

**LAPORAN
TUGAS BESAR
PEMROGRAMAN FUNGSIONAL**



**Data Nilai Mata Kuliah Bahasa Indonesia
Mahasiswa Kelas SIIF-07-O**

ANGGOTA KELOMPOK:

Adelia Sannomiya	(19102064)
M. Fikri Aqsha Zulfa Ismail	(19102136)
Risang Abdurrahman Gustiwa	(19102178)

KELAS S1 IF-07-O

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2021**

KATA PENGANTAR

Pertama – tama kami panjatkan puji syukur kepada Allah Subhanawata'ala yang telah melimpahkan rahmat, hidayah serta karunia-Nya kepada penulis sehingga dapat menyelesaikan Tugas Besar Mata Kuliah Pemrograman Fungsional dengan judul Data Nilai Mata kuliah Bahasa Indonesia Mahasiswa Kelas SIIF-07-O yang merupakan implementasi dari dua studi kasus yaitu studi kasus pertama Higher Order Functions dan kedua The Functools Module.

Tujuan dari pembuatan laporan Tugas Besar ini adalah sebagai penjelasan serta dokumentasi dari praktikum yang sudah kami lakukan dalam pembuatan program untuk menghitung nilai akhir mata kuliah Bahasa Indonesia mahasiswa kelas SIIF-07-O.

Terima kasih kami ucapkan kepada Bapak Agus Priyanto selaku dosen Mata Kuliah Pemrograman Fungsional yang selama satu semester ini telah membimbing kami dan memberikan ilmunya, semoga ilmu tersebut dapat menjadi amal jariyah dan dapat bermanfaat bagi pembaca. Yang tidak kalah pentingnya terima kasih kepada rekan – rekan satu kelompok yang sudah berpikir, berdiskusi dan bekerja sama untuk menyelesaikan Tugas Besar ini.

DAFTAR ISI

BAB 1. PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Tujuan	6
BAB 2. TINJAUAN PUSTAKA	7
2.1 Higher Order Functions	7
2.1.1 Lambda	7
2.1.2 Map	7
2.1.3 Filter	8
2.1.4 Sorted	8
2.2 Functool Module	9
2.2.1 @lru_cache()	9
2.2.2 Partial()	9
2.2.3 Reduce()	9
BAB 3. PEMBAHASAN	10
3.1 Program	10
3.2 Cara Kerja	21
BAB 4. PENUTUP	22
4.1 Kesimpulan	22
DAFTAR PUSTAKA	23

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang sangat pesat mendorong kita untuk selalu kreatif, inovatif dan cepat tanggap akan perubahan teknologi. Gordon E. Moore *co-founder* Intel pada tahun 1965 memprediksi bahwa jumlah transistor pada IC(*Integrated Circuit*) akan berkembang dua kali lipat setiap dua tahun. Observasi dia yang kemudian dikenal dengan nama Moore's law ini sangat relevan tidak hanya pada bidang elektronika, tetapi juga teknologi secara umum. Salah satu contoh perkembangan teknologi yang sangat pesat adalah komputer, komputer dulunya sebuah mainframe yang bentuknya besar dan butuh ruangan untuk menyimpannya dan tidak semua orang dapat mengakses atau bahkan memilikinya, berbeda dengan komputer sekarang. Sekarang komputer ada dimana-mana ada IoT, Ponsel Pintar, atau Raspberry Pi yang bisa dibilang chip tetapi memiliki fungsional seperti komputer.

Bahasa pemrograman merupakan komponen penting dalam perkembangan teknologi saat ini. Perkembangan teknologi juga berpengaruh terhadap perkembangan bahasa pemrograman sebagai alat untuk menciptakan sebuah teknologi, berbagai macam bahasa pemrograman juga bermunculan untuk memenuhi kebutuhan industri teknologi bahkan di Indonesia juga memiliki bahasa pemrograman seperti bahasa Nusa. Paradigma dalam bahasa pemrograman juga berkembang, kebanyakan bahasa pemrograman menggunakan beberapa paradigma pemrograman untuk menyediakan pemrogram untuk menggunakan gaya pemrograman yang paling cocok bagi mereka.

Pemrograman fungsional merupakan salah satu paradigma pemrograman yang masih digunakan saat ini. Meskipun pada industri teknologi paradigma pemrograman ini kurang populer dibandingkan paradigma misalnya imperative ataupun object-oriented tetapi beberapa sektor seperti keuangan, perdagangan, komunikasi(*concurrency*), data science, machine learning, dan juga bidang akademik pemrograman fungsional sangat populer, beberapa proyek skala besar juga menggunakan paradigma pemrograman fungsional.

Python merupakan salah satu bahasa pemrograman yang mendukung paradigma pemrograman fungsional dan cukup banyak digunakan saat ini. Sebuah permasalahan yang mudah diselesaikan dengan mengekspresikan sebuah fungsi merupakan salah satu keunggulan dari paradigma pemrograman fungsional

Maka dari itu pada laporan ini kita akan mengimplementasikan fitur-fitur yang ada di paradigma pemrograman fungsional terutama Python ke permasalahan dengan data *real*. Hal ini bertujuan agar kita mengetahui bagaimana modularitas pada paradigma pemrograman fungsional sangat berguna dan kita mengetahui gambaran bagaimana konsep paradigma pemrograman fungsional berbeda dengan paradigma pemrograman lainnya.

1.2 Tujuan

Tujuan dari dibuatnya program perhitungan nilai mahasiswa

1. Mengetahui nilai rata-rata keseluruhan mahasiswa kelas O.
2. Mengetahui nilai tertinggi dan terendah mahasiswa kelas O.
3. Mencari data nilai mahasiswa dengan cara mengetik nama.
4. Mengetahui nilai setiap mahasiswa kelas O.
5. Mengetahui cara menerapkan functool pada data real.
6. Mengetahui cara menerapkan Higher-Order-Function pada data real.

BAB 2. TINJAUAN PUSTAKA

2.1 Higher Order Functions

Higher-order Functions merupakan fungsi yang menggunakan fungsi lain sebagai parameter atau sebagai hasil return. Ada tiga jenis Higher-order Functions, yaitu sebagai berikut :

- Fungsi yang menerima fungsi sebagai salah satu argumennya.
- Fungsi yang mengembalikan fungsi.
- Fungsi yang menerima fungsi dan mengembalikan fungsi.

2.1.1 Lambda

Salah satu fitur python yang cukup berguna adalah fungsi lambda (anonim). Dikatakan anonim karena fungsi ini tidak diberikan nama pada saat didefinisikan. Pada fungsi biasa kita menggunakan kata kunci *def* untuk membuatnya, sedangkan fungsi anonim ini kita menggunakan kata kunci lambda. Fungsi lambda bisa mempunyai banyak argumen, tapi hanya boleh memiliki satu ekspresi.

Sintaks fungsi lambda :

```
lambda arguments: expression : lambda x: x**2
```

2.1.2 Map

Fungsi map () berfungsi untuk mengaplikasikan satu fungsi ke semua anggota dari iterable (list, tuple dan lain-lain) dan mengembalikan hasilnya berupa objek map. Objek map ini bisa dengan mudah diubah menjadi list baru yang anggotanya berupa hasil pemrosesan dari fungsi dengan menggunakan fungsi list (), tuple () maupun set ().

Fungsi map () mengambil dua buah parameter, yaitu:

- Function – fungsi yang akan memproses setiap item dari iterable.
- Iterable – iterable yang akan diproses (dimapping) menggunakan function.

Sintaks :

```
map(function, iterable, ...)
```

2.1.3 Filter

Fungsi filter () berfungsi menyaring anggota – anggota dari iterable menggunakan sebuah fungsi untuk menguji tiap anggota dari iterable tersebut. Hasilnya adalah iterable baru yang anggotanya berasal dari iterable yang lama yang bernilai true saat pengujian.

Fungsi filter () mengambil dua buah parameter, yaitu:

- Function – fungsi untuk menguji elemen-elemen dari iterable apakah True atau False.
- Iterable – iterable yang akan difilter, bisa string, list, tuple dan lain sebagainya.

Sintaks :

```
filter(function, iterable)
```

2.1.4 Sorted

Fungsi sorted () berfungsi untuk mengurutkan suatu iterable baik secara naik maupun turun.

Fungsi sorted () memiliki tiga parameter, yaitu:

- Iterable – sequence (string, tuple, list), koleksi (set, dictionary), maupun sembarang iterator.
- Reverse (opsional) – apakah list diurut terbalik atau tidak (ascending atau descending).
- Key (opsional) – fungsi kunci yang digunakan untuk perbandingan.

Sintaks :

```
sorted(iterable[, key][, reverse])
```


2.2 Functool Module

2.2.1 @lru_cache()

`lru_cache ()` adalah dekorator, yang membungkus fungsi dengan memoizing callable digunakan untuk menyimpan hingga `maxsize` yang hasil pemanggilan fungsi dan kembali nilai yang disimpan jika fungsi ini disebut dengan argumen yang sama lagi. Ini dapat menghemat waktu ketika fungsi mahal atau terikat I / O dipanggil secara berkala dengan argumen yang sama.

Contoh utilitas `lru_cache ()` dapat ditampilkan dalam mengoptimalkan kode yang menghasilkan faktorial angka

2.2.2 Partial()

Fungsi parsial adalah fungsi turunan yang memiliki beberapa parameter masukan yang telah ditetapkan sebelumnya. Misalnya, jika sebuah fungsi mengambil dua parameter mengatakan " **a**" dan " **b**" , fungsi parsial dapat dibuat darinya yang memiliki " **a**" sebagai argumen yang telah diisi dan kemudian dapat dipanggil dengan " **b**" sebagai satu-satunya parameter . Functool's `partial ()` digunakan untuk membuat fungsi / objek parsial dan ini adalah fitur yang berguna karena memungkinkan untuk:

1. Replikasi fungsi yang sudah ada dengan beberapa argumen sudah diteruskan.
2. Pembuatan versi terbaru dari fungsi yang ada dengan cara yang terdokumentasi dengan baik.

2.2.3 Reduce()

Fungsi **reduce()** menerima dua argumen, fungsi dan iterable. Ini menerapkan fungsi argumen secara kumulatif ke semua elemen iterable dari kiri ke kanan dan kemudian mengembalikan satu nilai.

Misalnya, `reduce ()` dapat digunakan untuk menghitung jumlah atau produk dari daftar dengan mudah.

Sintaks :

`reduce(fungsi, urutan)`

BAB 3. PEMBAHASAN

3.1 Program

```
#####  
#                                                                 #  
#   TUGAS BESAR IMPLEMENTASI STUDI KASUS :                       #  
#       1. Higher Order Function                                #  
#           > lambda                                           #  
#           > map                                              #  
#           > filter                                           #  
#           > sorted                                           #  
#       2. Functools Module                                    #  
#           > @lru_cache()                                     #  
#           > partial()                                       #  
#           > reduce()                                         #  
#   IMPLEMENTASI STUDI KASUS DATA MATA KULIAH KELAS SIIF-07-O   #  
#                                                                 #  
#####  
  
import functools  
import operator  
import os  
from prettytable import PrettyTable  
  
# element tuple :  
# element 0 = presensi ----> bobot 10%  
# element 1 = nilai tugas ---> bobot 30%  
# element 2 = nilai UTS ---> bobot 25%  
# element 3 = nilai UAS ----> bobot 35%  
  
# data nilai presensi, tugas, uts, uas, nama mahasiswa  
student_data = {  
    (100, 70, 77, 90, "Hikmah"), (100, 85, 87, 90, "Kiel"), (100, 75, 80, 72, "Eleven"),  
    (100, 86, 81, 75, "Shodik"), (100, 79, 75, 80, "Satrio"),  
    (90, 85, 88, 75, "Fadh"), (100, 80, 89, 83, "Nanda"), (80, 70, 70, 87, "Wildan"), (100,  
80, 77, 94, "Adel"), (100, 87, 78, 87, "Andriawan"),  
    (100, 80, 80, 86, "Faishal"), (100, 71, 77, 86, "Yuri"), (100, 70, 71, 80, "Nafis"),  
    (80, 75, 70, 89, "Lorance"), (85, 80, 72, 77, "Adi"),  
    (100, 80, 73, 80, "Kaka"), (90, 75, 70, 85, "Hersa"), (100, 78, 76, 80, "Attar"), (100,  
80, 72, 84, "Vindi"), (100, 85, 73, 72, "M Arif"),  
    (100, 76, 80, 80, "Bagus"), (100, 74, 81, 87, "Fikri"), (100, 74, 83, 80, "Fahri"),  
    (100, 79, 71, 81, "Herwin"), (100, 70, 78, 87, "Imam"),  
    (100, 75, 76, 87, "Pandu"), (100, 78, 90, 70, "Risang"), (100, 80, 80, 86, "Ulinda"),  
    (90, 75, 91, 85, "Salsa"), (95, 75, 92, 79, "Fakhri"),  
    (100, 70, 80, 90, "Putri"), (100, 75, 81, 87, "Ma'arif"), (100, 73, 91, 85, "Martin"),  
    (100, 76, 92, 72, "Galuh"), (100, 74, 81, 80, "Ihza"),  
    (100, 70, 78, 80, "Rangga"), (80, 72, 81, 87, "Nurdin"), (85, 70, 90, 86, "Andika"),  
    (80, 70, 91, 87, "Syifa"), (90, 70, 92, 88, "Bastian"),  
    (100, 70, 90, 87, "Hafid"), (85, 70, 88, 82, "Maldini"), (80, 77, 81, 87, "Aldi"),  
    (100, 76, 80, 80, "Regi"), (100, 76, 87, 80, "Annisa"),  
    (100, 75, 92, 80, "Trias"), (100, 70, 91, 89, "Kevin"), (80, 70, 91, 87, "Atha")  
}  
  
# hitung nilai keseluruhan data  
total_grade = sorted(list(map(lambda x: ((x[0] * 0.10) + (x[1] * 0.30) + (x[2] * 0.25) +  
(x[3] * 0.35)), x[4]), student_data)), reverse=True)  
  
# jumlahkan semua nilai di total grade
```

```

jumlah = functools.reduce(operator.add, [x[0] for x in total_grade])

# clear screen
def clear():
    os.system('cls' if os.name == 'nt' else 'clear')

# tampilkan data total_grade
def show_data(data_arg) :
    data_table = PrettyTable()
    data_table.field_names = ['Nama', 'Nilai']
    print('===== Data Nilai Akhir =====')
    t_grade = [[j, round(i, 1)] for i, j in data_arg]
    for i in t_grade:
        data_table.add_rows([i])
    print(data_table)

# menu utama
def show_menu():
    clear()
    print("=====")
    print("= Hasil Akhir Nilai Mata Kuliah Bahasa Indonesia      =")
    print("=====")
    print("= 1. Tampil Data                                           =")
    print("= 2. Cari Data                                             =")
    print("= 3. Nilai Tertinggi                                       =")
    print("= 4. Nilai Terendah                                        =")
    print("= 5. Rata - Rata                                           =")
    print("= 6. Keluar                                                =")
    print("=====")
    pilihan = input("Masukkan pilihan : ")
    return pilihan

# kembali ke menu
def back_menu():
    print("\n")
    input("Tekan enter untuk kembali...")
    entry_point()

# entry point program
def entry_point():
    while True:
        pilihan = show_menu()
        if (pilihan == "1"): # pilihan pertama tampilkan seluruh data
            clear()
            show_data(total_grade)
            back_menu()
        elif (pilihan == "2"): # pilihan kedua cari data berdasarkan nama
            clear()
            key = input("Masukkan nama : ") # key pencarian
            # cari berdasarkan elemen ke-1
            cari = list(filter(lambda x: x if x[1] == key else False, total_grade))
            show_data(cari)
            back_menu()
        elif (pilihan == "3"): # cari nilai maksimum data, tampilkan nama dan nilai
            clear()
            # cari nilai maksimum berdasarkan elemen ke-0
            maximum_val = max(total_grade, key=lambda x: x[0])
            maximum_val = [maximum_val]
            print("===== nilai tertinggi =====")

```

```

        show_data(maximum_val)
        back_menu()
    elif (pilihan == "4"): # cari nilai minimum data, tampilkan nama dan nilai
        clear()
        # cari nilai minimum berdasarkan elemen ke-0
        minimum_val = min(total_grade, key=lambda x: x[0])
        minimum_val = [minimum_val]
        show_data(minimum_val)
        back_menu()
    elif (pilihan == "5"): # cari nilai rata-rata semua data
        clear()
        table_rerata = PrettyTable()
        table_rerata.field_names = ['Rata - rata']
        rerata = lambda x, y: x / y # cari rata-rata
        rerata = round(rerata(jumlah, 48), 2)
        table_rerata.add_row([rerata])
        print(table_rerata)
        back_menu()
    elif(pilihan == "6"): # keluar dari program
        print("Program keluar")
        exit()
    else:
        print("Pilihan tidak tersedia") # jika pilihan tidak tersedia
        back_menu()

entry_point()

```

a. import modules

```

import functools
import operator
import os
from prettytable import PrettyTable

```

Di sini modul diimport ke dalam program terlebih dahulu. Modul-modul ini memiliki fungsi-fungsi yang akan digunakan untuk melakukan manipulasi data atau tingkah laku program.

- **functools**

Modul ini untuk memanggil higher-order *function* fungsi yang mengambil sebagai argumen atau mengembalikan fungsi lain yang dibutuhkan dalam proyek tugas besar ini, seperti `reduce()`, `@lru_cache`, atau `partial()`.

- **operator**

Modul operator menyediakan fungsi yang sama seperti operator di bahasa Python. Fungsi ini sangat berguna terutama ketika objek *callable* akan di simpan, sebagai argumen, atau dikembalikan sebagai *function result*. Beberapa contoh fungsi yang ada di modul ini diantaranya

operator.add() untuk penjumlahan, operator.mul() untuk perkalian, operator.pow() untuk pemangkatan, dan lain sebagainya.

- os

Modul os menyediakan fungsionalitas untuk melakukan tugas-tugas sistem operasi. Misalnya di modul ini menyediakan fungsi untuk membuat dan menghapus direktori(folder), mengambil isinya, mengubah dan mengidentifikasi direktori, contoh lainnya melakukan *clear screen* pada terminal atau *command prompt*.

- PrettyTable

PrettyTable adalah library python yang digunakan untuk menghasilkan tabel ASCII sederhana.

b. Data mahasiswa

```
# data nilai presensi, tugas, uts, uas, nama mahasiswa
student_data = {
    (100, 70, 77, 90, "Hikmah"), (100, 85, 87, 90, "Kiel"), (100, 75, 80, 72,
"Eleven"), (100, 86, 81, 75, "Shodik"), (100, 79, 75, 80, "Satrio"),
    (90, 85, 88, 75, "Fadh"), (100, 80, 89, 83, "Nanda"), (80, 70, 70, 87,
"Wildan"), (100, 80, 77, 94, "Adel"), (100, 87, 78, 87, "Andriawan"),
    (100, 80, 80, 86, "Faishal"), (100, 71, 77, 86, "Yuri"), (100, 70, 71, 80,
"Nafis"), (80, 75, 70, 89, "Lorance"), (85, 80, 72, 77, "Adi"),
    (100, 80, 73, 80, "Kaka"), (90, 75, 70, 85, "Hersa"), (100, 78, 76, 80,
"Attar"), (100, 80, 72, 84, "Vindi"), (100, 85, 73, 72, "M Arif"),
    (100, 76, 80, 80, "Bagus"), (100, 74, 81, 87, "Fikri"), (100, 74, 83, 80,
"Fahri"), (100, 79, 71, 81, "Herwin"), (100, 70, 78, 87, "Imam"),
    (100, 75, 76, 87, "Pandu"), (100, 78, 90, 70, "Risang"), (100, 80, 80, 86,
"Ulinda"), (90, 75, 91, 85, "Salsa"), (95, 75, 92, 79, "Fakhri"),
    (100, 70, 80, 90, "Putri"), (100, 75, 81, 87, "Ma'arif"), (100, 73, 91, 85,
"Martin"), (100, 76, 92, 72, "Galuh"), (100, 74, 81, 80, "Ihza"),
    (100, 70, 78, 80, "Rangga"), (80, 72, 81, 87, "Nurdin"), (85, 70, 90, 86,
"Andika"), (80, 70, 91, 87, "Syifa"), (90, 70, 92, 88, "Bastian"),
    (100, 70, 90, 87, "Hafid"), (85, 70, 88, 82, "Maldini"), (80, 77, 81, 87,
"Aldi"), (100, 76, 80, 80, "Regi"), (100, 76, 87, 80, "Annisa"),
    (100, 75, 92, 80, "Trias"), (100, 70, 91, 89, "Kevin"), (80, 70, 91, 87,
"Atha")
}
```

Student_data adalah sebuah set yang didalamnya berisi data-data mahasiswa. Set student_data berisi kumpulan tuple-tuple yang mana elemen di dalam tuple tersebut data-datanya sebagai berikut.

- Elemen tuple ke-0 atau elemen pertama adalah nilai presensi matakuliah bahasa indonesia mahasiswa kelas S1IF-07-O.

- Elemen tuple ke-1 atau elemen kedua adalah nilai tugas matakuliah bahasa indonesia kelas S1IF-07-O.
- Elemen tuple ke-2 atau elemen ketiga adalah nilai UTS matakuliah bahasa indonesia mahasiswa kelas S1IF-07-O.
- Elemen ke-3 atau elemen keempat adalah nilai UAS matakuliah bahasa indonesia mahasiswa kelas S1IF-07-O.
- Elemen ke-4 atau elemen kelima adalah nama mahasiswa kelas S1IF-07-O.

c. Hitung nilai keseluruhan data

```
# hitung nilai keseluruhan data
total_grade = sorted(list(map(lambda x: ((x[0] *
0.10) + (x[1] * 0.30) + (x[2] * 0.25) + (x[3] * 0.35)),
x[4]), student_data)), reverse=True)
```

Di sini program akan menghitung nilai presensi, tugas, uts, dan uas mahasiswa yang ada di set student data. Map() dalam hal ini berfungsi untuk menghitung nilai dari tuple-tuple di dalam set dengan bantuan fungsi lambda. Fungsi lambda akan menghitung nilai tuple pada elemen 0-3 yaitu nilai presensi, uts, dan uas dikalikan dengan bobot tiap nilai kemudian dijumlahkan ketiga nilai tersebut sehingga menjadi nilai akhir, lambda akan mengembalikan nilai berupa objek tuple yang memiliki 2 elemen saja, elemen ke-0 nilai akhir dan elemen ke-1 nama mahasiswa.

d. Jumlahkan nilai

```
# jumlahkan semua nilai di total grade
jumlah = functools.reduce(operator.add, [x[0] for x
in total_grade])
```

Setelah dihasilkan nilai akhir dari mahasiswa, di sini program akan menghitung jumlah semua nilai akhir dari mahasiswa yaitu nilai pada elemen pertama pada tiap tuple. Jumlah di sini berfungsi untuk nilai untuk mencari rata - rata.

e. Clear Screen

```
def clear():
    os.system('cls' if os.name == 'nt' else 'clear')
```

Fungsi clear screen digunakan untuk menghapus code sebelumnya. Terdapat fungsi dengan nama clear. Os.system digunakan untuk membuat efek clear screen pada program python.

f. Tampilkan data

```
# tampilkan data total_grade
def show_data(data_arg) :
    data_table = PrettyTable()
    data_table.field_names = ['Nama', 'Nilai']
    print('===== Data Nilai Akhir
=====')
    t_grade = [[j, round(i, 1)] for i, j in
data_arg]
    for i in t_grade:
        data_table.add_rows([i])
    print(data_table)
```

Fungsi show_data akan menampilkan data dari argumen data_arg. Data yang ditampilkan adalah berupa tabel yang memiliki dua kolom yaitu Nama dan Nilai. Variable t_grade adalah *list comprehension* nilai mahasiswa yang berupa tuple dengan elemen ke-0 adalah nilai dan elemen ke-1 adalah nama akan dibalik elemen ke-0 nama dan elemen ke-1 adalah nilai. Fungsi round akan memberikan satu digit angka dibelakang koma. Kemudian satu per satu nilai di dalam t_grade ditambahkan ke dalam tabel, kemudian tabel ditampilkan.

g. Tampilan menu

```
def show_menu() :
    clear()
    print("=====")
    print("= Hasil Akhir Nilai Mata Kuliah Bahasa Indonesia      =")
    print("=====")
    print("= 1. Tampil Data                                           =")
    print("= 2. Cari Data                                             =")
    print("= 3. Nilai Tertinggi                                       =")
    print("= 4. Nilai Terendah                                        =")
    print("= 5. Rata - Rata                                           =")
    print("= 6. Keluar                                                =")
    print("=====")
    pilihan = input("Masukkan pilihan : ")
    return pilihan
```

Untuk menampilkan menu disini dibuat sebuah fungsi agar lebih mudah dalam pemanggilannya. Setelah deklarasi fungsi menu terdapat fungsi clear

untuk menghapus code sebelumnya. Setelah itu terdapat print yang berisi tulisan dan sedikit dekorasi seperti tabel. Terdapat 6 pilihan pada menu dan di bawah menu tersebut terdapat input untuk memilih nomor pada menu agar dapat melihat hasilnya serta tidak lupa return untuk mengembalikan nilainya.

h. Back menu

```
def back_menu():  
    print("\n")  
    input("Tekan enter untuk kembali...")  
    entry_point()
```

Terdapat fungsi dengan nama back menu. Setelah itu terdapat fungsi \n untuk ganti baris selanjutnya. Kemudian terdapat input berupa “enter” yang digunakan untuk kembali ke menu utama dari submenu yang dipilih. Setelah itu terdapat entry point yang berfungsi fungsi pertama saat program dijalankan.

i. Entry point dan pilihan nomor 1

```
def entry_point():  
    while True:  
        pilihan = show_menu()  
        if (pilihan == "1"):  
            clear()  
            show_data(total_grade)  
            back_menu()
```

Terdapat fungsi bernama entry point berfungsi sebagai main program. Kemudian perulangan while dengan nilai true. Terdapat percabangan menggunakan fungsi if. Pada pilihan pertama sebelum menampilkan data nilai akhir serta nama mahasiswa, ada fungsi clear screen. Setelah data tampil diberikan fungsi back menu agar bisa kembali ke menu utama.

j. Pilihan menu nomor 2

```
elif (pilihan == "2"):  
    clear()  
    key = input("Masukkan nama : ")  
    cari = list(filter(lambda x: x if x[1]  
== key else False, total_grade))  
    show_data(cari)  
    back_menu()
```

Terdapat kondisi jika pilihan bernilai 2. Terdapat fungsi clear screen untuk menghapus code sebelumnya. Inisialisasi key dengan input berfungsi untuk mengetik nama mahasiswa yang akan dicari nilainya. Kemudian variabel

cari yang terdapat list untuk menampilkan data dalam bentuk list, filter untuk menyaring anggota iterable dan lambda yang di dalamnya terdapat expression dengan kondisi tersebut serta variabel total grade untuk mengambil nilainya dari situ. Setelah itu terdapat fungsi show data dengan variabel cari untuk menampilkan data berupa nama mahasiswa dan nilai akhirnya. Setelah data tampil diberikan fungsi back menu agar bisa kembali ke menu utama.

k. Pilihan menu nomor 3

```
elif (pilihan == "3"):  
    clear()  
    maximum_val = max(total_grade,  
key=lambda x: x[0])  
    maximum_val = [maximum_val]  
    print("===== nilai tertinggi  
=====")  
    show_data(maximum_val)  
    back_menu()
```

Terdapat kondisi apabila input pilihan bernilai 3. Terdapat fungsi clear screen untuk menghapus code sebelumnya. Variabel maximum dengan pemanggilan fungsi max yang di dalamnya terdapat total grade untuk mengambil nilai dari variabel tersebut serta key dengan lambda memiliki expression index ke 0 dari x. Variabel maximum val sama dengan fungsi maximum untuk mengambil nilai dari variabel tersebut. Kemudian terdapat fungsi show data dengan variabel maximum val untuk menampilkan nama mahasiswa dengan nilai akhir tertinggi. Setelah data tampil diberikan fungsi back menu agar bisa kembali ke menu utama.

l. Pilihan menu nomor 4

```
elif (pilihan == "4"):  
    clear()  
    minimum_val = min(total_grade,  
key=lambda x: x[0])  
    minimum_val = [minimum_val]  
    show_data(minimum_val)  
    back_menu()
```

Terdapat kondisi apabila input pilihan bernilai 4. Terdapat fungsi clear screen untuk menghapus code sebelumnya. Variabel minimum dengan pemanggilan fungsi min yang di dalamnya terdapat total grade untuk mengambil nilai dari variabel tersebut serta key dengan lambda memiliki expression index ke 0 dari x. Variabel minimum val sama dengan fungsi minimum untuk mengambil nilai dari variabel tersebut. Kemudian terdapat fungsi show data dengan variabel minimum val untuk menampilkan nama

mahasiswa dengan nilai akhir terendah. Setelah data tampil diberikan fungsi back menu agar bisa kembali ke menu utama.

m. Pilihan menu nomor 5

```
elif (pilihan == "5"):  
    clear()  
    table_rerata = PrettyTable()  
    table_rerata.field_names = ['Rata -  
rata']  
  
    rerata = lambda x, y: x / y  
    rerata = round(rerata(jumlah, 48), 2)  
    table_rerata.add_row([rerata])  
    print(table_rerata)  
    back_menu()
```

Pada menu ini program akan menghitung nilai rata-rata dari semua nilai akhir mahasiswa. Nilai rata-rata ini dihitung menggunakan fungsi lambda yang menerima dua argumen x dan y kemudian mengembalikan nilai berupa hasil pembagian dari x dan y. Perhitungan rata-rata dilakukan dengan memasukkan argumen berupa variabel jumlah yang nilainya adalah jumlah seluruh nilai akhir dari mahasiswa dan argumen kedua adalah bilangan 48 banyak mahasiswa kelas S1IF-07-O, fungsi round selanjutnya akan menampilkan dua digit angka setelah koma. Setelah rata-rata didapatkan maka di tampilkan adalah nilai rata-rata dari nilai akhir mahasiswa yang dimasukkan ke dalam tabel yang memiliki satu kolom Rata-rata.

n. Pilihan menu nomor 6

```
elif(pilihan == "6"):  
    print("Program keluar")  
    exit()
```

Pada pilihan keenam program atau proses berhenti. Artinya user telah keluar dari program atau memberhentikan program. Exit() adalah sebuah *method* untuk memberhentikan sebuah program atau proses.

```

=====
= Hasil Akhir Nilai Mata Kuliah Bahasa Indonesia =
=====
= 1. Tampil Data =
= 2. Cari Data =
= 3. Nilai Tertinggi =
= 4. Nilai Terendah =
= 5. Rata - Rata =
= 6. Keluar =
=====

```

Gambar 3.1.2 Menu utama program

```

===== Data Nilai Akhir =====
+-----+-----+
| Nama | Nilai |
+-----+-----+
| Kiel | 88.8 |
| Adel | 86.2 |
| Andriawan | 86.0 |
| Nanda | 85.3 |
| Kevin | 84.9 |
| Martin | 84.4 |
| Ulinda | 84.1 |
| Faishal | 84.1 |
| Salsa | 84.0 |

```

Gambar 3.1.3 Pillihan nomor satu

```

Masukkan nama : Bastian
===== Data Nilai Akhir =====
+-----+-----+
| Nama | Nilai |
+-----+-----+
| Bastian | 83.8 |
+-----+-----+

Tekan enter untuk kembali...

```

Gambar 3.1.4 Pilihan nomor dua

```
===== nilai tertinggi =====  
===== Data Nilai Akhir =====  
+-----+-----+  
| Nama | Nilai |  
+-----+-----+  
| Kiel | 88.8 |  
+-----+-----+  
  
Tekan enter untuk kembali...
```

Gambar 3.1.5 Pilihan nomor tiga

```
===== Data Nilai Akhir =====  
+-----+-----+  
| Nama | Nilai |  
+-----+-----+  
| Nafis | 76.8 |  
+-----+-----+  
  
Tekan enter untuk kembali...
```

Gambar 3.1.6 Pilihan nomor empat

```
+-----+  
| Rata - rata |  
+-----+  
| 81.73 |  
+-----+  
  
Tekan enter untuk kembali...
```

Gambar 3.1.7 Pilihan nomor lima

```
=====
= Hasil Akhir Nilai Mata Kuliah Bahasa Indonesia =
=====
= 1. Tampil Data =
= 2. Cari Data =
= 3. Nilai Tertinggi =
= 4. Nilai Terendah =
= 5. Rata - Rata =
= 6. Keluar =
=====
Masukkan pilihan : 6
Program keluar

D:\documents\functional-programming\pemrograman-fungsional>
```

Gambar 3.1.8 Pilihan nomor enam

3.2 Cara Kerja

Pada menu program ini terdapat lima pilihan yang dapat dipilih oleh pengguna. Jika memilih nomor satu maka pengguna dapat melihat nilai akhir berupa rata - rata dari (nilai tugas, quiz, UTS dan UAS) mata kuliah bahasa indonesia setiap mahasiswa kelas O. Kemudian tekan enter agar kembali ke menu utama. Jika pengguna memilih nomor kedua maka akan memasukkan kata kunci nama mahasiswa yang akan dicari data nilainya, setelah tekan enter akan muncul nilai akhir mahasiswa beserta namanya. Untuk kembali ke menu utama tekan enter. Apabila pengguna memilih nomor tiga maka akan menampilkan nilai akhir tertinggi mahasiswa beserta namanya. Tekan enter untuk kembali ke menu utama. Apabila pengguna memilih nomor empat maka akan menampilkan nilai akhir terendah mahasiswa beserta namanya. Tekan enter untuk kembali ke menu utama. Jika pengguna memilih nomor lima maka akan menampilkan nilai rata - rata keseluruhan mahasiswa kelas O pada mata kuliah bahasa Indonesia. Tekan enter untuk kembali ke menu utama. Apabila pengguna memilih nomor enam maka akan keluar dari program. Jika pengguna memilih nomor yang tidak tersedia pada menu utama maka akan muncul kata - kata “pilihan tidak tersedia” dan disarankan untuk tekan enter agar kembali bisa memilih nomor yang tersedia.

BAB 4. PENUTUP

4.1 Kesimpulan

Dalam melakukan manipulasi data dalam python terdapat banyak fungsi yang digunakan untuk menghasilkan output sesuai dengan kebutuhan. Pada program yang kelompok kami buat, untuk higher-order functions menggunakan fungsi max, min, lambda, map, filter dan sorted. Sedangkan untuk the functools module, kelompok kami hanya menggunakan fungsi reduce. Dengan menggunakan fungsi - fungsi tersebut, kita dapat memanipulasi data nilai mahasiswa yang memiliki banyak subnilai seperti nilai tugas, quiz, UTS dan UAS. Untuk mencari nilai tertinggi dan terendah kami menggunakan fungsi max dan min. Untuk menampilkan data nilai mahasiswa secara keseluruhan beserta namanya, kami menggunakan list. Untuk mencari data nilai mahasiswa sesuai dengan keinginan pengguna, terdapat menu pencarian yang dapat digunakan yaitu dengan cara mengetik kata kunci nama mahasiswa yang tersedia. Pada pilihan pencarian kami menggunakan fungsi filter untuk menyaring anggota - anggota iterable dan lambda untuk menulis expressionnya. Untuk menampilkan nilai rata - rata keseluruhan mahasiswa kami menggunakan fungsi lambda untuk expressionnya serta fungsi round untuk menampilkan jumlah digit yang berada di belakang koma.

DAFTAR PUSTAKA

- [1] Priyanto, Agus. Modul 6 - Higher-order Functions (PDF)
- [2] Priyanto, Agus. Modul 9 - The Functools Module (PDF)
- [3] Darvishi, Arian. Pengantar : Memperkenalkan Modul Functools Python. Ichi.Pro.
[Akses : 24 Januari 2021, pukul 14:00.] <https://ichi.pro/id/post/3044253398862>