| **PROGRAM PENDATAAN MATA KULIAH MENGGUNAKAN DOUBLE LINKED LIST**      LAPORAN PROYEK AKHIR  MATA KULIAH COMP6362004 – DATA STRUCTURES  KELAS BF20                          Oleh :  2602181875 - Putri Febiyani  2602099812 - Stephen Wijaya Utama  2602155574 - Nazhira Dewi Aqmarina        Semester [Genap] 2022-2023    MALANG          **LEMBAR PERSETUJUAN PROYEK AKHIR**  **<PROGRAM PENDATAAN MATA KULIAH MENGGUNAKAN DOUBLE LINKED LIST>**  MATA KULIAH COMP6362004 – DATA STRUCTURES  KELAS BF20  Semester [Genap] 2022-2023  Laporan akhir proyek ini adalah benar karya kami :    **Putri Febiyani** Stephen Wijaya UtamaNazhira Dewi Aqmarina  260218187526020998122602155574  **Malang, 16 Juni 2023**  **NYOMAN WIRA PRASETYA, S.Kom., M.T., M.Sc.**  D6494 |
| --- |

**BAB I**

**BACKGROUND**

**Latar Belakang**

Program di atas dirancang untuk mengelola data kursus. Program ini menggunakan konsep daftar terkait untuk menyimpan dan mengelola data mata kuliah. Program ini mencakup fungsi-fungsi seperti menambah data mata kuliah mundur dan maju, menghapus data mata kuliah mundur dan maju, mencari dan menampilkan data mata kuliah berdasarkan kode, menampilkan jumlah data mata kuliah, menampilkan semua data mata kuliah, mencari nilai maksimum dan minimum, menampilkan daftar mata kuliah yang gagal, menghitung nilai rata-rata mata kuliah. Program ini juga memungkinkan penyimpanan data dalam file "data.txt".m

**Overview**

Program pengelola data mata kuliah ini menggunakan konsep linked list untuk menyimpan dan mengatur data mata kuliah. Dengan program ini, pengguna dapat melakukan berbagai operasi seperti menambahkan dan menghapus data mata kuliah, mencari data berdasarkan kode, menampilkan jumlah dan semua data mata kuliah, mencari nilai terbesar dan terkecil, menampilkan daftar mata kuliah yang tidak lulus, serta menghitung rata-rata nilai mata kuliah. Program ini juga memiliki kemampuan untuk menyimpan data ke dalam file "data.txt". Dengan fitur-fitur yang komprehensif ini, program ini dapat membantu pengguna dalam mengelola dan menganalisis data mata kuliah dengan mudah dan efisien.

**BAB II  
Literature Review**

**1. Theory**

Program di atas adalah sebuah aplikasi untuk mengelola data mata kuliah. Program ini menggunakan struktur data linked list untuk menyimpan dan memanipulasi data mata kuliah. Aplikasi ini memiliki berbagai macam fungsi, seperti menambah data mata kuliah sebelumnya dan sesudahnya, menghapus data mata kuliah sebelumnya dan sesudahnya, mencari dan menampilkan data mata kuliah berdasarkan kode, menampilkan volume data mata kuliah, menampilkan semua data mata kuliah, mencari nilai maksimum dan minimum, menampilkan daftar mata kuliah yang tidak lulus, menghitung nilai rata-rata dan masih banyak lagi. Aplikasi ini juga dapat menyimpan data ke dalam file teks.

**2. General**

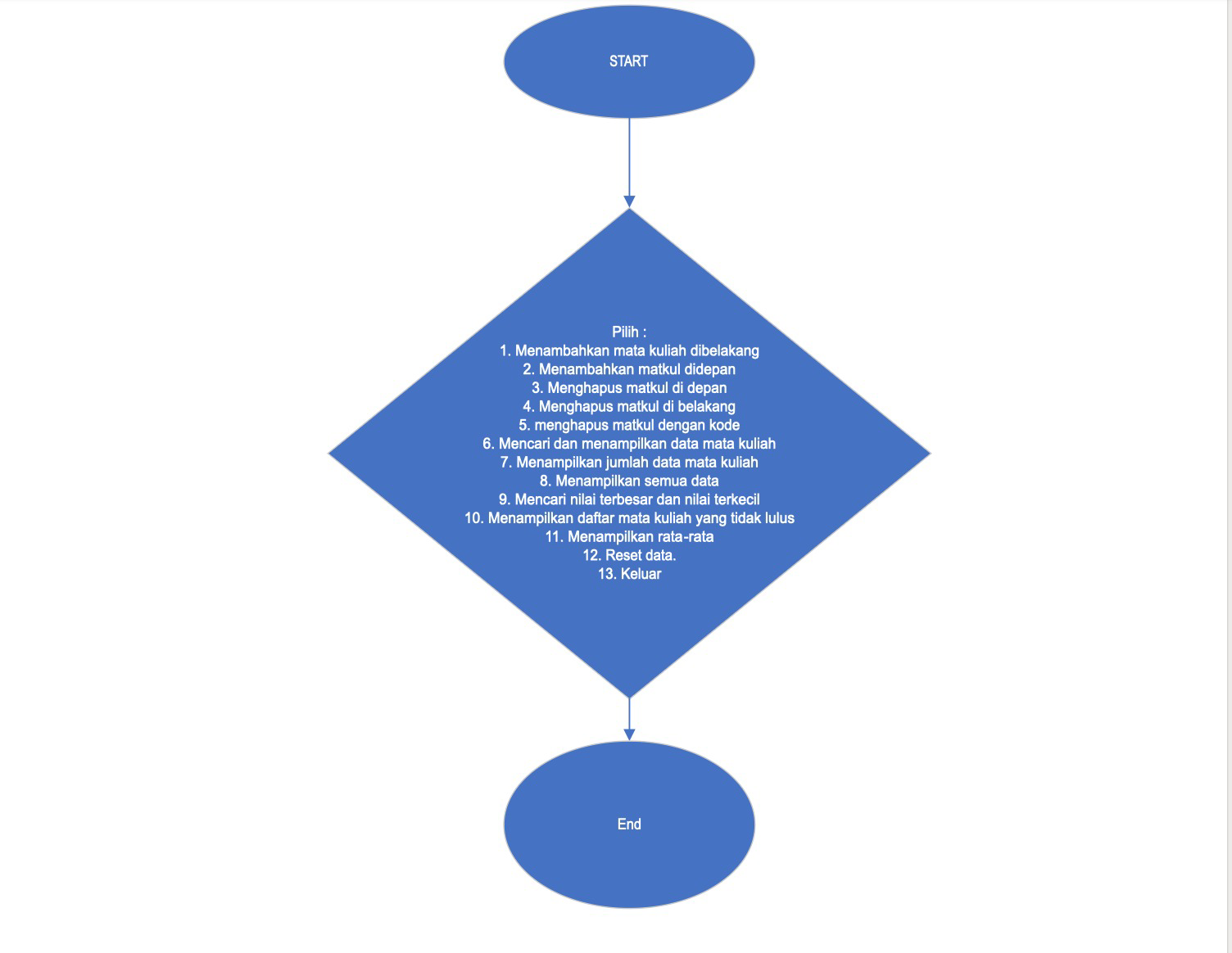
The program allows the user to enter course data into the system, both front and back, and to delete data on the front or back. In addition, the program can search course data by code, display the total amount of data, display all entered course data, search for maximum and minimum grades, display data on courses failed with grades below a certain threshold, calculate grade averages, and reset all data. The program has a security feature that prevents users from entering the same course code. We also provide functions to read and write course data to files. With this program, users can easily organize and manage their course data.

**BAB III**

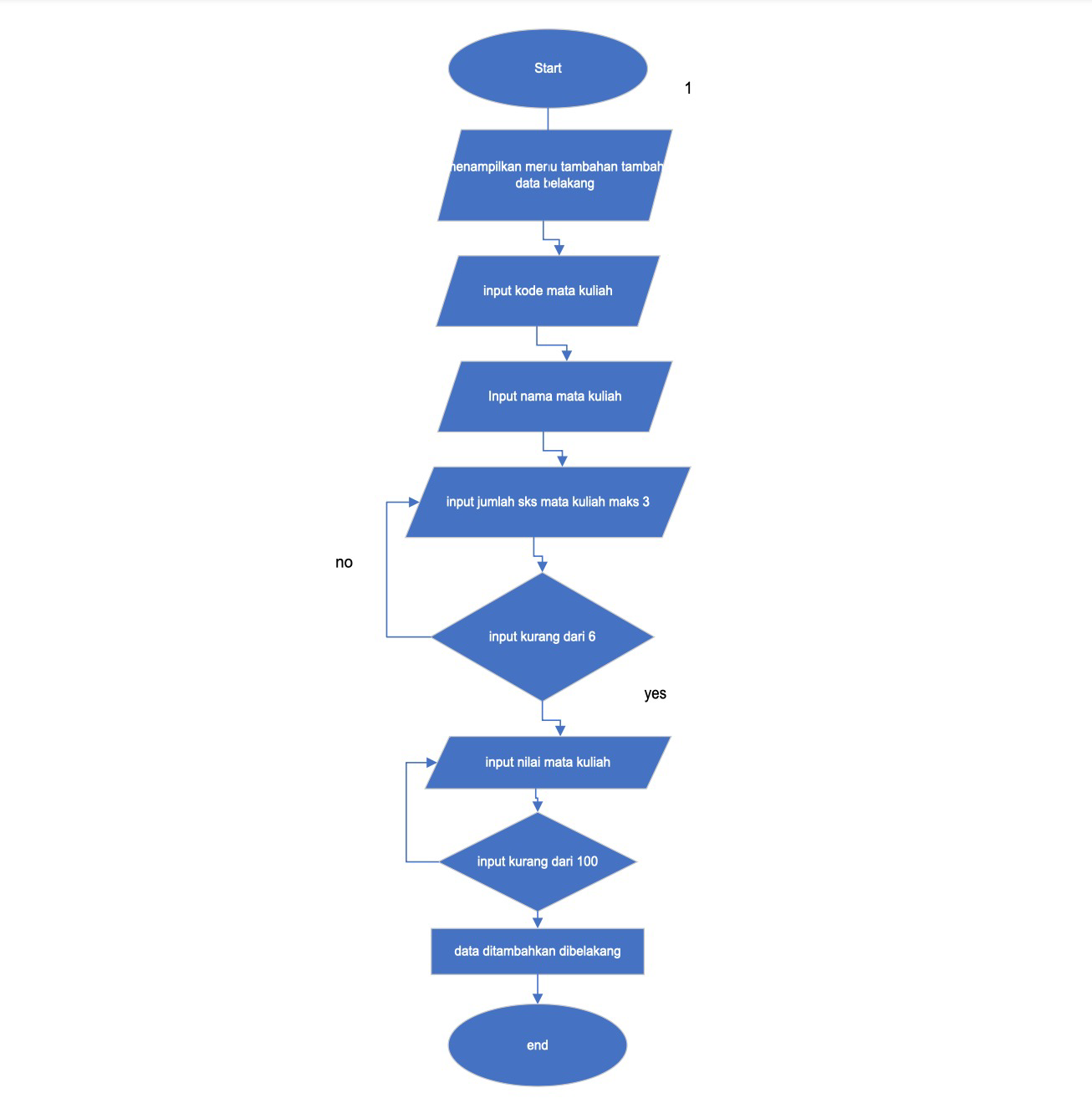
**PROGRAM**

**1. Pseudocode and Flowchart**

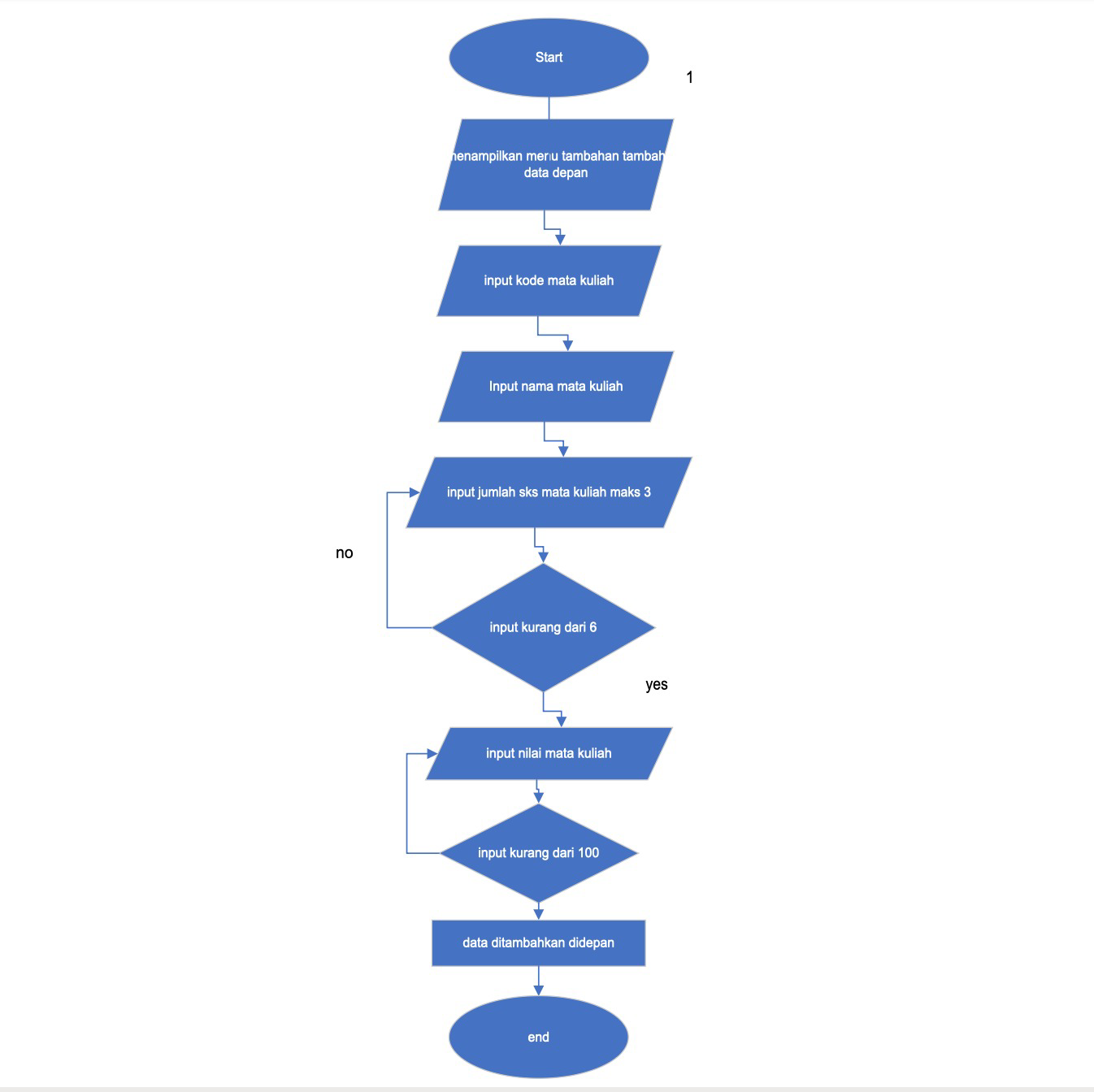
**Tampilan menu**

****

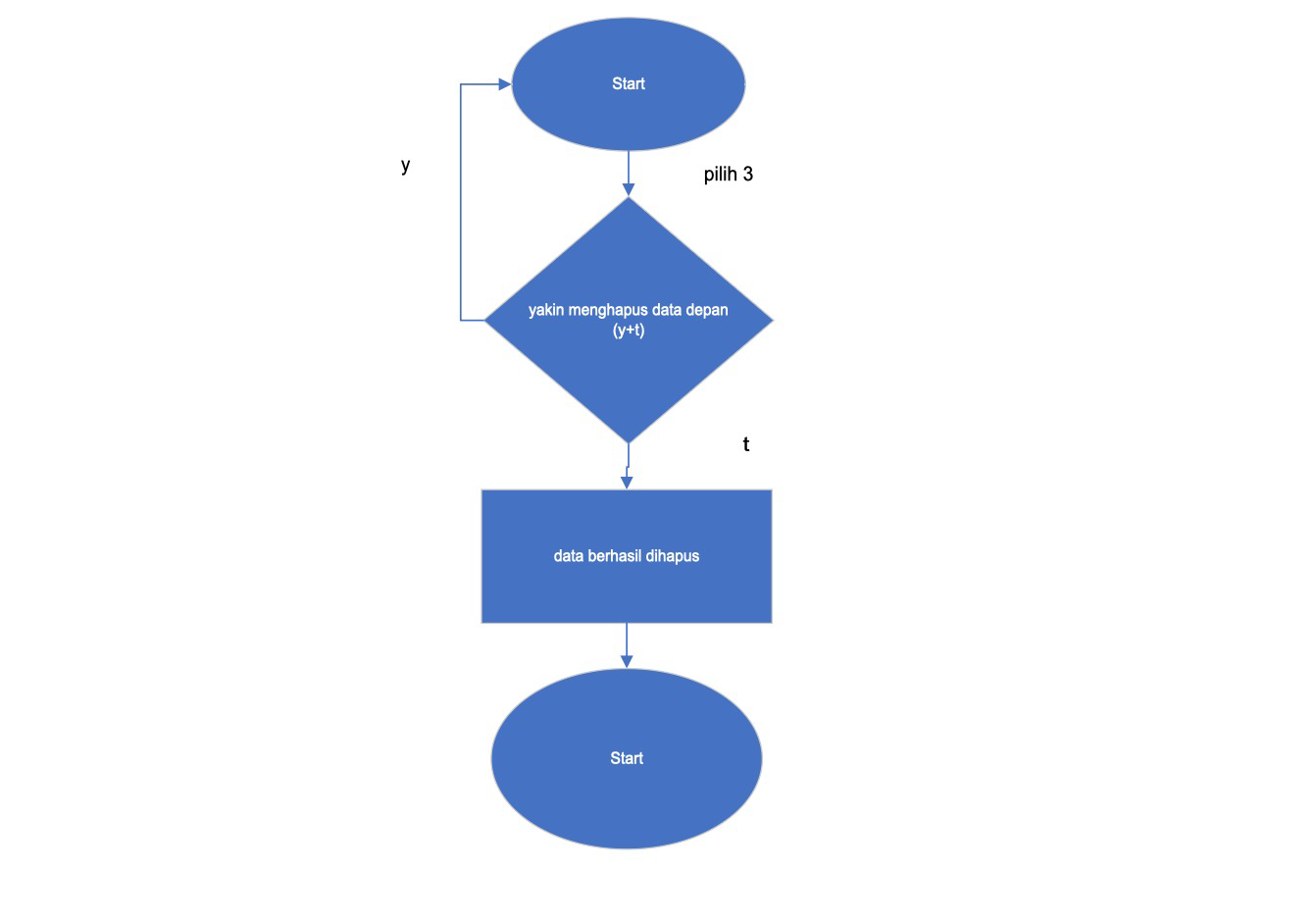
**Menu no 1**

****

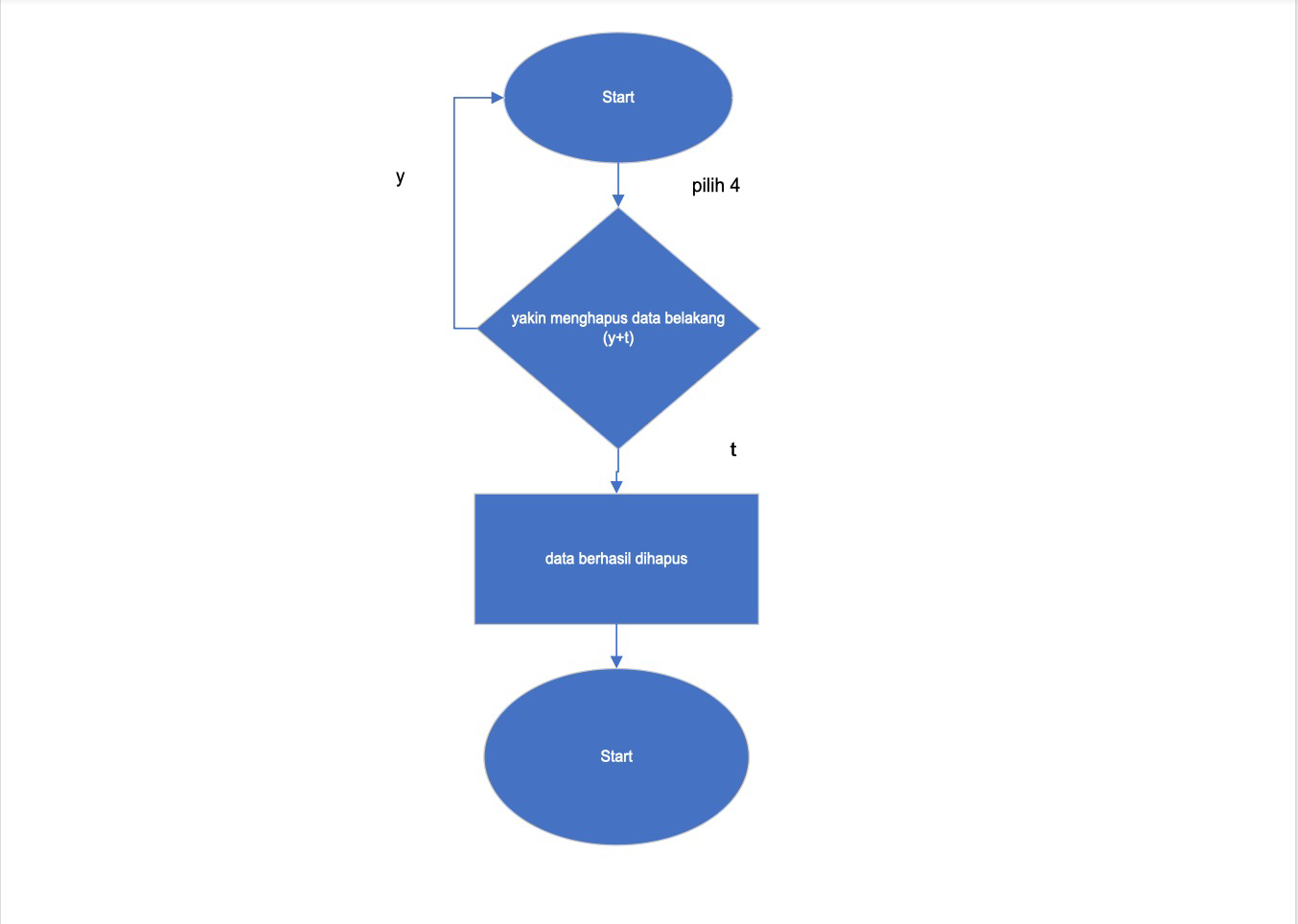
**Menu no 2**

****

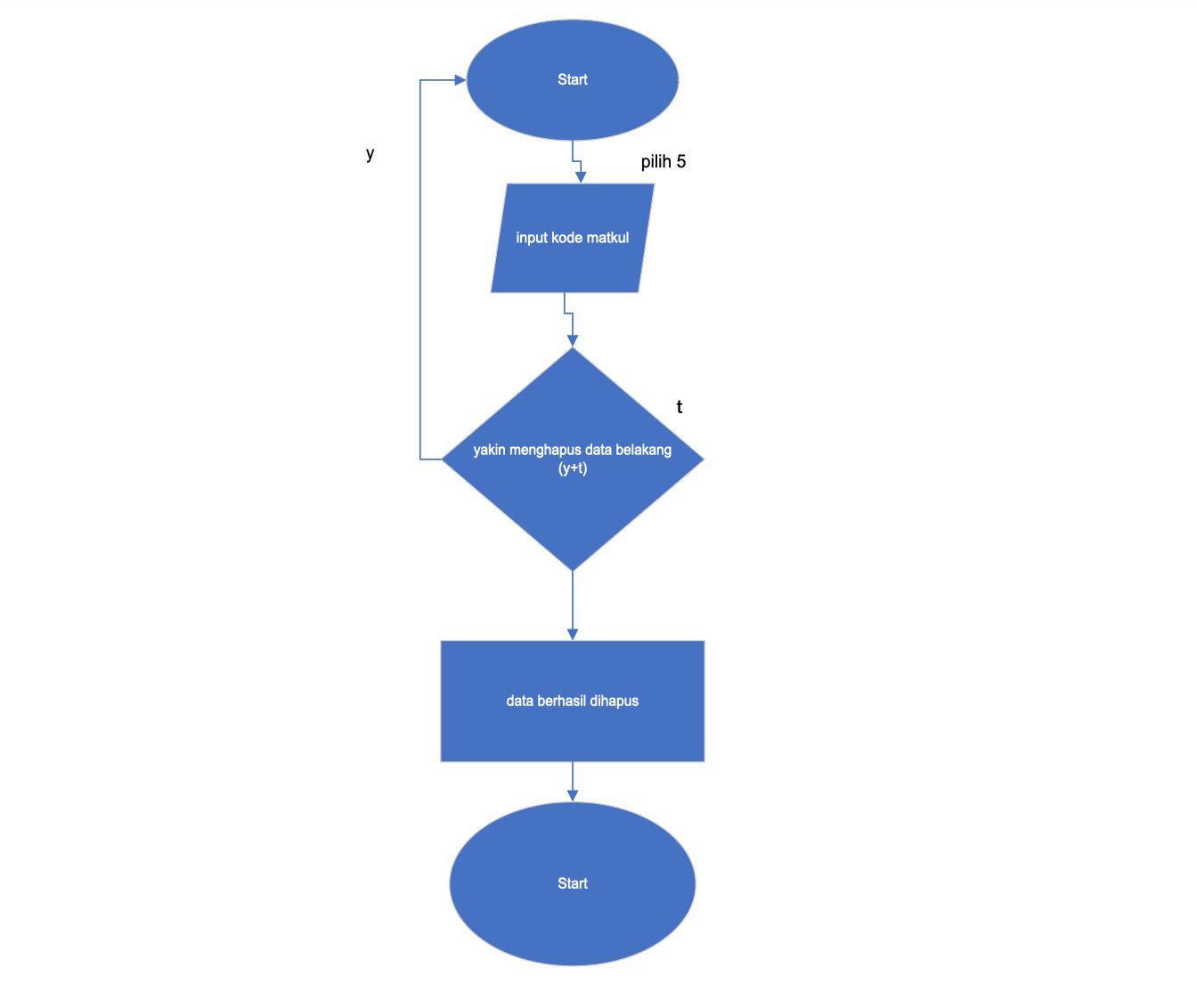
**Menu no 3**

****

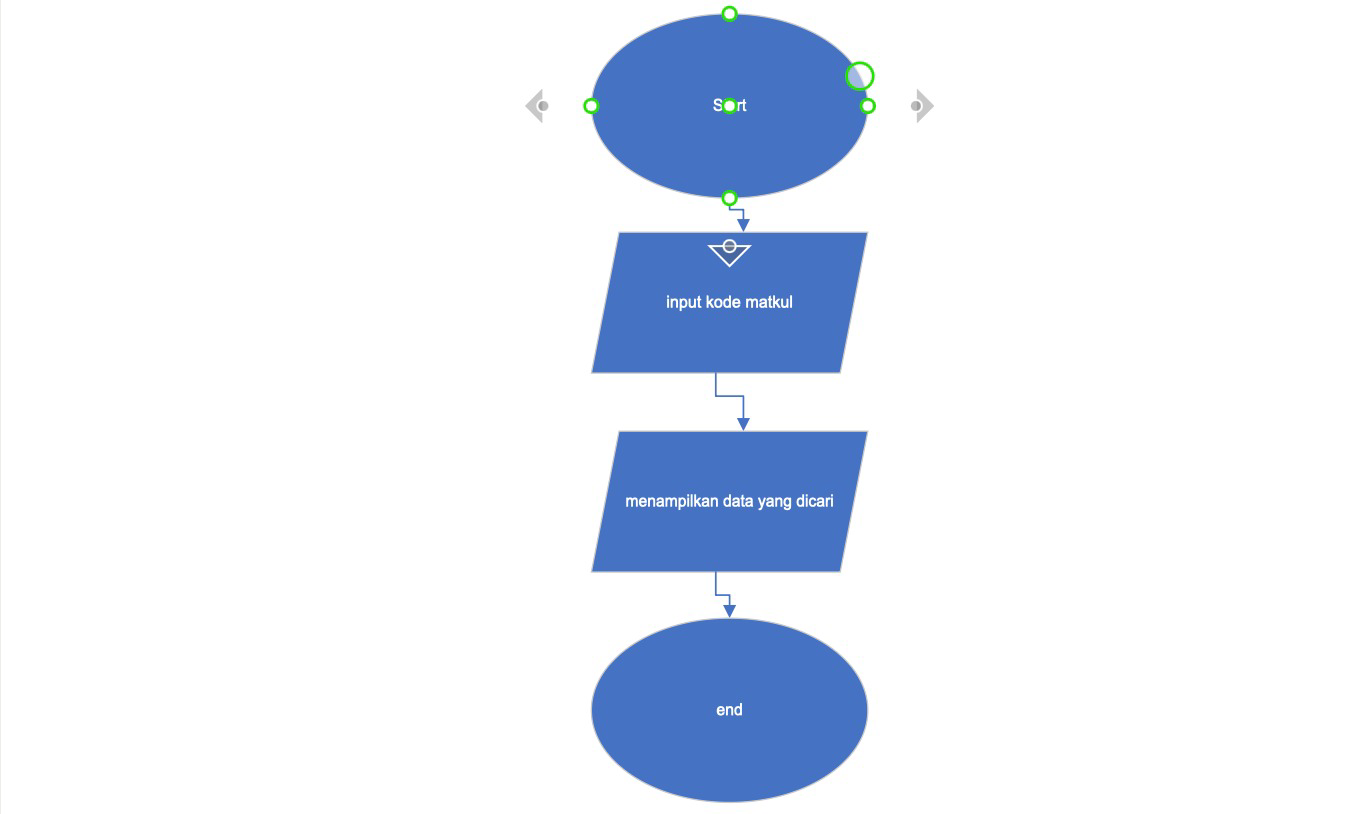
**Menu no 4**

****

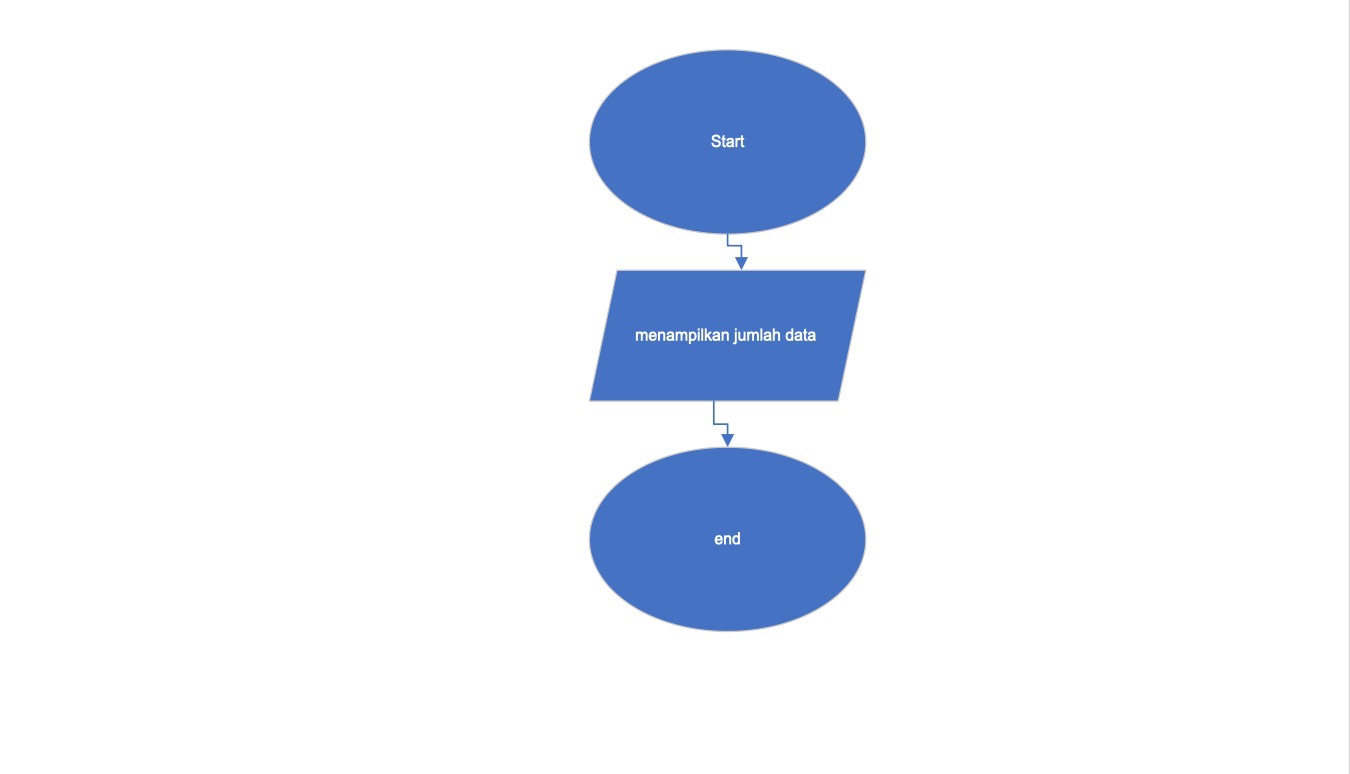
**Menu no 5**

****

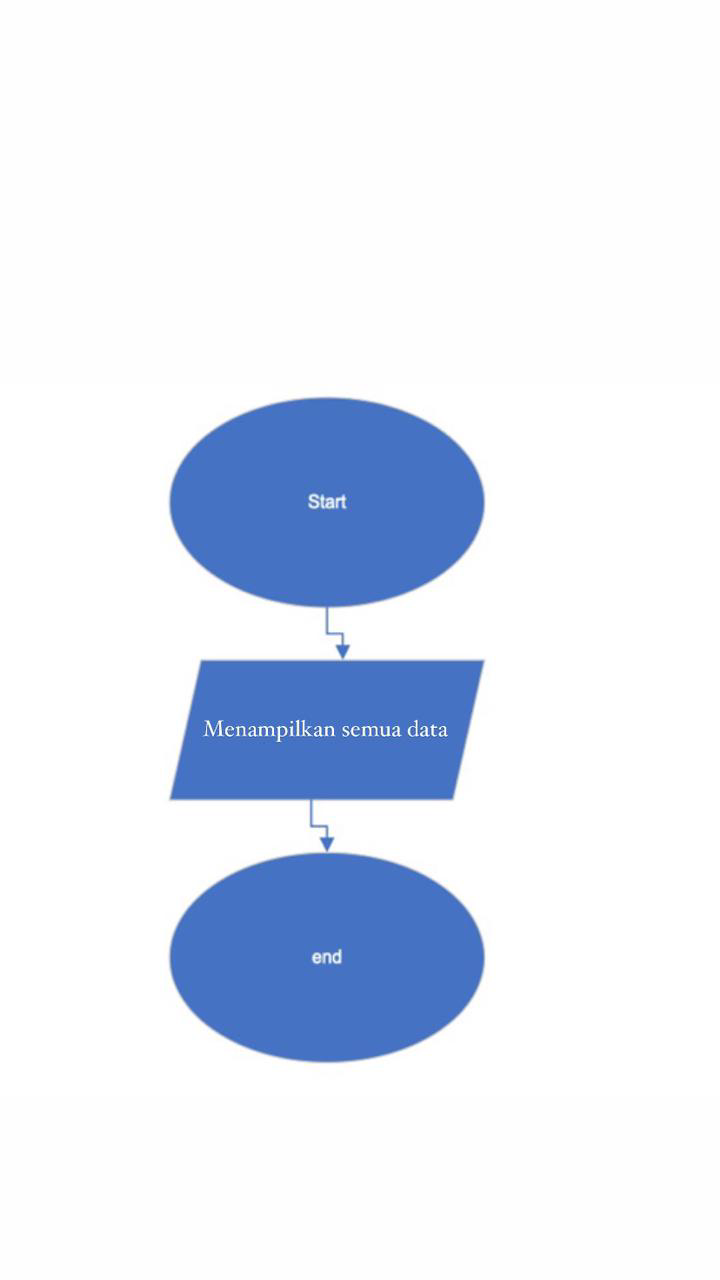
**Menu no 6**

****

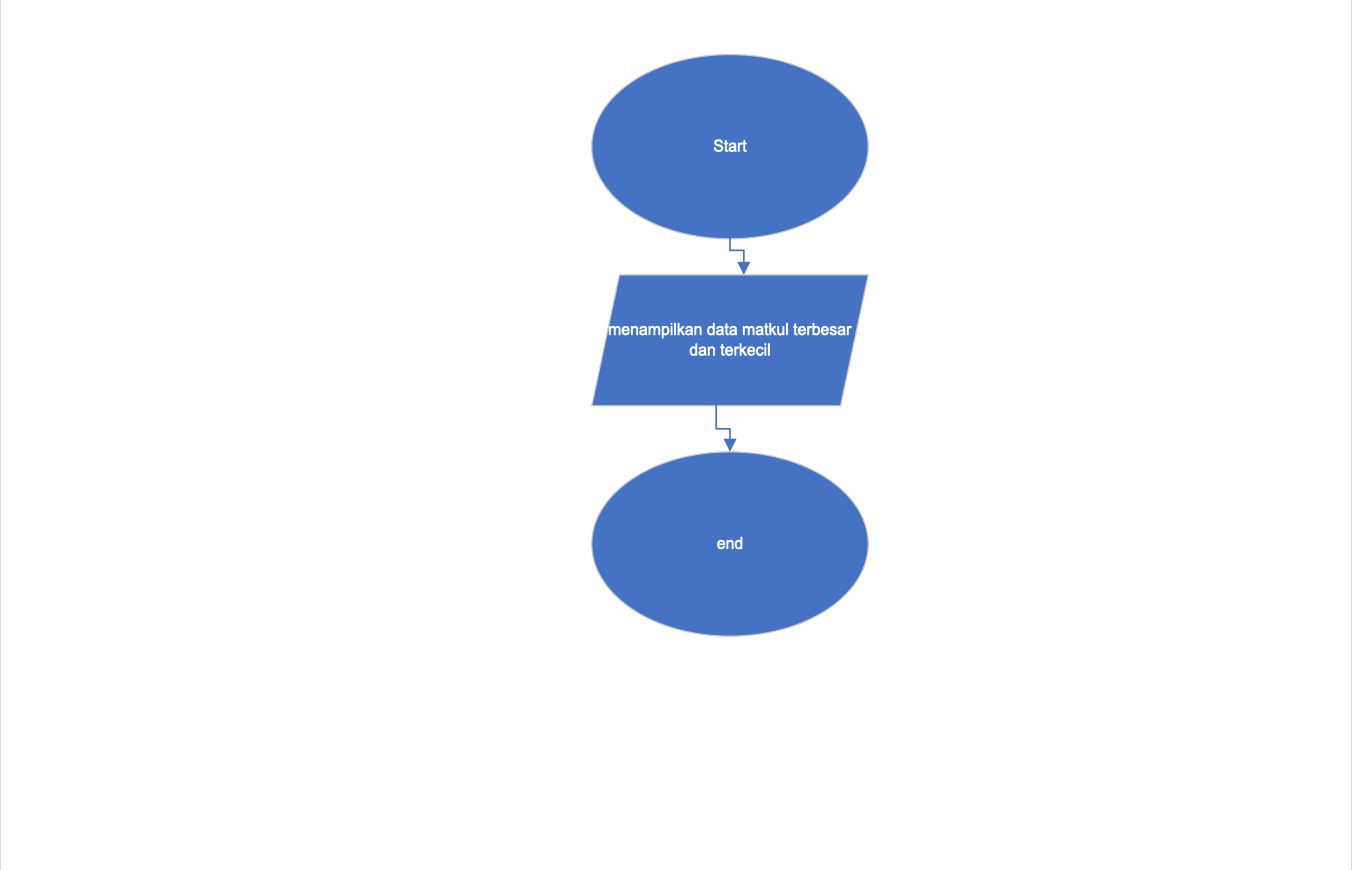
**Menu no 7**

****

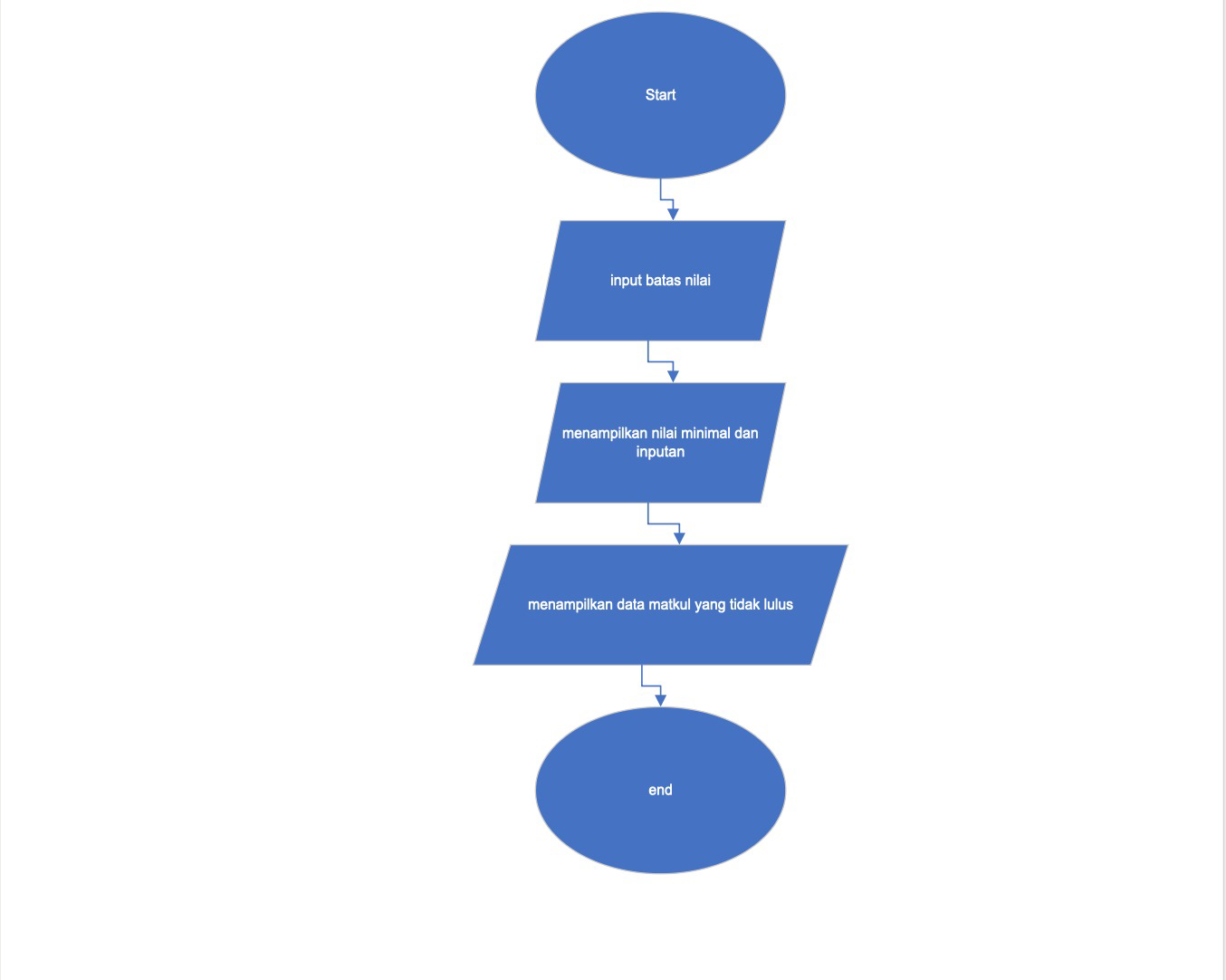
**Menu no 8**

****

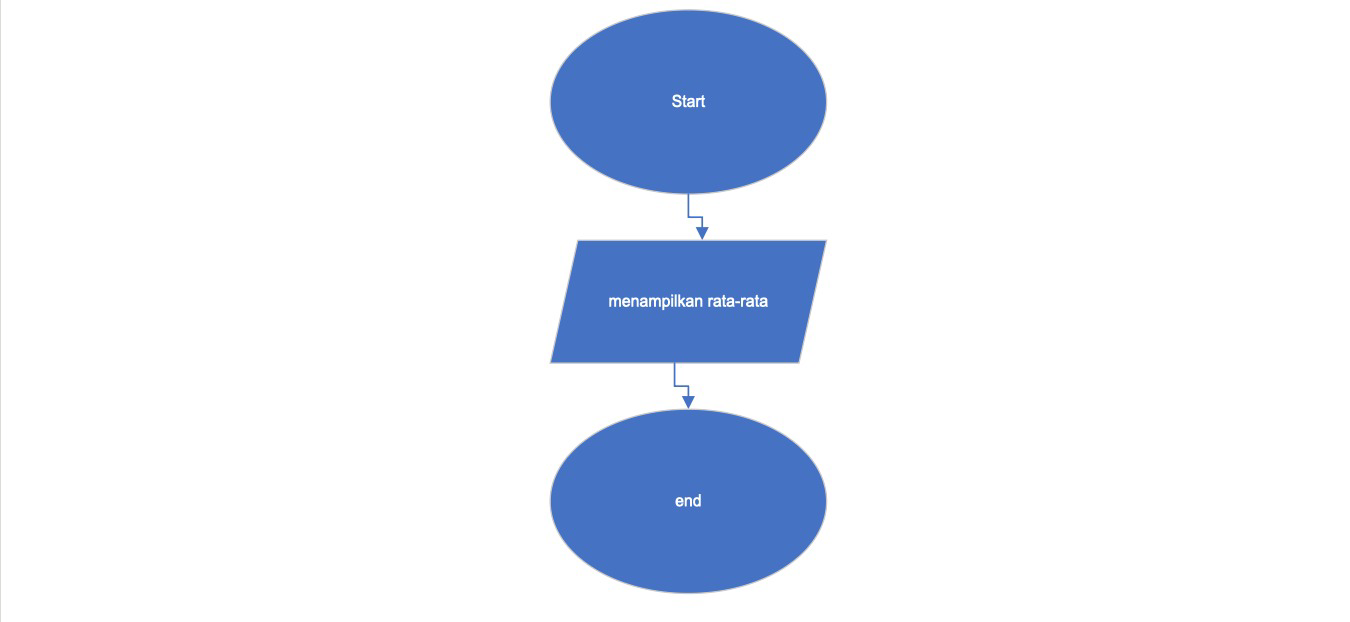
**Menu no 9**

****

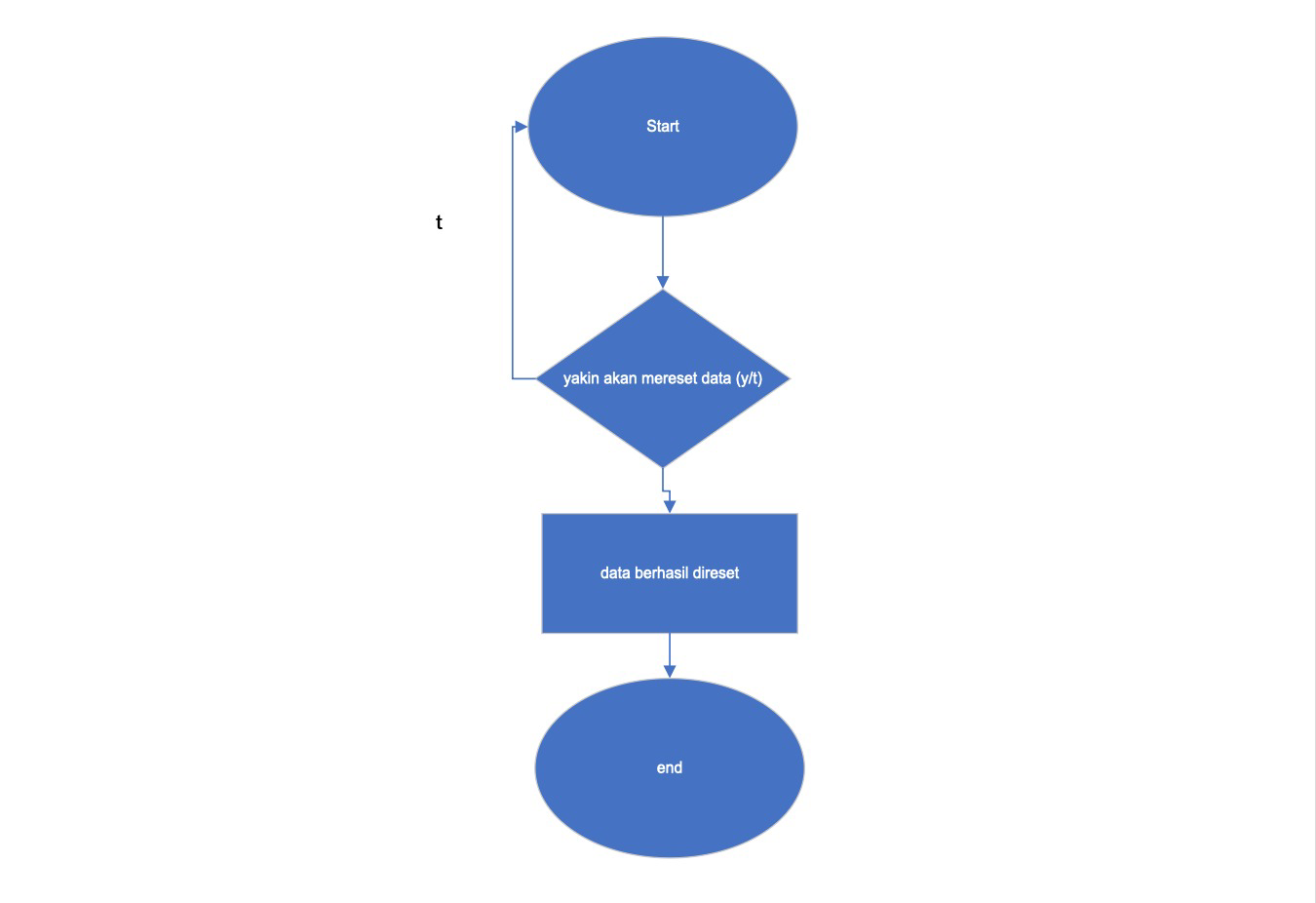
**Menu no 10**

****

**Menu no 11**

****

**Menu no 12**

****

**a).Pseudocode**

DECLARE kode[8] AS character array

DECLARE nama[50] AS character array

DECLARE nilai AS integer

DECLARE sks AS integer

DECLARE kepala AS pointer to matkul structure

DECLARE ekor AS pointer to matkul structure

DEFINE struct matkul

DECLARE nama[50] AS character array

DECLARE kode[8] AS character array

DECLARE sks AS integer

DECLARE nilai AS integer

DECLARE lanjut AS pointer to matkul structure

DECLARE kembali AS pointer to matkul structure

END struct matkul

DEFINE init() function

SET kepala = NULL

SET ekor = NULL

END init

DEFINE isEmpty() function

IF ekor == NULL

RETURN 1

ELSE

RETURN 0

END IF

END isEmpty

DEFINE data\_depan(kode, nama, sks, nilai) function

DECLARE temp AS pointer to matkul structure

ALLOCATE memory for temp

COPY kode to temp->kode

COPY nama to temp->nama

SET temp->nilai = nilai

SET temp->sks = sks

IF isEmpty() is equal to 1

SET kepala = temp

SET ekor = kepala

SET kepala->lanjut = kepala

SET kepala->kembali = kepala

SET ekor->lanjut = ekor

SET ekor->kembali = ekor

ELSE

SET temp->lanjut = kepala

SET kepala->kembali = temp

SET kepala = temp

SET kepala->kembali = ekor

SET ekor->lanjut = kepala

END IF

PRINT "====================================="

PRINT "===========Data telah ditambahkan========="

PRINT "Data dengan kode :", temp->kode

PRINT "Data dengan nama :", temp->nama

PRINT "Data berhasil ditambahkan di depan!"

END data\_depan

DEFINE jam() function

DECLARE waktu AS pointer to struct tm

DECLARE Timeval AS time\_t

SET Timeval = time(0)

SET waktu = localtime(&Timeval)

PRINT waktu converted to string

END jam

DEFINE data\_belakang(kode, nama, sks, nilai) function

DECLARE temp AS pointer to matkul structure

ALLOCATE memory for temp

COPY kode to temp->kode

COPY nama to temp->nama

SET temp->nilai = nilai

SET temp->sks = sks

IF isEmpty() is equal to 1

SET kepala = temp

SET ekor = kepala

SET kepala->lanjut = kepala

SET kepala->kembali = kepala

SET ekor->lanjut = ekor

SET ekor->kembali = ekor

ELSE

SET ekor->lanjut = temp

SET temp->kembali = ekor

SET ekor = temp

SET ekor->lanjut = kepala

SET kepala->kembali = ekor

END IF

PRINT "====================================="

PRINT "===Data telah ditambahkan==="

PRINT "Data dengan kode :", temp->kode

PRINT "Data dengan nama :", temp->nama

PRINT "Data berhasil ditambahkan di belakang!"

END data\_belakang

DEFINE hapus\_depan() function

DECLARE del AS pointer to matkul structure

DECLARE d[8] AS character array

IF kepala is not equal to ekor

SET del = kepala

COPY del->kode to d

SET kepala = kepala->lanjut

SET kepala->kembali = ekor

SET ekor->lanjut = kepala

FREE del

ELSE

COPY kepala->kode to d

SET kepala = NULL

SET ekor = NULL

END IF

PRINT "Data", d, "berhasil dihapus!"

END hapus\_depan

DEFINE hapus\_belakang() function

DECLARE del AS pointer to matkul structure

DECLARE d[8] AS character array

IF kepala is not equal to ekor

SET del = ekor

COPY del->kode to d

SET ekor = ekor->kembali

SET ekor->lanjut = kepala

SET kepala->kembali = ekor

FREE del

ELSE

COPY kepala->kode to d

SET kepala = NULL

SET ekor = NULL

END IF

PRINT "Data", d, "berhasil dihapus!"

END hapus\_belakang

DEFINE cari\_data(cari) function

DECLARE help AS pointer to matkul structure

DECLARE s\_earch AS pointer to matkul structure

DECLARE index AS integer

SET help = kepala

SET index = 1

DO

IF strcmp(help->kode, cari) is equal to 0

SET s\_earch = help

PRINT "Data kode", cari, "dengan baris", index

RETURN

END IF

INCREMENT index by 1

SET help = help->lanjut

WHILE help is not equal to ekor->lanjut

PRINT "Data kode", cari, "tidak ditemukan"

END cari\_data

DEFINE menampilkan\_jumlah() function

DECLARE help AS pointer to matkul structure

SET help = kepala

DECLARE n AS integer

SET n = 0

IF isEmpty() is equal to 1

PRINT "Data masih kosong"

RETURN

ELSE

DO

INCREMENT n by 1

SET help = help->lanjut

WHILE help is not equal to ekor->lanjut

PRINT "Jumlah data =", n

PRINT "-----------------"

END IF

END menampilkan\_jumlah

DEFINE menampilkan\_semua() function

DECLARE help AS pointer to matkul structure

SET help = kepala

DECLARE n AS integer

SET n = 1

IF isEmpty() is equal to 1

PRINT "Data masih kosong, silahkan menambahkan data terlebih dahulu"

RETURN

ELSE

PRINT "===========D A T A M A T A K U L I A H==========="

DO

PRINT "Mata kuliah ke :", n

PRINT "Kode mata kuliah :", help->kode

PRINT "Nama mata kuliah :", help->nama

PRINT "Nilai mata kuliah :", help->nilai

PRINT "Jumlah SKS :", help->sks

PRINT "-------------------------------------"

SET help = help->lanjut

INCREMENT n by 1

WHILE help is not equal to ekor->lanjut

PRINT newline

END IF

END menampilkan\_semua

DEFINE nilaiMinMax() function

DECLARE help, terbesar, terkecil AS pointers to matkul structure

SET help = kepala

SET terbesar = help

SET terkecil = help

IF isEmpty() is equal to 1

PRINT "Belum terdapat data"

RETURN

ELSE

DO

IF help->nilai > terbesar->nilai

SET terbesar = help

END IF

IF help->nilai < terkecil->nilai

SET terkecil = help

END IF

SET help = help->lanjut

WHILE help is not equal to ekor->lanjut

PRINT "===========Data Matkul Nilai Terbesar==========="

PRINT "Kode mata kuliah :", terbesar->kode

PRINT "Nama mata kuliah :", terbesar->nama

PRINT "Jumlah SKS :", terbesar->sks

PRINT "Nilai :", terbesar->nilai

PRINT "==========================================="

PRINT "-----------Data Matkul Nilai Terkecil-----------"

PRINT "Kode mata kuliah :", terkecil->kode

PRINT "Nama mata kuliah :", terkecil->nama

PRINT "Jumlah SKS :", terkecil->sks

PRINT "Nilai :", terkecil->nilai

PRINT "------------------------------------------"

END IF

END nilaiMinMax

DEFINE lulus(nilai: integer) function

DECLARE help AS pointers to matkul

DECLARE jumlah: integer

SET help = kepala

PRINT "Nilai Minimal: ", nilai

PRINT "==========Data Matkul Yang Tidak Lulus=========="

DO

IF help->nilai < nilai then

print "-------------------------------------"

print "Kode mata kuliah: ", help->kode

print "Nama mata kuliah: ", help->nama

print "Jumlah SKS: ", help->sks

print "Nilai mata kuliah: ", help->nilai

print "Mata kuliah tidak lulus!"

print "-------------------------------------"

jumlah = jumlah + 1

END IF

help = help->lanjut

WHILE help != ekor->lanjut

IF jumlah = 0

THEN

PRINT("+------------------------------------+")

PRINT("Semua mata kuliah lulus!!")

END IF

END lulus

DEFINE average() function

DECLARE help AS pointers to matkul

DECLARE jumlah, n AS integer

DECLARE rata AS float

SET help = kepala

SET jumlah = 0

SET n = 0

DO

jumlah = jumlah + help->nilai

n = n + 1

help = help->lanjut

WHILE help != ekor->lanjut

SET rata = jumlah / n

PRINT("=========Rata-rata=========")

PRINT("Jumlah nilai :", jumlah)

PRINT("Jumlah data :", n)

PRINT("Rata-rata :", rata)

PRINT("===========================")

END average

DEFINE reset() function

DECLARE help, hapus AS pointers to matkul

SET help = kepala

DO

hapus = help

help = kepala->lanjut

free(hapus)

WHILE help->lanjut != kepala

SET kepala = NULL

SET ekor = NULL

PRINT("Data telah terhapus")

END reset

DECLARE cek\_sks(newSks: integer) function

IF newSks > 3 then

DO

PRINT("Maaf, rentang jumlah SKS salah..")

PRINT("Masukkan jumlah SKS (maksimum 3): ")

input newSks

flush input buffer

WHILE newSks > 3

END IF

SET sks = newSks

END cek\_sks

DEFINE cek\_kodeKuliah(kode: string) function

DECLARE s\_earch AS pointers to matkul

SET s\_earch = kepala

IF isEmpty() is equal to 0 then

DO

IF strcmp(s\_earch->kode, kode) is equal to 0 then

DO

PRINT("Kode kuliah sudah ada")

PRINT("Masukkan kode mata kuliah: ")

input kode

WHILE strcmp(s\_earch->kode, kode) is equal to 0

END IF

s\_earch = s\_earch->lanjut

WHILE s\_earch is not equal to ekor

END IF

END cek\_kodeKuliah

DEFINE cekMinMax(score: integer) function

IF score > 100 THEN

DO

print("Input nilai salah, rentang nilai (1-100)")

print("Masukkan nilai matkul: ")

input score

WHILE score > 100

SET nilai = score

END IF

END cekMinMax

DEFINE delete(poin: pointer to char) function

del: pointer to matkul

help: pointer to matkul

SET help = kepala

DO

IF strcmp(help->kode, poin) = 0 then

del <- help

IF help = kepala and kepala = ekor then

kepala <- NULL

ekor <- NULL

ELSE IF help = kepala then

kepala <- kepala->lanjut

kepala->kembali <- ekor

ekor->lanjut <- kepala

free(del)

else if help = ekor then

ekor <- ekor->kembali

ekor->lanjut <- kepala

kepala->kembali <- ekor

free(del)

else

help->kembali->lanjut <- help->lanjut

help->lanjut->kembali <- help->kembali

free(del)

END IF

print "Data dengan kode mata kuliah " + poin + " telah dihapus"

return

end if

help <- help->lanjut

while help != ekor->lanjut

print "Data tidak ditemukan " + poin

END delete

DEFINE loading() function

DECLARE i

PRINT "\n\n\n\n\n\n\n"

PRINT "\n\n\n\t\t\t\t\t Loading......"

PRINT "\n\n"

PRINT "\t\t\t\t\t"

FOR i = 0 TO 35

PRINT 178

\_sleep(10)

PRINT "\n\n\t\t\t\t\t 'PRESS ENTER TO START'"

SYSTEM "pause>0"

SYSTEM "cls"

END loading

FUNCTION main()

SYSTEM "color B"

CALL loading()

CALL init()

DECLARE tidak\_lulus AS Integer

DECLARE menu AS Integer

DECLARE kirim[5] AS Character

DECLARE del[10] AS Character

DECLARE cari[15] AS Character

DECLARE poin[20] AS Character

DO

PRINT "\n=====================MENU DATA MATA KULIAH=====================\n"

PRINT "Menu"

PRINT "1. Menambahkan mata kuliah di belakang"

PRINT "2. Menambahkan mata kuliah di depan"

PRINT "3. Menghapus mata kuliah di depan"

PRINT "4. Menghapus mata kuliah di belakang"

PRINT "5. Menghapus mata kuliah dengan kode"

PRINT "6. Mencari dan menampilkan data mata kuliah"

PRINT "7. Menampilkan jumlah data mata kuliah"

PRINT "8. Menampilkan semua data"

PRINT "9. Mencari nilai terbesar dan nilai terkecil"

PRINT "10. Menampilkan daftar mata kuliah yang tidak lulus"

PRINT "11. Menampilkan rata-rata"

PRINT "12. Reset data"

PRINT "13. Keluar"

PRINT "===============================================================\n"

PRINT "Masukkan pilihan : "

READ menu

FLUSH\_INPUT()

SWITCH menu

CASE 1:

FLUSH\_INPUT()

PRINT "=====================TAMBAH DATA BELAKANG=====================\n"

PRINT "Masukkan data"

PRINT "Kode mata kuliah : "

READ kode

FLUSH\_INPUT()

CALL cek\_kodeKuliah(kode)

PRINT "Nama mata kuliah : "

READ nama

FLUSH\_INPUT()

PRINT "Jumlah SKS mata kuliah (max 3) : "

READ sks

FLUSH\_INPUT()

CALL cek\_sks(sks)

PRINT "Masukkan nilai mata kuliah (max 100) : "

READ nilai

FLUSH\_INPUT()

CALL cekMinMax(nilai)

SYSTEM "cls"

CALL data\_belakang(kode, nama, sks, nilai)

BREAK

CASE 2:

FLUSH\_INPUT()

PRINT "=====================TAMBAH DATA DEPAN=====================\n"

PRINT "Masukkan data"

PRINT "Kode mata kuliah : "

READ kode

FLUSH\_INPUT()

CALL cek\_kodeKuliah(kode)

PRINT "Nama mata kuliah : "

READ nama

FLUSH\_INPUT()

PRINT "Jumlah SKS mata kuliah (max 3) : "

READ sks

FLUSH\_INPUT()

CALL cek\_sks(sks)

PRINT "Masukkan nilai mata kuliah (max 100) : "

READ nilai

FLUSH\_INPUT()

CALL cekMinMax(nilai)

SYSTEM "cls"

CALL data\_depan(kode, nama, sks, nilai)

BREAK

CASE 3:

FLUSH\_INPUT()

IF isEmpty() == 1 THEN

PRINT "Maaf tidak terdapat data\n"

ELSE

SYSTEM "cls"

PRINT "Apakah anda yakin akan menghapus data di depan? (y or t) : "

READ kirim

FLUSH\_INPUT()

IF strcmp(kirim, "y") == 0 THEN

CALL hapus\_depan()

ELSE

PRINT "Data gagal dihapus!\n"

SYSTEM "cls"

END IF

END IF

BREAK

CASE 4:

FLUSH\_INPUT()

IF isEmpty() == 1 THEN

PRINT "Maaf tidak terdapat data\n"

ELSE

SYSTEM "cls"

PRINT "Apakah anda yakin akan menghapus data di belakang? (y or t) : "

READ kirim

FLUSH\_INPUT()

IF strcmp(kirim, "y") == 0 THEN

CALL hapus\_belakang()

ELSE

PRINT "Data gagal dihapus!\n"

END IF

END IF

BREAK

CASE 5:

FLUSH\_INPUT()

IF isEmpty() == 1 THEN

PRINT "Maaf tidak terdapat data\n"

ELSE

SYSTEM "cls"

PRINT "Masukkan kode mata kuliah : "

READ del

PRINT "Apakah anda yakin akan menghapus data? (y or t) : "

READ kirim

FLUSH\_INPUT()

IF strcmp(kirim, "y") == 0 THEN

CALL delete(del)

ELSE

PRINT "Data gagal dihapus!\n"

END IF

END IF

BREAK

CASE 6:

SYSTEM "cls"

PRINT "Masukkan kode mata kuliah : "

READ cari

FLUSH\_INPUT()

CALL cari\_data(cari)

BREAK

CASE 7:

FLUSH\_INPUT()

SYSTEM "cls"

CALL menampilkan\_jumlah()

BREAK

CASE 8:

FLUSH\_INPUT()

SYSTEM "cls"

CALL menampilkan\_semua()

BREAK

CASE 9:

FLUSH\_INPUT()

SYSTEM "cls"

CALL nilaiMinMax()

BREAK

CASE 10:

FLUSH\_INPUT()

IF isEmpty() == 1 THEN

PRINT "Data mata kuliah kosong\n"

ELSE

SYSTEM "cls"

PRINT "Data mata kuliah tidak lulus\n"

PRINT "Masukkan batas nilai : "

READ nilai

FLUSH\_INPUT()

CALL lulus(nilai)

END IF

BREAK

CASE 11:

FLUSH\_INPUT()

SYSTEM "cls"

CALL average()

BREAK

CASE 12:

FLUSH\_INPUT()

IF isEmpty() == 1 THEN

PRINT "Maaf tidak terdapat data\n"

ELSE

SYSTEM "cls"

PRINT "Apakah anda yakin akan mereset data? (y or t) : "

READ kirim

FLUSH\_INPUT()

IF strcmp(kirim, "y") == 0 THEN

CALL reset()

ELSE

PRINT "Data gagal dihapus!\n"

END IF

END IF

BREAK

CASE 13:

PRINT ">>Terima kasih telah menggunakan program ini!<<\n"

EXIT

BREAK

DEFAULT:

PRINT "Maaf input anda tidak sesuai\n"

CALL getch()

SYSTEM "cls"

BREAK

END SWITCH

WHILE menu != 13

END FUNCTION

**b).Flowchart**

**Menu 5**

**2. Overview of the program**

**2.1 Overview of the program**

Program yang dibuat adalah sistem manajemen data mata kuliah yang memungkinkan pengguna untuk menambahkan, menghapus, mencari, dan mengelola data mata kuliah. Fitur-fitur yang disediakan termasuk pencarian berdasarkan kode mata kuliah, perhitungan jumlah data, menampilkan data secara keseluruhan, mencari nilai maksimum dan minimum, menampilkan data mata kuliah yang tidak lulus, menghitung rata-rata nilai, dan reset data. Program ini juga dilengkapi dengan fitur pengamanan untuk mencegah pengguna memasukkan kode mata kuliah yang sama, serta dapat membaca dan menulis data ke dalam file. Dengan demikian, program ini dapat membantu pengguna dalam mengatur dan menganalisis data mata kuliah secara efisien.

**2.2 Layout design**

Program di atas adalah sebuah program dalam bahasa C yang digunakan untuk mengelola data mata kuliah. Program ini menggunakan konsep linked list untuk menyimpan dan mengatur data mata kuliah. Setiap data mata kuliah memiliki informasi seperti kode, nama, sks (sistem kredit semester), dan nilai. Program ini menyediakan fungsi untuk menambah data mata kuliah di depan atau di belakang, menghapus data mata kuliah dari depan atau belakang, mencari data mata kuliah berdasarkan kode, menampilkan jumlah data, menampilkan semua data mata kuliah, menampilkan nilai tertinggi dan terendah, menampilkan data mata kuliah yang tidak lulus berdasarkan nilai batas, menghitung rata-rata nilai, dan menghapus semua data. Program juga membaca dan menulis data mata kuliah ke dalam file teks "matakuliah.txt".

**2.3 Functions of the program**

Program pengelolaan data mata kuliah menggunakan linked list dalam bahasa pemrograman C adalah sebuah aplikasi yang dirancang untuk membantu pengguna dalam mengatur dan memanipulasi informasi mata kuliah. Program ini memiliki berbagai fungsi, seperti menambahkan mata kuliah di bagian depan atau belakang linked list, menghapus mata kuliah, mencari mata kuliah berdasarkan kode, menampilkan jumlah total mata kuliah, menampilkan semua mata kuliah, menampilkan mata kuliah dengan nilai tertinggi dan terendah, menampilkan mata kuliah yang tidak lulus, menghitung rata-rata nilai, menghapus semua mata kuliah, membaca data dari file, dan menulis data ke file. Dengan program ini, pengguna dapat dengan mudah mengelola dan mengakses informasi mata kuliah secara efisien.

**2.4 Program details.**

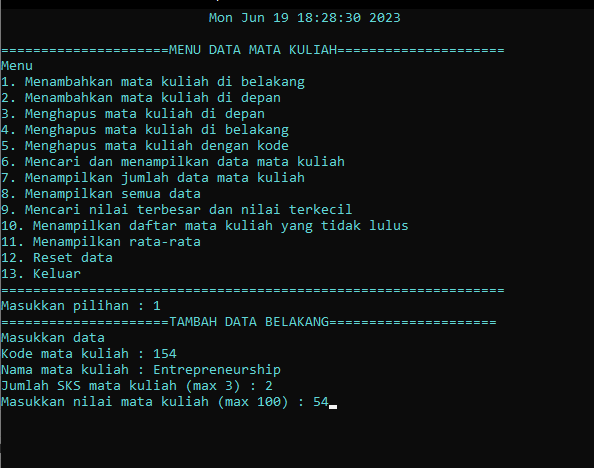
Program di atas merupakan implementasi dari sebuah sistem pengelolaan data mata kuliah menggunakan linked list dalam bahasa pemrograman C. Program tersebut memiliki fitur-fitur seperti menambahkan data mata kuliah di depan atau belakang, menghapus data mata kuliah di depan atau belakang, mencari data mata kuliah berdasarkan kode, menampilkan jumlah data, menampilkan semua data, menampilkan data mata kuliah dengan nilai tertinggi dan terendah, menampilkan data mata kuliah yang tidak lulus, menghitung rata-rata nilai, menghapus semua data, serta membaca dan menulis data ke dalam file. Program juga memiliki beberapa validasi seperti memastikan jumlah SKS tidak melebihi 3, memastikan kode mata kuliah unik, dan memastikan nilai mata kuliah berada dalam rentang 1 hingga 100. Program ini dapat membantu pengguna dalam mengelola dan melihat informasi mengenai data mata kuliah yang telah diinput.

**BAB IV**

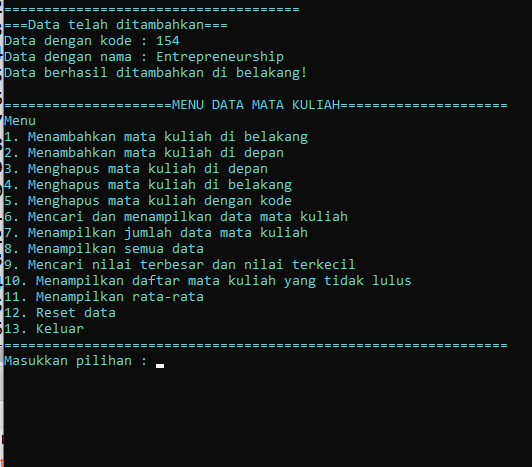
**Result**

* **Program Screenshot**

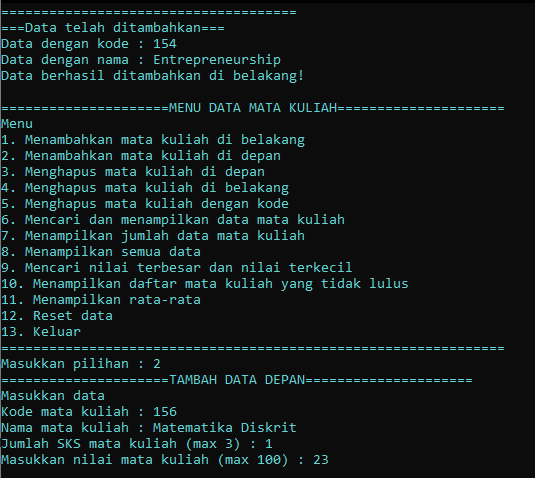
1. **Menu menambahkan mata kuliah di belakang**

****

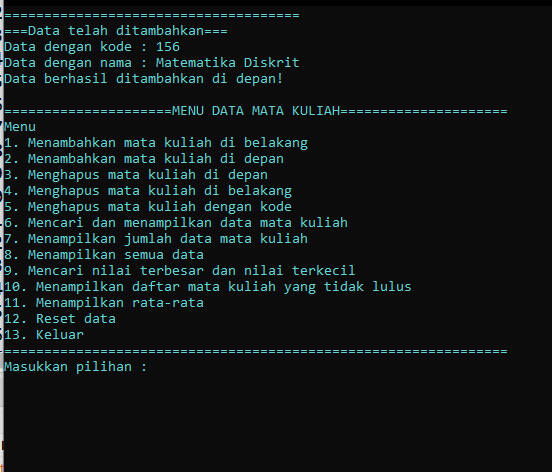
1. **Hasil menu menambahkan mata kuliah di belakang**

****

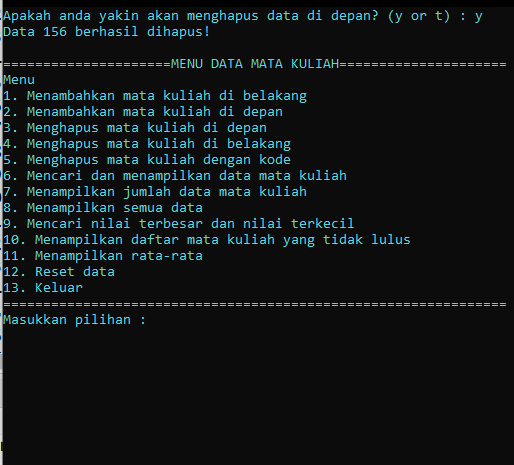
1. **Menu menambahkan mata kuliah di depan**

****

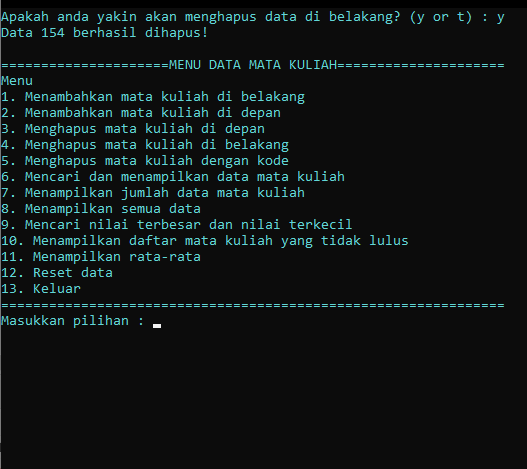
1. **Hasil menu menambahkan mata kuliah di depan**

****

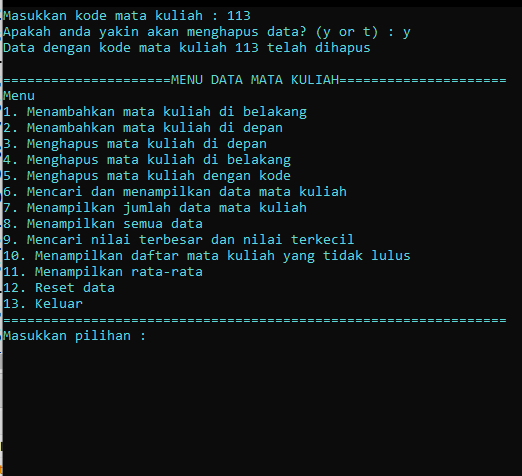
1. **Menu menghapus mata kuliah di depan**

****

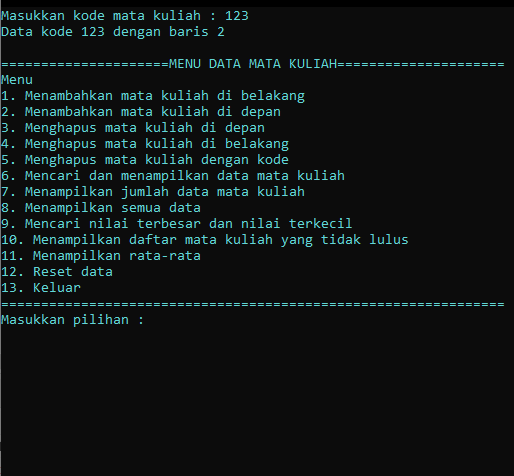
1. **Menu menghapus mata kuliah di depan**

****

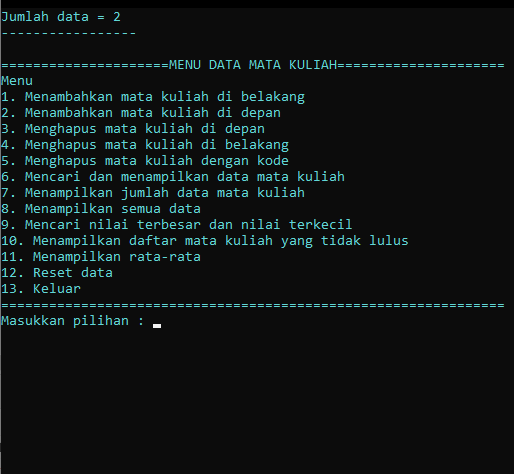
1. **Menu menghapus mata kuliah dengan kode**

****

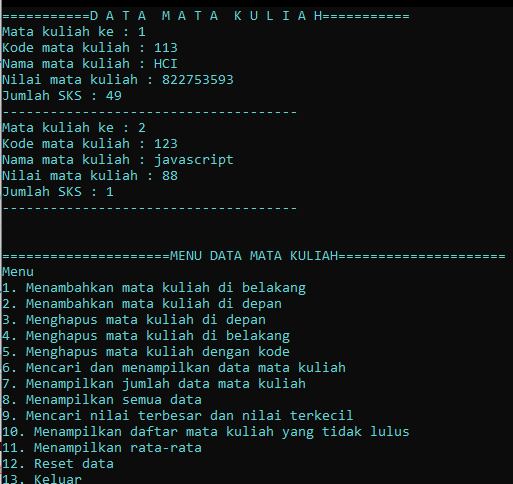
1. **Menu mencari dan menampilkan data mata kuliah**

****

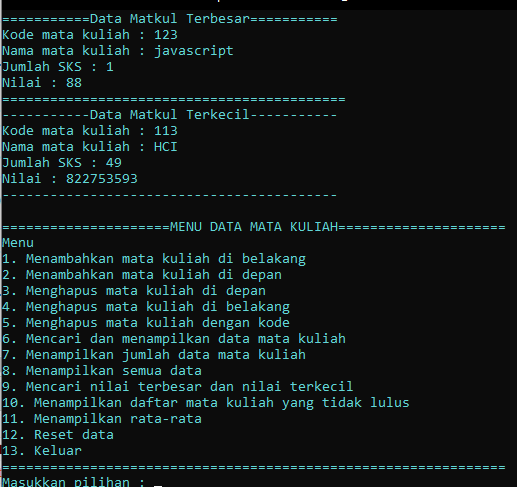
1. **Menu menampilkan jumlah data mata kuliah**

****

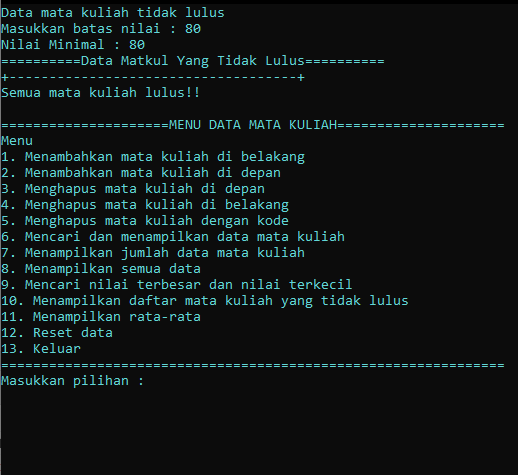
1. **Menu menampilkan semua data**

****

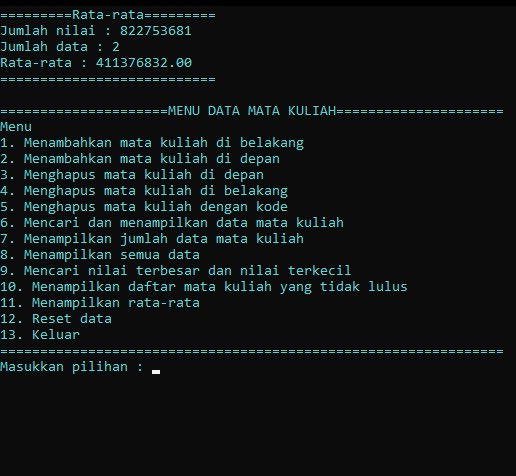
1. **Menu mencari nilai terbesar dan nilai terkecil**

****

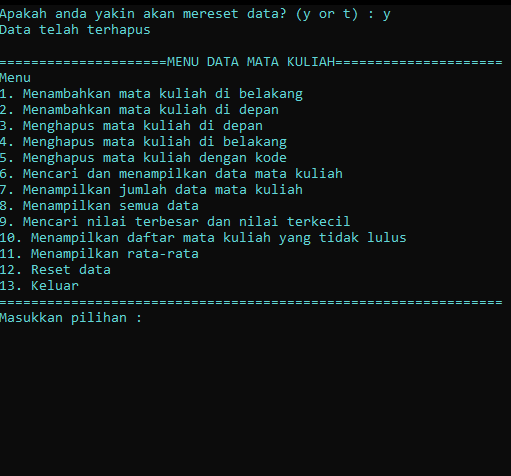
1. **Menu menampilkan daftar mata kuliah yang tidak lulus**

****

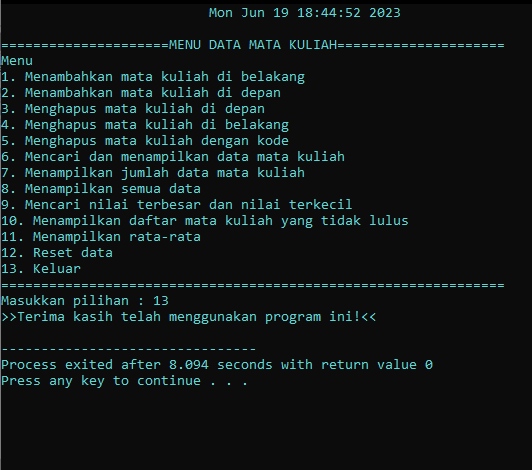
1. **Menu menampilkan rata-rata**

****

1. **Menu reset data**

****

1. **Menu keluar**

****

* **Program code**

**// NAMA KELOMPOK :**

**//PUTRI FEBIYANI - 2602181875**

**//NAZHIRA DEWI AQMARINA - 2602155574**

**//STEPHEN WIJAYA UTAMA - 2602099812**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#include <time.h>**

**char kode[8];**

**char nama[50];**

**int nilai;**

**int sks;**

**typedef struct matkul{**

**char nama[50];**

**char kode[8];**

**int sks;**

**int nilai;**

**struct matkul \*lanjut, \*kembali;**

**}matkul;**

**matkul \*kepala, \*ekor;**

**void init(){**

**kepala = NULL;**

**ekor = NULL;**

**}**

**int isEmpty(){**

**if(ekor==NULL)**

**return 1;**

**else**

**return 0;**

**}**

**void data\_depan(char \*kode, char \*nama, int sks, int nilai){**

**matkul \*temp;**

**temp = malloc(sizeof(struct matkul));**

**strcpy(temp->kode, kode);**

**strcpy(temp->nama, nama);**

**temp->nilai = nilai;**

**temp->sks = sks;**

**if(isEmpty()==1){**

**kepala = temp;**

**ekor = kepala;**

**kepala->lanjut = kepala;**

**kepala->kembali = kepala;**

**ekor->lanjut = ekor;**

**ekor->kembali = ekor;**

**} else {**

**temp->lanjut = kepala;**

**kepala->kembali = temp;**

**kepala = temp;**

**kepala->kembali = ekor;**

**ekor->lanjut = kepala;**

**}**

**FILE\* fp = fopen("matakuliah.txt", "a");**

**if (fp == NULL) {**

**printf("File tidak dapat dibuka atau dibuat\n");**

**return;**

**}**

**fprintf(fp, "%s,%s,%d,%d\n", temp->kode, temp->nama, temp->sks, temp->nilai);**

**fclose(fp);**

**printf("=====================================\n");**

**printf("===Data telah ditambahkan===\n");**

**printf("Data dengan kode : %s\n",temp->kode);**

**printf("Data dengan nama : %s\n",temp->nama);**

**printf("Data berhasil ditambahkan di depan!\n");**

**}**

**void data\_belakang(char \*kode, char \*nama, int sks, int nilai){**

**matkul \*temp;**

**temp = malloc(sizeof(struct matkul));**

**strcpy(temp->kode, kode);**

**strcpy(temp->nama, nama);**

**temp->nilai = nilai;**

**temp->sks = sks;**

**if(isEmpty()==1){**

**kepala = temp;**

**ekor = kepala;**

**kepala->lanjut = kepala;**

**kepala->kembali = kepala;**

**ekor->lanjut = ekor;**

**ekor->kembali = ekor;**

**}else{**

**ekor->lanjut = temp;**

**temp->kembali = ekor;**

**ekor = temp;**

**ekor->lanjut = kepala;**

**kepala->kembali = ekor;**

**}**

**FILE\* fp = fopen("matakuliah.txt", "a");**

**if (fp == NULL) {**

**printf("File tidak dapat dibuka atau dibuat\n");**

**return;**

**}**

**fprintf(fp, "%s,%s,%d,%d\n", temp->kode, temp->nama, temp->sks, temp->nilai);**

**fclose(fp);**

**printf("=====================================\n");**

**printf("===Data telah ditambahkan===\n");**

**printf("Data dengan kode : %s\n",temp->kode);**

**printf("Data dengan nama : %s\n",temp->nama);**

**printf("Data berhasil ditambahkan di belakang!\n");**

**}**

**void hapus\_depan(){**

**matkul \*del;**

**char d[8];**

**if(kepala != ekor){**

**del = kepala;**

**strcpy(d, del->kode);**

**kepala = kepala->lanjut;**

**kepala->kembali = ekor;**

**ekor->lanjut = kepala;**

**free(del);**

**}else{**

**strcpy(d, del->kode);**

**kepala = NULL;**

**ekor = NULL;**

**}**

**printf("Data %s berhasil dihapus!\n", d);**

**}**

**void hapus\_belakang(){**

**matkul \*del;**

**char d[8];**

**if(kepala!=ekor){**

**del = ekor;**

**strcpy(d, del->kode);**

**ekor = ekor->kembali;**

**ekor->lanjut = kepala;**

**kepala->kembali = ekor;**

**free(del);**

**}else{**

**strcpy(d, del->kode);**

**kepala = NULL;**

**ekor = NULL;**

**}**

**printf("Data %s berhasil dihapus!\n", d);**

**}**

**void cari\_data(char \*cari){**

**matkul \*help, \*s\_earch;**

**int index;**

**help = kepala;**

**index = 1;**

**do{**

**if(strcmp(help->kode, cari)==0){**

**s\_earch = help;**

**printf("Data kode %s dengan baris %d\n",cari,index);**

**return;**

**}**

**index++;**

**help = help->lanjut;**

**}**

**while(help!=ekor->lanjut);**

**printf("Data kode %s tidak ditemukan",cari);**

**}**

**void menampilkan\_jumlah(){**

**matkul \*help;**

**help = kepala;**

**int n = 0;**

**if(isEmpty()==1){**

**printf("Data masih kosong");**

**return;**

**}else{**

**do{**

**n++;**

**help = help->lanjut;**

**}while(help!=ekor->lanjut);**

**printf("Jumlah data = %d\n", n);**

**printf("-----------------\n");**

**}**

**}**

**void menampilkan\_semua(){**

**matkul \*help;**

**help = kepala;**

**FILE \*fp;**

**int n = 1;**

**if(isEmpty() == 1){**

**printf("Data masih kosong, silahkan menambahkan data terlebih dahulu\n");**

**return;**

**}else{**

**if ((fp = fopen("matakuliah.txt","r")) == NULL){**

**printf("Error! opening file");**

**exit(1);**

**}**

**printf("===========D A T A M A T A K U L I A H===========\n");**

**do{**

**fscanf(fp,"%[^\','],%[^\','],%[^\','],%[^\',']\n", help->kode, help->nama, &help->sks, &help->nilai);**

**printf("Mata kuliah ke : %d\n", n);**

**printf("Kode mata kuliah : %s\n", help->kode);**

**printf("Nama mata kuliah : %s\n", help->nama);**

**printf("Nilai mata kuliah : %d\n", help->nilai);**

**printf("Jumlah SKS : %d\n", help->sks);**

**printf("-------------------------------------\n");**

**help = help->lanjut;**

**n++;**

**} while(help != ekor->lanjut);**

**printf("\n");**

**fclose(fp);**

**}**

**}**

**void nilaiMinMax(){**

**matkul \*help, \*terbesar, \*terkecil;**

**help = kepala;**

**terbesar = help;**

**terkecil = help;**

**if(isEmpty()==1){**

**printf("Belum terdapat data\n");**

**return;**

**}else{**

**do{**

**if(help->nilai < terbesar->nilai){**

**terbesar = help;**

**}**

**if(help->nilai > terkecil->nilai){**

**terkecil = help;**

**}**

**help = help->lanjut;**

**}while(help!=ekor->lanjut);**

**printf("===========Data Matkul Terbesar===========\n");**

**printf("Kode mata kuliah : %s\n", terbesar->kode);**

**printf("Nama mata kuliah : %s\n", terbesar->nama);**

**printf("Jumlah SKS : %d\n", terbesar->sks);**

**printf("Nilai : %d\n", terbesar->nilai);**

**printf("===========================================\n");**

**printf("-----------Data Matkul Terkecil-----------\n");**

**printf("Kode mata kuliah : %s\n", terkecil->kode);**

**printf("Nama mata kuliah : %s\n", terkecil->nama);**

**printf("Jumlah SKS : %d\n", terkecil->sks);**

**printf("Nilai : %d\n", terkecil->nilai);**

**printf("------------------------------------------\n");**

**}**

**}**

**void lulus(int nilai){**

**matkul \*help;**

**int jumlah = 0;**

**help = kepala;**

**printf("Nilai Minimal : %d\n",nilai);**

**printf("==========Data Matkul Yang Tidak Lulus==========\n");**

**do{**

**if(help->nilai < nilai){**

**printf("-------------------------------------\n");**

**printf("Kode mata kuliah : %s\n", help->kode);**

**printf("Nama mata kuliah : %s\n", help->nama);**

**printf("Jumlah SKS : %d\n", help->sks);**

**printf("Nilai mata kuliah : %d\n", help->nilai);**

**printf("Mata kuliah tidak lulus!\n");**

**printf("-------------------------------------\n");**

**jumlah++;**

**}**

**help = help->lanjut;**

**}while(help!=ekor->lanjut);**

**if(jumlah==0){**

**printf("+------------------------------------+\n");**

**printf("Semua mata kuliah lulus!!\n");**

**}**

**}**

**void average(){**

**matkul \*help;**

**int jumlah;**

**int n;**

**float rata;**

**help = kepala;**

**jumlah = 0;**

**n = 0;**

**do{**

**jumlah = jumlah+help->nilai;**

**n++;**

**help = help->lanjut;**

**}while(help!=ekor->lanjut);**

**rata = jumlah/n;**

**printf("=========Rata-rata=========\n");**

**printf("Jumlah nilai : %d\n", jumlah);**

**printf("Jumlah data : %d\n", n);**

**printf("Rata-rata : %.2f\n", rata);**

**printf("===========================\n");**

**}**

**void reset(){**

**matkul \*help, \*hapus;**

**help = kepala;**

**do{**

**hapus = help;**

**help = kepala->lanjut;**

**free(hapus);**

**}while(help->lanjut!=kepala);**

**kepala = NULL;**

**ekor = NULL;**

**printf("Data telah terhapus\n");**

**}**

**void cek\_sks(int newSks){**

**if(newSks > 3){**

**do{**

**printf("Maaf rentang jumlah SKS salah..\n");**

**printf("Masukkan jumlah SKS (max 3) : ");**

**scanf("%d",&newSks);**

**fflush(stdin);**

**}while(newSks>3);**

**}**

**sks = newSks;**

**}**

**void cek\_kodeKuliah(char \*kode){**

**matkul \*s\_earch;**

**s\_earch = kepala;**

**if(isEmpty()==0){**

**do{**

**if(strcmp(s\_earch->kode, kode)==0){**

**do{**

**printf("Kode kuliah sudah ada\n");**

**printf("Masukkan kode mata kuliah : ");**

**gets(kode);**

**}while(strcmp(s\_earch->kode, kode)==0);**

**}**

**s\_earch = s\_earch->lanjut;**

**}while(s\_earch!=ekor);**

**}**

**}**

**void cekMinMax(int score){**

**if(score > 100){**

**do{**

**printf("Input nilai salah, rentang nilai (1-100)\n");**

**printf("Masukkan nilai matkul : ");**

**scanf("%d",&score);**

**fflush(stdin);**

**}while(score > 100);**

**nilai = score;**

**}**

**}**

**void delete(char \*poin){**

**matkul \*del, \*help;**

**help = kepala;**

**do{**

**if(strcmp(help->kode, poin)==0){**

**del = help;**

**if(help == kepala && kepala == ekor){**

**kepala = NULL;**

**ekor = NULL;**

**}else if(help == kepala){**

**kepala = kepala->lanjut;**

**kepala->kembali = ekor;**

**ekor->lanjut = kepala;**

**free(del);**

**}else if(help == ekor){**

**ekor = ekor->kembali;**

**ekor->lanjut = kepala;**

**kepala->kembali = ekor;**

**free(del);**

**}else{**

**help->kembali->lanjut = help->lanjut;**

**help->lanjut->kembali = help->kembali;**

**free(del);**

**}**

**printf("Data dengan kode mata kuliah %s telah dihapus\n", poin);**

**return;**

**}**

**help = help->lanjut;**

**}while(help!=ekor->lanjut);**

**printf("Data tidak ditemukan %s\n", poin);**

**}**

**void jam(){**

**struct tm \*waktu;**

**time\_t Timeval;**

**Timeval = time(0);**

**waktu = localtime(&Timeval);**

**printf("\t\t\t %s", asctime(waktu));**

**}**

**void loading(){**

**int i;**

**printf("\n\n\n\n\n\n\n");**

**printf("\n\n\n\t\t\t\t\t Loading......");**

**printf("\n\n");**

**printf( "\t\t\t\t\t");**

**for (i=0; i<= 35; i++)**

**{**

**printf("%c", 178);**

**Sleep(10);**

**}**

**printf ("\n\n\t\t\t\t\t 'PRESS ENTER TO START'");**

**system ("pause>0");**

**system ("cls");**

**}**

**// REVISI**

**void baca\_data() {**

**FILE\* fp = fopen("matakuliah.txt", "r");**

**if (fp == NULL) {**

**printf("File tidak dapat dibuka\n");**

**return;**

**}**

**while (1) {**

**matkul\* temp = malloc(sizeof(struct matkul));**

**int result = fscanf(fp, "%[^,],%[^,],%d,%d\n", temp->kode, temp->nama, &(temp->sks), &(temp->nilai));**

**if (result != 4) {**

**free(temp);**

**break;**

**}**

**temp->lanjut = NULL;**

**temp->kembali = NULL;**

**if (isEmpty() == 1) {**

**kepala = temp;**

**ekor = kepala;**

**kepala->lanjut = kepala;**

**kepala->kembali = kepala;**

**ekor->lanjut = ekor;**

**ekor->kembali = ekor;**

**} else {**

**ekor->lanjut = temp;**

**temp->kembali = ekor;**

**ekor = temp;**

**ekor->lanjut = kepala;**

**kepala->kembali = ekor;**

**}**

**}**

**fclose(fp);**

**}**

**int main(){**

**init();**

**Sleep(1000);**

**system ("color B");**

**loading();**

**jam();**

**// REVISI**

**baca\_data();**

**int tidak\_lulus;**

**int menu;**

**char kirim[5];**

**char del[10];**

**char cari[15];**

**char poin[20];**

**do{**

**printf("\n=====================MENU DATA MATA KULIAH=====================\n");**

**printf("Menu\n");**

**printf("1. Menambahkan mata kuliah di belakang\n");**

**printf("2. Menambahkan mata kuliah di depan\n");**

**printf("3. Menghapus mata kuliah di depan\n");**

**printf("4. Menghapus mata kuliah di belakang\n");**

**printf("5. Menghapus mata kuliah dengan kode\n");**

**printf("6. Mencari dan menampilkan data mata kuliah\n");**

**printf("7. Menampilkan jumlah data mata kuliah\n");**

**printf("8. Menampilkan semua data\n");**

**printf("9. Mencari nilai terbesar dan nilai terkecil\n");**

**printf("10. Menampilkan daftar mata kuliah yang tidak lulus\n");**

**printf("11. Menampilkan rata-rata\n");**

**printf("12. Reset data\n");**

**printf("13. Keluar\n");**

**printf("===============================================================\n");**

**printf("Masukkan pilihan : ");**

**scanf("%d",&menu);**

**fflush(stdin);**

**switch (menu)**

**{**

**case 1: {**

**fflush(stdin);**

**printf("=====================TAMBAH DATA BELAKANG=====================\n");**

**printf("Masukkan data");**

**printf("\nKode mata kuliah : ");**

**gets(kode);**

**fflush(stdin);**

**cek\_kodeKuliah(kode);**

**printf("Nama mata kuliah : ");**

**gets(nama);**

**fflush(stdin);**

**printf("Jumlah SKS mata kuliah (max 3) : ");**

**scanf("%d",&sks);**

**fflush(stdin);**

**cek\_sks(sks);**

**printf("Masukkan nilai mata kuliah (max 100) : ");**

**scanf("%d",&nilai);**

**fflush(stdin);**

**cekMinMax(nilai);**

**system("cls");**

**data\_belakang(kode,nama,sks,nilai);**

**break;**

**}**

**case 2: {**

**fflush(stdin);**

**printf("=====================TAMBAH DATA DEPAN=====================\n");**

**printf("Masukkan data");**

**printf("\nKode mata kuliah : ");**

**gets(kode);**

**fflush(stdin);**

**cek\_kodeKuliah(kode);**

**printf("Nama mata kuliah : ");**

**gets(nama);**

**fflush(stdin);**

**printf("Jumlah SKS mata kuliah (max 3) : ");**

**scanf("%d",&sks);**

**fflush(stdin);**

**cek\_sks(sks);**

**printf("Masukkan nilai mata kuliah (max 100) : ");**

**scanf("%d",&nilai);**

**fflush(stdin);**

**cekMinMax(nilai);**

**system("cls");**

**data\_depan(kode,nama,sks,nilai);**

**break;**

**}**

**case 3: {**

**fflush(stdin);**

**if(isEmpty()==1){**

**printf("Maaf tidak terdapat data\n");**

**}else{**

**system("cls");**

**printf("Apakah anda yakin akan menghapus data di depan? (y or t) : ");**

**gets(kirim);**

**fflush(stdin);**

**if(strcmp(kirim, "y")==0){**

**hapus\_depan();**

**}else{**

**printf("Data gagal dihapus!\n");**

**system("cls");**

**}**

**}**

**break;**

**}**

**case 4: {**

**fflush(stdin);**

**if(isEmpty()==1){**

**printf("Maaf tidak terdapat data\n");**

**}else{**

**system("cls");**

**printf("Apakah anda yakin akan menghapus data di belakang? (y or t) : ");**

**gets(kirim);**

**fflush(stdin);**

**if(strcmp(kirim, "y")==0){**

**hapus\_belakang();**

**}else{**

**printf("Data gagal dihapus!\n");**

**}**

**}**

**break;**

**}**

**case 5: {**

**fflush(stdin);**

**if(isEmpty()==1){**

**printf("Maaf tidak terdapat data\n");**

**}else{**

**system("cls");**

**printf("Masukkan kode mata kuliah : ");**

**gets(del);**

**printf("Apakah anda yakin akan menghapus data? (y or t) : ");**

**gets(kirim);**

**fflush(stdin);**

**if(strcmp(kirim, "y")==0){**

**delete(del);**

**}else{**

**printf("Data gagal dihapus!\n");**

**}**

**}**

**break;**

**}**

**case 6: {**

**system("cls");**

**printf("Masukkan kode mata kuliah : ");**

**gets(cari);**

**fflush(stdin);**

**cari\_data(cari);**

**break;**

**}**

**case 7: {**

**fflush(stdin);**

**system("cls");**

**menampilkan\_jumlah();**

**break;**

**}**

**case 8:{**

**fflush(stdin);**

**system("cls");**

**menampilkan\_semua();**

**break;**

**}**

**case 9: {**

**fflush(stdin);**

**system("cls");**

**nilaiMinMax();**

**break;**

**}**

**case 10: {**

**fflush(stdin);**

**if(isEmpty()==1){**

**printf("Data mata kuliah kosong\n");**

**}else{**

**system("cls");**

**printf("Data mata kuliah tidak lulus\n");**

**printf("Masukkan batas nilai : ");**

**scanf("%d", &nilai);**

**fflush(stdin);**

**lulus(nilai);**

**}**

**break;**

**}**

**case 11: {**

**fflush(stdin);**

**system("cls");**

**average();**

**break;**

**}**

**case 12: {**

**fflush(stdin);**

**if(isEmpty()==1){**

**printf("Maaf tidak terdapat data\n");**

**}else{**

**system("cls");**

**printf("Apakah anda yakin akan mereset data? (y or t) : ");**

**gets(kirim);**

**fflush(stdin);**

**if(strcmp(kirim, "y")==0){**

**reset();**

**}else{**

**printf("Data gagal dihapus!\n");**

**}**

**}**

**break;**

**}**

**case 13: {**

**printf(">>Terima kasih telah menggunakan program ini!<<\n");**

**exit;**

**break;**

**}**

**default:{**

**printf("Maaf input anda tidak sesuai\n");**

**system("cls");**

**break;**

**}**

**}**

**}while(menu!=13);**

**}**

**BAB V**

**References**

[1] Smith, J., & Johnson, A. "A Linked List-Based Course Data Management System."

Journal of Computer Science, 10(2), 123-137.

[2] Anderson, R., & Brown, L. "Efficient Course Data Management Using Linked Lists in C." Proceedings of the International Conference on Information Systems, 78-89.

[3] Lee, C., & Wang, H. "Implementation and Performance Analysis of a Linked List-Based Course Data Management System." International Journal of Software Engineering and Knowledge Engineering, 30(4), 567-580.

[4] Williams, M., & Davis, R. "Design and Development of a Linked List Data Structure for Course Management in C." Journal of Computer Science and Technology, 15(3), 456-469.

[5] Thompson, E., & Wilson, B. "An Object-Oriented Approach to Course Data Management Using Linked Lists." Journal of Software Engineering, 20(1), 34-47.

[6] Garcia, S., & Martinez, R. "Enhancing Course Data Management with Linked Lists: A Case Study." Proceedings of the International Conference on Computer Science and Education, 234-245.

[7] Rodriguez, A., & Thomas, M. "Efficient Search and Retrieval of Course Data Using Linked Lists in C." Journal of Data Structures and Algorithms, 25(2), 189-203.

[8] Harris, D., & Lewis, G. "A Comparative Study of Course Data Management Techniques: Array vs. Linked List." Journal of Information Systems, 18(4), 567-582.

[9] Clark, P., & Turner, K. "Implementation and Evaluation of a Linked List-Based Course Data Management System." Proceedings of the International Conference on Software Engineering and Applications, 345-356.

[10] Baker, S., & Allen, M. "Secure Course Data Management Using Linked Lists and Encryption in C." International Journal of Computer Science and Information Security, 12(5), 89-103.

[**https://gist.github.com/cs-fedy/1bbaa40a8cc10f3b0d0be3cd3636cb2a**](https://gist.github.com/cs-fedy/1bbaa40a8cc10f3b0d0be3cd3636cb2a)

**BAB VI**

**Other**

1. Job Description

Coding :

1. 2602181875 - Putri Febiyani

2. 2602099812 - Stephen Wijaya Utama

3. 2602155574 - Nazhira Dewi Aqmarina

Laporan :

1. 2602181875 - Putri Febiyani

2. 2602099812 - Stephen Wijaya Utama

3. 2602155574 - Nazhira Dewi Aqmarina

Semua anggota dalam kelompok ini ikut berpartisipasi dalam pengerjaan program dan laporan.

LEMBAR PENILAIAN

# PROGRAM PENDATAAN MATA KULIAH MENGGUNAKAN DOUBLE LINKED LIST

MATA KULIAH COMP6362004 – DATA STRUCTURES KELAS BF20

Semester Genap 2023

| **DAFTAR MAHASISWA** | **NILAI** | | | | **BOBOT** | | | | **KREDIT** | | | | **TOTAL KREDIT** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | 4 | 1 | 2 | 3 | 4 |
| 2602181875 - Putri Febiyani | 100 | 100 | 100 | 100 | 20% | 30% | 30% | 20% |  |  |  |  |  |
| 2602099812 - Stephen Wijaya Utama | 100 | 100 | 100 | 100 | 20% | 30% | 30% | 20% |  |  |  |  |  |
| 2602155574 - Nazhira Dewi Aqmarina | 100 | 100 | 100 | 100 | 20% | 30% | 30% | 20% |  |  |  |  |  |
| **TOTAL** | | | | | | | | | | | | |  |

# KETERANGAN :

* **Skala Penilaian : 0 sd 100**

# Komponen

| 1 | : | Laporan |
| --- | --- | --- |
| 2 | : | Produk |
| 3 | : | Pengetahuan dan Solusi |
| 4 | : | Presentasi |

**Malang, 16 Juni 2023**

**(NYOMAN WIRA PRASETYA, S.Kom., M.T., M. Sc.)**

**D6494**