

TUGAS ALGORITMA DAN PEMROGRAMAN
DESKRIPSI SERTA PENJELASAN DARI FLOWCHART, KODE PROGRAM,
SERTA OUTPUT PROGRAM PERHITUNGAN TARIF PEMBAYARAN AIR
MINUM



Disusun Oleh
I Gusti Ayu Krisna Kusuma Dewi - 2205551072

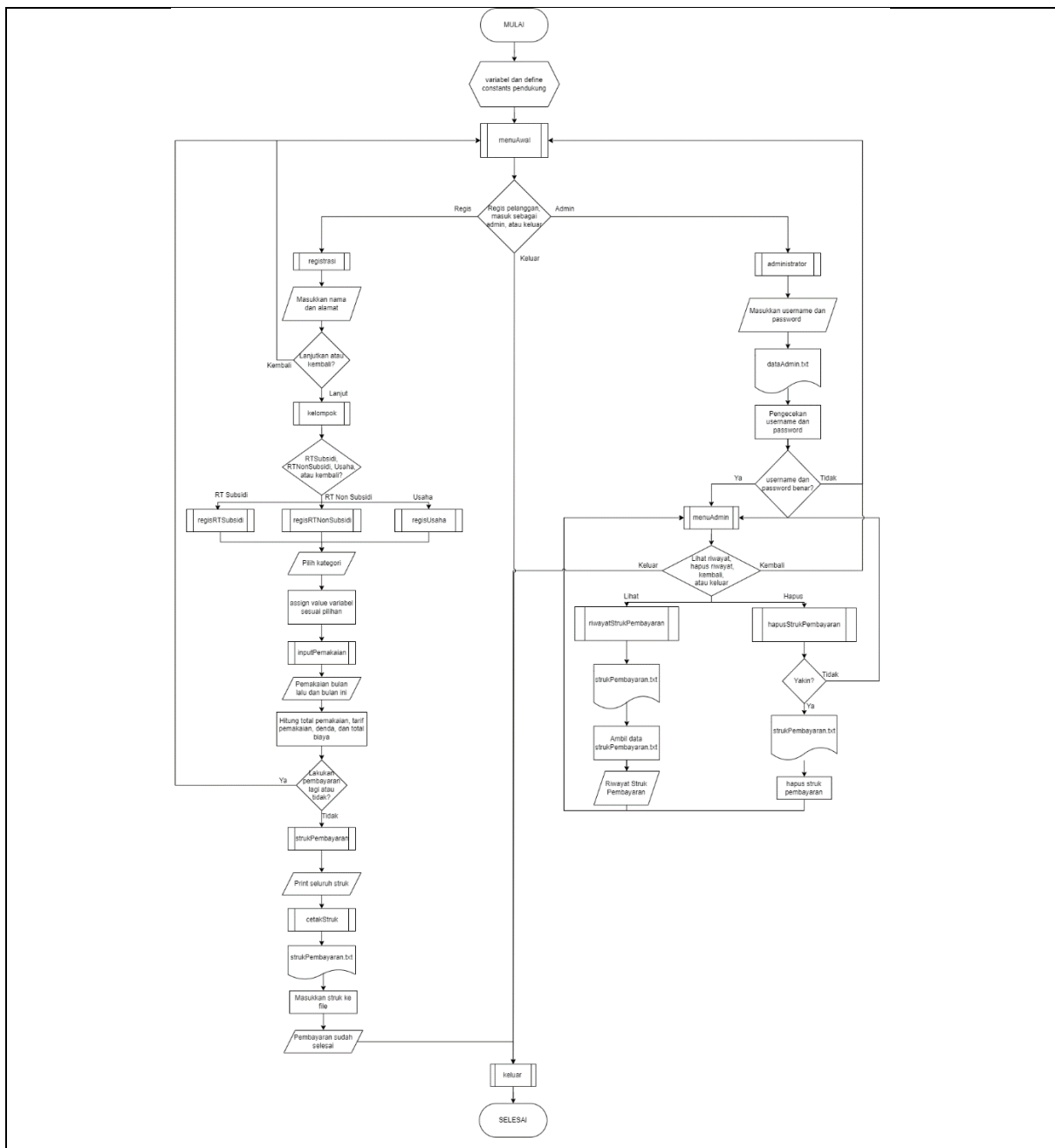
PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS UDAYANA
2022/2023

PEMBAHASAN

1. Flowchart Program

Flowchart merupakan suatu diagram yang digunakan untuk menggambarkan suatu *workflow* atau proses dari alur algoritma. Berikut adalah *flowchart* dari program perhitungan tarif pembayaran air minum.

1.1 Flowchart Sederhana

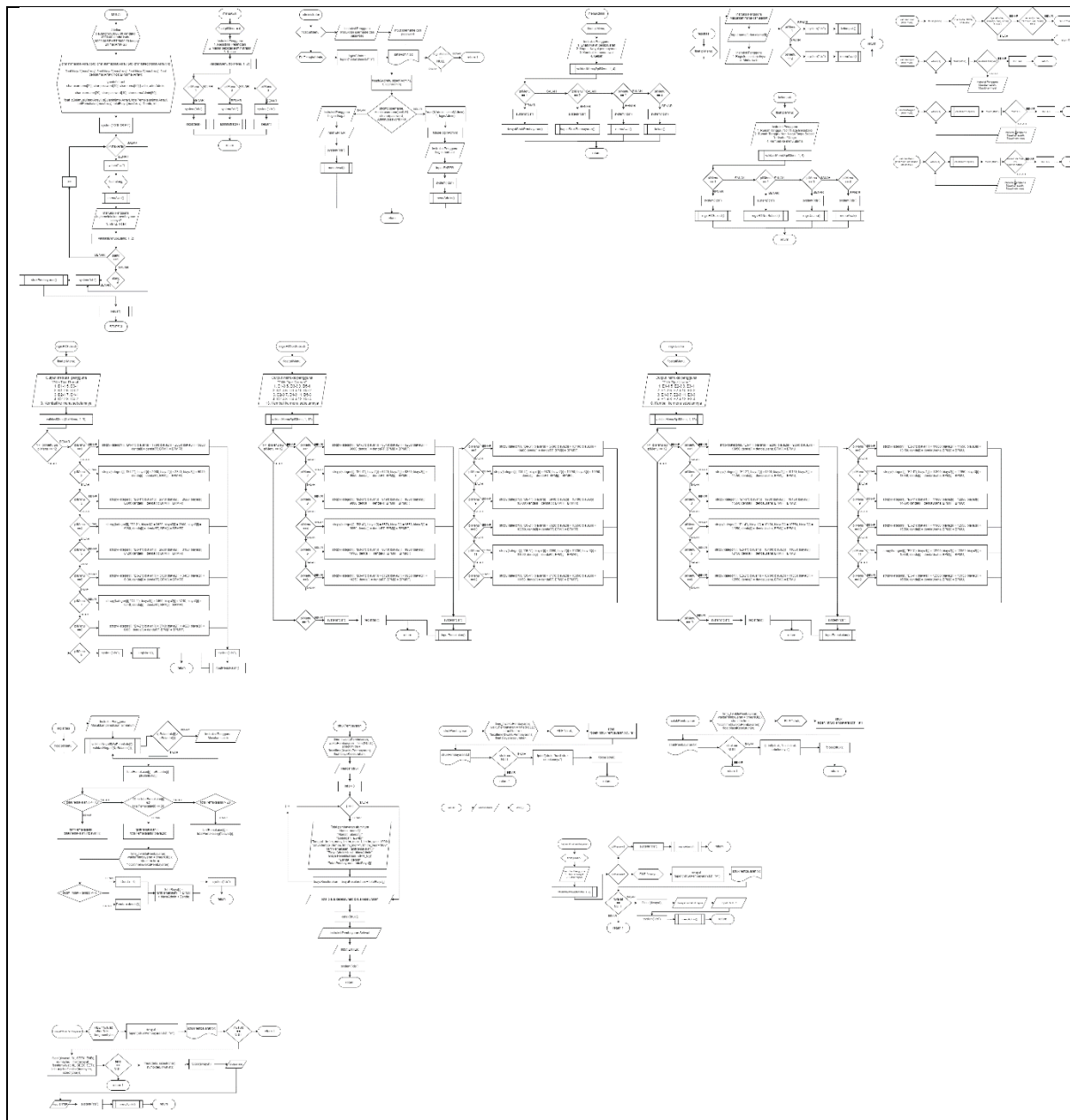


Gambar 1 Flowchart Sederhana

Demi mempermudah dalam memahami *flowchart* dan kode program dalam program ini, maka penulis telah menyiapkan suatu *flowchart* sederhana yang menjelaskan alur

berjalannya program. Sebagai pelanggan, pengguna dapat melakukan registrasi dan mendapatkan struk pembayarannya, yang mana dapat dicetak. Sebagai administrator, pengguna dapat melihat riwayat pembayaran atau menghapus riwayat struk pembayaran tersebut.

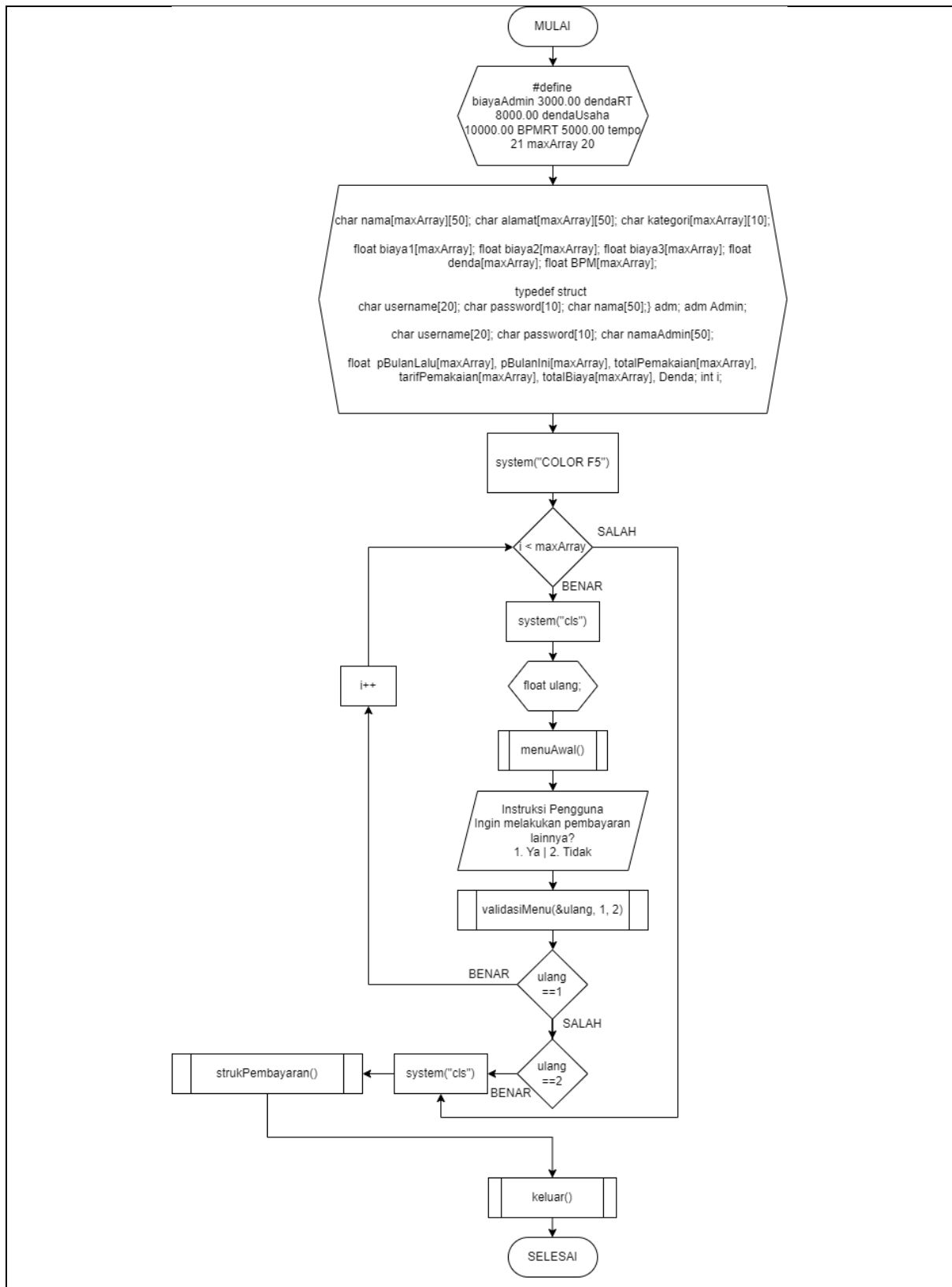
1.2 Flowchart Keseluruhan Program



Gambar 2 Flowchart Program

Flowchart di atas merupakan *flowchart* yang menjelaskan keseluruhan program. Sub-proses atau fungsi dan prosedur dijelaskan lebih lanjut pada *flowchart* di bawah ini.

1.3 Flowchart Fungsi Main()

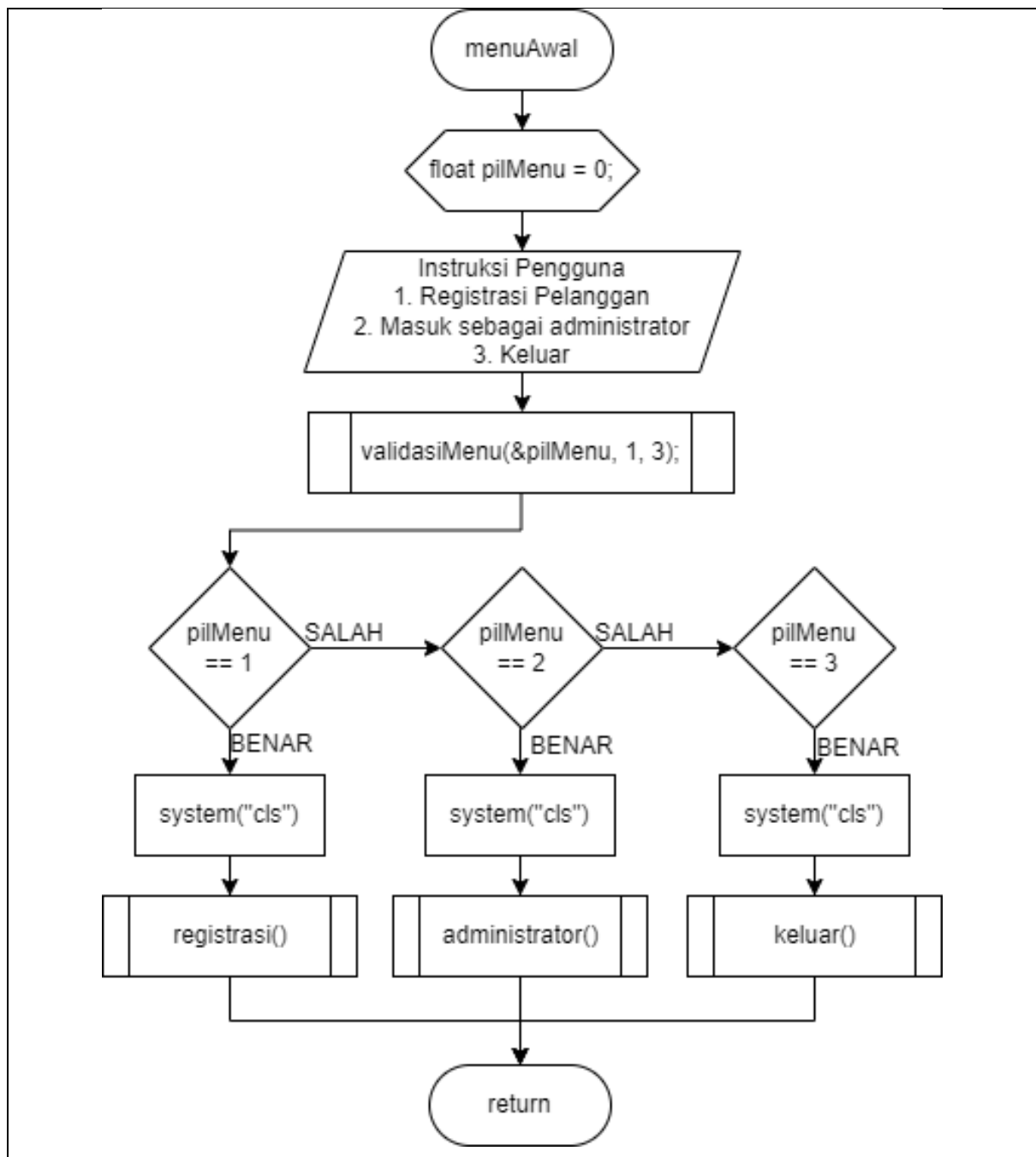


Gambar 3 Flowchart Fungsi `int main()`

Flowchart di atas merupakan alur dari fungsi `int main()`. Program ini berjalan dengan menggunakan *while loop* agar program dapat menerima lebih dari satu registrasi dan meng-assign array data dan informasi pengguna. Pada instruksi “*ingin melakukan pembayaran*

kembali?”, *loop* akan *berlanjut* jika pengguna memilih angka *1* dan pengguna pun dapat melakukan lebih dari satu registrasi pembayaran. Sebaliknya, *Loop* akan berhenti setelah pengguna memasukkan angka *2* setelah *instruksi pengguna*, yang berarti pengguna sudah selesai dalam mendaftarkan pembayarannya dan ingin menerima struk pembayaran. Program pun akan berhenti setelahnya atau saat *maxArray* sudah tercapai.

1.4 Flowchart Prosedur menuAwal()

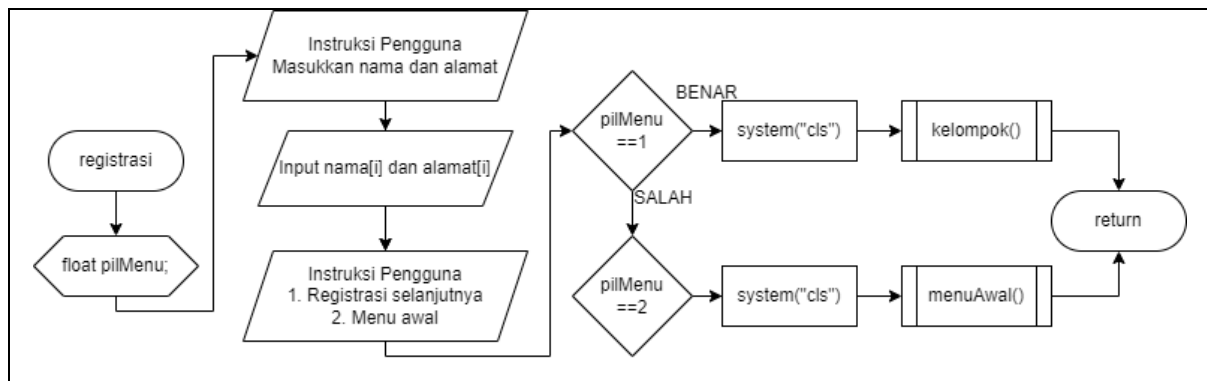


Gambar 4 Flowchart Prosedur menuAwal()

Flowchart di atas menjelaskan prosedur dari *menuAwal*. Pada bagian ini, pengguna dapat memilih sub-menu yang diinginkan. Prosedur ini bekerja dengan memanfaatkan *if-else* untuk memanggil prosedur berdasarkan pilihan pengguna, yang mana akan dijelaskan lebih lanjut pada bagian Kode Program. Tiga prosedur dapat dipanggil melalui *menuAwal* ini sesuai dengan pilihan pengguna. Setelah prosedur yang dipilih sudah dipanggil dan selesai berjalan,

maka *menuAwal* akan selesai dan kembali ke menu sebelum *menuAwal* dipanggil, yaitu fungsi *int main*.

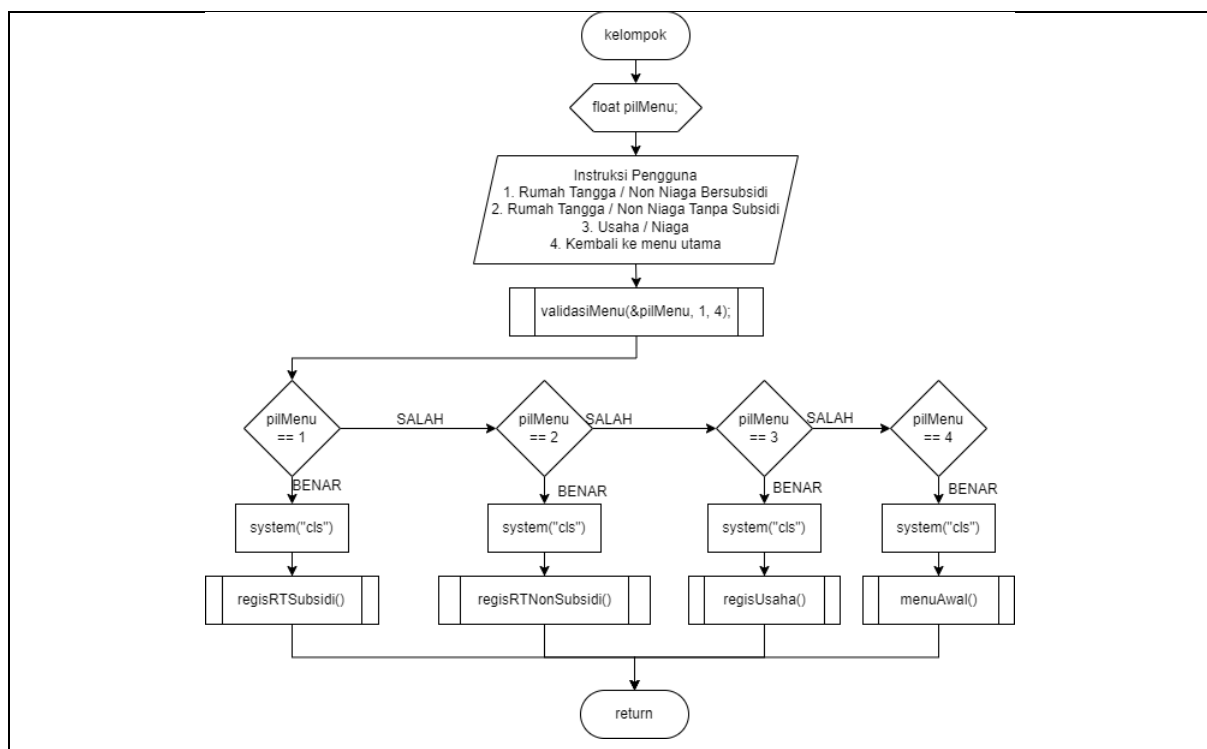
1.5 Flowchart Prosedur registrasi()



Gambar 5 Flowchart Prosedur registrasi()

Flowchart di atas merupakan alur dari prosedur *registrasi*. Pada bagian ini, pengguna akan diarahkan untuk mengisi nama serta alamat mereka. Bila ingin melanjutkan, maka pengguna dapat meng-*input* angka 1, lalu program pun akan mengarahkan pengguna ke prosedur *kelompok()* untuk memilih kategori mereka. Bila tidak, maka pengguna akan dikembalikan ke *menuAwal*.

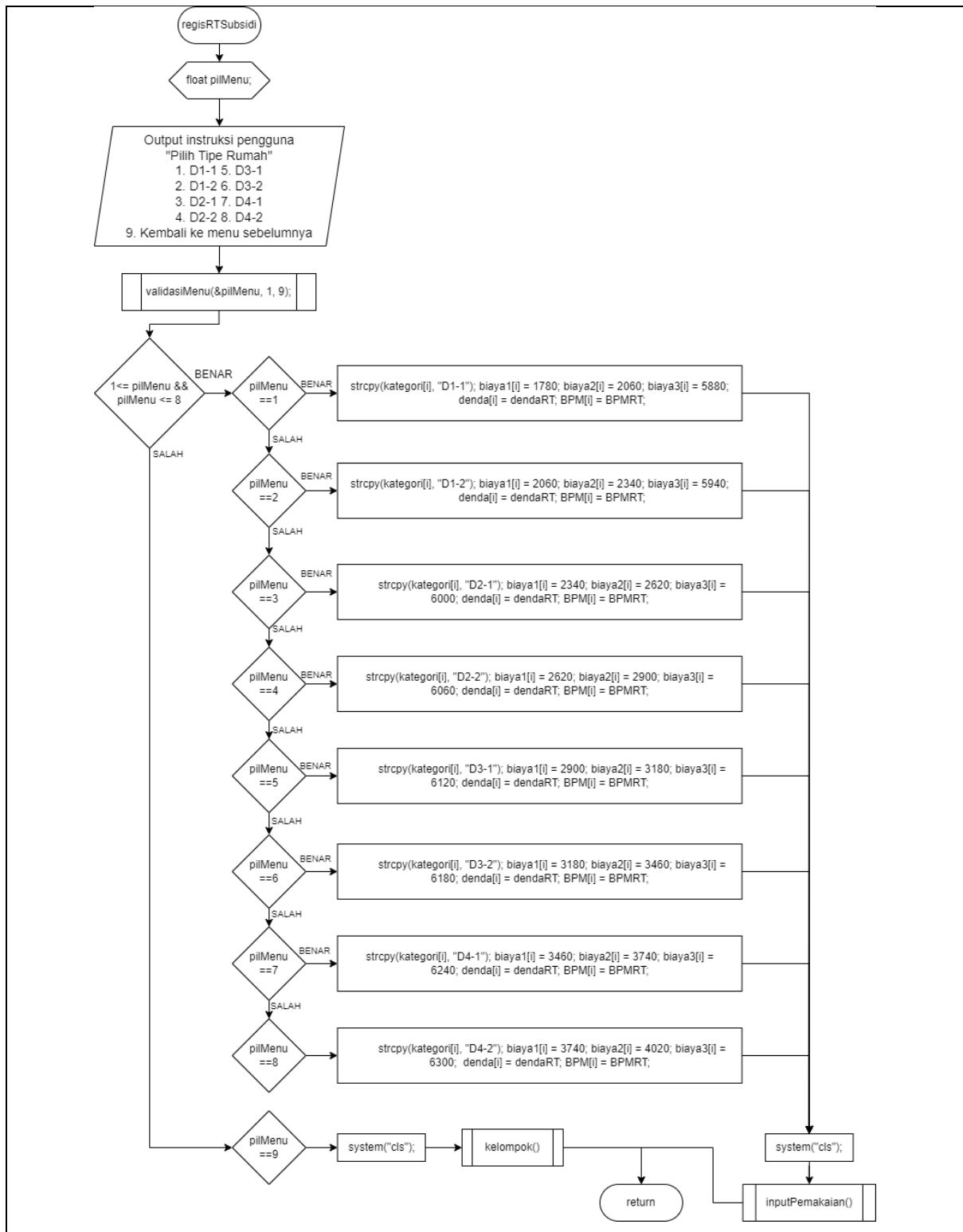
1.6 Flowchart Prosedur kelompok()



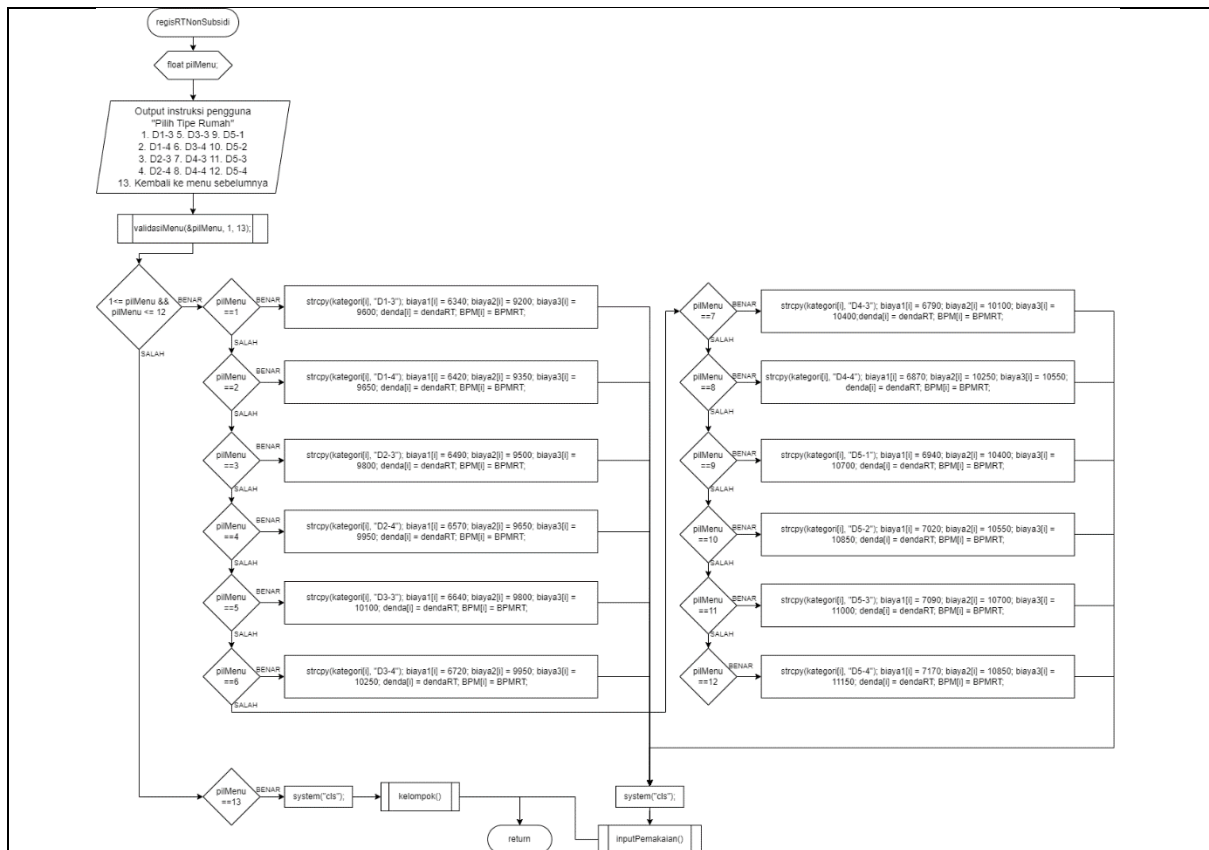
Gambar 6 Flowchart Prosedur kelompok()

Flowchart di atas menjelaskan alur dari prosedur *kelompok()*. Pada bagian ini, pengguna diberikan instruksi untuk memilih tipe kelompok bangunan mereka. Kelompok ini dibagi menjadi 3 untuk mempermudah registrasi. *Input* 1 akan mengarahkan ke prosedur *regisRTSubsidi*, 2 akan mengarahkan ke *regisRTNonSubsidi*, dan 3 akan mengarahkan ke *regisUsaha*. Bila ingin kembali ke menu sebelumnya, maka pengguna dapat meng-*input* angka 4.

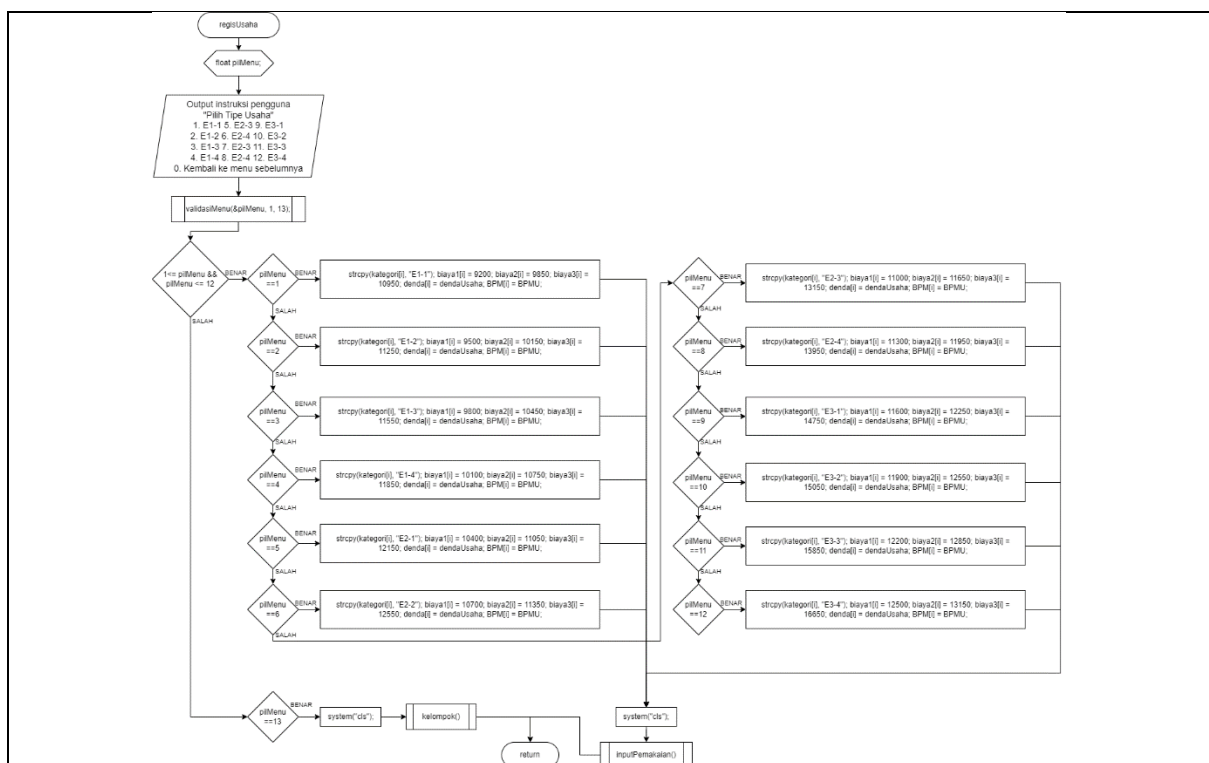
1.7 Flowchart Prosedur regisRTSubsidi, regisRTNonSubsidi, dan regisUsaha



Gambar 7 Flowchart Prosedur regisRTSubsidi()



Gambar 8 Flowchart Prosedur regisRTNonSubsidi()

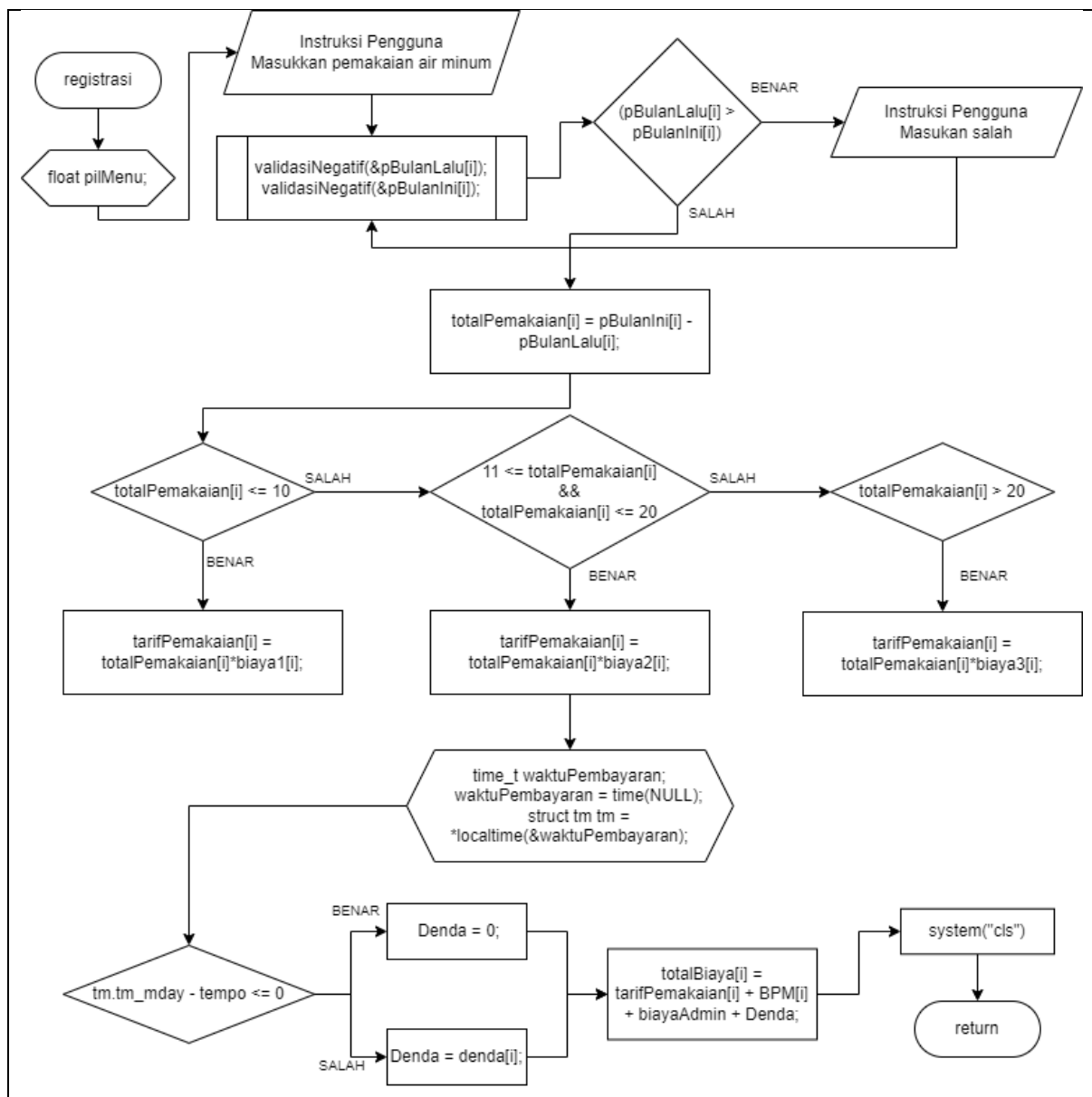


Gambar 9 Flowchart Prosedur regisUsaha()

Ketiga prosedur dalam *flowchart* di atas memiliki fungsi yang sama, yaitu memberikan instruksi bagi pengguna untuk memilih tipe kategori mereka dan men-assign nilai

yang sesuai dengan kategori tersebut ke dalam variabel yang nantinya akan digunakan untuk perhitungan tarif pembayaran. Setelah itu, pengguna pun akan diarahkan pada prosedur *inputPemakaian* untuk memasukkan penggunaan air mereka dan menghitung tarif pemakaian air.

1.8 Flowchart Prosedur *inputPemakaian*

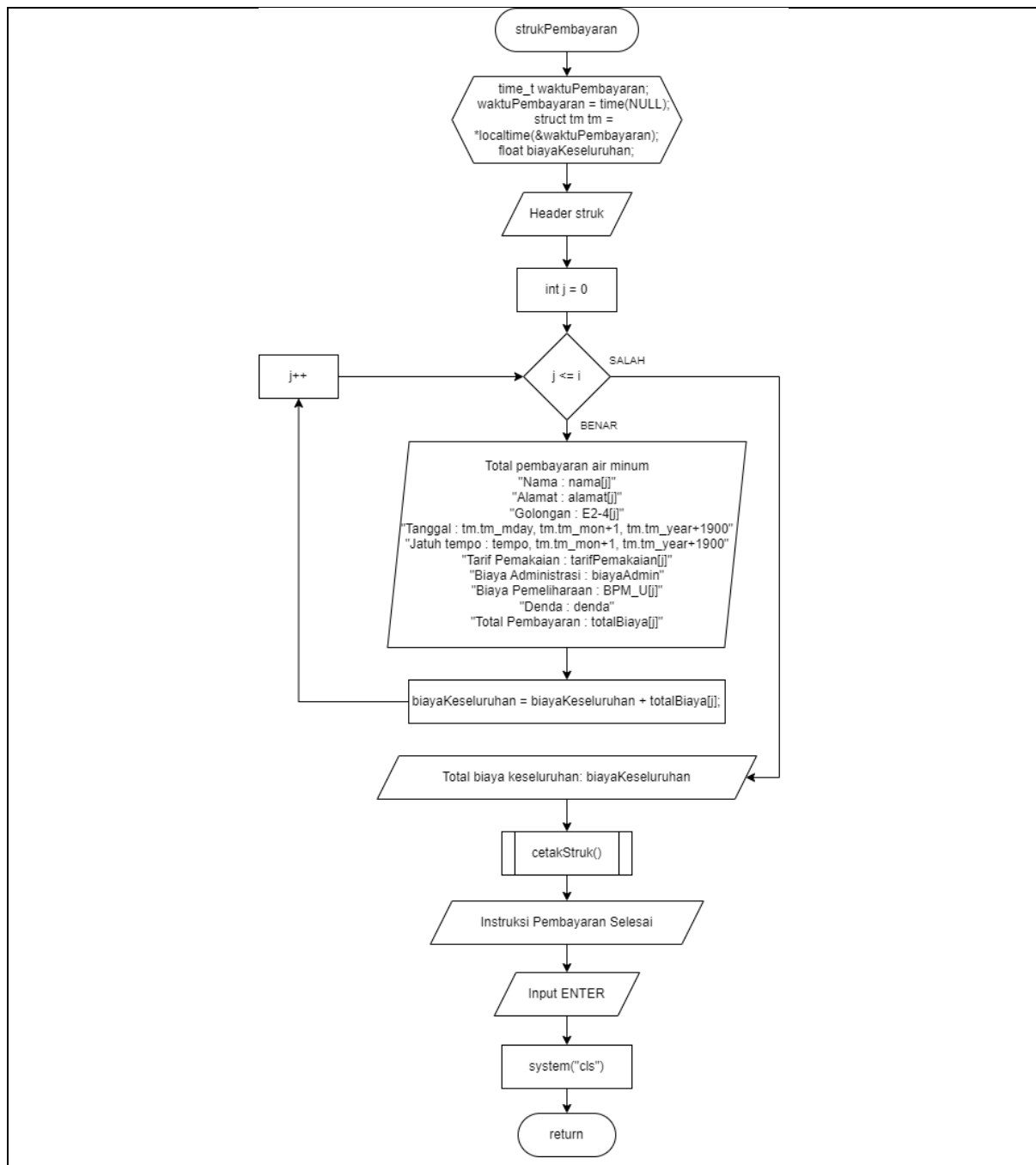


Gambar 10 Flowchart Prosedur *inputPemakaian()*

Flowchart di atas menjelaskan mengenai alur prosedur *inputPemakaian*. Setelah menyelesaikan registrasi, pengguna akan diarahkan untuk mengisi pemakaian air sebelum tanggal 21 serta pemakaian hingga saat ini. Selisih dari pemakaian tersebut adalah pemakaian yang akan digunakan sebagai perhitungan tarif. Tarif pemakaian pun dihitung sesuai dengan total pemakaian mereka, yang mana patokan tarifnya akan berubah sesuai dengan kondisi yang

dijelaskan pada *flowchart*. Setelahnya, denda pun ditentukan berdasarkan perbandingan tempo dengan waktu sistem pada saat registrasi dilaksanakan. Bila melewati tempo, maka akan ditambahkan denda. Keseluruhan total tarif pembayaran pun akan didapatkan dengan cara menjumlahkan tarif pemakaian, BPM, biaya administrasi, serta denda (jika ada). Prosedur ini pun dianggap selesai dan kembali ke dalam *int main*.

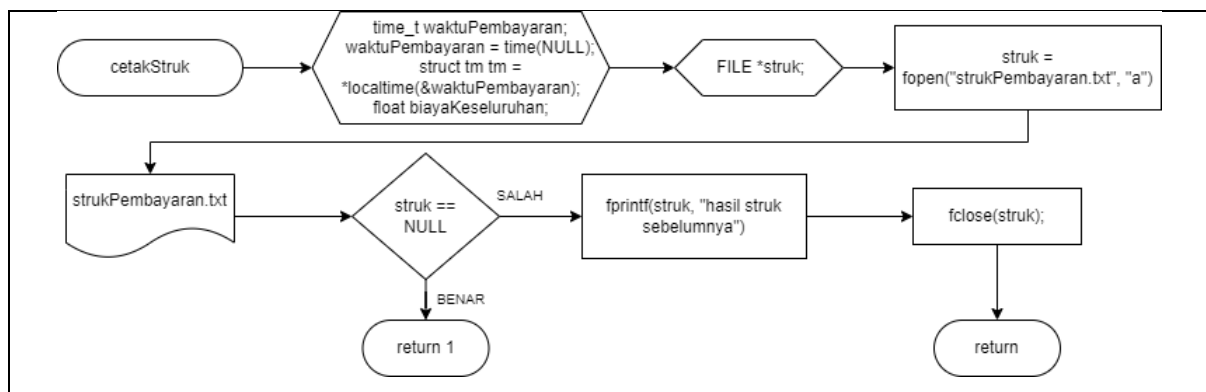
1.9 Flowchart Prosedur strukPembayaran



Gambar 11 Flowchart Prosedur strukPembayaran()

Flowchart di atas merupakan alur dari prosedur *strukPembayaran*. Setelah pengguna selesai melakukan registrasi dan meng-*input* pemakaian sekali atau lebih dari dua kali, maka pengguna akan diarahkan pada *strukPembayaran* untuk melihat rincian tarif air mereka. *strukPembayaran* ini menggunakan *for loop* agar dapat mencetak seluruh pembayaran yang dilakukan oleh pengguna. Prosedur ini juga memanggil prosedur *cetakStruk* untuk memasukkan struk pembayaran pengguna ke dalam file .txt, yang mana nantinya dapat dicetak secara fisik atau disimpan sebagai riwayat pembayaran.

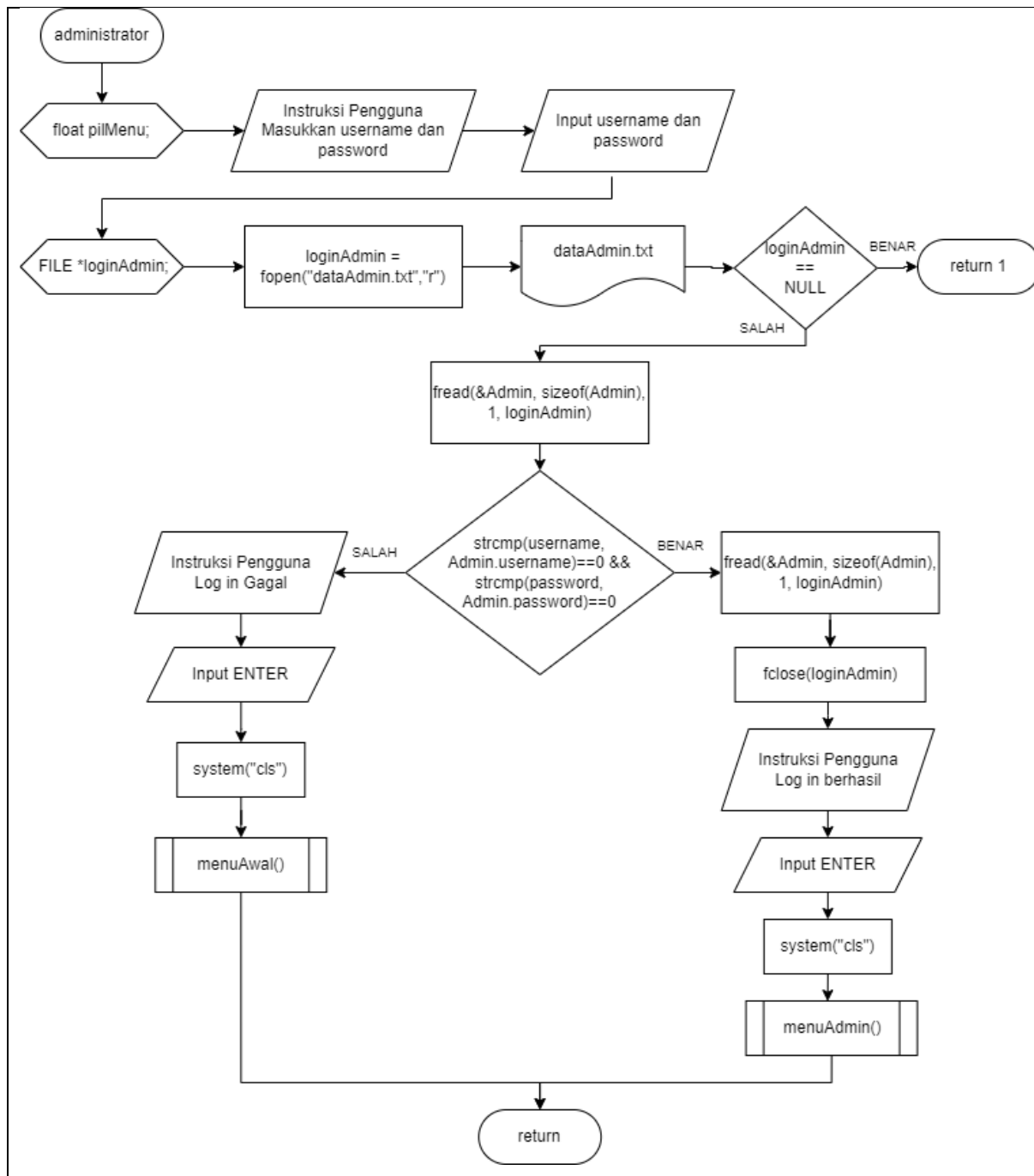
1.10 Flowchart Prosedur cetakStruk



Gambar 12 Flowchart Prosedur cetakStruk()

Pada *flowchart* prosedur *cetakStruk*, program akan membuka file *strukPembayaran.txt* untuk melakukan append atau menambahkan hasil struk pembayaran ke dalam file tersebut. File ini pun dapat diakses di luar program untuk melakukan pencetakan secara fisik, melihat riwayat pembayaran, dan bahkan dapat dimanipulasi tanpa membuka program. File ini juga tidak akan terhapus pada saat program selesai dijalankan. Cara kerja dari prosedur ini akan dijelaskan lebih lanjut di bagian Kode Program. Setelah selesai, maka program akan kembali ke prosedur atau bagian program sebelum *cetakStruk* dipanggil.

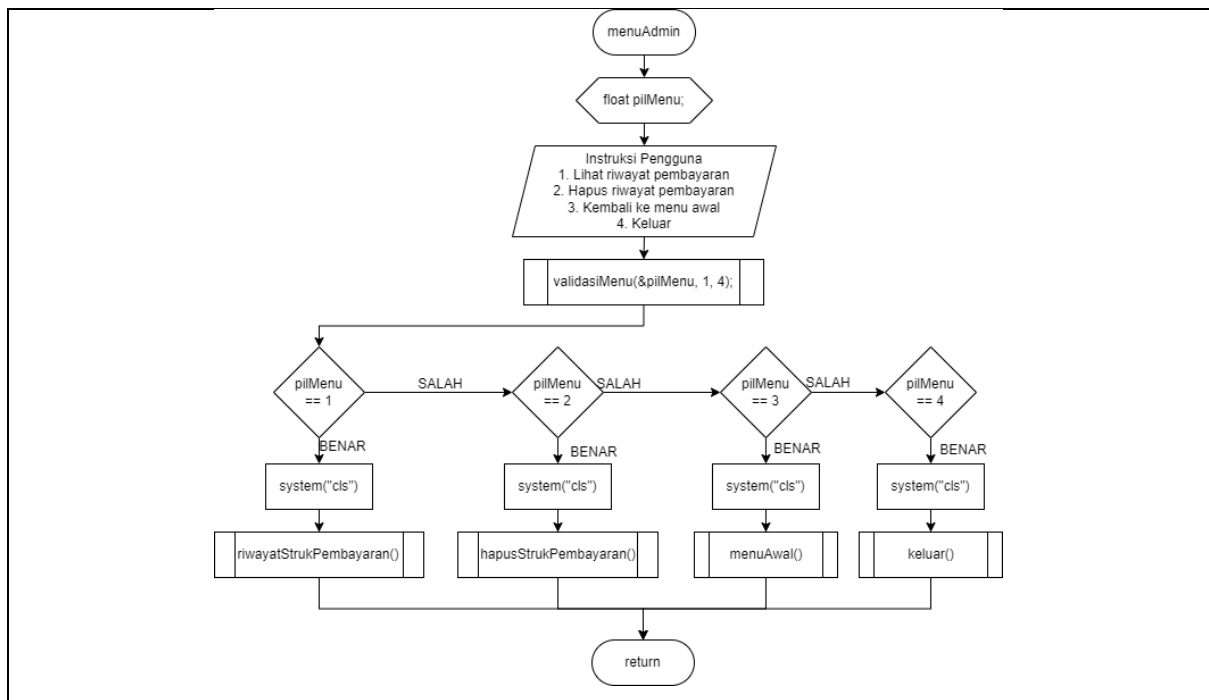
1.11 Flowchart Prosedur administrator



Gambar 13 Flowchart Prosedur administrator()

Flowchart ini menerangkan alur berjalannya prosedur *adminsitrator*. Prosedur ini akan mengarahkan pengguna untuk melakukan *log in* dengan cara memasukkan *username* dan *password* yang sebelumnya telah tersimpan pada file *dataAdmin.txt*. Jika ditemukan kecocokan, maka *log in* dinyatakan berhasil dan pengguna akan diarahkan ke *menuAdmin*. Jika salah, maka pengguna akan mendapatkan informasi bahwa *username* dan *password* salah dan lalu diarahkan kembali ke *menuAwal*. Setelah prosedur selesai, program akan kembali ke fungsi yang memanggil prosedur ini.

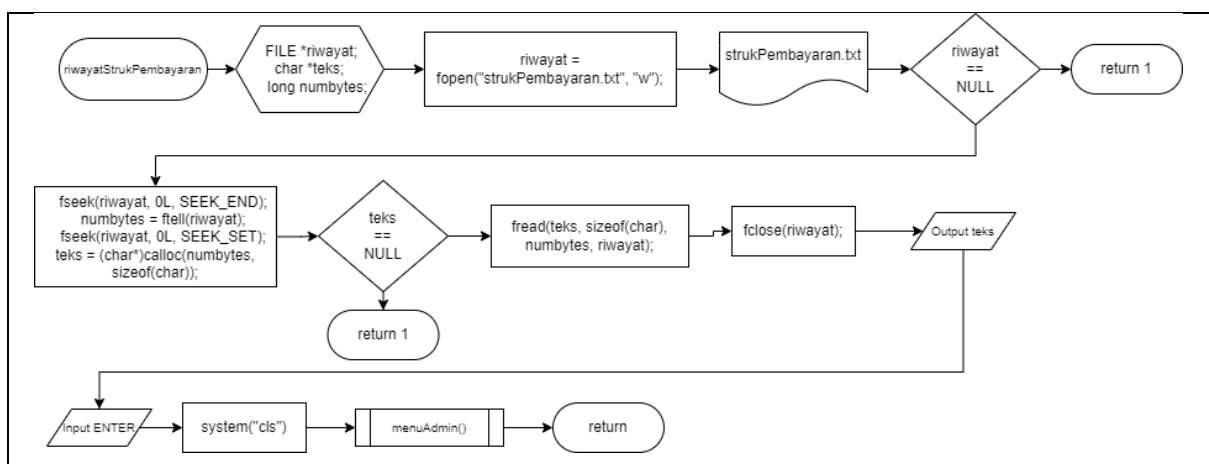
1.12 Flowchart Prosedur menuAdmin



Gambar 14 Flowchart Prosedur menuAdmin()

Flowchart di atas menggambarkan alur dari prosedur *menuAdmin*, yang mana hanya dapat diakses apabila pengguna berhasil melakukan *log in* pada prosedur sebelumnya. Dalam prosedur ini, pengguna dapat mengakses riwayat pembayaran, melakukan penghapusan riwayat pembayaran, atau pun kembali ke menu utama serta keluar dari program. Akses ini dilakukan dengan meng-*input* angka sesuai pilihan seperti menu sebelumnya.

1.13 Flowchart Prosedur riwayatStrukPembayaran

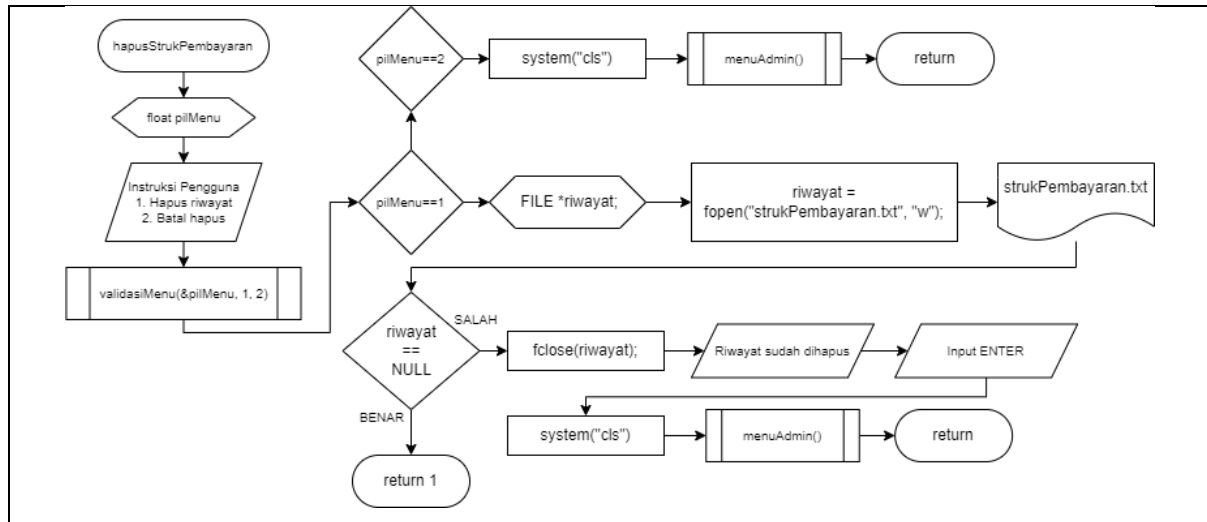


Gambar 15 Flowchart Prosedur riwayatStrukPembayaran

Flowchart di atas menjelaskan mengenai alur prosedur *riwayatStrukPembayaran*, yang mana dalam prosedur ini, file *strukPembayaran.txt* akan dibuka dan isinya ditampilkan pada layar

output program. Penjelasan lebih lanjut dapat dilihat di dalam Kode Program. Setelah selesai, maka pengguna akan kembali ke *menuAdmin*.

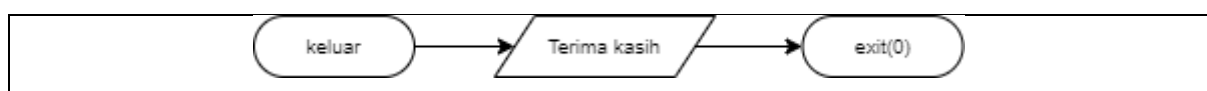
1.14 Flowchart Prosedur hapusStrukPembayaran



Gambar 16 Flowchart Prosedur hapusStrukPembayaran()

Flowchart di atas menjelaskan mengenai alur dari prosedur *hapusStrukPembayaran*. Dengan mengakses file *strukPembayaran.txt*, pengguna dapat melakukan penghapusan isi dari file tersebut. Program pun akan menanyakan apakah pengguna yakin ingin melakukan penghapusan atau tidak sebelum mengeksekusi perintah tersebut. Setelahnya, pengguna akan diarahkan pada *menuAdmin*.

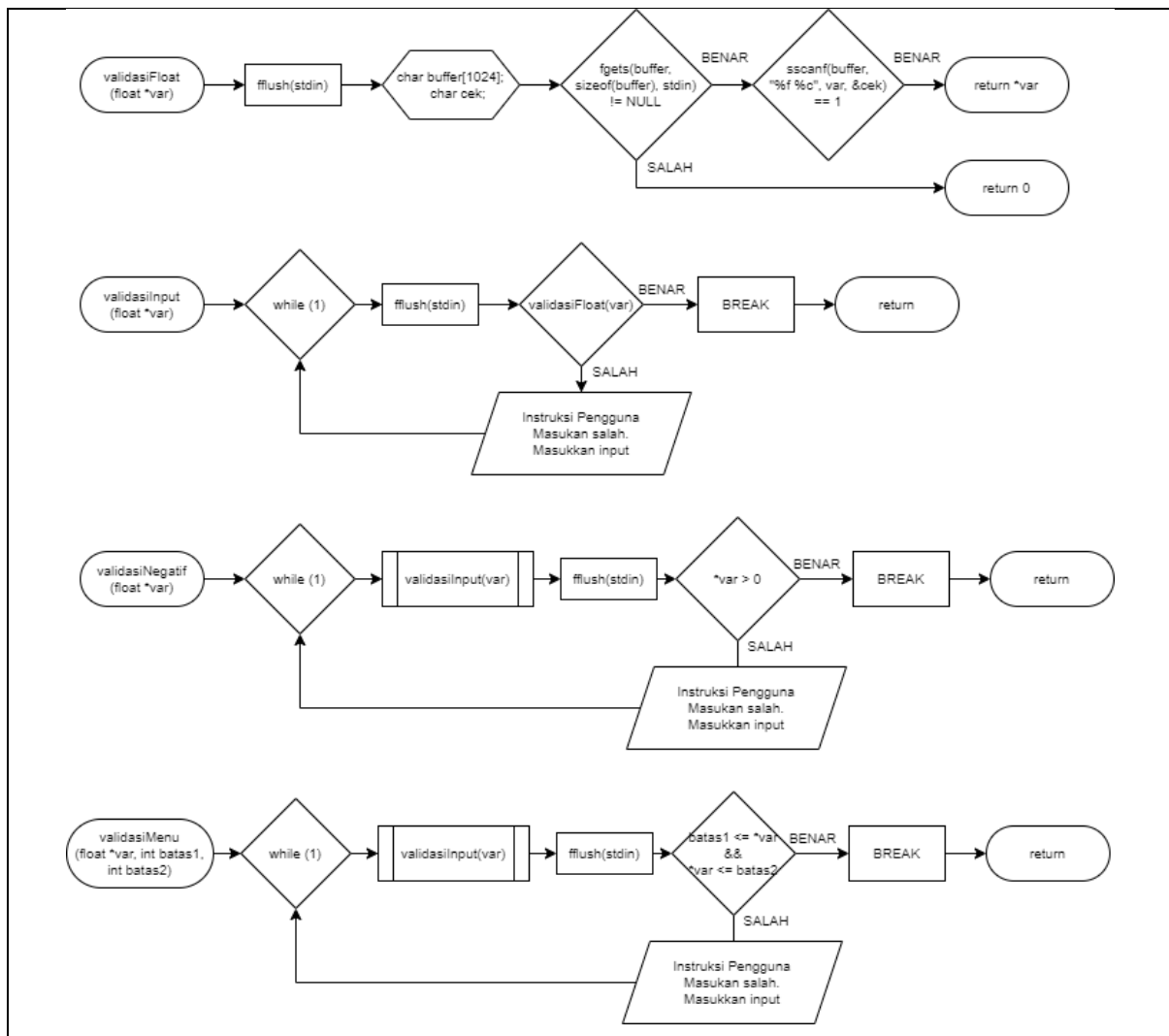
1.15 Flowchart Prosedur keluar



Gambar 17 Flowchart Prosedur keluar()

Flowchart di atas menjelaskan mengenai alur prosedur *keluar*, yang hanya terdiri dari *output Terima Kasih* dan akhirnya menghentikan berjalannya program.

1.16 Flowchart Fungsi dan Prosedur Validasi



Gambar 18 Flowchart Prosedur validasiFloat()

Flowchart ini merupakan gambaran alur dari fungsi dan prosedur yang terlibat dalam validasi *input* pengguna. Penjelasan lebih lanjut akan diberikan pada bagian Kode Program. Fungsi dan prosedur ini pun akan dipanggil untuk menggantikan *scanf()* pada umumnya.

2. Kode Program

Berikut adalah kode program lengkap dari program ini. Penjelasan lebih lanjut mengenai *syntax*, *library*, *variable*, dan lainnya dapat disimak di bawah kode program.

```

#include <stdio.h> //menunjuk kepada library standard input/output
#include <stdlib.h> //menunjuk kepada library standard library
#include <string.h> //menunjuk kepada library string
#include <time.h> //menunjuk kepada library time

#define biayaAdmin 3000.00 //Biaya administrasi Rp3000
#define dendaRT 8000.00 //Denda pemakaian melebihi
tempo untuk kategori RT Rp8000
  
```

[illegible]

```

printf("\t\t|+_____+|\n");
printf("\t\t||\n");
printf("\t\t|| Pembayaran Air Minum Anda Telah
Selesai. ||\n");
printf("\t\t|| Apakah Anda ingin melakukan pembayaran
lainnya? ||\n");
printf("\t\t||\n");
printf("\t\t|| [1] Ya || Kembali ke menu registrasi
||\n");
printf("\t\t|| [2] Tidak || Tampilkan struk dan keluar
||\n");

printf("\t\t|+_____+|\n");
printf("\t\t\n");
printf("\n\t\t\t Masukkan pilihan Anda : ");
validasiMenu(&ulang, 1, 2);
if(ulang==1) i++;
else if(ulang==2) {
    break;
}
}
system("cls");
strukPembayaran();
keluar();
return 0;
}

void menuAwal()
{
    float pilMenu = 0;
    printf("\t\t\t \n");
printf("\t\t|+_____+|\n");
printf("\t\t||\n");
printf("\t\t|| PROGRAM PENGHITUNG PEMBAYARAN AIR
||\n");
printf("\t\t|| PERUMDA AIR MINUM DENPASAR NUSANTARA
||\n");
printf("\t\t|| -----
||\n");
printf("\t\t|| [ I Gusti Ayu Krisna Kusuma Dewi || 2205551072 ]
||\n");
printf("\t\t|| =====
||\n");
printf("\t\t||\n");
printf("\t\t||\n");

```

```

printf("\t\t|| [1] Registrasi Pelanggan
||\n");
printf("\t\t|| [2] Masuk sebagai administrator
||\n");
printf("\t\t|| [3] Keluar
||\n");

printf("\t\t|+_____
+|\n");
printf("\t\t
\n");
printf("\n\t\t Pilih menu yang diinginkan: ");
validasiMenu(&pilMenu, 1, 3);
if(pilMenu==1){
    system("cls");
    registrasi();
} else if(pilMenu==2){
    system("cls");
    administrator();
} else if(pilMenu==3){
    system("cls");
    keluar();
}
}

void administrator()
{
    float pilMenu;
    printf("\t\t
\n");

printf("\t\t|+_____
+|\n");
printf("\t\t||
||\n");
printf("\t\t|| L O G I N A D M I N I S T R A T O R
||\n");
printf("\t\t|| Silahkan masukkan username dan password
Anda ||\n");
printf("\t\t||
||\n");

printf("\t\t|+_____
+|\n");
printf("\t\t
\n");
printf("\n");
printf("\t\t Username : ");
fgets(username, sizeof(username), stdin);
printf("\t\t Password : ");
fgets(password, sizeof(password), stdin);
fflush(stdin);

FILE *loginAdmin;

loginAdmin = fopen("dataAdmin.txt", "r");
if (loginAdmin==NULL) return 1;

```

```
fread(&Admin, sizeof(Admin), 1, loginAdmin);
    if(strcmp(username, Admin.username)==0 && strcmp(password,
Admin.password)==0) {
        strcpy(namaAdmin, Admin.nama);
        fclose(loginAdmin);
        printf("\t\t\n");
}

printf("\t\t|+_____\n");
printf("\t\t||\n");
printf("\t\t||\n");
printf("\t\tLOG IN BERHASIL\n");
printf("\t\tSelamat datang, %s",
namaAdmin);
printf("\t\t||\n");
printf("\t\t|+_____\n");
printf("\t\t\n");
printf("\t\tTekan ENTER untuk melanjutkan.\n");
getchar();
system("cls");
menuAdmin();
} else {
    printf("\t\t\n");
}

printf("\t\t|+_____\n");
printf("\t\t||\n");
printf("\t\t||\n");
printf("\t\tUSERNAME ATAU PASSWORD SALAH\n");
printf("\t\tMohon untuk memasukkan username dan\n");
printf("\t\tpassword yang benar ||\n");
printf("\t\t||\n");
printf("\t\t|+_____\n");
printf("\t\t\n");
printf("\t\tTekan ENTER untuk melanjutkan.\n");
getchar();
system("cls");
fflush(stdin);
menuAwal();
}

void menuAdmin()
{
```

[illegible]

```

printf("\t\t||
||\n");
printf("\t\t||
arahannya ||\n");
printf("\t\t||
||\n");

printf("\t\t|+
+|\n");
printf("\t\t
\n");
printf("\n\t\t Nama : ");
fgets(nama[i], sizeof(nama[i]), stdin);
printf("\t\t Alamat : ");
fgets(alamat[i], sizeof(alamat[i]), stdin);
fflush(stdin);
printf("\n\t\t
\n");
printf("\n\t\t Selamat datang, %s\n\n", nama[i]);
printf("\t\t|| Klik [1] untuk melanjutkan ke registrasi berikutnya.
||\n");
printf("\t\t|| Klik [2] untuk kembali ke menu awal.
||\n");
printf("\n\t\t
\n");
printf("\n\t\t Pilih menu yang diinginkan: ");
validasiMenu(&pilMenu, 1, 2);
if(pilMenu==1){
system("cls");
kelompok();
} else if(pilMenu==2){
system("cls");
menuAwal();
}
}

void kelompok()
{
float pilMenu;
printf("\t\t
\n");

printf("\t\t|+
+|\n");
printf("\t\t||
||\n");
printf("\t\t||
||\n");
printf("\t\t|| Mohon Pilih Sesuai dengan Tipe Kelompok Fasilitas
Air Anda ||\n");
printf("\t\t||
||\n");
printf("\t\t||
=====
||\n");
printf("\t\t||
||\n");

```

```

printf("\t\t|| [1] Rumah Tangga / Non Niaga Bersubsidi
||\n");
printf("\t\t|| [2] Rumah Tangga / Non Niaga Tanpa Subsidi
||\n");
printf("\t\t|| [3] Usaha / Niaga
||\n");
printf("\t\t|| [4] Kembali ke menu utama
||\n");

printf("\t\t|+
+|\n");
printf("\t\t
\n");
printf("\n\t\t Pilih tipe kelompok fasilitas : ");
validasiMenu(&pilMenu, 1, 4);
if(pilMenu==1){
    system("cls");
    regisRTSubsidi();
}else if(pilMenu==2){
    system("cls");
    regisRTNonSubsidi();
}else if(pilMenu==3){
    system("cls");
    regisUsaha();
}else if(pilMenu==4){
    system("cls");
    menuAwal();
}
}

void regisRTSubsidi()
{
    float pilMenu = 0;
    printf("\t\t
\n");

printf("\t\t|+
+|\n");
printf("\t\t||
||\n");
printf("\t\t|| RUMAH TANGGA / NON NIAGA BERSUBSIDI
||\n");
printf("\t\t|| Mohon untuk memilih tipe rumah Anda pada
daftar ini ||\n");
printf("\t\t||
||\n");
printf("\t\t||
=====
||\n");
printf("\t\t|| > Rumah Tangga A1 [Lebar Jalan : 0-3,99 meter]
||\n");
printf("\t\t|| [1] Daya Listrik 450 VA || D1-1
||\n");
printf("\t\t|| [2] Daya Listrik 900 VA || D1-2
||\n");
printf("\t\t|| -----
----- ||\n");

```



```

printf("\t\t|| > Rumah Tangga A2 [Lebar Jalan : 4-6,99 meter]
||\n");
printf("\t\t||          [3] Daya Listrik 450 VA || D2-1
||\n");
printf("\t\t||          [4] Daya Listrik 900 VA || D2-2
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| > Rumah Tangga A3 [Lebar Jalan : 7-10 meter]
||\n");
printf("\t\t||          [5] Daya Listrik 450 VA || D3-1
||\n");
printf("\t\t||          [6] Daya Listrik 900 VA || D3-2
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| > Rumah Tangga A4 [Lebar Jalan : >10 meter]
||\n");
printf("\t\t||          [7] Daya Listrik 450 VA || D4-1
||\n");
printf("\t\t||          [8] Daya Listrik 900 VA || D4-2
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t||          [9] Kembali ke menu sebelumnya
||\n");

printf("\t\t|+_____
+|\n");
printf("\t\t|_____

\n");
printf("\n\t\t Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
");
printf("\n\t\t Contoh: 1 ");
printf("\n\t\t Pilihan : ");
validasiMenu(&pilMenu, 1, 9);
if(1<= pilMenu && pilMenu <= 12) {
if(pilMenu==1){
strcpy(kategori[i], "D1-1"); biaya1[i] = 1780; biaya2[i] = 2060;
biaya3[i] = 5880;
denda[i] = dendaRT; BPM[i] = BPMRT;
}else if(pilMenu==2){
strcpy(kategori[i], "D1-2"); biaya1[i] = 2060; biaya2[i] = 2340;
biaya3[i] = 5940;
denda[i] = dendaRT; BPM[i] = BPMRT;
}else if(pilMenu==3){
strcpy(kategori[i], "D2-1"); biaya1[i] = 2340; biaya2[i] = 2620;
biaya3[i] = 6000;
denda[i] = dendaRT; BPM[i] = BPMRT;
}else if(pilMenu==4){
strcpy(kategori[i], "D2-2"); biaya1[i] = 2620; biaya2[i] = 2900;
biaya3[i] = 6060;
denda[i] = dendaRT; BPM[i] = BPMRT;
}else if(pilMenu==5){
strcpy(kategori[i], "D3-1"); biaya1[i] = 2900; biaya2[i] = 3180;
biaya3[i] = 6120;
denda[i] = dendaRT; BPM[i] = BPMRT;
}else if(pilMenu==6){
strcpy(kategori[i], "D3-2"); biaya1[i] = 3180; biaya2[i] = 3460;
biaya3[i] = 6180;

```

[illegible]

```

printf("\t\t|| -----
----- ||\n");
printf("\t\t|| > Rumah Tangga A4 [Lebar Jalan : >10 meter]
||\n");
printf("\t\t|| [7] Daya Listrik 1.300 VA || D4-3
||\n");
printf("\t\t|| [8] Daya Listrik >1.300 VA || D4-4
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| > Rumah Tangga A5 [Persil Rumah terdapat Jenis Usaha]
||\n");
printf("\t\t|| [9] Daya Listrik 450 VA || D5-1
||\n");
printf("\t\t|| [10] Daya Listrik 900 VA || D5-2
||\n");
printf("\t\t|| [11] Daya Listrik 1.300 VA || D5-3
||\n");
printf("\t\t|| [12] Daya Listrik >1.300 VA || D5-4
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| [13] Kembali ke menu sebelumnya
||\n");

printf("\t\t|+
+|\n");
printf("\t\t
\n");
printf("\n\t\t Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
");
printf("\n\t\t Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
");
printf("\n\t\t Contoh: 1 ");
printf("\n\t\t Pilihan : ");
validasiMenu(&pilMenu, 1, 13);
if(1<= pilMenu && pilMenu <= 12) {
    if(pilMenu==1){
        strcpy(kategori[i], "D1-3"); biaya1[i] = 6340; biaya2[i] = 9200;
biaya3[i] = 9600;
        denda[i] = dendaRT; BPM[i] = BPMRT;
    }else if(pilMenu==2){
        strcpy(kategori[i], "D1-4"); biaya1[i] = 6420; biaya2[i] = 9350;
biaya3[i] = 9650;
        denda[i] = dendaRT; BPM[i] = BPMRT;
    }else if(pilMenu==3){
        strcpy(kategori[i], "D2-3"); biaya1[i] = 6490; biaya2[i] = 9500;
biaya3[i] = 9800;
        denda[i] = dendaRT; BPM[i] = BPMRT;
    } else if(pilMenu==4) {
        strcpy(kategori[i], "D2-4"); biaya1[i] = 6570; biaya2[i] = 9650;
biaya3[i] = 9950;
        denda[i] = dendaRT; BPM[i] = BPMRT;
    } else if(pilMenu==5) {
        strcpy(kategori[i], "D3-3"); biaya1[i] = 6640; biaya2[i] = 9800;
biaya3[i] = 10100;
        denda[i] = dendaRT; BPM[i] = BPMRT;
    } else if(pilMenu==6) {
        strcpy(kategori[i], "D3-4"); biaya1[i] = 6720; biaya2[i] = 9950;
biaya3[i] = 10250;

```

[illegible]

```

printf("\t\t||      [3]      Daya   Listrik   1.300   VA   ||   E1-3
||\n");
printf("\t\t||      [4]      Daya   Listrik   >1.300   VA   ||   E1-4
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| >   Niaga   Sedang   [Lebar   Jalan   :   7-10   meter]
||\n");
printf("\t\t||      [5]      Daya   Listrik           450   VA   ||   E2-1
||\n");
printf("\t\t||      [6]      Daya   Listrik           900   VA   ||   E2-2
||\n");
printf("\t\t||      [7]      Daya   Listrik           1.300   VA   ||   E2-3
||\n");
printf("\t\t||      [8]      Daya   Listrik           >1.300   VA   ||   E2-4
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| >   Niaga   Besar   [Lebar   Jalan   :   >10   meter]
||\n");
printf("\t\t||      [9]      Daya   Listrik           450   VA   ||   E3-1
||\n");
printf("\t\t||      [10]     Daya   Listrik           900   VA   ||   E3-2
||\n");
printf("\t\t||      [11]     Daya   Listrik           1.300   VA   ||   E3-3
||\n");
printf("\t\t||      [12]     Daya   Listrik           >1.300   VA   ||   E3-4
||\n");
printf("\t\t|| -----
----- ||\n");
printf("\t\t||      [13]     Kembali   ke   menu   sebelumnya
||\n");

printf("\t\t|+_____
+|\n");
printf("\t\t|
\n");
printf("\n\t\t\t Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
");
printf("\n\t\t\t Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
");
printf("\n\t\t\t Contoh: 1 ");
printf("\n\t\t\t Pilihan : ");
validasiMenu(&pilMenu, 1, 13);
if(1<= pilMenu && pilMenu <= 12) {
    if(pilMenu==1) {
        strcpy(kategori[i], "E1-1"); biaya1[i] = 9200; biaya2[i] = 9850;
biaya3[i] = 10950;
        denda[i] = dendaUsaha; BPM[i] = BPMU;
    } else if(pilMenu==2) {
        strcpy(kategori[i], "E1-2"); biaya1[i] = 9500; biaya2[i] = 10150;
biaya3[i] = 11250;
        denda[i] = dendaUsaha; BPM[i] = BPMU;
    } else if(pilMenu==3) {
        strcpy(kategori[i], "E1-3"); biaya1[i] = 9800; biaya2[i] = 10450;
biaya3[i] = 11550;
        denda[i] = dendaUsaha; BPM[i] = BPMU;
    } else if(pilMenu==4) {
        strcpy(kategori[i], "E1-4"); biaya1[i] = 10100; biaya2[i] = 10750;
biaya3[i] = 11850;

```



```

printf("\t\t
\n");
printf("\t\t Mohon untuk memasukkan angka seperti contoh di bawah.
\n");
printf("\t\t Contoh: 150 atau 192.5 \n");
printf("\t\t Pemakaian air minum sebelum tanggal 21 (bulan
lalu) : ");
validasiNegatif(&pBulanLalu[i]);
printf("\t\t Total pemakaian air minum sebelum tanggal 21 hingga
sekarang (bulan ini) : ");
validasiNegatif(&pBulanIni[i]);

while (pBulanLalu[i] > pBulanIni[i]){
printf("\n\t\t Pemakaian bulan ini harus lebih besar dari bulan
lalu.\n");
printf("\t\t Pemakaian air minum sebelum tanggal 21 (bulan
lalu) : ");
validasiNegatif(&pBulanLalu[i]);
printf("\t\t Total pemakaian air minum sebelum tanggal 21 hingga
sekarang (bulan ini) : ");
validasiNegatif(&pBulanIni[i]);
}

totalPemakaian[i] = pBulanIni[i] - pBulanLalu[i];
if (totalPemakaian[i] <= 10) {
tarifPemakaian[i] = totalPemakaian[i]*biaya1[i];
} else if (11 <= totalPemakaian[i] && totalPemakaian[i] <= 20) {
tarifPemakaian[i] = totalPemakaian[i]*biaya2[i];
} else if (totalPemakaian[i] > 20) {
tarifPemakaian[i] = totalPemakaian[i]*biaya3[i];}

time_t waktuPembayaran;
waktuPembayaran = time(NULL);
struct tm tm = *localtime(&waktuPembayaran);

if ((tm.tm_mday - tempo) <= 0) {
Denda = 0;
} else {
Denda = denda[i];}

totalBiaya[i] = tarifPemakaian[i] + BPM[i] + biayaAdmin + Denda;
system("cls");
}

void strukPembayaran()
{
time_t waktuPembayaran;
waktuPembayaran = time(NULL);
struct tm tm = *localtime(&waktuPembayaran);
float biayaKeseluruhan;
printf("\t\t
\n");

printf("\t\t|+
+|\n");
printf("\t\t||
||\n");

```

```

printf("\t\t||
||\n");
printf("\t\t||
INUM ||\n");
printf("\t\t||
||\n");
for(int j = 0 ; j <= i ; j++){
printf("\t\t||
=====
||\n");
printf("\t\t|| [Identitas Pelanggan]
||\n");
printf("\t\t||
||\n");
printf("\t\t|| Nama : %s ", nama[j]);
printf("\t\t|| Alamat : %s ", alamat[j]);
printf("\t\t|| -----
----- ||\n");
printf("\t\t|| [Daftar Biaya]
||\n );
printf("\t\t||
||\n");
printf("\t\t|| Kategori : %s \n", kategori[j]);
printf("\t\t|| Total Pemakaian : %.2f m3\n",
totalPemakaian[j]);
printf("\t\t|| Tanggal : %d-%d-%d\n", tm.tm_mday,
tm.tm_mon+1, tm.tm_year+1900);
printf("\t\t|| Jatuh Tempo : %d-%d-%d\n", tempo,
tm.tm_mon+1, tm.tm_year+1900);
printf("\t\t|| Tarif Pemakaian : Rp%.2f \n",
tarifPemakaian[j]);
printf("\t\t|| Biaya Administrasi : Rp%.2f \n", biayaAdmin);
printf("\t\t|| Biaya Pemeliharaan : Rp%.2f \n", BPM[j]);
printf("\t\t|| Denda : Rp%.2f \n", Denda);
printf("\t\t|| -----
----- \n");
printf("\t\t|| Total Pembayaran : Rp%.2f \n", totalBiaya[j]);
biayaKeseluruhan = biayaKeseluruhan + totalBiaya[j];
}
printf("\t\t||
=====
||\n");
printf("\t\t||
||\n");
printf("\t\t|| Total biaya keseluruhan: %.2f\n", biayaKeseluruhan);
printf("\t\t|+
+|\n");
printf("\t\t|
\n\n");
cetakStruk();

printf("\t\t|
\n");
printf("\t\t|+
+|\n");
printf("\t\t|
||\n");

```



```
printf("\t\t||\n");
printf("\t\t||\n");
program.||\n";
printf("\t\t||\n");
printf("\t\t||\n");

printf("\t\t|+_____+|\n");
printf("\t\t\n\n");
printf("\t\t ");
getchar();
system("cls");
}

void cetakStruk()
{
    time_t waktuPembayaran;
    waktuPembayaran = time(NULL);
    struct tm tm = *localtime(&waktuPembayaran);

    FILE *struk;

    struk = fopen("strukPembayaran.txt", "a");
    if (struk == NULL) return 1;

    float biayaKeseluruhan;
    fprintf(struk, "\n\n\t\t\n");
    fprintf(struk, "\t\t|+_____+|\n");
    fprintf(struk, "\t\t||\n");
    fprintf(struk, "\t\t|| PERUMDA AIR MINUM DENPASAR NUSANTARA ||\n");
    fprintf(struk, "\t\t|| S T R U K P E M B A Y A R A N A I R M I N U M ||\n");
    fprintf(struk, "\t\t||\n");
    for(int j = 0 ; j <= i ; j++){
        fprintf(struk, "\t\t|| ===== |\n");
        fprintf(struk, "\t\t|| [Identitas Pelanggan] |\n");
        fprintf(struk, "\t\t||\n");
        fprintf(struk, "\t\t|| Nama : %s ", nama[j]);
        fprintf(struk, "\t\t|| Alamat : %s ", alamat[j]);
        fprintf(struk, "\t\t|| ----- |\n");
        fprintf(struk, "\t\t|| [Daftar Biaya] |\n");
        fprintf(struk, "\t\t||\n");
    }
}
```

```

        fprintf(struk, "\\t\\t|| Kategori : %s \\n", kategori[j]);
        fprintf(struk, "\\t\\t|| Total Pemakaian : %.2f m3\\n",
totalPemakaian[j]);
        fprintf(struk, "\\t\\t|| Tanggal : %d-%d-%d\\n", tm.tm_mday,
tm.tm_mon+1, tm.tm_year+1900);
        fprintf(struk, "\\t\\t|| Jatuh Tempo : %d-%d-%d\\n", tempo,
tm.tm_mon+1, tm.tm_year+1900);
        fprintf(struk, "\\t\\t|| Tarif Pemakaian : Rp%.2f \\n",
tarifPemakaian[j]);
        fprintf(struk, "\\t\\t|| Biaya Administrasi : Rp%.2f \\n", biayaAdmin);
        fprintf(struk, "\\t\\t|| Biaya Pemeliharaan : Rp%.2f \\n", BPM[j]);
        fprintf(struk, "\\t\\t|| Denda : Rp%.2f \\n", Denda);
        fprintf(struk, "\\t\\t|| -----
----- ||\\n");
        fprintf(struk, "\\t\\t|| Total Pembayaran : Rp%.2f \\n",
totalBiaya[j]);
        biayaKeseluruhan = biayaKeseluruhan + totalBiaya[j];
    }
    fprintf(struk, "\\t\\t||
=====
||\\n");
    fprintf(struk, "\\t\\t||
||\\n");
    fprintf(struk, "\\t\\t|| Total biaya keseluruhan: %.2f\\n",
biayaKeseluruhan);
    fprintf(struk,
"\\t\\t|+_____
+|\\n");
    fprintf(struk, "\\t\\t
_____
\\n\\n");
    fprintf(struk, "\\t\\t
_____
\\n\\n\\n");

    fclose(struk);
}

void keluar()
{
    system("cls");
    printf("\\t\\t
_____
\\n");

    printf("\\t\\t|+_____
+|\\n");
    printf("\\t\\t||
||\\n");
    printf("\\t\\t|| T E R I M A K A S I H
||\\n");
    printf("\\t\\t|| Sampai jumpa di lain waktu!
||\\n");
    printf("\\t\\t||
||\\n");

    printf("\\t\\t|+_____
+|\\n");

```

```
printf("\t\t\t\t\t");
\n");
exit (0);
}

float validasiFloat(float *var)
{
    fflush(stdin);
    char buffer[1024];
    char cek;
    if(fgets(buffer, sizeof(buffer), stdin) != NULL)
    {
        if(sscanf(buffer, "%f %c", var, &cek) == 1)
            return *var;
    }
    return 0;
}

void validasiInput(float *var)
{
    while (1)
    {
        fflush(stdin);
        if (validasiFloat(var))
            break;
        printf("\t\t\t\t\t Masukan salah. Mohon masukkan input sesuai arahan.\n");
        printf("\t\t\t\t\t Masukkan input: ");
    }
}

void validasiNegatif(float *var)
{
    while (1)
    {
        validasiInput(var);
        fflush(stdin);
        if (*var > 0)
            break;
        printf("\t\t\t\t\t Masukan salah. Mohon masukkan input sesuai arahan.\n");
        printf("\t\t\t\t\t Masukkan input: ");
    }
}

void validasiMenu(float *var, int batas1, int batas2)
{
    while (1)
    {
        validasiInput(var);
        fflush(stdin);
        if (batas1 <= *var && *var <= batas2)
            break;
        printf("\t\t\t\t\t Masukan salah. Mohon masukkan input sesuai arahan.\n");
        printf("\t\t\t\t\t Masukkan input: ");
    }
}
```

```
void hapusStrukPembayaran()
{
    float pilMenu;
    printf("\t\t\t\n");

printf("\t\t\t|+_____\n"+
_____|+\n");
    printf("\t\t\t||\n");
    printf("\t\t\t||          Apakah Anda yakin ingin menghapus riwayat
pembayaran?      ||\n");
    printf("\t\t\t||          Tekan [1] untuk YA dan [2] untuk TIDAK
||\n");
    printf("\t\t\t||\n");
    printf("\t\t\t||\n");

printf("\t\t\t|+_____\n"+
_____|+\n");
    printf("\t\t\t\n");

    printf("\t\t\t ");
    validasiMenu(&pilMenu, 1, 2);
    if(pilMenu == 1){
        FILE *riwayat;

        riwayat = fopen("strukPembayaran.txt", "w");
        if(riwayat == NULL) return 1;
        fclose(riwayat);

        printf("\t\t\t\n");

printf("\t\t\t|+_____\n"+
_____|+\n");
        printf("\t\t\t||\n");
        printf("\t\t\t||          Seluruh riwayat pembayaran telah
dihapus      ||\n");
        printf("\t\t\t||          Tekan ENTER untuk kembali ke menu
administrator.   ||\n");
        printf("\t\t\t||\n");
        printf("\t\t\t||\n");

printf("\t\t\t|+_____\n"+
_____|+\n");
        printf("\t\t\t\n");

        printf("\t\t\t ");
        getchar();
        system("cls");
        menuAdmin();
    } else if(pilMenu==2){
        system("cls");
        menuAdmin();
    }
}
```

```

void riwayatStrukPembayaran()
{
    FILE *riwayat;

    riwayat = fopen("strukPembayaran.txt", "r");

    char *teks;
    long numbytes;

    if(riwayat == NULL) return 1;

    fseek(riwayat, 0L, SEEK_END);
    numbytes = ftell(riwayat);
    fseek(riwayat, 0L, SEEK_SET);

    teks = (char*)calloc(numbytes, sizeof(char));
    if(teks == NULL)
        return 1;

    fread(teks, sizeof(char), numbytes, riwayat);
    fclose(riwayat);

    printf("%s", teks);

    printf("\t\t Tekan ENTER untuk melanjutkan.\n");
    printf("\t\t ");
    getchar();
    system("cls");
    menuAdmin();
}

```

Kode Program 1 Kode Program Lengkap

Kode Program di atas merupakan kode dari program perhitungan tarif pembayaran air ini. Detail serta penjelasan lebih lengkap dapat disimak pada pembahasan di bawah ini.

2.1 Library yang Digunakan

Program ini menggunakan *library* sebagai berikut.

```

#include <stdio.h>           //menunjuk kepada library standard input/output
#include <stdlib.h>          //menunjuk kepada library standard library
#include <string.h>          //menunjuk kepada library string
#include <time.h>            //menunjuk kepada library time

```

Kode Program 2 Deklarasi Library

Library sendiri merupakan kumpulan koleksi fungsi, definisi, serta simbol yang dapat digunakan oleh program. Setiap deklarasi *library* yang dilakukan dengan *syntax* `#include <standard include file header>`. `#include` sebagai *preprocessor directive* memiliki fungsi untuk mengarahkan program kepada *library file* yang akan digunakan dalam program.

`Stdio.h` merupakan *Standard Input/Output Header* yang berisi kumpulan fungsi serta simbol-simbol yang digunakan untuk melakukan perintah yang berhubungan dengan *input-output*, contohnya dalam program ini adalah penggunaan beberapa fungsi untuk mendapatkan

input dari pengguna atau pun memberikan *output* (`fgets()`, `scanf()`, `printf()` dll) serta mengakses, membaca, atau pun memanipulasi file luar (`fread()`, `fopen()`, `ftell()`, `fseek()`, `fprintf()`, dll), dan lainnya.

`stdlib.h` merupakan *Standard Library Header*, yang mana berisi koleksi fungsi yang digunakan untuk alokasi memori, *process control*, konversi, dan lainnya. Contoh penggunaan *predefined function* yang didapatkan dari `stdlib.h` dalam program ini adalah `exit()` untuk keluar dari program secara normal, `calloc` untuk mendapatkan ruang memori untuk sebuah *array* sebanyak *num* elemen, dan lalu meng-*initialize* setiap nilai dalam elemen tersebut menjadi 0, serta `system()` untuk memberikan *string* kepada *system command analyzer* (pada program ini menggunakan `system("cls")` untuk membersihkan layar *output* dan `system("COLOR F5")` untuk memberikan warna pada layar dan huruf).

`string.h` merupakan *String Library*, yang berisi kumpulan fungsi untuk melakukan manipulasi terhadap *string* pada program. Contoh penggunaan *predefined function* yang didapatkan dari *library* ini adalah `strcmp` untuk membandingkan nilai dari dua *string* dan mengembalikan nilai 1 atau 0 sesuai dengan hasil perbandingannya serta `strcpy` untuk menyalin dari *string* satu ke *string* lainnya.

`time.h` merupakan *Time Library* yang berisi fungsi-fungsi yang berhubungan dengan waktu dan tanggal, yang mana dapat digunakan untuk memberikan akses standar pada manipulasi waktu. Contoh penggunaan dari `time.h` ini adalah penggunaan `time()` untuk mengembalikan waktu kalender pada saat ini, `localtime` untuk konversi waktu menjadi *tm* yang nantinya disimpan dalam `struct tm`.

2.2 Konstanta dan Variabel yang Digunakan

Berdasarkan daftar referensi yang telah didapatkan, berikut adalah konstanta yang akan digunakan dalam program ini.

#define biayaAdmin	3000.00	//Biaya administrasi Rp3000
#define dendaRT	8000.00	//Denda pemakaian melebihi tempo
untuk kategori RT Rp8000		
#define dendaUsaha	10000.00	//Denda pemakaian melebihi tempo
untuk kategori Usaha Rp10000		
#define BPMRT	5000.00	//Biaya pemeliharaan meter rumah
tangga Rp.5000		
#define BPMU	5000.00	//Biaya pemeliharaan meter usaha
Rp.5000		
#define tempo	21	//Tempo / batas waktu pembayaran
tanggal 21		
#define maxArray	20	//Memberikan elemen maksimal yang
dapat disimpan di array		
		//untuk melakukan pengulangan

Kode Program 3 Deklarasi Konstanta #define

Konstanta tersebut akan dipanggil dengan menyebutkan nama konstanta yang telah ter-define melalui *preprocessor directive* #define. Jika *processor* melihat nama konstanta tersebut, maka akan digantikan dengan nilai yang telah ditentukan. Konstanta ini pun tidak dapat diubah nilainya.

Program ini juga mendeklarasikan beberapa variabel global yang akan digunakan. Berikut adalah variabel global tersebut beserta dengan *comment* yang menjelaskan fungsi dari variabel tersebut.

```
// prosedur serta fungsi yang digunakan
void menuAwal(); void registrasi(); void menuAdmin(); void administrator();
    void kelompok(); void regisRTSubsidi(); void regisRTNonSubsidi(); void
regisUsaha();
    void    inputPemakaian();    void    strukPembayaran();    void
hapusStrukPembayaran(); void riwayatStrukPembayaran();
    void keluar();

// array data pembayaran
char    nama[maxArray][50];    char    alamat[maxArray][50];    char
kategori[maxArray][10];
float biaya1[maxArray]; float biaya2[maxArray]; float biaya3[maxArray];
float denda[maxArray]; float BPM[maxArray];

//struct data admin
typedef struct {
    char username[20];
    char password[10];
    char nama[50];
} adm;
adm Admin;

//variabel penyimpan data admin selama log in
char username[20]; char password[10]; char namaAdmin[50];

// variabel perhitungan biaya
float    pBulanLalu[maxArray],    pBulanIni[maxArray],
totalPemakaian[maxArray],
    tarifPemakaian[maxArray], totalBiaya[maxArray], Denda;
```

Kode Program 4 Deklarasi Variabel, Array, dan Struct

Dapat dilihat pada Kode Program di atas bahwa ada beberapa deklarasi *array* serta *struct*. Program ini menggunakan *array* agar dapat menyimpan lebih dari satu nilai di dalam satu variabel yang sama, yang mana nantinya akan memungkinkan pengulangan registrasi bagi pengguna nantinya. *Array* ini pun dapat diakses atau dimanipulasi dengan menggunakan iterasi. *Struct* digunakan untuk menyimpan nilai pada saat admin ingin melakukan *log in*.

2.3 Statement serta Fungsi dan Prosedur yang Digunakan

Berikut adalah beberapa fungsi dan prosedur yang dideklarasikan dan didefinisikan di dalam kode program ini.

```
// fungsi dan prosedur untuk validasi input
```

```
float validasiFloat(float *var); void validasiInput(float *var); void
validasiNegatif(float *var);
    void validasiMenu(float *var, int batas1, int batas2);

// prosedur serta fungsi yang digunakan
void menuAwal(); void registrasi(); void menuAdmin(); void administrator();
    void kelompok(); void regisRTSubsidi(); void regisRTNonSubsidi(); void
regisUsaha();
    void inputPemakaian(); void strukPembayaran(); void
hapusStrukPembayaran(); void riwayatStrukPembayaran();
    void keluar();
```

Kode Program 5 Deklarasi Fungsi dan Prosedur

Kode program di atas merupakan deklarasi fungsi serta prosedur yang akan digunakan pada program ini. Definisi dari tiap fungsi dan prosedur tersebut terletak di bawah int main, oleh karena itu diperlukan deklarasi sebelum int main dijalankan. Alur berjalannya masing-masing fungsi dan prosedur di atas dapat disimak pada bagian *flowchart*.

2.3.1 Penggunaan Iterasi

Bila disimak lebih lanjut, program ini banyak menggunakan iterasi dalam menjalankan perintahnya. Iterasi tersebut adalah *while loop* pada int main, yang mana bersifat sebagai *infinity loop* sampai variabel ulang di-assign dengan nilai selain 1. Hal ini dilakukan agar dapat men-assign nilai pada elemen *array*. Beberapa *while loop* lainnya digunakan untuk mengulang beberapa perintah sampai kondisinya bernilai salah seperti pada validasi input yang akan dijelaskan setelah ini. *For loop* pun banyak digunakan untuk mengakses *array*.

2.3.2 Penggunaan If-Else

Program ini memanfaatkan if-else untuk mengakses menu atau prosedur tertentu. Pemanfaatan ini dapat dilihat pada berbagai prosedur yang menginstruksikan pengguna untuk meng-input angka tertentu untuk dapat mengakses menu yang diinginkan. *If-else* akan membandingkan nilai yang diberikan oleh pengguna dengan kondisinya masing-masing dan menjalankan *statement* yang dikaitkan dengannya bila bernilai benar. *If-else* juga digunakan pada saat menghitung tarif pembayaran seperti penentuan denda.

2.3.3 Validasi Input

Pengguna bisa saja memberikan *input* yang tidak sesuai dengan apa yang diminta oleh program. Oleh karena itu, diperlukan suatu validasi untuk menghindari *error*. Program ini memiliki setidaknya 4 fungsi validasi. Fungsi pertama yaitu fungsi float `validasiFloat(float *var)`, digunakan untuk menerima *input* dari pengguna dan

mendeteksi adanya *input* NULL (hanya menekan ENTER tanpa memberikan input apapun) serta *input* lain selain *float*.

`validasiFloat(float *var)` akan dipanggil oleh `validasiInput(float *var)` yang akan menerima nilai return dari `validasiFloat(float *var)` sebelumnya. Bila `validasiFloat(float *var)` mengembalikan nilai `var`, maka fungsi akan keluar dari *loop*. Bila salah, maka pengguna akan diberi peringatan untuk memasukkan *input* yang benar.

Selanjutnya adalah `validasiNegatif(float *var)` yang memanggil `validasiInput(float *var)`. Setelah `validasiInput(float *var)` memastikan bahwa nilai *input* adalah valid, maka `validasiNegatif(float *var)` akan memeriksa apakah nilai tersebut bernilai negatif atau tidak. Bila tidak, maka akan keluar dari *loop*. Bila iya, maka pengguna akan diberikan peringatan kembali.

Validasi terakhir adalah `validasiMenu(float *var, int batas1, int batas2)`, yang mana digunakan pada bagian-bagian program seperti menu atau pilihan kembali dan keluar. Fungsi ini memanggil `validasiInput(float *var)`. Jika benar, maka validasi akan dilanjutkan dengan memeriksa apakah nilai tersebut melebihi batas 1 dan batas 2. Pengguna akan diberikan peringatan bila ternyata nilai yang diberikan melebihi batas tersebut.

2.3.4 Penanganan File

Program ini memiliki kemampuan untuk mengakses atau memanipulasi file luar, yang mana dapat dilihat pada beberapa prosedur seperti `cetakStruk()` untuk memberikan *output* berupa struk ke dalam file yang telah ditentukan, `riwayatStrukPembayaran()` yang menggunakan berbagai fungsi file untuk mencetak isi file tersebut ke layar program (khususnya `fread()`, `fseek()`, `calloc()`, dan lainnya), atau `hapusStrukPembayaran()` untuk menghapus seluruh riwayat pembayaran.

Dalam mengakses file, program perlu membuka file terlebih dahulu melalui `fopen()`, yang mana alamat memorinya akan di-assign kepada variabel *pointer* yang telah disiapkan sebelumnya serta menentukan bagaimana file tersebut akan diperlakukan dalam program. Penentuan tersebut dilakukan dengan menambahkan *file opening mode* di samping nama file pada saat file dibuka. Contoh dari mode tersebut adalah "w" untuk melakukan *overwrite* pada isi file bila ada, "r" untuk membaca isi file dengan cara menaruh sebuah *pointer* pada karakter pertama dari file tersebut, serta "a" untuk menaruh sebuah *pointer* pada karakter terakhir dari file tersebut dan dapat melakukan *write* tanpa adanya *overwrite*. Bila file tidak ada, maka

program akan secara otomatis membuat file tersebut. Bila file ada namun tidak dapat dibuka, maka fungsi `fopen()` akan memberikan nilai `NULL`.

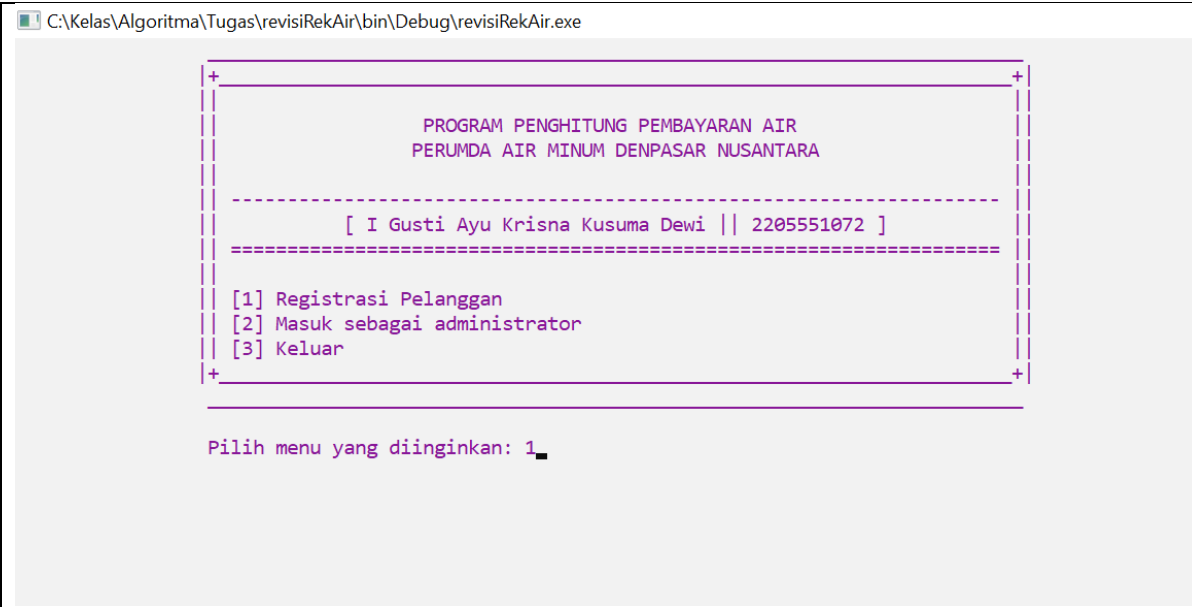
Setelah dibuka, maka program pun dapat melakukan manipulasi terhadap isi file tersebut sesuai dengan mode yang telah dideklarasikan sebelumnya. Contoh fungsi-fungsi yang digunakan adalah `fread()` dan `fscanf()` untuk membaca data dari file, `fprintf` untuk menuliskan data ke dalam file(), `fseek()` untuk menempatkan *pointer* pada posisi tertentu, `ftell()` untuk mengembalikan nilai berupa posisi *pointer* pada file saat ini, dan lainnya.

Setelah selesai, maka file akan ditutup dengan `fclose()`. Contoh *syntax* dari penanganan file tersebut dapat dilihat pada prosedur-prosedur di atas.

3. Tampilan Output

Berikut adalah tampilan *output* dari program perhitungan tarif pembayaran rekening air minum.

3.1 Menu Awal (`menuAwal()`)



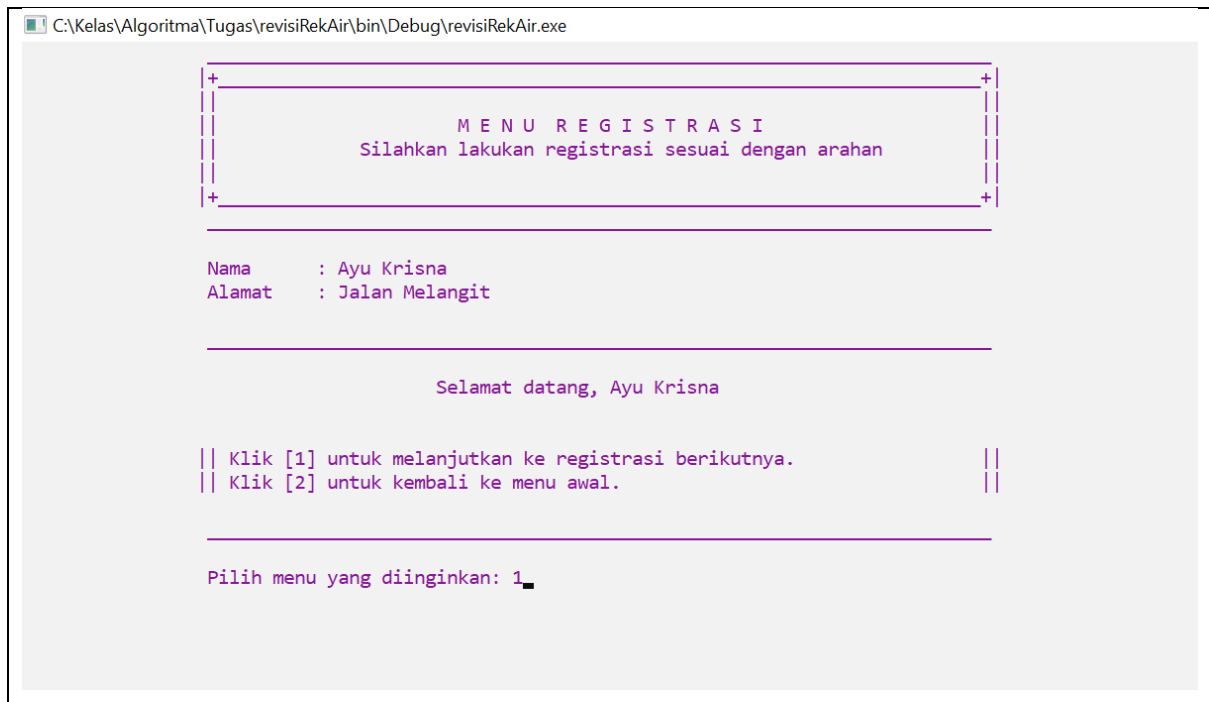
```
C:\Kelas\Algoritma\Tugas\revisiRekAir\bin\Debug\revisiRekAir.exe

+-----+
|               PROGRAM PENGHITUNG PEMBAYARAN AIR               |
|          PERUMDA AIR MINUM DENPASAR NUSANTARA          |
|-----|
| [ I Gusti Ayu Krisna Kusuma Dewi | 2205551072 ] |
|=====|
| [1] Registrasi Pelanggan                               |
| [2] Masuk sebagai administrator                         |
| [3] Keluar                                              |
+-----+

Pilih menu yang diinginkan: 1_
```

Gambar 19 Menu Awal (Output dari `menuAwal()`)

3.2 Menu Registrasi Pelanggan (registrasi(), kelompok(), regisRTSubsidi(), regisRTNonSubsidi(), regisUsaha())



Gambar 20 Menu Registrasi (Output dari registrasi())



Gambar 21 Pilih Kelompok (Output dari kelompok())

C:\Kelas\Algoritma\Tugas\revisiRekAir\bin\Debug\revisiRekAir.exe

```
+-----+
|                                     |
|          RUMAH TANGGA / NON NIAGA BERSUBSIDI          |
|      Mohon untuk memilih tipe rumah Anda pada daftar ini      |
|=====|
| > Rumah Tangga A1 [Lebar Jalan : 0-3,99 meter] |
| [1] Daya Listrik 450 VA || D1-1 |
| [2] Daya Listrik 900 VA || D1-2 |
|-----|
| > Rumah Tangga A2 [Lebar Jalan : 4-6,99 meter] |
| [3] Daya Listrik 450 VA || D2-1 |
| [4] Daya Listrik 900 VA || D2-2 |
|-----|
| > Rumah Tangga A3 [Lebar Jalan : 7-10 meter] |
| [5] Daya Listrik 450 VA || D3-1 |
| [6] Daya Listrik 900 VA || D3-2 |
|-----|
| > Rumah Tangga A4 [Lebar Jalan : >10 meter] |
| [7] Daya Listrik 450 VA || D4-1 |
| [8] Daya Listrik 900 VA || D4-2 |
|-----|
| [9] Kembali ke menu sebelumnya |
+-----+
```

Mohon masukkan pilihan sesuai dengan tipe rumah Anda.

Contoh: 1

Pilihan : 1_

Gambar 22 Tipe Rumah Tangga Subsidi (Output regisRTSubsidi())

C:\Kelas\Algoritma\Tugas\revisiRekAir\bin\Debug\revisiRekAir.exe

```
+-----+
|                                     |
|          USAHA / NIAGA              |
|      Mohon untuk memilih tipe usaha/niaga Anda pada daftar ini  |
|=====|
| > Niaga Kecil [Lebar Jalan : 0-6,99 meter] |
| [1] Daya Listrik 450 VA || E1-1 |
| [2] Daya Listrik 900 VA || E1-2 |
| [3] Daya Listrik 1.300 VA || E1-3 |
| [4] Daya Listrik >1.300 VA || E1-4 |
|-----|
| > Niaga Sedang [Lebar Jalan : 7-10 meter] |
| [5] Daya Listrik 450 VA || E2-1 |
| [6] Daya Listrik 900 VA || E2-2 |
| [7] Daya Listrik 1.300 VA || E2-3 |
| [8] Daya Listrik >1.300 VA || E2-4 |
|-----|
| > Niaga Besar [Lebar Jalan : >10 meter] |
| [9] Daya Listrik 450 VA || E3-1 |
| [10] Daya Listrik 900 VA || E3-2 |
| [11] Daya Listrik 1.300 VA || E3-3 |
| [12] Daya Listrik >1.300 VA || E3-4 |
|-----|
| [13] Kembali ke menu sebelumnya |
+-----+
```

Mohon masukkan pilihan sesuai dengan tipe rumah Anda.

Mohon masukkan pilihan sesuai dengan tipe rumah Anda.

Contoh: 1

Pilihan : 1_

Gambar 23 Tipe Rumah Tangga Non Subsidi (Output regisRTNonSubsidi())

```
C:\Kelas\Algoritma\Tugas\revisiRekAir\bin\Debug\revisiRekAir.exe

+-----+
|                                     |
|                               USAHA / NIAGA                               |
|               Mohon untuk memilih tipe usaha/niaga Anda pada daftar ini               |
|=====|
| > Niaga Kecil [Lebar Jalan : 0-6,99 meter]                                     |
| [1] Daya Listrik    450 VA || E1-1                                     |
| [2] Daya Listrik    900 VA || E1-2                                     |
| [3] Daya Listrik   1.300 VA || E1-3                                     |
| [4] Daya Listrik   >1.300 VA || E1-4                                     |
|-----|
| > Niaga Sedang [Lebar Jalan : 7-10 meter]                                     |
| [5] Daya Listrik    450 VA || E2-1                                     |
| [6] Daya Listrik    900 VA || E2-2                                     |
| [7] Daya Listrik   1.300 VA || E2-3                                     |
| [8] Daya Listrik   >1.300 VA || E2-4                                     |
|-----|
| > Niaga Besar [Lebar Jalan : >10 meter]                                     |
| [9] Daya Listrik    450 VA || E3-1                                     |
| [10] Daya Listrik   900 VA || E3-2                                     |
| [11] Daya Listrik  1.300 VA || E3-3                                     |
| [12] Daya Listrik  >1.300 VA || E3-4                                     |
|-----|
| [13] Kembali ke menu sebelumnya                                     |
|=====|
|
| Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
| Mohon masukkan pilihan sesuai dengan tipe rumah Anda.
| Contoh: 1
| Pilihan :
|
|
```

Gambar 24 Tipe Usaha (Output regisUsaha())

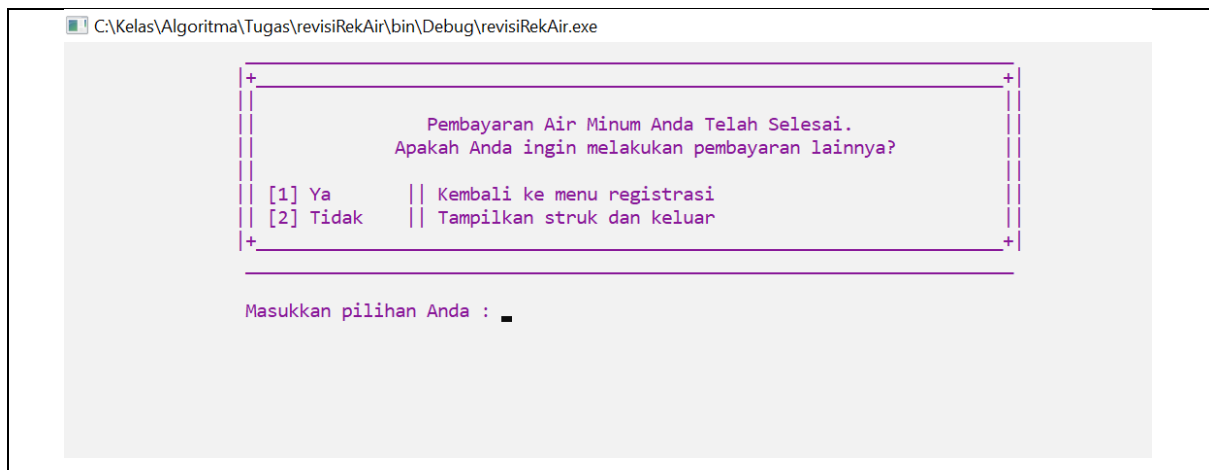
3.3 Input Pemakaian (inputPemakaian())

```
C:\Kelas\Algoritma\Tugas\revisiRekAir\bin\Debug\revisiRekAir.exe

+-----+
|                                     |
|                               INPUT PEMAKAIAN AIR MINUM                               |
|               Mohon Masukkan Pemakaian Air Minum Anda (dalam m3)               |
|=====|
|
| Mohon untuk memasukkan angka seperti contoh di bawah.
| Contoh: 150 atau 192.5
| Pemakaian air minum sebelum tanggal 21 (bulan lalu)                : 35
| Total pemakaian air minum sebelum tanggal 21 hingga sekarang (bulan ini) : 50
|
|
```

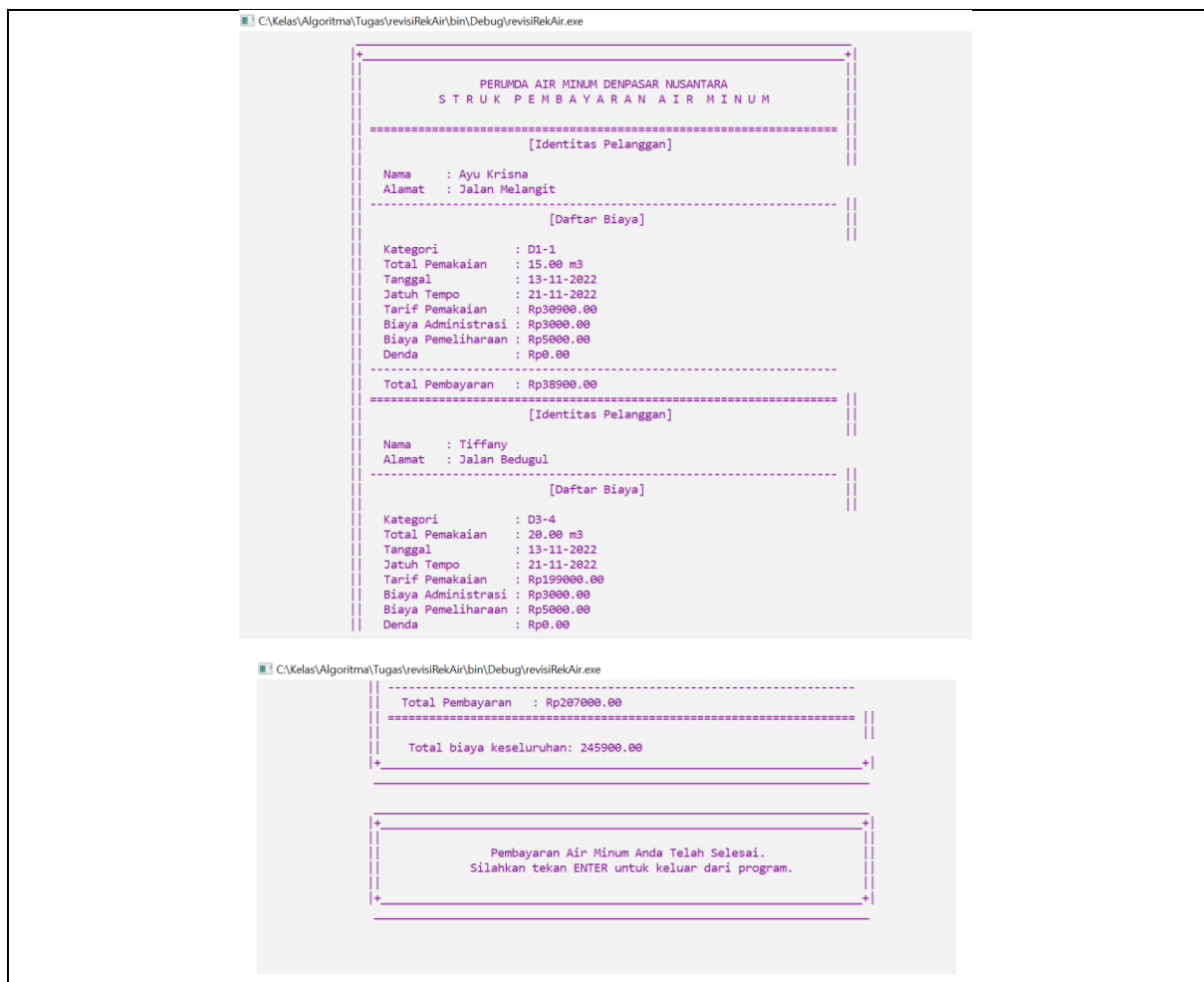
Gambar 25 Input Pemakaian Air Minum (Output inputPemakaian())

3.4 Pertanyaan Perulangan



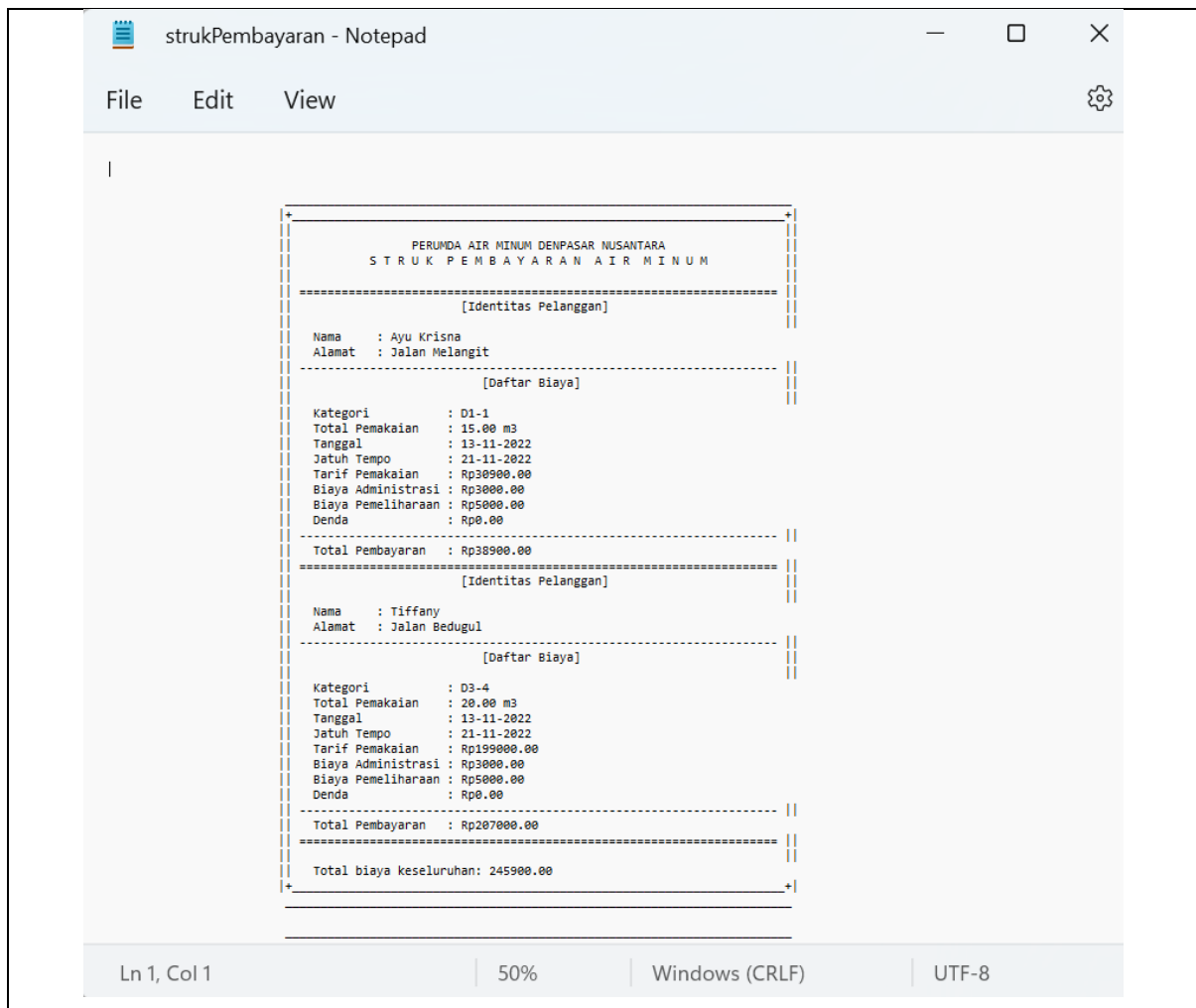
Gambar 26 Pertanyaan Perulangan (Output int main())

3.5 Struk Pembayaran (strukPembayaran())



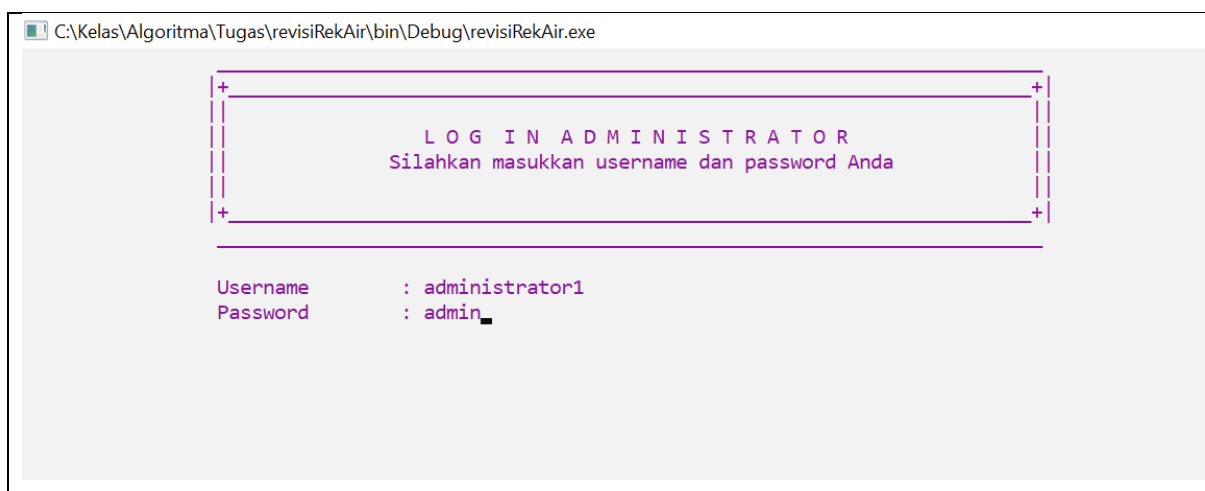
Gambar 27 Struk Pembayaran (strukPembayaran())

3.6 File strukPembayaran (setelah cetakStruk())



Gambar 28 File strukPembayaran setelah cetakStruk()

3.7 Masuk Sebagai Administrator (administrator())

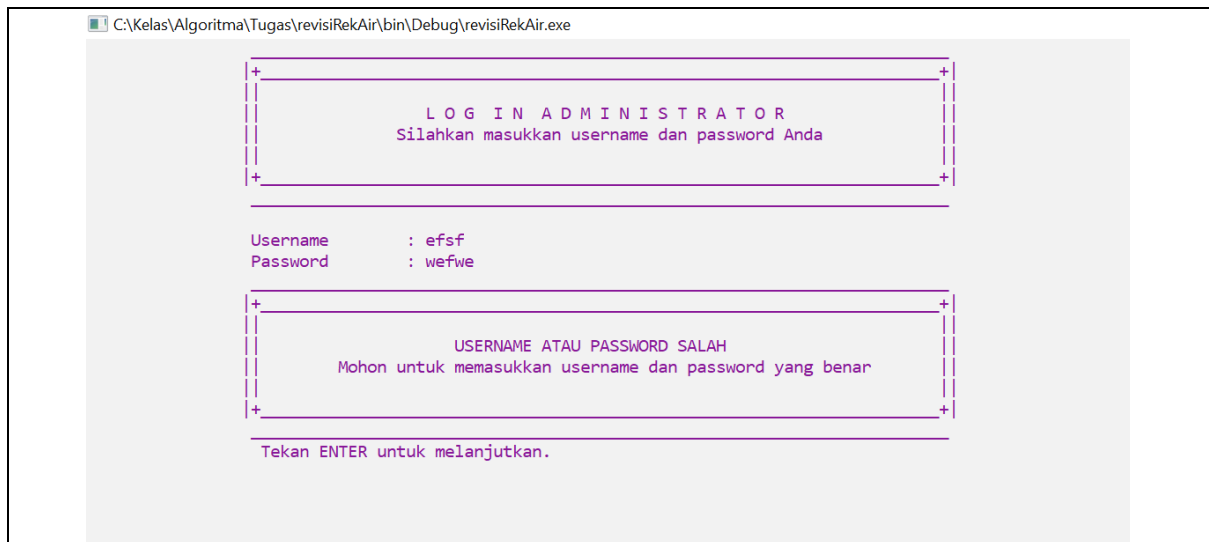


Gambar 29 Log In untuk Admin (administrator())

3.8 Log In Berhasil/Gagal

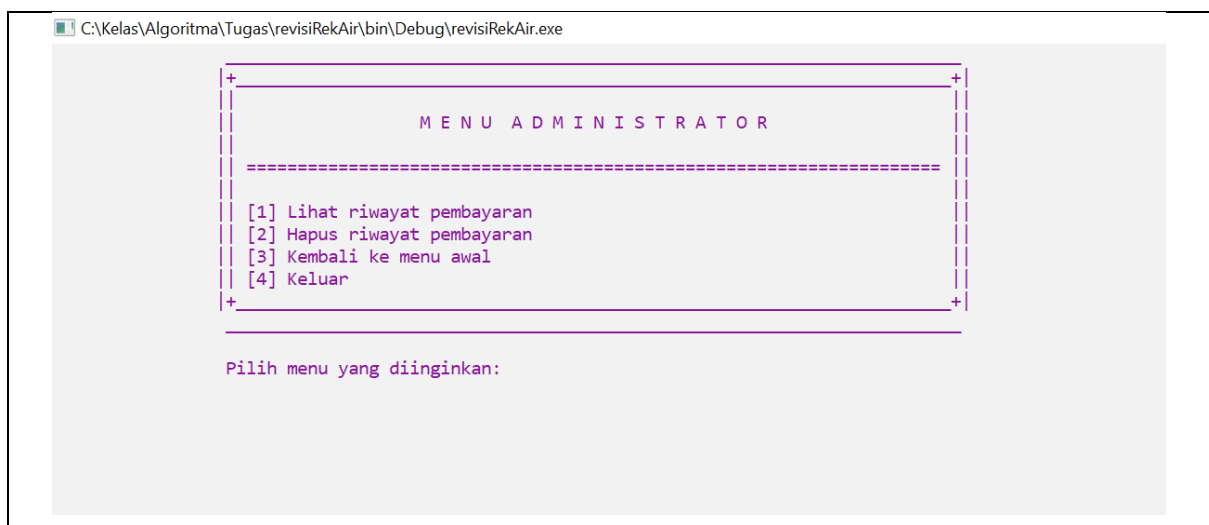


Gambar 30 Log In Berhasil (administrator())



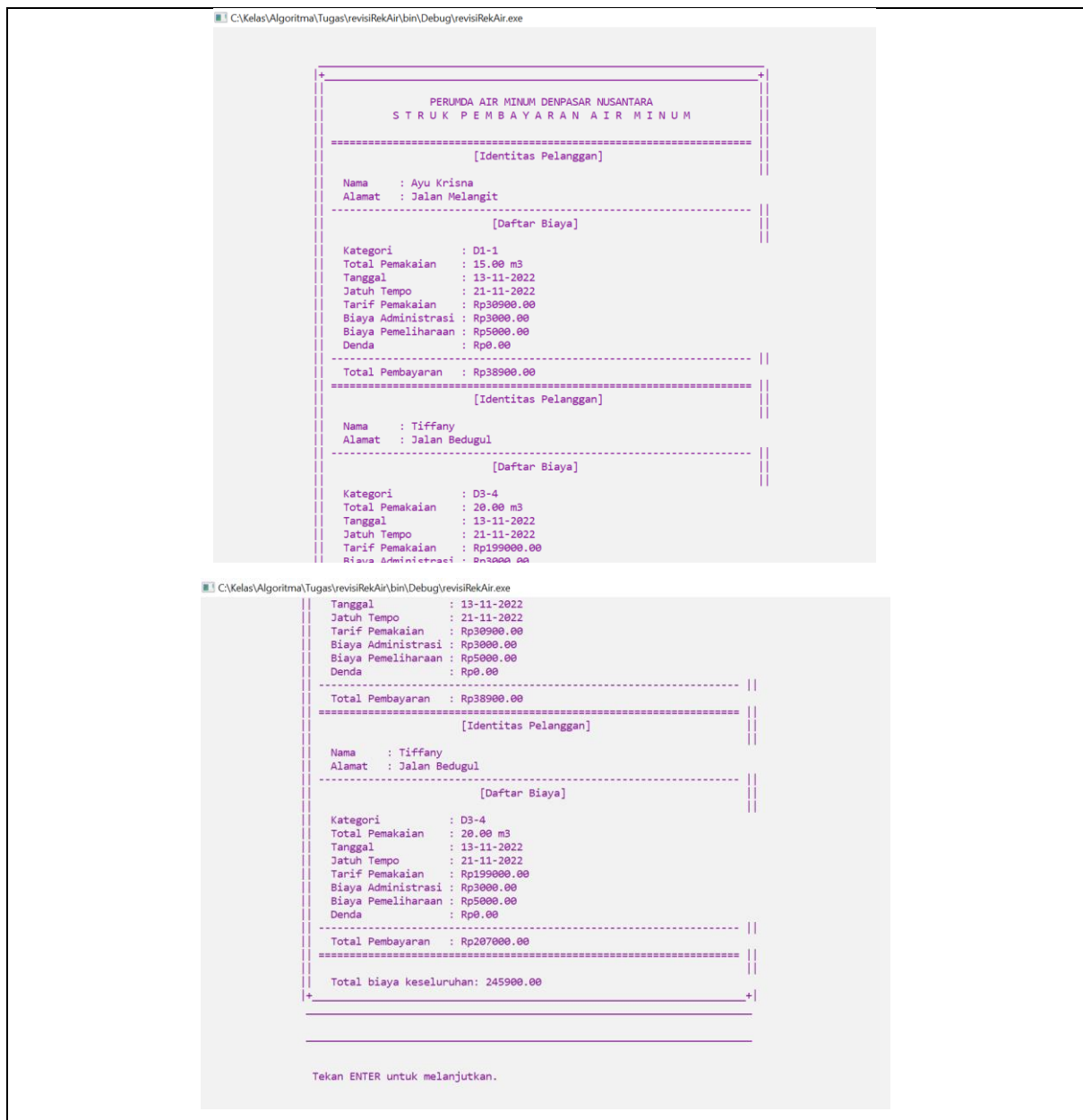
Gambar 31 Log In Gagal (administrator())

3.9 Menu Administrator (menuAdmin())



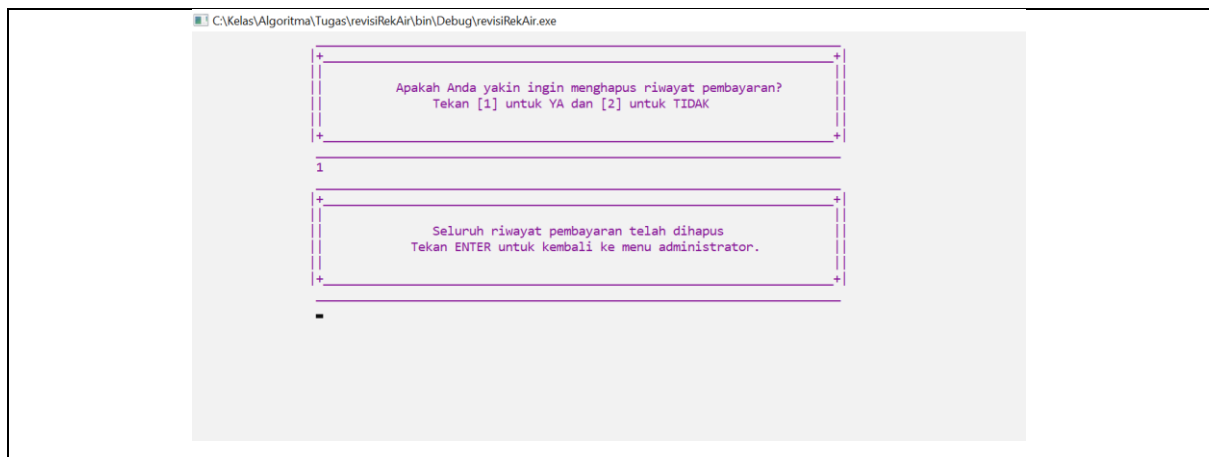
Gambar 32 Menu Administrator (menuAdmin())

3.10 Lihat Riwayat Pembayaran (riwayatStrukPembayaran())



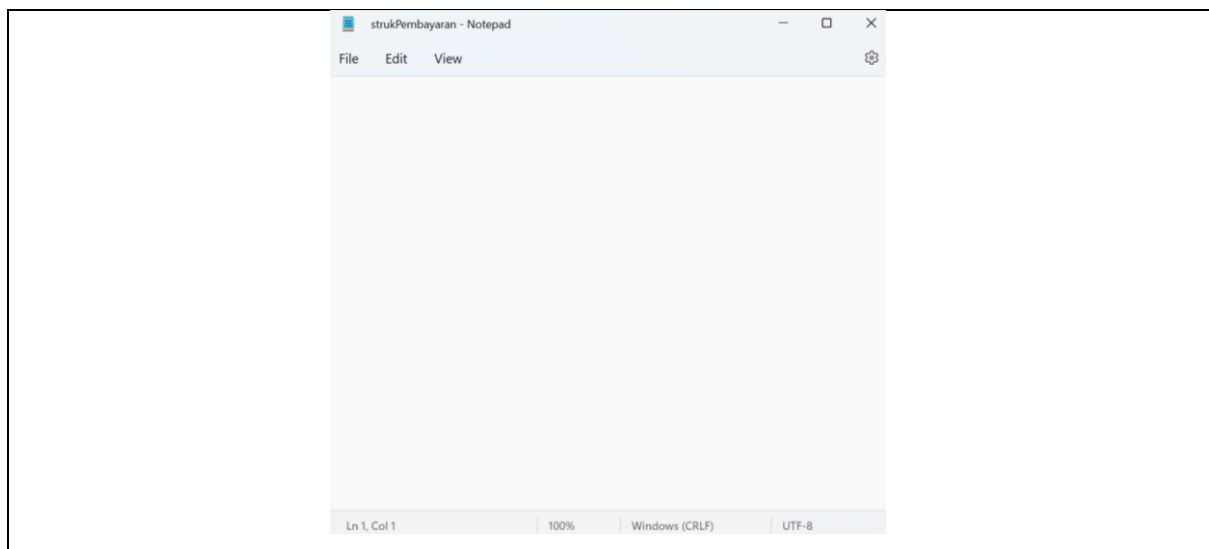
Gambar 33 Tampilan Riwayat Pembayaran (riwayatStrukPembayaran())

3.11 Hapus Riwayat Pembayaran (hapusRiwayatPembayaran())



Gambar 34 Tampilan Hapus Riwayat Pembayaran (hapusRiwayatPembayaran())

3.12 Tampilan File setelah hapusRiwayatPembayaran()



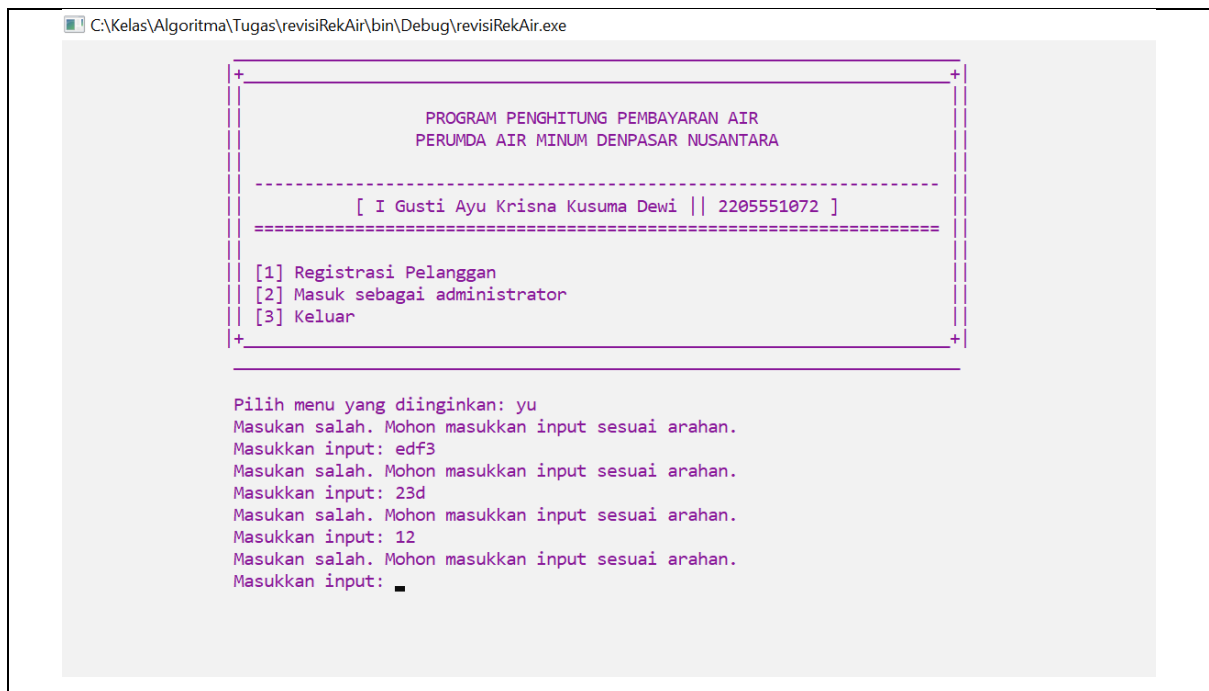
Gambar 35 Tampilan File setelah Isi Dihapus

3.13 Keluar dari Program (keluar())



Gambar 36 Terima Kasih dan Keluar dari Program (keluar())

3.14 Validasi Input



Gambar 37 Validasi Input

DAFTAR PUSTAKA PERATURAN TARIF AIR MINUM

- Admin, 2019. *Tarif Air Minum*. [Online]
Available at:
https://www.perumdmtkr.com/web/layanan_informasi/tarif_air_minum
[Accessed 22 Oktober 2022].
- Anon., 2016. *pdambadung*. [Online]
Available at: <http://pdambadung.id/usrAssets/docs/TARIF%20PDAM%202016.pdf>
[Accessed 22 Oktober 2022].
- Anon., 2022. *Tarif Air Minum Perusahaan Daerah Air Minum*. [Online]
Available at:
https://www.pdam.denpasarkota.go.id/public/uploads/download/download_220407120737_penyesuaian-tarif-perumda-air-minum-tirta-sewakadarma-2022.pdf
[Accessed 22 Oktober 2022].
- Anon., n.d. *BIAYA TAGIHAN PEMAKAIAN AIR*. [Online]
Available at:
<http://download.garuda.kemdikbud.go.id/article.php?article=2072725&val=15539&title=Analisis%20Sistem%20Informasi%20Akuntansi%20Pembayaran%20Biaya%20Tagihan%20Pemakaian%20Air%20Bersih%20Pada%20Perusahaan%20Daerah%20Air%20Minum%20PDAM%20Cabang%20Bengkalis>
[Accessed 22 Oktober 2022].