LAPORAN PRAKTIKUM

PEMROGRAMAN KOMPUTER

(PYTHON)



Disusun Oleh:

Rafli Pratama

H1101241008

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS TANJUNGPURA PONTIANAK

2024

# Praktikum 8

# Penanganan File

## Dasar Teori

* + 1. File Descriptor

File descriptor adalah sebuah integer unik yang diberikan oleh sistem operasi setiap kali file dibuka. File descriptor merepresentasikan sebuah referensi ke file atau perangkat lain (misalnya, *socket* atau *pipe*) dan digunakan oleh sistem operasi untuk mengidentifikasi dan mengelola file tersebut.

Dalam Python, file descriptor dapat diakses dengan properti fileno() dari objek file. Contoh penggunaan:

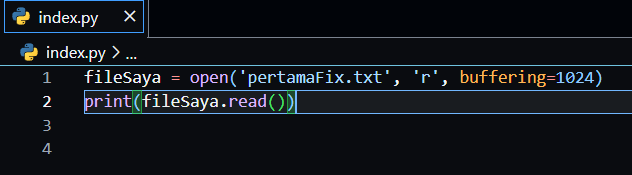


* + 1. Buffering

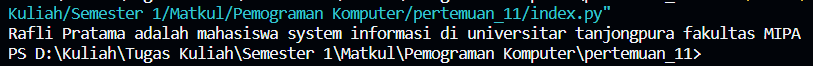
Buffering adalah mekanisme untuk mengelola aliran data antara memori dan disk dengan cara menyimpan data sementara di dalam *buffer* (memori sementara). Tujuan buffering adalah untuk meningkatkan efisiensi I/O karena akses ke memori lebih cepat dibandingkan akses langsung ke disk.

Buffering dapat diatur dengan parameter ketiga dari fungsi open(). Nilai defaultnya -1 (otomatis) yang menentukan tingkat buffering tergantung pada file dan sistem operasi. Terdapat tiga mode buffering utama:

* Unbuffered (0): Data ditulis atau dibaca langsung dari disk, tanpa penyimpanan di buffer.
* Line-buffered (1): Data di-buffer satu baris per satu waktu (umumnya untuk file teks).
* Fully buffered (>1 atau -1): Data di-buffer dengan ukuran blok tertentu (defaultnya sistem).



Output :

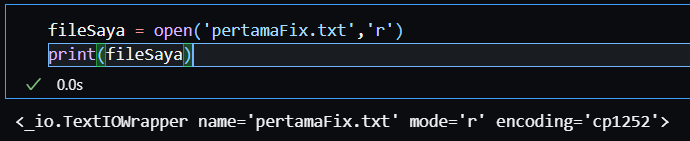


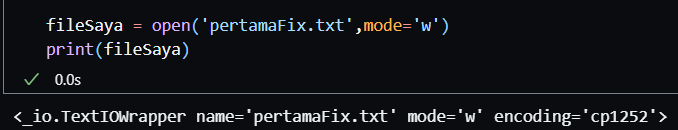
* + 1. Mode File

*File mode* menentukan cara membuka dan mengakses file. Python mendukung beberapa mode yang digunakan sebagai parameter kedua dari fungsi open():

* "r": *Read* - membuka file hanya untuk membaca.
* "w": *Write* - membuka file untuk menulis, menghapus konten jika file sudah ada.
* "a": *Append* - membuka file untuk menulis di akhir konten tanpa menghapus konten yang ada.
* "b": *Binary* - membuka file dalam mode biner, biasanya dikombinasikan dengan mode lain seperti "rb" atau "wb".
* "+": membuka file untuk *updating* (membaca dan menulis).

Contoh :



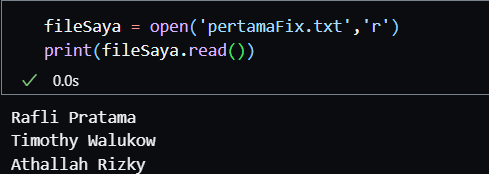


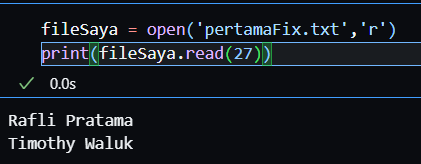
* + 1. Membuka File

Fungsi open() digunakan untuk membuka file dalam mode tertentu. Fungsi ini memiliki dua parameter utama: nama file dan mode akses file.

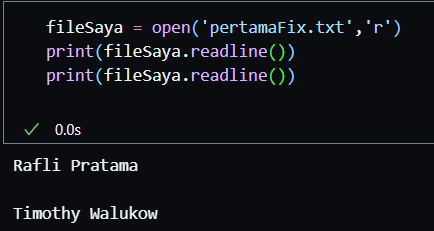
Berikut berbagai mode yang bisa digunakan dalam open():

* **'r' (Read)**: Membuka file hanya untuk dibaca. File harus sudah ada, dan pointer berada di awal file.
* **'w' (Write)**: Membuka file hanya untuk menulis. Jika file sudah ada, isinya akan dihapus; jika tidak, file baru akan dibuat.
* **'a' (Append)**: Membuka file untuk menulis di akhir file tanpa menghapus isinya. Jika file belum ada, maka akan dibuat.
* **'b' (Binary)**: Digunakan bersamaan dengan mode lain ('rb', 'wb', dll.) untuk membuka file dalam mode biner (bukan teks), yang biasanya digunakan untuk file non-teks seperti gambar atau video.
  1. **Fungsi read, readline, dan ridelines**
* **read()**: Membaca keseluruhan file atau sejumlah karakter tertentu. Jika size tidak diatur, read() akan membaca semua konten.

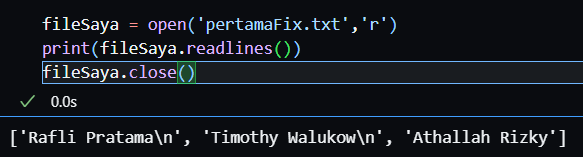




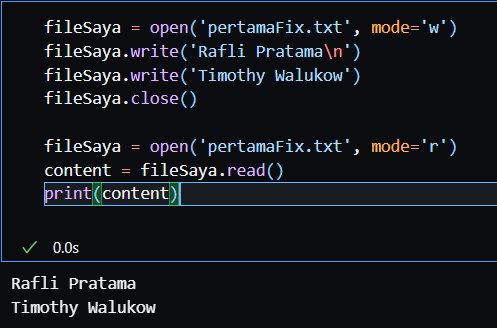
* **readline()**: Membaca satu baris dalam file. Setiap panggilan readline() mengembalikan baris berikutnya.



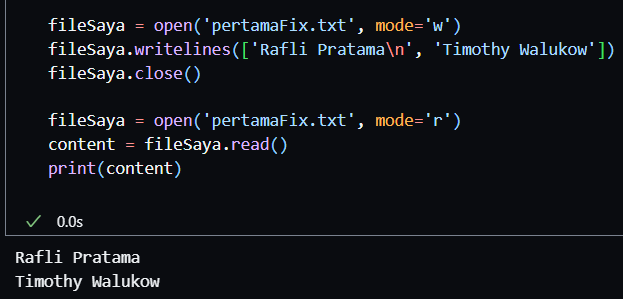
* **readlines()**: Membaca semua baris dalam file dan mengembalikannya sebagai list di mana setiap elemen adalah satu baris.



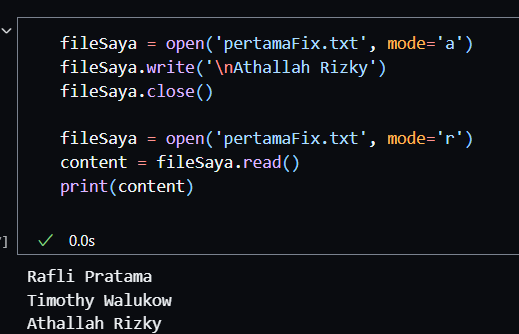
* 1. Fungsi Write dan Writelines
* **write()**: Menulis teks ke file. Jika file sudah ada dalam mode "w", semua konten lama akan dihapus. Untuk menambahkan teks tanpa menghapus, gunakan mode "a".



* **writelines()**: Menulis daftar string ke file. Setiap elemen dalam daftar ditulis tanpa karakter baris baru kecuali jika ditambahkan secara eksplisit.

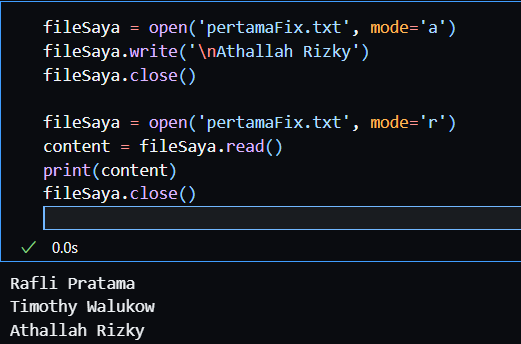


* 1. Append
* Mode "a" atau **append** dalam fungsi open() di Python digunakan untuk menulis data ke akhir file tanpa menghapus konten yang sudah ada. Mode ini dapat digunakan jika ingin menambahkan data baru tanpa memengaruhi data lama dalam file.



* 1. Fungsi Close()

Fungsi close() di Python digunakan untuk menutup file yang telah dibuka, baik untuk membaca, menulis, atau menambah data, sehingga memastikan semua data tersimpan dengan benar serta melepaskan sumber daya yang digunakan oleh file tersebut. Menutup file sangat penting karena membebaskan memori, mencegah kebocoran sumber daya, dan memastikan bahwa data yang masih berada di *buffer* disimpan secara permanen ke file. Jika close() tidak dipanggil, hal ini bisa menyebabkan error atau bahkan kehilangan data yang belum tersimpan.



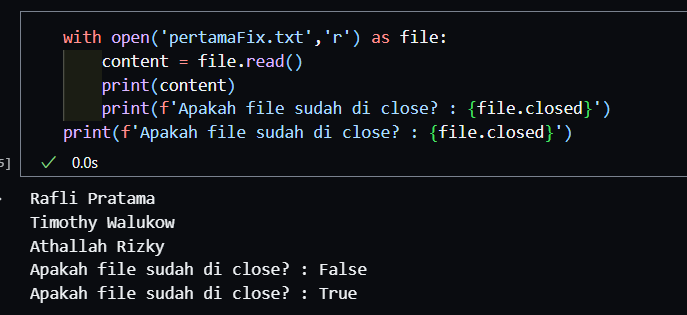
Karena fungsi close kita dapat melanjutkan program dengan mengubah mode listSaya yang sebelumnya w menjadi a tanpa mengalami erorr, lain hal jika kita tidak mengunakan fungsi close, program akan error atau bahkan mengalami bug dan resiko kehilangan data apabila data yang di buka banyak

* + 1. With

with adalah pernyataan yang digunakan untuk menangani file atau sumber daya lain dengan cara yang lebih aman dan efisien. Saat bekerja dengan file, with membantu membuka dan menutup file secara otomatis setelah blok kode selesai dijalankan, sehingga kita tidak perlu memanggil close() secara manual. Jika terjadi error atau pengecualian di dalam blok with, Python tetap akan memastikan bahwa file ditutup dengan benar. Menggunakan with adalah praktik terbaik saat bekerja dengan file, karena mencegah potensi kebocoran sumber daya dan membuat kode lebih bersih serta lebih mudah dibaca.

Mengapa with sangat penting :

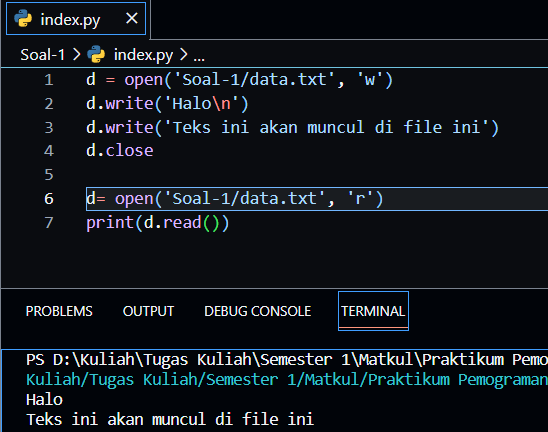
1. **Pengelolaan Sumber Daya yang Aman**: with memastikan bahwa sumber daya, seperti file, ditutup secara otomatis setelah digunakan. Ini mengurangi risiko kebocoran sumber daya yang dapat terjadi jika file tidak ditutup dengan benar.
2. **Penanganan Error**: Jika terjadi kesalahan atau pengecualian di dalam blok with, Python tetap akan menutup sumber daya tersebut. Ini membantu mencegah situasi di mana file tetap terbuka meskipun ada masalah.
3. **Kode yang Lebih Bersih dan Jelas**: Menggunakan with membuat kode lebih ringkas dan mudah dibaca. Hal ini menghilangkan kebutuhan untuk menulis kode penutupan terpisah, sehingga mengurangi kompleksitas.
4. **Meminimalkan Kesalahan Manusia**: Dengan otomatisasi proses penutupan, with membantu mengurangi kemungkinan kesalahan manusia, seperti lupa memanggil close() setelah selesai bekerja dengan file.

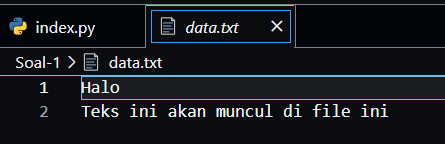


## 1.2. Percobaan Praktikum

### 1.2.1. Soal 1

Siapkan sebuah file txt dan isikan teks ke dalamnya. Kemudian tampilkan isi teks menggunakan bahasa python.



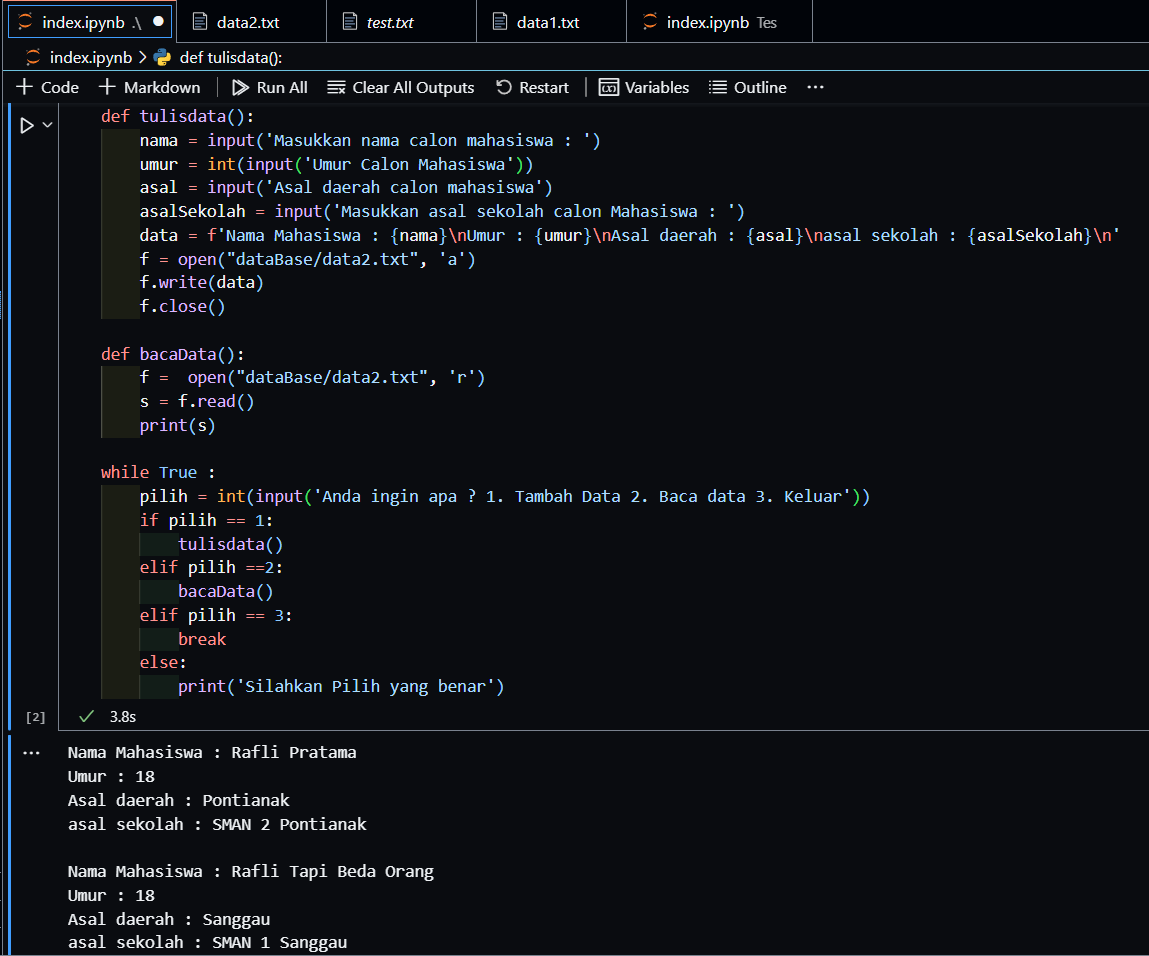


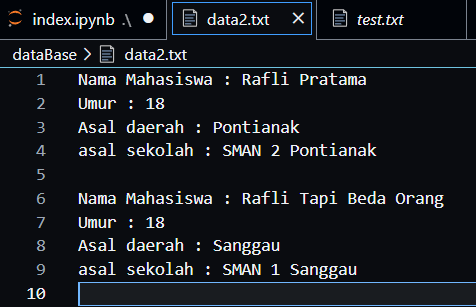
Penjelasan :

Dalam program tersebut kita menggunakan fungsi open() yang merupakan salah satu fungsi yang di gunakan untuk penangangan file pada python, pada baris pertama kita membuka file pada folder Soal-1 dengan nama file data.txt dan kita membuka nya dengan mode ’w’ atau write, sehingga kita dapat menambahkan konten ke dalam file tersebut, lalu kita mengclose itu menyimpan datanya dan kita buka kembali dengan mode ’r’ atau read yang kita gunakan untuk mengprint konten yang ada di dalam file data.txt.

### 1.2.1. Soal 2

Buat program sederhana pendaftaran mahasiswa. Identifikasi field yang diperlukan. Setelah proses pendaftaran selesai masukkan data mahasiswa tersebut kedalam sebuah file txt.





Penjelasan :

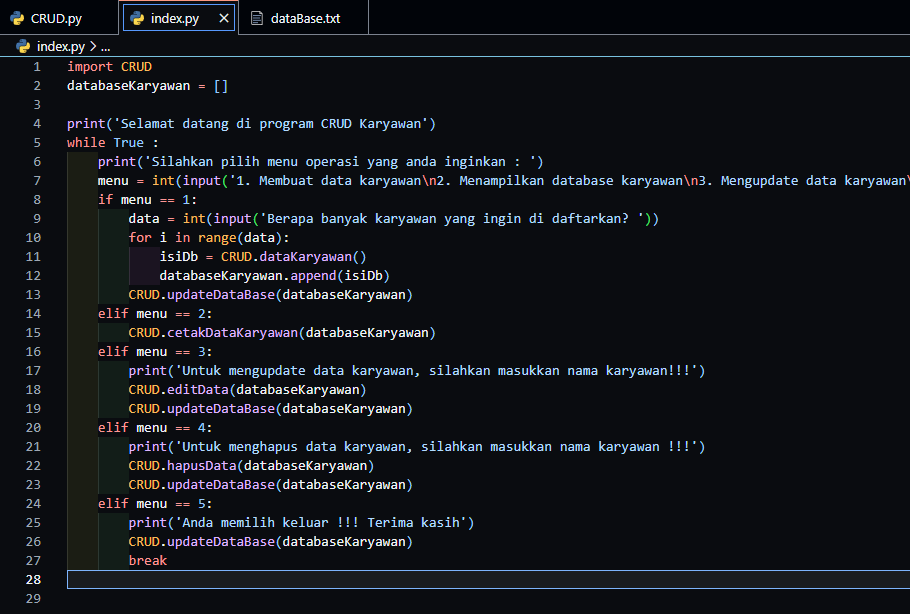
Dalam program kita membuat sebuah fungsi tulisData yang didalmnya berisi variabel yang menyimpan inputan dari user berupa data nama, umur, asal daerah, dan asal sekolah. Lalu kita membaut variabel data yang menyimpan data data yang user inputkan lalu kita membuka file data2.txt di folder dataBase dengan mode ’w’ atau write lalu kita menulis variabel data ke dalam file data2.txt

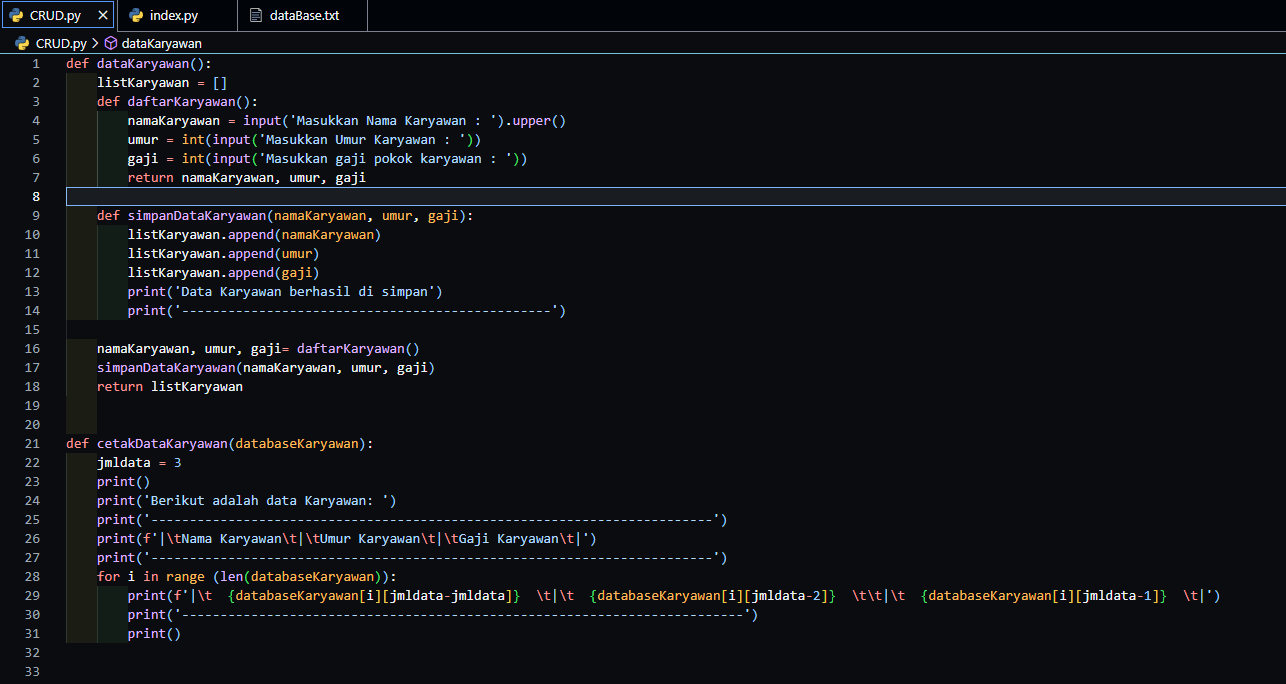
Dan di bawahnya kita membuat fungsi baca data yang membuka file data2.txt degan mode ’r’ atau read lalu kita menampilkan isinya dan di akhir kita membuat perulangan while yang masing masing kondisi menjalan fungsi fungsi yang telah kita buat

### 1.2.1. Soal 3

Gunakan file txt yang diberikan. Manipulasi data yang ada dan simpan dalam sebuah list. Kemudian buatlah program menu yang memiliki operasi dasar CRUD dari data yang sudah didapat.

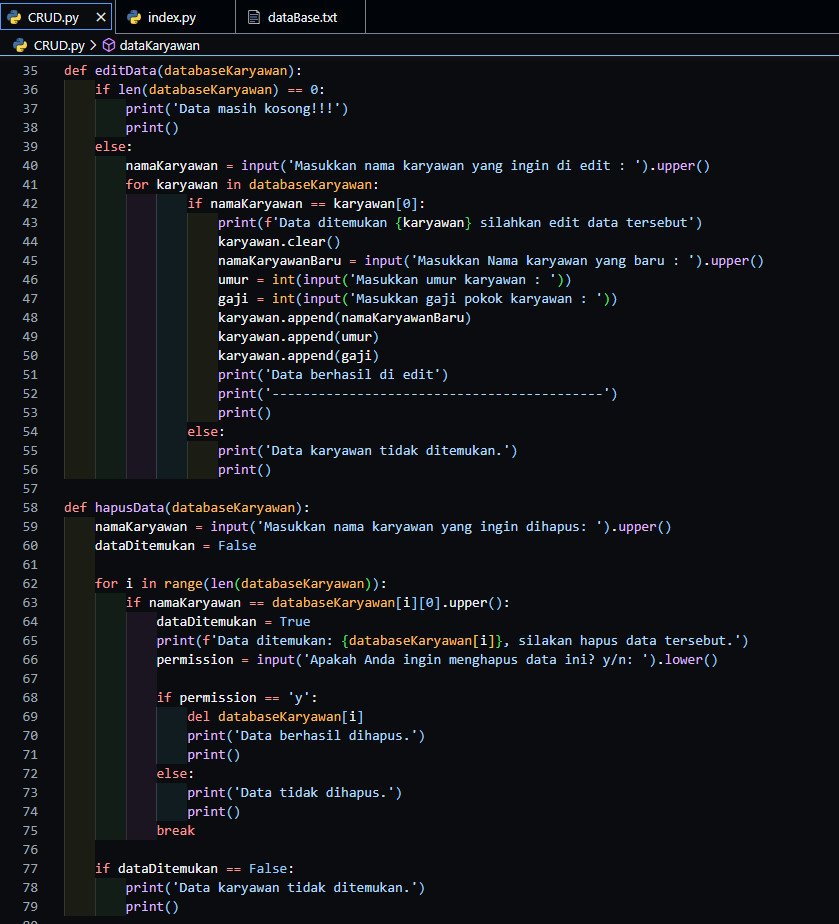
File index :

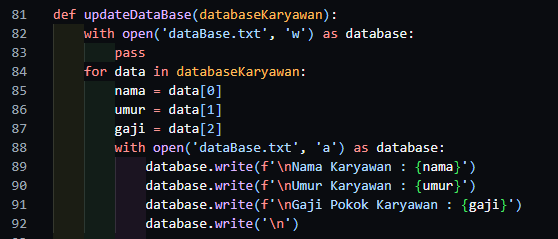


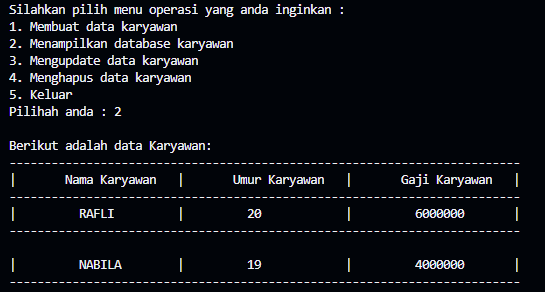


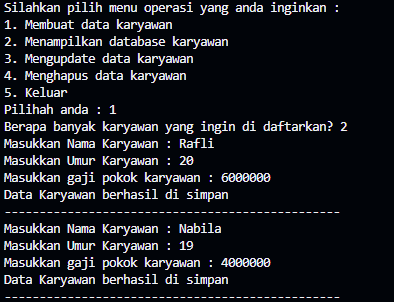
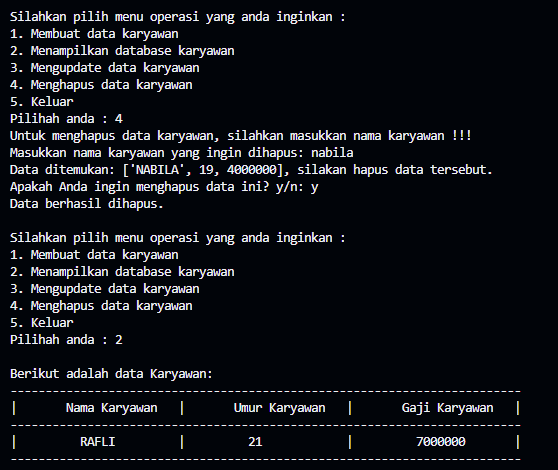
Penjelasan :

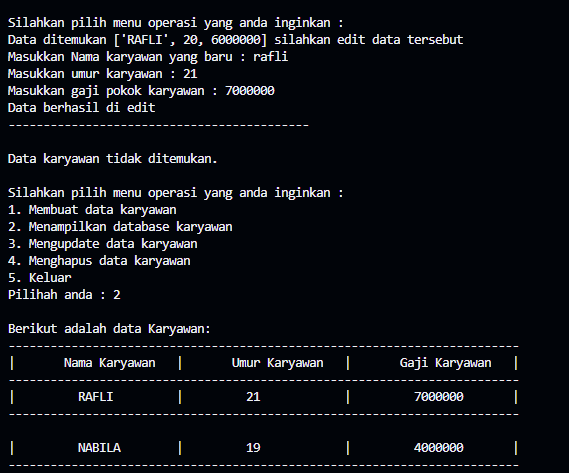
Pada program ini akan mengimport modul yang telah di buat,lalu membuat list kosong untuk menyimpan semua data karyawan, lalu akan masuk ke dalam perulangan while True, jika user menginputkan 1, maka program akan meminta user memasukkan jumlah karyawan yang ingin di data dan membuat perulangan sesuai jumlah tersebut dan di dalam perulangan itu akan di jalankan fungsi dataKaryawan yang ada di modul   
Jika user menginputkan 2, maka fungsi [ada modul CRUD yaitu cetakDataKaryawan akan di jalankan, jika 3 maka program pada fungsi editData akan dijalankan, jika 4 maka fungsi hapusData akan di jalankan dan jika memilih 5 maka program akan keluar dari perulangan, didalam setiap kondisi akan dijalan fungsi updateDataBase yang menuliskan data di dalam dataBaseKaryawan ke dalam file dataBase.txt

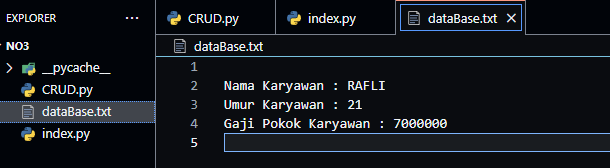
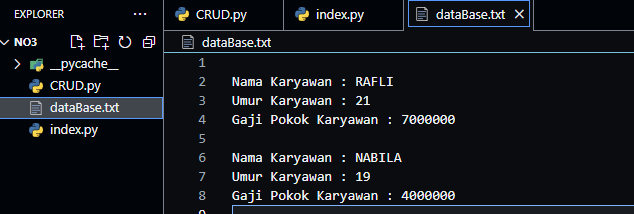




Output :







Penjelasan :

Pada modul CRUD, program akan membuat fungsi dataKaryawan yang berisi list kosong dan fungsi daftarKaryawan lalu di dalam nya terdapat inputan user yang berisi nama, umur, dan gaji pokok karyawan, dan mereturn nya, lalu dibuat juga fungsi simpanDataKaryawan yang menerima parameter umur dan gaji lalu meng append nya ke list kosong yang sudah di buat di atas setelah itu kita return list tersebut

pada modul CRUD didalamnya juga terdapat fungsi cetakDataKaryawan yang menerima parameter databaseKaryawan, setelah itu melakukan perulangan dan mencetak seluruh data yang ada di dalam list databaseKaryawan

Pada modul CRUD juga terdapat fungsi editData yang menerima parameter databaseKaryawan lalu di dalam nya kita melakukan pengecekan yang jika data di dalam list databaseKaryawan kosong maka data akan menampilkan bahwa data masih kosong, jika tidak user akan di minta memasukkan kata kunci berupa nama karyawan, jika ada nama karyawan yang sama dengan yang ada di dalam databaseKaryawan, maka data tersebut dapat di ganti dan di update

modul ini juga terdapat fungsiData yang cara kerja nya sama dengan fungsi editData di modul update, tetapi di akhir terdapat pilihan lagi sebagai konfirmasi ulang jika user memilih y, maka data akan di hapus dan selain itu data tidak akan di hapus.

Dan yang terakhir ada fungsi updateDataBase yang menerima parameter dataBaseKaryawan yang membuka file dataBase.txt dengan mode ’w’ untuk melakukan clear dan membuka nya lagi dengan mode ’a’ untuk menuliskan data yang terdapat dalam dataBaseKaryawan ke file dataBase.txt

## 1.3. Kesimpulan dan Saran

### 1.3.1. Kesimpulan

Dari praktikum ini dapat disimpulkan :

1. Python dapat melakukan handling file dengan menggunakan funsgi open() dengan beberapa mode yaitu read, write, dan append
2. Ketika kita mengbuka sebuah file dengan open() kita juga harus menutupnya dengan close()
3. Pada mode write, apabila file yang ingin di buka belum ada, maka file tersebut akan otomatis di buat
4. Write akan melakukan clear sebelum menulis kembali data, sedangkan append samas seperti write hanya saja tidak melakukan clear pada awal
5. With adalah metode paling aman karena dapat melakukan close file secara otomatis

### 1.3.2. Saran

Saran terhadap praktikum ini yaitu:

1. Pastikan kita menggunakan mode yang tepat ketika membuka sebuah file atau melakukan handling file
2. Ketika membuka file, kita di haruskan untuk menutupnya kembali dengan fungsi close() untuk mencegah error sekaligus menyimpan data tersebut
3. Anda bisa menggunakan with agar menghindari lupa menutup file, karena with akan menutup file secara otomatis kode yang di luar indentasi nya

# DAFTAR PUSTAKA

<https://dqlab.id/alternatif-mode-operasi-python-untuk-handling-file>

Modul *Perkuliahan dan Praktikum Algoritma dan Pemrograman* Oleh Ilhamsyah, S.Si., M.Cs.