

**LAPORAN TUGAS BESAR PENGENALAN KOMPUTASI
PROGRAM SISTEM ATM**

Ditujukan untuk memenuhi tugas besar mata kuliah Pengenalan Komputasi KU-
1102

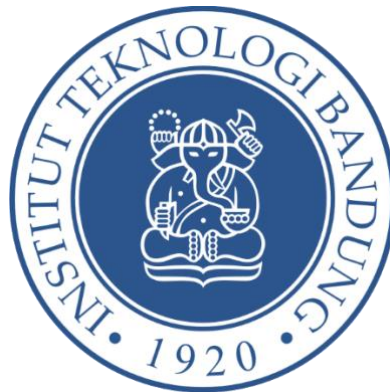
Oleh

16519097 – Reynaldo Averill Adji Putra

16519117 – Allief Nuriman

16519207 – Theodore Jonathan

16519217 – Mohammad Sheva Almeyda Sofjan



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2019**

BAB I

HASIL TUGAS I



Gambar 1.1: Contoh mesin ATM

ATM menurut Ellen Florian (2004) adalah "Alat telekomunikasi berbasis komputer yang menyediakan tempat bagi nasabah dalam melakukan transaksi keuangan tanpa membutuhkan seorang teller bank.", ATM dalam bahasa asing ialah *Automated Teller Machine* dan dalam bahasa Indonesia disebut sebagai Anjungan Tunai Mandiri. Sementara itu definisi ATM menurut Kasmir (2007: 327) "ATM merupakan mesin yang memberikan kemudahan kepada nasabah dalam melakukan transaksi perbankan secara otomatis selama 24 jam dalam 7 hari termasuk hari libur.



Gambar 1.2: Tampilan dari ATM

Secara umum, fungsi dari ATM ialah untuk melakukan penarikan uang tunai, namun masih ada fungsi-fungsi lain yang lebih spesifik dan mempermudah kepentingan nasabah melakukan aktivitas perbankan, contohnya:

- 1) Informasi Saldo
- 2) Pembayaran umum seperti tagihan telepon, kartu kredit, air, listrik, telepon, dan UKT
- 3) Pembelian pulsa
- 4) Transfer antar rekening atau antar bank
- 5) Pengubahan PIN

Dekomposisi persoalan:

1. Layar login
 - a. PIN
 - b. Cek jenis kartu ATM
 - c. Saldo
2. Setor tunai dan Informasi Saldo
 - a. Setor tunai
 - b. Informasi saldo
 - c. Transaksi terakhir
3. Pembayaran
 - a. Listrik
 - b. Air
 - c. Pendidikan
 - d. Pulsa
4. Transfer non tunai
 - a. Antar rekening
 - b. Antar bank
 - c. Virtual account

BAB 2

HASIL TUGAS 2

A. DESKRIPSI SIMULASI

Program ATM TARS memiliki beberapa menu utama, yaitu :

1. Layar login

Layar login merupakan antarmuka sebelum pengguna memasuki menu utama ATM. Di layar login, prosedur yang dilakukan adalah :

- a. Input : Pengguna akan memasukkan jenis kartu ATM dan PIN. Pengguna juga bisa memasukkan data diri baru.
- b. Proses : Program akan memeriksa PIN yang dimasukkan oleh pengguna. . Apabila PIN ditemukan di database, maka pengguna tersebut dapat lanjut ke menu utama. Program juga dapat menyimpan data yang baru dimasukkan ke database.
- c. Output : Saldo akan diambil dari database dan disesuaikan dengan PIN yang dimasukkan.

2. Setor Tunai dan Informasi Akun

Setor Tunai dan Informasi Akun memungkinkan pengguna untuk melakukan setor tunai, mengecek saldo, serta memeriksa tiga transaksi terakhir yang dilakukan pengguna. Prosedur yang dilakukan adalah :

- a. Input : Pengguna akan memilih jenis transaksi
- b. Proses : Jika dipilih menu setor tunai, maka pengguna harus menginput jumlah tunai yang akan disetor.
- c. Output : Jika dipilih menu setor tunai, maka saldo pengguna ditambahkan sesuai dengan jumlah setoran. Jika dipilih menu cek saldo, maka akan ditampilkan saldo pengguna sekarang. Jika dipilih menu cek riwayat transaksi, maka akan ditampilkan tiga transaksi terakhir yang digunakan pengguna.

3. Transfer

Menu Transfer memungkinkan pengguna untuk memindahkan sejumlah uang dari rekening pengguna ke rekening lain. Transfer antarbank dikenakan biaya administrasi Rp6.500,00 sedangkan transfer antarrekening tidak dikenakan biaya administrasi.

- a. Input : Pengguna menginput jenis bank dan nomor rekening tujuan serta menginput jumlah transfer.
- b. Proses : Akan dicek jika jenis bank bukan Bank TARS maka akan dikenakan biaya tambahan Rp6.500,00 dan akan dicek pula apakah saldo yang dimiliki pengguna mencukupi untuk melakukan transfer.
- c. Output : Saldo pengguna akan berkurang sebanyak jumlah transfer, ATM menunjukkan bahwa transaksi sukses.

4. Pembayaran

Menu Pembayaran memungkinkan pengguna untuk membayar tagihan, baik tagihan pendidikan, listrik, air, dan sebagainya. Prinsip dari menu Pembayaran adalah mengurangi saldo pengguna sebanyak nominal yang dibayarkan (mirip dengan transfer)

- a. Input : Pengguna menginput tujuan pembayaran (pendidikan, air, listrik, telepon, dan ke perusahaan mana) dan nominal pembayaran
- b. Proses : Akan dicek apakah saldo yang dimiliki pengguna mencukupi untuk melakukan pembayaran
- c. Output : Saldo pengguna akan berkurang sebanyak jumlah transfer, ATM menunjukkan bahwa transaksi sukses.

B. PSEUDOCODE

Program Simulasi_ATM

{Deskripsi Program}

Program merupakan simulasi dari ATM sebenarnya

KAMUS

{Deklarasi Variabel}

jumlahorang, jeniskartu, regisulang, ModeTransaksi, coba,

indeksorang, saldo, transaksi : integer

pilihan, yakin, setor, tujuantrf, norek, notelpon, jumlahtransfer,

kodekartu, kodebank : integer

atmbarunyala, cobapertamakali, pinadadidatabase, pinangkasemua :
boolean

history : 3d matriks of string and integer

data : matriks of string and integer

{Deklarasi Fungsi}

{Deklarasi Fungsi Registrasi}

Function registrasi(data:matriks)→ matriks

Kamus Lokal

Databaru : array of integer

Algoritma fungsi

input(databaru[0])

input(databaru[1])

While (databaru[1]not_integer) or (jumlah_karakter_data[1]!=6) do
 input(databaru[1])

input(databaru[2])

While (databaru[2]not_integer) do

 input(databaru[2])

Tambah_array_databaru_pada_matriks_data_sebagai_baris_baru

Jumlahorang ← jumlahorang+1

→ data

{Deklarasi Fungsi jejaktransaksi}

Function jejaktransaksi (history:matriks,nominal:integer, jenistransaksi:string) → matriks

Kamus lokal

history : matriks 3d of integer

nominal : int

jenistransaksi : str

Algoritma fungsi

history[indeksorang][2][0] ← history[indeksorang][1][0]

```

history[indeksorang][2][1] ← history[indeksorang][1][1]
history[indeksorang][1][0] ← history[indeksorang][0][0]
history[indeksorang][1][1] ← history[indeksorang][0][1]
history[indeksorang][0][0] ← nominal
history[indeksorang][0][1] ← jenistransaksi
→ history

```

{Deklarasi Fungsi Konfirmasi Pulsa}

Function konfirmasipulsa(saldo:int,history:matriks,nominalpulsa:int, nohp:int) → integer

Kamus lokal

saldo, nominal pulsa, nohp : int
history : 3d matriks of integer and string

Algoritma

```

if saldo>=nominalpulsa then
    Tampilkan_layar_konfirmasi
    input(yakin)
    if yakin==1 then
        saldo ← saldo-nominalpulsa
        jejaktransaksi(history,nominalpulsa,'Pembayaran
Pulsa')
    else then
        output('Tidak_valid')
else then
    output('saldo_tidak_cukup')
→ saldo

```

{Deklarasi Fungsi Pulsa}

Function pulsa(saldo:int,history:matriks,namakartu:str) → integer

Kamus lokal

saldo, kodepulsa : int
history : 3d matriks of integer and string
namakartu: string

Algoritma Fungsi

```

Tampilkan_pilihan_nominal
input(kodepulsa,nohp)
i traversal [0...3]
    if i+1 == kodepulsa then
        saldo=konfirmasipulsa(saldo,history,(i+1)*50000,nohp)
→ saldo

```

ALGORITMA UTAMA

{Layar Login}

```

Tampilkan_layar_login
input(jeniskartu)
If jeniskartu==1 then
    registrasi(data)
    jumlahorang ← jumlahorang+1
    Perluas_array_history
Else if jeniskartu==2 then
    regisulang ← 1
    while regisulang == 1 do
        registrasi(data)
        input(regisulang)
Else if jeniskartu==3 then
    jeniskartu ← 1

```

```

Else if jeniskartu==16519 then
    Hentikan_program
cobapertamakali ← True
Modetransaksi ← 1
While Modetransaksi ← 1 do
    coba ← 0
    pinadadidatabase ← False
    pinangkasemua ← False
    while (coba<3)and((pinangkasemua!=True) or
    (pinadadidatabase!=True)) do
        input(pin)
        pinadadidatabase ← False
        pinangkasemua ← False
        if (semua_pin_adalah_integer) and
        (jumlah_digit_pin_adalah_6)then
            pinangkasemua ← True
            If cobapertamakali==True then
                i traversal [0...jumlahorang-1]
                if(data[i][1]==pin) then
                    pinadadidatabase ← True
                    indeksorang ← i
            Else then
                If (data[indeksorang][1]==pin) then
                    pinadadidatabase ← True
            if(pinangkasemua == False)and(coba<2)then
                output('PIN salah')
            elif(pinadadidatabase ==False)and(coba<2)then
                output('PIN harus 6 angka')
            coba ← coba+1
        if (coba==3)and((pinangkasemua!=True)and((pinadadidatabase!=True)
        e) then
            output('PIN sudah dimasukkan sebanyak 3 kali')
            Hentikan_program

saldo ← data[indeksorang][2]
input(transaksi)

{Setor Tunai dan Informasi Akun}
if transaksi==1 then
    Tampilkan_Pilihan_Transaksi
    input(pilihan)
    {Setor Tunai}
    if (pilihan == 1)then
        input(setor)
        Konfirmasi_Terakhir
        input(yakin)
        if (yakin==1)then
            saldo ← saldo+setor
            jejaktransaksi(history, setor, 'setor
tunai')
        {Informasi Saldo}
    Else if pilihan == 2 then
        output(saldo)
    {Transaksi Terakhir}
    Else if pilihan == 3 then

```

```

        output(history)

{Pembayaran listik, air, pendidikan, telepon}
else if transaksi==2 then
    Tampilkan_Tujuan_Transfer
    input(tujuantrf)
    norek ← 0
    notelp ← 0
    jumlahtransfer ← 0

{Pembayaran Listrik}
if tujuantrf == 1 then
    input(norek)
    input(notelp)
    kirim_tagihan_ke_gawai
    input(jumlahtransfer)
    Tampilkan_layar_konfirmasi
    input(yakin)
    if yakin == 1 then
        if jumlahtransfer>saldo then
            output('Saldo_tidak_cukup')
        else then
            output('Transfer sukses')
            saldo ← saldo - jumlahtransfer

jejaktransaksi(history, setor, 'Pembayaran Listrik')
    else then
        ouput('Masukkan_Tidak_Valid')

{Pembayaran PDAM}
else if tujuantrf ==2 then
    Tampilkan_kode_PDAM
    input(norek)
    kirim_tagihan_ke_gawai
    input(jumlahtransfer)
    Tampilkan_layar_konfirmasi
    input(yakin)
    if yakin == 1 then
        if jumlahtransfer>saldo then
            output('Saldo_tidak_cukup')
        else then
            output('Transfer sukses')
            saldo ← saldo - jumlahtransfer

jejaktransaksi(history, setor, 'Pembayaran PDAM')
    else then
        output('Masukkan_Tidak_Valid')

{Pembayaran Pendidikan}
else if tujuantrf == 3 then
    Tampilkan_kode_PTN
    input(norek)
    input(jumlahtransfer)
    Tampilkan_layar_konfirmasi
    input(yakin)
    if yakin == 1 then

```



```

        if jumlahtransfer > saldo then
            output('Saldo_tidak_cukup')
        else then
            output('Transfer sukses')
            saldo ← saldo - jumlahtransfer

jejaktransaksi(history, setor, 'Pembayaran Pendidikan')
    else then
        output('Masukkan_Tidak_Valid')

{Pembayaran Pulsa}
else if tujuantrf == 4 then
    Tampilkan_Pilihan_Operator_Pulsa
    input(kodekartu)
    if kodekartu == 1 then
        saldo ← pulsa(saldo, history, 'XL')
    Else if kodekartu == 2 then
        saldo ← pulsa(saldo, history, 'Telkomsel')
    Else if kodekartu == 3 then
        saldo ← pulsa(saldo, history, 'Indosat')
    Else if kodekartu == 4 then
        saldo ← pulsa(saldo, history, 'Smartfren')

{Transfer antar rekening, antar bank, dan virtual account}
else if transaksi == 3 then
    Tampilkan_menu_transfer
    input(tujuantrf)
    norek ← 0
    jumlahtransfer ← 0
    {Transfer sesama bank}
    if tujuantrf == 1 then
        input(norek)
        if norek < 10000000000 then
            input(jumlahtransfer)
            Tampilkan_layar_konfirmasi
            input(yakin)
            if yakin == 1 then
                if jumlahtransfer > saldo then
                    output('Saldo tidak cukup')
                else then
                    output('Transfer berhasil')
                    saldo ← saldo - jumlahtransfer
                    jejaktransaksi(history, setor, 'Transfer Sesama Bank')

            else then
                output('Masukan tidak valid')
        {Transfer antar bank}
    else if tujuantrf == 2 then
        Tampilkan_kode_bank
        input(kode_bank)
        if 100 < kodebank < 110 then
            input(norek)
            if norek < 10000000000000000 then
                Tampilkan_peringatan_biaya_admin
                input(jumlahtransfer)

```

```

Tampilkan_layar_konfirmasi
input(yakin)
if yakin == 1 then
    if jumlahtransfer > saldo then
        output('Saldo tidak
cukup')
    else then
        output('Transfer
berhasil)

        saldo ← saldo - jumlahtransfer -
        4000
        jejaktransaksi(history, s
        etor, 'Transfer Antar Bank')
    else then
        output('Masukan tidak valid')
{Transfer virtual account}
else if tujuantrf == 3 then
    input(norek)
    if norek < 10**16 then
        input(jumlahtransfer)
        Tampilkan_layar_konfirmasi
        input(yakin)
        if yakin == 1 then
            if jumlahtransfer > saldo then
                output('Saldo tidak cukup')
            else then
                output('Transfer berhasil')
                saldo ← saldo - jumlahtransfer
                jejaktransaksi(history, setor,
'Virtual Account')
            else then ('Masukan tidak valid')
        else then
            output('Masukan tidak valid')
            Tampilkan_konfirmasi_melakukan_transaksi_lain
            input(ModeTransaksi)
            data[indeksorang] ← saldo
            cobapertamakali ← False
if jeniskartu == 1 then
    saldo ← saldo - 6500
    jejaktransaksi(history, setor, 'Biaya Administrasi')
data[indeksorang][2] ← saldo
atmbarunya ← False
output('Transaksi Selesai')
ATM_kembali_berjalan_dari_awal

```

BAB 3

HASIL TUGAS 3

Pada tugas 3, dekomposisi pada tugas 1 serta pseudocode pada tugas 2 akan diimplementasikan menjadi suatu program yang diketik dengan bahasa python. Program ATM TARS yang dibuat menggunakan fungsi - fungsi sederhana

1. Kondisional (IF)

Program ATM ini menggunakan banyak kondisional, karena penggunaan kondisional memungkinkan pengguna mengeluarkan input yang berbeda untuk masing-masing persyaratan. Beberapa kondisional yang digunakan pada simulasi mesin ATM ini adalah sebagai berikut.

- a. Penggunaan kondisional pada menu utama saat pengguna pertama kali memasukkan kartu ATM. Adanya kondisional memisahkan 3 kasus. Pertama, jika kartu ATM adalah TARS. Kedua, jika kartu ATM adalah kartu ATM bank lain, namun belum mengisi database. Terakhir, jika kartu ATM adalah TARS dan ingin registrasi data baru.

```
ATM Bank TARS
Tekan 0 jika kartu ATM anda adalah kartu ATM TARS
Tekan 1 jika kartu ATM anda berasal dari bank lain dan anda belum mengisi database di ATM ini
Tekan 2 jika anda ingin registrasi data baru, dan kartu ATM anda adalah kartu ATM TARS
```

- b. Penggunaan kondisional pada menu pilihan transaksi. Menu ini ditampilkan setelah pengguna memilih kartu ATM yang digunakan, dan telah memasukkan PIN. Kondisional memisahkan 3 transaksi yang mungkin dilakukan di ATM, yakni setoran tunai dan informasi saldo, pembayaran, dan transfer.

```
Pilihan transaksi :
1. Setoran tunai dan Informasi Rekening
2. Pembayaran
3. Transfer
Masukkan pilihan transaksi : █
```

- c. Penggunaan kondisional pada menu setoran tunai dan informasi rekening. Kondisional memisahkan 3 kemungkinan yang bisa dilakukan oleh ATM, yakni setor tunai, informasi saldo, dan transaksi terakhir. Setor tunai memungkinkan pengguna untuk melakukan penyetoran uang melalui mesin ATM. Informasi saldo memungkinkan pengguna untuk mengecek saldo yang tersisa. Transaksi terakhir menampilkan tiga transaksi terakhir yang dilakukan oleh pengguna.

```
Pilihan transaksi :
1. Setoran Tunai
2. Informasi Saldo
3. Transaksi Terakhir
Pilihan :
```

- d. Penggunaan kondisional pada menu pembayaran. Kondisional memisahkan empat jenis pembayaran yang mungkin dilakukan oleh ATM. Pembayaran yang mungkin adalah listrik, air, pendidikan, dan pulsa.

```

1. Listrik
2. Air
3. Pendidikan
4. Pulsa
Masukkan kode tujuan transfer :

```

- e. Penggunaan kondisional pada menu pembayaran pulsa. Kondisional memisahkan empat operator seluler yang pembayarannya dapat dilakukan melalui ATM Bank TARS. Keempat operator seluler tersebut adalah XL, Telkomsel, Indosat, dan Smartfren.
- f. Penggunaan kondisional pada menu transfer. Kondisional memisahkan tiga jenis transfer yang mungkin dilakukan pada ATM

```

Pilih kartu yang ingin Anda isi pulsanya
1. XL
2. Telkomsel
3. Indosat
4. Smartfren

```

Bank TARS, yaitu transfer antar rekening, transfer antar bank, dan transfer virtual account. Transfer antar rekening merupakan transfer yang digunakan untuk mengirim uang kepada sesama rekening Bank TARS. Transfer antar rekening digunakan untuk mengirim uang kepada rekening bank lain dan akan dikenakan biaya admin sebesar Rp6.000,00,-.

```

MENU TRANSFER
Silakan pilih tujuan transfer Anda.
1. Transfer Antar Rekening
2. Transfer Antar Bank
3. Transfer Virtual Account
Masukkan kode tujuan transfer : █

```

2. Loop (WHILE)

Looping merupakan suatu perintah dimana pengulangan akan terus dilakukan sampai kondisi di awal loop tidak terpenuhi. Terdapat empat looping yang digunakan pada algoritma utama program ATM ini, yaitu :

a. Looping utama ATM

Looping utama ATM memungkinkan ATM terus berjalan meskipun pengguna sudah menyelesaikan transaksinya. ATM dapat terus digunakan sehingga database yang tersimpan tidak terhapus. Karena looping yang digunakan adalah `while True`, maka looping akan terus berjalan. Looping hanya akan berhenti jika ditemukan perintah `exit()` pada program. Looping akan berhenti saat pengguna memasukkan kode "16519" di layar login utama ATM, atau saat pengguna salah memasukkan PIN sebanyak 3 kali.

```

116 # Layar login awal
117 while True:
118     print('ATM Bank TARS')

```

gambar penggunaan looping

```

154 | #Menghentikan program ATM
155 | elif(jeniskartu==16519):
156 |     print('Program ATM akan dihentikan')
157 |     exit()
158 |
188 |
189 | if((coba==3)and(pinangkasemua!=True)and(pin
    adadidatabase!=True)):
190 |     print('Anda telah memasukkan pin
    sebanyak 3 kali. Kartu ATM anda akan
    diblokir')
191 |     print('Silahkan datang ke kantor cabang
    Bank TARS terdekat di daerah anda')
192 |     exit()

```

Looping berhenti

b. Looping pengulangan transaksi

Looping pengulangan transaksi memungkinkan pengguna untuk melakukan transaksi lainnya setelah melakukan suatu transaksi, tanpa harus mengeluarkan kartu ATM dari mesin. Namun, untuk faktor keamanan pengguna harus tetap memasukkan PIN sebelum transaksi. Looping yang digunakan adalah `while ModeTransaksi == 1`. Selama pengguna ingin melakukan transaksi, maka looping akan terus berjalan.

```

160 | ModeTransaksi = 1
161 | while ModeTransaksi == 1 :
162 |     # PIN kartu
163 |     print('Silakan masukkan PIN anda')
164 |     print('Jagalah kerahasiaan PIN anda')

```

c. Looping PIN ATM

Looping PIN ATM memungkinkan pengguna untuk melakukan pengulangan apabila salah memasukkan PIN ATM. Looping akan terus berjalan jika PIN tidak terdapat di database, PIN tidak terdiri dari kombinasi 6 angka, atau pengguna belum memasukkan PIN sebanyak tiga kali. Looping akan berhenti ketika pengguna sudah salah mencoba PIN sebanyak 3 kali.

```

168 |
169 | while((coba<3)and((pinangkasemua!=True)or(p
    inadadidatabase!=True))):
170 |     pin=input()

```

d. Looping registrasi data

Looping registrasi data memungkinkan pengguna untuk melakukan registrasi data kembali setelah melakukan registrasi data sebelumnya.

Looping akan terus berjalan jika pengguna ingin melakukan registrasi data (regisulang==1)

```
143 | regisulang=1
144 | while(regisulang==1):
145 |     print('Silahkan lengkapi data berikut')
```

3. Array

Array memungkinkan ATM TARS untuk menyimpan database dari masing-masing pengguna. Terdapat array yang digunakan pada mesin ATM ini, yaitu :

- a. Array database username, PIN, dan Saldo

Program ATM TARS menggunakan array dua dimensi (matriks) untuk menyimpan data pengguna. Baris menyatakan data dari setiap pengguna. Kolom pertama berisi nama dari pengguna. Kolom kedua berisi PIN pengguna. Kolom ketiga berisi saldo pengguna.

```
Theodore Jonathan 519207 8000000
Reynaldo Averill 519097 9000000
Allief Nuriman 519117 8500000
Mohammad Sheva 519217 9500000
```

- b. Array history

Program ATM TARS menggunakan array tiga dimensi untuk menyimpan tiga transaksi terakhir yang dilakukan oleh pengguna. Array berukuran jumlah orang x 3 x 2. Untuk melihat isi dari array history, pengguna dapat login dengan PIN yang dimiliki, kemudian menggunakan menu "Transaksi Terakhir".

```
Tiga Transaksi Terakhir Anda:
| Nominal | Jenis Transaksi |
| 2500000 | Pembayaran Pendidikan |
| 1000000 | Pembayaran Pulsa |
| 590000 | Setor Tunai |
```

4. Fungsi

Fungsi memungkinkan ATM TARS untuk melakukan beberapa prosedur yang mirip, tanpa harus mengetik ulang *source code*. Terdapat empat fungsi yang dideklarasikan pada ATM TARS, yaitu :

- a. Fungsi registrasi

Fungsi registrasi digunakan untuk menambahkan data baru pada database. Fungsi registrasi menerima input berupa matriks. Kemudian, fungsi registrasi akan menambahkan data baru pada database. Terakhir, fungsi

memberikan kembaliant berupa matriks yang sudah berisi data pengguna baru.

b. Fungsi jejaktransaksi

Fungsi jejaktransaksi menerima tiga input, yaitu history (array tiga

```
16 # Deklarasi Fungsi Registrasi
17 def registrasi (data):
18     #Registrasi digunakan untuk menambahkan
    user baru pada database
19     #KAMUS LOKAL
20     #databaru : array of integer
21     #ALGORITMA FUNGSI
22     databaru=[0 for i in range (3)]
23     databaru[0]=input('Masukkan nama: ')
24     databaru[1]=input('Masukkan PIN anda
    (Terdiri dari 6 digit angka): ')
25     while((databaru[1].
    isdigit()==False)or(len(databaru[1])!=6)):
26         print('PIN harus terdiri dari 6 digit angka')
27         databaru[1]=input('Masukkan ulang PIN
    anda: ')
28     databaru[2]=input('Masukkan saldo: ')
29     while(databaru[2].isdigit()==False):
30         print('Saldo harus berupa bilangan bulat')
31         databaru[2]=input('Masukkan setoran
    awal anda: ')
32     data.append(databaru)
33     return data
```

dimensi), nominal (integer), dan jenistransaksi (string). Fungsi akan memasukkan transaksi terakhir yang dilakukan oleh pengguna ke array history. Terakhir, fungsi akan memberika kembalian berupa array tiga dimensi (array history yang telah berisi transaksi terbaru). Fungsi ini dibuat agar *source code* untuk mencatat transaksi tidak perlu diketik ulang di setiap transaksinya, namun cukup memanggil fungsi jejaktransaksi saja.

```

34 #Deklarasi Fungsi jejaktransaksi
35 def jejaktransaksi(history,nominal,
jenistransaksi):
36     #jejaktransaksi digunakan untuk mencatat
transaksi yang dilakukan suatu akun
37     #KAMUS LOKAL
38     #history : matriks 3d of integer
39     #nominal : int
40     #jenistransaksi : str
41     #ALGORITMA FUNGSI
42
history[indeksorang][2][0]=history[indeksorang]
[1][0]
43
history[indeksorang][2][1]=history[indeksorang]
[1][1]
44
history[indeksorang][1][0]=history[indeksorang]
[0][0]
45
history[indeksorang][1][1]=history[indeksorang]
[0][1]
46     history[indeksorang][0][0]=nominal
47     history[indeksorang][0][1]=jenistransaksi
48     return history

```

c. Fungsi konfirmasipulsa

Fungsi konfirmasipulsa akan menampilkan layar konfirmasi saat pengguna melakukan pembelian pulsa. Fungsi ini menerima empat input, yaitu saldo (integer), history (array 3d), nominal pulsa (integer), dan nohp (integer). Setelah menerima input, fungsi akan menampilkan layar konfirmasi yang disesuaikan dengan nominal pulsa dan no HP yang dimasukkan oleh pengguna. Terakhir, fungsi akan memberikan kembalian variabel saldo (return).


```

49 #Deklarasi Fungsi Konfirmasi Pulsa
50 def konfirmasipulsa(saldo,history,
    nominalpulsa,nohp) :
51     #Program menampilkan layar konfirmasi
    saat melakukan pembelian pulsa
52     #KAMUS LOKAL
53     #saldo, nominal pulsa, nohp : int
54     #history : 3d matriks of integer and string
55     #ALGORITMA FUNGSI
56     if saldo >= nominalpulsa:
57         print('Anda yakin ingin isi pulsa',
    nominalpulsa,'ke nomor',nohp,?)
58         print('0. Tidak')
59         print('1. Ya')
60         yakin=int(input())
61         if yakin == 1 :
62             print('Isi ulang berhasil.')
63             saldo=saldo-nominalpulsa
64             jejaktransaksi(history,nominalpulsa,
    "Pembayaran Pulsa")
65         else :
66             print('Masukan tidak valid.')
67     else:
68         print('Saldo tidak cukup')
69     return saldo

```

d. Fungsi pulsa

Fungsi pulsa menampilkan menu pilihan nominal pulsa. Fungsi ini menerima tiga input, yaitu saldo (integer), history (array 3 dimensi), dan namakartu (string). Setelah menerima input, program akan menampilkan layar pilihan nominal pulsa sesuai dengan jenis operator masing-masing. Terakhir, fungsi akan memberikan kembalian berupa saldo (integer).

```

70 #Deklarasi Fungsi Pulsa
71 def pulsa(saldo,history,namakartu):
72     #Program menampilkan menu pilihan
    pengisian pulsa
73     #KAMUS LOKAL
74     #saldo, kodepulsa : int
75     #history : 3d matriks of integer and string
76     #namakartu: string
77     #ALGORITMA FUNGSI
78     print('Anda akan mengisi pulsa kartu',
    namakartu)
79     print('Pilih nominal yang anda inginkan')
80     print('1. 50.000')
81     print('2. 100.000')
82     print('3. 150.000')
83     print('4. 200.000')
84     kodepulsa=int(input())
85     print('Masukkan nomor HP Anda')
86     nohp=int(input())
87     for i in range (4):
88         if i+1==kodepulsa :
89             saldo=konfirmasipulsa(saldo,history,((i+
    1)*50000),nohp)
90     return saldo

```

BAB 4

KESIMPULAN

A. Kesimpulan

Program pembuatan ATM adalah contoh yang baik dalam berlatih pemrograman karena dalam program ATM, khususnya di ATM TARS, membutuhkan dasar pemrograman yang kuat, khususnya pada bagian deklarasi dan pemanggilan fungsi, kondisional, dan *loop* yang merupakan operasi-operasi dasar yang bisa ditemukan di sebuah ATM.

Tentu saja dalam ATM di dunia nyata program yang digunakan tidak sesederhana yang dibuat pada tugas ini, dibutuhkan sistem yang lebih kompleks lagi.

B. Lesson Learned

Dalam program ATM dibutuhkan penggunaan fungsi untuk berbagai transaksi yang berbeda dalam ATM. Selain itu juga sangat diperlukan penggunaan kondisional (if) karena untuk transaksi yang berbeda-beda perlu dilakukan operasi yang berbeda-beda juga.

Karena transaksi yang bisa dilakukan di ATM sangat banyak maka diperlukan program yang sangat panjang dan cukup kompleks juga, sehingga dalam program ATM ini dibutuhkan kerapian dalam penulisan program supaya memudahkan dalam proses *debugging*.

C. Pembagian Tugas

- | | |
|------------------------------------|----------------------|
| 1. Layar login | - Reynaldo |
| a. PIN | |
| b. Cek jenis kartu ATM | |
| c. Saldo | |
| 2. Setor Tunai dan Informasi Saldo | - Sheva |
| a. Setor tunai | |
| b. Informasi saldo | |
| c. Transaksi terakhir | |
| 3. Pembayaran | - Allief |
| a. Listrik | |
| b. Air | |
| c. Pendidikan | |
| d. Pulsa | |
| 4. Transfer nontunai | - Theodore |
| a. Antarrekening | |
| b. Antarbank | |
| 5. Powerpoint | - Theodore |
| 6. Laporan Tugas 1 | - Allief, Sheva |
| 7. Laporan Tugas 2 | - Theodore, Reynaldo |
| 8. Laporan Tugas 3 | - Reynaldo, Sheva |

9. Kesimpulan dan Lesson Learned - Theodore