



**STMA**  
TRISAKTI

# Buku panduan

Penerapan Teknologi Dalam  
Pembelajaran Matematika

Fanny Novika, M.Si

---

## **Pendahuluan**

Teknologi informasi berkembang dengan cepat yang dapat dibuktikan dengan peningkatan Indeks Pembangunan Teknologi Informasi dan Komunikasi (IP-TIK) di Indonesia. Menurut survey BPS Indonesia, skala IP-TIK Indonesia tahun 2018 sebesar 5,07 yang meningkat dibandingkan IP-TIK tahun 2017 sebesar 4,96. Bagian subindeks yang tertinggi adalah bagian keahlian sebesar 5,76 yang diikuti subindeks akses dan infrastruktur sebesar 5,34 dan subindeks penggunaan sebesar 4,45. Keahlian dinilai sebagai subindeks tertinggi oleh karena itu sangat diperlukan untuk peningkatan keahlian di bidang teknik informasi.

Sangat disayangkan bahwa IP-TIK seluruh provinsi tersebar di dua kategori yaitu kategori sedang dan rendah. Hal ini menunjukkan bahwa tidak ada provinsi yang tertinggal pada kategori sangat rendah dan juga belum ada provinsi yang mencapai IP-TIK kategori tinggi. Empat dari 34 provinsi mengalami pergeseran kategori dari rendah ke sedang yaitu Jawa tengah, Sumatera Barat, Papua Barat dan Kalimantan Selatan. Peningkatan ini baru terlihat pada empat provinsi akan lebih baik apabila lebih banyak provinsi yang mengalami peningkatan dengan memperhatikan subindeks tertinggi yaitu keahlian.

Semakin tinggi teknologi yang dapat dihasilkan manusia, maka efisiensi dapat dilakukan terutama dalam pengajaran, salah satunya adalah pada bidang ilmu matematika. Ada baiknya peningkatan teknologi ini dapat diterapkan dalam pembelajaran, terutama mata pelajaran wajib yaitu matematika. Ada momok tersendiri saat mempelajarinya. Ada beberapa siswa yang cenderung mengalami respon emosional saat mempelajari matematika. Rasa takut, cemas dan kekhawatiran untuk sulit menjawab soal atau mendiskusikan matematika. Penelitian mengemukakan bahwa terdapat hubungan antara kecemasan matematika dengan hasil belajar siswa. Guru mempunyai peran terhadap keberhasilan belajar siswa agar dapat menciptakan suasana belajar yang kondusif.

Metode-metode pembelajaran terkini memanfaatkan teknologi yang sudah berkembang. Teknologi diperkenalkan guna menambah motivasi dan minat belajar. Selain itu, teknologi dapat membantu siswa untuk dapat bersaing di dunia global dan tidak tertinggal dengan negara lain. Salah satunya adalah penggunaan suatu aplikasi perhitungan matematika.

Python amat sangat mudah untuk dibaca. Sebagai interpreted language (bahasa pemrograman yang tidak perlu dikompilasi), Python tidak mengubah kodenya untuk menjadi terbaca oleh komputer. Bahasa ini juga merupakan bahasa pemrograman tujuan umum tingkat tinggi. Para pengembang mendesainnya untuk menjadi bunglon dari dunia pemrograman. Python bertujuan untuk menghasilkan kode yang lebih jelas dan lebih logis tidak hanya untuk proyek skala kecil tetapi juga untuk yang lebih besar. Pengkodean ini menggunakan kecerdasan buatan berupa *neural network*. Neural network adalah sistem yang terdiri dari algoritma yang didasarkan pada otak manusia. Pembuat jaringan pada python sangat canggih dan dapat digunakan dengan terlebih dahulu menganalisis contoh sebagai bentuk simulasi bahasa pemrograman.

Python dapat dimanfaatkan dalam penerapan pembelajaran matematika. Python adalah bahasa pemrograman interpretatif yang berfokus pada keterbacaan kode. Kode-kode yang digunakan pada python merupakan kode-kode yang jelas, lengkap dan mudah dipahami. Secara umum, bahasa python berbentuk pemrograman berorientasi objek, imperative dan fungsional. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak sebagai platform sistem operasi. Beberapa keunggulan yang dimiliki python yaitu:

- a. Memiliki koleksi modul yang siap pakai untuk keperluan pengolahan data atau pengembangan aplikasi
- b. Struktur bahasa dalam Python jelas, sederhana dan mudah dipelajari
- c. Berorientasi objek
- d. Memiliki sistem pengelolaan memori otomatis
- e. Mudah mengembangkan modul-modul baru yang dapat digunakan dalam bahasa Python (Enterprise, 2017)

Penerapan aplikasi pembelajaran matematika yang dapat diterapkan diantaranya adalah:

1. Membuat bahasa pemrograman operasi aljabar sederhana
2. Membuat bahasa pemrograman akar persamaan kuadrat
3. Membuat bahasa pemrograman operasi matriks
4. Membuat bahasa pemrograman perhitungan luas bidang dua dimensi
5. Membuat bahasa pemrograman perhitungan volume bidang tiga dimensi

6. Membuat bahasa pemrograman penerapan ekonomi matematika (menghitung besaran bunga, pertumbuhan dan peluruhan uang)
7. Membuat bahasa pemrograman penyajian data

Pada modul ini akan dibahas mengenai penerapan Python yang dapat digunakan dalam implementasi pengajaran matematika.

## Membuat Bahasa Pemrograman Operasi Aljabar Sederhana

Untuk membuat operasi aljabar, pertama-tama lakukan konsep pemrograman terlebih dahulu. Langkah konsep pemrograman operasi aljabar sederhana adalah:

1. Mengidentifikasi fungsi operasi aljabar yang akan diterapkan (dalam hal ini adalah operasi penjumlahan, pengurangan, perkalian dan pembagian).
2. Membentuk menu operasi sebagai arahan bagi sistem pemrograman. Identifikasikan kode bahwa 1 adalah operasi penjumlahan, 2 adalah operasi pengurangan, 3 adalah operasi perkalian dan 4 adalah operasi pembagian.
3. Membuat pembacaan untuk input operasi yang akan dipilih dengan menu "choice"
4. Membuat bahasa pemrograman untuk mengaktifkan operasi aljabar dengan menggunakan perintah "if"

Bahasa pemrograman untuk kegiatan tersebut adalah:

```
# fungsi penjumlahan
def add(x, y):
    return x + y
# fungsi pengurangan
def subtract(x, y):
    return x - y
# fungsi perkalian
def multiply(x, y):
    return x * y
# fungsi pembagian
def divide(x, y):
    return x / y
# menu operasi
print("Pilih Operasi.")
print("1.Jumlah")
print("2.Kurang")
print("3.Kali")
print("4.Bagi")
# Meminta input dari user
choice = input("Masukkan pilihan(1/2/3/4): ")
num1 = int(input("Masukkan bilangan pertama: "))
num2 = int(input("Masukkan bilangan kedua: "))
```

```

if choice == '1':
print(num1,"+",num2,"=", add(num1,num2))
elif choice == '2':
print(num1,"-",num2,"=", subtract(num1,num2))
elif choice == '3':
print(num1,"*",num2,"=", multiply(num1,num2))
elif choice == '4':
print(num1,"/",num2,"=", divide(num1,num2))
else:
print("Input salah")

```

## Membuat Bahasa Pemrograman Akar Persamaan Kuadrat

Untuk membuat bahasa pemrograman akar persamaan kuadrat, tahapan pemrograman yang dapat dilakukan adalah:

1. Menginput konstanta persamaan kuadrat dalam bentuk  $ax^2 + bx + c$
2. Merumuskan nilai diskriminan  $d = b^2 - 4ac$
3. Mencari akar persamaan pertama, yaitu  $x_1 = \frac{-b+\sqrt{d}}{2a}$
4. Mencari akar persamaan kedua, yaitu  $x_2 = \frac{-b-\sqrt{d}}{2a}$
5. Mendefinisikan input konstanta kuadrat
6. *Print* hasil pada lembar jawaban

Bahasa pemrograman operasi tersebut adalah

```

def rumusakarkuadrat (a, b, c):
a= float(a); b=float (b); c=float(c)
d=b**b-4*a*c
x1=(-b+akar(d))/2*a
x2=(-b-akar(d))/2*a
akar =(x1, x2)
return hasil

a=input("Masukkan konstanta kuadrat:")
b=input("Masukkan konstanta x:")
c=input("Masukkan konstanta")

print rumusakarkuadrat(a,b,c)

```

## Membuat Bahasa Pemrograman Operasi Matriks

Untuk membuat bahasa pemrograman operasi matriks, langkah-langkah yang dapat dilakukan adalah:

1. Mendefinisikan matriks yang akan dioperasikan
2. Membuat bahasa pemrograman penjumlahan matriks
3. Membuat bahasa pemrograman pengurangan matriks
4. Membuat bahasa pemrograman perkalian matriks

```
mat1 = [  
    [5, 0],  
    [2, 6],  
]
```

```
mat2 = [  
    [1, 0],  
    [4, 2],  
]
```

```
mat3 = []
```

```
#penjumlahan matriks
```

```
for x in range(0, len(mat1)):  
    for y in range(0, len(mat1[0])):  
        print (mat1[x][y] + mat2[x][y], end=' '),  
    print
```

```
#pengurangan matriks
```

```
for x in range(0, len(mat1)):  
    for y in range(0, len(mat1[0])):  
        print (mat1[x][y] - mat2[x][y], end=' '),  
    print
```

```
#perkalian matriks
```

```
for x in range(0, len(mat1)):  
    row = []  
    for y in range(0, len(mat1[0])):  
        total = 0  
        for z in range(0, len(mat1)):  
            total = total + (mat1[x][z] * mat2[z][y])  
        row.append(total)
```

```

mat3.append(row)

for x in range(0, len(mat3)):
    for y in range(0, len(mat3[0])):
        print (mat3[x][y], end=' ')
    print ()

```

## Membuat Bahasa Pemrograman Perhitungan Luas Bidang Dua Dimensi

Luas dua dimensi yang akan dibuat bahasa pemrogramannya adalah:

- a. Segiempat
- b. Segitiga
- c. Trapesium
- d. Jajargenjang

```

#menghitung Luas segitiga
a = float(input("Masukkan panjang alas: "))
t = float(input("Masukkan tinggi segitiga: "))
luassegitiga = 0.5*a*t
print("Luas segitiga adalah : "+ str(luassegitiga))

#menghitung Luas segiempat
p = float(input("Masukkan panjang segiempat: "))
l = float(input("Masukkan lebar segiempat: "))
luassegieempat = p*t
print("Luas segiempat adalah : "+ str(luassegieempat))

#menghitung trapesium
sa = float(input("Masukkan sisi sejajar atas trapesium: ")
)
sb = float(input("Masukkan sisi sejajar bawah trapesium: ")
))
ti = float(input("Masukkan tinggi trapesium: "))
luastrapesium = 0.5*(sa+sb)*ti
print("Luas trapesium adalah : "+ str(luastrapesium))

#menghitung jajargenjang
aj = float(input("Masukkan alas jajar genjang: "))
tj = float(input("Masukkan tinggi jajar genjang: "))
luasjajargenjang = at*tj

```



```
print("Luas jajargenjang adalah : "+ str(luasjajargenjang)
)
```

## Membuat Bahasa Pemrograman Perhitungan Volume Bidang Tiga Dimensi

Volume bidang tiga dimensi yang akan dihitung adalah:

- a. Volume kubus
- b. Volume balok
- c. Volume tabung
- d. Volume kerucut
- e. Volume bola

```
#menghitung volume kubus
```

```
s = int(input('masukan sisi kubus: '))
volumekubus = s*s*s
print("Volume kubus adalah : "+ str(volumekubus))
```

```
#menghitung volume balok
```

```
p = int(input('masukan panjang balok: '))
l = int(input('masukan lebar balok: '))
t = int(input('masukan tinggi balok: '))
volumebalok = p*l*t
print("Volume balok adalah : "+ str(volumebalok))
```

```
#menghitung volume tabung
```

```
r = int(input('masukan jari-jari tabung: '))
h = int(input('masukan tinggi tabung: '))
volumetabung = 3.14*r*r*h
print("Volume tabung adalah : "+ str(volumetabung))
```

```
#menghitung volume kerucut
```

```
rk = int(input('masukan jari-jari kerucut: '))
hk = int(input('masukan tinggi kerucut: '))
volumekerucut = (1/3)*3.14*rk*rk*hk
print("Volume kerucut adalah : "+ str(volumekerucut))
```

```
#menghitung volume bola
```

```
rb = int(input('masukan jari-jari bola: '))
volumebola = (4/3)*3.14*rb*rb*rb
print("Volume kerucut adalah : "+ str(volumebola))
```

## Membuat Bahasa Pemrograman Penerapan Ekonomi Matematika (Menghitung Tingkat Suku Bunga, Pertumbuhan Dan Peluruhan Uang)

Penerapan ekonomi matematika sangat luas pengaplikasiannya. Beberapa problematika soal-soal ekonomi dapat diaplikasikan dengan berbagai perumusan dalam pengkodean bahasa pemrograman. Sebagai dasar, berikut merupakan masalah sederhana yang akan dicari langkah pemrogramannya:

1. Mencari akumulasi tabungan pada suku bunga tertentu dengan bunga efektif (konsep *future value*)

Nilai akumulasi untuk tabungan mencari nilai masa yang akan datang dengan asumsi tidak ada penambahan biaya adalah

$$A = C * (1 + i)^n$$

dengan:

A: Nilai akumulasi tabungan

C: Nilai tabungan awal

i: tingkat suku bunga

n: lamanya tabungan disimpan (dalam tahun)

Bahasa pemrograman dalam mencari akumulasi tersebut adalah:

```
#menghitung besaran tabungan
C = int(input('masukan nilai tabungan awal: '))
i = int(input('masukan nilai bunga efektif: '))
n = int(input('masukan nilai lama tabungan (tahun): '))
Akumulasi= C*(1+i)**n
print("Besarnya tabungan anda adalah : "+ str(Akumulasi))
```

2. Mencari akumulasi *past value* dengan suku bunga efektif

Nilai akumulasi untuk mencari nilai pada masa masa lalu dengan asumsi tidak ada nilai lagi yang dilakukan adalah:

$$P = \frac{C}{(1 + i)^n}$$

dengan:

P: Nilai akumulasi masa lalu

C: Nilai tabungan awal

i: tingkat suku bunga

n: lamanya nilai masa lalu akan diprediksi (dalam tahun)

Bahasa pemrograman untuk mencari nilai past value adalah:

```
#menghitung besaran past value
C = int(input('masukan nilai tabungan awal: '))
i = int(input('masukan nilai bunga efektif: '))
n = int(input('masukan nilai lama tabungan (tahun): '))
pastvalue= C*((1+i)**(-1*n))
print("Besarnya tabungan anda adalah : "+ str(pastvalue))
```

## Membuat Bahasa Pemrograman Penyajian Data

Pada aplikasi Python, dapat juga digunakan untuk melakukan pemrograman penyajian data statistika. Ilustrasi yang disajikan pada modul ini adalah dengan melakukan olah data sederhana, yaitu mencari nilai:

- a. Rata-rata
- b. Median
- c. Modus
- d. Ragam
- e. Standar deviasi

Langkah yang digunakan adalah dalam menyusun bahasa pemrograman tersebut adalah:

1. Melakukan instalasi program untuk menghitung operasi statistika
2. Membuat perintah untuk memasukkan data ke sistem
3. Membuat fungsi rata-rata, median, modus, ragam dan standar deviasi
4. Menampilkan hasil pengolahan data

Bahasa pemrograman untuk mencari rata-rata adalah:

```
import os
import collections
import matplotlib.pyplot as plt
os.system("cls")
#memasukkan data
banyaknyadata = int(input("masukan nilai banyaknya data: "))
data = input("masukan data:")
x= data.split(",")
nilaidata = [int(z) for z in x]

#mencari nilai rata-rata
rata2 = sum(nilaidata)/banyaknyadata
print ("nilai rata-rata dari data anda adalah: ", float(rata2))
```

Bahasa pemrograman untuk mencari median adalah:

```
#mencari median
os.system("cls")
nilaidata.sort()
if len(nilaidata) %2 == 0
    a=int((len(nilaidata)/2))
    b=(nilaidata[a-1]+nilaidata[a])/2
    median= str(b)

else
    a=int((len(nilaidata)+1)/2)
    median=str(nilaidata[a-1])
```

Bahasa pemrograman untuk mencari modus adalah:

```
#mencari modus
os.system("cls")
modus=max(set(nilaidata), key=nilaidata.count)
print ("nilai modus dari data anda adalah: ", float(modus))
```

## **Daftar Pustaka**

- Badan Pusat Statistika Indonesia. 2018. Indeks Pembangunan Teknologi, Informasi, dan Komunikasi. Jakarta: BPS-Statistics Indonesia
- Enterprise, Jubilee. 2017. Otodidak Pemrograman Phyton. Jakarta: Elex Media Komputindo
- Irsyad, Rahadian. 2018. Penggunaan Python Web Framework Flask Untuk Pemula. OSF Preprint. oi: [10.31219/osf.io/t7u5r](https://doi.org/10.31219/osf.io/t7u5r)
- Kadir, Abdul. 2010. Logika Pemrograman Phyton. Jakarta: PT Elex Media Komputindo