

Penjabaran Masalah

1. Elemen Komputasi

Pada kode program kami, ada beberapa elemen komputasi yang digunakan yaitu :

```
1 def showStudentData():
```

```
1 def changeStudentData():
```

a. Operasi matematika, untuk menaikkan indeks dalam perulangan.

```
1 i += 1
```

b. Operasi string, untuk manipulasi string / proses input pengguna

```
1 studentNIM = input("Input ID Students\t\t\t: ")
```

c. Pernyataan kondisi, untuk memproses program berdasarkan kondisi

```
1 if not students:
```

```
1 if searchIDStudent in index["ID STUDENT"]:
```

d. Melakukan Perulangan, untuk mengulangi program yang sama dalam kondisi benar

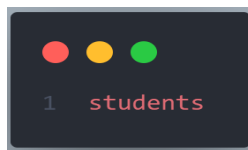
```
1 while True:
```

2. Elemen Dekomposisi

a. Fungsi, untuk membuat program satu kesatuan dan membaginya menjadi sub-sub kecil

```
1 def addStudentsData():
```

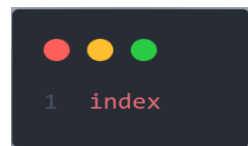
b. Struktur data, untuk menyimpan informasi yang mewakili detail program



```
1 students
```

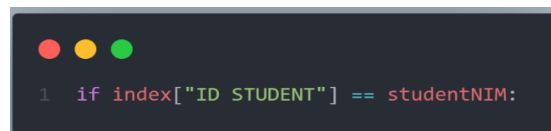
3. Elemen Pattern Recognition

a. Pencocokan String, cocok dengan



```
1 index
```

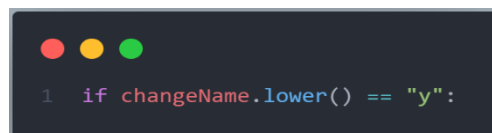
untuk memeriksa apakah inputan pengguna pencarian



```
1 if index["ID STUDENT"] == studentNIM:
```

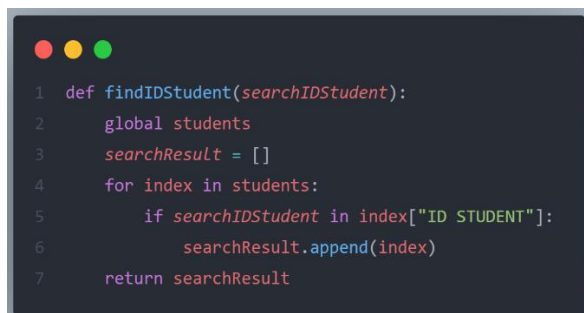
4. Generalisasi

a. Struktur data generik dan desain yang modular, untuk menyimpan data sebagai dan membuatnya lebih mudah beradaptasi dengan program lain tanpa modifikasi/pembuatan program terbaru.



```
1 if changeName.lower() == "y":
```

kamus



```
1 def findIDStudent(searchIDStudent):
2     global students
3     searchResult = []
4     for index in students:
5         if searchIDStudent in index["ID STUDENT"]:
6             searchResult.append(index)
7     return searchResult
```



```
1 def mergeSort(n, process):
2     if len(n) <= 1:
3         return n
4
5     midProcess = len(n) // 2
6     leftProcess = mergeSort(n[:midProcess], process)
7     rightProcess = mergeSort(n[midProcess:], process)
8
9     return merge(leftProcess, rightProcess, process)
```

Penjabaran Algoritma beserta kompleksitas dan data gambar disertai hasil program

1. Fitur Tambah Data Mahasiswa

```

1  def addStudentsData():
2      global students
3      print("="*70)
4      print("\t\t\t Add Student Data")
5      print("="*70)
6      studentNIM = input("Input ID Students\t\t\t: ")
7      studentName = input("Input Name\t\t\t\t: ")
8      studentPlaceOfBirth = input("Input Place of Birth\t\t\t: ")
9      studentDateOfBirth = input("Input Date of Birth (dd/mm/yyyy)\t: ")
10     studentMajor = input("Input Major\t\t\t\t: ")
11     studentEntryYear = input("Input Entry Year\t\t\t: ")
12     students.append({
13         "ID STUDENT": studentNIM,
14         "NAME": studentName,
15         "PLACE of BIRTH": studentPlaceOfBirth,
16         "DATE of BIRTH": studentDateOfBirth,
17         "MAJORITY": studentMajor,
18         "ENTRY YEAR": studentEntryYear
19     })
20     print("\nStudent data added successfully!")

```

Hasil :

```

=====
                        Add Student Data
=====
Input ID Students           : 231113410
Input Name                  : Rasidin Harahap
Input Place of Birth        : Majelangka
Input Date of Birth (dd/mm/yyyy) : 28/05/2003
Input Major                 : Teknik Informatika
Input Entry Year             : 2023

Student data added successfully!

```

Algoritma :

```

1 # D. INISIALISASI FUNGSI ADDSTUDENTDATA UNTUK MENAMBAHKAN DATA
2 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
3 # - CETAK GARIS PEMBATAS DAN JUDUL
4 # - LAKUKAN INPUTAN MULAI DARI NIM, NAMA, TEMPAT DAN TANGGAL LAHIR, JURUSAN DAN TANGGAL MASUK
5 # - TAMBAHKAN SELURUH DATA INPUTAN KE DALAM VARIABEL MAHASISWA
6 # - CETAK PEMBERITAHUAN BAHWA DATA TELAH BERHASIL DITAMBAHKAN

```

Kompleksitas waktu : $O(1)$ → fungsi ini menambahkan data mahasiswa baru ke variabel students dan membutuhkan waktu yang konstan.

Kompleksitas Ruang : $O(1)$ → fungsi ini membuat objek dictionary untuk menyimpan data secara ruangnya konstan.

2. Fitur Tampilkan Data Mahasiswa

```

1 def showStudentData():
2     global students
3     if not students:
4         print("\nThere is no student data")
5         return
6     print("\n=====")
7     print("\t\t\tStudent Data:")
8     print("=====")
9     allDataStudents = []
10    for index in students:
11        allDataStudents.append(
12            [index["ID STUDENT"], index["NAME"], index["MAJORITY"], index["ENTRY YEAR"]])
13    print(tabulate(allDataStudents, headers=[
14        "ID STUDENT", "NAME", "MAJORITY", "ENTRY YEAR"], tablefmt="grid"))
15

```

Hasil :

ID STUDENT	NAME	MAJORITY	ENTRY YEAR
231112518	Tasya Syafriza	Teknik Informatika	2021
231111619	Delarosa Butar Butar	Sistem Informasi	2021
231111908	Purba Tua Siregar	Akutansi	2023
231113410	Rasidin Harahap	Teknik Informatika	2023
231111654	Tasya Farsya	Manajemen Informatika	2020
231115418	Enjelina Sitorus	Teknik Informatika	2021
231111398	Farhan Batu Bara	Sistem Informasi	2022
231113498	Ahmad Yaya	Manajemen Informatika	2023
231111459	Talitha Anggraini	Teknik Informatika	2023
231111678	Rara Salsabila Putri	Sistem Informasi	2020

Algoritma :

```

1 # E. INISIALISASI FUNGSI SHOWSTUDENTDATA
2 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
3 # - JIKA MAHASISWA TIDAK ADA, CETAK TIDAK ADA DATA MAHASISWA
4 # - CETAK GARIS PEMBATAS DATA MAHASISWA DAN JUDULNYA
5 # - INISIALISASI TEMPAT UNTUK MENAMPUNG KESELURUHAN DATA DIDALAM LIST
6 # - LAKUKAN PERULANGAN INDEX DENGAN DATA YANG ADA
7 # - TAMBAHKAN DATANYA KE MASING-MASING FORMAT
8 # - CETAK DATA MAHASISWA DENGAN TABEL DARI LIBRARY TABULATE

```

Kompleksitas waktu : $O(n)$ → fungsi ini harus mengiterasi semua n data mahasiswa

Kompleksitas ruang : $O(n)$ → fungsi ini membuat daftar sementara `allDataStudents` untuk menyimpan data mahasiswa yang telah diformat

3. Fitur Ubah Data Mahasiswa

```

1 def changeStudentData():
2     global students
3     studentNIM = input("The student's ID will be changed: ")
4     for index in students:
5         if index["ID STUDENT"] == studentNIM:
6             print("\nFollowing are the student data : ")
7             allDataStudents = [
8                 {"FIELD": "ID STUDENT", "DETAIL": index["ID STUDENT"]},
9                 {"FIELD": "NAME", "DETAIL": index["NAME"]},
10                {"FIELD": "PLACE of BIRTH", "DETAIL": index["PLACE of BIRTH"]},
11                {"FIELD": "DATE of BIRTH", "DETAIL": index["DATE of BIRTH"]},
12                {"FIELD": "MAJORITY", "DETAIL": index["MAJORITY"]},
13                {"FIELD": "ENTRY YEAR", "DETAIL": index["ENTRY YEAR"]}
14            ]
15            print(tabulate(allDataStudents, headers="keys", tablefmt="grid"))
16

```

```

1 while True:
2     print(
3         "\n=====")
4     print("\t\t\t Latest Student Data")
5     print(
6         "=====")
7     changeName = input("Change Name (y/t)\t\t\t: ")
8     dataQuestion = 0
9     if changeName.lower() == "y":
10         index["NAME"] = input("Enter a new Name\t\t\t: ")
11         dataQuestion = True
12     elif changeName.lower() == "t":
13         pass
14     else:
15         print("Invalid choice! Please enter 'y' or 't'.")
16         changeName = input("Change Name (y/t)\t\t\t: ")
17         if changeName.lower() == "y":
18             index["NAME"] = input("Enter a new Name\t\t\t: ")
19             dataQuestion = True
20         elif changeName.lower() == "t":
21             pass

```



```
1  changePlaceOfBirth = input("Change Place of Birth (y/t)\t\t: ")
2  if changePlaceOfBirth.lower() == "y":
3      index["PLACE of BIRTH"] = input(
4          "Enter the new Place of Birth\t\t: ")
5      dataQuestion = True
6  elif changePlaceOfBirth.lower() == "t":
7      pass
8  else:
9      print("Invalid choice! Please enter 'y' or 't'.")
10     changePlaceOfBirth = input(
11         "Change Place of Birth (y/t)\t\t: ")
12     if changePlaceOfBirth.lower() == "y":
13         index["PLACE of BIRTH"] = input(
14             "Enter the new Place of Birth\t\t: ")
15         dataQuestion = True
16     elif changePlaceOfBirth.lower() == "t":
17         pass
```



```
1  changeDateOfBirth = input("Change Date of Birth (y/t)\t\t: ")
2  if changeDateOfBirth.lower() == "y":
3      index["DATE of BIRTH"] = input(
4          "Enter new Date of Birth\t\t\t: ")
5      dataQuestion = True
6  elif changeDateOfBirth.lower() == "t":
7      pass
8  else:
9      print("Invalid choice! Please enter 'y' or 't'.")
10     changeDateOfBirth = input(
11         "Change Date of Birth (y/t)\t\t: ")
12     if changeDateOfBirth.lower() == "y":
13         index["DATE of BIRTH"] = input(
14             "Enter new Date of Birth\t\t\t: ")
15         dataQuestion = True
16     elif changeDateOfBirth.lower() == "t":
17         pass
18
```




```
1  changeMajor = input("Change Majority (y/t)\t\t\t: ")
2  if changeMajor.lower() == "y":
3      index["MAJORITY"] = input(
4          "Enter new Majority\t\t\t: ")
5      dataQuestion = True
6  elif changeMajor.lower() == "t":
7      pass
8  else:
9      print("Invalid choice! Please enter 'y' or 't'.")
10     changeMajor = input(
11         "Change Majority (y/t)\t\t\t: ")
12     if changeMajor.lower() == "y":
13         index["MAJORITY"] = input(
14             "Enter new Majority\t\t\t: ")
15         dataQuestion = True
16     elif changeMajor.lower() == "t":
17         pass
```



```
1  changeYearIn = input("Change Entry Year (y/t)\t\t\t: ")
2  if changeYearIn.lower() == "y":
3      index["ENTRY YEAR"] = input(
4          "Enter the new Entry Year\t\t: ")
5      dataQuestion = True
6      break
7  elif changeYearIn.lower() == "t":
8      break
9  else:
10     print("Invalid choice! Please enter 'y' or 't'.")
11     changeYearIn = input("Change Entry Year (y/t)\t\t\t: ")
12     if changeYearIn.lower() == "y":
13         index["ENTRY YEAR"] = input(
14             "Enter the new Entry Year\t\t: ")
15         dataQuestion = True
16         break
17     elif changeYearIn.lower() == "t":
18         break
```



```

1     if dataQuestion:
2         print(f"Data with ID {
3             studentNIM} has been successfully changed .")
4     else:
5         print("No Data Changed")
6     return
7 if ["ID STUDENT"] != studentNIM:
8     print(f"Sorry, ID {studentNIM} was not found.")

```

Hasil :

```

The student's ID will be changed: 231111654

Following are the student data :
+-----+-----+
| FIELD          | DETAIL          |
+=====+=====+
| ID STUDENT     | 231111654       |
+-----+-----+
| NAME           | Tasya Farsya    |
+-----+-----+
| PLACE of BIRTH | Papua Nugini     |
+-----+-----+
| DATE of BIRTH  | 26/03/2001      |
+-----+-----+
| MAJORITY       | Manajemen Informatika |
+-----+-----+
| ENTRY YEAR     | 2020             |
+-----+-----+

```

```

=====
                          Latest Student Data
=====
Change Name (y/t)          : y
Enter a new Name           : Tasya Farasya Amelia
Change Place of Birth (y/t) : t
Change Date of Birth (y/t) : t
Change Majority (y/t)      : t
Change Entry Year (y/t)    : t
Data with ID 231111654 has been successfully changed .

```

Algoritma :

```
1 # F. INISIALISASI FUNGSI CHANGESTUDENTDATA UNTUK MENGUBAH DATA
2 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
3 # - INPUT NIM MAHASISWA
4 # - LAKUKAN PERULANGAN INDEX DENGAN DATA YANG ADA
5 # - JIKA NIM YANG DIINPUT == NIM YANG ADA DI DATA, MAKA TAMBAHKAN DATA SESUAI FORMAT
6 # - CETAK DATA DENGAN TABEL MENGGUNAKAN LIBRARY TABULATE
7 # - LAKUKAN PERULANGAN WHILE SELAMA KONDISINYA BENAR
8 # - CETAK GARIS PEMBATAS DAN JUDUL
9 # - INISIALISASI VARIABEL DATAQUESTION = 0
10 # - LAKUKAN PERTANYAAN MENGENAI DATA YANG INGIN DI UBAH DENGAN PERKONDISIAN
11 # - JIKA PENGGUNA MENGINPUTNYA DATA TERBARU, MAKA SETTING DATAQUESTION MENJADI TRUE UNTUK MENYIMPAN DATANYA
12 # - JIKA SUDAH SELESAI MAKA CETAK PEMBERITAHUAN BAHWA DATA TELAH DI UBAH
13 # - APABILA PENGGUNA TIDAK MENGINPUT DATA TERBARU, CETAK PEMBERITAHUANNYA
14 # - JIKA NIM YANG DI INPUT TIDAK SAMA DENGAN NIM YANG ADA DI DATA, MAKA CETAK PEMBERITAHUAN MAAF, NIM TERSEBUT TIDAK DITEMUKAN DI DATA MAHASISWA
```

Kompleksitas waktu : $O(n)$ → fungsi ini harus mengiterasi semua n data mahasiswa untuk menemukan data dengan NIM yang diinput.

Kompleksitas ruang : $O(1)$ → fungsi ini tidak mempengaruhi struktur data sebelumnya jika tidak diubah

4. Fitur Urutkan Data Mahasiswa

```
1 def sortStudentData():
2     global students
3
4     while True:
5         print("\n=====")
6         print("\t\t\t Sort Student Data")
7         print("=====")
8         print("1. ID Students")
9         print("2. Name")
10        print("3. Majority")
11        print("4. Entry Year")
12        print("5. Close")
13        print("=====")
14
15        to = int(input("Sort type (Number Input): "))
16
17        if to == 1:
18            process = "ID STUDENT"
19            break
20        elif to == 2:
21            process = "NAME"
22            break
23        elif to == 3:
24            process = "MAJORITY"
25            break
26        elif to == 4:
27            process = "ENTRY YEAR"
28            break
29        elif to == 5:
30            return
31        else:
32            print("INVALID!")
```



```
1  def mergeSort(n, process):
2      if len(n) <= 1:
3          return n
4
5      midProcess = len(n) // 2
6      leftProcess = mergeSort(n[:midProcess], process)
7      rightProcess = mergeSort(n[midProcess:], process)
8
9      return merge(leftProcess, rightProcess, process)
10
11 def merge(leftProcess, rightProcess, bagian):
12     students = []
13     i = 0
14     j = 0
15     while i < len(leftProcess) and j < len(rightProcess):
16         if leftProcess[i][bagian] <= rightProcess[j][bagian]:
17             students.append(leftProcess[i])
18             i += 1
19         else:
20             students.append(rightProcess[j])
21             j += 1
22
23     students += leftProcess[i:]
24     students += rightProcess[j:]
25
26     return students
27
28 students = mergeSort(students, process)
29 print("Succeed! The data has been sorted accordingly ", process)
30 showStudentData()
31 sortStudentData()
```

Hasil :

```
=====
                        Sort Student Data
=====
1. ID Students
2. Name
3. Majority
4. Entry Year
5. Close
=====
Sort type (Number Input): █
```

Contoh urutan berdasarkan nama :

ID STUDENT	NAME	MAJORITY	ENTRY YEAR
231113498	Ahmad Yaya	Manajemen Informatika	2023
231111619	Delarosa Butar Butar	Sistem Informasi	2021
231115418	Enjelina Sitorus	Teknik Informatika	2021
231111398	Farhan Batu Bara	Sistem Informasi	2022
231111908	Purba Tua Siregar	Akutansi	2023
231111678	Rara Salsabila Putri	Manajemen Informatika	2020
231113410	Rasidin Harahap	Teknik Informatika	2023
231111459	Talitha Anggraini	Teknik Informatika	2023
231111654	Tasya Farasya Amelia	Manajemen Informatika	2020
231112518	Tasya Syafriza	Teknik Informatika	2021

Algoritma :

```
1 # G. INISIALISASI FUNGSI SORTSTUDENTDATA UNTUK MENGURUTKAN DATA
2 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
3 # - LAKUKAN PERULANGAN WHILE SELAMA KONDISI BENAR
4 # - CETAK GARIS PEMBATAS DAN JUDUL
5 # - CETAK MENU PILIHAN PENGURUTAN UNTUK PENGGUNA
6 # - LAKUKAN INPUT NILAI ATAS PILIHAN PENGGUNA
7 # - JIKA INPUTAN == 1, MAKA DATA AKAN DIURUTKAN BERDASARKAN NIM
8 # - JIKA INPUTAN == 2, MAKA DATA AKAN DIURUTKAN BERDASARKAN NAMA
9 # - JIKA INPUTAN == 3, MAKA DATA AKAN DIURUTKAN BERDASARKAN JURUSAN
10 # - JIKA INPUTAN == 4, MAKA DATA AKAN DIURUTKAN BERDASARKAN TAHUN MASUK
11 # - JIKA INPUTAN == 5, MAKA AKAN KELUAR KE MENU UTAMA
12 # - JIKA TIDAK SEMUANYA, MAKA CETAK INVALID!
13 # - INISIALISASI FUNGSI MERGESORT DENGAN PARAMETER N DAN PROSES
14 # - JIKA PANJANG DATA N KURANG DARI 1, RETURN/KEMBALIKAN NILAI N
15 # - PROSES PERTAMA IALAH MEMBAGI DUA DATA LEWAT VARIABEL MID PROSES
16 # - PROSES KEDUA IALAH MENGURUTKAN DATA DARI SISI KIRI
17 # - PROSES KETIGA IALAH MENGURUTKAN DATA DARI SISI KANAN
18 # - RETURN/KEMBALIKAN NILAI MERGE KIRI, KANAN, DAN PROSESNYA
19 # - INISIALISASI FUNGSI MERGE DENGAN PARAMETER LEFTPROSES, RIGHTPROSES, BAGIAN
20 # - INISIALISASI VARIABEL MAHASISWA DENGAN LIST KOSONG
21 # - INISIALISASI NILAI I DAN J = 0
22 # - LAKUKAN PERULANGAN I JIKA LEBIH KECIL DARI PANJANG PROSES BAGIAN KIRI DAN J LEBIH KECIL DARI PROSES BAGIAN KANAN
23 # - JIKA INDEKS DARI PROSES KIRI LEBIH KECIL DARI YANG KANAN, TAMBAHKAN NILAINYA KE VARIABEL MAHASISWA. LANJUT KE NODE BERIKUTNYA
24 # - JIKA TIDAK, MAKA TAMBAHKAN NILAI DARI PROSES KANAN KE VARIABEL MAHASISWA. LANJUT KE NODE BERIKUTNYA
25 # - KESELURUHAN DATA INDEKS BAGIAN KIRI DAN KANAN DIMASUKKAN KE DALAM LIST MAHASISWA. RETURN/KEMBALIKAN NILAI
26 # - PANGGIL KEMBALI FUNGSI MERGE SORT DENGAN ARGUMEN MAHASISWA DAN PROSES
27 # - CETAK PEMBERITAHUAN BAHWA DATA TELAH DIURUTKAN
28 # - TAMPILKAN DATANYA DAN ULANGI FUNGSI SORTSTUDENTDATA
```

Kompleksitas waktu : $O(n \log n)$ → fungsi mergeSort menggunakan algoritma merge sort untuk mengurutkan n elemen

Kompleksitas ruang : $O(n)$ → Dengan menggunakan algoritma merge sort, maka akan ada ruang tambahan untuk array sementara selama proses pengurutan yang selaras dengan jumlah data mahasiswa

5. Fitur Cari Data Mahasiswa

```
1 def findIDStudent(searchIDStudent):
2     global students
3     searchResult = []
4     for index in students:
5         if searchIDStudent in index["ID STUDENT"]:
6             searchResult.append(index)
7     return searchResult
8
9
10 def findNameStudent(searchNameStudent):
11     global students
12     searchResult = []
13     for index in students:
14         if searchNameStudent.lower() in index["NAME"].lower():
15             searchResult.append(index)
16     return searchResult
```

```
1 def searchForStudentData():
2     while True:
3         print("Select a section to search:")
4         print("1. Student ID")
5         print("2. Student Name")
6         to = int(input("Search Type(Enter number): "))
7         if to == 1:
8             searchIDStudent = input(
9                 "Enter the keywords you are looking for : ")
10            searchResult = findIDStudent(searchIDStudent)
11            print("Search complete.")
12            break
13        elif to == 2:
14            searchNameStudent = input(
15                "Enter the keywords you are looking for : ")
16            searchResult = findNameStudent(searchNameStudent)
17            break
18        else:
19            print("INVALID!")
20
21    if searchResult:
22        print("\nSucceed! Here's the data:")
23        allDataStudents = []
24        for index in searchResult:
25            allDataStudents.append([index["ID STUDENT"], index["NAME"], index["PLACE of BIRTH"],
26                                    index["DATE of BIRTH"], index["MAJORITY"], index["ENTRY YEAR"]])
27        print(tabulate(allDataStudents, headers=[
28            "NO", "ID STUDENT", "NAME", "PLACE of BIRTH", "DATE of BIRTH", "MAJORITY", "ENTRY YEAR"], tablefmt="grid"))
29    else:
30        print("Your search results were not found!")
31
```


Hasil :

```
Select a section to search:
1. Student ID
2. Student Name
Search Type(Enter number): 1
Enter the keywords you are looking for : 231111
Search complete.

Succeed! Here's the data:
+-----+-----+-----+-----+-----+-----+
| NO | ID STUDENT | NAME | PLACE of BIRTH | DATE of BIRTH | MAJORITY |
+-----+-----+-----+-----+-----+-----+
| 231111678 | Rara Salsabila Putri | Sulawesi Tengah | 18/04/2001 | Manajemen Informatika | 2020 |
+-----+-----+-----+-----+-----+-----+
| 231111654 | Tasya Farasya Amelia | Papua Nugini | 26/03/2001 | Manajemen Informatika | 2020 |
+-----+-----+-----+-----+-----+-----+
| 231111619 | Delarosa Butar Butar | Padang | 12/01/2003 | Sistem Informasi | 2021 |
+-----+-----+-----+-----+-----+-----+
| 231111398 | Farhan Batu Bara | Kalimantan | 24/04/2004 | Sistem Informasi | 2022 |
+-----+-----+-----+-----+-----+-----+
| 231111908 | Purba Tua Siregar | Pekanbaru | 12/09/2004 | Akutansi | 2023 |
+-----+-----+-----+-----+-----+-----+
| 231111459 | Talitha Anggraini | Lampung | 29/01/2003 | Teknik Informatika | 2023 |
+-----+-----+-----+-----+-----+-----+
```

```
Select a section to search:
1. Student ID
2. Student Name
Search Type(Enter number): 2
Enter the keywords you are looking for : tasya

Succeed! Here's the data:
+-----+-----+-----+-----+-----+-----+
| NO | ID STUDENT | NAME | PLACE of BIRTH | DATE of BIRTH | MAJORITY |
+-----+-----+-----+-----+-----+-----+
| 231111654 | Tasya Farasya Amelia | Papua Nugini | 26/03/2001 | Manajemen Informatika | 2020 |
+-----+-----+-----+-----+-----+-----+
| 231112518 | Tasya Syafriza | Medan | 23/01/2003 | Teknik Informatika | 2021 |
+-----+-----+-----+-----+-----+-----+
```

```
Select a section to search:
1. Student ID
2. Student Name
Search Type(Enter number): 2
Enter the keywords you are looking for : ra

Succeed! Here's the data:
+-----+-----+-----+-----+-----+-----+
| NO | ID STUDENT | NAME | PLACE of BIRTH | DATE of BIRTH | MAJORITY |
+-----+-----+-----+-----+-----+-----+
| 231111678 | Rara Salsabila Putri | Sulawesi Tengah | 18/04/2001 | Manajemen Informatika | 2020 |
+-----+-----+-----+-----+-----+-----+
| 231111654 | Tasya Farasya Amelia | Papua Nugini | 26/03/2001 | Manajemen Informatika | 2020 |
+-----+-----+-----+-----+-----+-----+
| 231111398 | Farhan Batu Bara | Kalimantan | 24/04/2004 | Sistem Informasi | 2022 |
+-----+-----+-----+-----+-----+-----+
| 231113410 | Rasidin Harahap | Majelangka | 28/05/2003 | Teknik Informatika | 2023 |
+-----+-----+-----+-----+-----+-----+
| 231111459 | Talitha Anggraini | Lampung | 29/01/2003 | Teknik Informatika | 2023 |
+-----+-----+-----+-----+-----+-----+
```


Algoritma :

```
1 # H. INISIALISASI FUNGSI FINDIDSTUDENT DENGAN PARAMETER SEARCHIDSTUDENT
2 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
3 # - INISIALISASI LIST SEARCHRESULT KOSONG
4 # - LAKUKAN PERULANGAN INDEX DENGAN DATA
5 # - JIKA SEARCHIDSTUDENT ADA DI DATA NIM, TAMBAHKAN NILAINYA KE SEARCHRESULT. RETURN/KEMBALIKAN NILAI
6 # I. INISIALISASI FUNGSI FINDNAMESTUDENT DENGAN PARAMETER SEARCHNAMESTUDENT
7 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
8 # - INISIALISASI LIST SEARCHRESULT KOSONG
9 # - LAKUKAN PERULANGAN INDEX DENGAN DATA
10 # - JIKA SEARCHNAMESTUDENT ADA DI DATA NAMA, TAMBAHKAN NILAINYA KE SEARCHRESULT. RETURN/KEMBALIKAN NILAI
11 # J. INISIALISASI FUNGSI SEARCHFORSTUDENTDATA
12 # - LAKUKAN PERULANGAN WHILE SELAMA KONDISI BENAR
13 # - CETAK PILIHAN UNTUK PENGGUNA
14 # - LAKUKAN INPUTAN UNTUK MEMILIH TIPE PENCARIAN
15 # - JIKA PILIHANNYA == 1, MAKA PENGGUNA HARUS MENGINPUTKAN KATA KUNCI YANG INGIN DICARI. KEMUDIAN PANGGIL FUNGSI FINDIDSTUDENT DAN CETAK DATA DITEMUKAN. KELUAR DARI PERULANGAN
16 # - JIKA PILIHANNYA == 2, MAKA PENGGUNA HARUS MENGINPUTKAN KATA KUNCI YANG INGIN DICARI. JALANKAN FUNGSI FINDNAMESTUDENT DAN KELUAR DARI PERULANGAN.
17 # - JIKA TIDAK KEDUANYA, CETAK PEMBERITAHUAN INVALID!
18 # - LAKUKAN PENGKONDISIAN UNTUK SEARCHRESULT
19 # - CETAK PEMBERITAHUAN BERIKUT DATA YANG DICARI
20 # - INISIALISASI VARIABEL ALLDATASTUDENTS DENGAN LIST KOSONG
21 # - LAKUKAN PERULANGAN INDEX DENGAN DATA YANG ADA DI SEARCHRESULT
22 # - TAMBAHKAN DATANYA KE VARIABEL ALLDATASTUDENTS SESUAI FORMAT KEMUDIAN CETAK DENGAN TABEL
23 # - JIKA DATANYA TIDAK ADA, MAKA CETAK PEMBERITAHUAN BAHWA DATA YANG DICARI TIDAK DITEMUKAN
```

Kompleksitas waktu fungsi findIDStudent & findNameStudent : $O(n)$ → iterasi kecocokan pada semua n data mahasiswa berdasarkan NIM atau nama

Kompleksitas ruang fungsi findIDStudent & findNameStudent : $O(n)$ → membuat daftar sementara untuk menyimpan data mahasiswa yang cocok dan selaras dengan jumlah data

Kompleksitas waktu fungsi searchForStudentData : $O(n)$ → memanggil fungsi findIDStudent & findNameStudent yang memiliki kompleksitas lanjar alias linear

Kompleksitas ruang fungsi searchForStudentData : $O(n)$ → mewarisi dari fungsi find/pencarian yang telah mendasarinya

6. Fitur Menu Utama / Tampilan Program Data Mahasiswa

```
1 def main():
2     global students
3     while True:
4         print("\n=====")
5         print("\t\t\tStudent Data Application")
6         print("=====")
7         print("1. Add Student Data")
8         print("2. Show Student Data")
9         print("3. Change Student Data")
10        print("4. Sort Student Data")
11        print("5. Search for Student Data")
12        print("6. Close")
13        print("=====")
14        to = int(input("Enter to (1 - 6): "))
15        if to == 1:
16            addStudentsData()
17        elif to == 2:
18            showStudentData()
19        elif to == 3:
20            changeStudentData()
21        elif to == 4:
22            sortStudentData()
23        elif to == 5:
24            searchForStudentData()
25        elif to == 6:
26            print(
27                "Thank you for using our group program")
28            break
29        else:
30            print("INVALID!")
31
```

Hasil :

```
=====
                        Student Data Application
=====
1. Add Student Data
2. Show Student Data
3. Change Student Data
4. Sort Student Data
5. Search for Student Data
6. Close
=====
Enter to (1 - 6): █
```

Algoritma :

```
1 # K. INISIALISASI FUNGSI MAIN
2 # - PANGGIL VARIABEL MAHASISWA KE DALAM FUNGSI INI
3 # - LAKUKAN PERULANGAN SELAMA KONDISI BENAR
4 # - CETAK GARIS PEMBATAS DAN JUDUL
5 # - CETAK MENU PILIHAN PENGGUNA
6 # - LAKUKAN INPUTAN ATAS PILIHAN YANG ADA DI MENU
7 # - JIKA PILIHAN == 1, MAKA JALANKAN FUNGSI ADDSTUDENTDATA
8 # - JIKA PILIHAN == 2, MAKA JALANKAN FUNGSI SHOWSTUDENTDATA
9 # - JIKA PILIHAN == 3, MAKA JALANKAN FUNGSI CHANGESTUDENTDATA
10 # - JIKA PILIHAN == 4, MAKA JALANKAN FUNGSI SORTSTUDENTDATA
11 # - JIKA PILIHAN == 5, MAKA JALANKAN FUNGSI SEARCHFORSTUDENTDATA
12 # - JIKA PILIHAN == 6, MAKA CETAK TERIMA KASIH SUDAH MENGGUNAKAN PROGRAM KELOMPOK KAMI
13 # - JIKA PILIHAN TIDAK SEMUANYA, MAKA CETAK PEMBERITAHUAN INVALID!
```

Berdasarkan data kode program keseluruhan, berikut total keseluruhan kompleksitas :

```
1 # KOMPLEKSITAS WAKTU :
2 # - ADDSTUDENTDATA -->  $O(1)$  // KONSTAN
3 # - SHOWSTUDENTDATA -->  $O(n)$  // LANJAR
4 # - CHANGESTUDENTDATA -->  $O(n)$  // LANJAR
5 # - SORTSTUDENTDATA -->  $O(n \log n)$ 
6 # - FINDIDSTUDENT -->  $O(n)$  // LANJAR
7 # - FINDNAMESTUDENT -->  $O(n)$  // LANJAR
8 # - SEARCHFORSTUDENTDATA -->  $O(n)$  // LANJAR
9 # TOTAL KESELURUHAN =  $O(n \log n)$ 
10
11 # KOMPLEKSITAS RUANG :
12 # - ADDSTUDENTDATA -->  $O(1)$  // KONSTAN
13 # - SHOWSTUDENTDATA -->  $O(n)$  // LANJAR
14 # - CHANGESTUDENTDATA -->  $O(1)$  // KONSTAN
15 # - SORTSTUDENTDATA -->  $O(n)$  // LANJAR
16 # - FINDIDSTUDENT -->  $O(n)$  // LANJAR
17 # - FINDNAMESTUDENT -->  $O(n)$  // LANJAR
18 # - SEARCHFORSTUDENTDATA -->  $O(n)$  // LANJAR-
19 # TOTAL KESELURUHAN =  $O(n)$ 
```