

**LAPORAN TUGAS BESAR**  
**WI-1102 BERPIKIR KOMPUTASIONAL**  
**“PREDIKSI PENJURUSAN”**

Dosen : Tricya Esterina Widagdo, S.T., M.Sc. dan Dr. Maya Nabila, S.Si., M.Si.



Disusun Oleh :  
Kelompok 11

Banu Agil Triyanto	19623046
Fakhri Athallah Dwi Andeo	19625111
Faiq Fadhlul bahri	19625107
Steve Bradley Hoeij	19625163
Muhammad Razka Adhitya	19625051

## KATA PENGANTAR

Dengan mengucapkan puji syukur kepada Tuhan Yang Maha Esa karena kasih dan karunia-Nya sehingga penulis dapat menyelesaikan laporan akhir tugas besar “Program Prediksi Penjurusan”. Laporan Tugas Besar ini disusun untuk memenuhi tugas besar mata kuliah Berpikir Komputasional (WI1102).

Penulis menyampaikan ucapan terima kasih kepada Yth. Ibu Tricya Esterina Widagdo, S.T., M.Sc. dan Dr. Maya Nabila, S.Si., [M.Si](#) selaku dosen pengampu yang telah memberikan arahan dan bimbingan kepada penulis selama proses penyelesaian laporan tugas besar ini.

Penyusunan laporan ini masih jauh dari kesempurnaan. Untuk itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan laporan tugas besar ini. Akhir kata, penulis mengucapkan terima kasih.

Sumedang, 11 Desember 2025

Tim Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iv</b>
<b>PEMBAGIAN TUGAS ANGGOTA KELOMPOK.....</b>	<b>v</b>
<b>TUGAS I.....</b>	<b>1</b>
A. Eksplorasi Program.....	1
B. Dekomposisi.....	2
C. Pengenalan Pola.....	4
D. Abstraksi.....	4
E. Algoritma.....	5
<b>TUGAS II.....</b>	<b>6</b>
A. Deskripsi Simulasi.....	6
B. Diagram Alir.....	7
<b>TUGAS III.....</b>	<b>11</b>
A. Penjelasan Program.....	11
1. Sekuens.....	11
2. Sections.....	12
3. Repetisi.....	14
4. Array.....	16
5. Subprogram.....	17
<b>PENUTUP.....</b>	<b>19</b>
A. Kesimpulan.....	19
B. Lesson Learned.....	19
C. Dokumentasi.....	19

## DAFTAR GAMBAR

Gambar 1.1 Dekomposisi Umum Program.....	3
Gambar 1.2 Dekomposisi Bagian Input IPK.....	4
Gambar 1.3 Dekomposisi Bagian Analisis Peringkat.....	4
Gambar 1.4 Dekomposisi Bagian Prediktor Penjurusan.....	5
Gambar 1.5 Dekomposisi Bagian Pemberian Rekomendasi.....	5
Gambar 2.1 Diagram Alir Keseluruhan.....	9
Gambar 2.2 Fungsi rank_index.....	9
Gambar 2.3 Fungsi open_file.....	10
Gambar 2.4 Fungsi skew_norm_cdf.....	10
Gambar 2.5 Fungsi compute_xi.....	11
Gambar 2.6 Fungsi retry_input_range.....	11
Gambar 2.7 Fungsi Phi, Erf, dan Psi.....	11
Gambar 2.8 Fungsi Owen's T.....	12
Gambar 2.9 Fungsi truncated_skew_norm_cdf.....	12
Gambar 3.1 Inisialisasi Variabel Rata-rata.....	13
Gambar 3.2 Data	14
Gambar 3.3 Input IPK Pengguna.....	14
Gambar 3.4 Mengecek Pilihan.....	15
Gambar 3.5 Cek Dataset.....	15
Gambar 3.6 Rekomendasi.....	16
Gambar 3.7 Pengulangan.....	16
Gambar 3.8 Loop membaca file.....	17
Gambar 3.9 Loop menghitung peringkat.....	17
Gambar 3.10 Loop memilih jurusan.....	17
Gambar 3.11 Array Data.....	18
Gambar 3.12 Menyimpan nama jurusan.....	18
Gambar 3.13 Menyimpan pilihan jurusan pengguna.....	18
Gambar 3.14 Menyimpan probabilitas IPK.....	19
Gambar 3.15 Fungsi Rank Index.....	19
Gambar 3.16 Fungsi Open File.....	20
Gambar 4.1 Asistensi Jumat, 5 Desember 2025.....	23

## PEMBAGIAN TUGAS ANGGOTA KELOMPOK

No.	Nama	NIM	Tugas
1.	Banu Agil Triyanto	19625095	Membantu commit algoritma di git ( memperbaiki algoritma yang error ), membantu bikin laporan, membantu flowchart, dan edit video
2.	Fakhri Athallah Dwi Andeo	19625111	Membuat Laporan dan Membantu membuat flowchart
3.	Steve Bradley Hoeij	19625163	Membantu pembuatan diagram alir utuh program, membantu pembuatan program bagian math_functions.py, membantu penulisan laporan bagian pilar-pilar CT, dan membantu pembuatan power point.
4.	Muhammad Razka Adhitya	19625051	Membantu membuat laporan dan membantu membuat ppt.
5.	Faiq Fadhlul bahri	19625107	Membantu membuat laporan dan membantu membuat ppt.

# TUGAS I

## A. Eksplorasi Program

Sistem Analisis Peluang Penjurusan adalah perangkat lunak berbasis Python yang dirancang untuk membantu mahasiswa Sekolah Teknik Elektro dan Informatika Komputasi (STEI-K) dalam memperkirakan posisi akademik mereka terkait pemilihan jurusan. Pengguna cukup memasukkan Indeks Prestasi Kumulatif (IPK) mereka, kemudian sistem akan mengolah data tersebut menggunakan metode statistik untuk memberikan peringkat, probabilitas kelulusan, serta rekomendasi akademik yang relevan. Sistem ini bekerja dengan membandingkan data pengguna terhadap dataset historis mahasiswa (Informatika dan Sistem Teknologi Informasi). Sistem ini dirancang secara modular, memisahkan logika matematika, utilitas program, dan penanganan data utama. Secara umum, alur kerja program ini mencakup input data pengguna, pemrosesan statistik menggunakan distribusi *Truncated Skew Normal*, dan penyajian output berupa data numerik maupun saran kualitatif. Berikut adalah fitur dan tahapan utama dalam sistem ini

### 1. Input Data Pengguna

Pada tampilan awal, sistem akan meminta pengguna untuk memasukkan data Indeks Prestasi Kumulatif (IPK).

### 2. Validasi Input Sistem

Digunakan modul `prog_utils.py` untuk memvalidasi input (berupa angka *float* antara 0,00 hingga 4,00). Jika input di luar jangkauan, sistem akan meminta input ulang.

Menu Utama & Fitur Analisis, setelah data IPK diterima, pengguna disuguhkan menu interaktif dengan tiga fitur utama:

### 1. Analisis Peringkat (Ranking)

- Informatika (Ganesha & Jatinangor)
- Sistem dan Teknologi Informasi (Ganesha & Jatinangor)
- Total gabungan seluruh mahasiswa

Sistem akan membaca file eksternal (seperti “`data_ipk_total.csv`”), membandingkan IPK pengguna dengan data yang ada, lalu menampilkan peringkat pengguna (misal: Peringkat 50 dari 285 mahasiswa).

## 2. Menghitung Peluang

Pengguna akan diminta untuk mengurutkan jurusan di STEI-K sebagai urutan pilihan dari yang paling diinginkan (pilihan 1) hingga yang paling tidak diinginkan (pilihan 4). Lalu sistem menggunakan model distribusi *skew-normal* terpotong, untuk menghitung probabilitas mendapatkan jurusan tersebut dengan parameter yang disesuaikan pada rata-rata IPK masing-masing jurusan. Nilai IPK pengguna dievaluasi dalam distribusi tersebut untuk menghasilkan peluang kumulatif. Peluang tersebut juga ditampilkan dalam bentuk persentase.

## 3. Rekomendasi Akademik

Fitur rekomendasi akademik memberikan saran akademik berdasarkan posisi atau peringkat IPK pengguna terhadap distribusi IPK seluruh angkatan STEI-K. Berdasarkan nilai peluang kumulatif, pengguna diklasifikasikan ke dalam salah satu dari tiga kategori, yaitu

- Prestasi Sangat Tinggi ( $P \geq 85\%$ )
- Prestasi Baik ( $30\% \leq P < 85\%$ )
- Perlu Peningkatan Akademik ( $P < 30\%$ )

Setiap kategori disertai dengan rekomendasi akademik yang relevan serta perbandingan terhadap rata-rata IPK angkatan.

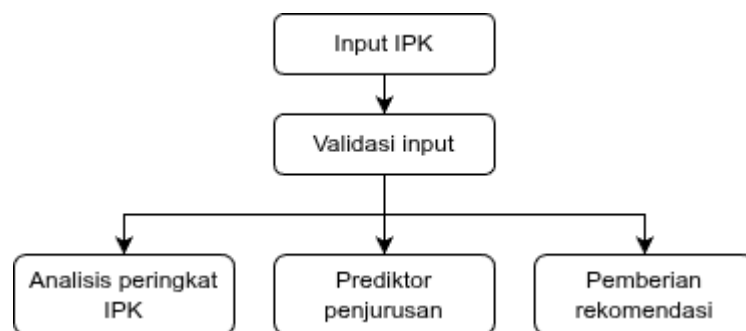
Berdasarkan eksplorasi fitur yang tersedia, program ini mampu memberikan gambaran posisi akademik mahasiswa secara kuantitatif melalui peringkat IPK, mengestimasi peluang memasuki jurusan, serta merekomendasikan strategi peningkatan prestasi akademik berbasis statistik. Pendekatan distribusi *skew-normal* terpotong memungkinkan analisis IPK yang lebih realistis karena memperhatikan batas nilai IPK. Secara keseluruhan, program berfungsi sebagai alat bantu pengambilan keputusan akademik yang informatif dan terstruktur.

## B. Dekomposisi

Dekomposisi merupakan metode untuk memecah permasalahan kompleks menjadi beberapa bagian yang lebih kecil dan terstruktur agar lebih mudah dianalisis serta diimplementasikan ke dalam program. Pada sistem prediksi penjurusan ini,

permasalahan utama berupa penentuan peluang masuk jurusan berdasarkan nilai IPK pengguna dipecah ke dalam beberapa komponen utama, yaitu pengambilan data IPK, pengolahan data statistik, perbandingan dengan data historis, perhitungan peluang, serta pemberian rekomendasi akademik. Dengan pemecahan masalah secara bertahap, sistem dapat bekerja secara sistematis dan menghasilkan output yang informatif serta mudah dipahami oleh pengguna.

## 1. Dekomposisi Umum



**Gambar 1.1** Dekomposisi Umum Program

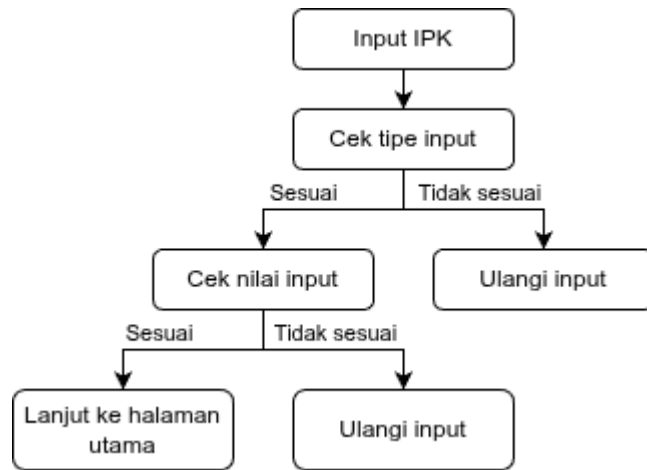
Secara umum, proses dalam sistem prediksi penjurusan dimulai dari masukan nilai IPK sebagai data awal. Data IPK ini kemudian divalidasi untuk memastikan bahwa nilai sudah berada dalam rentang yang sesuai. Setelah itu, sistem menampilkan menu utama untuk memilih fitur analisis yang diinginkan, seperti analisis peringkat IPK, perhitungan peluang memasuki jurusan, serta pemberian rekomendasi akademik. Kemudian, sistem akan memproses data IPK pengguna dengan membandingkannya terhadap data historis IPK mahasiswa STEI-K yang tersimpan dalam lima berkas CSV, yaitu “data\_ipk\_if\_g.csv”, “data\_ipk\_if\_j.csv”, “data\_ipk\_sti\_g.csv”, “data\_ipk\_sti\_j.csv”, dan “data\_ipk\_total.csv”. Hasil pengolahan tersebut ditampilkan dalam bentuk peringkat, persentase peluang, serta kategori rekomendasi akademik. Proses ini akan terus berulang selama pengguna tidak memilih untuk keluar dari program.

## 2. Dekomposisi Bagian

Pada tingkat yang lebih rinci, sistem prediksi penjurusan dapat diuraikan ke dalam beberapa bagian fungsional:

### a. Pengambilan dan Validasi Data IPK

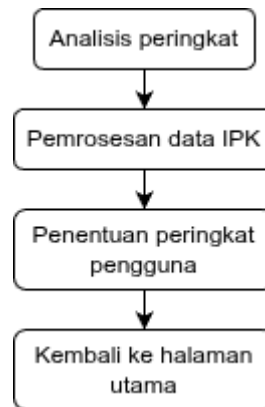




**Gambar 1.2** Dekomposisi Bagian Input IPK

Pada tahap awal, sistem menerima input nilai IPK dari pengguna. Selanjutnya dilakukan proses validasi untuk memastikan bahwa IPK yang dimasukkan berupa bilangan desimal dengan rentang 0,00 hingga 4,00. Jika input tidak valid, sistem akan meminta pengguna untuk menginput ulang hingga diperoleh masukan yang sesuai.

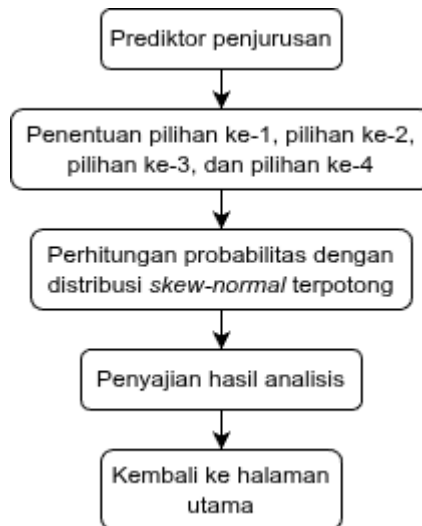
b. Analisis Peringkat (*Ranking*)



**Gambar 1.3** Dekomposisi Bagian Analisis Peringkat

Setelah masukan IPK valid, sistem membaca data IPK dari berkas CSV. Nilai IPK pengguna kemudian dibandingkan dengan seluruh data yang tersedia untuk menentukan posisi peringkat pengguna, baik pada tingkat jurusan maupun pada tingkat gabungan seluruh mahasiswa STEI-K.

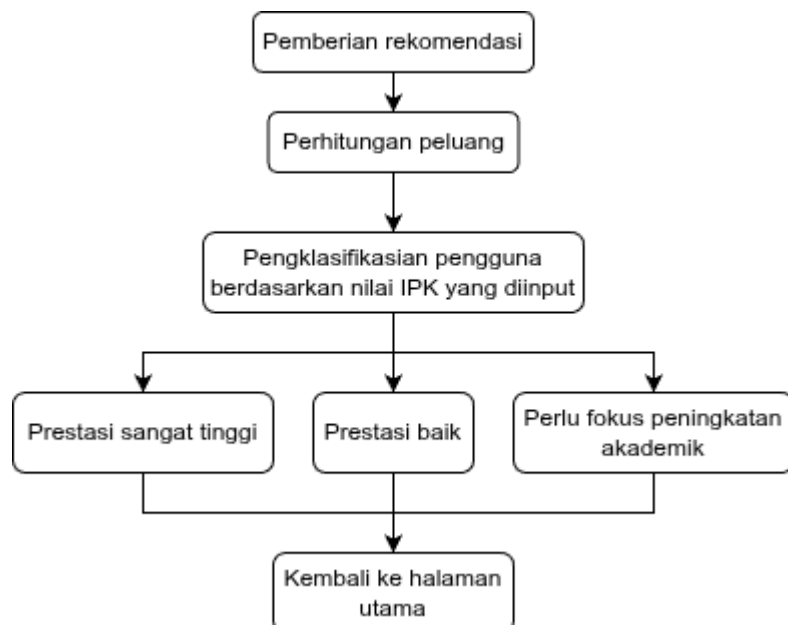
c. Prediktor Penjurusan



**Gambar 1.4** Dekomposisi Bagian Prediktor Penjurusan

Pada bagian ini, pengguna mengurutkan jurusan berdasarkan tingkat ketertarikan. Sistem kemudian menggunakan model distribusi statistik *skew-normal* terpotong untuk menghitung peluang diterima pada masing-masing jurusan sesuai dengan parameter distribusi IPK jurusan tersebut. Hasil perhitungan ditampilkan dalam bentuk persentase peluang.

d. Pemberian Rekomendasi Akademik



**Gambar 1.5** Dekomposisi Bagian Pemberian Rekomendasi

Berdasarkan hasil peringkat dan peluang yang diperoleh, sistem mengklasifikasikan pengguna ke dalam kategori prestasi tertentu, yaitu

prestasi sangat tinggi, baik, atau perlu peningkatan. Setiap kategori disertai dengan rekomendasi akademik yang bertujuan membantu pengguna dalam meningkatkan atau mempertahankan performa akademiknya.

### **C. Pengenalan Pola**

Pada tahap pengenalan pola, sistem akan mencoba untuk menemukan hubungan antara masukan dan data yang tersedia. Pola yang dianalisis meliputi tren rata-rata IPK pada setiap jurusan serta hubungan antara nilai IPK dengan peluang diterima di masing-masing jurusan. Melalui pengamatan pola tersebut, dapat diketahui bahwa setiap jurusan memiliki karakteristik distribusi IPK yang berbeda-beda serta tingkat persaingan yang bervariasi. Pola-pola inilah yang kemudian dijadikan dasar dalam memahami kecenderungan data dan menentukan metode perhitungan yang paling tepat dalam proses estimasi peluang masuk jurusan agar hasil yang diperoleh lebih akurat dan representatif.

### **D. Abstraksi**

Pada tahap abstraksi, informasi dunia nyata yang bersifat kompleks disederhanakan menjadi elemen-elemen inti yang relevan dengan tujuan sistem. Nilai akademik mahasiswa direpresentasikan dalam satuan angka IPK, sementara keseluruhan data historis disusun dalam bentuk daftar nilai yang mudah diolah oleh program. Hasil analisis yang semula berupa perhitungan statistik yang kompleks kemudian diringkas menjadi kategori-kategori prestasi tertentu. Proses penyederhanaan ini memungkinkan sistem untuk berfokus pada informasi yang benar-benar penting dalam pengambilan keputusan akademik, sehingga hasil yang diberikan menjadi lebih jelas, efisien, dan mudah dipahami oleh pengguna.

### **E. Algoritma**

Pada tahap algoritma, sistem disusun berdasarkan langkah-langkah berpikir yang teratur dan sistematis. Proses diawali dengan pengguna memasukkan nilai IPK sebagai data awal, kemudian pengguna memilih menu atau fitur analisis yang diinginkan.

Selanjutnya, sistem memproses data IPK tersebut menggunakan metode yang sesuai, menghitung peluang memasuki jurusan berdasarkan hasil pengolahan data, dan akhirnya menghasilkan keluaran berupa rekomendasi akademik. Urutan logis ini memastikan setiap tahapan saling terhubung secara runtut, sehingga output yang dihasilkan bersifat konsisten, dapat diprediksi, serta mendukung pengambilan keputusan secara sistematis.

## **TUGAS II**

### **A. Deskripsi Simulasi**

Pengguna memasukkan nilai IPK-nya ke dalam program, lalu program akan menampilkan menu utama dengan input, proses, dan output sebagai berikut:

#### **a. Input**

Pengguna akan diberikan empat pilihan dalam layar utama:

1. Menampilkan nilai indeks angkatan
2. Menghitung peluang memasuki jurusan
3. Memberikan rekomendasi
4. Keluar

#### **b. Proses**

Program akan membaca input pengguna.

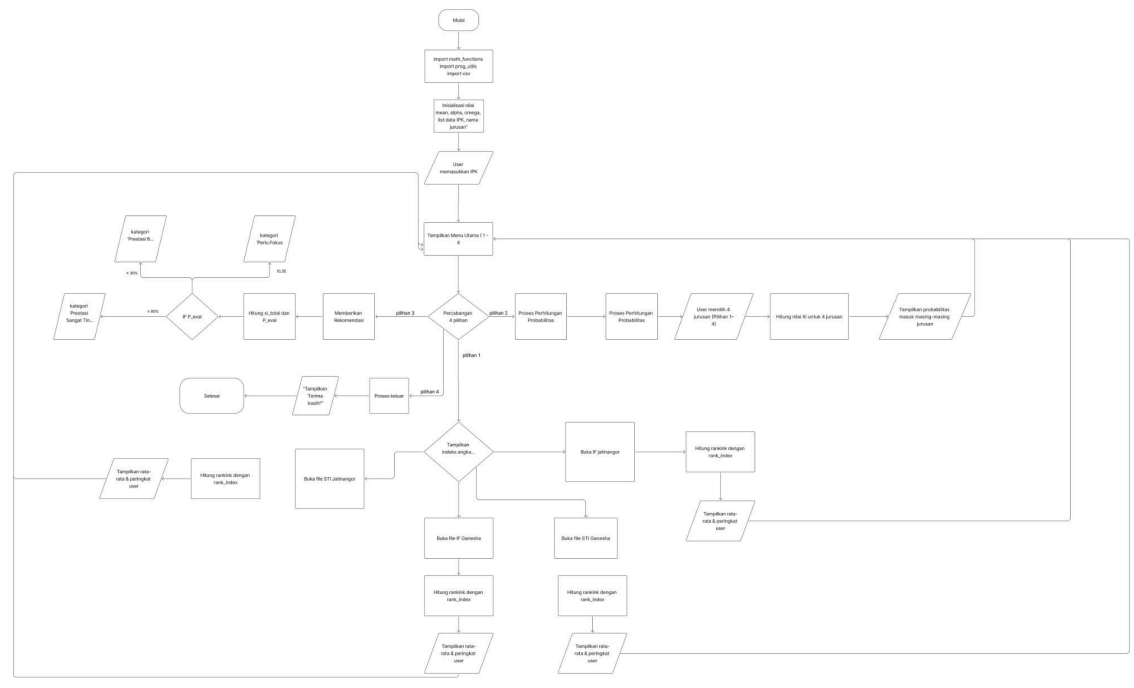
1. Pilihan 1: pengguna diarahkan untuk memilih menampilkan nilai indeks setiap jurusan di STEI-K atau total gabungan untuk dibandingkan dengan IPK pengguna.
2. Pilihan 2: Pengguna akan diarahkan untuk mengurutkan pilihan jurusan sesuai dengan yang paling diinginkan, lalu program akan mengevaluasi IPK pengguna untuk memperoleh peluang memasuki jurusan.
3. Pilihan 3: Mengklasifikasikan IPK pengguna berdasarkan posisi kuantil terhadap distribusi IPK angkatan.
4. Pilihan 4: Program akan berhenti berjalan.

#### **c. Output**

1. Informasi peringkat IPK pengguna terhadap salah satu jurusan di STEI-K atau terhadap seluruh angkatan STEI-K.
2. Persentase peluang memasuki masing-masing jurusan di STEI-K sesuai pilihan.
3. Menunjukkan kategori prestasi akademik dan rekomendasi pengembangan akademik yang sesuai.
4. Program akan menampilkan pesan “Terima kasih!”

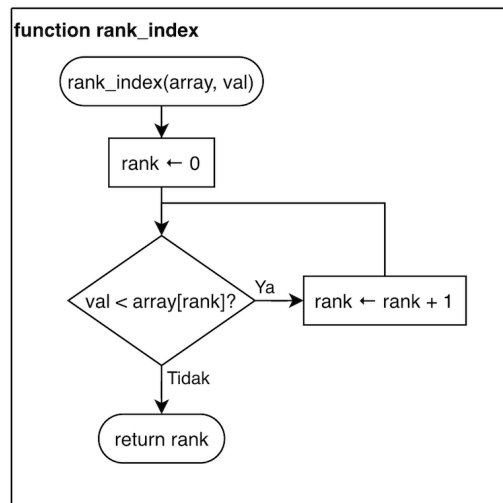
## B. Diagram Alir

Diagram alir dapat diakses pada [bit.ly/TubesII11](http://bit.ly/TubesII11)



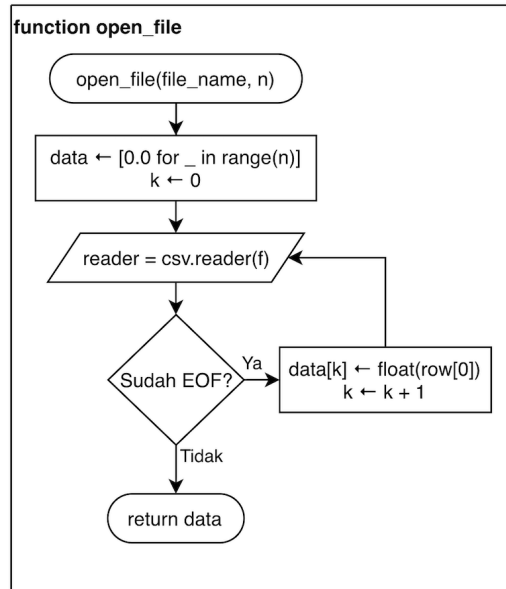
**Gambar 2.1** Diagram Alir Keseluruhan

### a. Fungsi Rank Index



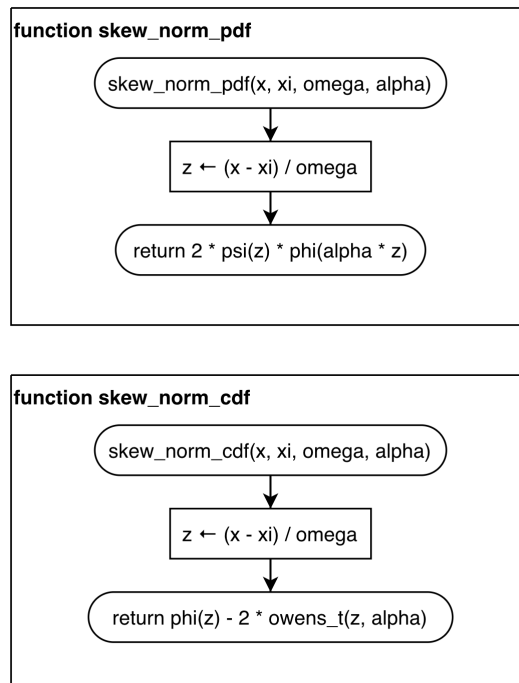
**Gambar 2.2** Fungsi rank\_index

b. Fungsi Open File



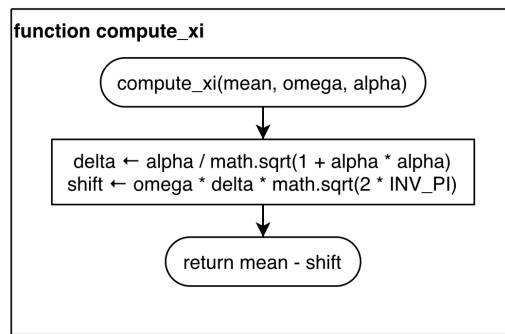
**Gambar 2.3** Fungsi open\_file

c. Fungsi Skew Norm PDF dan CDF



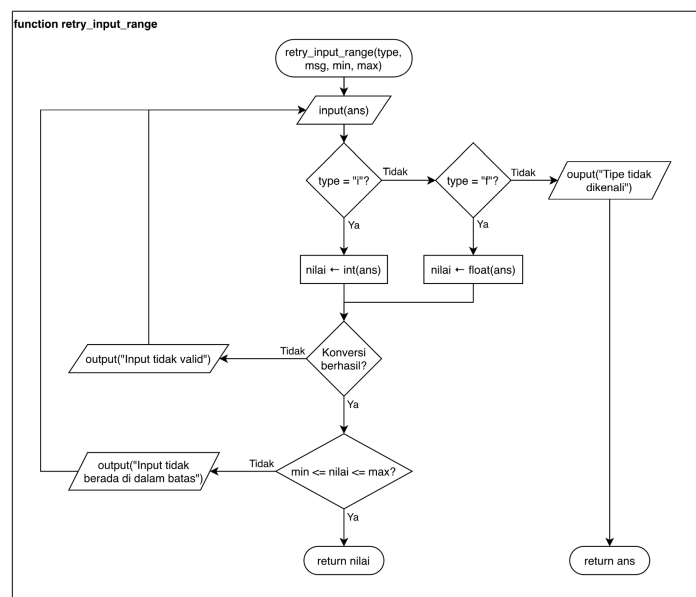
**Gambar 2.4** Fungsi skew\_norm\_cdf

d. Fungsi Compute xi



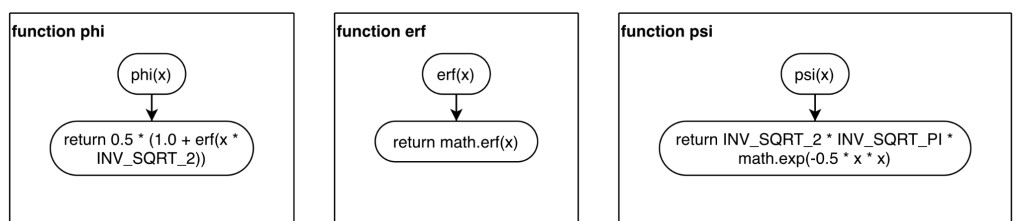
**Gambar 2.5** Fungsi compute\_xi

e. Fungsi Retry Input Range



**Gambar 2.6** Fungsi retry\_input\_range

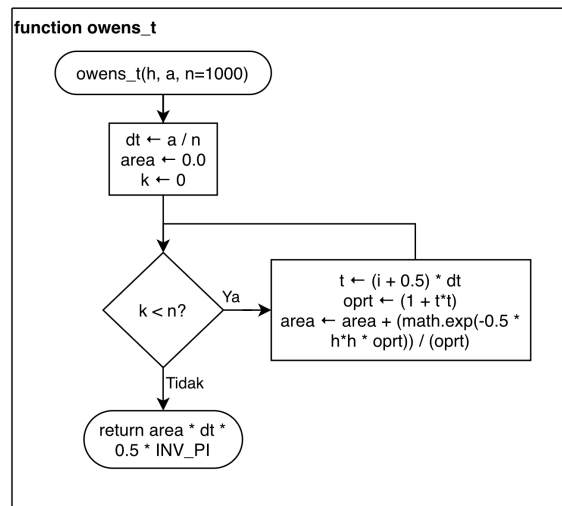
f. Fungsi phi, erf, dan psi



**Gambar 2.7** Fungsi Phi, Erf, dan Psi

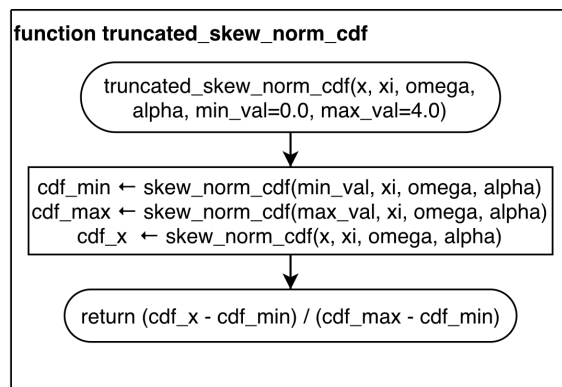
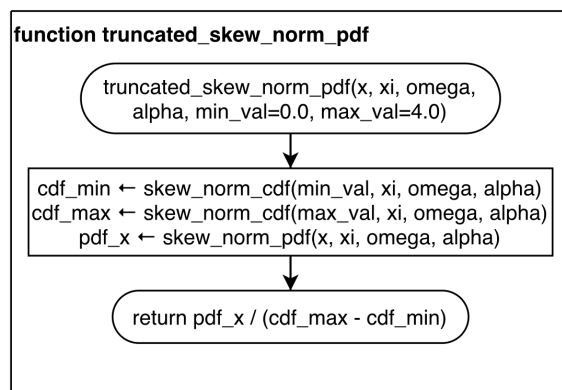


g. Fungsi Owen's T



**Gambar 2.8** Fungsi Owen's T

h. Fungsi Truncated Skew Norm PDF dan CDF



**Gambar 2.9** Fungsi truncated\_skew\_norm\_cdf

## TUGAS III

### A. Penjelasan Program

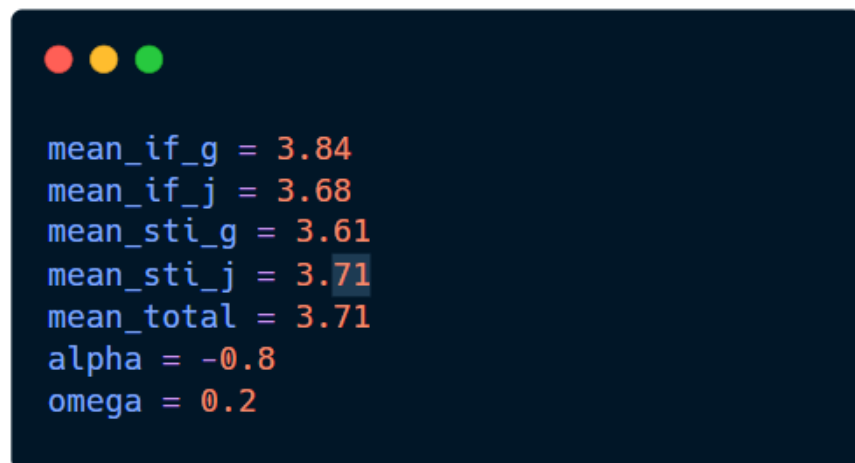
Program ini berjalan berurutan dari atas ke bawah sesuai logika utama prediktor untuk jurusan. Alur utama program adalah

- a. Pengguna memasukkan nilai IPK.
- b. Program menampilkan menu utama.
- c. Pengguna memilih salah satu fitur analisis (nilai indeks angkatan, peluang memasuki jurusan, atau rekomendasi).
- d. Program memproses data IPK menggunakan metode yang sesuai dengan fitur terpilih.
- e. Hasil analisis ditampilkan kepada pengguna.
- f. Pengguna dapat mengulang proses atau keluar dari program.

#### 1. Sekuens

Sekuens adalah langkah-langkah yang dieksekusi secara berurutan dari atas ke bawah. Dalam program ini, sekuens muncul pada:

- i. Inisialisasi variabel rata-rata dan parameter distribusi

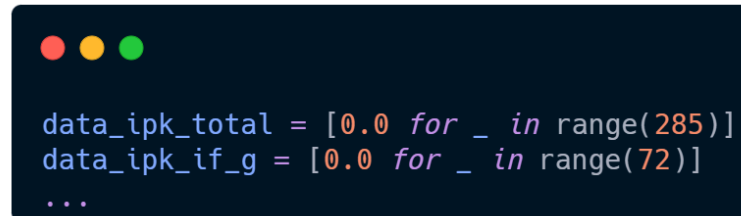


```
mean_if_g = 3.84
mean_if_j = 3.68
mean_sti_g = 3.61
mean_sti_j = 3.71
mean_total = 3.71
alpha = -0.8
omega = 0.2
```

**Gambar 3.1** Inisialisasi Variabel Rata-rata

Nilai-nilai ini adalah parameter statistik IPK (mean, skewness). Program membutuhkan ini untuk menghitung probabilitas (fungsi distribusi kumulatif *skew-normal*). Instruksi dijalankan berurutan tanpa kondisi.

ii. Pembuatan List Data

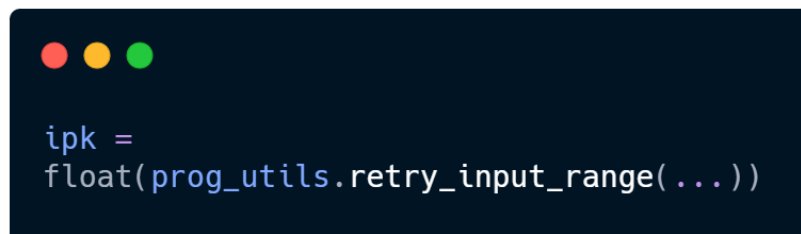


```
data_ipk_total = [0.0 for _ in range(285)]
data_ipk_if_g = [0.0 for _ in range(72)]
...
```

**Gambar 3.2** Data

Program membuat list kosong berisi bilangan *float* sesuai jumlah mahasiswa. Tujuannya adalah menyiapkan tempat untuk menyimpan IPK yang akan dibaca dari CSV.

iii. Input IPK pengguna



```
ipk =
float(prog_utils.retry_input_range(...))
```

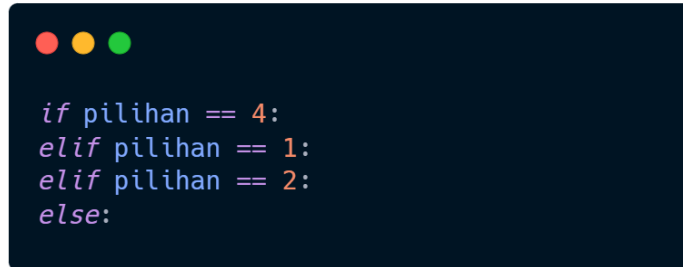
**Gambar 3.3** Input IPK Pengguna

Program meminta pengguna untuk memasukkan IPK. Karena digunakan fungsi `retry_input_range`, input dipastikan sudah valid sebelum diproses oleh program. Ini tetap bagian sekuens karena tidak ada percabangan; hanya memanggil fungsi.

## 2. Sections

Program menggunakan seleksi ketika harus memilih jalur logika berdasarkan kondisi tertentu, sehingga setiap keputusan yang diambil sistem, seperti memvalidasi input pengguna, menentukan menu yang akan dijalankan, atau mengecek ketersediaan stok dapat diproses secara tepat sesuai dengan situasi yang sedang terjadi.

i. If, Elif, dan Else



```
if pilihan == 4:
elif pilihan == 1:
elif pilihan == 2:
else:
```

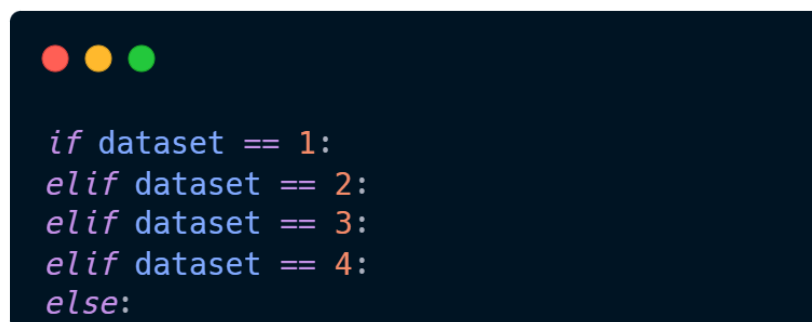
**Gambar 3.4** Mengecek Pilihan

Setelah user memilih menu, program mengecek nilainya.

- Jika 4: keluar
- Jika 1: tampilkan isi dataset
- Jika 2: hitung peluang
- Jika 3: rekomendasi jurusan

Selain itu, program akan menampilkan “input tidak valid” dan meminta masukan lagi.

ii. Seleksi *dataset*

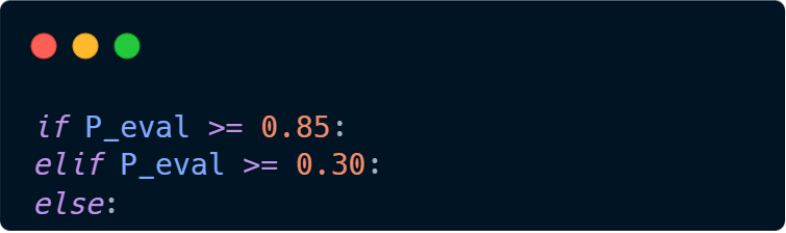


```
if dataset == 1:
elif dataset == 2:
elif dataset == 3:
elif dataset == 4:
else:
```

**Gambar 3.5** Cek Dataset

Program mengecek pilihan *dataset* yang akan digunakan: IF Ganesha, IF Jatinangor, STI Ganesha, STI Jatinangor, dan total gabungan.

iii. Seleksi kategori rekomendasi



```
if P_eval >= 0.85:  
elif P_eval >= 0.30:  
else:
```

**Gambar 3.6** Rekomendasi


Hasil probabilitas IPK pengguna dibandingkan dengan distribusi total, kemudian dikategorikan:

- $\geq 85\%$ : Sangat Baik
- 30–85%: Baik
- $< 30\%$ : Perlu Perhatian

Seleksi ini menentukan kategori akhir yang ditampilkan

### 3. Repetisi

i. Loop utama program

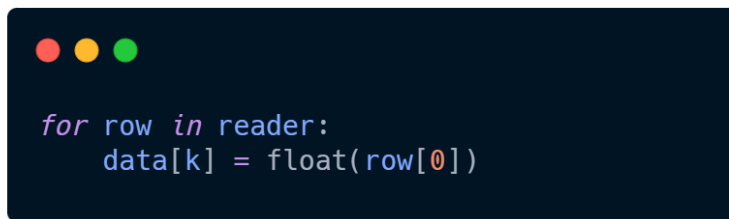


```
while True:
```

**Gambar 3.7** Pengulangan

Program terus berputar menampilkan menu sampai user memilih keluar. Ini adalah *infinite loop* yang hanya berhenti ketika terkena kondisi **break** di pilihan

ii. Loop membaca file CSV

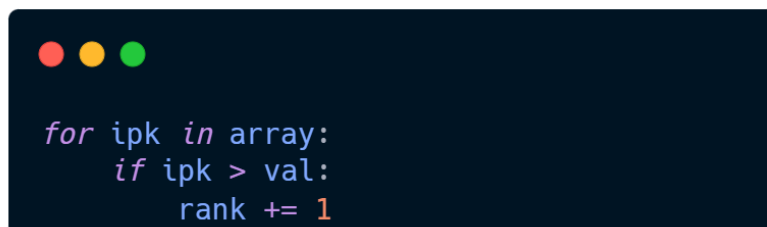


```
for row in reader:
    data[k] = float(row[0])
```

**Gambar 3.8** Loop membaca file

Membaca setiap baris file CSV IPK, lalu menyimpannya ke array. Setiap iterasi mengambil 1 angka IPK.

iii. Loop menghitung peringkat

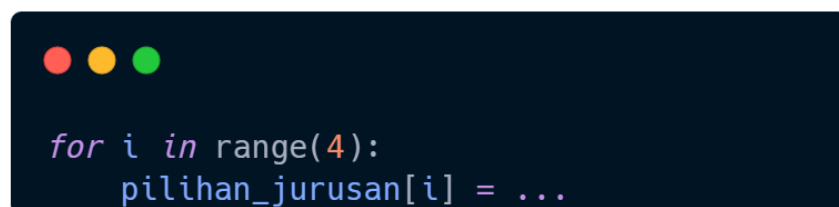


```
for ipk in array:
    if ipk > val:
        rank += 1
```

**Gambar 3.9** Loop menghitung peringkat

Membandingkan IPK user dengan setiap IPK mahasiswa lain. Jika lebih kecil → peringkat bertambah. Ini digunakan untuk menentukan ranking user di jurusan tertentu.

iv. Loop input pilihan jurusan



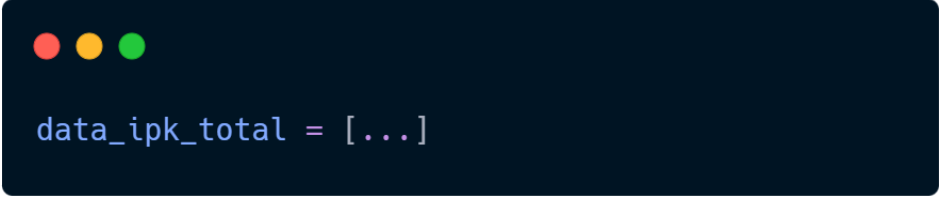
```
for i in range(4):
    pilihan_jurusan[i] = ...
```

**Gambar 3.10** Loop memilih jurusan

Program meminta user memasukkan 4 pilihan jurusan. Loop menjaga agar tidak perlu menulis empat kali input secara manual.

#### 4. Array

- i. Menyimpan *dataset* IPK

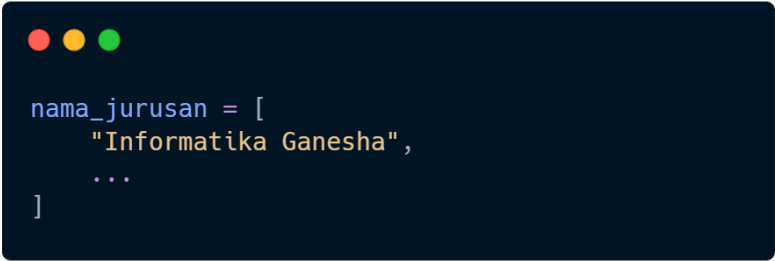


```
data_ipk_total = [...]
```

**Gambar 3.11** Array Data

List ini menyimpan semua IPK dari seluruh mahasiswa (285 orang). Program membutuhkan ini untuk menentukan probabilitas dan peringkat.

- ii. Menyimpan daftar nama jurusan



```
nama_jurusan = [  
    "Informatika Ganesha",  
    ...  
]
```

**Gambar 3.12** Menyimpan nama jurusan

List digunakan agar memudahkan akses nama jurusan berdasarkan indeks. Misal, `nama_jurusan[0] = Informatika Ganesha`

- iii. Menyimpan 4 pilihan jurusan user

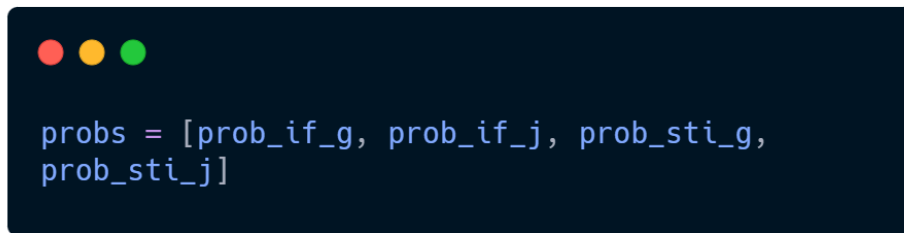


```
pilihan_jurusan = [0 for _ in range(4)]
```

**Gambar 3.13** Menyimpan pilihan jurusan pengguna

List ini menyimpan indeks jurusan yang dipilih user (1 s.d. 4). Nilai ini akan dipakai untuk menghitung probabilitas di tiap pilihan jurusan.

- iv. Menyimpan probabilitas IPK



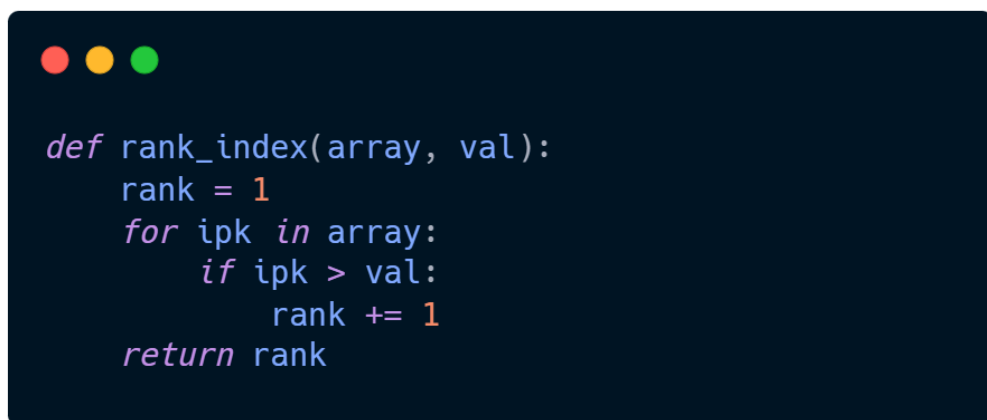
```
probs = [prob_if_g, prob_if_j, prob_sti_g,
         prob_sti_j]
```

**Gambar 3.14** Menyimpan probabilitas IPK

List ini menyimpan peluang penjurusan berdasarkan pilihan sebelumnya. Contohnya, jika pengguna memilih secara berturut-turut IF-G, IF-J, STI-G, dan STI-J, peluang akan disimpan di list “probs” sesuai dengan urutan tersebut (lihat gambar).

## 5. Subprogram

- i. Fungsi rank\_index(array, val)



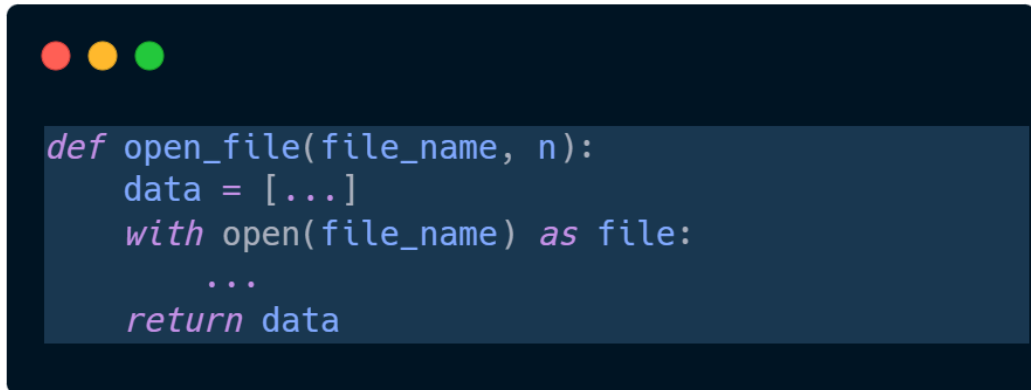
```
def rank_index(array, val):
    rank = 1
    for ipk in array:
        if ipk > val:
            rank += 1
    return rank
```

**Gambar 3.15** Fungsi Rank Index

Menghitung peringkat IPK user dibandingkan seluruh mahasiswa dalam jurusan. Mengembalikan nilai (return), jadi ini adalah fungsi.



ii. Fungsi `open_file(file_name, n)`



```
def open_file(file_name, n):  
    data = []  
    with open(file_name) as file:  
        ...  
    return data
```

**Gambar 3.16** Fungsi Open File

Membaca file CSV, mengisi array IPK, lalu mengembalikannya. Fungsi ini mempermudah pengisian dataset tanpa mengulang kode

iii. Subprogram dari modul lain

- `compute_xi(...)`
- `truncated_skew_norm_cdf(...)`

Dipakai untuk menghitung probabilitas distribusi IPK (CDF). Program tidak menghitung rumus statistiknya langsung, tetapi memanggil subprogram.

iv. Fungsi utama `main()` sebagai prosedur

`main()` tidak mengembalikan nilai, tetapi mengatur seluruh jalannya program (menu, input, perhitungan, output). Secara konsep, ini adalah prosedur.

## PENUTUP

### A. Kesimpulan

Program Prediktor untuk Jurusan merupakan program berbasis teks yang digunakan untuk menganalisis dan memprediksi posisi akademik mahasiswa berdasarkan nilai IPK. Program ini menyediakan fitur peringkat IPK, estimasi peluang memasuki jurusan, serta rekomendasi akademik berbasis distribusi statistik.

Secara logika, program menerapkan percabangan untuk pemilihan fitur, perulangan menu untuk interaksi berkelanjutan, serta *array* untuk penyimpanan data IPK mahasiswa, sehingga program ini telah menerapkan konsep dasar berpikir komputasional terhadap sistem prediktor untuk jurusan secara sederhana namun fungsional menggunakan bahasa pemrograman Python.

### B. Lesson Learned

Program Prediksi Penjurusan, penggunaan perulangan (*looping*) diperlukan agar sistem dapat terus berjalan selama pengguna masih ingin melakukan analisis hingga akhirnya memilih untuk keluar dari program. Selain itu, penggunaan struktur kondisional (*if, elif, else*) sangat penting karena setiap pilihan menu, seperti menampilkan peringkat IPK, menghitung peluang memasuki jurusan, atau memberikan rekomendasi akademik, memiliki proses dan alur kerja yang berbeda. Program ini juga menekankan pentingnya penggunaan struktur data *list* untuk menyimpan data IPK historis mahasiswa agar data dapat diakses dan diolah secara dinamis.

Selain itu, kami juga menerapkan fungsi dan prosedur untuk memisahkan setiap proses utama seperti validasi input, perhitungan peluang, serta pemberian rekomendasi agar program lebih terstruktur dan mudah dikembangkan. Struktur program disusun secara modular dengan pemisahan fungsi dan pengelolaan data pada file terpisah sehingga program menjadi lebih terorganisasi, mudah dipelihara, dan lebih efisien dalam pengembangan.

### **C. Dokumentasi**

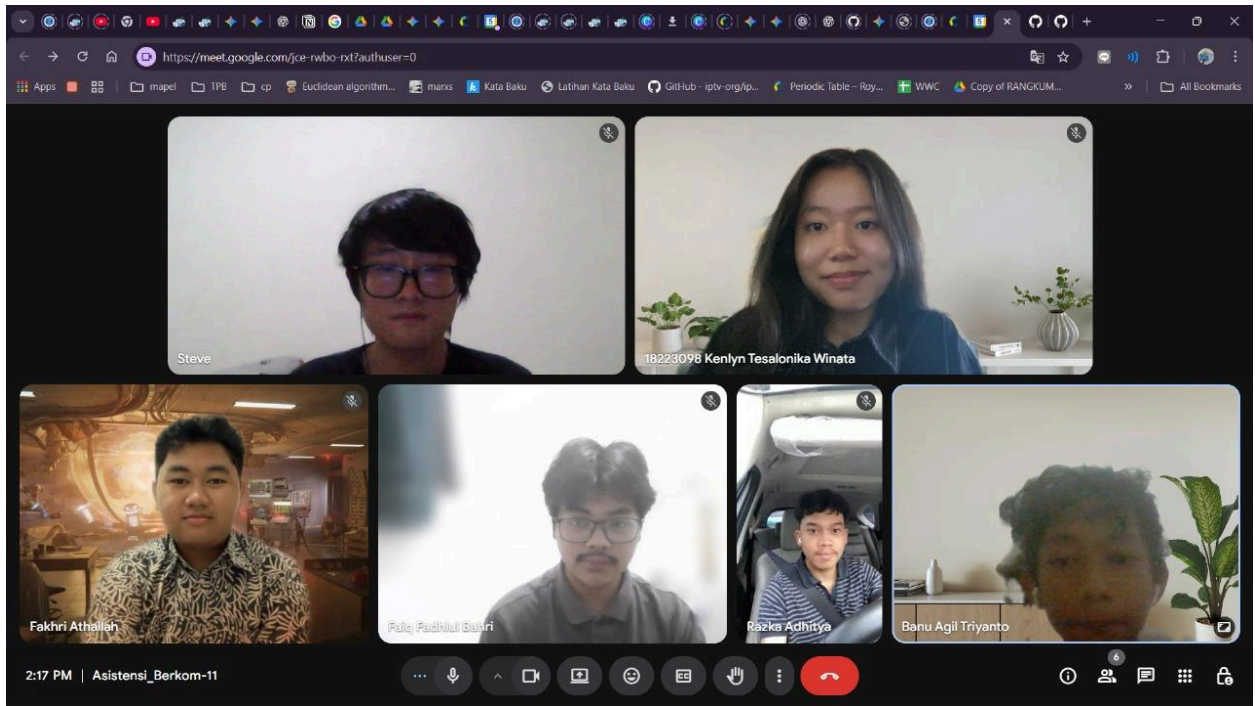
Github:

[https://github.com/EnnEffDec/Project\\_2\\_Berkom](https://github.com/EnnEffDec/Project_2_Berkom)

PPT dan Video

<https://drive.google.com/drive/folders/1PaAD2ldQmu8pR-OKxJSX9nmqsb2WlhmG?usp=sharing>

## LAMPIRAN



**Gambar 4.1** Asistensi Jumat, 5 Desember 2025