

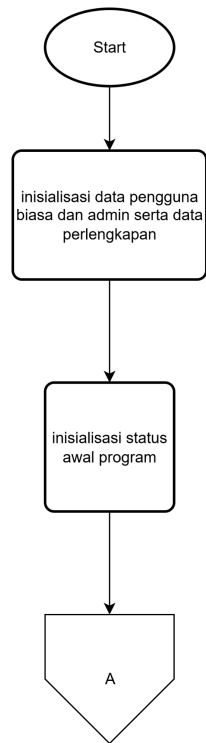
**LAPORAN PRAKTIKUM**  
**POSTTEST 5**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Nama (2509106021)**  
**Kelas (A1 '25)**

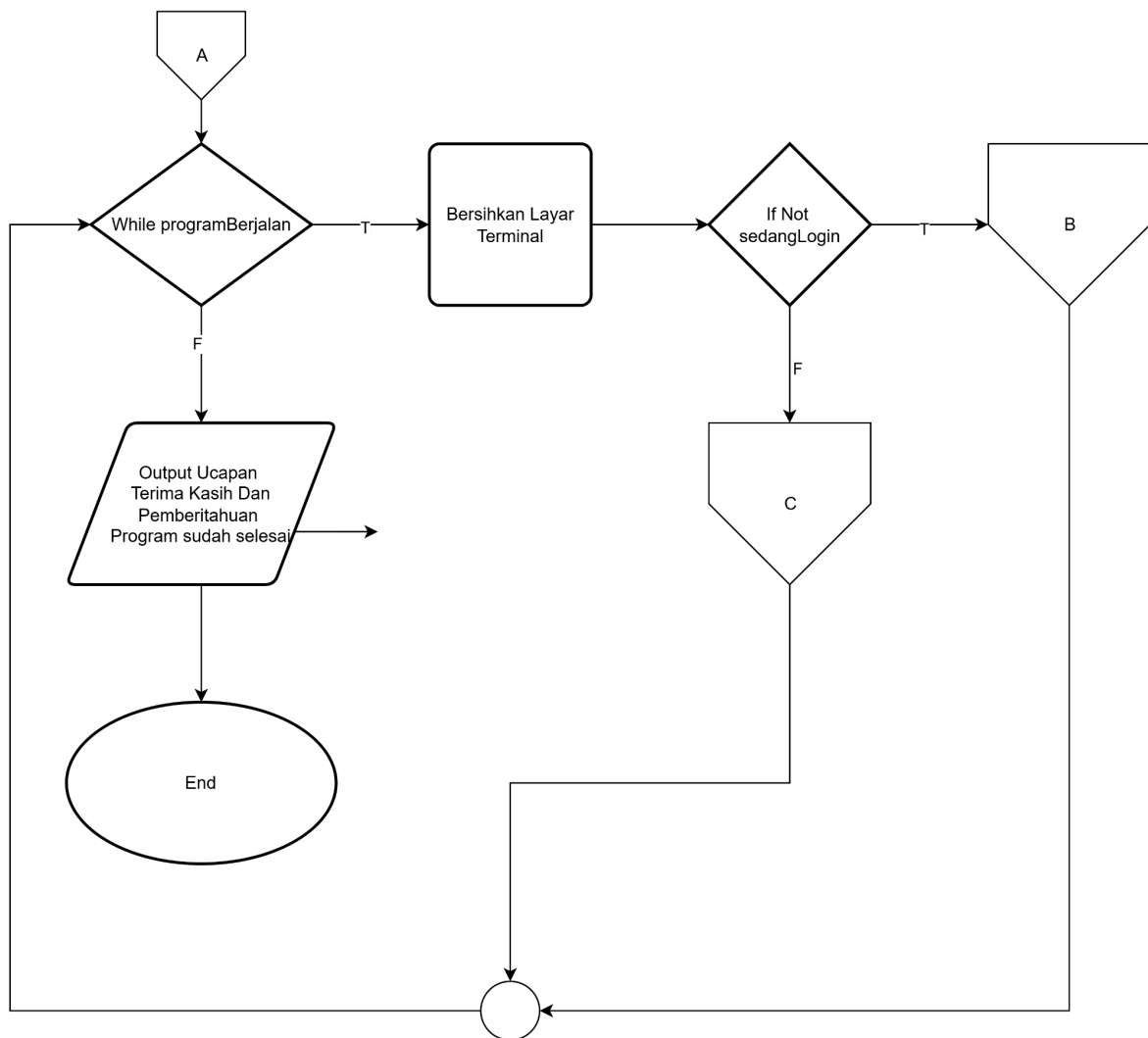
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart



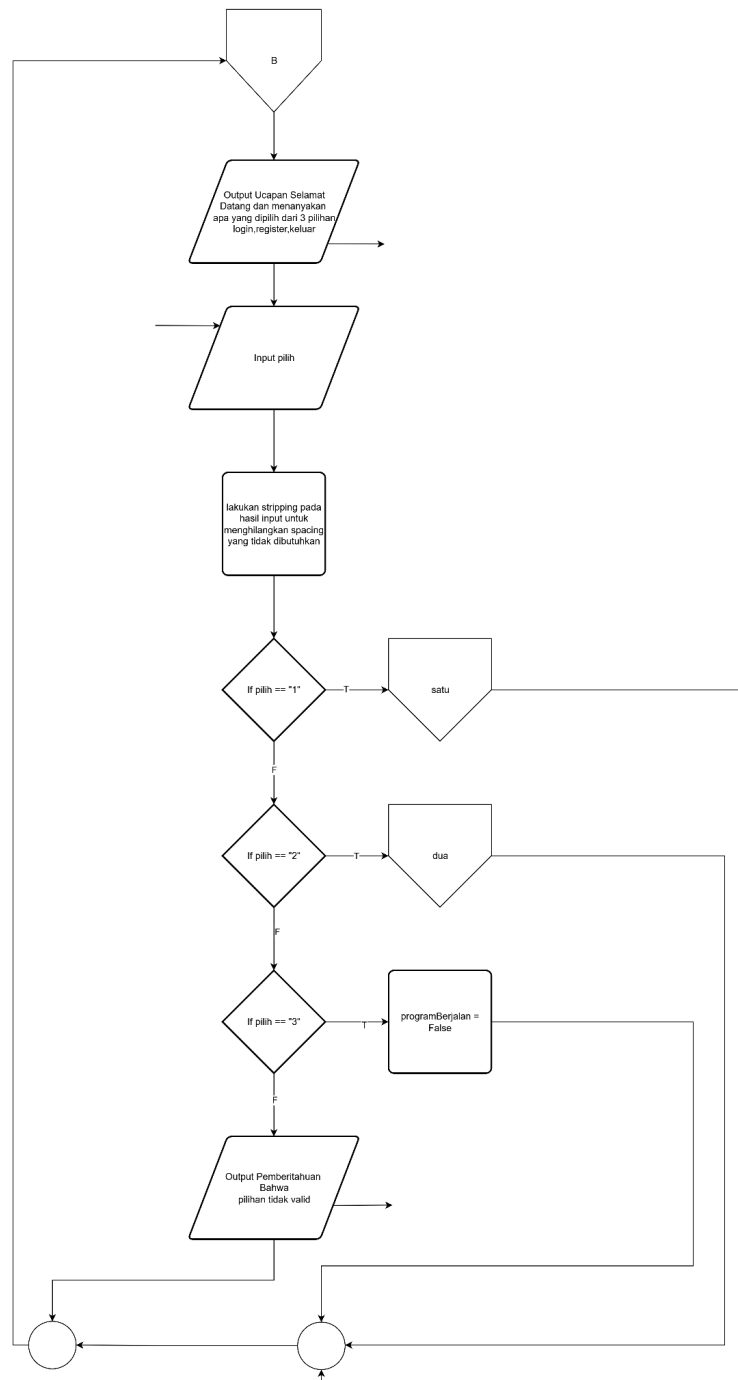
**Gambar 1.1** *Inisialisasi*

**Gambar 1.1** menunjukkan diagram alir (flowchart) yang menggambarkan tahapan awal inisialisasi program sebelum memasuki proses utama. Program dimulai dari simbol “Start”, lalu melakukan inisialisasi data pengguna meliputi data pengguna biasa dan admin serta inisialisasi data perlengkapan yang akan digunakan dalam sistem. Setelah itu, program mengatur status awal, seperti variabel atau kondisi sistem yang diperlukan untuk menjalankan logika berikutnya. Alur kemudian diarahkan ke simbol connector berlabel “A”, yang berfungsi sebagai penghubung ke bagian lanjutan dari flowchart (biasanya di halaman atau bagian berikutnya) untuk melanjutkan eksekusi program. Tahapan ini penting untuk memastikan seluruh komponen sistem telah siap sebelum memasuki menu atau proses inti aplikasi.



**Gambar 1.2 Proses Looping**

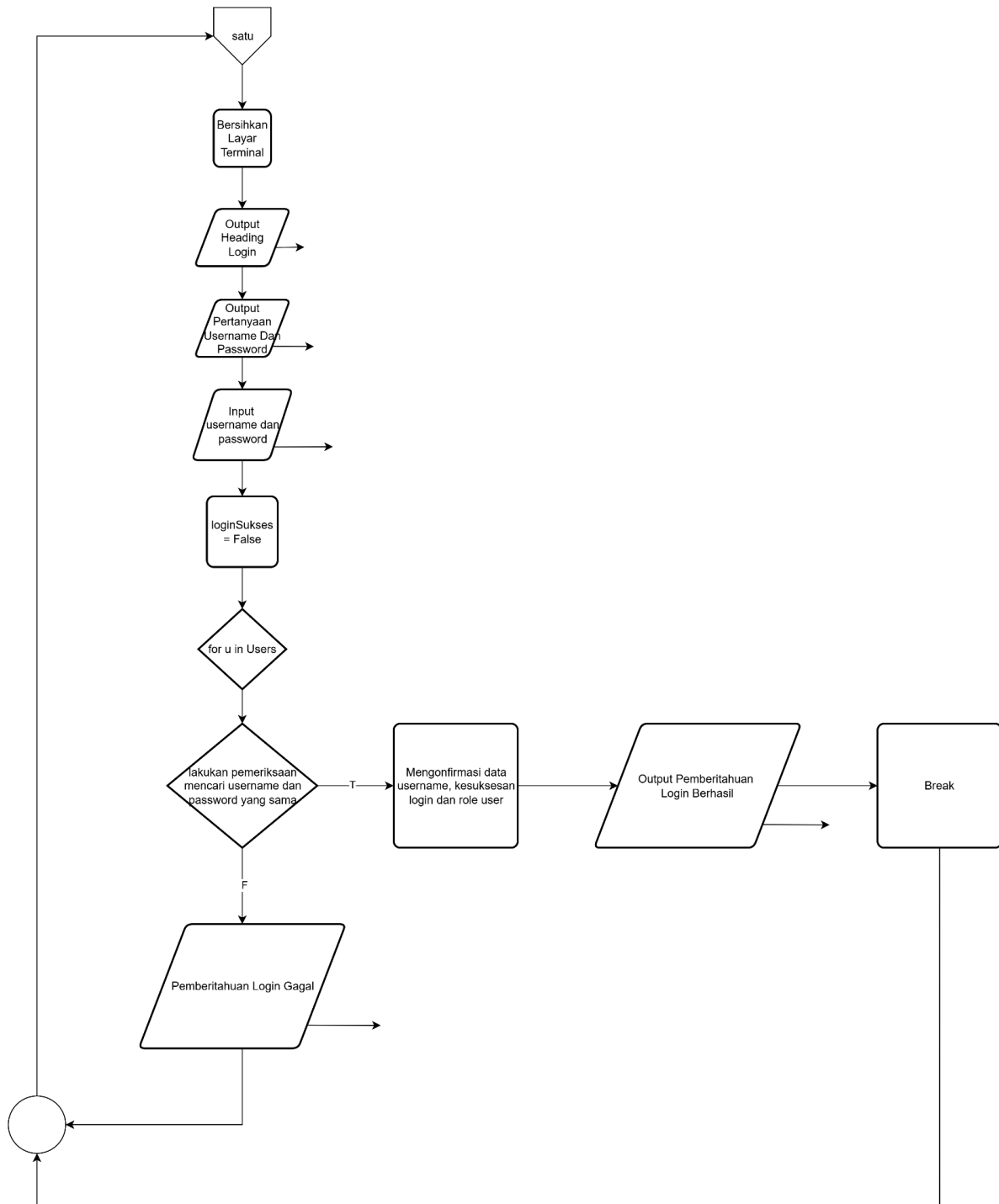
**Gambar 1.2** melanjutkan alur program dari **Gambar 1.1** melalui connector “A”, dan menggambarkan struktur utama pengendali program berbasis loop serta pengecekan status login. Program memasuki perulangan “While programBerjalan” yang akan terus berjalan selama kondisi tersebut bernilai benar (True), di mana setiap iterasi dimulai dengan membersihkan layar terminal untuk menampilkan tampilan yang rapi. Di dalam loop ini, program kemudian memeriksa kondisi “If Not sedangLogin”; jika pengguna belum login (kondisi True), alur dialihkan ke connector “B” untuk memulai proses autentikasi atau login. Namun, jika pengguna sudah login (kondisi False), alur dialihkan ke connector “C”, yang kemungkinan besar mengarah ke menu utama atau fitur-fitur program yang dapat diakses oleh pengguna yang telah terautentikasi. Jika pada suatu titik kondisi “While programBerjalan” menjadi salah (False), maka program keluar dari loop, menampilkan pesan ucapan terima kasih dan pemberitahuan bahwa program telah selesai, lalu berakhir di simbol “End”. Dengan demikian, Gambar 1.2 menggambarkan logika inti program yang mengelola siklus hidup aplikasi, mulai dari menjaga program tetap berjalan hingga menangani kondisi login dan akhirnya menutup program secara teratur.



**Gambar 1.3 Penentuan Pilihan**

**Gambar 1.3** melanjutkan alur dari connector B pada **Gambar 1.2** dan menggambarkan proses menu utama sebelum login. Program menampilkan pesan selamat datang dan memberikan tiga pilihan kepada pengguna: login, register, atau keluar. Setelah pengguna memasukkan pilihannya, input tersebut dibersihkan dari spasi berlebih menggunakan fungsi strip. Program kemudian memeriksa nilai input tersebut. Jika input bernilai 1, alur dialihkan ke connector satu yang kemungkinan besar mengarah ke proses login. Jika input bernilai 2, alur dialihkan ke connector dua yang kemungkinan mengarah ke proses registrasi. Jika input bernilai 3, variabel programBerjalan diubah menjadi False,

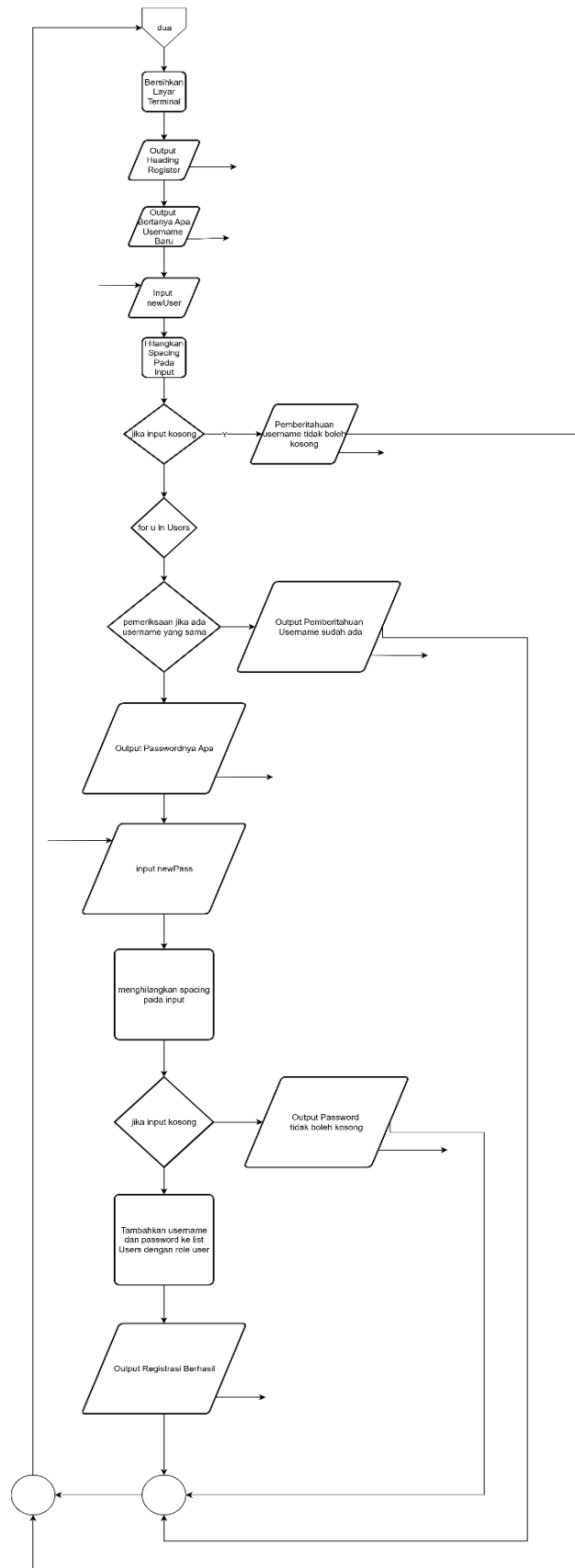
sehingga program akan berhenti dan berakhir sesuai alur pada **Gambar 1.2**. Jika input tidak sesuai dengan ketiga pilihan tersebut, program menampilkan pesan bahwa pilihan tidak valid, lalu kembali ke awal loop untuk menunggu input baru. Dengan demikian, **Gambar 1.3** berfungsi sebagai menu awal yang mengatur akses pengguna sebelum masuk ke sistem.



**Gambar 1.4** pilihan satu

**Gambar 1.4** melanjutkan alur dari connector satu pada **Gambar 1.3** dan menggambarkan proses login pengguna. Program pertama-tama membersihkan layar terminal, lalu menampilkan heading atau judul untuk menu login, diikuti dengan permintaan

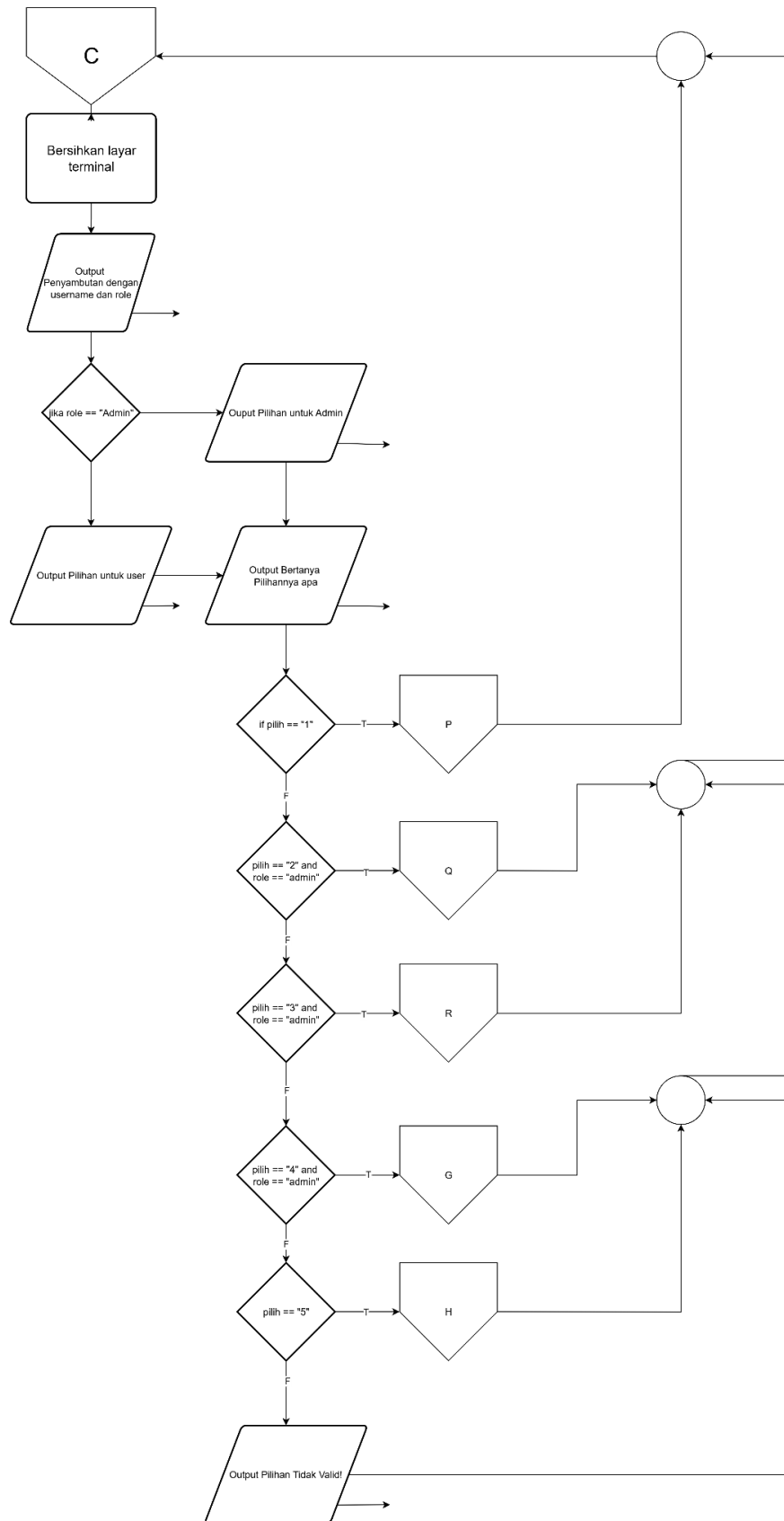
input username dan password dari pengguna. Sebelum memulai pencarian, variabel loginSukses diatur ke False sebagai nilai awal. Program kemudian melakukan perulangan untuk memeriksa setiap data pengguna yang tersimpan dalam daftar Users. Di dalam perulangan ini, program membandingkan username dan password yang dimasukkan oleh pengguna dengan setiap pasangan data di dalam daftar. Jika ditemukan kecocokan, maka proses login dikonfirmasi sebagai berhasil, status loginSukses diubah menjadi True, dan program menampilkan pesan pemberitahuan bahwa login berhasil. Setelah itu, perulangan dihentikan secara paksa menggunakan perintah Break agar tidak terus mencari data lain. Namun, jika setelah semua data diperiksa tidak ditemukan kecocokan, maka program akan menampilkan pesan pemberitahuan bahwa login gagal. Dengan demikian, **Gambar 1.4** menjelaskan logika validasi autentikasi pengguna sebelum akses ke fitur-fitur sistem diberikan.



**Gambar 1.5 pilihan dua**

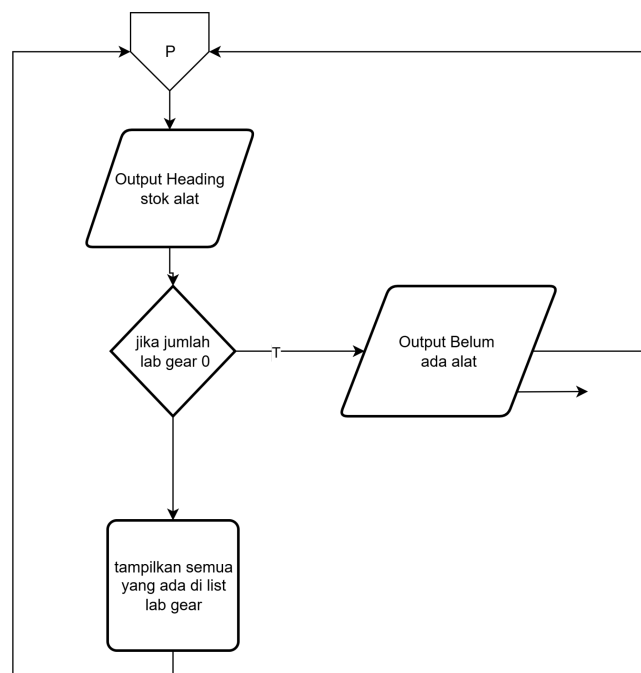
**Gambar 1.5** melanjutkan alur dari connector dua pada **Gambar 1.3** dan menggambarkan proses pendaftaran akun baru. Program dimulai dengan membersihkan layar terminal, lalu menampilkan judul menu register dan meminta pengguna untuk memasukkan username baru. Setelah input diterima, program melakukan pembersihan spasi berlebih dari input tersebut. Selanjutnya, program memeriksa apakah input username kosong; jika iya, akan ditampilkan pesan bahwa username tidak boleh kosong. Jika username tidak kosong, program kemudian memeriksa seluruh daftar pengguna yang sudah ada untuk memastikan tidak ada username yang sama. Jika ditemukan username yang sudah terdaftar, program akan menampilkan pesan bahwa username tersebut sudah ada. Jika username valid dan unik, program akan meminta pengguna untuk memasukkan password baru, lalu membersihkan spasi berlebih dari input password tersebut. Program juga memeriksa apakah password yang dimasukkan kosong; jika iya, akan ditampilkan pesan bahwa password tidak boleh kosong. Jika semua validasi terpenuhi, username dan password baru akan ditambahkan ke dalam daftar pengguna dengan peran default sebagai user, dan program akan menampilkan pesan bahwa registrasi berhasil. Dengan demikian, **Gambar 1.5** menjelaskan langkah-langkah lengkap dalam proses pendaftaran akun baru beserta validasi data yang diperlukan.





**Gambar 1.6**

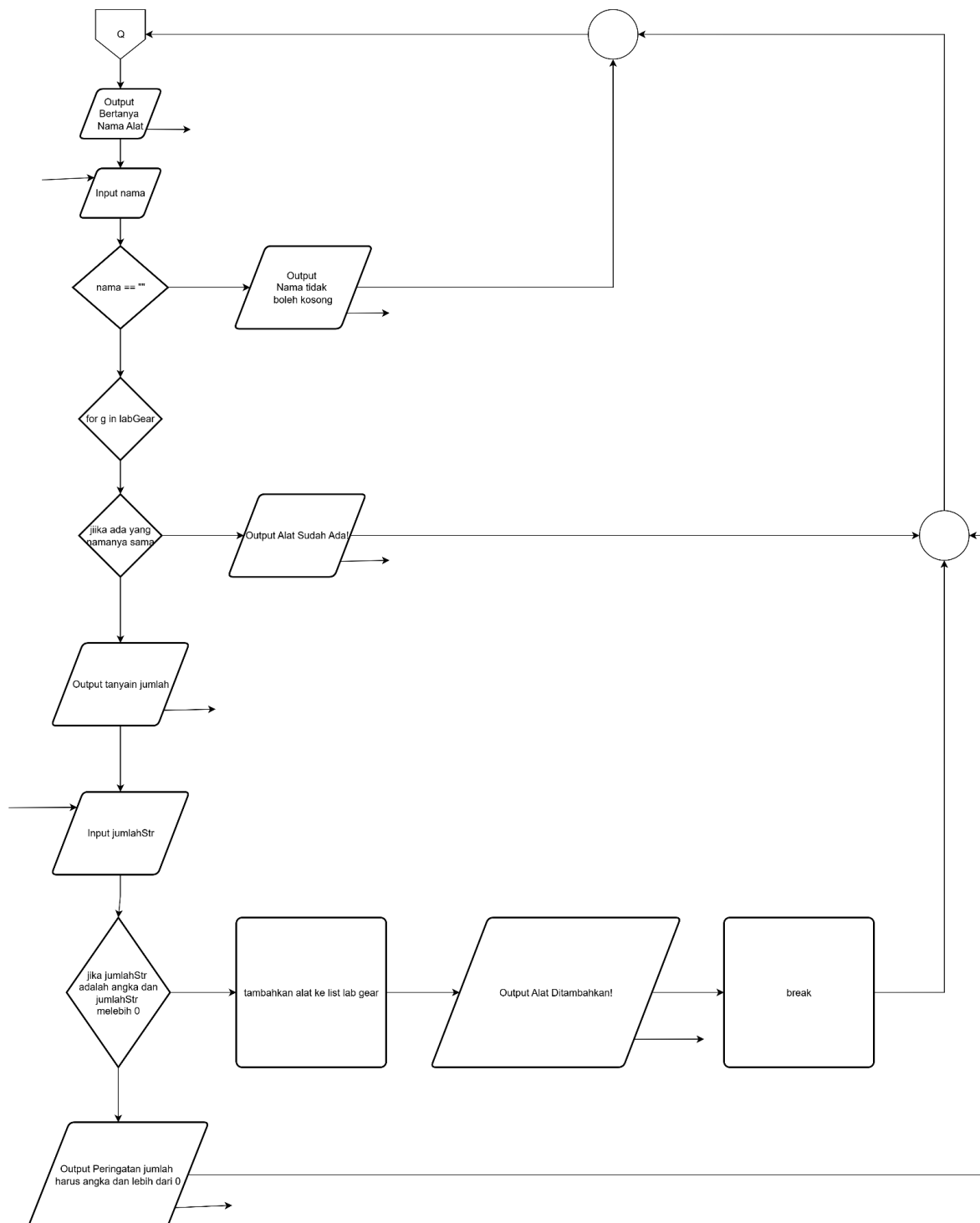
**Gambar 1.6** melanjutkan alur dari connector C pada **Gambar 1.2** dan menggambarkan menu utama setelah pengguna berhasil login. Program pertama-tama membersihkan layar terminal, lalu menampilkan pesan sambutan yang menyebutkan username dan peran (role) pengguna. Selanjutnya, program memeriksa apakah peran pengguna adalah "Admin"; jika iya, akan ditampilkan pilihan-pilihan khusus untuk admin, namun jika bukan admin, maka akan ditampilkan pilihan-pilihan standar untuk user biasa. Setelah pilihan ditampilkan, program meminta input dari pengguna dan memprosesnya. Jika pengguna memilih opsi 1, alur dialihkan ke connector P. Jika pengguna memilih opsi 2 dan perannya adalah admin, alur dialihkan ke connector Q. Jika pengguna memilih opsi 3 dan perannya adalah admin, alur dialihkan ke connector R. Jika pengguna memilih opsi 4 dan perannya adalah admin, alur dialihkan ke connector G. Jika pengguna memilih opsi 5, alur dialihkan ke connector H, yang kemungkinan besar mengarah ke proses logout atau keluar dari akun. Jika input tidak sesuai dengan semua kondisi di atas, program akan menampilkan pesan bahwa pilihan tidak valid dan kembali ke awal loop untuk meminta input ulang. Dengan demikian, Gambar 1.6 berfungsi sebagai pengendali menu utama yang membedakan fitur antara pengguna biasa dan admin.



**Gambar 1.7**

**Gambar 1.7** melanjutkan alur dari connector P pada **Gambar 1.6** dan menggambarkan proses untuk menampilkan stok alat. Program pertama-tama menampilkan judul atau heading "stok alat". Selanjutnya, program memeriksa apakah jumlah alat di dalam daftar lab gear sama dengan nol; jika iya, maka akan ditampilkan pesan bahwa belum ada alat yang terdaftar. Namun, jika jumlah alat tidak nol, program akan menampilkan seluruh data

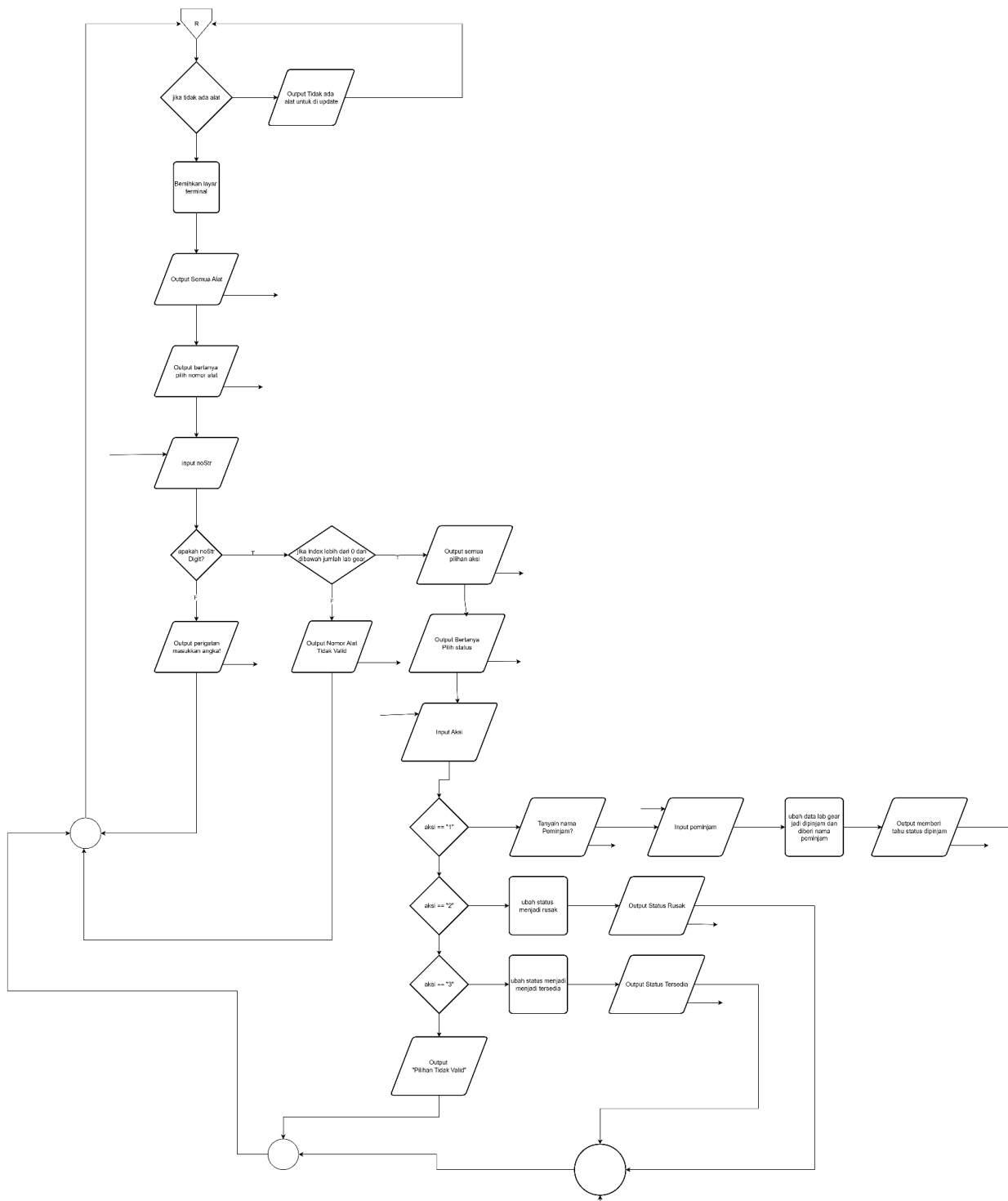
alat yang ada di dalam daftar lab gear. Dengan demikian, **Gambar 1.7** menjelaskan logika tampilan stok alat yang memastikan pengguna diberi informasi sesuai dengan kondisi data yang ada, baik ketika stok kosong maupun ketika ada alat yang tersedia.



**Gambar 1.8**

**Gambar 1.8** melanjutkan alur dari connector Q pada **Gambar 1.6** dan menggambarkan proses penambahan alat baru oleh admin. Program pertama-tama meminta admin untuk memasukkan nama alat, lalu memeriksa apakah input nama kosong; jika iya, akan ditampilkan pesan bahwa nama tidak boleh kosong. Jika nama tidak kosong, program

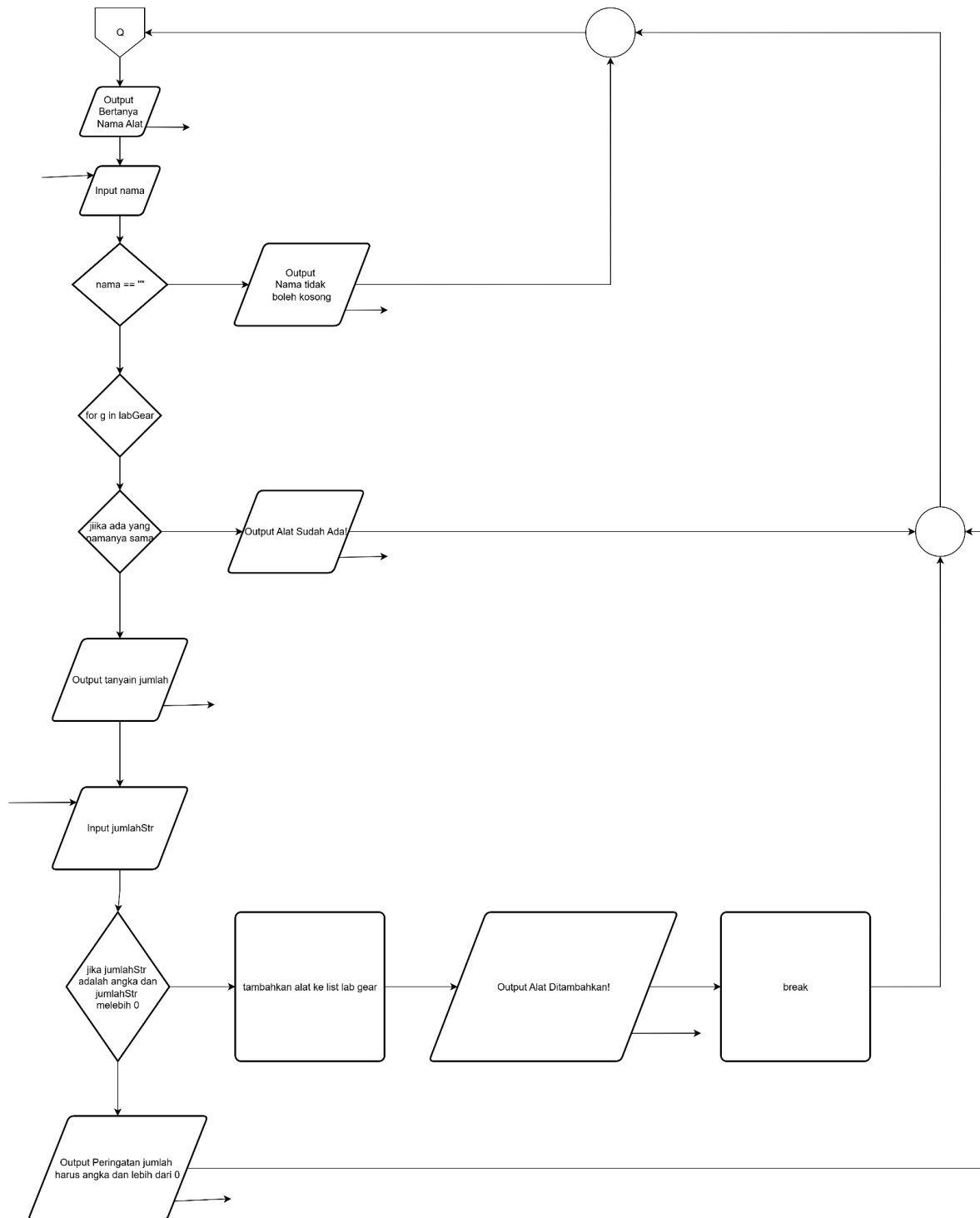
kemudian memeriksa seluruh daftar alat yang sudah ada untuk memastikan tidak ada nama alat yang sama; jika ditemukan nama yang sudah ada, akan ditampilkan pesan bahwa alat tersebut sudah terdaftar. Jika nama alat unik, program akan meminta admin untuk memasukkan jumlah alat, lalu memeriksa apakah input jumlah adalah angka dan lebih besar dari nol; jika tidak, akan ditampilkan peringatan bahwa jumlah harus berupa angka dan lebih dari 0. Jika semua validasi terpenuhi, alat baru dengan nama dan jumlah yang dimasukkan akan ditambahkan ke dalam daftar lab gear, program akan menampilkan pesan bahwa alat berhasil ditambahkan, dan proses penambahan dihentikan dengan perintah break. Dengan demikian, Gambar 1.8 menjelaskan langkah-langkah lengkap dalam proses penambahan alat beserta validasi data yang diperlukan untuk memastikan konsistensi dan keakuratan data stok.



**Gambar 1.9**

**Gambar 1.9** melanjutkan alur dari connector R pada Gambar 1.6 dan menggambarkan proses pembaruan atau update status alat oleh admin. Program pertama-tama memeriksa apakah ada alat yang terdaftar; jika tidak ada, akan ditampilkan pesan bahwa tidak ada alat untuk diupdate. Jika ada alat, program membersihkan layar, menampilkan daftar semua alat, lalu meminta admin untuk memilih nomor alat yang ingin diupdate. Program kemudian memvalidasi input nomor tersebut: jika input bukan angka, akan muncul peringatan untuk memasukkan angka; jika angka yang dimasukkan tidak valid

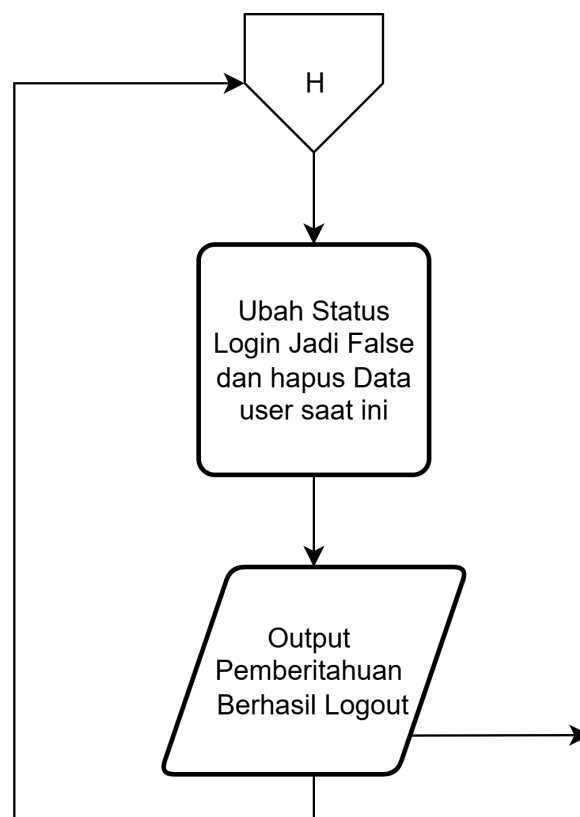
(misalnya kurang dari 1 atau melebihi jumlah alat), akan ditampilkan pesan bahwa nomor alat tidak valid. Setelah nomor alat divalidasi, program menampilkan pilihan aksi yang bisa dilakukan: meminjamkan alat, merusakkan alat, atau menjadikannya tersedia kembali. Jika admin memilih opsi 1, program akan meminta nama peminjam, lalu mengubah status alat menjadi "dipinjam" dan mencatat nama peminjam, disertai pesan konfirmasi. Jika memilih opsi 2, status alat akan diubah menjadi "rusak" dan ditampilkan pesan status rusak. Jika memilih opsi 3, status alat akan diubah menjadi "tersedia" dan ditampilkan pesan status tersedia. Jika admin memilih opsi lain selain ketiga pilihan tersebut, akan ditampilkan pesan bahwa pilihan tidak valid. Dengan demikian, Gambar 1.9 menjelaskan logika lengkap untuk memperbarui status alat, termasuk validasi input dan penanganan berbagai jenis aksi yang dapat dilakukan oleh admin.



**Gambar 1.10**

**Gambar 1.10** melanjutkan alur dari connector Q pada **Gambar 1.6** dan menggambarkan proses penambahan alat baru oleh admin. Program pertama-tama meminta admin untuk memasukkan nama alat, lalu memeriksa apakah input nama kosong; jika iya, akan ditampilkan pesan bahwa nama tidak boleh kosong. Jika nama tidak kosong, program kemudian memeriksa seluruh daftar alat yang sudah ada untuk memastikan tidak ada nama

alat yang sama; jika ditemukan nama yang sudah ada, akan ditampilkan pesan bahwa alat tersebut sudah terdaftar. Jika nama alat unik, program akan meminta admin untuk memasukkan jumlah alat, lalu memeriksa apakah input jumlah adalah angka dan lebih besar dari nol; jika tidak, akan ditampilkan peringatan bahwa jumlah harus berupa angka dan lebih dari 0. Jika semua validasi terpenuhi, alat baru dengan nama dan jumlah yang dimasukkan akan ditambahkan ke dalam daftar lab gear, program akan menampilkan pesan bahwa alat berhasil ditambahkan, dan proses penambahan dihentikan dengan perintah break. Dengan demikian, **Gambar 1.10** menjelaskan langkah-langkah lengkap dalam proses penambahan alat beserta validasi data yang diperlukan untuk memastikan konsistensi dan keakuratan data stok.



**Gambar 1.11**

**Gambar 1.11** melanjutkan alur dari connector H pada Gambar 1.6 dan menggambarkan proses logout pengguna. Program pertama-tama mengubah status login menjadi False untuk menandakan bahwa pengguna telah keluar, lalu menghapus data pengguna yang sedang aktif dari memori atau sesi program agar tidak bisa diakses lagi. Setelah itu, program menampilkan pesan pemberitahuan bahwa proses logout telah berhasil dilakukan. Dengan demikian, Gambar 1.11 menjelaskan langkah sederhana namun penting dalam mengakhiri sesi pengguna secara aman sebelum kembali ke menu awal atau menutup program.



## 2. Deskripsi Singkat Program

Program ini merupakan sistem manajemen laboratorium berbasis teks yang memungkinkan dua jenis pengguna admin dan user biasa untuk berinteraksi dengan data alat laboratorium. Pengguna dapat mendaftar, login, melihat stok alat, serta (jika berstatus admin) menambahkan alat baru dan memperbarui status alat (misalnya: dipinjam, rusak, atau tersedia). Program menggunakan struktur menu berbasis loop dengan validasi input yang ketat untuk memastikan keakuratan data dan pengalaman pengguna yang terkendali. Sesi pengguna diatur melalui status login, dan sistem secara otomatis kembali ke menu awal setelah logout atau menutup program secara rapi saat pengguna memilih keluar.

## 3. Source Code

### A. Fitur Pilihan

```
print("Selamat Datang Di Lab GEAR")
print("1. Login")
print("2. Register")
print("3. Keluar")
pilih = input("Pilihan : ").strip()
```

Fitur ini merupakan bagian dari menu utama sebelum login yang berfungsi untuk menyambut pengguna dan memberikan pilihan aksi awal. Pada fitur ini, program menampilkan teks Selamat Datang Di Lab GEAR beserta tiga opsi: 1 Login, 2 Register, dan 3 Keluar. Pengguna diminta memasukkan pilihan melalui perintah `input("Pilihan : ")`, dan metode `strip()` digunakan untuk membersihkan spasi kosong di awal atau akhir input. Penggunaan `strip()` merupakan bagian penting dari fitur ini karena meningkatkan ketahanan program terhadap kesalahan input, seperti ketikan spasi berlebih, sehingga memastikan validasi pilihan berjalan dengan akurat. Fitur ini menjadi pintu masuk utama bagi pengguna baru maupun yang sudah terdaftar untuk mulai berinteraksi dengan sistem.

## B. Fitur Login

```
if pilih == "1":
    # Login
    os.system("cls || clear")
    print("===LOGIN===")
    inpUsername = input("Username : ")
    inpPassword = input("Password : ")

    loginSukses = False
    for u in Users:
        if u[0] == inpUsername and u[1] == inpPassword:
            sedangLogin = True
            role = u[2]
            username = inpUsername
            loginSukses = True
            print("Login Berhasil!")
            input("Tekan Enter... ")
            break

    if not loginSukses:
        print("Gagal Login!")
        input("Tekan Enter ...")
```

Fitur Login berfungsi untuk mengautentikasi pengguna sebelum dapat mengakses menu utama sistem. Saat pengguna memilih opsi 1 di menu awal, program akan membersihkan layar lalu menampilkan form login yang meminta username dan password. Input dari pengguna kemudian dibandingkan dengan data pengguna yang tersimpan dalam daftar Users. Jika ditemukan kecocokan antara username dan password, status login diaktifkan (`sedangLogin = True`), peran (`role`) dan username pengguna disimpan, serta muncul pesan Login Berhasil. Program kemudian menunggu pengguna menekan Enter sebelum melanjutkan. Jika tidak ditemukan data yang cocok, program menampilkan pesan Gagal Login dan juga menunggu konfirmasi dari pengguna. Fitur ini memastikan hanya pengguna terdaftar yang dapat masuk ke sistem, sekaligus menentukan hak akses berdasarkan perannya (admin atau user biasa).

## C. Fitur Register

```

elif pilih == "2":
    # Register
    os.system("cls || clear")
    print("===REGISTER===")
    newUser = input("Username Baru : ").strip()
    if newUser == "":
        print("Username tidak boleh kosong")
    else:
        usernameAda = False
        for u in Users:
            if u[0] == newUser:
                usernameAda = True
        if usernameAda:
            print("Username sudah dipakai!")
        else:
            newPass= input("Password : ").strip()
            if newPass == "":
                print("Password tidak boleh kosong")
            else:
                Users.append([newUser,newPass,"user"])
                print("Registrasi Berhasil")
        input("tekan enter... ")

```

Fitur register memungkinkan pengguna baru untuk membuat akun pada sistem. Saat memilih opsi 2 di menu awal, program membersihkan layar dan menampilkan form registrasi. Pengguna diminta memasukkan username baru, yang kemudian diverifikasi agar tidak kosong dan belum digunakan oleh pengguna lain. Jika username valid, pengguna diminta memasukkan password, yang juga diperiksa agar tidak kosong. Jika kedua data memenuhi syarat, akun baru akan ditambahkan ke daftar Users dengan peran default sebagai user. Setelah proses selesai, baik berhasil maupun gagal, program menampilkan pesan status dan menunggu pengguna menekan Enter sebelum kembali ke menu utama. Fitur ini memastikan pendaftaran akun berjalan aman dan terstruktur dengan validasi input yang memadai.

#### E. Fitur CRUD

```

# Menu Setelah Login
os.system("cls || clear")
print(f"====Masuk Sebagai: {username} ({role})====")
print("1. Lihat Stok")
if role == "Admin":

```

```
print("2. Tambah Alat")
print("3. Update Status")
print("4. Hapus Alat")
print("5. Logout")
pilih = input("Pilih : ").strip()
```

Fitur ini merupakan bagian dari menu utama setelah pengguna berhasil login dan berfungsi sebagai antarmuka untuk mengakses fitur-fitur CRUD (Create, Read, Update, Delete) dalam sistem manajemen laboratorium. Program pertama-tama membersihkan layar dan menampilkan identitas pengguna beserta perannya. Semua pengguna, baik admin maupun user biasa, dapat memilih opsi 1 untuk melihat stok alat (Read). Namun, opsi tambahan yaitu Tambah Alat (Create), Update Status (Update), dan Hapus Alat (Delete) hanya ditampilkan dan dapat diakses jika peran pengguna adalah Admin. Opsi 5 tersedia untuk semua pengguna guna melakukan logout. Dengan struktur ini, fitur ini menerapkan kontrol akses berbasis peran, memastikan bahwa hanya admin yang dapat melakukan perubahan data, sementara user biasa hanya dapat melihat informasi stok.

#### 4. Hasil Output

```
Selamat Datang Di Lab GEAR
1. Login
2. Register
3. Keluar
Pilihan : █
```

```
===LOGIN===
Username : admin
Password : admin123
Login Berhasil!
Tekan Enter... █
```

```
====Masuk Sebagai: admin (Admin)====
1. Lihat Stok
2. Tambah Alat
3. Update Status
4. Hapus Alat
5. Logout
Pilih : █
```

```
===REGISTER===
Username Baru : sigma
Password : sigma
Registrasi Berhasil
tekan enter... █
```

## 5. Langkah-langkah GIT

### 5.1 GIT Add

```
PS C:\Users\Yiban\Documents\posttest\2\praktikum-apd> git add .
```

Simpan Perubahan

### 5.2 GIT Commit

```
(.venv) PS C:\Users\Yiban\Documents\posttest\2\praktikum-apd> git commit -m "posttest5"
[main 9f39e1d] posttest5
8 files changed, 1790 insertions(+)
create mode 100644 post-test/post-test-apd-4/flowchartimage/Untitled Diagram.drawio.svg
create mode 100644 post-test/post-test-apd-5/2509106021-MuhammadZidaneAbdulkadir-PT-5.pdf
create mode 100644 post-test/post-test-apd-5/2509106021-MuhammadZidaneAbdulkadir-PT-5.py
create mode 100644 post-test/post-test-apd-5/__pycache__/data.cpython-313.pyc
create mode 100644 post-test/post-test-apd-5/__pycache__/status.cpython-313.pyc
create mode 100644 post-test/post-test-apd-5/data.py
create mode 100644 post-test/post-test-apd-5/posttest5.drawio
create mode 100644 post-test/post-test-apd-5/status.py
```

tambahkan perubahan pada repositori

### 5.3 GIT Push

```
• (.venv) PS C:\Users\Yiban\Documents\posttest\2\praktikum-apd> git push
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 16 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 1.28 MiB | 3.19 MiB/s, done.
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To github.com:ExtraYiban/praktikum-apd.git
   c253347..9f39e1d  main -> main
```

Kirim Perubahan Ke Cloud (Github)