

LAPORAN PRAKTIKUM
POSTTEST (6)
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:

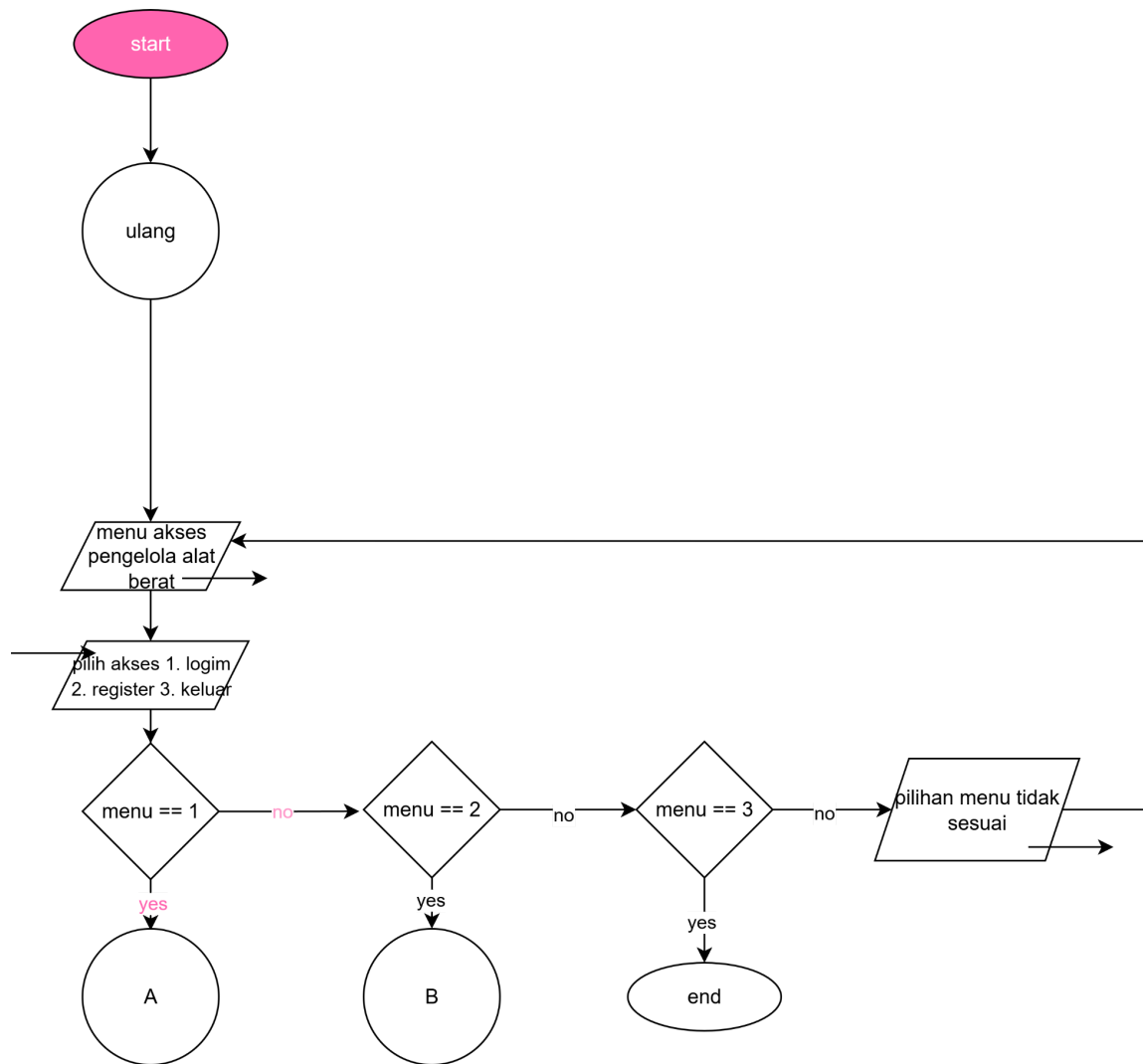
Alia Fransiska Dewi Arum Trilestari (2509106015)

Kelas (A1 '25)

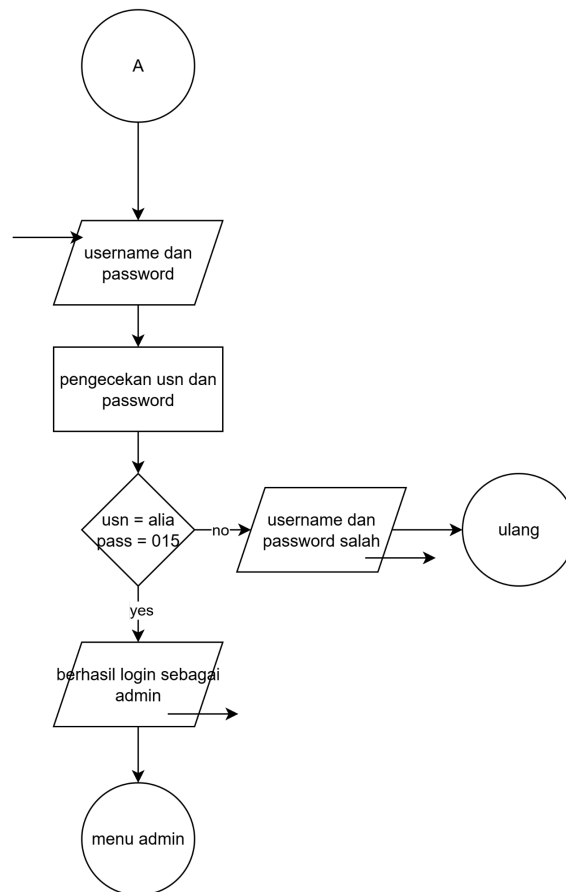
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

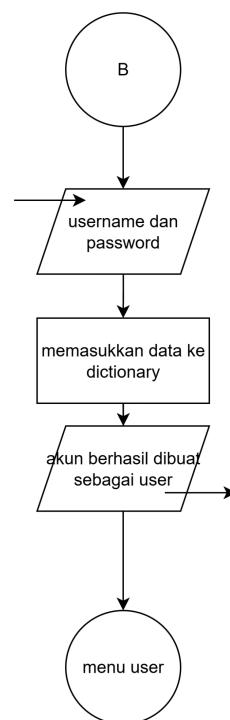
1. Flowchart



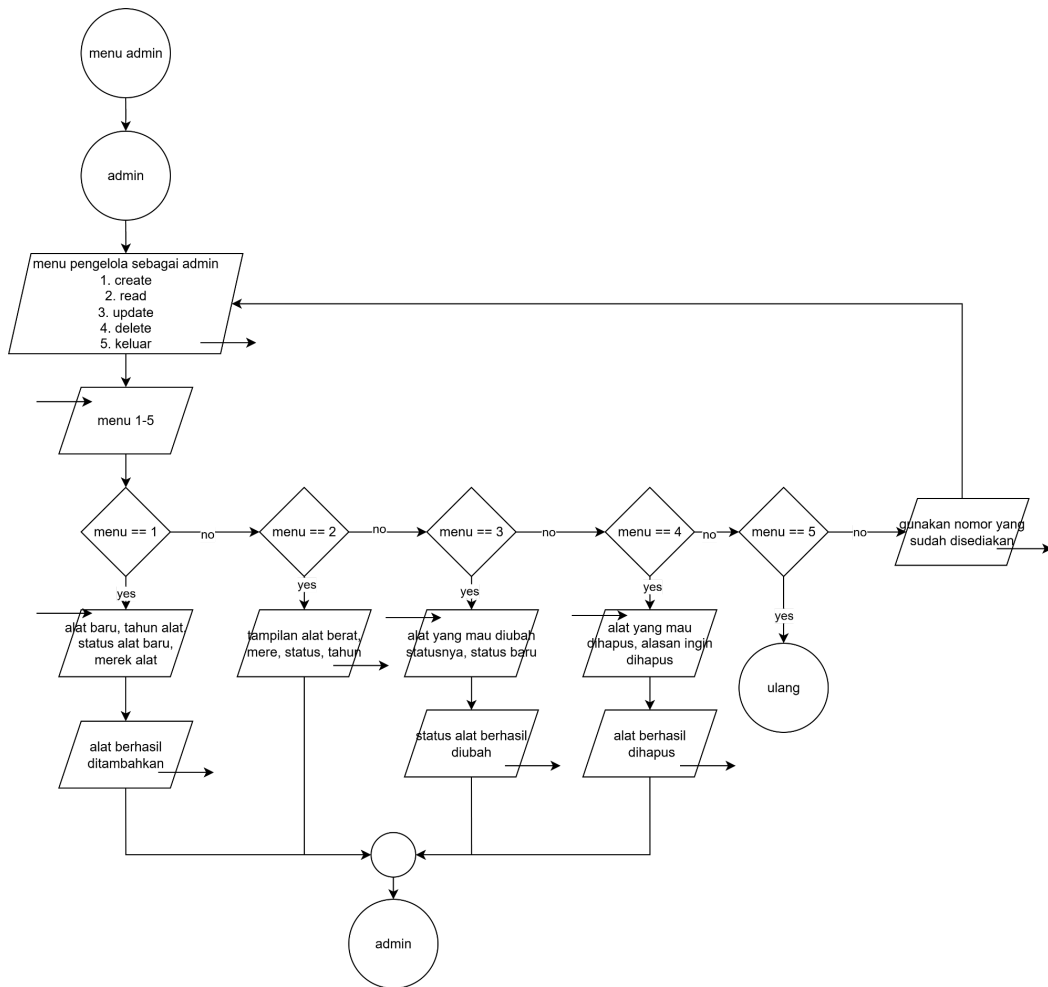
Gambar Flowchart 1.1



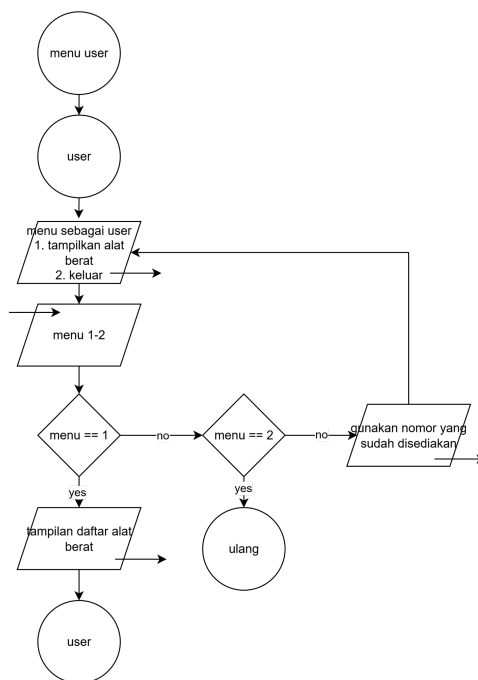
Gambar Flowchart 1.2



Gambar Flowchart 1.3



Gambar Flowchart 1.4



Gambar Flowchart 1.5

Flowchart ini untuk menggambarkan program pengelola alat berat di proyek konstruksi, program membuat akses untuk pengelola alat berat yaitu akses sebagai admin dan akses sebagai pengguna biasa. Pertama program akan menampilkan menu untuk akses yaitu ada admin, register (membuat akun baru), dan keluar. Jika pengguna memilih login untuk admin, maka pengguna harus menginput username dan password sesuai dengan yang ditentukan. Jika berhasil maka program akan langsung mengarahkan admin ke menu pengelola alat berat. Bedanya akses admin dan pengguna biasa adalah, kalau akses admin pengguna dapat melakukan menu 1-5 pada menu pengelola alat berat yaitu 1). Create, 2). Read, 3). Update, 4). Delete, dan 5). Keluar. Dan akses pengguna biasa hanya ada 2 menu yaitu menampilkan alat berat dan keluar dari program. Akses pengguna biasa juga harus menginput username baru dan password baru, setelah itu program akan langsung menampilkan menu.

2. Deskripsi singkat program

Program ini dibuat untuk sistem pengelola alat berat, program memiliki dua akses untuk pengelola alat berat yaitu admin dan pengguna biasa. Bedanya akses admin dan user yaitu admin dapat melakukan CRUD pada alat berat ada penambahan alat berat, menambahkan alat berat baru, mengubah status alat berat, dan menghapus alat berat yang sudah tidak dipakai atau sedang diperbaiki. Admin login dengan menggunakan username dan password yang sudah ditentukan. Jika login admin berhasil maka admin bisa melakukan proses CRUD. Lalu pengguna biasa login dengan menggunakan username dan password baru yang sudah dibuat dan pengguna biasa hanya dapat melihat daftar alat berat yang ada tidak bisa melakukan proses CRUD.

3. Source Code

```
pengguna = {  
    "alia": {"password": "015", "akses": "admin"}  
}  
  
alat_berat = {  
    "Excavator": {"merek": "Hitachi", "tahun": 2010, "status": "Siap Pakai"},  
    "Bulldozer": {"merek": "Caterpillar", "tahun": 2020, "status": "Rusak"},  
    "Compactor": {"merek": "Bomag", "tahun": 2005, "status": "Dalam Perbaikan"}  
}
```

Ada variabel alat_berat, tahun, merek, dan status_alat, ini berisi daftar alat berat yang ada, tahun alat, merek alat, dan status alat berat. Ada variabel pengguna yang berupa nested dictionary isi dari dictionary itu digunakan untuk login akses pengguna admin. Variabel akses ini untuk menampung akses pengguna yang login, untuk mengetahui apakah pengguna adalah

admin atau pengguna biasa.

```
while True:
    os.system("cls" if os.name == "nt" else "clear")
    print("=== SISTEM PENGELOLA ALAT BERAT ===")
    print("1. Login")
    print("2. Register")
    print("3. Keluar")

    menu = input("Pilih menu (1-3): ")
```

Program akan menampilkan menu untuk pengguna pilih ada 1). Login, 2). Register, 3). Keluar. Setelah itu, pengguna harus memilih menu antara dari 1-3 untuk lanjut ke proses berikutnya.

```
if menu == "1":
    print("=== LOGIN ===")
    username = input("Masukkan Username: ").strip()
    password = input("Masukkan Password: ").strip()

    if username in pengguna and pengguna[username]["password"] == password:
        akses = pengguna[username]["akses"]
        print(f"Login berhasil sebagai {akses}")
        input("Tekan Enter untuk melanjutkan...")
    else:
        print("Username atau password salah!")
        input("Tekan Enter untuk melanjutkan...")
        continue
```

Jika pengguna memilih menu 1 yaitu login, program akan mengarahkan untuk login admin dengan menggunakan username dan password yang sudah ditentukan. Program akan mengecek apakah username dan password sudah sesuai atau belum, jika username dan password sesuai maka program akan menampilkan login berhasil sebagai admin, dan program akan melanjutkan proses berikutnya. Jika username dan password yang diinput salah maka program akan menampilkan “Username dan password salah”. Kemudian pengguna harus menekan Enter. Setelah itu, “continue” akan membuat program melewati sisa kode dalam loop, sehingga pengguna dapat login lagi.

```
# MENU ADMIN
while akses == "admin":
    os.system("cls" if os.name == "nt" else "clear")
    print("=== MENU ADMIN ===")
    print("1. Tambah alat baru")
    print("2. Lihat semua alat")
    print("3. Ubah status alat")
    print("4. Hapus alat")
    print("5. Keluar")
```

```
pilihan = input("Pilih menu (1-5): ")
```

Jika berhasil login admin program akan menampilkan daftar menu pengelolaan alat berat, ada menambah data alat baru (Create), menampilkan daftar alat yang ada (Read), mengubah status alat (Update), menghapus alat yang sudah rusak atau tidak digunakan (Delete), dan ada pilihan untuk keluar dari menu. Setelah itu, pengguna harus memilih salah satu menu.

```
# CREATE
if pilihan == "1":
    os.system("cls" if os.name == "nt" else "clear")
    print("=== TAMBAH ALAT BARU ===")
    alat = input("Masukkan nama alat: ").strip()

    if alat in alat_berat:
        print("Alat sudah tersedia!")
    else:
        merek = input("Masukkan merek alat: ").strip()
        tahun_input = int(input("Masukkan tahun alat: "))
        status = input("Masukkan status alat (Siap Pakai/Rusak/Dalam Perbaikan): ").strip()

        alat_berat.update({
            alat: {"merek": merek, "tahun": tahun_input, "status": status}
        })

        print(f"Alat '{alat}' berhasil ditambahkan.")
        input("Tekan Enter untuk melanjutkan...")
```

Saat admin memilih menu 1 maka program akan mengarahkan ke tambah alat baru, lalu program meminta admin untuk menginput nama, merek, tahun, dan status dari alat baru tersebut. Setelah menginput semua data program akan menampilkan pesan jika alat baru sudah ditambahkan. Jika nama alat yang dimasukkan sudah ada di data alat_berat maka program akan menampilkan pesan alat sudah tersedia.

```
# READ
elif pilihan == "2":
    os.system("cls" if os.name == "nt" else "clear")
    print("=== DAFTAR ALAT BERAT ===")
    table = PrettyTable()
    table.field_names = ["Nama", "Merek", "Tahun", "Status"]
    for nama, value in alat_berat.items():
        table.add_row([nama, value["merek"], value["tahun"], value["status"]])
    print(table)
    input("\nTekan Enter untuk melanjutkan...")
```

Jika admin memilih menu 2 maka program akan menampilkan tabel yang berisi nama, merek, tahun, dan status alat berat dengan menggunakan PrettyTable dan admin dapat melihat data dari data berat.

```
# UPDATE

elif pilihan == "3":
    os.system("cls" if os.name == "nt" else "clear")
    print("=== UBAH STATUS ALAT ===")
    alat = input("Masukkan nama alat yang ingin diubah: ").strip()
    if alat in alat_berat:
        status_baru = input("Masukkan status baru: ").strip()
        alat_berat[alat].update(["status"] = status_baru)
        print(f"Status alat '{nama}' berhasil diubah menjadi '{status_baru}'")
    else:
        print("Tidak ada alat tersebut")
        input("Tekan Enter untuk melanjutkan...")
```

Ketika pengguna memilih menu nomor 3, meminta pengguna untuk memasukkan nama alat yang ingin diubah statusnya. Program kemudian akan memeriksa apakah nama alat tersebut ada di dalam dictionary alat_berat. Jika ada maka pengguna diminta untuk memasukkan status baru, lalu nilai pada "status" di dalam dictionary alat tersebut akan diperbarui dengan status baru yang dimasukkan. Setelah itu, program menampilkan status alat berhasil diubah. Jika nama alat yang dimasukkan tidak ada di dalam data, program akan menampilkan "Tidak ada alat tersebut".

```
# DELETE

elif pilihan == "4":
    os.system("cls" if os.name == "nt" else "clear")
    print("=== HAPUS ALAT ===")
    alat = input("Masukkan nama alat yang ingin dihapus: ").strip()
    if alat in alat_berat:
        alat_berat.pop(alat)
        status_dihapus = input("Kenapa ingin menghapus alat ini (Rusak/Ingin Diperbaiki): ")
        print(f"Alat '{alat}' berhasil dihapus, karena '{status_dihapus}'.")
    else:
        print("Tidak ada alat tersebut")
        input("Tekan Enter untuk melanjutkan...")
```

Ketika pengguna memilih menu nomor 4, lalu pengguna diminta untuk memasukkan nama alat yang ingin dihapus. Program kemudian memeriksa apakah nama alat tersebut ada di dalam dictionary alat_berat. Jika ada, data alat tersebut akan dihapus dengan perintah alat_berat.pop(alat). Program lalu menanyakan alasan penghapusan alat (karena rusak atau ingin diperbaiki), dan menampilkan pesan alat berhasil dihapus dengan alasannya. Jika nama

alat tidak ditemukan di dalam dictionary, maka akan muncul pesan “Tidak ada alat tersebut”.

```
elif pilihan == "5":  
    break  
else:  
    print("Gunakan Urutan Nomor Pada Menu Yang Sudah  
Disediakan")
```

Jika admin memilih menu 5 yaitu “Keluar” maka program akan langsung berhenti. Jika admin menginput nomor selain 1-5 maka program akan menampilkan output “Gunakan Urutan Nomor Pada Menu Yang Sudah Disediakan”, maka program akan menanyakan kembali mau memilih menu apa hanya 1-5 menu saja.

```
# REGISTER  
elif menu == "2":  
    os.system("cls" if os.name == "nt" else "clear")  
    print("=== REGISTER AKUN BARU ===")  
    username = input("Masukkan username baru: ").strip()  
    password = input("Masukkan password baru: ").strip()  
  
    if username in pengguna:  
        print("Username sudah terdaftar!")  
    else:  
        pengguna[username] = {"password": password, "akses": "user"}  
        print(f"Akun '{username}' berhasil dibuat sebagai user!")  
  
    input("Tekan Enter untuk melanjutkan...")
```

Saat pengguna memilih menu 2, program akan mengarahkan pengguna untuk melakukan register untuk pengguna biasa. Program meminta pengguna untuk memasukkan username dan password baru. Setelah itu, program memeriksa apakah username sudah ada di dalam dictionary pengguna. Jika nama pengguna sudah ada, program akan menampilkan pesan username sudah terdaftar. Jika belum ada berarti akun baru yang dibuat secara otomatis memiliki akses sebagai user (bukan admin). Terakhir, program menampilkan pesan akun berhasil dibuat, lalu pengguna menekan Enter sebelum lanjut ke proses berikutnya. Jika akses menggunakan user program menampilkan menu untuk user ada dua pilihan: 1). Tampilkan alat berat untuk melihat daftar alat yang tersedia, dan 2). Keluar untuk keluar dari menu user. Program kemudian meminta input pilihan menu dari pengguna input("Masukkan Nomor Pada Menu (1-2): ").

```
if pilihan == "1":  
    os.system("cls" if os.name == "nt" else "clear")  
    print("=== DAFTAR ALAT BERAT ===")  
    table = PrettyTable()  
    table.field_names = ["Nama", "Merek", "Tahun", "Status"]
```

```

        for nama, data in alat_berat.items():
            table.add_row([nama, data["merek"], data["tahun"],
data["status"]])
        print(table)
        input("Tekan Enter untuk melanjutkan...")
    elif pilihan == "2":
        break
    else:
        print("Gunakan urutan nomor yang sudah disediakan.")
        input("Tekan Enter untuk melanjutkan...")

```

Jika pengguna memilih pilihan 1, program akan menampilkan tabel yang berisi Nama Alat, Tahun, Merek, dan Status. Jika pengguna memilih opsi 2, program akan menghentikan perulangan menggunakan perintah break, yang berarti keluar dari menu. Jika pengguna memasukkan pilihan selain “1” atau “2”, program akan menampilkan pesan agar pengguna menggunakan nomor yang sesuai dengan menu yang telah disediakan.

```

elif menu == "3":
    os.system("cls" if os.name == "nt" else "clear")
    print("Program berhenti")
    break
else:
    print("Pilihan menu tidak sesuai")

```

Jika pengguna memilih pilihan 3, program menampilkan pesan “Program berhenti” sebagai tanda bahwa proses akan berhenti. Jika pengguna memasukkan pilihan selain 1-3 bagian else akan dijalankan, dan menampilkan pesan “Pilihan tidak valid.”

4. Hasil output

```

=== LOGIN ===
Masukkan Username: alia
Masukkan Password: 015
Login berhasil sebagai admin
Tekan Enter untuk melanjutkan...

```

Gambar 4.1 Login Menggunakan Username dan Password Yang Sesuai dan Berhasil Login Sebagai Admin

```

=== MENU ADMIN ===
1. Tambah alat baru
2. Lihat semua alat
3. Ubah status alat
4. Hapus alat
5. Keluar
Pilih menu (1-5): █

```

Gambar 4.2 Output Menu Untuk Pengelola Alat Berat

```

=== TAMBAH ALAT BARU ===
Masukkan nama alat: Crane
Masukkan merek alat: Terex
Masukkan tahun alat: 2024
Masukkan status alat (Siap Pakai/Rusak/Dalam Perbaikan): Siap Pakai
Alat 'Crane' berhasil ditambahkan.
Tekan Enter untuk melanjutkan...█

```

Gambar 4.3 Menu 1 Menambahkan Alat Berat Baru

Nama	Merek	Tahun	Status
Excavator	Hitachi	2010	Siap Pakai
Bulldozer	Caterpillar	2020	Rusak
Compactor	Bomag	2005	Dalam Perbaikan
Crane	Terex	2024	Siap Pakai

Gambar 4.4 Memilih Menu 2 Tampilan Alat Berat dan Ada Tambahan Alat Baru

```

=== UBAH STATUS ALAT ===
Masukkan nama alat yang ingin diubah: Bulldozer
Masukkan status baru (Siap Pakai/Rusak/Dalam Perbaikan): Siap Pakai
Status alat 'Bulldozer' berhasil diubah menjadi 'Siap Pakai'
Tekan Enter untuk melanjutkan...█

```

Gambar 4.5 Menu 3 Mengubah Status Alat Berat Baru

```
=== DAFTAR ALAT BERAT ===
+-----+-----+-----+-----+
| Nama | Merek | Tahun | Status |
+-----+-----+-----+-----+
| Excavator | Hitachi | 2010 | Siap Pakai |
| Bulldozer | Caterpillar | 2020 | Siap Pakai |
| Compactor | Bomag | 2005 | Dalam Perbaikan |
| Crane | Terex | 2024 | Siap Pakai |
+-----+-----+-----+-----+
Tekan Enter untuk melanjutkan...
```

Gambar 4.6 Output Setelah Mengubah Status Alat Berat

```
=== HAPUS ALAT ===
Masukkan nama alat yang ingin dihapus: Compactor
Kenapa ingin menghapus alat ini (Rusak/Ingin Diperbaiki): Ingin Diperbaiki
Alat 'Compactor' berhasil dihapus, karena 'Ingin Diperbaiki'.
Tekan Enter untuk melanjutkan...
```

Gambar 4.7 Memilih Menu 4 Untuk Menghapus Alat

```
=== DAFTAR ALAT BERAT ===
+-----+-----+-----+-----+
| Nama | Merek | Tahun | Status |
+-----+-----+-----+-----+
| Excavator | Hitachi | 2010 | Siap Pakai |
| Bulldozer | Caterpillar | 2020 | Siap Pakai |
| Crane | Terex | 2024 | Siap Pakai |
+-----+-----+-----+-----+
Tekan Enter untuk melanjutkan...
```

Gambar 4.8 Output Setelah Menghapus Alat Berat

```
=== REGISTER AKUN BARU ===
Masukkan username baru: arum
Masukkan password baru: 1
Akun 'arum' berhasil dibuat sebagai user!
Tekan Enter untuk melanjutkan...
```

Gambar 4.9 Membuat Akun Baru dan Menampilkan Menu Untuk User

```
=== MENU USER ===
1. Tampilkan alat berat
2. Keluar
Masukkan Nomor Pada Menu (1-2): █
```

Gambar 4.10 Menu Untuk User

5. Instalasi Git

5.1 GIT Add

```
PS C:\praktikum-apd> git add .
```

Gambar 5.1 Perintah Git Add

Git Add digunakan untuk menambahkan file baru atau perubahan baru pada program yang dibuat.

5.2 GIT Commit

```
PS C:\praktikum-apd> git commit -m 'posttestapd'
[main 4fc40bb] posttestapd
1 file changed, 1 deletion(-)
```

Gambar 5.2 Perintah Git Commit

Git Commit digunakan untuk menyimpan perubahan pada program yang dibuat.

5.3 GIT Push

```
PS C:\praktikum-apd> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 447 bytes | 447.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/AliaFransiskaDewiArumTrilestari-A25/praktikum-apd.git
   fc46030..4fc40bb  main -> main
PS C:\praktikum-apd> █
```

Gambar 5.3 Perintah Git Push

Git Push digunakan untuk mengupload perubahan dan mengirimkannya ke repositori github.