|  |  |
| --- | --- |
| A picture containing text, clipart  Description automatically generated | **Laporan Praktikum Algoritma dan Pemrograman**  Semester Genap 2023/2024 |

|  |  |
| --- | --- |
| **NIM** | **71231005** |
| **Nama Lengkap** | **Josephine Marcelia** |
| **Minggu ke / Materi** | **07 / Pengolahan String** |

**SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.**

**SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.**

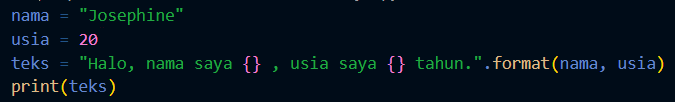
**PROGRAM STUDI INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
YOGYAKARTA  
2024**

# BAGIAN 1

Pengertian String

String pada Python merupakan sekumpulan karakter yang saling berurutan dan biasanya dikelilingi oleh tanda kutip tunggal (') atau ganda ("). Karakter-karakter tersebut dapat diakses dan diubah sesuai dengan indeksnya, yang dimulai dari 0 sampai panjang string dikurangi 1. String digunakan untuk merepresentasikan teks dan dapat berisi huruf, angka, simbol, atau bahkan karakter khusus seperti baris baru (\n).

Berikut merupakan contoh penggunaan string untuk format teks :

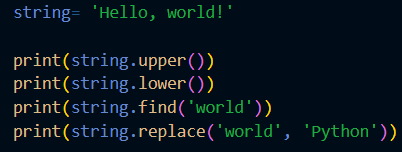


Output :

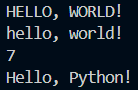


String juga dapat diolah dengan bantuan berbagai macam method yang disediakan oleh Python, seperti pembuatan kapital (uppercase), membuat huruf kecil (lowercase), mencari substring, dan mengganti substring.

Berikut adalah contoh penggunaan beberapa method tersebut :



Output :



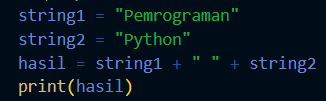
Manipulasi String

Pada memori komputer, string disimpan secara urut menggunakan list yang berisi huruf-huruf dengan indeks yang dimulai dari nol. Kita dapat memilih salah satu atau beberapa karakter berdasarkan index yang ada.

Berikut merupakan contoh-contoh manipulasi string :

1. Penggabungan Kalimat

Kita dapat menggabungkan kalimat dengan menggunakan operator +

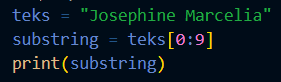


Output :



1. Pemotongan Kalimat

Pada contoh dibawah ini bisa dilihat bahwa karakter yang diambil adalah karakter dari indeks 0 sampai dengan 9. Sehingga indeks 10 dan seterusnya tidak termasuk.

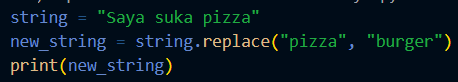


Output :



3. Penggantian Kalimat

Dengan menggunaskan fungsi replace(), kita dapat mengubah string pada python



Output :

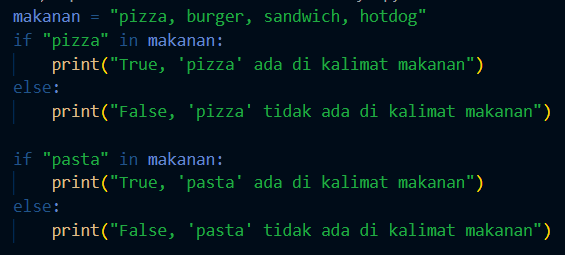


Operator dan Metode String

**OPERATOR in**

Contoh :

Operator in pada program dibawah ini dapat digunakan untuk memeriksa apakah kata pizza dan pasta ada pada string makanan atau tidak. Jika ada program akan menghasilkan output True, tetapi jika tidak program akan menghasilkan output False.

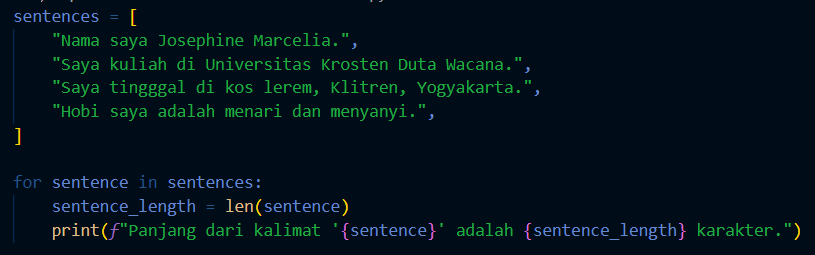
****

Output :

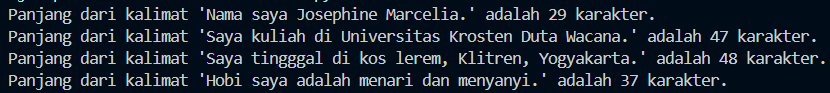


**FUNGSI len**

Pada Python, Fungsi len digunakan untuk menghitung panjang dari sebuah string, yaitu jumlah karakter yang ada dalam string , termasuk spasi dan symbol.



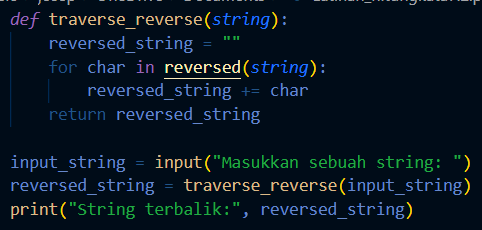
Output :



Traversing String

Traversing string pada Python merujuk pada proses pengaksesan setiap karakter dalam sebuah string secara berurutan untuk melakukan operasi tertentu, seperti pencarian, manipulasi, atau analisis. Hal ini melibatkan iterasi melalui setiap karakter dalam string, satu per satu, biasanya menggunakan struktur kontrol seperti loop (for atau while).

Berikut merupakan contoh program traversing string dalam Python yang mencetak setiap karakter string dalam urutan terbalik:



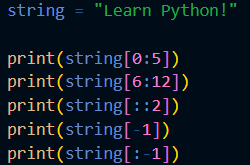
Output :



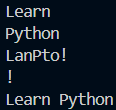
String Slice

String slice pada Python digunakan untuk memanipulasi string dengan cara mengambil beberapa bagian atau karakter dari string. Hal ini dilakukan dengan menggunakan indeks yang ditentukan, dan menetapkan nilai awal dan akhir dari slicing. Slice string akan mengembalikan substring yang dimulai dari indeks awal dan diakhiri sebelum indeks akhir.

Contoh :



Output :

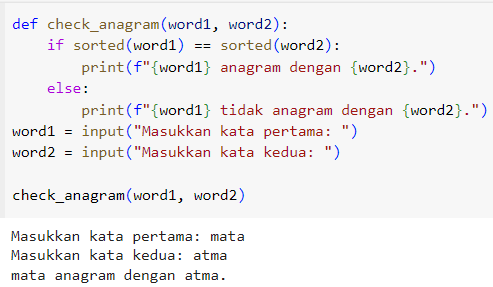


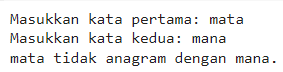
Beberapa Method String yang sering digunakan :

|  |  |  |
| --- | --- | --- |
| Nama Fungsi | Kegunaan | Penggunaan |
| Capitalize | Mengubah string menjadi huruf besar | String.capitalize() |
| Count | Menghitung jumlah substring yang muncul dari sebuah string | String.count() |
| Starts With | Mengetahui apakah suatu string diawali dengan string yang diinputkan | String.startswith() |
| Ends With | Mengetahui apakah suatu string diakhiri dengan string yang diinputkan | String.endswith() |
| Find | Mengembalikan indeks pertama string jika ditemukan string yang dicari | String.find() |
| Isupper/lower | Mencari tahu apakah sebuah string kapital atau kecil | String.isupper()  String.islower() |
| Is Digit | Mencari tahu apakah sebuah string sebuah digit atau tidak | String.isdigit() |
| Strip | Menghapus whitespace didepan dan dibelakang string | String.strip() |
| Split | Memisah sebuah string berdasarkan sebuah parameter | String.split(parameter) |

BAGIAN 2: LATIHAN MANDIRI

Lat 7.1





Penjelasan :

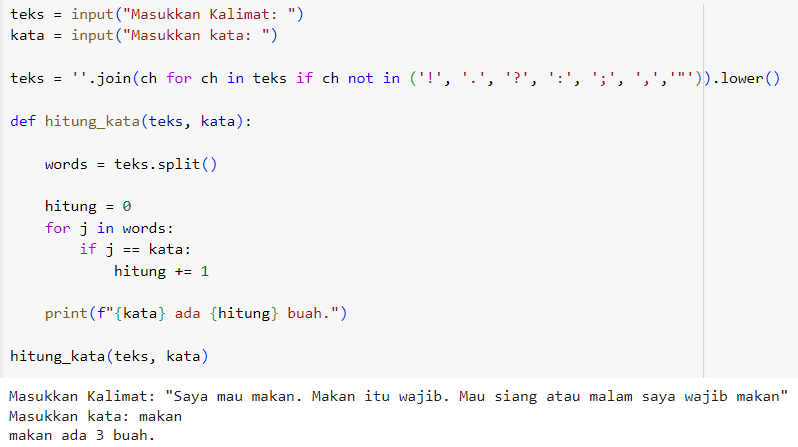
- Pada fungsi bernama check\_anagram saya menggunakan dua parameter, yaitu word1 dan word2, yang akan digunakan untuk memeriksa apakah kata yang diinput merupakan anagram atau bukan

- Pada baris if sorted(word1) == sorted(word2), kedua kata yang dimasukkan (word1 dan word2) diurutkan secara alfabetis menggunakan fungsi sorted(). Jika kedua kata tersebut memiliki susunan huruf yang sama setelah diurutkan, maka mereka dianggap sebagai anagram.

- Jika kata pada word1 anagram dengan word2 maka output akan menghasilkan “(word1) anagram dengan (word2)”

- Jika kata pada word1 tidak anagram dengan word2, maka output akan menghasilkan “(word1) tidak anagram dengan (word2)”

Lat 7.2



**Penjelasan :**

- Saya menggunakan fungsi bernama hitung\_kata yang menerima dua parameter, yaitu teks (kalimat yang dimasukkan oleh pengguna) dan kata (kata yang ingin dicari jumlah kemunculannya).

- Pada baris ini teks = ''.join(ch for ch in teks if ch not in ('!', '.', '?', ':', ';', ',','"')).lower() saya gunakan untuk menghilangkan tanda baca seperti tanda seru, titik, tanya, dua titik, titik koma, dan koma dari kalimat menggunakan ekspresi generator. Setelah itu, semua huruf dalam kalimat tersebut dikonversi menjadi huruf kecil agar pencarian kata menjadi case-insensitive.

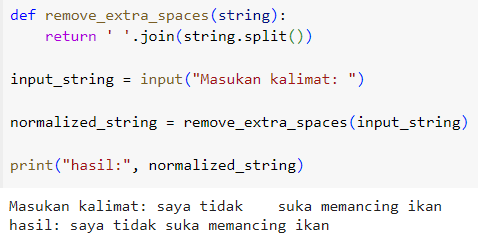
- Pada baris hitung = 0, variabel hitung saya gunakan untuk menyimpan jumlah kemunculan kata yang dicari.

- Pada baris for j in words:, program melakukan iterasi pada setiap kata dalam list words.

- Pada baris if j == kata: berarti jika kata yang sedang dievaluasi sama dengan kata yang ingin dicari, maka hitung akan bertambah satu.

- Program diatas kemudian akan mencetak hasil jumlah kata yang ada pada kalimat pada fungsi print(f" {kata} ada {hitung} buah.")

**Lat 7.3**

****

**Penjelasan :**

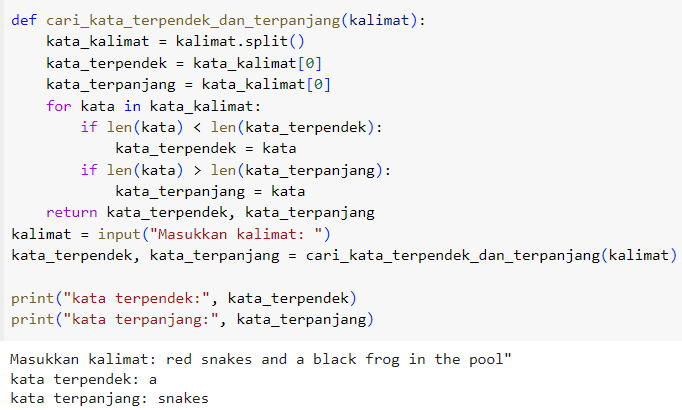
- Fungsi bernama remove\_extra\_spaces akan mengambil satu parameter, yaitu string, yang merupakan string yang akan diproses.

- Pada baris return ' '.join(string.split()): , string digunakan untuk menghapus spasi ekstra. Dimana string akan dipecah menjadi kata-kata menggunakan metode split(). Kemudian, kata-kata tersebut digabungkan kembali menggunakan metode join() dengan spasi sebagai pemisah.

- Pada baris normalized\_string = remove\_extra\_spaces(input\_string):, program akan memanggil fungsi remove\_extra\_spaces untuk menghapus spasi ekstra dari kalimat yang dimasukkan oleh pengguna.

- Kemudian program akan mencetak hasil dari proses penghapusan spasi ekstra, yaitu kalimat yang sudah dinormalisasi. Melalui fungsi print("hasil:", normalized\_string):

Lat 7.4



Penjelasan :

- Pada fungsi bernama cari\_kata\_terpendek\_dan\_terpanjang kita menerima satu parameter yaitu kalimat, yang merupakan kalimat yang akan diproses.

- Pada program diatas kalimat akan diubah menjadi daftar kata menggunakan split() dan disimpan dalam variabel kata\_kalimat

- kata\_terpendek dan kata\_terpanjang kemudiandiinisialisasi dengan kata\_kalimat[0]

- pada baris for kata in kata\_kalimat:, program melakukan iterasi pada setiap kata dalam list kata\_kalimat.

- if len(kata) < len(kata\_terpendek)::artinya jika panjang kata yang sedang dievaluasi lebih kecil dari panjang kata terpendek yang saat ini, maka kata terpendek akan diperbarui dengan kata yang sedang dievaluasi.

- if len(kata) > len(kata\_terpanjang): artinya jika panjang kata yang sedang dievaluasi lebih besar dari panjang kata terpanjang yang saat ini, maka kata terpanjang akan diperbarui dengan kata yang sedang dievaluasi.

- fungsi return kata\_terpendek, kata\_terpanjang: saya gunakan untuk mengembalikan dua nilai, yaitu kata terpendek dan kata terpanjang.

- Pada baris kata\_terpendek, kata\_terpanjang = cari\_kata\_terpendek\_dan\_terpanjang(kalimat):, program akan memanggil fungsi cari\_kata\_terpendek\_dan\_terpanjang untuk mencari kata terpendek dan terpanjang dalam kalimat yang dimasukkan oleh pengguna.

- Kemudian Program akan mencetak kata terpendek yang ditemukan dalam kalimat melalui fungsi print("kata terpanjang:", kata\_terpanjang)

Link Github:

<https://github.com/Josephinemrc/Tugas7-PrakAlpro.git>