

**LAPORAN PROYEK**  
**ANTRIAN PENGAJUAN E-KTP**



Disusun Oleh :

Fadhel Pramono Gartam (NPM 1606951254)

Michael Bonardo Setiawira (NPM 1506672666)

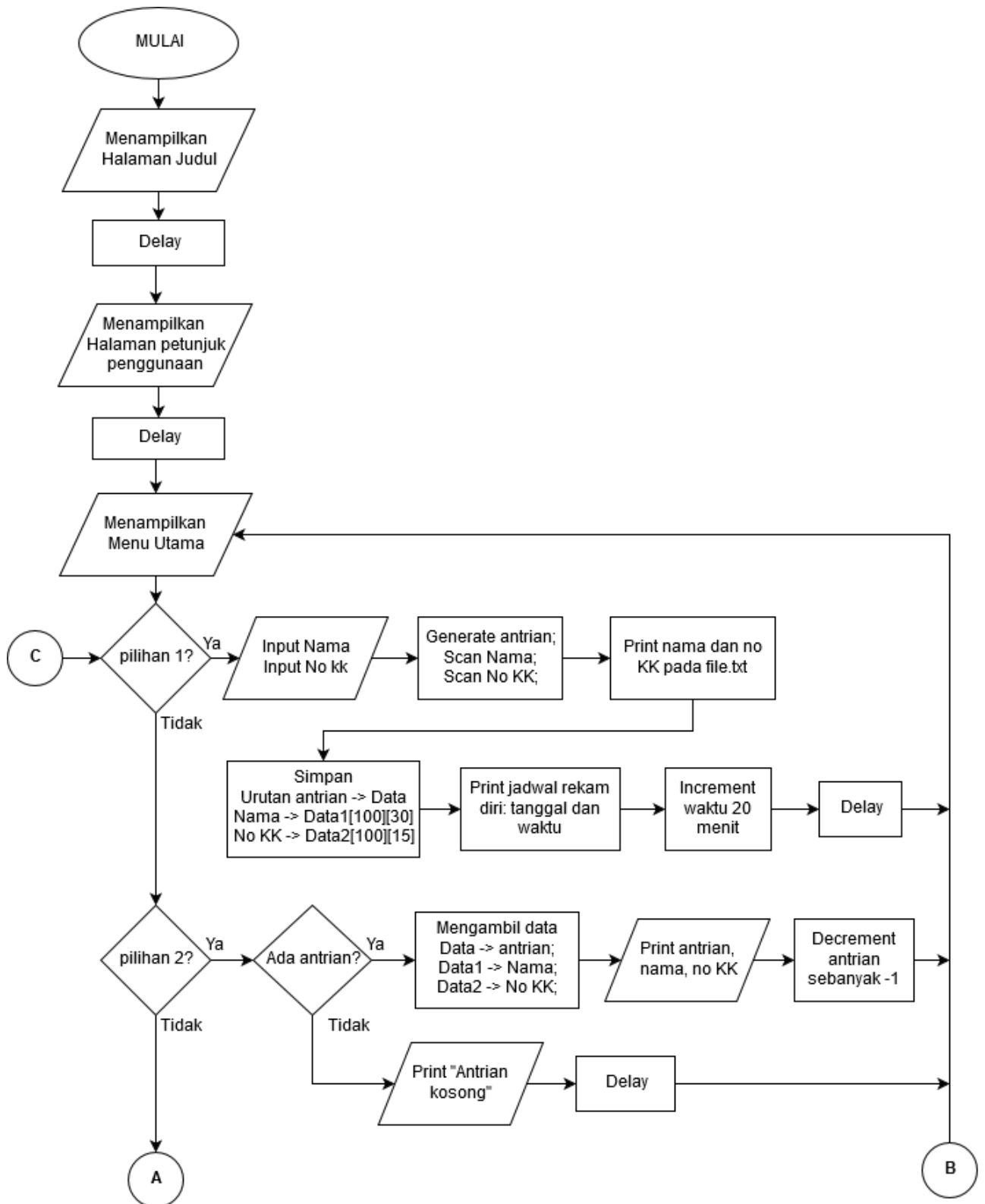
**DEPARTEMEN TEKNIK ELEKTRO**  
**FAKULTAS TEKNIK UNIVERSITAS INDONESIA**  
**DEPOK**  
**2018**

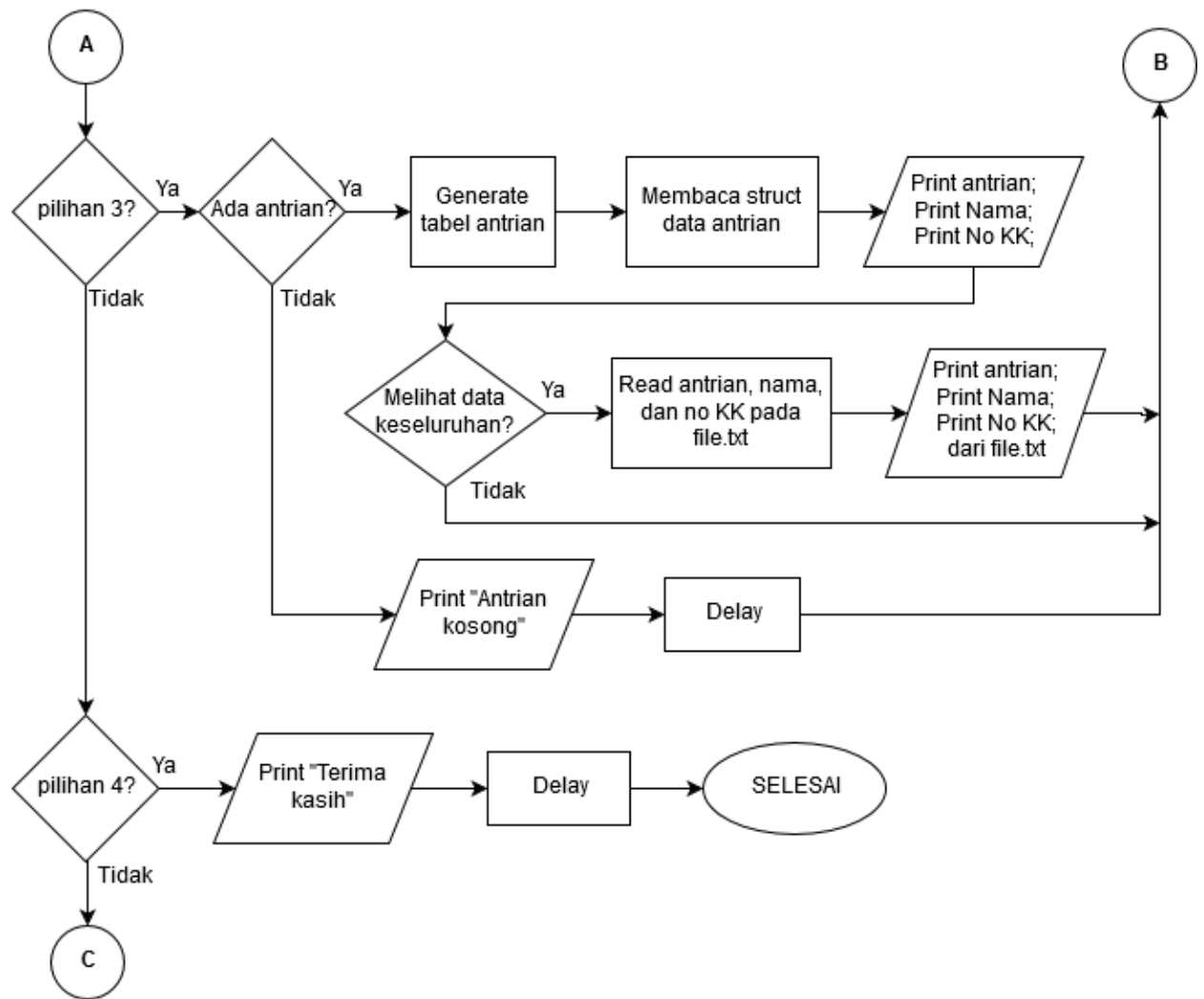
## 1. Pemaparan Fitur Program

Program ini merupakan program antrian pengajuan E-KTP yang memfasilitasi masyarakat yang akan mengajukan pembuatan E-KTP, serta untuk mengetahui jadwal giliran rekam data diri di Kantor Kecamatan setempat (*enqueue*). Jadwal rekam data diri yang dimaksud berupa hari, tanggal, dan jam. Hari dan tanggal perekaman data diri awal dimulai pada 1 Januari 2018, sedangkan jam antrian pertama setiap harinya dimulai dari pukul 09.00 sampai pukul 15.00. Pada pukul 12.00 sampai dengan pukul 13.00 merupakan jam istirahat pegawai kecamatan, sehingga tidak ada jadwal rekam data diri warga pada waktu tersebut. Durasi proses rekam data diri warga adalah masing-masing 20 menit. Program difasilitasi untuk memanggil antrian secara berurutan (*dequeue*) untuk melakukan rekam data diri. Program ini juga dapat menyimpan data keseluruhan dari pengaju beserta nomor antrian pengajuan E-KTP dengan ekstensi “Pengajuan.txt”.

Pada program ini terdapat tantangan yang diberikan, yaitu untuk memasukan data pengaju E-KTP, namun data tersebut tidak dimasukan pada program saat berjalan melainkan pada file penyimpanan data dengan ekstensi .txt. Tantangan ini berhasil diselesaikan. Terdapat penambahan menu tambahan pada saat menampilkan data pengaju E-KTP pada program, yaitu untuk menampilkan histori antrian sehingga seluruh data antrian baik yang sudah terpanggil (*dequeue*) ataupun data yang dimasukan pada file penyimpanan dengan ekstensi .txt akan ditampilkan pada program ini.

## 2. Diagram Alir Program





### 3. Pemaparan Aspek Teknis Pada Pemograman

- Implementasi pemrograman modular

Modularisasi diimplementasikan pada pemograman. Fungsi modularisasi adalah untuk menyederhakan proses eksekusi dari sekuensial instruksi pada koding yang telah

```
int antrianKosong(){
    if (antrian.belakang==1)
        return 1;
    else
        return 0;}

int antrianTerisi(){
    if(antrian.belakang==100)
        return 1;
    else
        return 0;}

int antrianKosong(){
    if (antrian.belakang==1)
        return 1;
    else
        return 0;}

int antrianTerisi(){
    if(antrian.belakang==100)
        return 1;
    else
        return 0;}
```

dibuat. Pada proyek ini terdapat beberapa mododularisasi program, diantaranya :

```
void Enqueue(int data, char data1[20], char data2[30]){
    if(antrianKosong()==1){
        antrian.depan=antrian.belakang=0;
        nama.depan=nama.belakang=0;
        nokk.depan=nokk.belakang=0;
        antrian.data[antrian.belakang]=data;
        for(i=0;i<20;i++){
            nama.data1[nama.belakang][i]=data1[i];
        }
        for(i=0;i<15;i++){
            nokk.data2[nokk.belakang][i]=data2[i];
        }
        printf("\n\t\t\t\t\t =====DATA TERSIMPAN===== ");
        jadwal(data);delay(3000);}
    else if(antrianTerisi()==0){
        antrian.belakang++;
        nama.belakang++;
        nokk.belakang++;
        antrian.data[antrian.belakang]=data;
        for(i=0;i<20;i++){
            nama.data1[nama.belakang][i]=data1[i];
        }
        for(i=0;i<15;i++){
            nokk.data2[nokk.belakang][i]=data2[i];
        }
    }
}

int Dequeue(){
    if(antrianKosong()==0){
        int i,e ;char a[20], b[30];
        e=antrian.data[antrian.depan];
        for(i=0;i<20;i++){
            a[i]=nama.data1[nama.depan][i];
        }
        for(i=antrian.depan;i<antrian.belakang;i++){
            antrian.data[i]=antrian.data[i+1];
            for(j=0;j<20;j++){
                nama.data1[i][j]=nama.data1[i+1][j];
            }
            for(j=0;j<15;j++){
                nokk.data2[i][j]=nokk.data2[i+1][j];
            }
        }
        antrian.belakang--;
        nama.belakang--;
        nokk.belakang--;
        kop();
        printf("\n\t\t\t\t\tAntrian %i dengan nama %s\n", e, a);
        printf("\n\t\t\t\t\tSilahkan masuk counter rekam data!\n\n\n");delay(3000);
        return e;
        return a[20];
    }
    else{
        kop();
        printf("\n\n\n\n\n\n\n\t\t\t\t\t =====ANTRIAN KOSONG===== ");
    }
}
```

- Implementasi header

Header digunakan untuk menyederhanakan program dengan menjadikan beberapa instruksi modular menjadi dalam suatu header. Header juga dapat menjadi suatu library yang suatu waktu dapat digunakan. Adapun header yang digunakan adalah sebagai berikut,

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <time.h>
#include "subroutines.h"
```

- Implementasi array

Pengimplementasian array digunakan untuk menyimpan nomor antrian, nama lengkap dan nomor kartu keluarga. Pada array nama lengkap dan nomor kartu keluarga digunakan array 2 dimensi, sedangkan untuk menyimpan nomor antrian digunakan array satu dimensi. Berikut penggunaan array pada program ini,

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <time.h>
#include "subroutines.h"

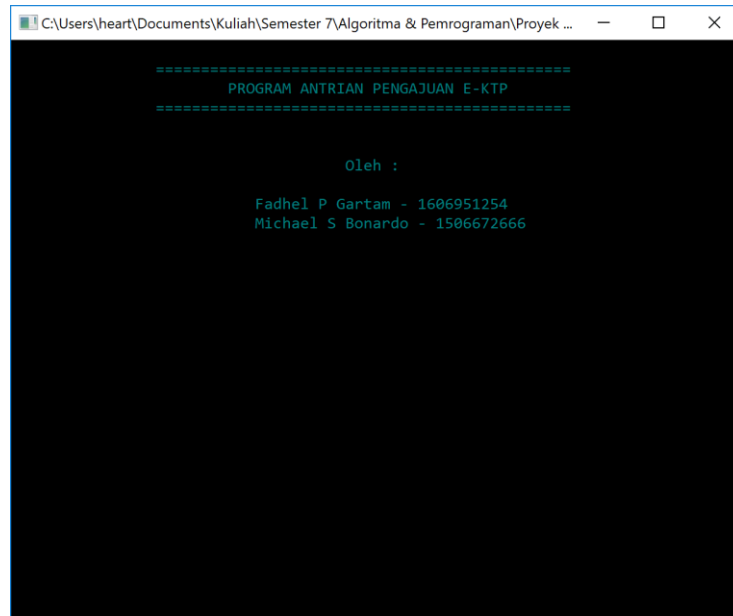
typedef struct{
    int data[100];char data1[100][20];
    char data2[100][30];char data3[100][15];
    int depan;int belakang;}
Queue;Queue antrian;Queue nokk;Queue nama;

int pilihan, data, i, j,n;
char data1[20], data2[30], data3[15];
```

- Implementasi user friendly

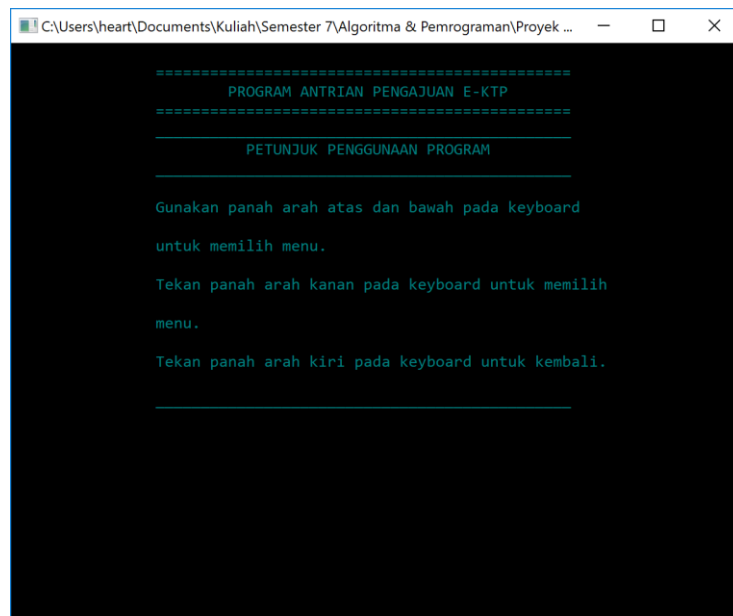
Pengimplementasian unsur user friendly adalah dengan adanya judul program, petunjuk penggunaan program, notifikasi error, penggunaan program sesuai dengan

pengguna inginkan. Adapun masing-masing bagian dari unsur user friendly yang diimplemetasikan pada program adalah sebagai berikut,



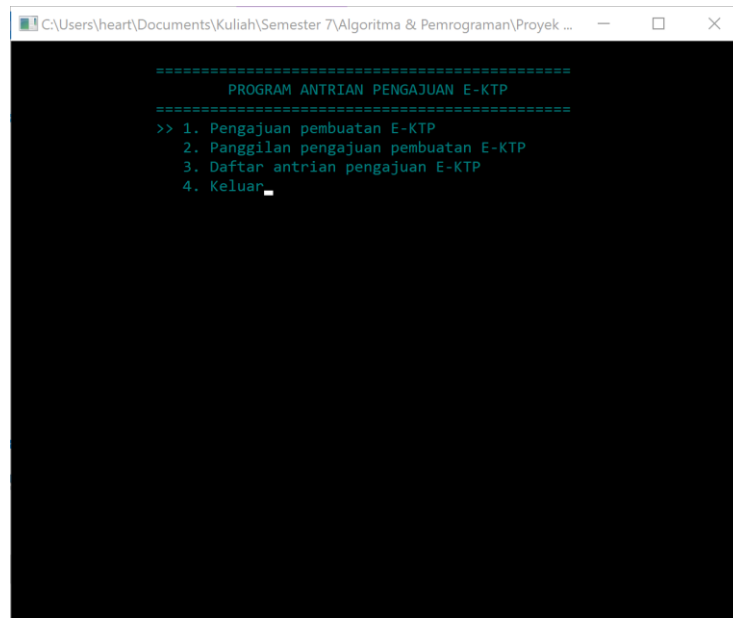
```
C:\Users\heart\Documents\Kuliah\Semester 7\Algoritma & Pemrograman\Proyek ...  
=====   
PROGRAM ANTRIAN PENGAJUAN E-KTP  
=====   
  
Oleh :  
  
Fadhel P Gartam - 1606951254  
Michael S Bonardo - 1506672666
```

**Gambar 01.** Judul dan identitas pembuat



```
C:\Users\heart\Documents\Kuliah\Semester 7\Algoritma & Pemrograman\Proyek ...  
=====   
PROGRAM ANTRIAN PENGAJUAN E-KTP  
=====   
PETUNJUK PENGGUNAAN PROGRAM  
=====   
Gunakan panah arah atas dan bawah pada keyboard  
untuk memilih menu.  
  
Tekan panah arah kanan pada keyboard untuk memilih  
menu.  
  
Tekan panah arah kiri pada keyboard untuk kembali.  
=====
```

**Gambar 02.** Petunjuk penggunaan program



**Gambar 03.** Menu Utama

- Implementasi struktur data

Struktur data merupakan suatu pengelompokan data yang dapat dimanfaatkan untuk fungsi tertentu. Pengimplementasian struktur data pada program adalah sebagai berikut,

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <time.h>
#include "subroutines.h"

typedef struct{
    int data[100];char data1[100][20];
    char data2[100][30];char data3[100][15];
    int depan;int belakang;}
Queue;Queue antrian;Queue nokk;Queue nama;

int pilihan, data, i, j,n;
char data1[20], data2[30], data3[15];

```

- Implementasi pointer

pointer digunakan untuk memudahkan pengeksekusian instruksi ddengan melibatkan alamat referensi untuk menghemat memori. Pengaplikasian pointer pada penginstruksian menulis dan membaca data dari file.txt. adapun format penulisan pointer adalah sebagai berikut,



```

void savefile(int antrian, char nama[20], char nokk[30]){
    FILE *out=fopen("Pengajuan.txt", "a");
    fprintf(out, "\n%d\t\t%s\t\t%s\n", antrian, nama, nokk);
    fclose(out);
}

```

## 4. Aspek Penilaian

### 3.1 Executable Program

- Kompleksitas Program

Program memfasilitasi pengguna untuk mendapat nomor antrian(enqueue) dan jadwal (Hari, tanggal, dan waktu) perekaman data diri dengan asumsi setiap orang menghabiskan waktu 20menit untuk rekam data diri, serta asumsi proses rekam data diri dimulai pada tanggal 1 Januari 2018. Program juga memfasilitasi untuk memanggil antrian(dequeue). Program ini dilengkapi fasilitas untuk menyimpan histori data antrian dan dapat disimpan pada file .txt, serta dapat menginput data dari file tersebut dan dapat terbaca oleh program. Adapun program yang dipaparkan diatas dapat dilihat pada gambar berikut,

```

105 case 1 :
106     system("CLS");kop();
107     printf("\n\t\tAntrian      : %i\n", a);data=a; //Generate nomor antrian
108     printf("\n\t\tNama Lengkap   : "); scanf("%[^\n]",&data1);fflush(stdin);
109     printf("\n\t\tNomor Kartu Keluarga : "); scanf("%s",&data2);fflush(stdin);
110     savefile(data,data1,data2);
111     Enqueue(data, data1, data2);a++;break; //Nomor antrian bertambah dengan a++

```

**Gambar 04.** Program membaca data pengaju E-KTP dan generate antrian

```

28 void Enqueue(int data, char data1[20], char data2[30]){
29     if(antrianKosong()==1){
30         antrian.depan=antrian.belakang=0;
31         nama.depan=nama.belakang=0;
32         nokk.depan=nokk.belakang=0;
33         antrian.data[antrian.belakang]=data;
34         for(i=0;i<20;i++){
35             nama.data1[nama.belakang][i]=data1[i];
36         }
37         for(i=0;i<15;i++){
38             nokk.data2[nokk.belakang][i]=data2[i];
39             printf("\n\t\t\t\t\t=====DATA TERSIMPAN===== ");
40             jadwal(data);delay(3000);}
41     else if(antrianTerisi()==0){
42         antrian.belakang++;
43         nama.belakang++;
44         nokk.belakang++;
45         antrian.data[antrian.belakang]=data;
46         for(i=0;i<20;i++){
47             nama.data1[nama.belakang][i]=data1[i];
48         }
49         for(i=0;i<15;i++){
50             nokk.data2[nokk.belakang][i]=data2[i];
51             printf("\n\t\t\t\t\t=====DATA TERSIMPAN===== ");
52             jadwal(data);delay(3000);}

```

**Gambar 05.** Program membentuk antrian pengajuan E-KTP

```

46 int jadwal(int n){
47     #include<stdio.h>
48
49     int x;int menit=0;int jam=9;int day=1; //nilai awal tanggal dan bulan perekaman data diri
50     int tanggal=1;int minggu=1;int bulan=1;
51     int tahun=2018;
52     void hari(int x);void month(int x);
53     for( x=1;x<n;x++){
54         menit=menit+20;
55         if(menit>=60){
56             jam=jam+1;menit=0;}
57         if(jam>=15){
58             day=day+1;tanggal=tanggal+1;jam=9;}
59         if (day>5){
60             minggu=minggu+1;day=1;}
61         if (tanggal>=31){
62             bulan=bulan+1;tanggal=1;} //Asumsi jumlah 31 hari tiap bulannya
63         if (bulan==2){
64             if (tanggal>=28){
65                 bulan=bulan+1;tanggal=1;}} //bukan tahun kabisat
66         if (bulan>=12){
67             tahun=tahun+1;bulan=1;}
68     }
69     printf("\n\t\t\t\t\t=====");
70     printf("\n\t\t\tJadwal rekam identitas diri anda,\n");
71     printf("\n\t\t\tHari,Tanggal : ");
72     hari(day);printf(" %d ",tanggal);month(bulan);printf(" %d\n",tahun);
73     printf("\n\t\t\tWaktu      : Pukul %d.%d WIB\n",jam,menit);
74 }

```

**Gambar 06.** Program generate jadwal rekam data diri(hari/tanggal/waktu)

```

52 int Dequeue(){
53     if(antrianKosong()==0){
54         int i,e ;char a[20], b[30];
55         e=antrian.data[antrian.depan];
56         for(i=0;i<20;i++){
57             a[i]=nama.data1[nama.depan][i];
58
59             for(i=antrian.depan;i<antrian.belakang;i++){
60                 antrian.data[i]=antrian.data[i+1];
61                 for(j=0;j<20;j++){
62                     nama.data1[i][j]=nama.data1[i+1][j];
63                     for(j=0;j<15;j++){
64                         nokk.data2[i][j]=nokk.data2[i+1][j];
65                     }
66                 }
67                 antrian.belakang--;
68                 nama.belakang--;
69                 nokk.belakang--;
70                 kop();
71                 printf("\n\t\tAntrian %i dengan nama %s\n", e, a);
72                 printf("\n\t\tSilahkan masuk counter rekam data!\n\n");delay(3000);
73                 return e;
74                 return a[20];
75     }
76     else{
77         kop();
78         printf("\n\n\n\n\n\n\t\t =====ANTRIAN KOSONG===== ");
79     }
80 }

```

**Gambar 07.** Program memanggil antrian ke konter rekam data

```

129 void savefile(int antrian,char nama[20],char nokk[30]){
130     FILE *out=fopen("Pengajuan.txt","a");
131     fprintf(out,"\n%d\t\t%s\t\t%s\n",antrian,nama,nokk);
132     fclose(out);
133 }
134
135 void menureadfile(int pilihan){
136     char arah1[3]={' ',' ',' '};
137     arah1[pilihan]='>';
138     kop();
139     printf("\n\t\tLihat keseluruhan data pendaftar?");
140     printf("\n\t\t%c 1. Buka",arah1[1],arah1[1]);
141     printf("\n\t\t%c 2. Kembali",arah1[2],arah1[2]);
142 }

```

**Gambar 08.** Program menyimpan data antrian ke file .txt dan membaca input data dari file untuk ditampilkan pada program.

```
C:\Users\heart\Documents\Kuliah\Semester 7\Algoritma & Pemrograman\Proyek ...  
  
=====PROGRAM ANTRIAN PENGAJUAN E-KTP=====  
  
Antrian : 1  
Nama Lengkap : Fadhel  
Nomor Kartu Keluarga : 12938172  
  
=====DATA TERSIMPAN=====  
Jadwal rekam identitas diri anda,  
  
Hari,Tanggal : Senin, 1 Januari 2018  
Waktu : Pukul 9.0 WIB
```

**Gambar 09.** Tampilan isian data pengaju E-KTP beserta generate jadwal rekam data diri.

- Terdapat Judul Program

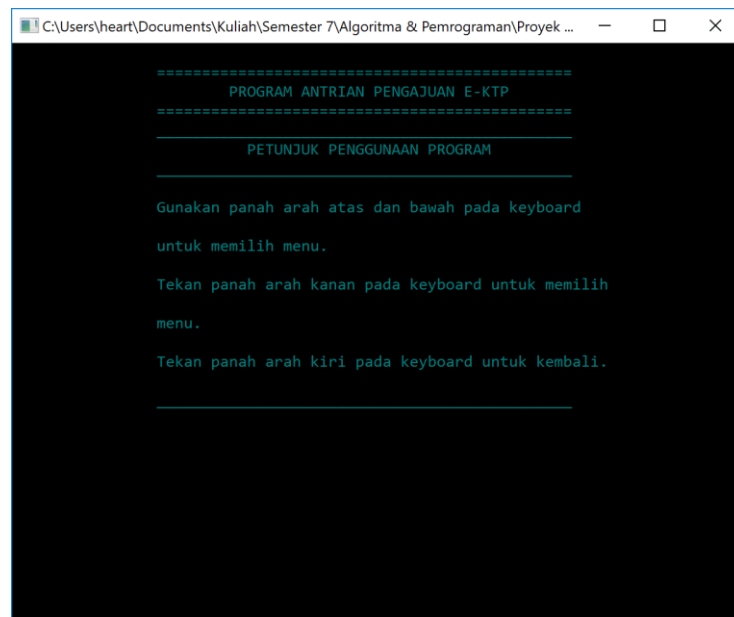
Halaman judul pada program memuat judul program dan identitas pembuat dengan warna dan ukuran window yang telah ditentukan oleh pembuat sebagai pertimbangan estetika. Halaman judul dirancang hanya ditampilkan selama 2 detik dengan menggunakan instruksi delay.

```
C:\Users\heart\Documents\Kuliah\Semester 7\Algoritma & Pemrograman\Proyek ...  
  
=====PROGRAM ANTRIAN PENGAJUAN E-KTP=====  
  
Oleh :  
Fadhel P Gartam - 1606951254  
Michael S Bonardo - 1506672666
```

**Gambar 10.** Tampilan Identitas Pembuat Program

- Terdapat fasilitas HELP/Petunjuk pengoperasian

Halaman petunjuk pengoperasian program muncul setelah halaman identitas yang ditampilkan selama 5 detik agar pengguna dapat membaca cara menggunakan program. Pada halaman petunjuk ini memuat pemaparan bagaimana cara memilih opsi pada menu utama, cara menyimpan data input dan cara untuk kembali atau keluar program. Adapun halaman petunjuk yang dipaparkan diatas adalah sebagai berikut,



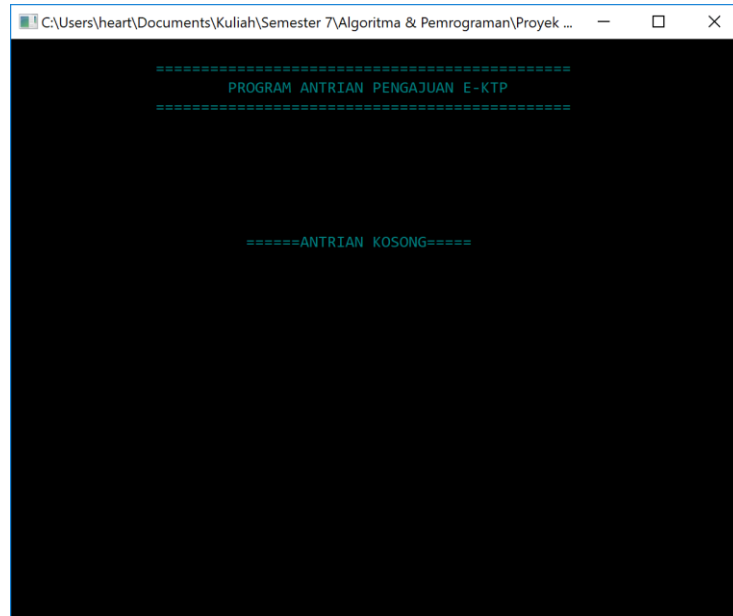
**Gambar 011.** Halaman petunjuk penggunaan program

- Program dapat dijalankan berulang-ulang

Program ini dapat digunakan secara berulang, bahkan untuk berulang memasukan data pengajuan E-KTP sebanyak 100 data. Jumlah 100 data diatur oleh jumlah elemen array 2 dimensi yang dideklarasikan pada program. Bukti bahwa program dapat dilakukan secara berulang adalah adanya menu kembali pada setiap halaman dengan menekan opsi kembali atau tekan tombol panah arah kiri pada keyboard.

- Terdapat pesan kesalahan

Penggunaan pemilihan opsi dengan kursor menyebabkan pengguna tertabulasi pada pilihan yang ada sehingga tidak diperlukan pesan kesalahan saat memilih menu, namun pada opsi melihat daftar antrian bila tidak ada antrian maka akan ada notifikasi antrian kosong sebagai pesan kesalahan. Berikut merupakan respon pesan kesalahan pada program ini,



**Gambar 012.** Halaman pesan kesalahan bila tidak diinput data

### 3.2 Source Code

- Modularity program dan program reusable

Terdapat cukup banyak sekali program modular yang berfungsi untuk menyederhanakan program dan memudahkan pemahaman program keseluruhan. Program modular dapat dilihat pada sumber kode keseluruhan.

- Menggunakan minimal 1 buah modul dengan parameter passing

Penggunaan parameter passing terlihat pada koding program pada saat membaca seluruh data pada file penyimpanan dengan ekstensi .txt. Adapun koding tersebut adalah sebagai berikut,

```

125 //buka data keseluruhan sumber dari file
126 int tombol=1,tombolmax=2;
127 unsigned char ch;FILE *f;char s[5000];char c[1000];
128 while(tombol!=1){
129     if(tombol>tombolmax){
130         tombol=1;
131         //nilai but berputar dari 1 ke 2 dan kembali lagi ke 1
132     } else if(tombol<1){
133         tombol=tombolmax;
134     }
135     system("cls");menureadfile(tombol);ch=getch();
136     if (ch==0){
137         //initial read input char dari keyboard
138         ch=getch();
139     }
140     if (ch==72){
141         //decrement nilai button bila ditekan panah atas
142         tombol--;
143     }
144     else if (ch==80){
145         //increment nilai button bila ditekan panah bawah
146         tombol++;
147     }
148     if(ch==77){
149         //memilih opsi bila ditekan panah kanan
150         system("CLS");
151         switch (tombol){
152             case 1: kop();printf("\nAntrian\tNama\t\tNomor KK\n");
153                 f=fopen("Pengajuan.txt", "r");
154                 if(!f)
155                     return 1;
156                 while (fgets(s,5000,f)!=NULL) {
157                     printf("%s", s);
158                 }
159                 fclose(f);delay(2000);
160                 break;

```

**Gambar 013.** Penggunaan parameter passing

- Menggunakan ARRAY

Pada program utama menggunakan array satu dimensi dan dua dimensi untuk menyimpan data nama, nomor kartu keluarga, nomor antrian. Array tersebut dideklarasikan pada inisialisasi program. Adapun array yang dimaksud adalah sebagai berikut,

```
1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  #include <time.h>
5  #include "subroutines.h"
6
7  typedef struct{
8      int data[100];char data1[100][20];
9      char data2[100][30];char data3[100][15];
10     int depan,int belakang;}
11 Queue;Queue antrian;Queue nokk;Queue nama;
```

**Gambar 014.** Penggunaan array pada program

- Menggunakan POINTER yang sesuai dengan kebutuhan

Pointer digunakan untuk merujuk instruksi membuka file dengan format .txt. Berikut merupakan implementasi pointer pada program,

```
129 void savefile(int antrian,char nama[20],char nokk[30]){
130     FILE *out=fopen("Pengajuan.txt","a");
131     fprintf(out,"%d\t\t%s\t\t%s\n",antrian,nama,nokk);
132     fclose(out);
133 }
```

**Gambar 015.** Penggunaan pointer pada program

- Data dibuat dalam bentuk STRUCT

Pada program ini data struct digunakan untuk pengelompokan data nama, nomor KK, dan jadwal rekam diri setiap antrian. Hal tersebut dibuktikan pada inisialisasi data struct sebagai berikut,

```

1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  #include <time.h>
5  #include "subroutines.h"
6
7  typedef struct{
8      int data[100];char data1[100][20];
9      char data2[100][30];char data3[100][15];
10     int depan;int belakang;}
11     Queue;Queue antrian;Queue nokk;Queue nama;

```

**Gambar 016.** Penggunaan data struct pada program

- Adanya penjelasan dalam komentar program

Penjelasan pada program hanya dipaparkan pada instruksi-instruksi tertentu yang dianggap tidak terlalu jelas penggunaannya. Berikut merupakan penggunaan komentar pada koding program,

```

85 int main(){
86     system("color F0");system("mode 81, 20");           //warna tulisan,warna background,ukuran layar output
87     int but=1;int butmax=4;unsigned char ch;int n;int a=1;buat();
88     judul();delay(1000);system("CLS");petunjuk();delay(1000);system("CLS");
89     back:
90     while(but!=1){
91         if(but>butmax){
92             but=1;                                         //nilai but berputar dari 1 ke 4 dan kembali lagi ke 1
93         }else if(but<1){
94             but=butmax;
95             system("CLS");menu(but);ch=getch();
96         }if (ch==0){
97             ch=getch();                                   //initial read input char dari keyboard
98         }if (ch==72){
99             but--;                                         //decrement nilai button bila ditekan panah ke atas
100        }else if (ch==80){
101            but++;                                         //increment nilai button bila ditekan panah ke bawah
102        }if(ch==77){
103            system("CLS");
104            switch (but){
105                case 1 :
106                    system("CLS");kop();
107                    printf("\n\t\tAntrian          : %i\n", a);data=a;           //Generate nomor antrian
108                    printf("\n\t\tNama Lengkap      : "); scanf("%s",&data1);fflush(stdin);

```

**Gambar 017.** Penggunaan komentar pada koding program

- Semua variable yang digunakan merepresentasikan artinya

Variabel menjelaskan fungsinya, namun terdapat beberapa variable yang sengaja disingkat penulisannya agar lebih singkat. Contohnya : button disingkat but, buttonmax disingkat butmax.

- Penulisan program rapi

Koding program ini terbagi atas beberapa program modular, dimana diantaranya dituliskan pada header subroutine.h, sedangkan program modular utama ditulis satu file dengan program main().

- Terdapat implementasi Stack atau Queue



Queue diimplementasikan pada program ini untuk membentuk suatu antrian. Tidak hanya itu dequeue pun diimplementasikan dengan format first in first out.

## 5. Koding Program

### Isi file Antrian Pengajuan E-KTP

```
1  #include <stdio.h>
2  #include <conio.h>
3  #include <string.h>
4  #include <time.h>
5  #include "subroutines.h"
6
7  typedef struct{
8      int data[100];char data1[100][20];
9      char data2[100][30];char data3[100][15];
10     int depan;int belakang;}
11 Queue;Queue antrian;Queue nokk;Queue nama;
12
13 int pilihan, data, i, j,n;
14 char data1[20], data2[30], data3[15];
15
16 int antrianKosong(){
17     if (antrian.belakang==--1)
18         return 1;
19     else
20         return 0;}
21
22 int antrianTerisi(){
23     if(antrian.belakang==100)
24         return 1;
25     else
26         return 0;}
27
28 void Enqueue(int data, char data1[20], char data2[30]){
29     if(antrianKosong()==1){
30         antrian.depan=antrian.belakang=0;
31         nama.depan=nama.belakang=0;
32         nokk.depan=nokk.belakang=0;
33         antrian.data[antrian.belakang]=data;
34         for(i=0;i<20;i++){
35             nama.data1[nama.belakang][i]=data1[i];}
36         for(i=0;i<15;i++){
37             nokk.data2[nokk.belakang][i]=data2[i];}
38         printf("\n\t\t\t\t\t =====DATA TERSIMPAN===== ");
39         jadwal(data);delay(3000);}
40     else if(antrianTerisi()==0){
41         antrian.belakang++;
42         nama.belakang++;
43         nokk.belakang++;
44         antrian.data[antrian.belakang]=data;
45         for(i=0;i<20;i++){
46             nama.data1[nama.belakang][i]=data1[i];}
47         for(i=0;i<15;i++){
48             nokk.data2[nokk.belakang][i]=data2[i];}
49         printf("\n\t\t\t\t\t =====DATA TERSIMPAN===== ");
50         jadwal(data);delay(3000);}}
```

```

51
52 int Dequeue(){
53     if(antrianKosong()==0){
54         int i,e ;char a[20], b[30];
55         e=antrian.data[antrian.depan];
56         for(i=0;i<20;i++){
57             a[i]=nama.data1[nama.depan][i];}

58
59         for(i=antrian.depan;i<antrian.belakang;i++){
60             antrian.data[i]=antrian.data[i+1];
61             for(j=0;j<20;j++){
62                 nama.data1[i][j]=nama.data1[i+1][j];}
63             for(j=0;j<15;j++){
64                 nokk.data2[i][j]=nokk.data2[i+1][j];}}
65         antrian.belakang--;
66         nama.belakang--;
67         nokk.belakang--;
68         kop();
69         printf("\n\t\tAntrian %i dengan nama %s\n", e, a);
70         printf("\n\t\tSilahkan masuk counter rekam data!\n\n");delay(3000);
71         return e;
72         return a[20];}
73     else{
74         kop();
75         printf("\n\n\n\n\n\n\t\t\t =====ANTRIAN KOSONG===== ");}
76 }
77
78 void buat(){
79     antrian.depan=antrian.belakang=-1;
80     nama.depan=nama.belakang=-1;
81     nokk.depan=nokk.belakang=-1;
82 }
83
84

```

```

85 int main(){
86     system("color F0");system("mode 81, 20");           //warna tulisan,warna background,ukuran layar output
87     int but=1;int butmax=4;unsigned char ch;int n;int a=1;buat();
88     judul();delay(1000);system("CLS");petunjuk();delay(1000);system("CLS");
89     back:
90     while(but!=-1){
91         if(but>butmax){
92             but=1;                                       //nilai but berputar dari 1 ke 4 dan kembali lagi ke 1
93         }else if(but<1){
94             but=butmax;
95             system("CLS");menu(but);ch=getch();
96         }if (ch==0){
97             ch=getch();
98         }if (ch==72){
99             but--;                                       //decrement nilai button bila ditekan panah ke atas
100        }else if (ch==80){
101            but++;                                       //increment nilai button bila ditekan panah ke bawah
102        }if(ch==77){
103            system("CLS");
104            switch (but){
105                case 1 :
106                    system("CLS");kop();
107                    printf("\n\t\tAntrian      : %i\n", a);data=a;           //Generate nomor antrian
108                    printf("\n\t\tNama Lengkap   : "); scanf("%s",&data1);fflush(stdin);
109                    printf("\n\t\tNomor Kartu Keluarga : "); scanf("%s",&data2);fflush(stdin);
110                    savefile(data,data1,data2);
111                    Enqueue(data, data1, data2);a++;break;   //Nomor antrian bertambah dengan a++
112                case 2 :
113                    system("CLS");
114                    Dequeue();delay(1000);break;
115                case 3 :
116                    system("CLS");
117                    if(antrianKosong()==1){
118                        kop();
119                        printf("\n\n\n\n\n\n\n\t\t\t =====ANTRIAN KOSONG===== ");delay(2000);break;}

```

```

120     kop();
121     puts("\nAntrian      Nomor KK      Nama      Jadwal rekam id");
122     for(i=antrian.depan;i<=antrian.belakang;i++){
123         printf("%i      %s      %s      ", antrian.data[i], nokk.data2[i], nama.data1[i]);
124         n=antrian.data[i];cetak(n);delay(2000);
125         //buka data keseluruhan sumber dari file
126         int tombol=1,tombolmax=2;
127         unsigned char ch;FILE *f;char s[5000];char c[1000];
128         while(tombol!=-1){
129             if(tombol>tombolmax){
130                 tombol=1;                                       //nilai but berputar dari 1 ke 2 dan kembali lagi ke 1
131             }else if(tombol<1){
132                 tombol=tombolmax;
133                 system("cls");menureadfile(tombol);ch=getch();
134             }if (ch==0){
135                 ch=getch();
136             }if (ch==72){
137                 tombol--;                                       //decrement nilai button bila ditekan panah atas
138             }else if (ch==80){
139                 tombol++;                                       //increment nilai button bila ditekan panah bawah
140             }if(ch!=72||80){
141                 system("cls");printf("Invalid Input");delay(1000);
142             }
143             if(ch==77){
144                 system("CLS");
145                 switch (tombol){
146                     case 1 : kop();printf("\nAntrian\tNama\t\tNomor KK\n");
147                             f=fopen("Pengajuan.txt", "r");
148                             if(!f)
149                                 return 1;
150                             while (fgets(s,5000,f)!=NULL) {
151                                 printf("%s", s);
152                             }
153                             fclose(f);delay(2000);
154                             break;

```



```

69     printf("\n\t\t=====");
70     printf("\n\t\tJadwal rekam identitas diri anda,\n");
71     printf("\n\t\tHari,Tanggal  : ");
72     hari(day);printf(" %d ",tanggal);month(bulan);printf(" %d\n",tahun);
73     printf("\n\t\tWaktu      : Pukul %d.%d WIB\n",jam,menit);
74 }

```

```

75
76
77 void hari(int x){
78     switch(x){
79         case 1: printf("Senin,");break;
80         case 2: printf("Selasa,");break;
81         case 3: printf("Rabu,");break;
82         case 4: printf("Kamis,");break;
83         case 5: printf("Jumat,");break;}
84 }

```

```

85
86 void month(int x){
87     switch(x){
88         case 1: printf(" Januari");break;
89         case 2: printf(" Februari");break;
90         case 3: printf(" Maret");break;
91         case 4: printf(" April");break;
92         case 5: printf(" Mei");break;
93         case 6: printf(" Juni");break;
94         case 7: printf(" Juli");break;
95         case 8: printf(" Agustus");break;
96         case 9: printf(" September");break;
97         case 10:printf(" Oktober");break;
98         case 11:printf(" November");break;
99         case 12:printf(" Desember");break;}
100 }

```

```

101
102 void cetak(n){
103     #include<stdio.h>
104     int x;int menit=0;int jam=9;int day=1;int tanggal=1;int minggu=1;int bulan=1; int tahun=2018;//nilai awal tanggal dan bulan perekaman data diri
105
106     for( x=1;x<n;x++){
107         menit=menit+20;
108         if(menit>=60){
109             jam=jam+1;menit=0;}
110
111         if(jam>=15){
112             day=day+1;tanggal=tanggal+1;jam=9;}
113
114         if (day>5){
115             minggu=minggu+1;day=1;}
116
117         if (tanggal>=31){
118             bulan=bulan+1;tanggal=1;}
119
120         if (bulan==2){
121             if (tanggal>=28){
122                 bulan=bulan+1;tanggal=1;}}//bulan tahun kabisat
123
124         if (bulan>=12){
125             tahun=tahun+1;bulan=1;}
126         hari(day);printf(" %d ",tanggal);month(bulan);printf(" %d",tahun);printf(" Pukul %d.%d WIB\n",jam,menit);
127     }
128
129 void savefile(int antrian,char nama[20],char nokk[30]){
130     FILE *out=fopen("Pengajuan.txt","a");
131     fprintf(out,"\n%d\t\t%s\t\t%s\n",antrian,nama,nokk);
132     fclose(out);
133 }

```

```

134
135 void menureadfile(int pilihan){
136     char arah1[3]={' ',' ',' '};
137     arah1[pilihan]='>';
138     kop();
139     printf("\n\t\tLihat keseluruhan data pendaftar?");
140     printf("\n\t\t%c 1. Buka",arah1[1],arah1[1]);
141     printf("\n\t\t%c 2. Kembali",arah1[2],arah1[2]);
142 }

```