

**LAPORAN PRAKTIKUM
POSTTEST (8)
ALGORITMA PEMROGRAMAN DASAR**



Disusun oleh:
Ahmad Rafi' Irsyad Nugraha
(2509106034)
Kelas (A2 '25)

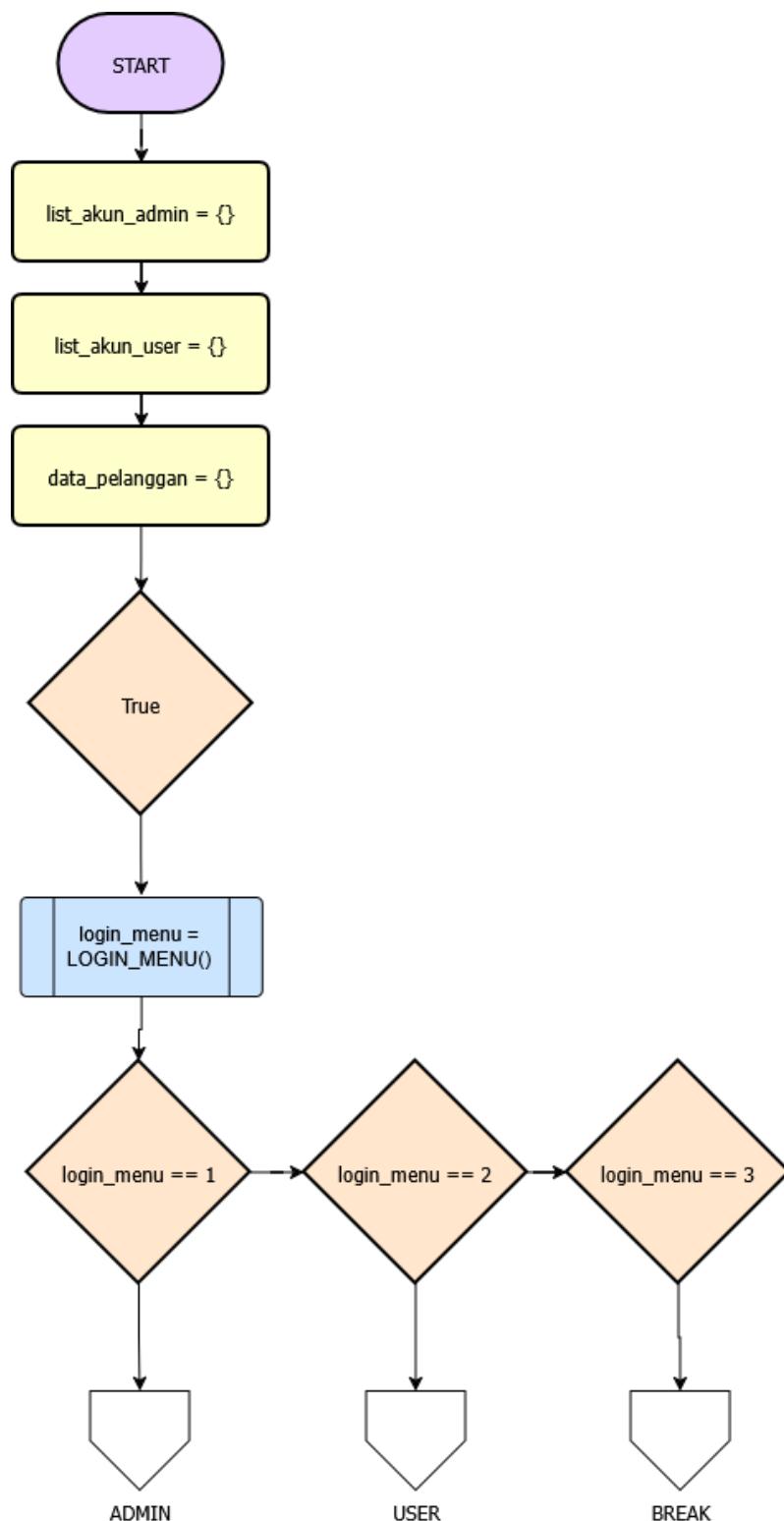
**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA**

2025

1. Flowchart

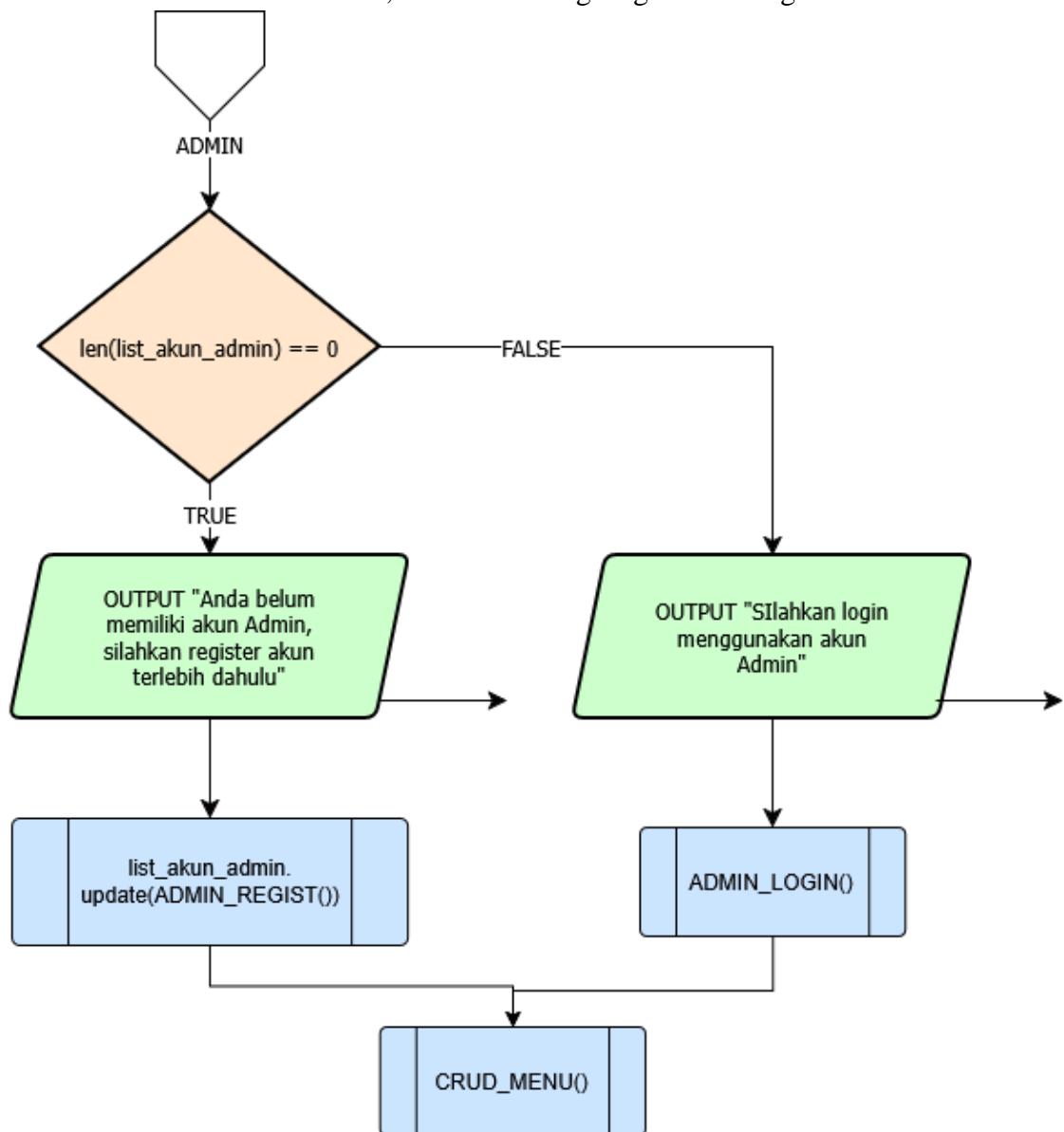
1.1. Main Menu

Pada menu utama, program menyimpan beberapa variabel Global dan memanggil fungsi paling awal program, yaitu fungsi LOGIN_MENU yang berisi library Inquirer untuk user dapat memilih sebagai apa login di program



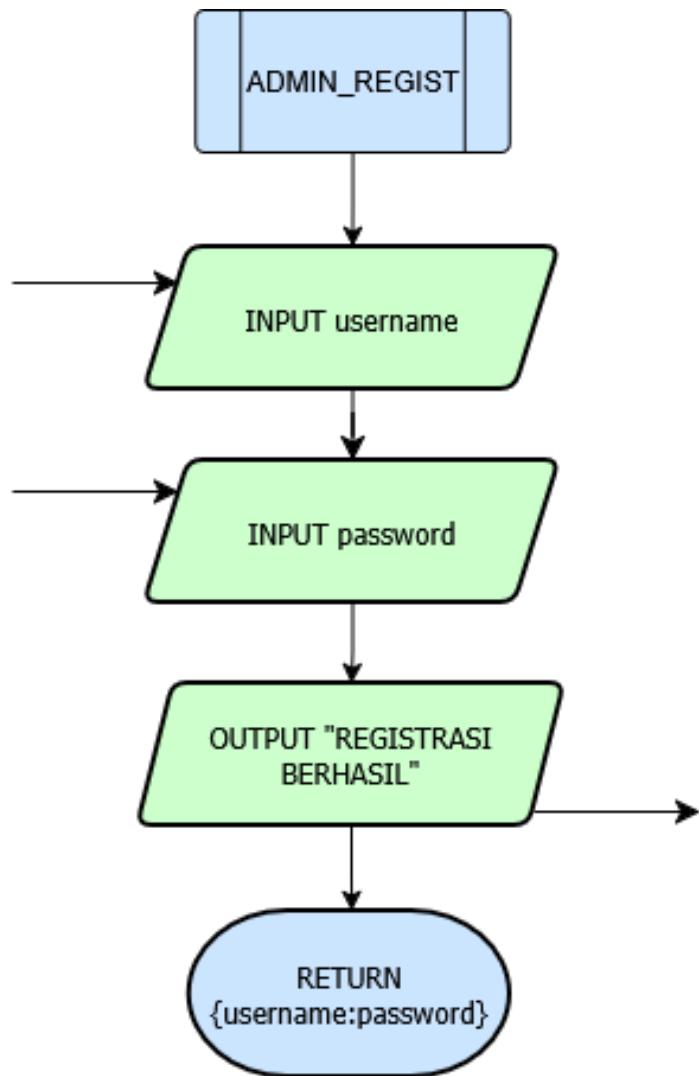
1.2. Admin

Jika user memilih sebagai Admin, maka pertama-tama akan dicek apakah sudah ada akun admin dari list_akun_admin. Jika tidak maka user akan diminta untuk register terlebih dahulu. Jika sudah ada, maka akan langsung diminta login



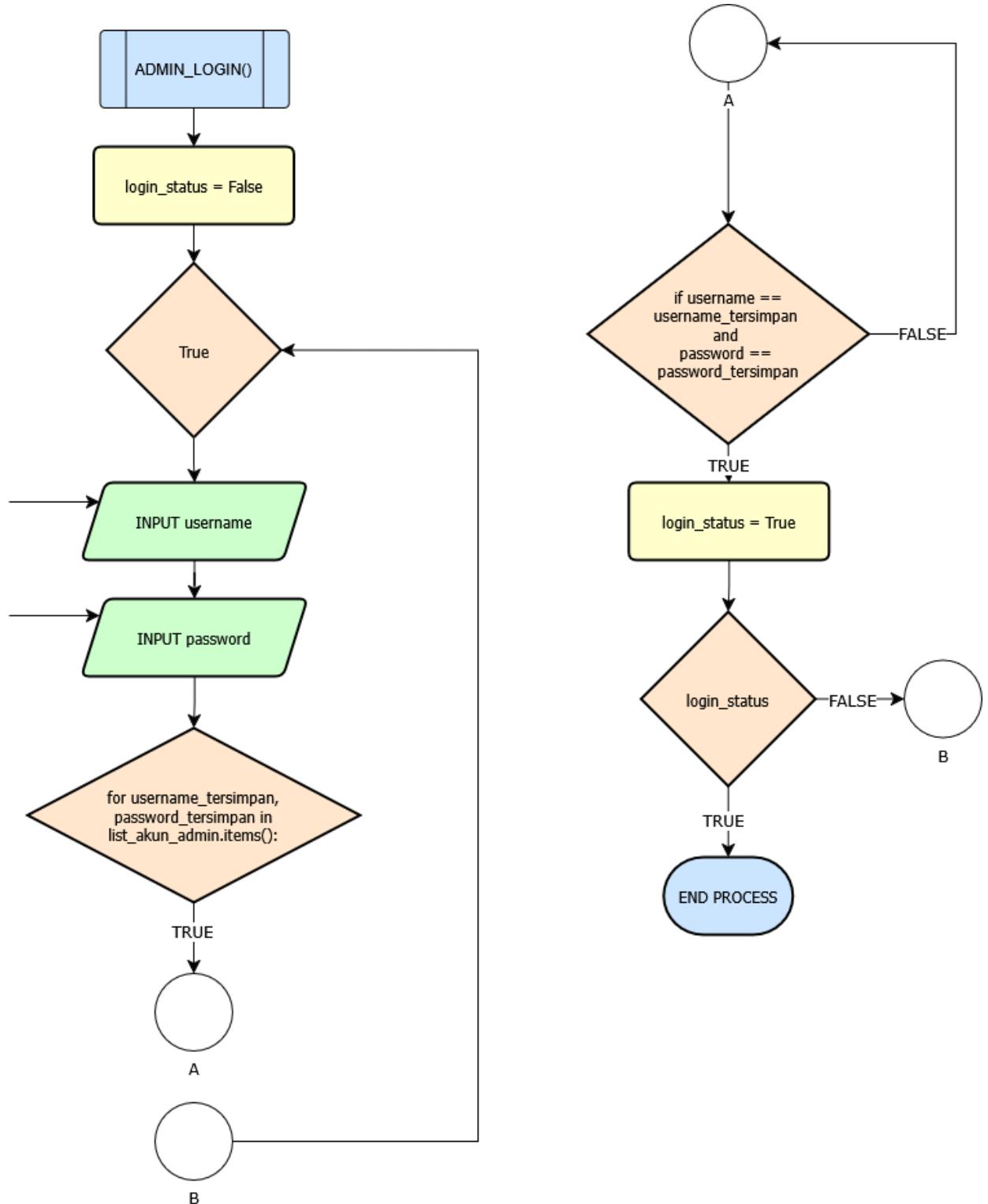
1.3. Admin Regist

Pada Bagian ini, User diminta untuk membuat akun admin baru dengan menginputkan username dan password yang kemudian kedua variabel tersebut diupdate dalam bentuk dictionary. Kemudian regist dinyatakan berhasil



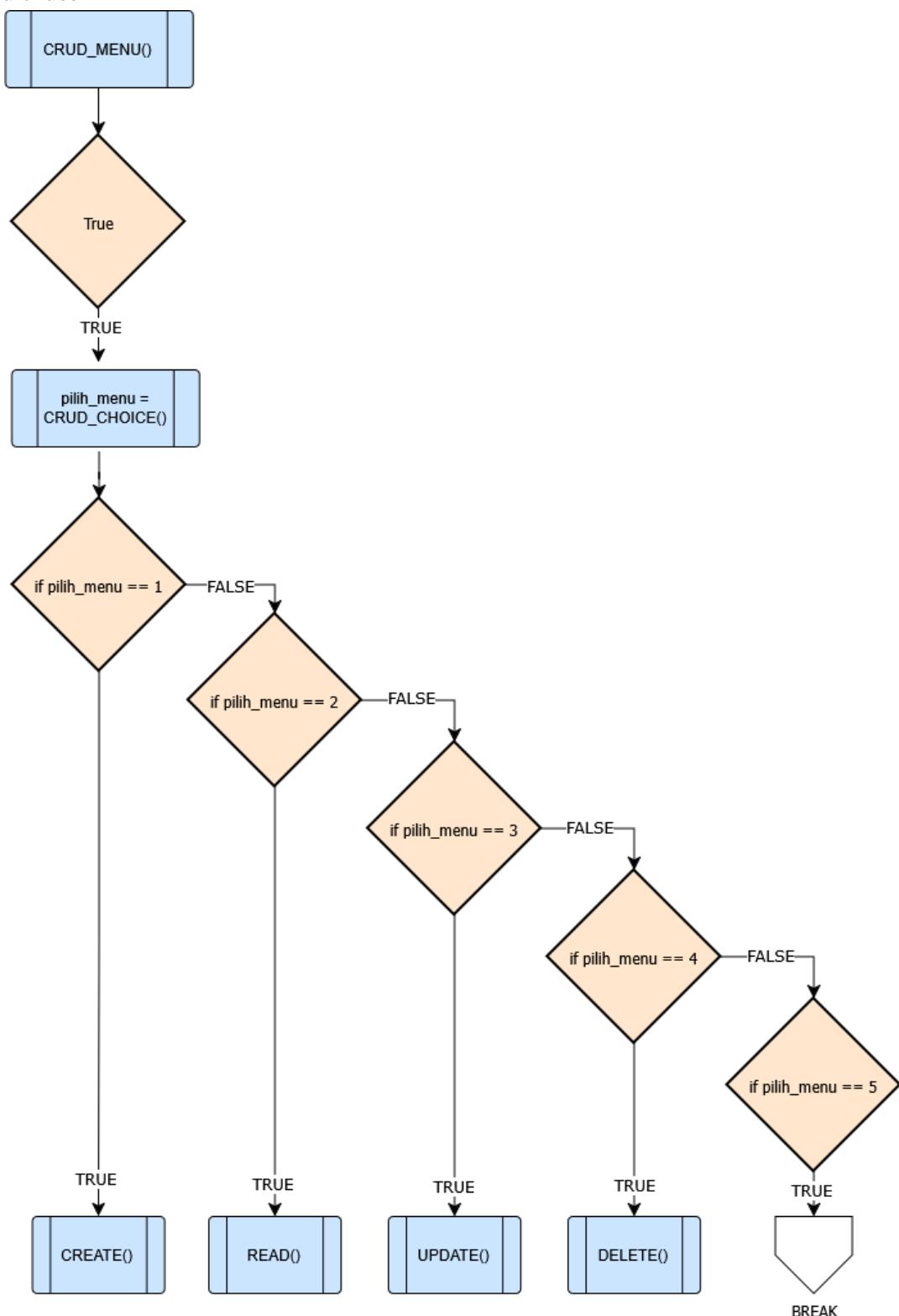
1.4. Admin Login

Pertama-tama login_status diset False. Lalu user diminta untuk menginput username dan password, lalu dicocokan satu persatu dengan data yang ada di list_akun_admin menggunakan for loop. Jika ada yg cocok maka status_login diset True dan login dinyatakan berhasil. Jika tidak maka user diminta untuk login ulang



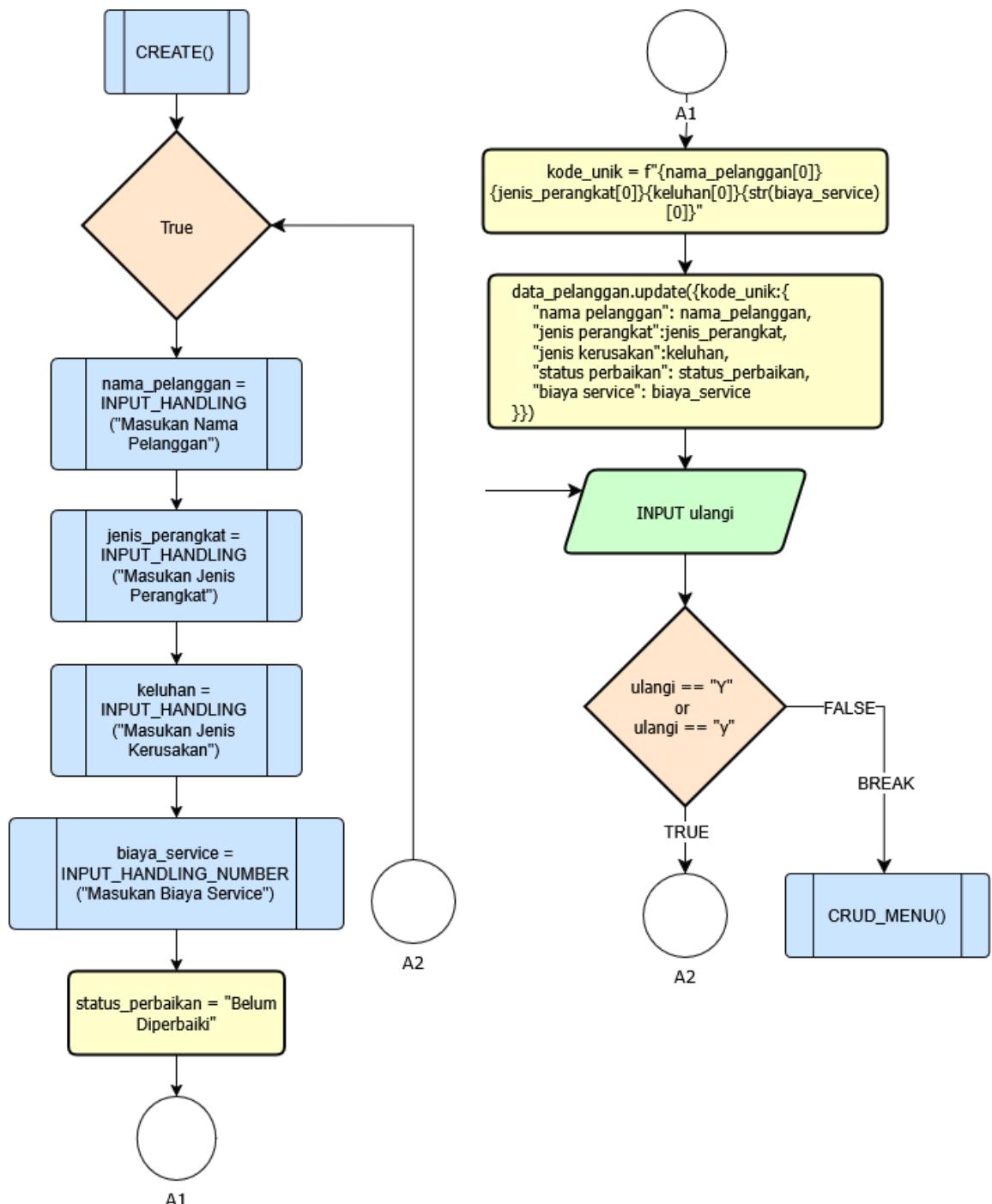
1.5. CRUD Menu

Pada Bagian ini, user yang sudah terdefinisi sebagai Admin, dapat memilih Menu CRUD yang ada menggunakan fungsi `CRUD_CHOICE()` yang menggunakan library `inquirer`, user dapat memilih menu CREATE, READ, UPDATE, DELETE. Atau memilih keluar, dan akan *break* ke bagian login multi user



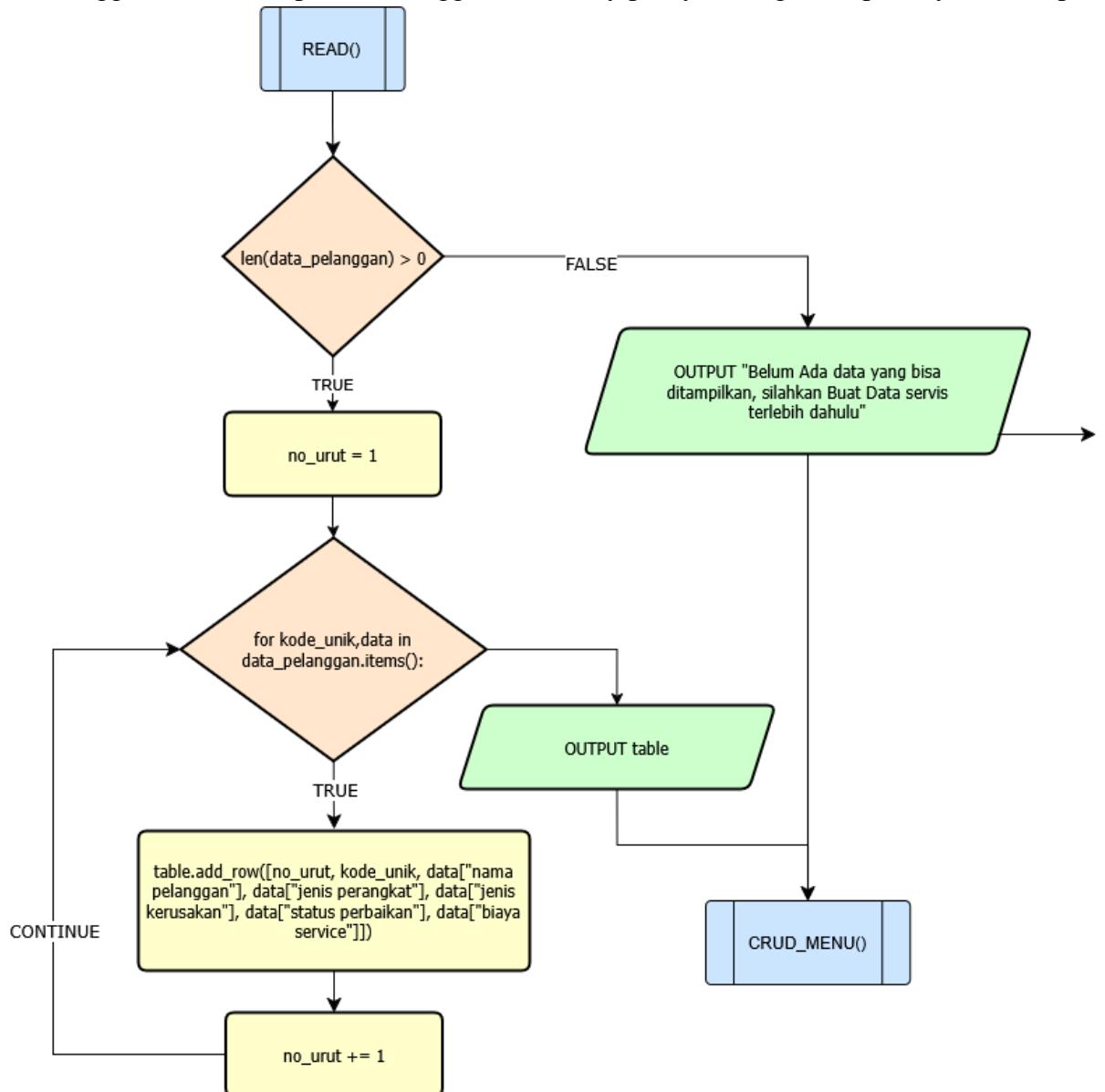
1.6. Menu Tambah Data Servis (CREATE)

Pada bagian ini. User siminta untuk menginput nama_pelanggan, jenis_perangkat,jenis_kerusakan dan biaya service. Serta membuat kode unik dengan menggabungkan huruf awal dari nama,jenis perangkat, jenis kerusakan dan angka awal dari harga. Lalu data-data tersebut diupdate dalam bentuk array sehingga nanti bentuknya akan menjadi nested array.



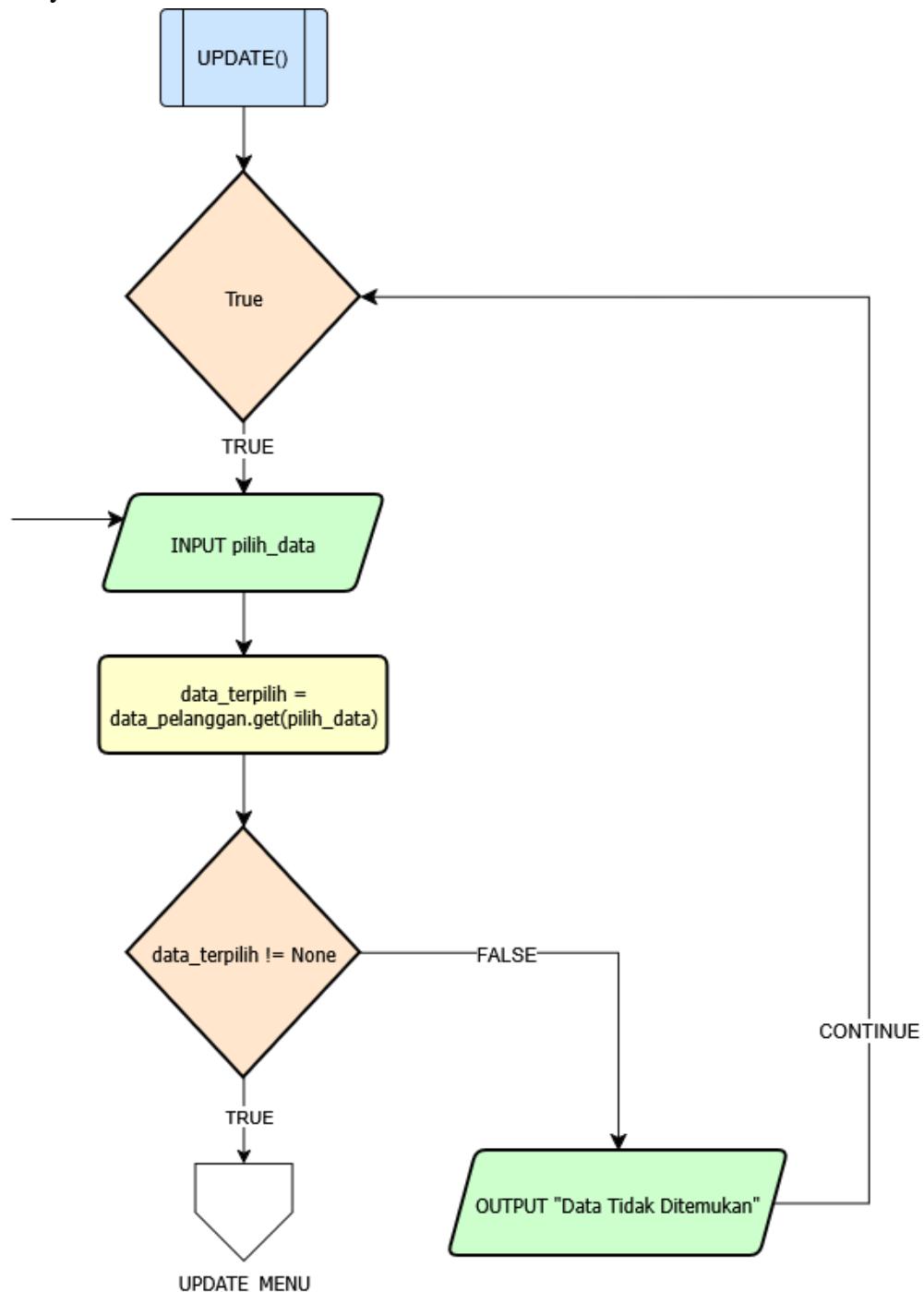
1.7. Menu Menampilkan Data Servis (READ)

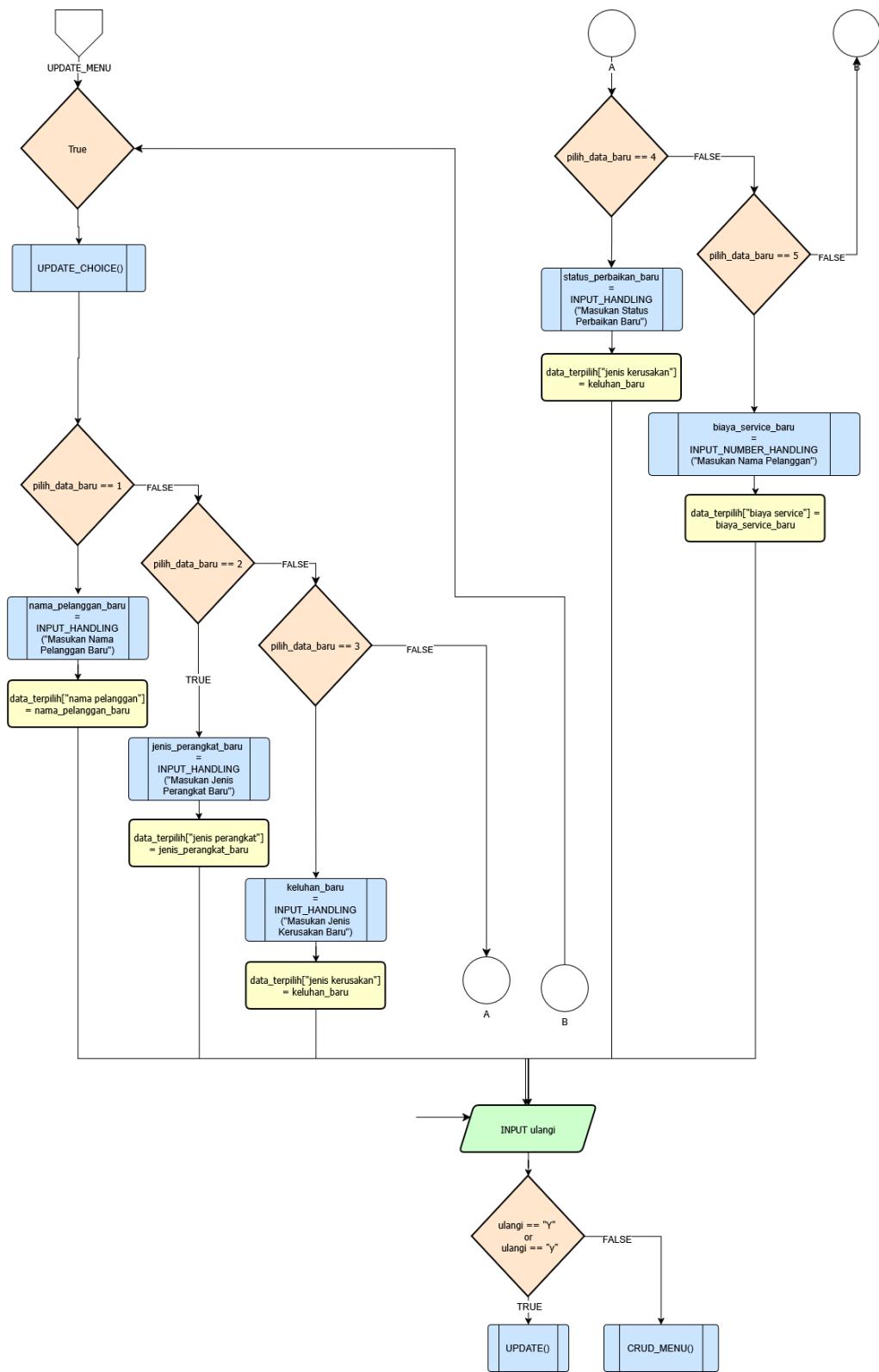
Pada bagian ini program menampilkan satu persatu data yang ada pada data_pelanggan menggunakan for loop, serta menggunakan library prettytable agar tampilannya lebih rapih



1.8. Menu Mengubah Data Service (UPDATE)

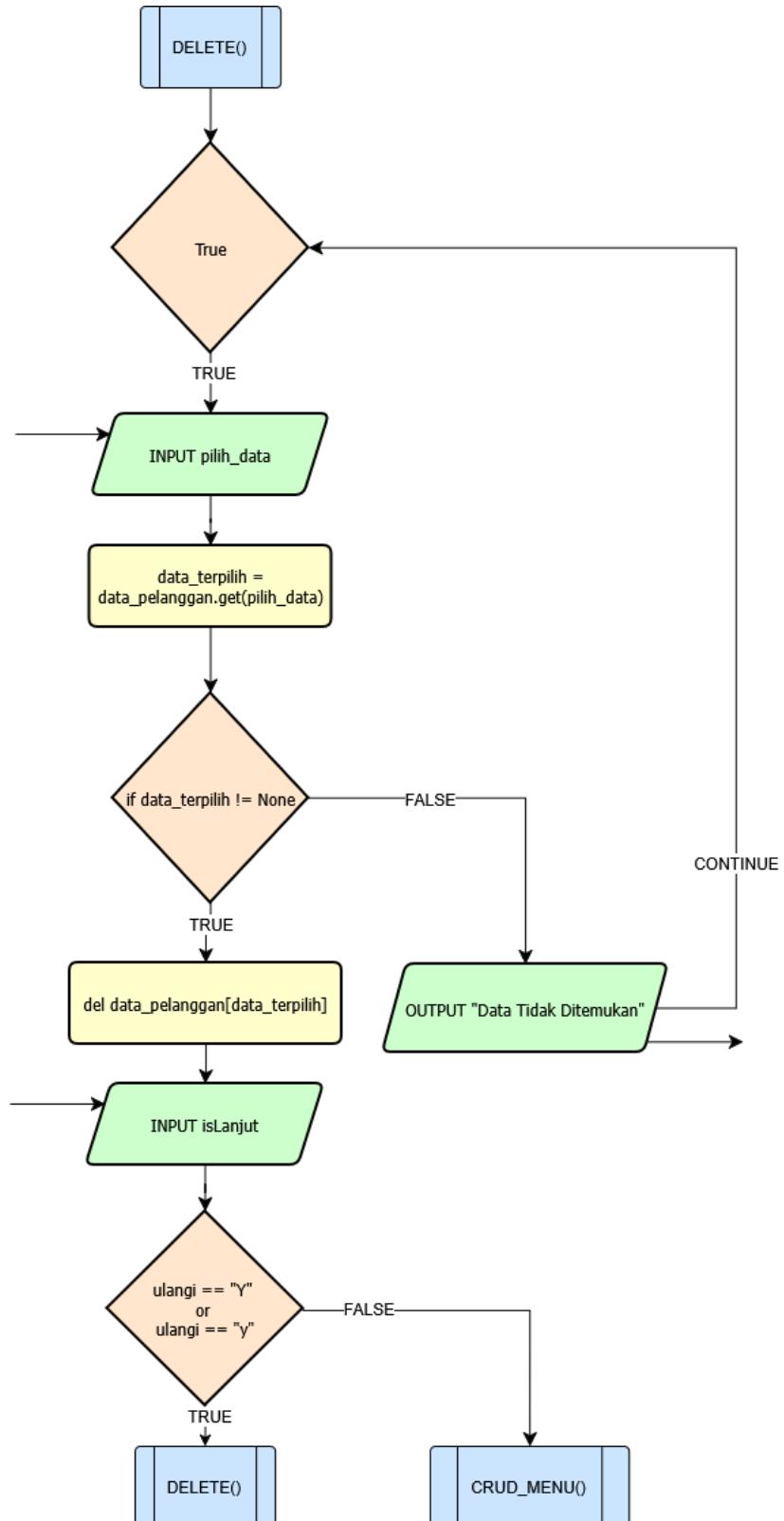
Pertama-tama, user (Admin) diminta untuk memilih data pelanggan mana yang ingin diubah menggunakan kode unik yang merupakan key dari masing masing value berupa dictionary yang ada. Jika kode yang diinput tidak terdapat pada key data_pelanggan. Maka data dinyatakan tidak ditemukan, dan diminta untuk menginput ulang hingga data ditemukan. Jika data ditemukan, maka admin diminta untuk memilih jenis data yang ingin diubah, mulai dari nama_pelanggan, jenis_perangkat,jenis_kerusakan, status_perbaikan dan biaya_service. Lalu admin meniginput data baru yang nantinya akan menimpa data lama yang telah dipilih sebelumnya.





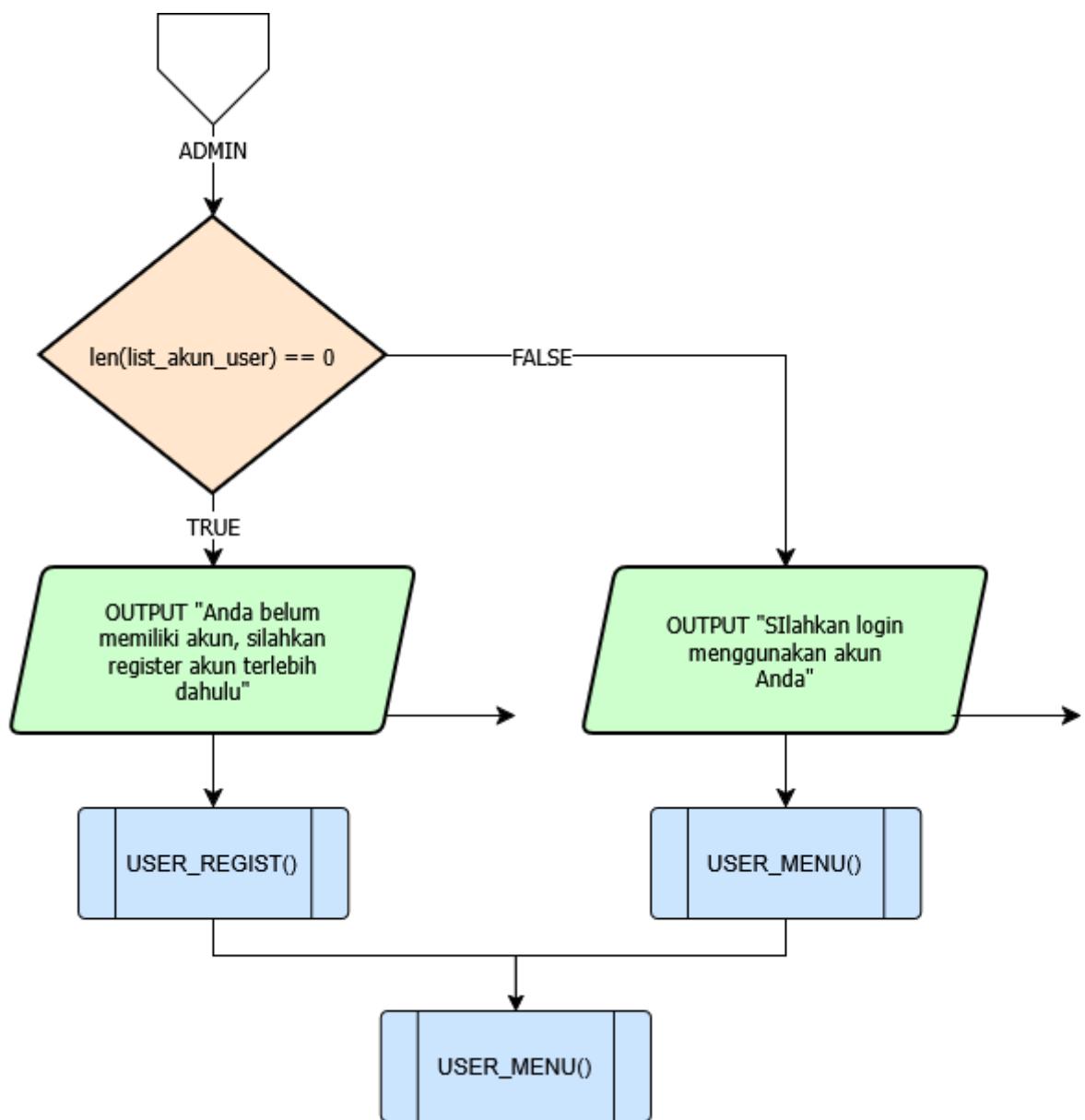
1.9. Menu Menghapus Data Servis (DELETE)

Pertama-tama, user (Admin) diminta untuk memilih data pelanggan mana yang ingin diubah menggunakan kode unik yg merupakan key dari dictionary data_pelanggan yang berisi value dictionary pula. Jika kode yang diinput tidak ada yang sesuai dengan key yang ada. Maka data dinyatakan tidak ditemukan, dan diminta untuk menginput ulang hingga data ditemukan. Jika telah ditemukan, maka data tersebut dihapus dari dictionary data_pelanggan



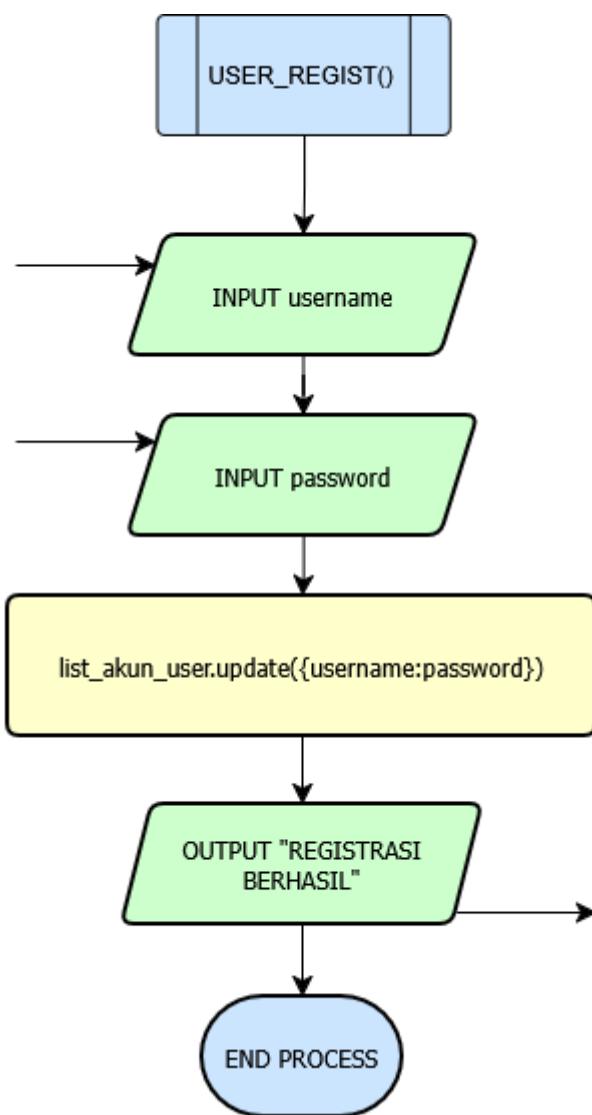
1.10. User

Jika user memilih menu USER, maka pertama-tama akan dicek apakah sudah ada akun admin dari list_akun_user. Jika tidak maka user akan diminta untuk register terlebih dahulu. Jika sudah ada, maka akan langsung diminta login



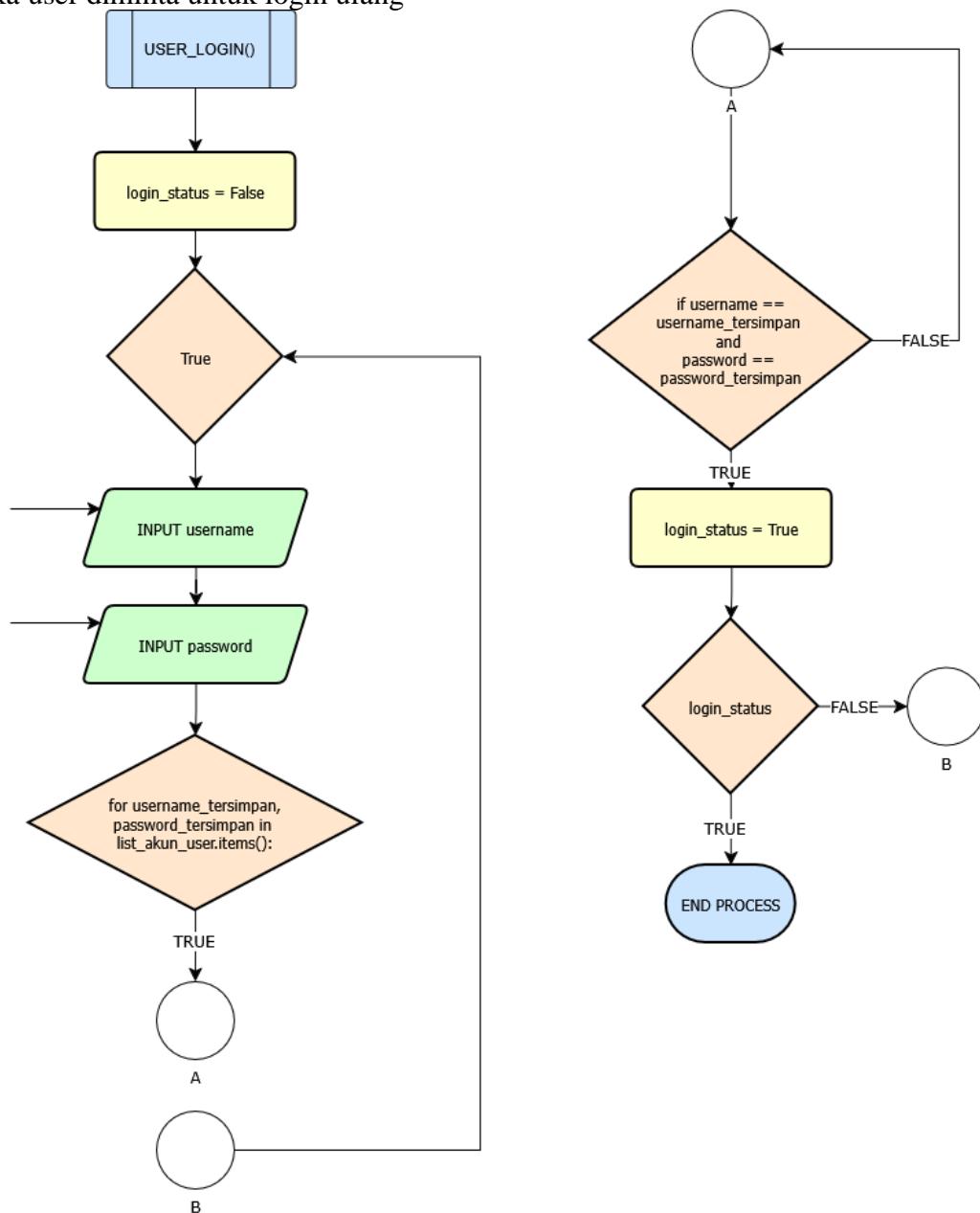
1.11. User Regist

Pada Bagian ini, User diminta untuk membuat akun user baru dengan menginputkan username dan password yang kemudian kedua variabel tersebut diupdate dalam bentuk dictionary. Kemudian regist dinyatakan berhasil



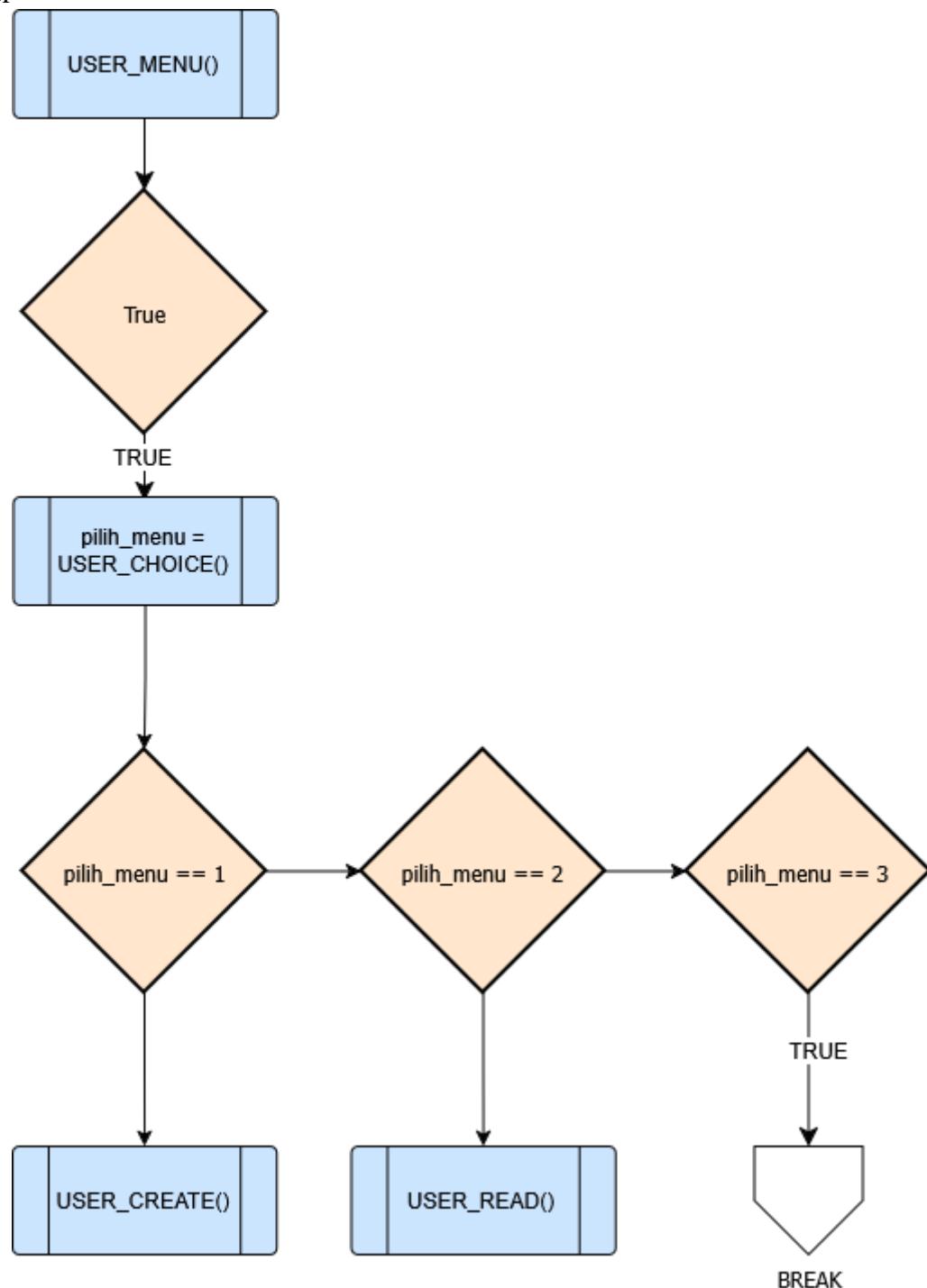
1.12. User Login

Pertama-tama login_status diset False. Lalu user diminta untuk menginput username dan password, lalu dicocokan satu persatu dengan data yang ada di list_akun_user menggunakan for loop. Jika ada yg cocok maka status_login diset True dan login dinyatakan berhasil. Jika tidak maka user diminta untuk login ulang



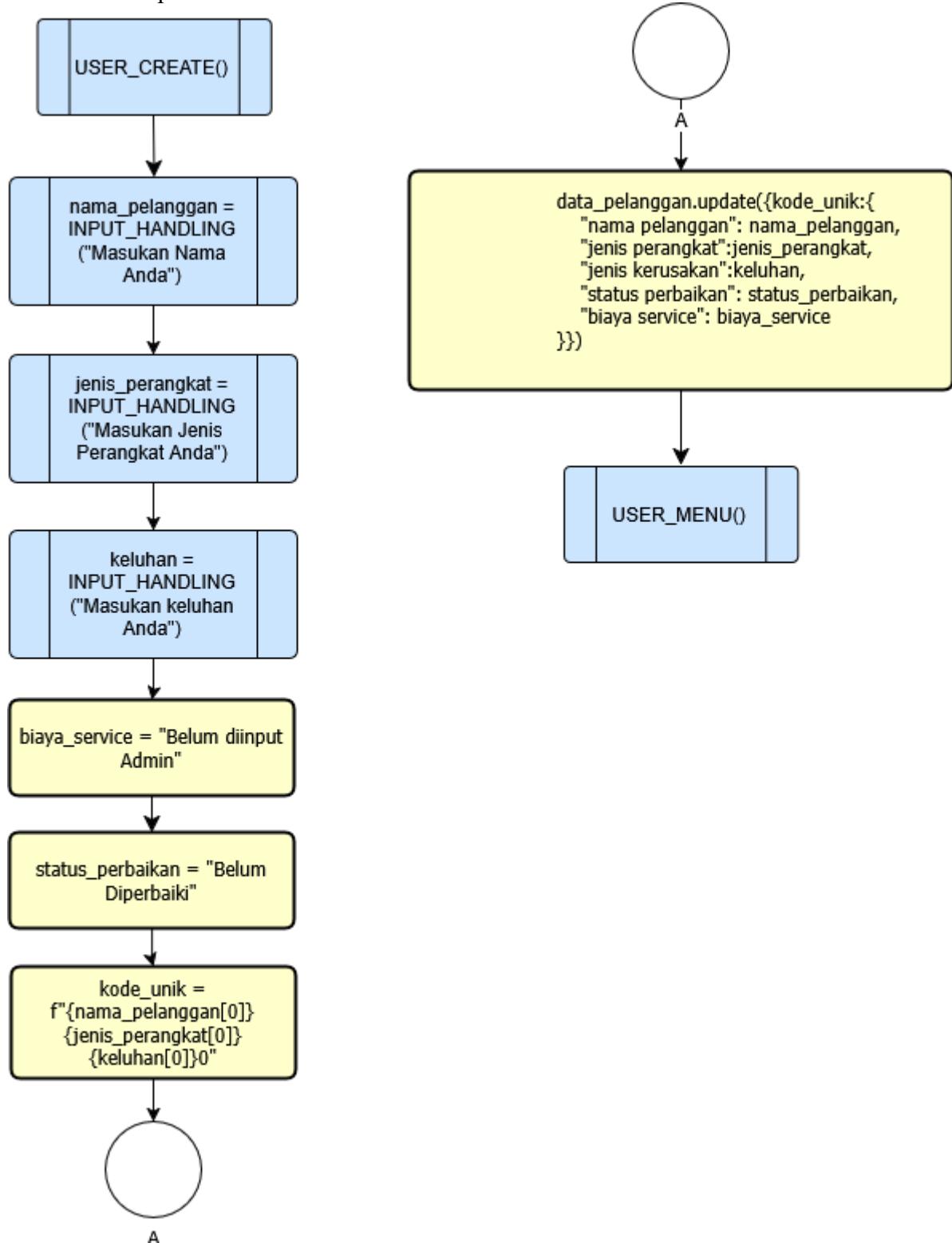
1.13. User Menu

Sedikit Berbeda dengan Admin menu, di sini user hanya bisa menambahkan data servis (CREATE) dan Menampilkan data miliknya sendiri (READ) serta menu untuk keluar (LOGOUT). Pada user menu ini juga program memanggil fungsi user_choice yang menggunakan library Inquirer untuk menambah efektivitas program dan menghindari error dari kesalahan input user



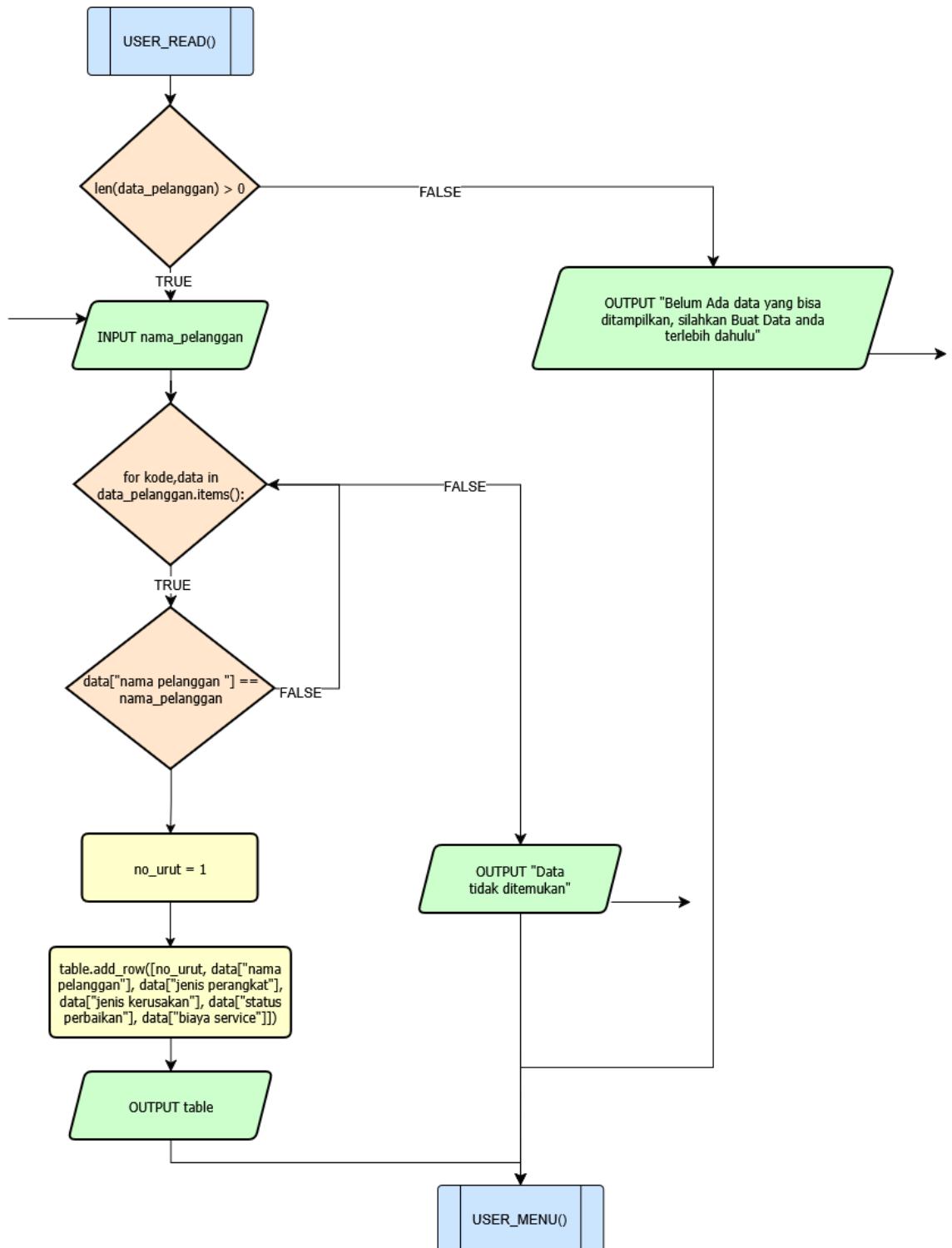
1.14. User Create

Pada Menu Create, User hanya akan mengisi nama_pelanggan, jenis_perangkat dan keluhan, sedangkan untuk status_perbaikan akan diset default ke “Belum diperbaiki” dan biaya service “Belum diinput Admin”



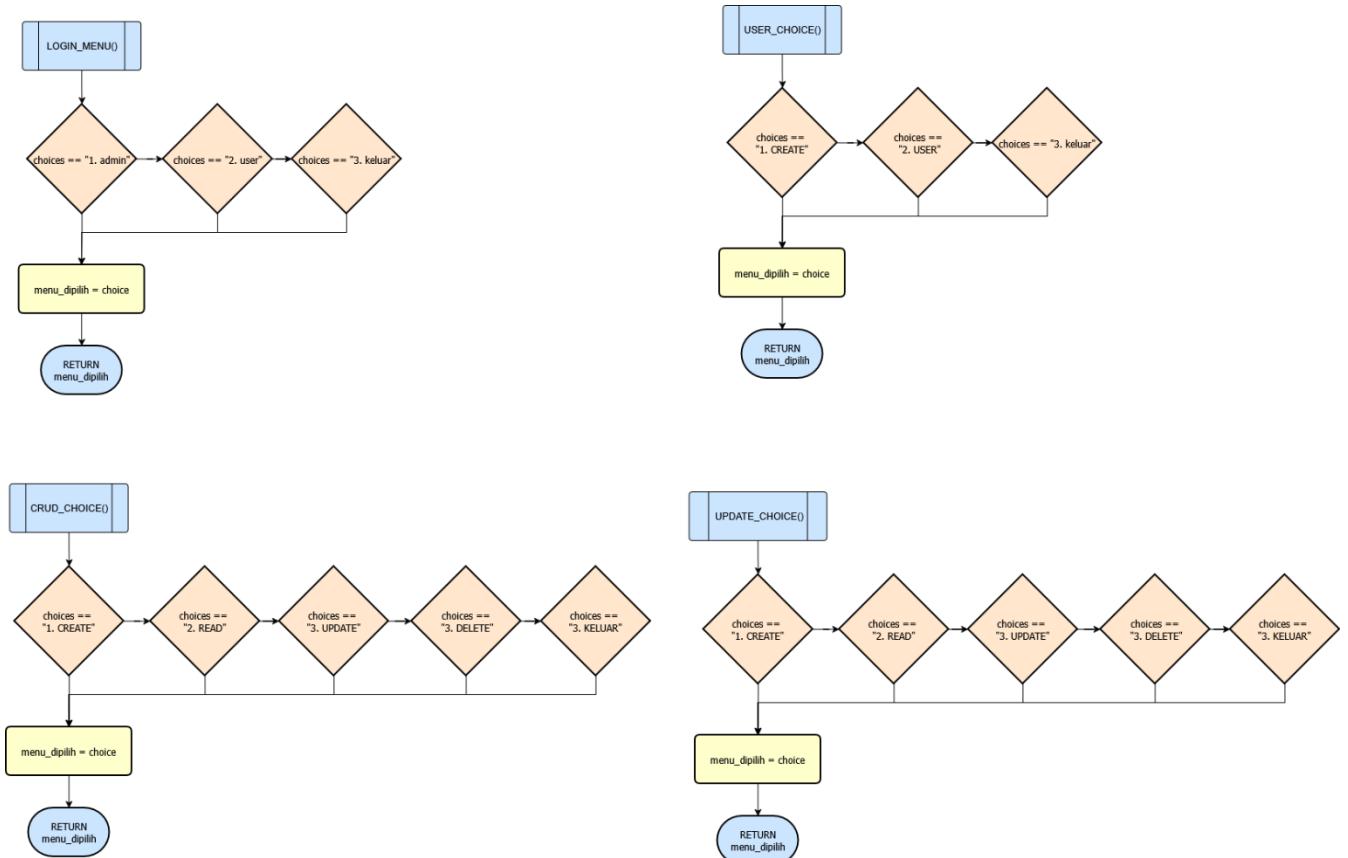
1.15. User Read

Sedikit berbeda dengan READ yang ada pada Admin, pada User data yang ditampilkan hanya miliknya sendiri dengan cara menginput nama yang nantinya akan dicek satu persatu menggunakan for loop. Jika nama ditemukan di dalam variabel nama_pelanggan, maka data akan ditampilkan, jika tidak maka program akan kembali ke menu User. Pada menu user read ini menggunakan library prettytable agar tampilannya lebih rapih



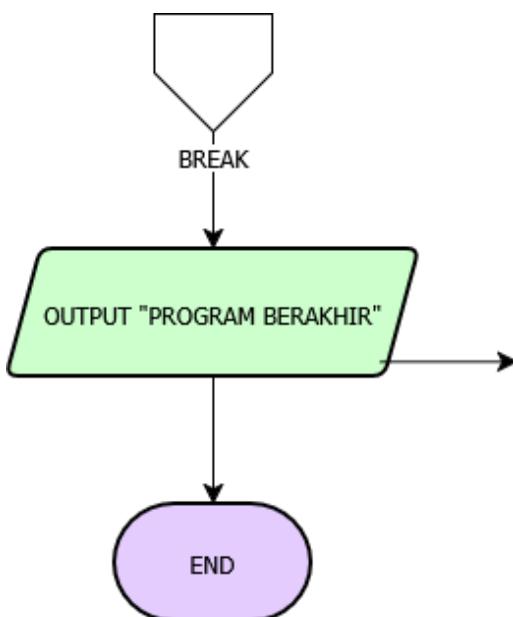
1.16. DISPLAY MENU (INQUIRER)

Ini merupakan sekumpulan fungsi untuk menampilkan Menu-menu yang ada di program menggunakan library inquirer. Tujuannya adalah agar program menjadi lebih dinamis dan meminimalisir kemungkinan error yang berasal dari kesalahan input user



1.17. BREAK

Jika User memilih Keluar pada menu login multi_user, maka program akan *break* sampai ke bagian akhir, dan program dinyatakan selesai



2. Deskripsi Singkat Program

Program ini merupakan penerapan dari CRUD Sederhana yang dapat CREATE data Service baru, READ atau menampilkan data, UPDATE atau mengubah data yang sudah ada, serta DELETE atau menghapus data.

Program ini dibagi menggunakan fitur Multi User, yaitu Admin dan User biasa. Admin dapat mengakses semua fitur CRUD, sedangkan User hanya dapat menggunakan CREATE dan READ dengan terbatas yang di mana pada bagian READ user hanya dapat menampilkan data miliknya sendiri, bukan keseluruhan data.

Program ini Juga memiliki Error Handling yang mencegah pesan error muncul ditengah-tengah program karena user menginputkan elemen yang tidak sesuai dengan kebutuhan program

Dengan bantuan library Prettytable dan Inquirer. Program ini memiliki tampilan yang dinamis dan rapih untuk meningkatkan *user experience* yang lebih baik

Program dibuat dalam bentuk Modular Programming untuk memudahkan *maintanance* serta agar struktur program menjadi lebih tertata

3. Source Code

3.1. Main Menu



```
1 if __name__ == "__main__":
2     while True:
3         pilih_login = login_menu()
4         if pilih_login == "1":
5             os.system("cls || clear")
6             if len(list_akun_admin) == 0:
7                 list_akun_admin.update(admin_regist())
8             else:
9                 admin_login(list_akun_admin)
10            crud_operation()
11
12
13
14
15        elif pilih_login == "2":
16            os.system("cls || clear")
17
18            if len(list_akun_user) == 0:
19                print("Anda belum memiliki akun, silahkan register akun terlebih dahulu\n")
20                list_akun_user.update(user_regist())
21
22            else:
23                user_login(list_akun_user)
24
25            user_operation()
26
27        elif pilih_login == "3":
28            break
29        else:
30            print("Input tidak dikenal, silahkan ulangi login")
31            continue
32
33    print("\n=====PROGRAM BERAKHIR=====")
```

3.2. DISPLAY MENU

3.2.1. Login Menu

```
● ○ ●

1 def login_menu():
2     os.system("cls || clear")
3     print("=====SISTEM SERVIS PERANGKAT ELEKTRONIK=====\n")
4     print("----SEBAGAI APA ANDA MENGGUNAKAN PROGRAM INI?----")
5
6     pilih_menu = [inquirer.List(
7         "Menu",
8         message="Pilih Role Anda",
9         choices=[
10            "1. Admin",
11            "2. User",
12            "3. Keluar"]
13        )
14    ]
15
16     menu_dipilih = inquirer.prompt(pilih_menu)
17     return menu_dipilih["Menu"][0]
```

3.2.2. CRUD Menu

```
● ○ ●

1 def crud_menu():
2     print("=====SISTEM SERVIS PERANGKAT ELEKTRONIK=====\n")
3     print("Anda Login sebagai Admin, Silahkan pilih menu dibawah:")
4     print("\n=====")
5     pilih_menu = [inquirer.List(
6         "Menu",
7         message="Pilih Menu",
8         choices=[
9             "1. Tambah Data Servis (Create)",
10            "2. Tampilkan Data Servis (Read)",
11            "3. Ubah Data Service (Update)",
12            "4. Hapus Data Pelanggan (Delete)",
13            "5. Keluar"]
14        )
15    ]
16     menu_dipilih = inquirer.prompt(pilih_menu)
17     print("=====")
18     return menu_dipilih["Menu"][0]
```

3.2.3. User Menu

```
● ● ●

1 def user_menu():
2     print("=====SISTEM SERVIS PERANGKAT ELEKTRONIK=====\n")
3     print("Anda Login sebagai User, Silahkan pilih menu dibawah:")
4
5     print("\n=====")
6
7
8     pilih_menu = [inquirer.List(
9         "Menu",
10        message="Pilih Menu",
11        choices=[
12            "1. Buat Laporan Servis",
13            "2. Tampilkan Data Servis Saya",
14            "3. Keluar"]
15        )
16    ]
17     menu_dipilih = inquirer.prompt(pilih_menu)
18
19     return menu_dipilih["Menu"][0]
```

3.2.4. Update Menu

```
● ● ●

1 def update_menu():
2     os.system("cls || clear")
3     print("\n=====")
4     print("Pilih jenis data servis yang ingin anda ubah: ")
5     print("=====")
6     pilih_menu = [inquirer.List(
7         "Menu",
8         message="Pilih Menu",
9         choices=[
10            "1. Nama Pelanggan",
11            "2. Jenis Perangkat",
12            "3. Jenis Kerusakan",
13            "4. Status Perbaikan",
14            "5. Biaya Service",]
15        )
16    ]
17     menu_dipilih = inquirer.prompt(pilih_menu)
18
19     return menu_dipilih["Menu"][0]
```

3.3. ADMIN AUTHENTICATION

```
1 import os
2
3 def admin_regist():
4     print("Anda belum memiliki akun Admin, silahkan register akun terlebih dahulu\n")
5     username = input("Buat Username Anda: ")
6     password = input("Buat Password Anda: ")
7     {username:password}
8     os.system("cls || clear")
9     print("\n-----REGISTRASI BERHASIL!-----\n")
10    return {username:password}
11
12 def admin_login(list_akun_admin):
13     login_status = False
14     print("Silahkan login menggunakan akun Admin")
15     while True:
16         username = input("Masukan Username Admin: ")
17         password = input("Masukan Password Admin: ")
18         for username_tersimpan,password_tersimpan in list_akun_admin.items():
19             if username == username_tersimpan and password == password_tersimpan:
20                 os.system("cls || clear")
21                 print("\n-----Login Berhasil!-----\n")
22                 login_status = True
23                 break
24
25         if login_status:
26             break
27     else:
28         os.system("cls || clear")
29         print("\n=====Username dan Password tidak sesuai, silahkan ulangi lagi=====")
30         continue
31
```

3.4. CRUD Select Menu

```
1 def crud_operation():
2     while True:
3         crud_menu()
4         pilih_menu = input("Pilih menu [1-5]: ")
5         if pilih_menu == "1":
6             create_data(data_pelanggan)
7
8         elif pilih_menu == "2":
9             read_data(data_pelanggan)
10
11        elif pilih_menu == "3":
12            update_data(data_pelanggan)
13
14        elif pilih_menu == "4":
15            delete_data(data_pelanggan)
16
17        elif pilih_menu == "5":
18            break
19
20    else:
21        print("\nPilihan tidak dikenali, mohon input pilihan antara angka 1 sampai 5!\n")
22        continue
```

3.5. Menu Tambah Data Servis (CREATE)

```
● ● ●

1 def create_data(data_pelanggan):
2
3     os.system("cls || clear")
4     print("\n=====Silahkan Buat Data pelanggan baru=====\\n")
5
6     nama_pelanggan = input_handling("Masukan nama pelanggan: ")
7     jenis_perangkat = input_handling("Masukan jenis Perangkat: ")
8     keluhan = input_handling("Masukan jenis kerusakan: ")
9     biaya_service = input_number_handling("Masukan biaya service: ")
10    status_perbaikan = "Belum Diperbaiki"
11
12    kode_unik = f"{nama_pelanggan[0]}{jenis_perangkat[0]}{keluhan[0]}{str(biaya_service)[0]}"
13
14
15    data_pelanggan.update({kode_unik:{
16        "nama pelanggan": nama_pelanggan,
17        "jenis perangkat":jenis_perangkat,
18        "jenis kerusakan":keluhan,
19        "status perbaikan": status_perbaikan,
20        "biaya service": biaya_service
21        }})
22    print("\nData Berhasil Ditambahkan!\\n")
23    ulangi = input("Apakah Anda ingin menambahkan data lagi? ketik Y jika iya: ")
24    if ulangi == "Y" or ulangi == "y":
25        os.system("cls || clear")
26        create_data(data_pelanggan)
27    else:
28        os.system("cls || clear")
29
```

3.6. Menu Menampilkan Data Servis (READ)

3.6.1. Admin Read

```
● ● ●

1 def read_data(data_pelanggan):
2
3     os.system("cls || clear")
4
5     table = PrettyTable()
6     no_urut = 1
7     if len(data_pelanggan) > 0:
8         table.field_names = ["No. ", "Kode", "Nama Pelanggan", "Jenis Perangkat", "Jenis Kerusakan", "Status Perbaikan",
9         "Biaya Service"]
10
11         for kode_unik,data in data_pelanggan.items():
12             table.add_row([no_urut, kode_unik, data["nama pelanggan"], data["jenis perangkat"], data["jenis kerusakan"],
13             data["status perbaikan"], data["biaya service"]])
14             no_urut+=1
15     print(table)
16     print()
17     else:
18         print("\nBelum ada Data yang tersimpan, silahkan Buat Data servis terlebih dahulu\\n")
```

3.6.2. User Read



```
1 def user_read(data_pelanggan):
2     os.system("cls || clear")
3     table = PrettyTable()
4
5     if len(data_pelanggan) > 0:
6         nama_pelanggan = input("Masukan nama untuk melihat data servis Anda: ")
7         for kode,data in data_pelanggan.items():
8             if data["nama pelanggan"] == nama_pelanggan:
9                 no_urut = 1
10                table.field_names = ["No. ", "Nama Pelanggan", "Jenis Perangkat", "Jenis Kerusakan", "Status Perbaikan",
11 "Biaya Service"]
12
13                table.add_row([no_urut, data["nama pelanggan"], data["jenis perangkat"], data["jenis kerusakan"], data
14 ["status perbaikan"], data["biaya service"]])
15
16                print(table)
17                print()
18                data_ditemukan = True
19                break
20            else:
21                data_ditemukan = False
22        if(data_ditemukan == False):
23            os.system("cls || clear")
24            print("Mohon Maaf, data anda tidak ditemukan")
25    else:
26        os.system("cls || clear")
27        print("Belum Ada data pelanggan, silahkan buat data anda terlebih dahulu")
```

3.7. Menu Mengubah Data Service (UPDATE)

```
● ● ●

1 def update_data(data_pelanggan):
2     read_data(data_pelanggan)
3     if len(data_pelanggan) > 0:
4
5         while True:
6
7             pilih_data = input("\npilih data yang ingin anda ubah berdasarkan Kodenya: ")
8
9             data_terpilih = data_pelanggan.get(pilih_data)
10            if data_terpilih != None:
11                break
12            else:
13                print("Data tidak ditemukan, Coba lagi")
14                continue
15
16        while True:
17
18            pilih_data_baru = update_menu()
19            if pilih_data_baru == "1":
20                nama_pelanggan_baru = input_handling("Masukan Nama pelanggan Baru: ")
21                data_terpilih["nama pelanggan"] = nama_pelanggan_baru
22                break
23
24            elif pilih_data_baru == "2":
25                jenis_perangkat_baru = input_handling("Masukan Jenis Perangkat Baru: ")
26                data_terpilih["jenis perangkat"] = jenis_perangkat_baru
27                break
28
29            elif pilih_data_baru == "3":
30                keluhan_baru = input_handling("Masukan Jenis Kerusakan Baru: ")
31                data_terpilih["jenis kerusakan"] = keluhan_baru
32                break
33
34            elif pilih_data_baru == "4":
35                status_perbaikan_baru = input_handling("Masukan Status Perbaikan Baru: ")
36                data_terpilih["status perbaikan"] = status_perbaikan_baru
37                break
38
39            elif pilih_data_baru == "5":
40                biaya_service_baru = input_number_handling("Masukan Biaya service Baru: ")
41                data_terpilih["biaya service"] = biaya_service_baru
42                break
43
44            else:
45                os.system("cls || clear")
46                print("Pilihan tidak dikenali, silahkan ulangi lagi")
47                continue
48
49            print("\n=====Data sukses Diubah!=====\n")
50            ulangi = input("Apakah ada data yang ingin anda ubah lagi? ketik Y jika iya: ")
51            if ulangi == "Y" or ulangi == "y":
52                os.system("cls || clear")
53                update_data(data_pelanggan)
54            else:
55                os.system("cls || clear")
```

3.8. Menu Menghapus Data Servis (DELETE)

```
● ● ●

1 import os
2 from READ import read_data
3
4 def delete_data(data_pelanggan):
5     read_data(data_pelanggan)
6     if len(data_pelanggan) > 0:
7         pilih_data = input("\n\npilih data yang ingin anda hapus berdasarkan Kodenya: ")
8
9         while True:
10             data_terpilih = data_pelanggan.get(pilih_data)
11             if data_terpilih != None:
12                 del data_pelanggan[pilih_data]
13                 print("\n=====Data Berhasil Dihapus!=====\\n")
14                 break
15             else:
16                 print("Data tidak ditemukan, Coba lagi")
17                 continue
18
19             ulangi = input("Apakah ada data yang ingin anda hapus lagi? ketik Y jika iya: ")
20             if ulangi == "Y" or ulangi == "y":
21                 os.system("cls || clear")
22                 delete_data(data_pelanggan)
23             else:
24                 os.system("cls || clear")
25
26
```

3.9. USER AUTHENTICATION

```
● ● ●

1 def user_regist():
2
3     username = input("Buat Username Anda: ")
4     password = input("Buat Password Anda: ")
5     os.system("cls || clear")
6     print("\n-----REGISTRASI BERHASIL!-----\n")
7     return {username:password}
8
9 def user_login(list_akun_user):
10    login_status = False
11
12
13    pilih_menu_login = [inquirer.List(
14        "Menu Login",
15        message="Silahkan login dengan akun yang telah ada atau buat akun baru",
16        choices=[
17            "1. Buat Akun",
18            "2. Login dengan akun yang ada"
19        ]
20    )]
21
22    menu_dipilih = inquirer.prompt(pilih_menu_login)["Menu Login"][0]
23
24    if menu_dipilih == "1":
25        user_regist()
26    else:
27
28        while True:
29            username = input("Masukan Username Anda: ")
30            password = input("Masukan Password Anda: ")
31            for username_tersimpan,password_tersimpan in list_akun_user.items():
32                if username == username_tersimpan and password == password_tersimpan:
33                    os.system("cls || clear")
34                    print("\n-----Login Berhasil!-----\n")
35                    login_status = True
36                    break
37
38                if login_status:
39                    break
40                else:
41                    os.system("cls || clear")
42                    print("\n=====Username dan Password tidak sesuai, silahkan ulangi lagi=====")
43                    continue
```

3.10. MENAMBAH DATA UNTUK USER

```
1  def user_create(data_pelanggan):
2      os.system("cls || clear")
3      print("\n=====Silahkan Buat Data Servis Anda=====\\n")
4
5      nama_pelanggan = input_handling("Masukan nama Anda: ")
6      jenis_perangkat = input_handling("Masukan jenis Perangkat Anda: ")
7      keluhan = input_handling("Masukan Keluhan Anda: ")
8      biaya_service = "Belum diinput Admin"
9      status_perbaikan = "Belum Diperbaiki"
10     kode_unik = f"{nama_pelanggan[0]}{jenis_perangkat[0]}{keluhan[0]}0"
11     data_pelanggan.update({kode_unik:{})
12         "nama pelanggan": nama_pelanggan,
13         "jenis perangkat":jenis_perangkat,
14         "jenis kerusakan":keluhan,
15         "status perbaikan": status_perbaikan,
16         "biaya service": biaya_service
17     }})
18
19     os.system("cls || clear")
20     print("\nData Anda Berhasil Ditambahkan!\\n")
21
```

3.11. ERROR Handling

```
1 def input_number_handling(input_message):
2     while True:
3         try:
4             user_input = int(input(input_message))
5             return user_input
6         except ValueError:
7             print("Mohon Masukan Angka!\n")
8             continue
9
10 def input_handling(input_message):
11     while True:
12
13     try:
14         user_input = input(input_message)
15         if user_input == "" or user_input == " "*len(user_input):
16             raise ValueError("Input tidak boleh kosong, silahkan coba lagi!\n")
17         else:
18             return user_input
19     except ValueError as e:
20         print(e)
21         continue
22
```

4. Hasil Output

```
=====SISTEM SERVIS PERANGKAT ELEKTRONIK=====

-----SEBAGAI APA ANDA MENGGUNAKAN PROGRAM INI?-----
[?] Pilih Role Anda:
 > 1. Admin
 2. User
 3. Keluar
```

Gambar 4.1 Menu

Multi User Login

```
Anda belum memiliki akun Admin, silahkan register akun terlebih dahulu

Buat Username Anda: rafi
Buat Password Anda: admin123
```

Gambar 4.2 Register

Akun Admin

```
Silahkan login menggunakan akun Admin
Masukan Username Admin: rafi
Masukan Password Admin: admin123
```

Gambar 4.3 Login
Akun Admin (Berhasil)

```
Silahkan login menggunakan akun Admin  
Masukan Username Admin: rafi  
Masukan Password Admin: admin123
```

Gambar 4.4 Login
Akun Admin (Gagal)

```
-----Login Berhasil!-----  
=====SISTEM SERVIS PERANGKAT ELEKTRONIK=====  
Anda Login sebagai Admin, Silahkan pilih menu dibawah:  
=====  
[?] Pilih Menu:  
> 1. Tambah Data Servis (Create)  
    2. Tampilkan Data Servis (Read)  
    3. Ubah Data Service (Update)  
    4. Hapus Data Pelanggan (Delete)  
    5. Keluar
```

Gambar 4.5
CRUD Menu (Admin)

```
=====Silahkan Buat Data pelanggan baru=====
```

```
Masukan nama pelanggan: Bakil  
Masukan jenis Perangkat: laptop  
Masukan jenis kerusakan: lcd copot  
Masukan biaya service: 950000
```

```
Data Berhasil Ditambahkan!
```

```
Apakah Anda ingin menambahkan data lagi? ketik Y jika iya:
```

Gambar 4.6 Membuat/Menambah Data Baru
(CREATE)

No.	Kode	Nama Pelanggan	Jenis Perangkat	Jenis Kerusakan	Status Perbaikan	Biaya Service
1	BLL9	Bakil	Laptop	Lcd copot	Belum Diperbaiki	950000
2	RHB5	Rafi	Hp	Baterai kembung	Belum Diperbaiki	500000
3	JPT6	Jopan	Printer	Tinta macet	Belum Diperbaiki	670000

Gambar 4.7 Menampilkan
Data (READ)

```
=====
Pilih jenis data servis yang ingin anda ubah:
=====
[?] Pilih Menu:
    1. Nama Pelanggan
    2. Jenis Perangkat
    3. Jenis Kerusakan
    > 4. Status Perbaikan
    5. Biaya Service
>
Masukan Status Perbaikan Baru: Sudah Diperbaiki

=====Data sukses Diubah!=====

Apakah ada data yang ingin anda ubah lagi? ketik Y jika iya: █
```

Gambar 4.8 Mengubah
Data (UPDATE)

No.	Kode	Nama Pelanggan	Jenis Perangkat	Jenis Kerusakan	Status Perbaikan	Biaya Service
1	BLL9	Bakil	Laptop	Lcd copot	Belum Diperbaiki	950000
2	RHB5	Rafi	Hp	Baterai kembung	Sudah diperbaiki	500000
3	JPT6	Jopan	Printer	Tinta macet	Belum Diperbaiki	670000

Gambar 4.9 Tampilan Data
Setelah diupdate

```
=====
| No. | Kode | Nama Pelanggan | Jenis Perangkat | Jenis Kerusakan | Status Perbaikan | Biaya Service |
+---+---+-----+-----+-----+-----+-----+
| 1 | BLL9 | Bakil | Laptop | Lcd copot | Belum Diperbaiki | 950000 |
| 2 | RHB5 | Rafi | Hp | Baterai kembung | Sudah diperbaiki | 500000 |
| 3 | JPT6 | Jopan | Printer | Tinta macet | Belum Diperbaiki | 670000 |
>+-----+-----+-----+-----+-----+-----+-----+
pilih data yang ingin anda hapus berdasarkan Kodenya: RHB5

=====Data Berhasil Dihapus!=====

Apakah ada data yang ingin anda hapus lagi? ketik Y jika iya: █
```

Gambar 4.10 Menghapus
Data (DELETE)

No.	Kode	Nama Pelanggan	Jenis Perangkat	Jenis Kerusakan	Status Perbaikan	Biaya Service
1	BLL9	Bakil	Laptop	Lcd copot	Belum Diperbaiki	950000
2	JPT6	Jopan	Printer	Tinta macet	Belum Diperbaiki	670000

Gambar 4.11 Tampilan Data
Setelah dihapus

```
-----REGISTRASI BERHASIL!-----
>=====SISTEM SERVIS PERANGKAT ELEKTRONIK=====

Anda Login sebagai User, Silahkan pilih menu dibawah:

=====
[?] Pilih Menu:
    > 1. Buat Laporan Servis
        2. Tampilkan Data Servis Saya
        3. Keluar
```

Gambar 4.12 User
Menu

```
=====Silahkan Buat Data Servis Anda=====

>Masukan nama Anda: Nurman
Masukan jenis Perangkat Anda: Laptop
Masukan Keluhan Anda: RAM tidak terbaca
```

Gambar 4.13 Tambah Data
Untk User

Masukan nama untuk melihat data servis Anda: Nurman						
No.	Nama Pelanggan	Jenis Perangkat	Jenis Kerusakan	Status Perbaikan	Biaya Service	
1	Nurman	Laptop	Ram tidak terbaca	Belum Diperbaiki	Belum diinput Admin	

Gambar 4.14 Menampilkan Data
Hanya Milik User

5. Langkah-langkah GIT

5.1 GIT Init

Menginisiasi repository Git baru di lokal

```
PS C:\Users\MyBook Z Series\Desktop\AHMAD RAFI\praktikum-apd> git init
Initialized empty Git repository in C:/Users/MyBook Z Series/Desktop/AHMAD RAFI/praktikum-apd/.git/
```

5.2 GIT Add

Menambahkan file/perubahan pada file ke staging area sebelum di commit ke repository lokal

```
PS C:\Users\MyBook Z Series\Desktop\AHMAD RAFI\praktikum-apd> git add *
```

5.3 GIT Commit

Menyimpan snapshot perubahan yang ada di staging area ke repository lokal

```
PS C:\Users\MyBook Z Series\Desktop\AHMAD RAFI\praktikum-apd> git commit -m "add python program"
[main 23df0e1] add python program
 1 file changed, 323 insertions(+)
 create mode 100644 post-test/post-test-apd-6/2509106034-AHMAD RAFI' IRSYAD NUGRAHA-PT-6.py
```

5.4 GIT Remote

Menghubungkan repository lokal ke repository yang ada di GitHub

```
PS C:\Users\MyBook Z Series\Desktop\AHMAD RAFI\praktikum-apd> git remote add origin https://github.com/Raafx/praktikum-apd.git
```

5.5 GIT Push

Mengirim commit dari repository lokal ke repository remote (GitHub)

```
PS C:\Users\MyBook Z Series\Desktop\AHMAD RAFI\praktikum-apd> git push origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 971 bytes | 161.00 KiB/s, done.
Total 8 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 6 local objects.
To https://github.com/Raafx/praktikum-apd.git
 89e6385..518e06a  main -> main
```