

Nama : Kartika Indah Widyas Tantri
NIM : 222410102001
Prodi : Teknologi Informasi
Mata Kuliah : Praktikum Algoritma & Pemrograman (Kelas C)

Soal 1

```
print('!','*20,!!')  
print(' '*7," SOAL 1 ",'*7')  
print('!','*20,!!')
```

code diatas hanya untuk menampilkan header soal ke 1,2, dan 3 di setiap program

```
! ===== !  
          SOAL 1  
! ===== !
```

```
jumlah = int(input("Masukkan jumlah kata : "))
```

Code diatas adalah untuk menginputkan data ke dalam variabel jumlah, data yang diinputkan harus berupa integer karena jika tidak, akan muncul error berupa ValueError karena value yang diterima tidak bertipe integer

```
Exception has occurred: ValueError ×  
invalid literal for int() with base 10: 'a'
```

```
for i in range(jumlah):  
    in_str = input("Masukkan kata ke-{0} : ".format(i+1))  
  
    print("\nHasil : ")  
    temp_arr = [i for a,i in enumerate(in_str)]  
    for y in range(len(temp_arr)-1):  
        banding = ""  
        if ord(temp_arr[y]) > ord(temp_arr[y+1]):  
            banding = "lebih"  
            selisih = ord(temp_arr[y]) - ord(temp_arr[y+1])  
        elif ord(temp_arr[y]) < ord(temp_arr[y+1]):  
            banding = "kurang"  
            selisih = ord(temp_arr[y+1]) - ord(temp_arr[y])  
  
        print("{char1} {banding} dari {char2}, selisihnya adalah {selisih}".format(char1 =  
temp_arr[y], char2 = temp_arr[y+1], banding = banding, selisih = selisih))  
    print()
```

```
for i in range(jumlah): # lakukan perulangan sebanyak variabel jumlah
```

Perulangan sebanyak variabel jumlah

```
in_str = input("Masukkan kata ke-{0} : ".format(i+1))
```

Pada syntax input terdapat **{0}** dan **.format()**, itu adalah string format yang dimana fungsi nya untuk membuat supaya kalimat “Masukkan kata ke-n” menjadi dinamis mengikuti increment dari variabel **i**, **i+1** didalam format tersebut adalah output yang akan muncul di dalam **{0}** tersebut, kenapa **i + 1** karena index awal dari loop tersebut dimulai dari 0 jadi jika tidak + **1** maka output kalimat tersebut akan menjadi dimulai dari 0 yang dimana hal tersebut tidak umum, karena bilangan biasanya dimulai dari 1.

```
Masukkan kata ke-1 : aranyaka
```

```
temp_arr = [i for a,i in enumerate(in_str)]
```

Code diatas adalah untuk memecah string menjadi sebuah list character, cara nya dengan lakukan looping pada sebuah **enumerate(in_str)**, a = index dari **enumerate(in_str)** dan i = valuenya, lalu kita ambil value nya saja maka tambahkan i di paling kiri, maka akan tercipta sebuah list baru berupa pecahan character dari string sebelumnya

```
for y in range(len(temp_arr)-1):
```

Selanjutnya lakukan perulangan sejumlah panjang list **temp_arr - 1**, Kenapa - 1 karena perbandingan masing-masing karakter tidak bisa lebih atau sama dengan panjang **temp_arr**, sesuai dengan output soal yang dimana perbandingan terakhir adalah karakter ke-n dengan karakter ke-akhir dan karakter-akhir tidak dibandingkan lagi.

```
    if ord(temp_arr[y]) > ord(temp_arr[y+1]):  
        banding = "lebih"  
        selisih = ord(temp_arr[y]) - ord(temp_arr[y+1])  
    elif ord(temp_arr[y]) < ord(temp_arr[y+1]):  
        banding = "kurang"  
        selisih = ord(temp_arr[y+1]) - ord(temp_arr[y])
```

percabangan untuk membandingkan nilai dari karakter kiri dengan kanan, untuk bisa membandingkannya harus mendapatkan unicode dari karakter tersebut dengan menggunakan function **ord()** lalu baru kita dapat membandingkannya, jika karakter kiri lebih besar dari kanan maka variabel banding akan berisi string “lebih” ini akan digunakan pada print nantinya, lalu variabel selisih berisi pengurangan dari unicode karakter kiri dikurangi kanan. Jika kondisi pertama

tidak terpenuhi maka akan di cek lagi pada kondisi berikutnya yaitu jika karakter kiri lebih kecil dari karakter kanan maka variabel banding akan berisi string “kurang” dan variabel selisih berisi pengurangan dari karakter kanan dikurangi karakter kiri

```
print("{char1} {banding} dari {char2}, selisihnya adalah {selisih}".format(char1 = temp_arr[y], char2 = temp_arr[y+1], banding = banding, selisih = selisih))
```

Lalu selanjutnya baru print hasil perhitungan sebelumnya, char1 untuk menampung karakter kiri dan char2 untuk menampung karakter kanan, lalu banding akan menampung nilai dari variabel banding, dan selisih akan menampung nilai variabel selisih. Dan hasil akhirnya seperti ini contohnya

```
a kurang dari r, selisihnya adalah 17
r lebih dari a, selisihnya adalah 17
a kurang dari n, selisihnya adalah 13
n kurang dari y, selisihnya adalah 11
y lebih dari a, selisihnya adalah 24
a kurang dari k, selisihnya adalah 10
k lebih dari a, selisihnya adalah 10
```

Proses-proses diatas akan diulangi sebanyak jumlah yang diinputkan user.

Soal 2

```
jumlah_data = int(input("Masukkan jumlah sapi : "))
```

Input data ke dalam variabel **jumlah_data** yang nanti dipakai untuk melakukan perulangan untuk input kode sapi sama seperti penjelasan Soal 1 maka jika input tidak berupa angka akan terjadi error

```
jumlah_hari = 0
```

Pendeklarasian variabel **jumlah_hari = 0** adalah untuk menampung jumlah hari dari masing-masing sapi yang dipilih oleh user

```
for i in range(jumlah_data):
```

Perulangan sejumlah panjang jumlah_data

```
inp = int(input("Masukkan kode sapi : "))
```

Lalu inputkan kode angka sapi sesuai pada soal yang diberikan

```
    if(inp == 1):
        jumlah_hari += 690
    elif(inp == 2):
        jumlah_hari += 420
    elif(inp == 3):
        jumlah_hari += 530
    elif(inp == 4):
        jumlah_hari += 330
```

Lalu selanjutnya barulah cek dengan percabangan kode sapi yang diinputkan user sebelumnya,

1. jika 1 maka tambahkan nilai 690 ke variabel jumlah_hari
2. jika 2 maka tambahkan nilai 420 ke variabel jumlah_hari
3. jika 3 maka tambahkan nilai 530 ke variabel jumlah_hari
4. jika 4 maka tambahkan nilai 330 ke variabel jumlah_hari

Proses diatas akan diulangi sampai perulangan mencapai nilai variabel jumlah_data

```
tahun = jumlah_hari // 365
```

```
bulan = (jumlah_hari - tahun * 365) // 30
```

```
hari = (jumlah_hari - tahun * 365 - bulan * 30)
```

Setelah proses diatas selesai baru selanjutnya akan dikalkulasikan nilai total dari jumlah hari tersebut agar menjadi berformat tahun, bulan, hari

```
tahun = jumlah_hari // 365
```

Untuk mendapatkan tahunnya rumusnya adalah jumlah_hari di bagi 365, kenapa pada syntax nya double “//” karena untuk membuat hasil dari pembagian tersebut dibulatkan kebawah

```
bulan = (jumlah_hari - tahun * 365) // 30
```

Untuk mendapatkan total bulan kita bisa menggunakan rumus $(\text{jumlah_hari} - \text{tahun} * 365) // 30$

```
hari = (jumlah_hari - tahun * 365 - bulan * 30)
```

Dan untuk mendapatkan total hari maka rumusnya seperti ini $(\text{jumlah_hari} - \text{tahun} * 365 - \text{bulan} * 30)$

Setelah mendapatkan tahun, bulan dan hari yang dibutuhkan baru selanjutnya print hasil tersebut

```
print('Jumlah hari yang diperlukan ialah : {0} Tahun, {1} Bulan, dan {2} Hari'.format(tahun, bulan, hari))
```

{0} untuk tahun, {1} untuk bulan, dan {2} untuk hari, hasilnya akan menjadi seperti ini contohnya

```
Jumlah hari yang diperlukan ialah : 4 Tahun, 6 Bulan, dan 0 Hari
```

Soal 3

```
KIRI = ['q', 'w', 'e', 'r', 't', 'a', 's', 'd', 'f', 'g', 'z', 'x', 'c', 'v', 'b']  
KANAN = ['y', 'u', 'i', 'o', 'p', 'h', 'j', 'k', 'l', 'n', 'm']
```

Definisikan constant variabel KIRI dan KANAN nanti untuk meng-cek typing nyaman atau tidak.

```
def isKanan(ch):  
    if(ch in KANAN):  
        return True  
    return False
```

Buat fungsi isKanan dengan parameter ch, fungsi ini nanti akan digunakan untuk cek apakah karakter yang dimasukkan adalah bagian Kanan, jika ch ada di list KANAN maka akan mengembalikan nilai boolean True jika tidak maka akan mengembalikan nilai boolean False.

```
def isKiri(ch):  
    if(ch in KIRI):  
        return True  
    return False
```

fungsi ini sama saja seperti fungsi isKanan akan tetapi pada fungsi ini untuk meng-cek apakah karakter termasuk bagian kiri atau tidaknya, dan sama juga fungsi ini akan mengembalikan nilai boolean True jika terpenuhi dan False jika tidak terpenuhi.

```
def isNyaman(nyaman, char1 = "", char2 = "", sisi = ""):  
    print(nyaman)  
    if(nyaman):  
        print("Penjelasan: Setiap karakter selalu bergantian tangan.")  
    else:  
        print("Penjelasan: Karakter yang berdempetan yakni \"{char1}\" dan \"{char2}\" berada  
di satu tangan ({sisi})".format(char1 = char1, char2 = char2, sisi = sisi))
```

Fungsi isNyaman adalah untuk meng-cek hasil akhir dari variabel nyaman yang dimana variabel tersebut bertipe boolean, jika variabel nyaman bernilai True maka akan muncul output "Penjelasan: Setiap karakter selalu bergantian tangan." Dan jika nilainya False maka akan memunculkan output "Penjelasan: Karakter yang berdempetan yakni \"{char1}\" dan \"{char2}\" berada di satu tangan ({sisi})". Parameter char1, char2, sisi akan digunakan pada saat variabel nyaman bernilai False karena kita perlu untuk mencetak karakter apa dan karakter apa yang berdempetan dan di sisi tangan mana itu terjadi maka dari itu diperlukan parameter tersebut.

Sekarang lanjut ke code utamanya

```
in_str = input("Masukkan Kata/kalimat : ")
```

Input kata atau kalimat ke dalam variabel in_str

```
nyaman = True
```

Definisikan variabel nyaman menjadi True variabel ini yang nantinya akan menentukan nyaman tidaknya kalimat yang diberikan user

```
temp_arr = [i for a,i in enumerate(in_str)]
```

Ubah variabel in_str menjadi list proses nya sama dengan penjelasan pada soal 1

```
for y in range(len(temp_arr)-1):
```

Lalu lakukan perulangan sejumlah panjang temp_arr – 1, untuk mengecek masing-masing value di dalam temp_arr apakah termasuk Kiri atau Kanan

```
    if isKanan(temp_arr[y]):  
        if isKanan(temp_arr[y+1]):  
            nyaman = False  
            isNyaman(nyaman, temp_arr[y], temp_arr[y+1], 'Kanan')  
            break
```

Di dalam perulangan lalu pertama kita cek dengan memanggil fungsi isKanan untuk karakter ber-index i jika terpenuhi alias karakter tersebut termasuk kanan maka lanjut cek apakah karakter dengan index i+1 (ini dimaksud untuk cek karakter selanjut dari index i) apakah kanan juga atau tidak, jika kanan maka set variabel **nyaman** menjadi **False**, lalu panggil fungsi isNyaman dengan parameter nyaman, karakter index i, karakter index i+1, 'kanan', dan program akan menjalankan fungsi isNyaman yang sudah dijelaskan diatas dan di cek apakah variabel nyaman pada parameter True atau False. Karena False maka akan keluar output sesuai dengan penjelasan fungsi nyaman bagian kondisi false nya

```
Masukkan Kata/kalimat : halo  
False  
Penjelasan: Karakter yang berdempetan yakni "l" dan "o" berada di satu tangan (Kanan)
```

```
elif isKiri(temp_arr[y]):  
    if isKiri(temp_arr[y+1]):  
        nyaman = False  
        isNyaman(nyaman, temp_arr[y], temp_arr[y+1], 'Kiri')  
        break
```

Untuk penjelasan bagian ini mirip seperti penjelasan pengecekan kanan diatas. Bedanya hanya untuk cek dari kiri

```
if(nyaman):  
    isNyaman(nyaman)
```

Terakhir apa bila perulangan selesai yang artinya inputan user tidak merujuk ke satu sudut saja baik kanan maupun kiri saja maka selanjutnya kita cek apakah variabel masih bernilai nyaman, jika iya maka jalankan fungsi isNyaman dengan hanya membawa variabel nyaman dan karena nyaman merupakan bernilai True maka akan muncul output berupa.

```
Masukkan Kata/kalimat : qpwoeiruv  
True  
Penjelasan: Setiap karakter selalu bergantian tangan.
```

Sedikit penjelasan tentang fungsi, kita bisa membuat fungsi yang memiliki nilai default yaitu dengan mencontoh code ini

```
def isNyaman(nyaman, char1 = "", char2 = "", sisi = ""):
```

jadi disaat memanggil fungsi ini maka kita hanya wajib mengisi satu parameter variabel saja yaitu nyaman, kenapa demikian karena ia tidak memiliki value default, beda dengan char1, char2, sisi mereka sudah didefinisikan dengan value "" jika mereka tidak dimasukkan pada saat pemanggilan fungsi tersebut. Maksud dimasukkan seperti ini

isNyaman(nyaman)

Hanya nyaman saja yang perlu wajib kita masukkan sedangkan yang lain itu opsional

Lalu bagaimana jika variabel char1,char2,sisi tidak memiliki default value seperti nyaman, maka jika dilakukan pemanggilan seperti contoh dimasukkan diatas maka akan menjadi error karena parameter yang wajib dimasukkan/wajib diisi tidak ditemukan pada saat menjalankan fungsi tersebut. Sebagai contoh saya mengubah fungsi isnyaman menjadi seperti ini

```
def isNyaman(nyaman, char1, char2 = "", sisi = ""):
```

Maka saat pemanggilan hanya mengisi 1 parameter saja akan terjadi error

isNyaman(nyaman)

Akan muncul error seperti ini

```
Exception has occurred: TypeError ×  
isNyaman() missing 1 required positional argument: 'char1'
```