

**PRAKTIKUM**  
**ALGORITMA DAN STRUKTUR DATA**  
**MODUL 11 (QUIZ)**



**Nama : NICKY JULYATRIKA SARI**

**NIM : L200200101**

**PROGRAM STUDI**  
**INFORMATIKA**  
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**  
**UNIVERSITAS MUHAMMADIYAH SURAKARTA**  
**TAHUN 2021/2022**

1. Buatlah fungsi regex untuk mengecek sebuah inputan yang akan menampilkan nilai True apabila inputan tersebut berupa email dan false apabila tidak sesuai format email

Ex:

fajri@gmail.com => True

fajri@yahoo.co.id => True

fajri@com => False

```
1.py - C:\Users\asus\Downloads\1.py (3.8.6)
File Edit Format Run Options Window Help
#Nicky Julyatrika Sari NIM L200200101
#QUIZ
#Nomer 1

import re

regex = re.compile(r'([A-Za-z0-9]+[._-])*[A-Za-z0-9]+@[A-Za-z0-9-]+(\.[A-Z|a-z]{2,})+')

def check(email):
    if re.search(regex, email):
        print("True")
    else:
        print("False")

email = input('Input Email : ')
check(email)
```

```
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [M8C v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\asus\Downloads\1.py =====
Input Email : nicky@yahoo.co.id
True
>>>
===== RESTART: C:\Users\asus\Downloads\1.py =====
Input Email : nicky@gmail.com
True
>>>
===== RESTART: C:\Users\asus\Downloads\1.py =====
Input Email : nicky@com
False
>>>
```

2. Buat fungsi regex untuk mengecek email ums

ex :

fajri@ums.ac.id => True

fajri@uns.ac.id => False

fajri@student.ums.ac.id => True

fajri@ums.com => False

```
quiz no2.py - D:\MATKUL SMT 4\Praktikum ASD\quiz\quiz no2.py (3.8.6)
File Edit Format Run Options Window Help
#Nicky Julyatrika Sari NIM L200200101
#QUIZ
#Nomer 2

import re

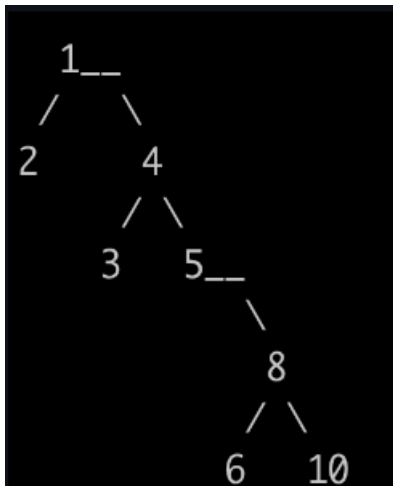
email = (input("Masukkan Email : "))
cocok = re.search("ums[^\s]+[^\s]+\.[^\s]+", email)
if cocok:
    print("True")
else:
    print("False")
```

```

Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum ASD\quiz\quiz no2.py =====
Masukkan Email : nicky@ums.ac.id
True
>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum ASD\quiz\quiz no2.py =====
Masukkan Email : nicky@ums.com
False
>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum ASD\quiz\quiz no2.py =====
Masukkan Email : nicky@student.ums.ac.id
True
>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum ASD\quiz\quiz no2.py =====
Masukkan Email : nicky@uns.ac.id
False
>>> |

```

3. Buatlah BTree seperti gambar dibawah ini :



```

*quiz no3.py - D:/MATKUL SMT 4/Praktikum ASD/quiz/quiz no3.py (3.8.6)*
File Edit Format Run Options Window Help
#Nicky Julyatrika Sari NIM L200200101
#QUIZ
#Nomer 3

from binarytree import Node
root = Node(1)
root.left = Node(2)
root.right = Node(4)
root.right.left = Node(3)
root.right.right = Node(5)
root.right.right.right = Node(8)
root.right.right.right.left = Node(6)
root.right.right.right.right = Node(10)

print(root)

```

```

Python 3.8.6 Shell*
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\MATKUL SMT 4\Praktikum ASD\quiz\quiz no3.py =====
1
 / \
2   4
 / \ \
3  5 8
   \ \
   6 10
>>> |

```

4. Tampilkan data btree dari no 3 dengan metode (hasil dan penjelasan | BUKAN PROGRAM) :

a. pre order

Hasil : [ 1, 2, 4, 3, 5, 8, 6, 10]

Penjelasan : pengurutan dimulai dari Node paling atas(root), kemudian dilanjutkan ke sub pohon sebelah kiri dan diikuti sub pohon sebelah kanan.

Proses tersebut diulangi sampai ke level bawah dan berhenti jika menemukan akhir dari simpul.

b. in order

Hasil : [2, 1, 3, 4, 5, 6, 8, 10]

Penjelasan : pengurutan dimulai dari mengunjungi simpul sub pohon sebelah kiri, lalu mengunjungi simpulnya. Kemudian dilanjutkan dengan mengunjungi sub pohon bagian kanan. Proses tersebut diulangi sampai ke level bawah dan berhenti jika menemukan akhir dari simpul.

c. post order

Hasil : [2, 3, 6, 10, 8, 5, 4, 1]

Penjelasan : pengurutan dimulai dari sub pohon sebelah kiri, kemudian ke sub pohon sebelah kiri bagian bawah. Pada post order, simpul akhirnya dikunjungi paling akhir. Alur pengerjaan post order adalah menelusuri bagian kiri, dilanjutkan ke bagian kanan dari bawah ke atas, dan yang terakhir ke simpul akar.