



TATA TERTIB UJIAN

- ✓ Mahasiswa **wajib** mengisi Pakta Integritas (Tata Tertib dan Kode Etik Ujian) di SCeLE agar dapat mengikuti ujian. Ujian yang dikerjakan tanpa mengisi Pakta tersebut akan mendapat **nilai 0**.
- ✓ **Dokumen soal dan tempat pengumpulan jawaban** terdapat pada Google Form yang tautannya akan diberikan di SCeLE.
- ✓ Mahasiswa diminta menuliskan jawaban (**tulis tangan dengan jelas dan rapi**) di atas kertas, lalu **di-scan/foto dan diunggah**. Jangan lupa tuliskan nama, NPM, dan kelas di berkas jawaban. Berkas jawaban terkumpul dalam **satu dokumen .pdf**. Gunakan aplikasi Adobe Scan, CamScanner, dan sejenisnya untuk *scan* dan konversi. **Dokumen jawaban selain format .pdf akan diberikan penalti 10 poin**. Konsekuensi terhadap penilaian dari tulisan yang tidak terbaca jelas ditanggung sepenuhnya oleh mahasiswa.
- ✓ Berkas jawaban menggunakan format penamaan: **Kelas_NPM_Nama.pdf** (Contoh: C_12345_GuidoRossum.pdf). **Kesalahan penulisan NPM maupun Kelas akan diberikan penalti 10 poin**.
- ✓ Durasi total 3 jam dengan rincian **2.5 jam (150 menit) untuk mengerjakan dan menulis jawaban + 0.5 jam (30 menit) untuk scan/foto dan mengunggah jawaban di Google Form**. Jadi, Anda diharapkan dapat mengunggah jawaban tepat waktu, yakni pada durasi **pukul 08.00 - 11.00**.
- ✓ Anda dapat mengunggah jawaban beberapa kali di Google Form, tetapi **hanya jawaban yang terakhir saja yang akan dinilai**.
- ✓ Diberikan **10 menit buffer time** dengan **penalti 20% dari total nilai**. Jadi, Anda masih dapat mengunggah jawaban hingga pukul 11.10 dengan konsekuensi penalti nilai. Setelah itu, Google Form akan otomatis tertutup, sehingga **jika terlambat lebih dari 10 menit, maka ujian Anda akan mendapat nilai 0**.
- ✓ Ujian bersifat *open all*.
- ✓ Mahasiswa **HARUS bekerja secara mandiri** (tidak bekerja sama dengan mahasiswa lain melalui media apapun). Jika ditemukan kecurangan akan diberikan penalti dengan **nilai akhir E**.
- ✓ Mahasiswa **HARUS mengerjakan tanpa menggunakan bantuan Python Interpreter** (baik menggunakan IDLE, Visual Studio Code, maupun IDE lainnya).
- ✓ Tiap jawaban wajib disertai **penjelasan/proses pengerjaan**. Jawaban tanpa penjelasan **tidak dinilai (nilai poin tersebut = 0)**.
- ✓ Kami tidak menerima pertanyaan tentang soal selama ujian berlangsung. Jika ada soal yang dirasa kurang jelas, **silakan menuliskan asumsi Anda** sepanjang tidak bertentangan dengan soal.

Bagian I. Isian

Bacalah soal dengan teliti. Tuliskan jawaban Anda pada lembar jawaban. Terdapat 20 soal dengan bobot penilaian per soal sebesar 3 poin, sehingga total nilai maksimum adalah 60 poin.

1. Perhatikan program berikut ini:

```
#Masukkan NPM anda
score = input('Masukkan NPM anda: ')
mystery = 0

for chara in score:

    if int(chara) == 9:
        break
    if int(chara) % 2 > 0:
        continue
    else:
        mystery += int(chara)

print(mystery)
```

Apa *output* yang dihasilkan dari program tersebut jika **score** diisi dengan NPM Anda? Berikan penjelasan alur dari program tersebut!

Jawaban:

Dengan NPM: **1106022755**, maka output dari program tersebut adalah: **10**

Program ini **menjumlahkan seluruh angka genap yang ada pada NPM, namun bila ada angka 9 maka program akan berhenti menghitung.**

2. Perhatikan program berikut ini:

```

n = 5
output = ''

for i in range(0,n):

    output += str(n)
    n += 1

print(output)

```

Apa *output* yang dihasilkan dari program di atas? Berikan penjelasan alur dari program tersebut!

Jawaban:

Output: **56789**

range hanya didefinisikan sekali menggunakan nilai **n = 5**

Hal ini membuat kita akan mendapatkan angka 5, 6, 7, 8, 9 dalam iterasi for **i in range (0, n)**

3. Berapa **nilai terkecil dari counter** yang dibutuhkan program di bawah ini, sehingga *output* yang dihasilkan adalah "**Nilai x = 1 dan y = 1**"? Jelaskan juga alur program berikut ketika diberikan nilai **counter terkecil** yang Anda gunakan!

```

x, y = 1, -1

#TODO Implement
counter =

for index in range(0,counter):

    if x >= y:
        x, y = y, x
    elif x < y:
        x = x + 1
    else:
        continue

print('Nilai x = {} dan y = {}'.format(x,y))

```

Jawaban:

Kita akan melakukan iterasi tanpa mengetahui batasannya terlebih dahulu:

Iterasi 1: $x = 1, y = -1$, masuk klausa `if x >= y`. Hasil: $x = -1, y = 1$

Iterasi 2: $x = -1, y = 1$, masuk klausa `if x < y`. Hasil: $x = 0, y = 1$

Iterasi 3: $x = 0, y = 1$, masuk klausa `if x < y`. Hasil: $x = 1, y = 1$

Karena pada Iterasi 3 kita sudah mendapatkan hasil yang diinginkan, maka nilai **counter** yang kita butuhkan adalah **3**

4. Perhatikan program berikut:

```
a = 6
b = 12

while b:
    a, b = b, a%b

print(a)
```

Apa *output* yang dihasilkan dari program di atas? Berikan penjelasan alur dari program tersebut!

Jawaban:

Ingat, pada `while conditional`, dimana nilai dari `conditional` adalah selain 0, maka iterasi akan tetap berjalan. Lakukan *tracing*:

Iterasi 1: $a = 6, b = 12 \rightarrow a = 12, b = 6\%12 \rightarrow a = 12, b = 6$

Iterasi 2: $a = 6, b = 12 \rightarrow a = 6, b = 12\%6 \rightarrow a = 6, b = 0$

Maka output dari program adalah 6

5. Ubah `for loop` berikut ini menjadi `while loop` yang ekuivalen. Berikan penjelasan alur dari program Anda!

```
result=1
lst=[1, 3, 5, 7]
for element in lst:
    result=result*element
```

Jawaban:

```
i=0;
result=1;
lst=[1,3,5,7]
while( i<len(lst):
    result = result * lst[i];
    i+=1;
```

6. Berapakah representasi **heksadesimal** dari **unsigned binary** 1110110101 ? Jelaskan proses perhitungannya untuk melakukan konversi tersebut.

Jawaban: 3B5

7. Berapakah nilai desimal (basis 10) dari **two's complement 8 bit** 10111101 ? Jelaskan proses perhitungannya untuk melakukan konversi tersebut.

Jawaban: -67

8. Lengkapi fungsi **count_positive_even()**, yang menerima input list integer dua dimensi (list of lists), dan mengembalikan list 1 dimensi berisi banyaknya bilangan genap positif dari setiap elemen list input. Berikan penjelasan alur dari program Anda!

Contoh:

```
>>> count_positive_even ([[2,-1,0], [4,3,4]])
[1,2]
```

```

def count_positive_even(lst):
    result=[]
    for i in lst :
        count=0
        for j in _____ :
            if lst[j] > 0 and _____ :
                _____
    return result

```

Jawaban:

```

def count_positive_even(lst):
    result=[]
    for i in lst :
        count=0
        for j in range(len(i)) :
            if i[j] > 0 and i[j]%2==0 :
                count+=1
        result.append(count)
    return result

```

```

def count_positive_even(lst):
    result=[]
    for i in lst :
        count=0
        for j in i :
            if j > 0 and j%2==0 :
                count+=1
        result.append(count)
    return result

```

9. **Pak Bejo** adalah seorang pedagang alat transportasi tradisional. Beliau menjual becak (seharga 3.000.000 – tiga juta – per satuan), delman (seharga 10.000.000 – sepuluh

juta – per satuan), dan sepeda kumbang (seharga 2.000.000 – dua juta – per satuan). Selama ini, beliau menghitung total hasil penjualan secara tradisional, yakni dengan pena dan kertas. Sayang sekali, cara penghitungan ini terkadang kurang cepat serta rentan kesalahan.

Sebagai salah seorang pelanggan setia toko Pak Bejo, Anda tertarik untuk menemukan cara untuk menghitung hasil penjualan alat transportasi tradisional secara nontradisional. Buatlah **fungsi Python** yang dapat menghitung total hasil penjualan, dengan spesifikasi sebagai berikut:

- Nama fungsi: **total_penjualan()**
- Terdapat 3 argument dengan ekspektasi tipe int: **becak, delman, dan sepeda**.
- Fungsi **mengembalikan int** yang merupakan total penjualan.

Berikut contoh pemanggilan fungsi beserta kembaliannya:

```
>>> total_penjualan(2,2,0)
```

```
26000000
```

```
>>> total_penjualan(1,1,10)
```

```
33000000
```

Buatlah fungsi tersebut dan berikan penjelasan alur dari program Anda!

Jawaban:

1	def total_penjualan(becak, delman, sepeda):
2	return becak*3000000 + delman*10000000 + sepeda*2000000

Kode ini tentang fungsi total_penjualan() dengan 3 argumen, yakni becak, delman, dan sepeda. Fungsi tersebut mengembalikan nilai total penjualan yang merupakan hasil penjumlahan dari perkalian antara jumlah transportasi tradisional dengan harganya.

10. **Perhatikan** kode berikut. Berikan penjelasan (boleh menggunakan ilustrasi gambar yang disertai narasi), kenapa *output*-nya **[3, 2, 7]**, bukan **[3, 2, 1]** ataupun *output* yang lain?

1	x = [3,2,1]
2	y = x
3	z = y
4	y.append(7)
5	z.remove(1)
6	print(x)

Jawaban:

Pada Baris 1, list [3,2,1] di-assign ke var x. Baris 2 melakukan assignment dari var x ke var y. Di sini var y akan **menunjuk ke object list yang sama** yang ditunjuk oleh var x. Baris 3 pada dasarnya melakukan hal yang serupa dengan Baris 2, yakni var z akan menunjuk ke object list yang ditunjuk oleh var y (dan juga var x). Oleh karena itu, perubahan-perubahan pada y dan z akan juga berdampak ke x, sehingga print(x) di Baris 6 menghasilkan [3,2,7].

11. **Apa yang** dilakukan oleh kode fungsi berikut apabila parameter x dipastikan bertipe list of booleans? Berikan penjelasan alur dari program ini!

1	def m(x):
2	z = []
3	for y in x:
4	if (y and True) or False:
5	z.append(y)
6	return len(z)

Jawaban:

Long story short, **menghitung banyaknya elemen True pada list x**. Perhatikan bahwa Baris 3 melakukan **iterasi elemen demi elemen** (yakni var y) pada list x, dan Baris 4-5 akan **menambahkan elemen y ke list z jika y bernilai True**. Fungsi kemudian akan **mengembalikan panjang list z**.

12. **Lengkapi** Baris 7 pada kode berikut ini sehingga dihasilkan *output*: [9, 7, 8]
Berikan penjelasan alur dari program Anda.


```
1 siji = [[1,2,3]]
2 loro = [[5,2,2]]
3 telu = [[3,3,3]]
4
5 res = []
6 for i in range(3):
7     res.append(siji[___][___] + loro[___][___] + telu[___][___])
8
9 print(res) # [9,7,8]
```

Jawaban:

Kode untuk melengkapi: `res.append(siji[0][i] + loro[0][i] + telu[0][i])`

Penjelasan: Teknik pengaksesan list 2D, yakni kita menggunakan access by index secara berantai. Index yang awal selalu 0, karena untuk mengakses elemen (yang juga bertipe list) di dalam list siji, loro, dan telu. Index selanjutnya, memanfaatkan var i, karena iterasi dari kiri ke kanan.

13. Lengkapi bagian kode yang kosong ("_____") di bawah ini agar mengeluarkan hasil yang diharapkan. Berikan penjelasan alur dari program Anda!

Kode:

```
mylist = ['you', 'I', 'we']
for word in mylist:
    print("_____".format(word))
```

Hasil yang diharapkan:

```
.....you
.....I
.....we
```

Jawaban:

`{: >10}` atau `{0: >10}` atau `{0: >10s}` atau `.*(10-len(word))+"`

Ini merupakan format string sepanjang 10 yang menggunakan padding kanan dan karakter kosong sepanjang 10 - len(word) diisi dengan "."

14. Apa *output* dari program berikut? Berikan penjelasan alur dari program ini!

```
word = "pemrograman"
word.replace("r", "")
print(word)
```

Jawaban:

pemrograman

Penjelasan: string adalah immutable object. Method replace tidak mengubah string. Jika ingin mengeluarkan hasil replace dan menyimpannya sebagai word, maka perlu ada assignment ke word.

15. Apa *output* dari program berikut? Berikan penjelasan alur dari program ini!

```
kata = "terang"
kata[0] = "k"
if kata[0] == "k":
    print("kerang")
else:
    print("terang")
```

Jawaban:

Error / Type error

String adalah immutable object. Assignment seperti pada baris kedua tidak diizinkan

16. Apa *output* dari program berikut? Berikan penjelasan alur dari program ini!

```
kal = "kuku kaki kakekku kaku-kaku"
print((kal.split('k')).count('u'))
```

Jawaban:

Hasil split dari variable kal oleh 'k' adalah = ['', 'u', 'u ', 'a', 'i ', 'a', 'e', '', 'u ', 'a', 'u-', 'a', 'u']

Sehingga, hasil count 'u' pada list tersebut adalah **2**

17. Lengkapi paragraf berikut ini!

Saat membuka file menggunakan perintah open() di Python, kita membuka jalur untuk arus data; dimana data (file) dialirkan dari memori _____(A)_____ ke tempat penyimpanan di memori _____(B)_____ yang disebut *buffer*.

Salah satu alasan dibuatnya *buffer* adalah agar program yang sedang dijalankan _____(C)_____.

Jawaban:

A	disk / permanen / sekunder
B	RAM / primer
C	dapat langsung mengakses isi file dari buffer di memori (tidak perlu bolak-balik membaca dari disk yg membutuhkan waktu lama), sehingga program dapat berjalan dengan lebih cepat

18. Lengkapi fungsi `daftar()` di bawah ini. Fungsi `daftar()` menerima parameter berupa (1) `file_name`, dan (2) `email_addr` yang ingin ditambahkan di baris paling akhir pada berkas `file_name` tersebut. Jika email berhasil ditambahkan, akan dikembalikan `True`, dan jika tidak berhasil, dikembalikan `False`. Berikan penjelasan alur dari program ini!

```
1 def daftar(file_name, email_addr):
2     try:
3         file_object = _____
4         _____
5         file_object.close()
6         _____
7     except:
8         _____
9
10    nama_file = "emails.txt"
11    email = input("Masukkan alamat email: ")
12    if daftar(nama_file, email_input):
13        print("Email berhasil didaftarkan.")
14    else:
15        print("Email tidak berhasil didaftarkan")
```

Berkas **emails.txt** diasumsikan terdiri dari beberapa baris dengan masing-masing baris berisi sebuah email. Contoh file **emails.txt** sebagai berikut:

```
nobita.nobi@ui.ac.id
shizuka.minamoto@ui.ac.id
```

Berikut adalah contoh jalannya program:

```
Masukkan alamat email: dekisugi@ui.ac.id
Email berhasil didaftarkan.
```

Selanjutnya, berkas **emails.txt** akan berubah menjadi:

```
nobita.nobi@ui.ac.id
shizuka.minamoto@ui.ac.id
dekisugi@ui.ac.id
```

Berikan penjelasan alur dari program Anda!

Jawaban:

```
def daftar(file_name, email_addr):
    try:
        file_object = open(file_name, 'a') # 1
        #file_object = open(file_name, 'a+') # 1 alt answer
        print(email_addr, file = file_object) # 2
        #file_object.write(email_addr) # 2 alt answer
        file_object.close()
        return True # 3
    except:
        return False # 4

nama_file = "emails.txt"
email = input("Masukkan alamat email: ")
if daftar(nama_file, email): # ada typo tadinya
    print("Email berhasil didaftarkan.")
else:
    print("Email tidak berhasil didaftarkan.")
```

19. Lengkapi isian (1), (2), dan (3) masing-masing dengan tepat satu pilihan ekspresi/statement (A)-(H) agar program berikut mencetak string "error kedua". Berikan penjelasan untuk alur program Anda!

```

1  try:
2      ____ (1) ____
3      ____ (2) ____
4  except IndexError:
5      print("error pertama")
6  except ____ (3) ____:
7      print("error kedua")
8  except:
9      print("error ketiga")

```

Pilihan (A)-(H):

A	NameError
B	ZeroDivisionError
C	TypeError
D	m = [0]*3
E	m = "000"
F	print(m[m[0]])
G	print(1+m[1])
H	print(1/m[3])

Jawaban:

```

try:
    m = "000" # __1__ ALIAS E
    print(1+m[1]) # __2__ ALIAS G
# alternative: print(m[m[0]]) ALIAS F
except IndexError:
    print('error pertama')
except TypeError: # __3__ ALIAS C
    print('error kedua')
except:
    print('error ketiga')

```

20. Lengkapi isian (1), (2), dan (3) masing-masing dengan tepat satu pilihan ekspresi/statement (A)-(H) agar program berikut mencetak string "eror ketiga". Berikan penjelasan untuk alur program Anda!

```

1  try:
2     ____ (1) ____
3     ____ (2) ____
4  except IndexError:
5     print("eror pertama")
6  except ____ (3) ____:
7     print("eror kedua")
8  except:
9     print("eror ketiga")

```

Pilihan (A)-(H):

A	NameError
B	ZeroDivisionError
C	TypeError
D	m = [0]*3
E	m = "000"
F	print(m[m[0]])
G	print(1+m[1])
H	print(1/m[3])

Jawaban:

```

try:
    m = "000" # __1__ ALIAS E
    print(m[m[0]])# __2__ ALIAS F
except IndexError:
    print('eror pertama')
except NameError:# ALIAS A
#alternative ZeroDivisionError ALIAS B
    print('eror kedua')
except:
    print('eror ketiga')

```


Bagian 2. Esai

Bacalah soal dengan teliti. Tuliskan uraian Anda pada lembar jawaban. Terdapat 5 soal esai dengan bobot penilaian per soal sebesar 8 poin, sehingga total nilai maksimum adalah 40 poin.

1. Diberikan suatu list sebagai berikut:

```
[ [0,0], [0,2], [0,4], [2,0], [2,2], [2,4], [4,0], [4,2], [4,4] ]
```

- Buatlah program **tanpa menggunakan list comprehension** untuk menghasilkan list di atas. Lalu, berikan alur penjelasan untuk program Anda!
- Buatlah program **dengan menggunakan list comprehension** untuk menghasilkan list di atas. Lalu, berikan alur penjelasan untuk program Anda!

Perhatikan bahwa program yang Anda buat tidak boleh langsung mengenumerasi suatu *list* yang sudah jadi.

Jawaban:

a.

```
res = []  
  
for x in range(0,5,2):  
    for y in range(0,5,2):  
        res.append([x,y])  
  
print(res)
```

b.

```
res = [[x,y] for x in range(0,5,2) for y in range(0,5,2)]  
  
print(res)
```


2. Anda diminta untuk membuat program dengan ketentuan sebagai berikut:

- Program akan meminta masukan sebuah bilangan N , dimana $2 < N < 10$
- Bilangan N akan menentukan besarnya gambar huruf X yang akan dihasilkan

Berikut **contoh output** dari program bilamana N sesuai dengan kriteria:

$N = 3$	$N = 4$	$N = 5$	$N = 6$
X X X X X	X X XX XX X X	X X X X X X X X X	X X X X XX XX X X X X

Bilamana N tidak sesuai dengan kriteria, maka program **tidak akan mengeluarkan output** apapun.

Lengkapi 6 bagian pada program di bawah ini sehingga bisa memenuhi kebutuhan. Berikan penjelasan alur program yang Anda buat!

```
n = int(input('Masukkan nilai N: '))
index = 0

#Fill the matrix to draw X
while index __1__ n and __2__ and __3__:

    empty_space_list = [' '] * n
    empty_space_list[__4__] = 'x'
    empty_space_list[__5__] = 'x'
    __6__
    print(''.join(empty_space_list))
```

Jawaban:

```
n = int(input('Masukkan nilai N: '))
index = 0

#Fill the matrix to draw X
while index < n and n > 2 and n < 10:

    empty_space_list = [' '] * n
    empty_space_list[index] = 'x'
    empty_space_list[len(empty_space_list) - index - 1] = 'x'
    index += 1
    print(''.join(empty_space_list))
```

3. Fungsi berikut dimaksudkan untuk mencari bilangan **ganjil maksimal** dari suatu list bilangan bulat (int). Jika list kosong atau tidak ada bilangan ganjil sama sekali, kembalikan None. Akan tetapi, terdapat bug pada fungsi tersebut.
- Untuk *bug* tersebut, **buat dan jelaskan contoh** pemanggilan fungsi beserta kembaliannya (hasil return dari pemanggilan tersebut) yang memberikan justifikasi kenapa Anda menganggap itu sebagai *bug*.
 - Berikan dan jelaskan **solusi bug fix-nya**.

Petunjuk: Terdapat **lebih dari satu bug**.

```
1 def maks_ganjil(lst):
2
3     m = None
4
5     if len(lst):
6         return m
7
8     for e in lst:
9         if e % 2 and m == None:
10            m = e
11            elif e % 2 or e > m:
12                m = e
13
14     return m
```

Jawaban:

Pada kode tersebut, kondisi pada Baris 5 seharusnya bukan **len(lst)** tetapi **not len(lst)**. Contoh input-output yang buggy:

maks_ganjil([5]) mengembalikan None, dari seharusnya 5.

Asumsikan bug Baris 5 sudah diperbaiki. Kemudian, operator pada Baris 11 seharusnya bukan or tapi **and**. Contoh input-output yang buggy:

maks_ganjil([5,8]) mengembalikan 8, dari seharusnya 5.

maks_ganjil([5,1]) mengembalikan 1, dari seharusnya 5.

```
1 def maks_ganjil(lst):
2
3     m = None
4
5     if not len(lst):
6         return m
7
8     for e in lst:
9         if e % 2 and m == None:
10            m = e
11            elif e % 2 and e > m:
12                m = e
13
14     return m
```

Penjelasan:

Pada Baris 5-6, blok kode yang mengandung `not len(lst)` akan return `None` jika `lst` kosong, yang merupakan expected behavior.

Pada Baris 11, penggunaan **and** menjamin correctness dari fungsi, yakni kedua syarat (sebagai berikut: ganjil dan lebih besar dari `m`) harus terpenuhi baru `var m` bisa diupdate.

4. Buatlah satu fungsi bernama **berstemmer** yang menerima 1 parameter berupa kalimat. Selanjutnya fungsi tersebut mengembalikan kalimat baru dimana awalan be- dan ber- pada suatu kata (jika ada) akan dihapus.

Gunakan template berikut dan lengkapi bagian fungsi agar dapat menghasilkan luaran yang diinginkan. **Anda DILARANG menggunakan regular expression (regex) untuk menjawab persoalan ini.** Berikan penjelasan untuk alur program yang Anda buat!

```
def berstemmer(kalimat):
    stem_kalimat = _____ #a
    for word in kalimat.split():
        if _____: #b
            word = _____ #c
        elif _____: #d
            word = _____ #e
        stem_kalimat = _____ #f

    return stem_kalimat
```

Contoh pemanggilan fungsi

```
kalimat = "ayah sedang bekerja"
print(f"input = {kalimat}")
print(f"output = {berstemmer(kalimat)}")

kalimat = "aku rajin berlatih pemrograman Python"
print(f"input = {kalimat}")
print(f"output = {berstemmer(kalimat)}")

kalimat = "atlet sekaliber Kevin Sanjaya pun tetap harus rajin berlatih"
print(f"input = {kalimat}")
print(f"output = {berstemmer(kalimat)}")
```

Contoh *output* pemanggilan fungsi di atas:

```
input = ayah sedang bekerja
output = ayah sedang kerja
input = aku rajin berlatih pemrograman Python
output = aku rajin latihan pemrograman Python
input = atlet sekaliber Kevin Sanjaya pun tetap harus rajin berlatih
output = atlet sekaliber Kevin Sanjaya pun tetap harus rajin latihan
```

Asumsikan bahwa program yang Anda buat ini belum sempurna sehingga kata-kata dasar dengan awalan be- atau ber- juga akan terpotong. Misalnya jika diberikan:

```
kalimat = "aku memelihara seekor kucing betina"
print(f"input = {kalimat}")
print(f"output = {berstemmer(kalimat)}")

kalimat = "adikku menemukan berang-berang di sungai "
print(f"input = {kalimat}")
print(f"output = {berstemmer(kalimat)}")
```

Maka dihasilkan:

```
input = aku memelihara seekor kucing betina
output = aku memelihara seekor kucing tina
input = adikku menemukan berang-berang di sungai
output = adikku menemukan ang-berang di sungai
```

Jawaban:

```
def berstemmer(kalimat):

    stem_kalimat = ''
    for word in kalimat.split():
        if len(word) >= 3 and word[:3].lower() == 'ber':
            word = word[3:]
        elif len(word) >= 2 and word[:2].lower() == 'be':
            word = word[2:]
        stem_kalimat = stem_kalimat + word + ' '

    return stem_kalimat
```

5. Buatlah sebuah program yang menghitung **frekuensi (jumlah kemunculan)** masing-masing alfabet A-Z di sebuah file teks. Jumlah kemunculan disimpan dalam sebuah list berukuran 26, kemudian dicetak. Pencetakan diurutkan berdasarkan alfabet. Huruf kapital dan huruf kecil dianggap sama (contoh: frekuensi huruf A pada 'Algoritma' adalah 2). Karakter lain pada file teks seperti spasi, tanda baca, numerik, dan sebagainya, diabaikan.

Gunakan try-except dan while-loop untuk meminta pengguna memasukkan nama file teksnya hingga file dengan nama tersebut ditemukan; Jika tidak ada, program mencetak pesan untuk memasukkan nama file teks lagi.

Sebagai contoh, file bernama **input.txt** berisi baris-baris teks sebagai berikut:

```
Ikan hiu lagi ngaca
I lope you buat yang baca
ikan hiu naik ambulance
Hiu hiu hiu hiu
```

Jika diberikan file **input.txt** seperti di atas, dan diketahui tidak terdapat file `1nput.txt` dan `inpoet.txt`, maka eksekusi program adalah sebagai berikut:

```
Masukkan nama file: 1nput.txt
Masukkan nama file: inpoet.txt
Masukkan nama file: input.txt
Huruf A memiliki jumlah kemunculan 12.
Huruf B memiliki jumlah kemunculan 3.
Huruf C memiliki jumlah kemunculan 3.
Huruf D memiliki jumlah kemunculan 0.
Huruf E memiliki jumlah kemunculan 2.
Huruf F memiliki jumlah kemunculan 0.
Huruf G memiliki jumlah kemunculan 3.
Huruf H memiliki jumlah kemunculan 6.
Huruf I memiliki jumlah kemunculan 11.
Huruf J memiliki jumlah kemunculan 0.
Huruf K memiliki jumlah kemunculan 3.
Huruf L memiliki jumlah kemunculan 3.
Huruf M memiliki jumlah kemunculan 1.
Huruf N memiliki jumlah kemunculan 6.
Huruf O memiliki jumlah kemunculan 2.
Huruf P memiliki jumlah kemunculan 1.
Huruf Q memiliki jumlah kemunculan 0.
Huruf R memiliki jumlah kemunculan 0.
Huruf S memiliki jumlah kemunculan 0.
Huruf T memiliki jumlah kemunculan 1.
Huruf U memiliki jumlah kemunculan 9.
Huruf V memiliki jumlah kemunculan 0.
Huruf W memiliki jumlah kemunculan 0.
Huruf X memiliki jumlah kemunculan 0.
Huruf Y memiliki jumlah kemunculan 2.
Huruf Z memiliki jumlah kemunculan 0.
```

Gunakan *template* program di bawah ini. Berikan penjelasan alur dari program Anda!

```
1 while True:
2     try:
3         namafile = input("Masukkan nama file: ")
4         fileku =
5         freq = [0]*26 #list freq untuk mencatat jumlah kemunculan tiap huruf
6         for baris in fileku:
7             for char in baris:
8
9
10
11
12         for i in range(26):
13
14
15
16         fileku.close()
17
18
19
20     except:
21
```

Jawaban:

```
while True:
    try:
        namafile = input("Masukkan nama file: ")
        fileku = open(namafile, "r")

        freq = [0]*26 #list freq untuk mencatat jumlah kemunculan tiap
huruf

        for baris in fileku:
            for char in baris:
                char = char.upper()
                ind = ord(char) - 65
                if ind >= 0 and ind <= 25:
                    freq[ind] += 1

        for i in range(26):
            print('Huruf {} memiliki jumlah kemunculan
{}'.format(chr(i+65), freq[i]))

        fileku.close()
```



```
except IOError:  
    namafile = input("Masukkan nama file: ")
```

Good luck :-)