



PYTHON PROGRAMMING

STUDI KASUS PENJADWALAN TUGAS

2023

@ariyoatmojo

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa karena berkat rahmat dan hidayah-Nya lah dapat menyelesaikan Buku yang berjudul “Python Programming Studi kasus Penjadwalan Tugas”.

Dalam kesempatan ini penulis tidak lupa menyampaikan ucapan terima kasih kepada semua pihak yang telah memberikan bantuan langsung maupun tidak langsung dalam menyelesaikan Buku ini.

Diharapkan Buku yang berjudul “ Python Programming Studi kasus Penjadwalan Tugas“ dapat menjadi Buku yang lebih baik dengan hasil yang dapat bermanfaat bagi para pembaca.

Bandung, 12 Juli 2023

Penulis

DAFTAR ISI

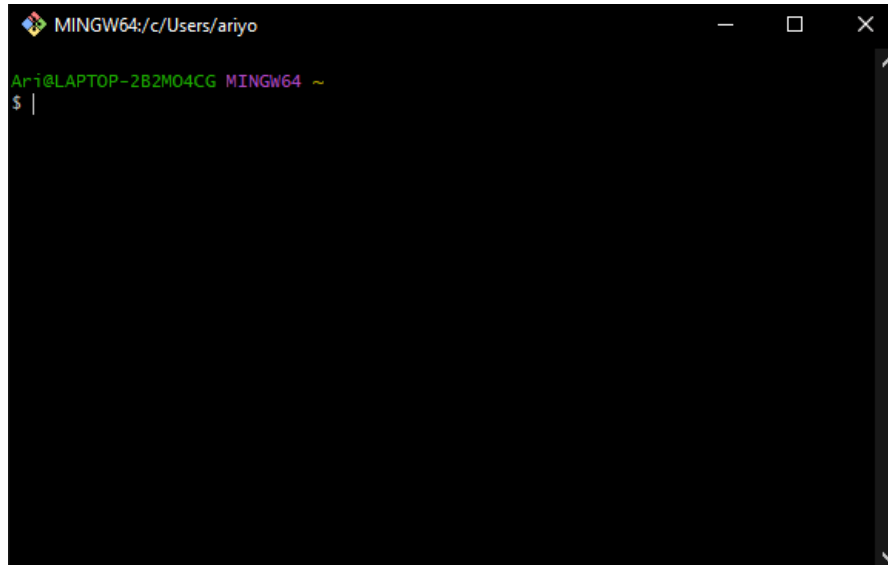
| | |
|--|----|
| KATA PENGANTAR | 2 |
| DAFTAR ISI | 3 |
| Chapter 1 SSH Key Tutorial | 4 |
| Chapter 2 Menginputkan Tugas..... | 6 |
| Chapter 3 Menjadwalkan Tugas..... | 7 |
| Chapter 4 Menghitung Waktu Pengerjaan Tugas | 9 |
| Chapter 5 Menampilkan Tugas Berdasarkan Waktu Pengerjaan Terpendek | 11 |
| Chapter 6 Menghitung Total Waktu Pengerjaan..... | 13 |

Chapter 1 SSH Key Tutorial

Download git-scm dari <https://git-scm.com/downloads>

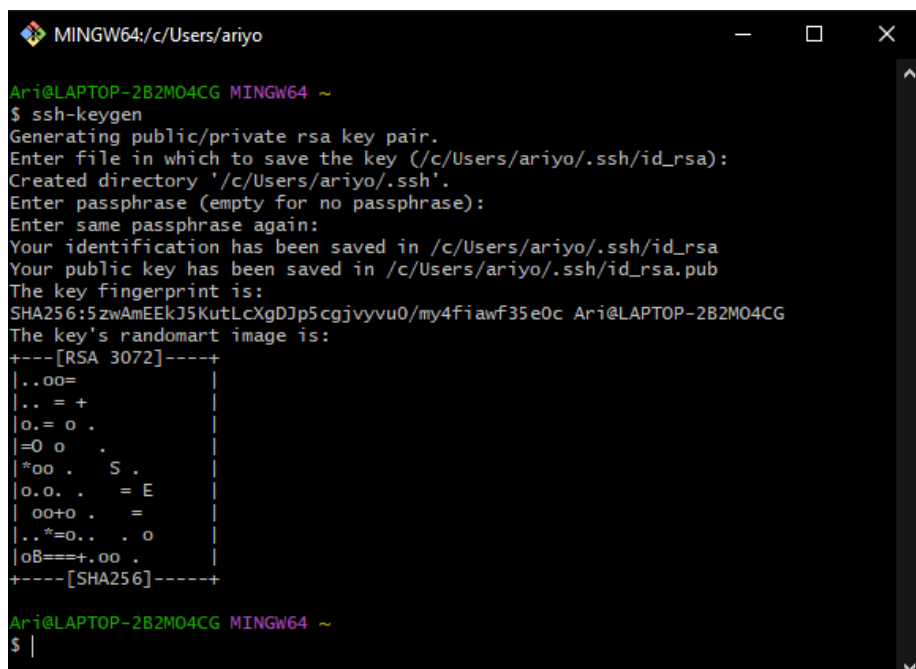
Cara membuat ssh key

1. Buka git bash.



```
MINGW64: c:/Users/ariyo
Ari@LAPTOP-2B2M04CG MINGW64 ~
$ |
```

2. Ketik ssh-keygen lalu tekan enter untuk memakai nama file default dan tekan enter 2x untuk mengosongkan passphrase.



```
MINGW64: c:/Users/ariyo
Ari@LAPTOP-2B2M04CG MINGW64 ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/ariyo/.ssh/id_rsa):
Created directory '/c/Users/ariyo/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/ariyo/.ssh/id_rsa
Your public key has been saved in /c/Users/ariyo/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5zwAmEEkJ5KutLcXgDjp5cgjvyvu0/my4fiawf35e0c Ari@LAPTOP-2B2M04CG
The key's randomart image is:
+---[RSA 3072]-----+
|..00=|
|.. = +|
|0. = 0 .|
|=0 o .|
|*00 . S .|
|o.o. . = E|
| oo+o . =|
|..*=0.. . o|
|oB===+.oo .|
+---[SHA256]-----+
Ari@LAPTOP-2B2M04CG MINGW64 ~
$ |
```

3. Setelah itu masuk ke github pilih setting lalu pilih ssh dan GPG keys.

SSH and GPG keys

- Pilih new ssh key untuk memasukan nama dan key.
- Masukan nama contoh title: SSH Windows Ariyo dan untuk Key buka file explorer di folder C:\Users\nama_user.ssh dan buka nama file id_rsa.pub menggunakan vscode lalu copy semua isi dalam file dengan mengetik ctrl+a+c dan paste ke dalam key setelah itu klik Add SSH key.

SSH keys / Add new

Title

SSH Windows Ariyo

Key type

Authentication Key

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQGDgEiDlrmTBLcVkv8/fmdF4iWfu5CWpZP2LwF2Nt/kVL6emFJIC4kf4oSgtTG5fCxOea
nEqRkbBkgUF6MkU9mbMP2e+UNl+uJqkZlv5HkliUlm24wmD5PczO0LFVA1fKELbep815TuemvmQYjJMSIV/pPjk1G7ba6V
MQN2VBCrIR+AQAKeU9c7XOfbFotglolnziApHOxLmdXiQgprrop49wJ1FrGN5K907QyFUWpdlotAgPCzgwVKg/XZrJTio+u
NYvq5qLks8sTfjJLVN25elgiW1uJ+IBvAQkd9KC2MdVhZ5s6DwiryXYFYFCJY/JFT3pfe+D3cW4pRkRFxyKDTVBoAZI9PWvop+
gerXNcN2cW276jzCoECPvv6B5J4dqAwYi3WkqL/GuuBPth8zmO4HUR+/8i2PKlgVC2wNeJ63tPUVaFaS3im2/JwOyd+omM
6FgJ8n9yFlkysjFTQYNTMkx/2qlmNU1fCt1yv+VsAwPjSpxTTzulGa+ROlna6pzqk= Ari@LAPTOP-2B2MO4CG
```


Add SSH key

- Hasil

SSH keys New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication Keys



SSH

SSH Windows Ariyo

SHA256: 5zwAmEEkJ5KutLcXgDJp5cgjvyvu0/my4fiawf35e0c

Added on Jul 5, 2023

Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

Chapter 2 Menginputkan Tugas

```
# Membuat list kosong untuk menyimpan tugass
daftar_tugas = []

# Menginputkan jumlah tugas
jumlah_tugas = int(input("Masukkan jumlah tugas: "))

# Menginputkan nama tugas dan menyimpannya ke dalam list
for i in range(jumlah_tugas):
    tugas = input("Masukkan nama tugas ke-{}: ".format(i + 1))
    daftar_tugas.append(tugas)

# Menampilkan daftar tugas
print("Daftar tugas:")
for tugas in daftar_tugas:
    print(tugas)
```

Penjelasan :

- Kode program di atas digunakan untuk menginputkan dan menyimpan daftar tugas.
- Pertama membuat list kosong dengan nama daftar_tugas untuk menyimpan tugas-tugas yang diinputkan.
- Selanjutnya, meminta pengguna untuk memasukkan jumlah tugas yang akan dijadwalkan.
- Setelah itu, menggunakan loop 'for', kita meminta pengguna untuk memasukkan nama tugas sebanyak jumlah tugas yang telah diinputkan. Nama tugas tersebut kemudian disimpan dalam list daftar_tugas.
- Terakhir, kita menampilkan daftar tugas yang telah diinputkan oleh pengguna.

Chapter 3 Menjadwalkan Tugas

```
# Membuat list kosong untuk menyimpan jadwal dan daftar tugas
jadwal_tugas = []
daftar_tugas = []

# Menginputkan jumlah tugas
jumlah_tugas = int(input("Masukkan jumlah tugas: "))

# Menginputkan nama tugas dan menyimpannya ke dalam list
for i in range(jumlah_tugas):
    tugas = input("Masukkan nama tugas ke-{}: ".format(i + 1))
    daftar_tugas.append(tugas)

# Menampilkan daftar tugas
print("Daftar tugas:")
for tugas in daftar_tugas:
    print(tugas)

# Menjadwalkan tugas
jumlah_tugas = len(daftar_tugas)
jumlah_hari = int(input("Masukkan jumlah hari: "))
tugas_per_hari = jumlah_tugas // jumlah_hari

# Menyusun jadwal tugas
for i in range(jumlah_hari):
    jadwal = daftar_tugas[i*tugas_per_hari:(i+1)*tugas_per_hari]
    jadwal_tugas.append(jadwal)

# Menampilkan jadwal tugass
for i, jadwal in enumerate(jadwal_tugas):
    print("Hari {}: {}".format(i+1, jadwal))
```

Penjelasan :

- Kode program di atas digunakan untuk menjadwalkan tugas-tugas yang telah diinputkan.
- Pertama, kita membuat list kosong dengan nama 'jadwal_tugas' untuk menyimpan jadwal tugas.
- Setelah itu, menggunakan loop 'for', kita meminta pengguna untuk memasukkan nama tugas sebanyak jumlah tugas yang telah diinputkan. Nama tugas tersebut kemudian disimpan dalam list daftar_tugas.

- Setelah itu, kita meminta pengguna untuk memasukkan jumlah hari yang tersedia untuk menjadwalkan tugas.
- Kemudian, kita menghitung berapa banyak tugas yang akan dijadwalkan dalam setiap hari dengan membagi jumlah tugas dengan jumlah hari.
- Dalam loop 'for', kita membagi daftar tugas menjadi bagian-bagian yang sesuai dengan jumlah tugas per hari, dan menyimpannya dalam list 'jadwal_tugas'.
- Terakhir, kita menampilkan jadwal tugas untuk setiap hari.

Chapter 4 Menghitung Waktu Pengerjaan Tugas

```
# Membuat dictionary kosong untuk menyimpan waktu pengerjaan tugas
waktu_pengerjaan = {}

# Membuat list kosong untuk menyimpan tugas
daftar_tugas = []

# Menginputkan jumlah tugas
jumlah_tugas = int(input("Masukkan jumlah tugas: "))

# Menginputkan nama tugas dan menyimpannya ke dalam list
for i in range(jumlah_tugas):
    tugas = input("Masukkan nama tugas ke-{: }.format(i + 1))
    daftar_tugas.append(tugas)

# Menampilkan daftar tugas
print("Daftar tugas:")
for tugas in daftar_tugas:
    print(tugas)

# Menginputkan waktu pengerjaan tugas
for tugas in daftar_tugas:
    waktu = float(input("Masukkan waktu pengerjaan untuk tugas {: }.format(tugas)))
    waktu_pengerjaan[tugas] = waktu

# Menampilkan waktu pengerjaan tugas
print("Waktu pengerjaan tugas:")
for tugas, waktu in waktu_pengerjaan.items():
    print("{: }: {: } jam".format(tugas, waktu))
```

Penjelasan :

- Kode program di atas digunakan untuk menginputkan dan menyimpan waktu pengerjaan untuk setiap tugas.
- Pertama, kita membuat dictionary kosong dengan nama 'waktu_pengerjaan' untuk menyimpan waktu pengerjaan tugas.
- Selanjutnya, menggunakan loop 'for', kita meminta pengguna untuk memasukkan waktu pengerjaan untuk setiap tugas yang ada dalam daftar tugas. Waktu pengerjaan

tersebut kemudian disimpan dalam dictionary 'waktu_pengerjaan', dengan nama tugas sebagai kunci.

- Terakhir, menampilkan waktu pengerjaan untuk setiap tugas.

Chapter 5 Menampilkan Tugas Berdasarkan Waktu Pengerjaan Terpendek

```
# Membuat dictionary kosong untuk menyimpan waktu pengerjaan tugas
waktu_pengerjaan = {}

# Membuat list kosong untuk menyimpan tugas
daftar_tugas = []

# Menginputkan jumlah tugas
jumlah_tugas = int(input("Masukkan jumlah tugas: "))

# Menginputkan nama tugas dan menyimpannya ke dalam list
for i in range(jumlah_tugas):
    tugas = input("Masukkan nama tugas ke-{}: ".format(i + 1))
    daftar_tugas.append(tugas)

# Menampilkan daftar tugas
print("Daftar tugas:")
for tugas in daftar_tugas:
    print(tugas)

# Menginputkan waktu pengerjaan tugas
for tugas in daftar_tugas:
    waktu = float(input("Masukkan waktu pengerjaan untuk tugas {}: ".format(tugas)))
    waktu_pengerjaan[tugas] = waktu

# Menampilkan waktu pengerjaan tugas
print("Waktu pengerjaan tugas:")
for tugas, waktu in waktu_pengerjaan.items():
    print("{}: {} jam".format(tugas, waktu))

# Menampilkan tugas berdasarkan waktu pengerjaan terpendek
tugas_terpendek = min(waktu_pengerjaan, key=waktu_pengerjaan.get)
print("Tugas dengan waktu pengerjaan terpendek: {}".format(tugas_terpendek))
```

Penjelasan :

- Kode program di atas untuk menampilkan tugas dengan waktu pengerjaan terpendek dari daftar tugas yang telah diinputkan.
- Kita menggunakan fungsi 'min()' untuk mencari kunci dengan nilai terkecil dari dictionary 'waktu_pengerjaan', dengan menggunakan argumen 'key=waktu_pengerjaan.get'.

- Hasilnya, kita akan mendapatkan tugas dengan waktu pengerjaan terpendek, yang kemudian ditampilkan.

Chapter 6 Menghitung Total Waktu Pengerjaan

```
# Membuat dictionary kosong untuk menyimpan waktu pengerjaan tugas
waktu_pengerjaan = {}

# Membuat list kosong untuk menyimpan tugas
daftar_tugas = []

# Menginputkan jumlah tugas
jumlah_tugas = int(input("Masukkan jumlah tugas: "))

# Menginputkan nama tugas dan menyimpannya ke dalam list
for i in range(jumlah_tugas):
    tugas = input("Masukkan nama tugas ke-{}: ".format(i + 1))
    daftar_tugas.append(tugas)

# Menampilkan daftar tugas
print("Daftar tugas:")
for tugas in daftar_tugas:
    print(tugas)

# Menginputkan waktu pengerjaan tugas
for tugas in daftar_tugas:
    waktu = float(input("Masukkan waktu pengerjaan untuk tugas {}: ".format(tugas)))
    waktu_pengerjaan[tugas] = waktu

# Menampilkan waktu pengerjaan tugas
print("Waktu pengerjaan tugas:")
for tugas, waktu in waktu_pengerjaan.items():
    print("{}: {} jam".format(tugas, waktu))

# Menghitung total waktu pengerjaan
total_waktu = sum(waktu_pengerjaan.values())
print("Total waktu pengerjaan: {} jam".format(total_waktu))
```

Penjelasan :

- Kode program di atas untuk menghitung total waktu pengerjaan untuk semua tugas yang telah diinputkan.
- Kita menggunakan fungsi 'sum()' untuk menjumlahkan nilai-nilai dalam dictionary 'waktu_pengerjaan', dengan menggunakan metode 'values()'.
- Hasilnya, kita akan mendapatkan total waktu pengerjaan dari semua tugas yang ditampilkan.