

Nama : Salsabia Vebi Natasya  
NPM : 1194066  
Kelas : D4TI-3B

## 1 Cara menampilkan output

```
nama = "Salsabila Vebi"  
print "Hello",nama
```

## 2 Cara mendapatkan hostname system

```
import socket  
print(socket.gethostname())
```

## 3 Menulis data dictionary ke CSV

```
import csv  
  
with open('contacts.csv', mode='a') as csv_file :  
    # menentukan label  
    fieldnames = ['NO', 'NAMA', 'TELEPON']  
  
    # membuat objek writer  
    writer = csv.DictWriter( csv_file , fieldnames=fieldnames)  
  
    # menulis baris ke file CSV  
    writer.writeheader()  
    writer.writerow({'NO': '10', 'NAMA': 'Salsa', 'TELEPON': '0821194512'})  
    writer.writerow({'NO': '11', 'NAMA': 'Vebi', 'TELEPON': '08213218211'})  
  
print("Writing Done!")
```

## 4 cara mendapatkan key dan value dan menggabungkan dalam daftar

```
data = [{"name": "Anne", "followers": ["Brian"]}, {"name": "Cindy", "  
        followers": ["Brian", "G osh", "Anne"]}, {"name": "Dave", "followers": []}]
```

output : [{ "name": ["Brian"] , " follows":[" Anne","Cindy"]},...] etc ...

my code **from** now :

**from** operator **import** itemgetter

```
data = data = [{"name": "Anne", "followers": ["Brian"]}, {"name": "Cindy", "followers": ["Brian", "Gosh", "Anne"]}, {"name": "Dave", "followers": []}]
```

```
x = list(map(itemgetter('followers'), data))
y = list(map(itemgetter('name'), data))
print("name : " + str(x), " follow : " + str(y))
```

## 5 Cara menemukan semua kombinasi dari 3 kerangka data dan mengembalikannya sebagai daftar

```
>>> from datar.all import f, tibble, bind_cols, expand, nesting
>>>
>>> df1 = tibble(
...     name=["John", "Nick", "Eric"], job=["engineer", "architect", "
...     deisgner"]
... )
>>> df2 = tibble(
...     city=["London", "Montresor", "Esslingen"],
...     bigness=["captical", "villege", "town"],
... )
>>> df3 = tibble(
...     street=["street1", "street2", "street3"],
...     population=["high", "low", "average"],
... )
>>>
>>> df = bind_cols(df1, df2, df3)
>>> df >> expand(
...     nesting(f.name, f.job),
...     nesting(f.city, f.bigness),
...     nesting(f.street, f.population),
... )
```

## 6 Cara membuat list

```
list1 = ['kimia', 'fisika ', 1993, 2017]
list2 = [1, 2, 3, 4, 5 ]
list3 = ["a", "b", "c", "d"]
\end{lstlisting}
```

```
\section {Menggabungkan dua buah list}
\begin{lstlisting}
listone = [1, 2, 3]
listtwo = [4, 5, 6]
```

```
joinedlist = listone + listtwo
```

## 7 Cara mengakses nilai dalam list

```
list1 = ['fisika ', 'kimia', 1993, 2017]
list2 = [1, 2, 3, 4, 5, 6, 7 ]
```

```
print ("list1 [0]: ", list1 [0])
print ("list2 [1:5]: ", list2 [1:5])
```

## 8 Cara mendapatkan waktu Saat ini

```
import time;

localtime = time.localtime(time.time())
print "Waktu lokal saat ini :", localtime
```

## 9 Cara menggunakan Waktu selektif pada database python

```
df.sort_values('Submit Date').drop_duplicates(subset=['customer'], keep='
last')
```

## 10 Convert date ke datetime pada python

```
from datetime import date
from datetime import datetime

dt = datetime.combine(date.today(), datetime.min.time())
```

## 11 Cara menggunakan waktu Tick

```
import time;

ticks = time.time()
print "Berjalan sejak 12:00am, January 1, 1970:", ticks
```

## 12 Menggunakan fungsi create

```
def create_contact():
    clear_screen()
    with open(csv_filename, mode='a') as csv_file:
        fieldnames = ['NO', 'NAMA', 'TELEPON']
        writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

        no = input("No urut: ")
        nama = input("Nama lengkap: ")
        telepon = input("No. Telepon: ")

        writer.writerow({'NO': no, 'NAMA': nama, 'TELEPON': telepon})
        print("Berhasil disimpan!")

    back_to_menu()
```

## 13 Membuat fungsi search

```
def search_contact():
    clear_screen()
    contacts = []

    with open(csv_filename, mode="r") as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            contacts.append(row)

    no = input("Cari berdasarkan nomer urut> ")

    data_found = []
```

```

# mencari contact
indeks = 0
for data in contacts:
    if (data['NO'] == no):
        data_found = contacts[indeks]

    indeks = indeks + 1

if len(data_found) > 0:
    print("DATA DITEMUKAN: ")
    print(f>Nama: {data_found['NAMA']}")
    print(f"Telepon: {data_found['TELEPON']}")
else:
    print("Tidak ada data ditemukan")
    back_to_menu()

```

## 14 Membuat fungsi edit

```

def edit_contact():
    clear_screen()
    contacts = []

    with open(csv_filename, mode="r") as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            contacts.append(row)

    print("NO \t NAMA \t\t TELEPON")
    print("—" * 32)

    for data in contacts:
        print(f"{data['NO']} \t {data['NAMA']} \t {data['TELEPON']}")

    print("-----")
    no = input("Pilih nomer kontak> ")
    nama = input("nama baru: ")
    telepon = input("nomer telepon baru: ")

    # mencari contact dan mengubah datanya
    # dengan data yang baru

```

```

indeks = 0
for data in contacts:
    if (data['NO'] == no):
        contacts[indeks]['NAMA'] = nama
        contacts[indeks]['TELEPON'] = telepon
        indeks = indeks + 1

# Menulis data baru ke file CSV (tuliskan ulang)
with open(csv_filename, mode="w") as csv_file:
    fieldnames = ['NO', 'NAMA', 'TELEPON']
    writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
    writer.writeheader()
    for new_data in contacts:
        writer.writerow({'NO': new_data['NO'], 'NAMA': new_data['NAMA'], '
            TELEPON': new_data['TELEPON']})

back_to_menu()

```

## 15 Menggunakan fungsi delete

```

def delete_contact():
    clear_screen()
    contacts = []

    with open(csv_filename, mode="r") as csv_file:
        csv_reader = csv.DictReader(csv_file)
        for row in csv_reader:
            contacts.append(row)

    print("NO \t NAMA \t TELEPON")
    print("—" * 32)

    for data in contacts:
        print(f'{data["NO"]} \t {data["NAMA"]} \t {data["TELEPON"]}')

    print("-----")
    no = input("Hapus nomer> ")

    # mencari contact dan mengubah datanya
    # dengan data yang baru

```

```

indeks = 0
for data in contacts:
    if (data['NO'] == no):
        contacts.remove(contacts[indeks])
        indeks = indeks + 1

    # Menulis data baru ke file CSV (tuliskan ulang)
    with open(csv_filename, mode="w") as csv_file:
        fieldnames = ['NO', 'NAMA', 'TELEPON']
        writer = csv.DictWriter( csv_file , fieldnames=fieldnames)
        writer.writeheader()
        for new_data in contacts:
            writer.writerow({'NO': new_data['NO'], 'NAMA': new_data['NAMA'], '
                TELEPON': new_data['TELEPON']})

print("Data sudah terhapus")
back_to_menu()

```

## 16 Menggunakan Daftar Lists sebagai Tumpukan Stacks

```

>>> stack = [3, 4, 5]
>>> stack.append(6)
>>> stack.append(7)
>>> stack
[3, 4, 5, 6, 7]
>>> stack.pop()
7
>>> stack
[3, 4, 5, 6]
>>> stack.pop()
6
>>> stack.pop()
5
>>> stack
[3, 4]

```

## 17 membuat menu dari input pengguna

```

def functions():

```

```

max_length = int(input("how many products in your card ? : "))
select_function = input("press 1 to add product names to the menu or 2 to
    assign prices : ")
select_function = int(select_function)
products = []
while select_function == 1 and len(products) != max_length :
items = input("enter product name : ")
items = items.split ()
products.append(items)
if len(products) == max_length :
    select_function == 2
    price = []
    while select_function ==2 and len(price) != max_length :
items = input("enter product price : ")
items = items.split ()
price.append(items)
menu = dict(zip(products,price))
print(menu)

```

## 18 Cara menggunakan operator relasi sama dengan

```

lulus = raw_input("Apakah kamu lulus? [ya/tidak]: ")

if lulus == "tidak":
print("Kamu harus mengulang ujian")

```

## 19 Penggunaan If/Else

```

umur = input("Berapa umur kamu: ")

if umur >= 17:
print("Kamu boleh membuat KTP")
else:
print("Kamu belum boleh membuat KTP")

```

## 20 Parsing XML di Python



```

import xml.dom.minidom as minidom

def main():
    # gunakan fungsi parse() untuk me-load xml ke memori
    # dan melakukan parsing
    doc = minidom.parse("mahasiswa.xml")

    # Cetak isi doc dan tag pertamanya
    print doc.nodeName
    print doc.firstChild.tagName

if __name__ == "__main__":
    main()

```

## 21 Mengakses nilai dalam dict python

```

dict = {'Name': 'Salsa', 'Age': 20, 'Class': 'First'}
print ("dict ['Name']: ", dict['Name'])
print ("dict ['Age']: ", dict['Age'])

```

## 22 Cara menggunakan Lambda Expression

```

greeting = lambda name: print(f'Hello, {name}')
sapa = greeting
greeting("Andi")
sapa("Salsa")

```

## 23 Cara membuat dictionary

```

Test = {
    "nama": "Salsabila Vebi Natasya",
    "umur": 20,
    "hobi": ["coding", "membaca", "tidur"],
    "menikah": False,
    "sosmed": {
        "facebook": "salsa",
        "twitter": "@salsa"
    }
}

```

```
}  
}
```

## 24 Mengubah nilai item Dictionary

```
skill = {  
    "utama": "Tidur",  
    "lainnya": ["PHP", "Java", "HTML"]  
}
```

```
# Mencetak isi skill utama  
print(skill ["utama"])
```

```
# mengubah isi skill utama  
skill ["utama"] = "Mencetak gol"
```

```
# Mencetak isi skill utama  
print(skill ["utama"])
```

## 25 Mengambil panjang atau length Dictionary

```
books = {  
    "python": "Menguasai Python dalam 2028 jam",  
    "java": "Tutorial Belajar untuk Pemula",  
    "php": "Membuat aplikasi web dengan PHP"  
}
```

```
# mencetak jumlah data yang ada di dalam dictionary  
print("total buku: %d" % len(books))
```

## 26 Penggunaan Range pada perulangan for

```
for nomer in range(10):  
    print "mahasiswa—" + str(nomer)
```

## 27 mendapatkan 2 nilai per item

```

n = 1000000000
l = int(n/6)
f1 = lambda x: (6*x)-1
f3 = lambda x: (6*x)+1
primeCandidate = [f(i) for i in range(1,l+1) for f in (f1,f3)]

```

## 28 Cara menggunakan generator expressions

```

def squares(length):
    for n in range(length):
        yield n ** 2

```

## 29 Mendapatkan semua kemungkinan urutan boolean untuk panjang daftar tertentu

```

from itertools import product

[seq for seq in product((True, False), repeat=3)][1:-1]

```

## 30 Cara mendapatkan nama kolom kerangka data dari nilai dalam array numpy

```

threshold = .5

for j in range(loads.shape[1]):
    print(df.columns[loads[:,j]>threshold])

```

## 31 Menghapus duplikat dari korelasi matrix

```

def view_corr(df):
    df = df.unstack()
    corr_f = df.sort_values(kind="quicksort", ascending=False)
    corr_f = corr_f.dropna().drop_duplicates() # <<<----- here
    corr_f = corr_f[corr_f<1]
    print(corr_f[corr_f>0.10])

```

### 32 Pengambilan panjang Tuple

```
# Membuat Tuple
hari = ('Senin', 'Selasa', 'Rabu', 'Kamis', 'Jum\'at', 'Sabtu', 'Minggu')

# Mengambil panjang tuple hari
print("Jumlah hari: %d" % len(hari))
```

### 33 Perulangan Nested Loop

```
i = 2
while(i < 100):
    j = 2
    while(j <= (i/j)):
        if not(i%j): break
    j = j + 1
    if (j > i/j) : print(i, " is prime")
    i = i + 1

print("Good bye!")
```

### 34 Membuat list dengan besaran yang ditentukan

```
>>> Z = [Yes] * 3
>>> Z
[Yes, Yes, Yes]
```

### 35 Melakukan pengecekan list kosong

```
if not a:
print("List is empty")
```

### 36 Menulis data JSON ke file

```
import json
with open('data.json', 'w') as f:
    json.dump(data, f)
```

### 37 Menangani Exception

```
while True:
    try:
        x = int(input("Please enter a number: "))
        break
    except ValueError:
        print("Oops! That was no valid number. Try again ...")
```

### 38 Mengimport semua submodul

```
import * from sound.effect
```

### 39 menghapus item dari list

```
>>> a = [-1, 1, 66.25, 333, 333, 1234.5]
>>> del a[0]
>>> a
[1, 66.25, 333, 333, 1234.5]
>>> del a[2:4]
>>> a
[1, 66.25, 1234.5]
>>> del a[:]
>>> a
[]
```

### 40 menghapus seluruh variabel

```
>>> del a
```

### 41 Mengulang urutan secara terbalik

```
>>> for i in reversed(range(1, 10, 2)):
    print(i)
```

```
9
7
```

```
5
3
1
```

## 42 Perulangan if

```
a = 8
b = 10
if b > a:
    print("b lebih besar dari a")
```

## 43 Perulangan While

```
count = 0
while (count < 9):
    print ("The count is: ", count)
    count = count + 1

print ("Good bye!")
```

## 44 Perulangan For

```
angka = [1,2,3,4,5]
for x in angka:
    print(x)

buah = ["nanas", "apel", "jeruk"]
for makanan in buah:
    print ("Saya suka makan", makanan)
```

## 45 Penggunaan Variabel

```
nama = "John Doe"
print(nama)
```

```
#nilai dan tipe data dalam variabel dapat diubah
umur = 20 #nilai awal
```

```

print(umur)           #mencetak nilai umur
type(umur)           #mengecek tipe data umur
umur = "dua puluh satu" #nilai setelah diubah
print(umur)           #mencetak nilai umur
type(umur)           #mengecek tipe data umur

```

```

namaDepan = "Budi"
namaBelakang = "Susanto"
nama = namaDepan + " " + namaBelakang
umur = 22
hobi = "Berenang"
print("Biodata\n", nama, "\n", umur, "\n", hobi)

```

```

#contoh variabel lainnya
inivariabel = "Halo"
ini_juga_variabel = "Hai"
_inivariabeljuga = "Hi"
inivariabel222 = "Bye"

```

```

panjang = 10
lebar = 5
luas = panjang * lebar
print(luas)

```

#### 46 Membuat Instance Object

```

emp1 = Employee("Zara", 2000)
emp2 = Employee("Manni", 5000)

```

#### 47 Menginstall package menggunakan pip berdasarkan requirements.txt

```

pip install -r /path/to/requirements.txt

```

#### 48 Penggunaan dari \*args dan \*\*kwargs

```

# membuat fungsi dengan parameter *args
def kirim_sms(*nomer):
print nomer

```

```

# membuat fungsi dengan parameter **kwargs
def tulis_sms(**isi):
    print isi

# Pemanggilan fungsi *args
 kirim_sms(123, 888, 4444)

# pemanggilan fungsi **kwargs
 tulis_sms(tujuan=123, pesan="apa kabar")

```

#### 49 Cara memilih item secara acak dari daftar

```

import random

foo = ['battery', 'correct', 'horse', 'staple']
secure_random = random.SystemRandom()
print(secure_random.choice(foo))

```

#### 50 Import files dari folder berbeda

```

# some_file.py
import sys
# insert at 1, 0 is the script path (or '' in REPL)
sys.path.insert(1, '/path/to/application/app/folder')

import file

```

#### 51 operasi pada array menggunakan NumPy

```

import numpy as np
a = np.array([1, 2, 3])
f = np.array([1.1, 2.2, 3.3])
a*f

```

#### 52 Membuat DataFrame Menggunakan Pandas



```

import pandas as pd
data = {'kota' : ['semarang', 'semarang', 'semarang', 'bandung', 'bandung',
                 bandung],
        'tahun' : ['2016', '2017', '2018', '2016', '2017', '2018'],
        'populasi': [1.5, 2.1, 3.2, 2.3, 3.2, 4.5]}
frame = pd.DataFrame(data)
frame

```

### 53 Plotting dasar menggunakan Matplotlib

```

import numpy as np
import matplotlib.pyplot as plt
x = np.array ([1,2,3,4,5], float) # membuat nilai array sumbu x
y = np.array ([1,4,9,16,25], float) # membuat nilai array sumbu y
plt.plot(x,y) # menggunakan pylab untuk memplot x dan y
[<matplotlib.lines.Line2D object at 0x0000000006064CC0>]
plt. show() # menampilkan hasil plot pada layar

```

### 54 Plotting Histogram menggunakan Matplotlib

```

import matplotlib.pyplot as plt
plt.show()
data = np.random.normal(5., 3., 1000)
# membuat histogram dari data array

plt.hist(data)
(array([ 1., 13., 43., 146., 244., 251., 185., 91., 20., 6.]),
 array([ -6.05563989, -3.93179046, -1.80794103, 0.3159084 ,
 2.43975783, 4.56360726, 6.68745669, 8.81130612,
10.93515555, 13.05900498, 15.18285441])), <a list of 10 Patch objects>)
plt.xlabel("data")
<matplotlib.text.Text object at 0x000000000B6969B0>
plt.show()

```

### 55 Multiple Plotting Dalam Sebuah Canvas

```

import matplotlib.pyplot as plt
fig1 = plt.figure(1)
plt.subplot(211)
<matplotlib.axes._subplots.AxesSubplot object at 0x0000000005229320>

```

## 56 Plotting Data Dalam Suatu File

```

import matplotlib.pyplot as plt
data = np.loadtxt('databoehongan.txt')
plt.plot(data[:,0], data[:,1], 'bo')
[<matplotlib.lines.Line2D object at 0x0000000004A78E10>]
plt.xlabel('sumbu x')
<matplotlib.text.Text object at 0x0000000004913278>
plt.ylabel('sumbu y')
<matplotlib.text.Text object at 0x00000000049D7780>
plt.title('Plotting Data ASCII')
<matplotlib.text.Text object at 0x0000000004A087B8>
plt.xlim(0., 10.)
(0.0, 10.0)
plt.show()

```

## 57 Mencari rata-rata pada sebuah list

```
l = [15, 18, 2, 36, 12, 78, 5, 6, 9]
```

```

import statistics
statistics.mean(l)

```

## 58 Logging

```

import logging

logger = logging.getLogger()

def f():

```

```

try:

    flaky_func()

except Exception:

    logger.exception()

raise

```

## 59 perulangan while dengan inputan

```

a = int(input('Masukkan bilangan ganjil lebih dari 50: '))

while a % 2 != 1 or a <= 50:
    a = int(input('Salah, masukkan lagi: '))

print('Benar')

```

## 60 Menuliskan File

```

f = file("baru.txt", "w")
f.write("Baris pertama")
f.write("masih di baris pertama")
f.write("\n masuk ke baris kedua")
f.close()

```

## 61 Unzipping files

```

import zipfile
with zipfile.ZipFile( path_to_zip_file , 'r') as zip_ref :
    zip_ref.extractall( directory_to_extract_to )

```

## 62 Iterasi Pada Array

```

a = np.array ([1,2,8], int)
for x in a:
    print x

```

### 63 Penugasan Berganda pada progres Iterasi Pada Array

```
a = np.array ([[1,2], [3,4], [5,6]], float)
for (x, y) in a:
    print x * y
```

### 64 Menggunakan Fungsi Put pada Array

```
a = np.array ([0,1,2,3,4,5], float)
b = np.array ([6,7,8], float)
a.put ([0,3], b)
a
```

### 65 perulangan while untuk list

```
listKota = ['Jakarta', 'Surabaya', 'Depok', 'Bekasi', 'Solo', 'Jogjakarta',
            'Semarang', 'Makassar']

# bermain index
i = 0
while i < len(listKota):
    print(listKota[i])
    i += 1
```

### 66 Bilangan Acak

```
np.random.rand(15)
```

### 67 Date string ke date object

```
import datetime
datetime.datetime.strptime('24052010', "%d%m%Y").date()
datetime.date(2010, 5, 24)
```

## 68 Mencari dan mereplace elemen pada list

```
a=[1,2,3,1,3,2,1,1]
[4 if x==1 else x for x in a]
[4, 2, 3, 4, 3, 2, 4, 4]
```

## 69 Perbandingan string dengan case-sensitive

```
string1 = 'Hello'
string2 = 'hello'

if string1.casefold() == string2.casefold():
    print("The strings are the same (case insensitive)")
else:
    print("The strings are NOT the same (case insensitive)")
```

## 70 Membuat list kosong dengan besaran yang ditentukan

```
l = [None] * 10
l
[None, None, None, None, None, None, None, None, None, None]
```

## 71 Menentukan Beberapa Nilai Sekaligus

```
v = ('a', 2, True)
(x, y, z) = v
>>> x
'a'
>>> y
2
>>> z
True
```

## 72 Melakukan trigonometri dasar

```
>>> import math
>>> math.pi
```

```

3.1415926535897931
>>> math.sin(math.pi / 2)
1.0
>>> math.tan(math.pi / 4)
0.9999999999999999

```

### 73 filter membuat daftar elemen yang mengembalikan fungsi benar

```

number_list = range(-5, 5)
less_than_zero = list(filter(lambda x: x < 0, number_list))
print(less_than_zero)

```

### 74 Penggunaan Reduce

```

product = 1
list = [1, 2, 3, 4]
for num in list:
    product = product * num

```

### 75 Penggunaan Set

```

some_list = ['a', 'b', 'c', 'b', 'd', 'm', 'n', 'n']
duplicates = []
for value in some_list:
    if some_list.count(value) > 1:
        if value not in duplicates:
            duplicates.append(value)
print(duplicates)

```

### 76 Ternary Operators

```

nice = True
personality = ("mean", "nice")[nice]
print("The cat is ", personality)

```

## 77 Penggunaan Map

```
def multiply(x):  
    return (x*x)  
def add(x):  
    return (x+x)  
funcs = [multiply, add]  
for i in range(5):  
    value = list(map(lambda x: x(i), funcs))  
    print(value)
```

## 78 Menggunakan main loop

```
if __name__ == "__main__":  
    while True:  
        show_menu()
```

## 79 perulangan while dengan break

```
listKota = [  
    'Jakarta', 'Surabaya', 'Depok', 'Bekasi', 'Solo',  
    'Jogjakarta', 'Semarang', 'Makassar'  
]  
  
kotaYangDicari = input('Masukkan nama kota yang dicari: ')  
  
i = 0  
while i < len(listKota):  
    if listKota[i].lower() == kotaYangDicari.lower():  
        print('Ketemu di index', i)  
        break  
  
    print('Bukan', listKota[i])  
    i += 1
```

## 80 Mencetak exception dengan Python

```
except Exception as e: print(e)
```

## 81 Menggunakan Fungsi Fill

```
a = np.array([1,2,8], float)
>>> a
array([ 1.,  2.,  8.])
a.fill(6)
>>> a
array([ 6.,  6.,  6.])
```

## 82 Perintah In

```
>>> a = np.array([[1,2,3], [4,5,6],[1,2,4]], float)
>>> 2 in a
True
>>> 0 in a
False
```

## 83 Fungsi Zeros dan ones

```
>>> np.ones((2,3), dtype = float)
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.]])
>>> np.zeros(7, dtype = int)
array([0, 0, 0, 0, 0, 0, 0])
```

## 84 mengetahui nilai terendah dan tertinggi dari elemen – elemen dalam suatu array:

```
>>> a = np.array([1,2,8], float)
>>> a.min()
1.0
>>> a.max()
8.0
```

## 85 mengurutkan elemen – elemen dalam array



```
>>> a = np.array([5,1,4,-2,0], float)
>>> sorted(a)
[-2.0, 0.0, 1.0, 4.0, 5.0]
>>> a.sort()
>>> a
array([-2., 0., 1., 4., 5.])
```

## 86 Memotong array sesuai dengan rentang nilai tertentu

```
a = np.array ([6,2,5,-1,0], float)
a.clip (0,4)
array([ 4., 2., 4., 0., 0.]
```

## 87 Mengekstraksi elemen – elemen unik dalam array

```
a = np.array ([1,1,1,2,2,3,4,4,4,4,5,5,5,5,5], float)
np.unique(a)
array([ 1., 2., 3., 4., 5.]
```

## 88 Mengekstraksi elemen – elemen diagonal dalam suatu array

```
>>> a = np.array([[1,2,3], [4,5,6], [7,8,9]], float)
>>> a.diagonal()
array([ 1., 5., 9.]
```

## 89 Membandingkan suatu array dengan nilai tunggal

```
>>> a = np.array([1,2,8], float)
>>> a > 2
array([False, False,  True], dtype=bool)
```

## 90 Menerapkan broadcast dalam fungsi where

```
>>> np.where(a > 0, 3, 2)
array([3, 3, 2])
```

## 91 Penggunaan Fungsi Nonzero

```
>>> a = np.array([[0,1], [1,0]], float)
>>> a.nonzero()
(array([0, 1], dtype=int64), array([1, 0], dtype=int64))
```

## 92 Memeriksa keberadaan nilai NaN (not a number) dan bilangan hingga (finite) dalam suatu array

```
>>> a = np.array([1, np.NaN, np.Inf], float)
>>> a
array([ 1., nan, inf])
>>> np.isnan(a)
array([False,  True, False], dtype=bool)
>>> np.isfinite(a)
array([ True, False, False], dtype=bool)
```

## 93 Melakukan Perkalian Titik

```
>>> a = np.array([1,2,0], float)
>>> b = np.array([1,2,8], float)
>>> np.dot(a,b)
5.0
```

## 94 Menghitung Determinan dari suatu matriks

```
>>> a = np.array([[1,2,8], [1,2,0], [2,2,0]], float)
>>> a
array([[ 1.,  2.,  8.],
       [ 1.,  2.,  0.],
       [ 2.,  2.,  0.]])
>>> np.linalg.det(a)
-15.999999999999998
```

## 95 Mengetahui polinom mana yang menghasilkan akar – akar

```
>>> np.poly([-1, 1, 1, 10])
array([ 1., -11.,  9., 11., -10.] )
```

## 96 Menghapus duplikasi pada list

```
>>> t = [1, 2, 3, 1, 2, 5, 6, 7, 8]
>>> t
[1, 2, 3, 1, 2, 5, 6, 7, 8]
>>> list(set(t))
[1, 2, 3, 5, 6, 7, 8]
>>> s = [1, 2, 3]
>>> list(set(t) - set(s))
[8, 5, 6, 7]
```

## 97 Menentukan Bilangan Ganjil Genap

```
number = int(input("masukkan bilangan:"))

if number % 2 == 0:
    print("%i adalah bilangan genap" % number)
else:
    print("%i adalah bilangan ganjil" % number)
```

## 98 Mengganti Nama File

```
import os
try:
    os.rename('absen.txt', 'daftar-hadir.txt')
    print "Nama file sudah diubah.."
except (IOError, OSError), e:
    print "proses error karena : ", e
```

## 99 Penggunaan “elif” pada “if”

```
print "Masukkan dua buah angka.."
print "Dan Anda akan check hubungan kedua angka tersebut"
angka1 = raw_input("Masukkan angka pertama : ")
angka1 = int(angka1)
angka2 = raw_input("Masukkan angka kedua : ")
angka2 = int(angka2)
if angka1 == angka2 :
```

```

print "%d sama dengan %d" % (angka1, angka2)
elif angka1 != angka2 :
print "%d tidak sama dengan %d" % (angka1, angka2)
elif angka1 < angka2 :
print "%d kurang dari %d" % (angka1, angka2)
elif angka1 > angka2 :
print "%d lebih dari %d" % (angka1, angka2)
elif angka1 <= angka2 :
print "%d kurang dari sama dengan %d" % (angka1, angka2)
elif angka1 >= angka2 :
print "%d lebih dari sama dengan %d" % (angka1, angka2)

```

## 100 Membuat teks yang rata kiri dan/atau rata kanan dalam string

```

>>> s = "apple".ljust(10) + "orange".rjust(10) + "\n" \
... + "grape".ljust(10) + "pear".rjust(10)
>>> print s
apple orange
grape pear

```