LAPORAN PRAKTIKUM

POSTTEST 8

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Muhammad Zidane Abdul Kadir (2509106021)

Kelas (A1 ‘25)

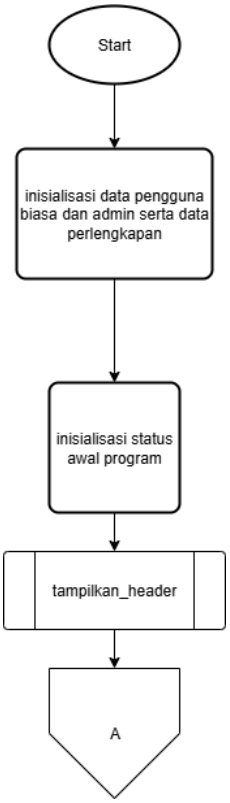
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

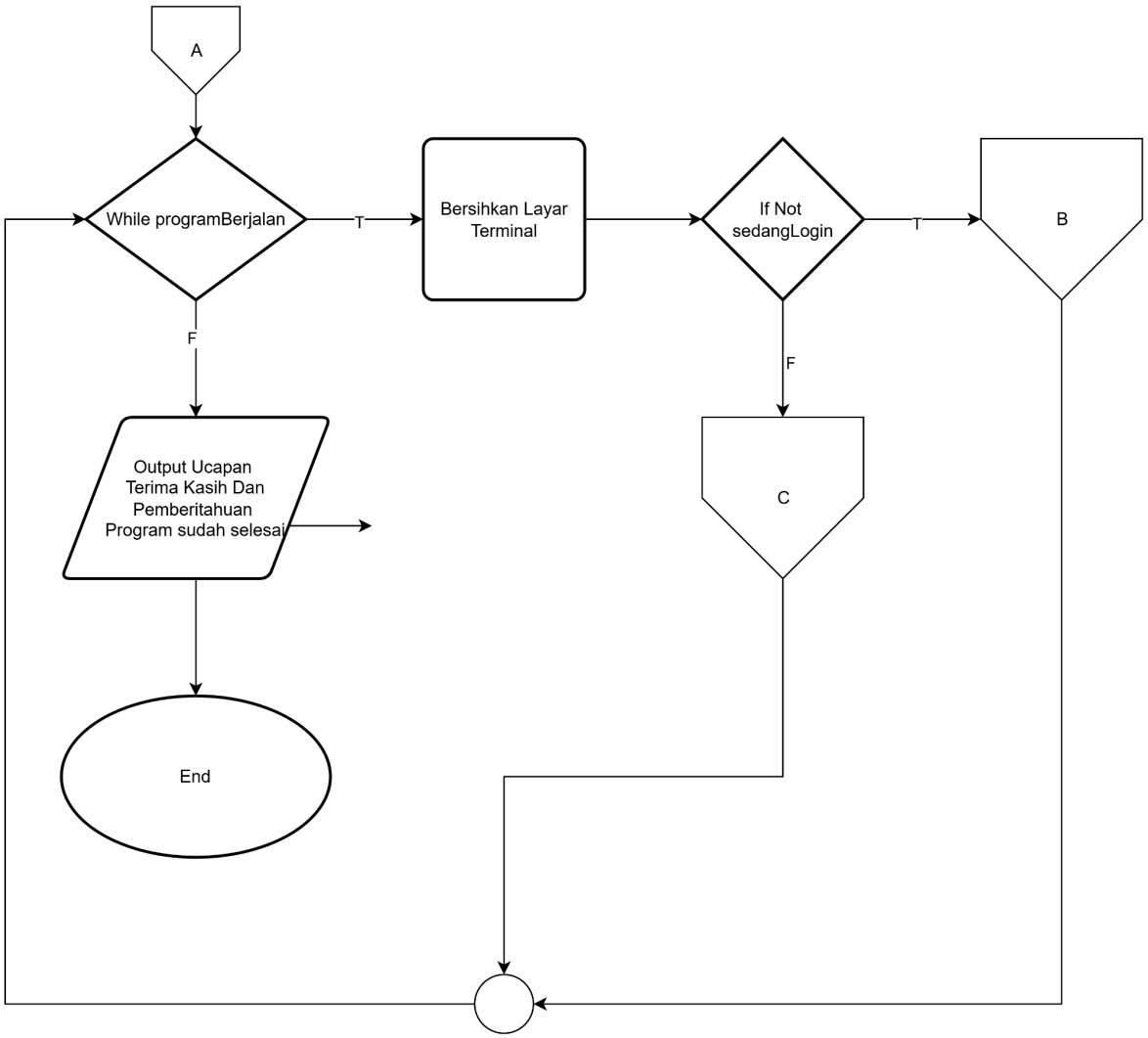
2025

1. Flowchart



Gambar 1.1 Inisialisasi

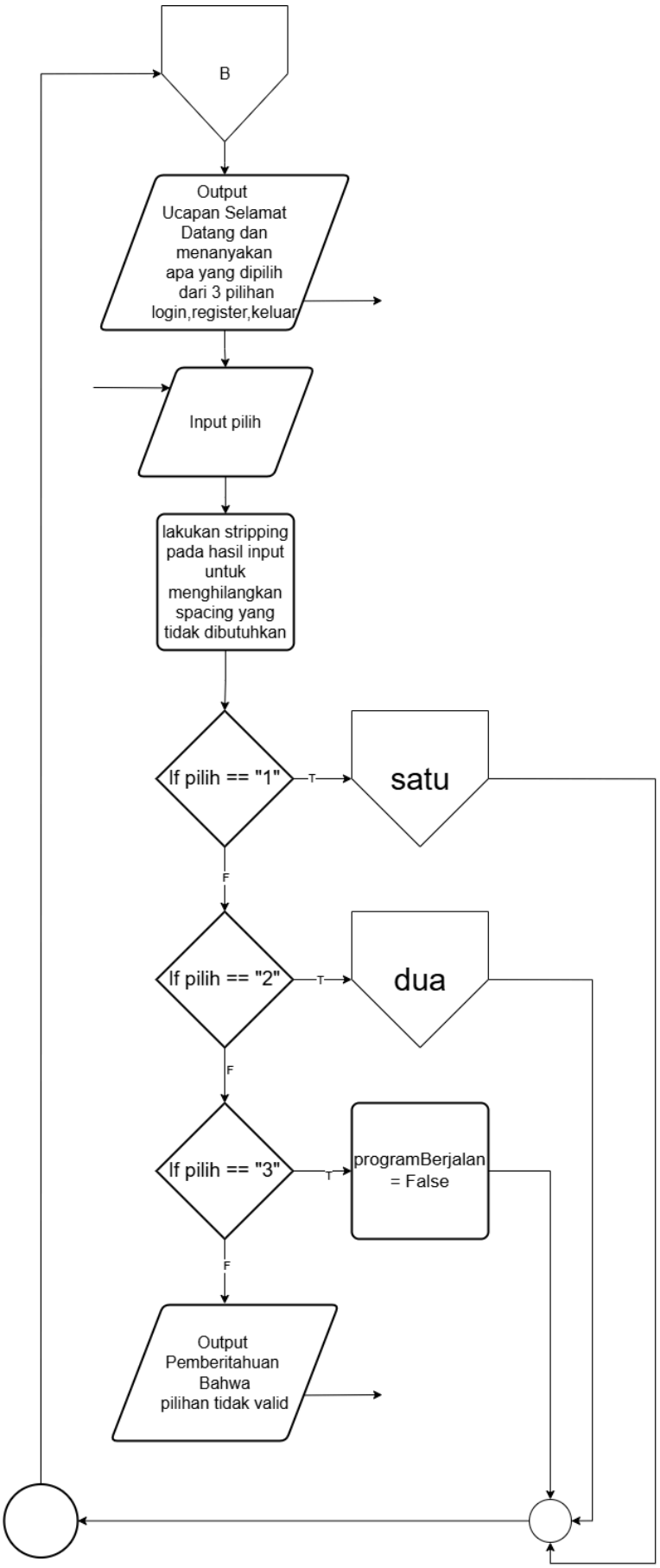
Gambar 1.1 menunjukkan diagram alir yang menjelaskan proses inisialisasi awal program sebelum memasuki bagian utama. Proses dimulai dari simbol “Start”, kemudian program melakukan inisialisasi data pengguna biasa dan admin serta data perlengkapan yang akan digunakan dalam sistem. Selanjutnya, program melakukan inisialisasi status awal program, yaitu pengaturan nilai awal variabel atau kondisi yang dibutuhkan agar program dapat berjalan dengan benar. Setelah status awal diatur, program menjalankan fungsi tampilkan_header untuk menampilkan tampilan awal atau judul program di layar. Alur kemudian berlanjut ke simbol konektor “A” yang menjadi penghubung ke bagian berikutnya dari flowchart untuk melanjutkan proses utama program. Tahapan ini memastikan bahwa seluruh data, status, dan tampilan awal telah siap sebelum program masuk ke logika inti.



Gambar 1.2 Proses Looping

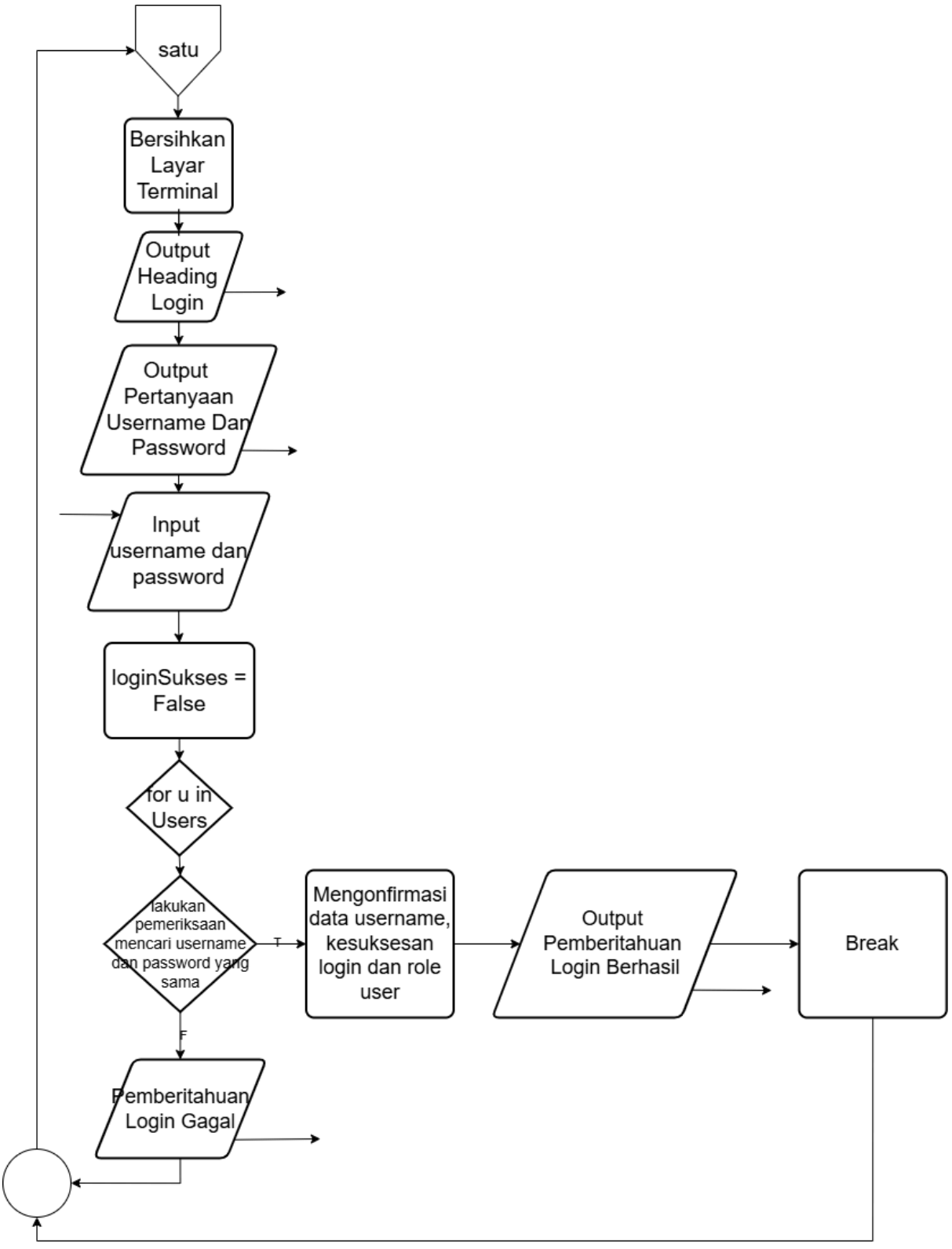
Gambar 1.2 melanjutkan alur program dari **Gambar 1.1** melalui connector “A”, dan menggambarkan struktur utama pengendali program berbasis loop serta pengecekan status login. Program memasuki perulangan “While programBerjalan” yang akan terus berjalan selama kondisi tersebut bernilai benar (True), di 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

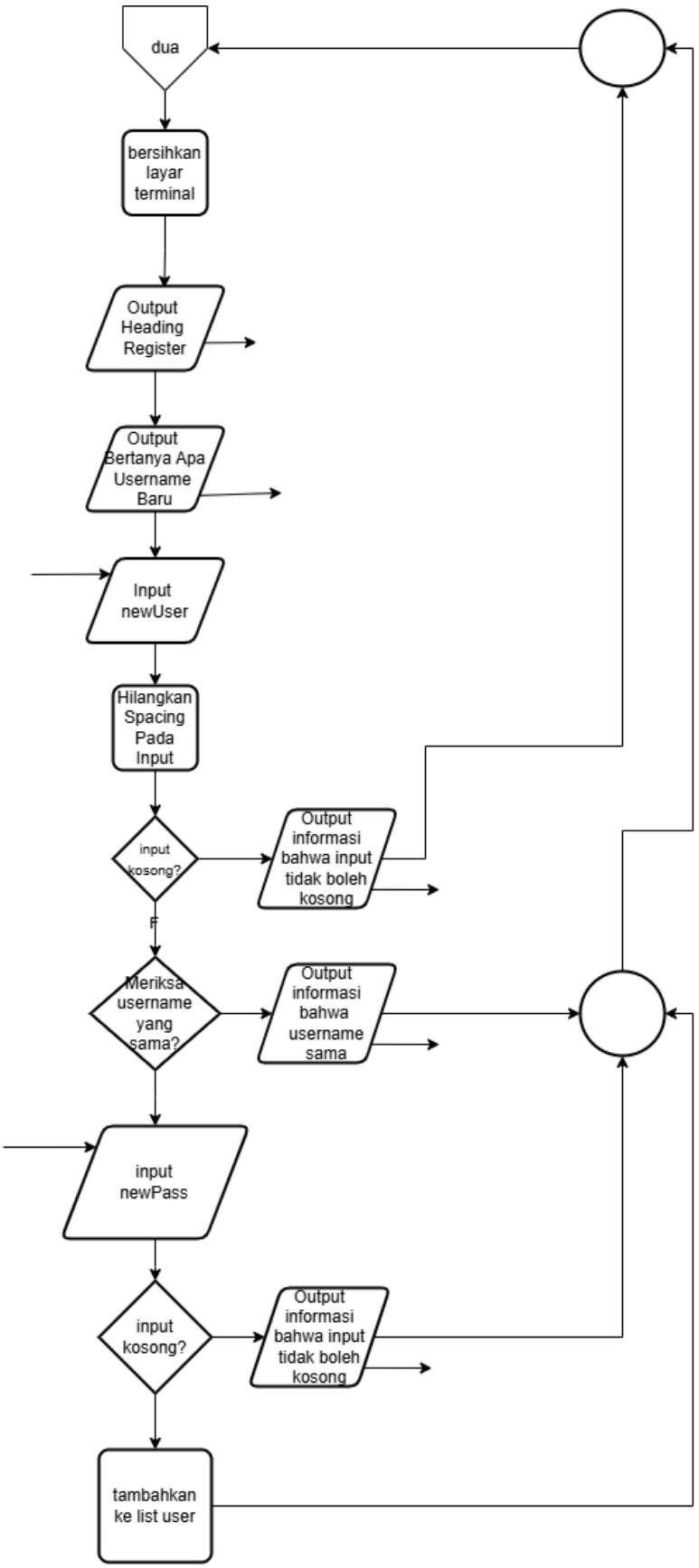
Gambar 1.3 melanjutkan alur dari connector B pada **Gambar 1.2** dan menggambarkan proses menu utama sebelum login. Program menampilkan pesan selamat datang dan memberikan tiga pilihan kepada pengguna: login, register, atau keluar. Setelah pengguna memasukkan pilihannya, input tersebut dibersihkan dari 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

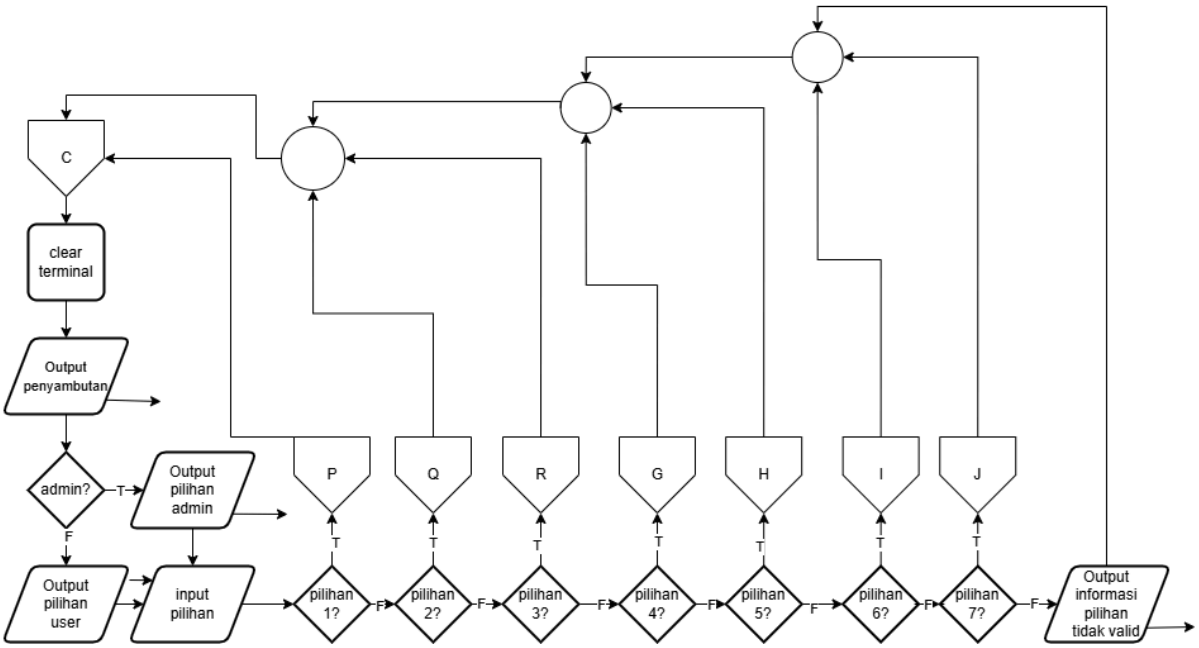
Gambar 1.4 melanjutkan alur dari connector satu pada **Gambar 1.3** dan menggambarkan proses login pengguna. Program pertama-tama membersihkan layar terminal, lalu menampilkan heading atau judul untuk menu login, diikuti dengan permintaan input username dan password dari pengguna. Sebelum memulai pencarian,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



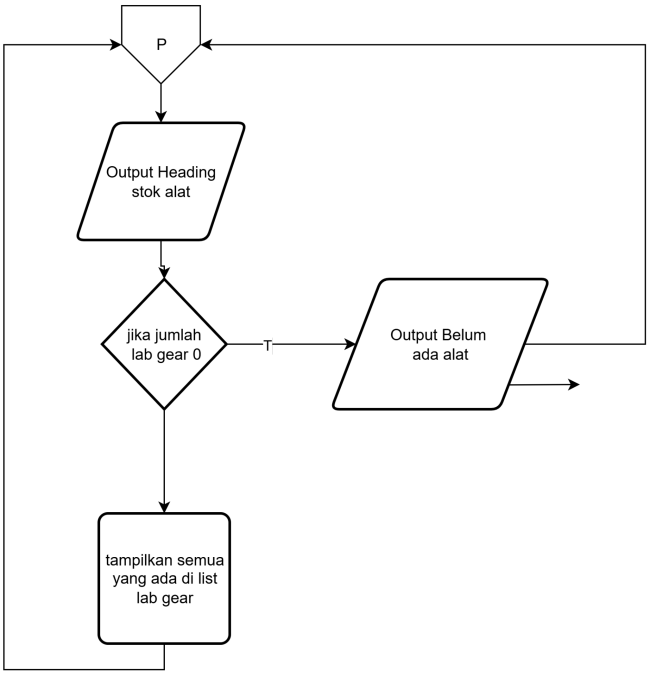
Gambar 1.5 pilihan dua

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.



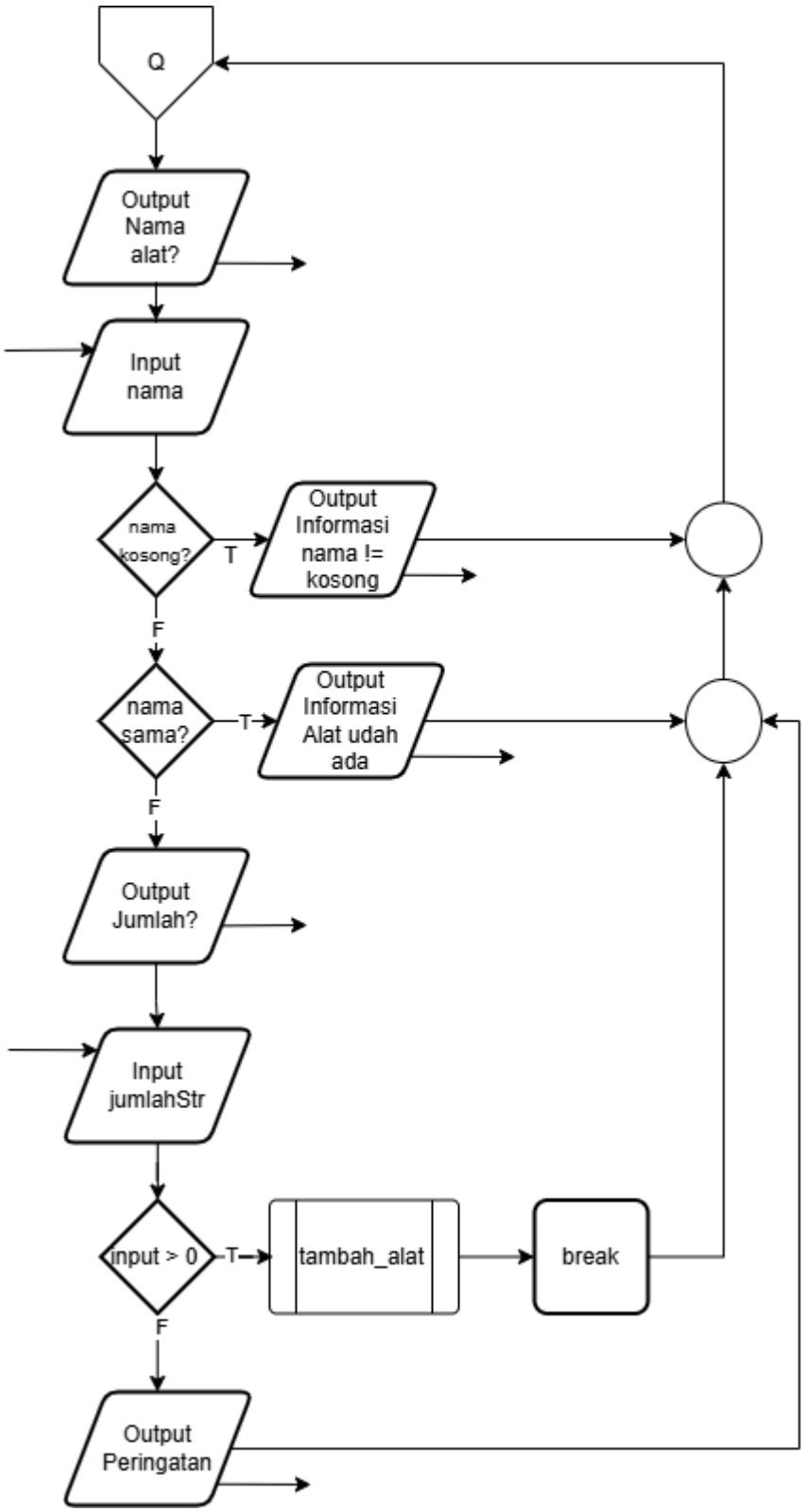
Gambar 1.6

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

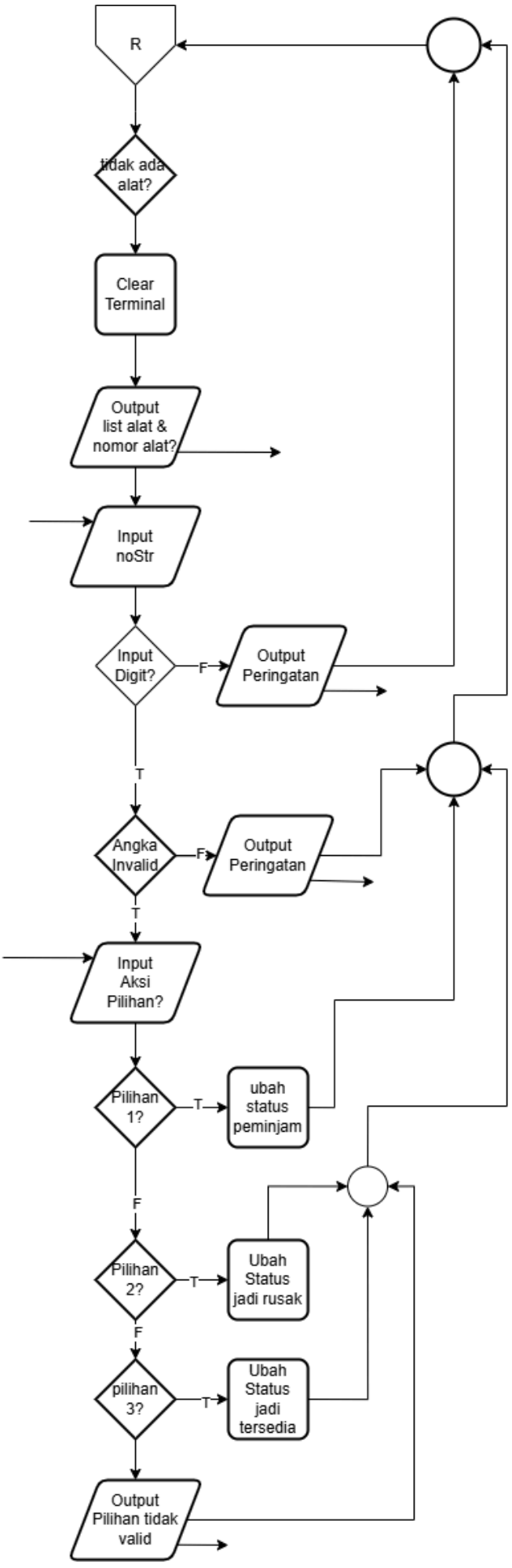
Gambar 1.7 melanjutkan alur dari connector P pada **Gambar 1.6** dan menggambarkan proses untuk menampilkan stok alat. Program pertama-tama menampilkan judul atau heading "stok alat". Selanjutnya, program memeriksa apakah jumlah alat di dalam daftar lab gear sama dengan nol; jika iya, maka akan ditampilkan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

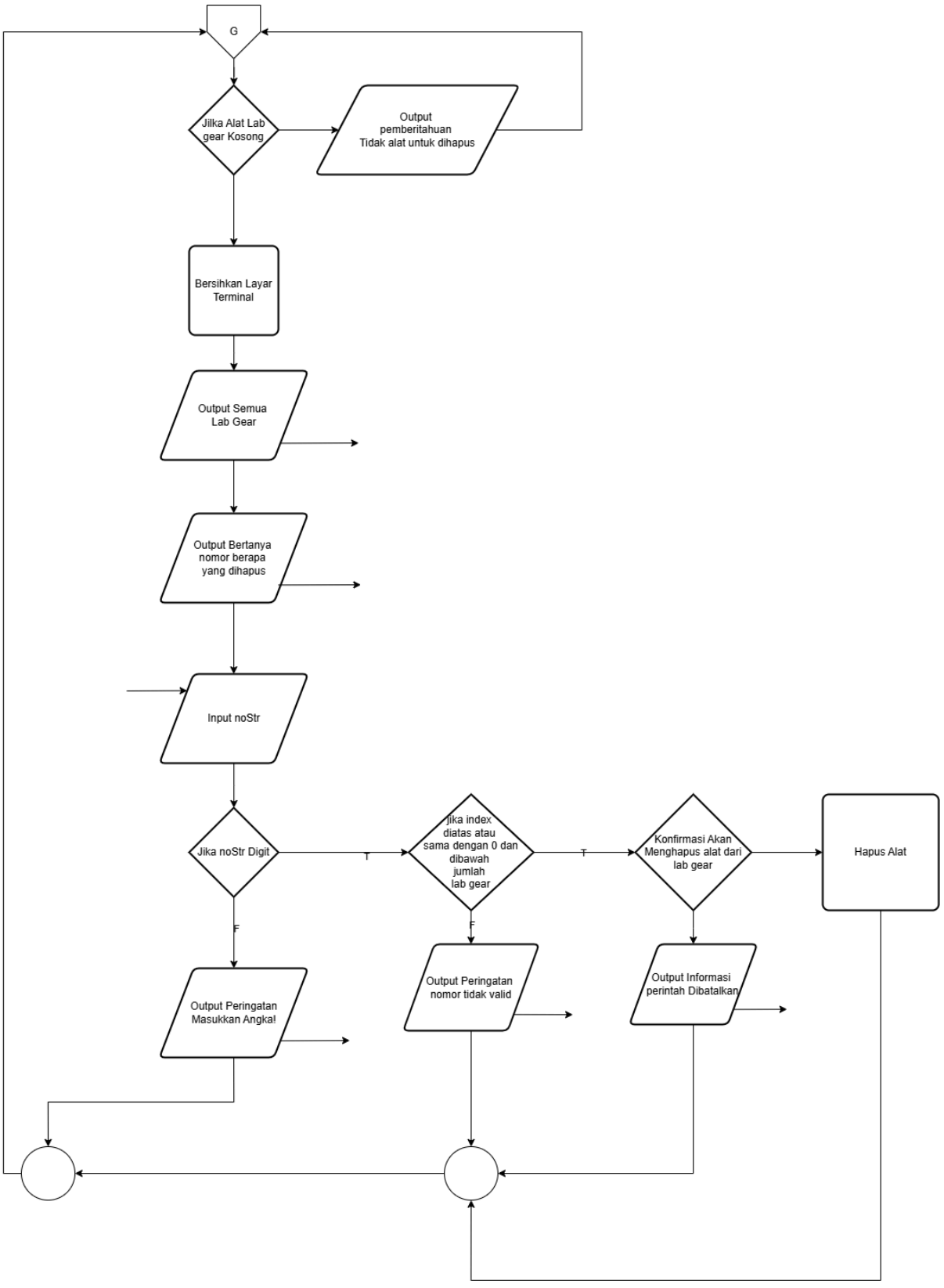
Gambar 1.8 melanjutkan alur dari connector Q pada **Gambar 1.6** dan menggambarkan proses penambahan alat baru oleh admin. Program pertama-tama meminta admin untuk memasukkan nama alat, lalu memeriksa apakah input nama kosong; jika iya, akan ditampilkan pesan bahwa nama tidak boleh kosong. Jika nama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


R
tidak ada alat?
Clear Terminal
Output list alat & nomor alat?
Input noStr
Input Digit?
F
Output Peringatan
T
Angka Invalid
F
Output Peringatan
T
Input Aksi Pilihan?
Pilihan 1?
T
ubah status peminjam
F
Pilihan 2?
T
Ubah Status jadi rusak
F
pilihan 3?
T
Ubah Status jadi tersedia
Output Pilihan tidak valid

Gambar 1.9

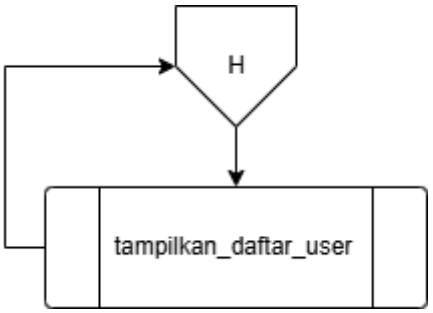
Gambar 1.9 melanjutkan alur dari connector R pada Gambar 1.6 dan menggambarkan proses pembaruan atau update status alat oleh admin. Program pertama-tama memeriksa apakah ada alat yang terdaftar; jika tidak ada, akan ditampilkan pesan bahwa tidak ada alat untuk diupdate. Jika ada alat, program membersihkan 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

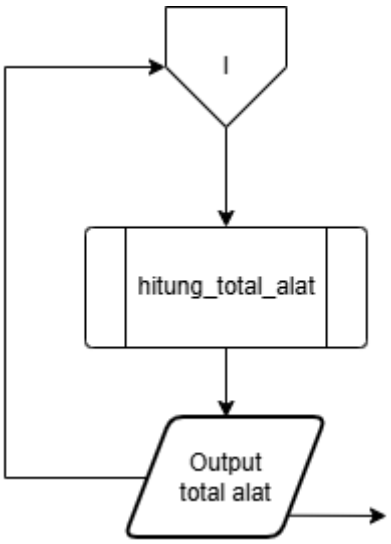
Gambar 1.10 menunjukkan alur proses penghapusan alat laboratorium (Lab Gear) oleh admin. Proses ini dimulai dari connector G, yang kemudian memeriksa apakah daftar

alat laboratorium kosong. Jika tidak ada alat, program menampilkan pesan pemberitahuan bahwa tidak ada alat yang dapat dihapus. Apabila daftar alat tidak kosong, sistem akan membersihkan layar terminal dan menampilkan seluruh daftar Lab Gear yang tersedia. Setelah itu, program menampilkan permintaan input kepada admin untuk menentukan nomor alat yang ingin dihapus. Input tersebut disimpan dalam variabel *noStr*, lalu diperiksa apakah merupakan digit angka yang valid. Jika bukan angka, maka program akan menampilkan peringatan untuk memasukkan angka. Jika input berupa angka, program selanjutnya memeriksa apakah nomor alat berada dalam rentang yang valid, yaitu antara 1 hingga jumlah alat yang tersedia. Bila nomor tidak valid, akan muncul peringatan bahwa nomor tidak valid. Jika semua pemeriksaan valid, program akan menampilkan konfirmasi apakah admin benar-benar ingin menghapus alat tersebut. Jika admin mengonfirmasi, maka sistem akan menghapus alat dari daftar Lab Gear. Namun, jika admin membatalkan, maka program menampilkan informasi bahwa perintah dibatalkan. Setelah salah satu kondisi tersebut, alur kembali ke proses sebelumnya. Dengan demikian, Gambar 1.10 menggambarkan secara lengkap mekanisme penghapusan alat laboratorium, mencakup pemeriksaan data, validasi input, hingga konfirmasi penghapusan untuk memastikan tindakan dilakukan dengan benar dan aman



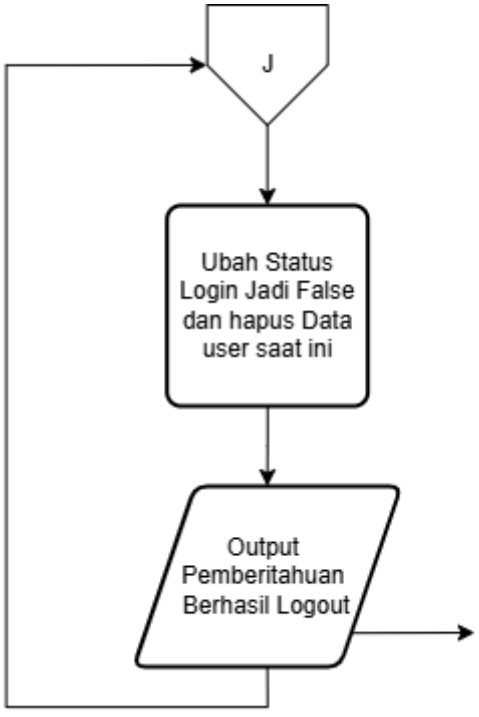
Gambar 1.11

Gambar 1.11 melanjutkan alur dari connector H pada Gambar 1.6 dan menggambarkan proses penampilan daftar pengguna (user). Pada bagian ini, program diarahkan untuk menjalankan prosedur tampilkan_daftar_user, yaitu menampilkan seluruh data pengguna yang tersimpan dalam sistem. Proses ini tidak melibatkan input atau percabangan logika tambahan, melainkan hanya berfungsi untuk menampilkan informasi pengguna secara lengkap sesuai dengan data yang telah ada.



Gambar 1.12

Gambar 1.12 melanjutkan alur dari connector I pada Gambar 1.6 dan menggambarkan proses perhitungan total alat laboratorium yang terdapat dalam sistem. Pada tahap ini, program menjalankan prosedur hitung_total_alat untuk menjumlahkan seluruh jumlah alat yang terdaftar pada data Lab Gear. Setelah proses perhitungan selesai, sistem akan menampilkan output berupa total keseluruhan alat yang ada. Dengan demikian, Gambar 1.12 menjelaskan mekanisme perhitungan dan penampilan total alat laboratorium secara otomatis, yang berguna bagi admin untuk memantau jumlah keseluruhan alat yang tersedia di dalam sistem.



Gambar 1.13

Gambar 1.13 melanjutkan alur dari connector J pada Gambar 1.6 dan menggambarkan proses logout pengguna dari sistem. Pada tahap ini, program akan mengubah status login menjadi False dan menghapus data pengguna yang sedang aktif untuk menandakan bahwa pengguna telah keluar dari sistem. Setelah itu, program menampilkan output berupa pemberitahuan bahwa proses logout telah berhasil dilakukan. Dengan demikian, Gambar 1.13 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(INFO + "1. Login")
        print(INFO + "2. Register")
        print("3. Keluar" + RESET)
        try:
            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":
                os.system("cls || clear")
                print(TITLE + "=== LOGIN ===" + RESET)
                usernameInput = input("Username : ").strip()
                passwordInput = input("Password : ").strip()

                if usernameInput in Users and
Users[usernameInput]["password"] == passwordInput:
                    sedangLogin = True
                    username = usernameInput
                    role = Users[usernameInput]["role"]
                    print(SUCCESS + "Login Berhasil!" + RESET)
                else:
                    print(ERROR + "Login gagal! Username atau
password salah." + RESET)
                input("Tekan Enter...")
```

Fitur login digunakan untuk memverifikasi identitas pengguna sebelum masuk ke sistem. Saat pengguna memilih menu 1, program akan membersihkan tampilan terminal menggunakan perintah os.system("cls || clear") agar layar terlihat rapi. Setelah itu program menampilkan judul “=== LOGIN ===” dan meminta pengguna untuk memasukkan username serta password. Data yang dimasukkan akan dicek di dictionary Users yang berisi daftar akun lengkap dengan password dan perannya. Jika username ditemukan dan password yang dimasukkan cocok, maka variabel sedangLogin diubah menjadi True sebagai tanda bahwa pengguna berhasil login. Username dan role juga disimpan agar sistem bisa mengetahui siapa yang sedang login dan apa hak aksesnya. Program kemudian menampilkan pesan “Login Berhasil!” dan menunggu pengguna menekan Enter untuk melanjutkan. Namun jika username tidak ditemukan atau password salah, maka program menampilkan pesan “Login gagal! Username atau password salah.” dan juga menunggu pengguna menekan Enter. Fitur

ini memastikan bahwa hanya pengguna yang terdaftar dengan data yang benar yang bisa masuk ke dalam sistem.

C. Fitur Register

```
elif pilih == "2":
                os.system("cls || clear")
                print(TITLE + "=== REGISTER ===" + RESET)
                newUser = input("Username Baru : ").strip()

                if newUser == "":
                    print(ERROR + "Username tidak boleh kosong!" +
RESET)
                elif newUser in Users:
                    print(WARNING + "Username sudah digunakan!" +
RESET)
                else:
                    newPass = input("Password : ").strip()
                    if newPass == "":
                        print(ERROR + "Password tidak boleh
kosong!" + RESET)
                    else:
                        Users[newUser] = {"password": newPass,
"role": "user"}
                        print(SUCCESS + "Registrasi Berhasil!" +
RESET)
                input("Tekan Enter...")
```

Fitur register digunakan untuk menambahkan akun baru ke dalam sistem. Saat pengguna memilih menu 2, program akan membersihkan layar menggunakan perintah os.system("cls || clear") lalu menampilkan judul “=== REGISTER ===”. Setelah itu pengguna diminta untuk memasukkan username baru. Program kemudian memeriksa apakah input username kosong, jika kosong maka akan muncul pesan “Username tidak boleh kosong!”. Jika username sudah ada di dalam dictionary Users, maka akan muncul pesan “Username sudah digunakan!”. Namun jika username valid dan belum terdaftar, program akan meminta pengguna untuk memasukkan password. Password juga dicek agar tidak kosong, dan jika semua data sudah benar, akun baru akan disimpan ke dalam dictionary Users dengan struktur {"password": newPass, "role": "user"}. Setelah proses selesai, muncul pesan “Registrasi Berhasil!” dan program menunggu pengguna menekan Enter untuk melanjutkan. Fitur ini memungkinkan pengguna baru untuk membuat akun sendiri agar bisa login dan menggunakan sistem

.

D. Fitur CRUD

```
os.system("cls || clear")
        print(TITLE + f"=== Login sebagai {username} ({role}) ==="
+ RESET)
        print(INFO + "1. Lihat Stok Alat" + RESET)
        if role == "Admin":
            print("2. Tambah Alat")
            print("3. Update Status Alat")
            print("4. Hapus Alat")
            print("5. Lihat Daftar User")
        print("6. Hitung Total Jumlah Alat (rekursif)")
        print("7. Logout")
```

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.

E. Fitur Read

```
if pilih == "1":
                tampilkan_stok_alat()
                tanya = input("\nIngin cek jumlah alat tertentu? (y/n):
").lower().strip()
                if tanya == "y":
                    nama_cek = input("Masukkan nama alat: ").strip()
                    jumlah = cek_jumlah_alat(nama_cek)
                    if jumlah is not None:
                        print(SUCCESS + f"Jumlah alat '{nama_cek}'
adalah: {jumlah}" + RESET)
                input("Tekan Enter...")
```

Fitur Read berfungsi untuk menampilkan daftar stok alat laboratorium kepada pengguna. Saat pengguna memilih opsi 1 di menu utama setelah login, program membersihkan layar dan menampilkan judul Stok Alat. Jika daftar labGear masih kosong, program akan menampilkan pesan Belum Ada Alat. Namun, jika terdapat data alat, program menampilkan setiap alat secara berurutan dengan informasi lengkap, meliputi nama alat, jumlah, kondisi (misalnya tersedia, dipinjam, atau rusak), dan nama peminjam (jika ada). Setelah menampilkan data, program menunggu pengguna menekan Enter sebelum kembali ke

menu utama. Fitur ini memungkinkan pengguna, baik admin maupun user biasa, untuk memantau ketersediaan dan status alat secara transparan dan real-time.

F. Fitur Create

```
elif pilih == "2" and role == "Admin":
                os.system("cls || clear")
                print(TITLE + "=== TAMBAH ALAT ===" + RESET)
                nama = input("Nama Alat : ").strip()
                if nama == "":
                    print(ERROR + "Nama alat tidak boleh kosong!" +
RESET)
                elif nama in labGear:
                    print(WARNING + "Alat sudah ada!" + RESET)
                else:
                    tambah_alat(nama)
                input("Tekan Enter...")
```

Fitur create berfungsi untuk menambahkan data alat baru ke dalam sistem, dan hanya bisa diakses oleh pengguna dengan peran admin. Saat admin memilih menu 2, program akan membersihkan layar menggunakan perintah os.system("cls || clear") lalu menampilkan judul "=== TAMBAH ALAT ===". Setelah itu admin diminta untuk memasukkan nama alat yang ingin ditambahkan. Program akan memeriksa apakah input nama alat kosong, jika kosong maka muncul pesan "Nama alat tidak boleh kosong!". Jika nama alat sudah ada di dalam dictionary labGear, maka akan muncul pesan "Alat sudah ada!". Namun jika nama alat valid dan belum terdaftar, program meminta input jumlah alat. Input jumlah akan dicek agar berupa angka dan nilainya lebih dari 0. Jika valid, data alat baru disimpan ke dalam dictionary labGear dengan format {"jumlah": int(jumlah), "status": "tersedia", "peminjam": "-"}. Setelah itu program menampilkan pesan "Alat berhasil ditambahkan!" dan menunggu admin menekan Enter untuk melanjutkan. Fitur ini digunakan agar admin dapat menambah data alat laboratorium yang bisa dipinjam oleh pengguna lain.

G. Fitur Update

```
elif pilih == "3" and role == "Admin":
            os.system("cls || clear")
            print("=== UPDATE STATUS ===")
            if not labGear:
                print("Tidak ada alat yang bisa diupdate.")
            else:
                for i, alat in enumerate(labGear.keys(), start=1):
                    print(f"{i}. {alat} -
{labGear[alat]['status']}")
                no = input("Pilih nomor alat: ").strip()
                if no.isdigit():
                    idx = int(no) - 1
                    if 0 <= idx < len(labGear):
                        namaAlat = list(labGear.keys())[idx]
                        print("1. Dipinjam\n2. Rusak\n3.
Tersedia")
                        aksi = input("Pilih status : ").strip()
                        if aksi == "1":
                            peminjam = input("Nama Peminjam :
").strip()
                            if peminjam:
                                labGear[namaAlat]["status"] =
"dipinjam"
                                labGear[namaAlat]["peminjam"] =
peminjam
                                print("Status alat diperbarui
menjadi Dipinjam.")
                            else:
                                print("Nama peminjam tidak boleh
kosong.")
                        elif aksi == "2":
                            labGear[namaAlat]["status"] = "rusak"
                            labGear[namaAlat]["peminjam"] = "-"
                            print("Status alat diperbarui menjadi
Rusak.")
                        elif aksi == "3":
                            labGear[namaAlat]["status"] =
"tersedia"
                            labGear[namaAlat]["peminjam"] = "-"
                            print("Status alat diperbarui menjadi
Tersedia.")
                        else:
```

```
                        print("Pilihan tidak valid.")
                else:
                    print("Nomor alat tidak valid!")
            else:
                print("Masukkan angka!")
        input("Tekan Enter...")
```

Fitur update digunakan oleh admin untuk mengubah status alat yang sudah terdaftar di sistem. Saat admin memilih menu 3, program akan membersihkan layar menggunakan perintah os.system("cls || clear") lalu menampilkan judul "=== UPDATE STATUS ===". Program terlebih dahulu memeriksa apakah dictionary labGear kosong, jika kosong maka akan muncul pesan "Tidak ada alat yang bisa diupdate.". Jika terdapat data alat, program akan menampilkan daftar alat beserta statusnya dalam bentuk nomor urut. Admin kemudian diminta memasukkan nomor alat yang ingin diubah. Input nomor akan dicek apakah berupa angka dan valid berdasarkan jumlah data yang ada. Jika valid, program menampilkan pilihan status baru yaitu: 1. Dipinjam, 2. Rusak, dan 3. Tersedia. Jika admin memilih opsi 1, maka sistem akan meminta nama peminjam. Jika nama peminjam diisi, status alat akan diubah menjadi "dipinjam" dan nama peminjam akan disimpan di data alat tersebut. Jika peminjam tidak diisi, muncul pesan "Nama peminjam tidak boleh kosong.". Jika admin memilih opsi 2, status alat akan diubah menjadi "rusak" dan nama peminjam direset menjadi "-". Jika admin memilih opsi 3, status alat akan diubah menjadi "tersedia" dan peminjam juga direset menjadi "-". Namun jika pilihan status tidak sesuai, program akan menampilkan pesan "Pilihan tidak valid.". Setelah proses selesai, program akan menunggu admin menekan Enter untuk melanjutkan. Fitur ini berfungsi agar admin dapat memperbarui kondisi atau ketersediaan alat sesuai keadaan sebenarnya di laboratorium.

H. Fitur Delete

```
    elif pilih == "4" and role == "Admin":
        os.system("cls || clear")
        print("=== HAPUS ALAT ===")
        if not labGear:
            print("Tidak ada alat yang bisa dihapus.")
        else:
            for i, alat in enumerate(labGear.keys(), start=1):
                print(f"{i}. {alat}")
            no = input("Pilih nomor alat: ").strip()
            if no.isdigit():
                idx = int(no) - 1
```

```
                    if 0 <= idx < len(labGear):
                        namaAlat = list(labGear.keys())[idx]
                        konfirm = input(f"Hapus {namaAlat}? (y/n):
").lower().strip()
                        if konfirm == "y":
                            del labGear[namaAlat]
                            print("Alat berhasil dihapus.")
                        else:
                            print("Dibatalkan.")
                    else:
                        print("Nomor alat tidak valid.")
                else:
                    print("Masukkan angka!")
            input("Tekan Enter...")
```

Fitur delete digunakan oleh admin untuk menghapus data alat yang sudah tidak diperlukan dari sistem. Saat admin memilih menu 4, program akan membersihkan layar menggunakan perintah os.system("cls || clear") lalu menampilkan judul "=== HAPUS ALAT ===". Program kemudian memeriksa apakah dictionary labGear kosong, jika kosong maka muncul pesan "Tidak ada alat yang bisa dihapus.". Jika ada data alat, program akan menampilkan daftar alat dalam bentuk nomor urut. Admin diminta memasukkan nomor alat yang ingin dihapus, dan input tersebut diperiksa apakah berupa angka yang valid. Jika nomor yang dimasukkan sesuai dengan daftar alat, program akan menampilkan konfirmasi penghapusan dengan pertanyaan "Hapus (nama alat)? (y/n)". Jika admin mengetik "y", maka alat tersebut akan dihapus dari dictionary labGear menggunakan perintah del labGear[namaAlat], dan program menampilkan pesan "Alat berhasil dihapus.". Jika admin mengetik selain "y", maka proses dibatalkan dan muncul pesan "Dibatalkan.". Namun jika nomor alat tidak valid atau bukan angka, program akan menampilkan pesan kesalahan yang sesuai. Setelah proses selesai, program menunggu admin menekan Enter untuk melanjutkan. Fitur ini berfungsi agar admin bisa menghapus data alat yang sudah rusak permanen, hilang, atau tidak lagi digunakan di laboratorium.

I. Fitur Warna

```
from colorama import Fore, Style, init
init(autoreset=True)
```

```
TITLE = Fore.CYAN + Style.BRIGHT
SUCCESS = Fore.GREEN + Style.BRIGHT
ERROR = Fore.RED + Style.BRIGHT
WARNING = Fore.YELLOW + Style.BRIGHT
INFO = Fore.MAGENTA + Style.BRIGHT
RESET = Style.RESET_ALL
```

Bagian kode ini berfungsi untuk mengatur warna teks pada tampilan terminal menggunakan modul Colorama agar output program terlihat lebih menarik dan mudah dibaca. Modul colorama diinisialisasi dengan init(autoreset=True) agar setiap kali mencetak teks berwarna, warna tersebut otomatis kembali ke warna default setelah baris tersebut selesai ditampilkan. Beberapa variabel kemudian didefinisikan untuk menyimpan kombinasi warna dan gaya teks yang umum digunakan dalam program, yaitu TITLE menggunakan warna cyan terang untuk judul atau header, SUCCESS dengan warna hijau terang untuk pesan keberhasilan, ERROR dengan warna merah terang untuk menampilkan kesalahan, WARNING dengan warna kuning terang untuk peringatan, INFO dengan warna magenta terang untuk pesan informasi, dan RESET untuk mengembalikan tampilan teks ke kondisi standar tanpa warna. Dengan penggunaan warna-warna ini, tampilan program menjadi lebih konsisten, jelas, dan mudah dibaca oleh pengguna.

4. Hasil Output

```
=== SISTEM LAB GEAR UNIVERSITAS ===
Selamat datang di sistem manajemen alat laboratorium

1. Login
2. Register
3. Keluar
Pilihan :
```

```
=== LOGIN ===
Username :
```

```
=== Login sebagai admin (Admin) ===
1. Lihat Stok Alat
2. Tambah Alat
3. Update Status Alat
4. Hapus Alat
5. Lihat Daftar User
6. Hitung Total Jumlah Alat (rekursif)
7. Logout
Pilih Menu :
```

```
=== REGISTER ===
Username Baru :
```

5. Langkah-langkah GIT

5.1 GIT Add

```
PS D:\kuliah\semester1\praktikum\algoritma\praktikum-apd> git add .
```

Simpan Perubahan

5.2 GIT Commit

```
PS D:\kuliah\semester1\praktikum\algoritma\praktikum-apd> git commit -m "PT8"
[main 14bbe61] PT8
 2 files changed, 1 deletion(-)
```

tambahkan perubahan pada repositori

5.3 GIT Push

```
PS D:\kuliah\semester1\praktikum\algoritma\praktikum-apd> git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 16 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 687 bytes | 687.00 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 github.com:ExtraYiban/praktikum-apd.git
   8db5864..14bbe61  main -> main
```

Kirim Perubahan Ke Cloud (Github)