

LAPORAN PRAKTIKUM
POSTTEST (5)
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:

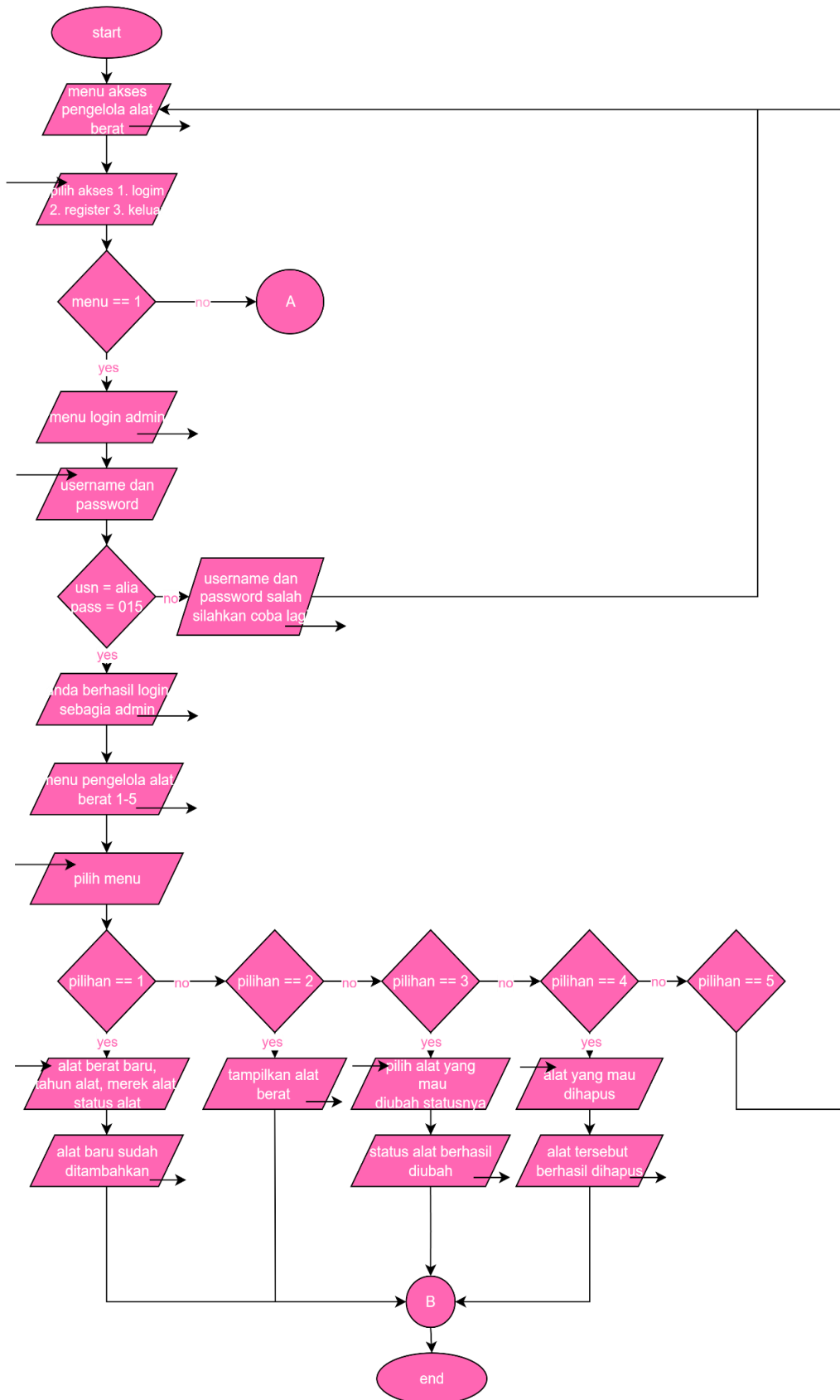
Alia Fransiska Dewi Arum Trilestari (2509106015)

Kelas (A1 '25)

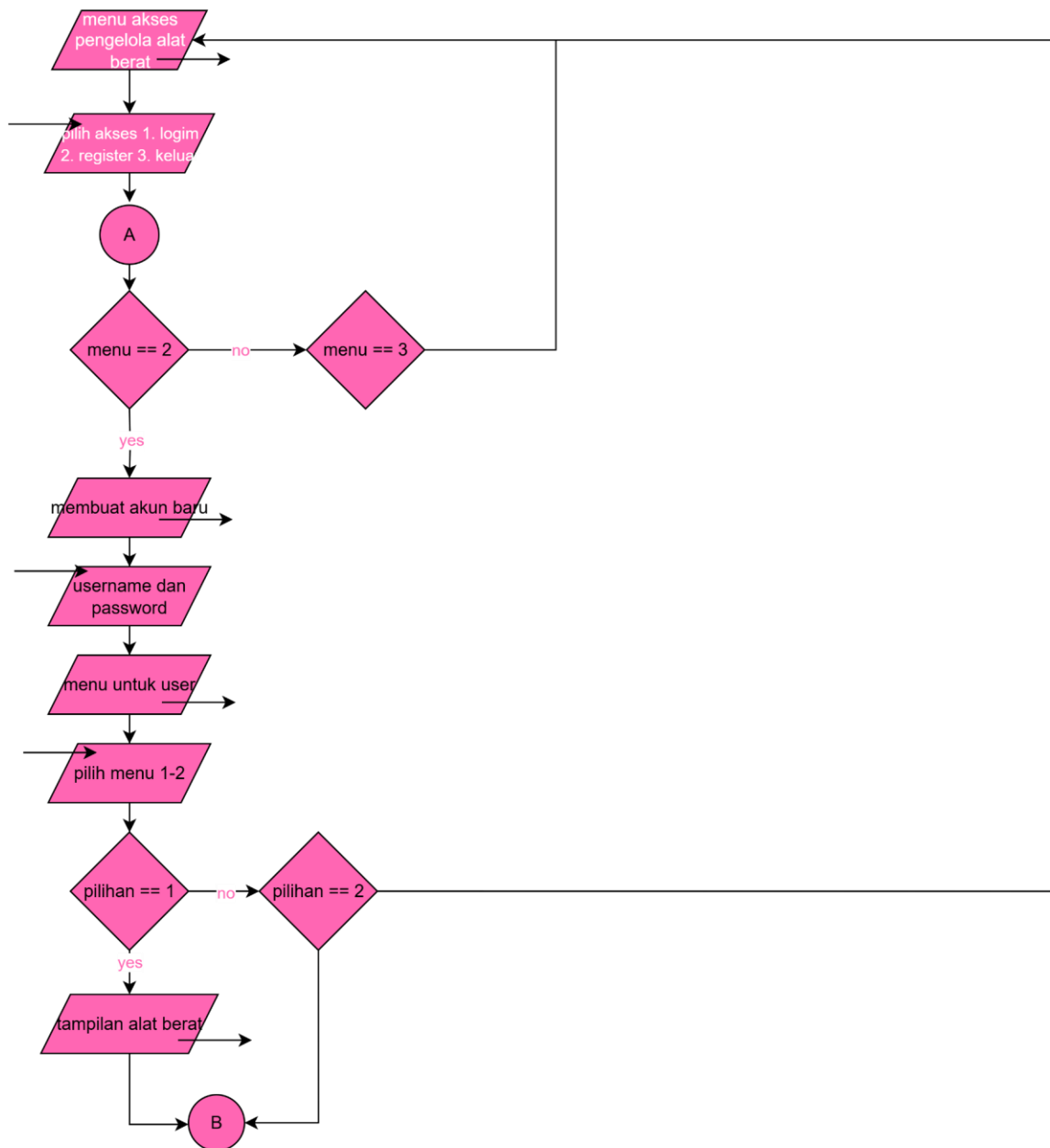
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

1. Flowchart



Gambar Flowchart 1.1



Gambar Flowchart 1.2

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", "015", "admin"]
]

alat_berat = ["Excavator", "Bulldozer", "Compactor"]
tahun = [2010, 2020, 2005]
merek = ["Hitachi", "Caterpillar", "Bomag"]
status_alat = ["Siap Pakai", "Rusak", "Dalam Perbaikan"]

akses = ""
```

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 list isi dari list 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("=== AKSES 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 ADMIN ===")
    username = input("Masukkan Username: ").strip()
    password = input("Masukkan Password: ").strip()

    akses = ""
    for up in pengguna:
        if up[0] == username and up[1] == password:
            akses = up[2]
            break
    if akses == "":
        print("Username atau password salah")
    else:
        print(f"Login berhasil sebagai {akses}")

```

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".

Untuk up[0] itu adalah indeks 0 yang berisi nama yang sudah kita buat di list variabel pengguna, lalu up[1] itu adalah indeks 1 yang berisi password di list pengguna, dan up[2] itu indeks untuk akses login yang sudah dilakukan yaitu admin. Disini saya menggunakan variabel up untuk menemapung data.

```

while akses == "admin":
    print("\n=== MENU PENGELOLA ALAT BERAT ===")
    print("1. Create = Tambah alat berat baru")
    print("2. Read = Tampilkan alat berat")
    print("3. Update = Mengubah status alat berat")
    print("4. Delete = Menghapus alat yang sudah tidak dipakai  
atau rusak")
    print("5. Keluar")

    pilihan = input("Pilih menu untuk pengelola alat berat (1-5): ")

```

Program 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 pilihan.

```

# CREATE

if pilihan == "1":
    print("\n=== TAMBAH ALAT BARU ===")

```

```

        nama_alat_baru = input("Masukkan nama alat: ")
        merek_alat_baru = input("Masukkan merek alat: ")
        tahun_alat_baru = input("Masukkan tahun alat
tersebut: ")

        status_alat_baru = input("Masukkan status alat (Siap
Pakai/Rusak/Dalam Perbaikan): ")
        alat_berat.append(nama_alat_baru)
        merek.append(merek_alat_baru)
        tahun.append(tahun_alat_baru)
        status_alat.append(status_alat_baru)
        print(f'Alat: {nama_alat_baru}, Merek:
{merek_alat_baru}, Tahun: {tahun_alat_baru}, Status: {status_alat_baru} Sudah
Ditambahkan')

```

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.

```

# READ

        elif pilihan == "2":
            table = PrettyTable()
            table.field_names = ["Nama", "Merek", "Tahun",
"Status"]

            for i in range(len(alat_berat)):
                table.add_row([alat_berat[i], merek[i],
tahun[i], status_alat[i]])
            print(table)

```

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":
            print("\n=== UBAH STATUS ALAT ===")
            if alat_berat == []:
                print("Belum ada alat yang bisa diubah.")
            else:
                no = 0
                for alat in alat_berat:
                    print(no + 1, ".", alat, "-",
status_alat[no])

                    no = no + 1

                ubah_status = int(input("Masukkan nomor alat
yang ingin diubah: "))

                status_baru = input("Masukkan status baru: ")

```

```

        indeks = ubah_status - 1
        status_alat[indeks] = status_baru
        print(f"Status alat '{alat_berat[indeks]}'
berhasil diubah")

```

Jika admin memilih menu 3 maka program akan mengarahkan untuk mengubah status dari alat berat. Program akan memeriksa jika data alat berat kosong, jika belum ada data maka program akan menampilkan belum ada alat yang bisa diubah. Jika data tersedia, program menampilkan seluruh alat berat dan statusnya. Admin menginput nomor alat yang ingin diubah, dengan status baru yang akan diberikan. Setelah itu, program menghitung indeks karena dalam daftar list, indeks dimulai dari 0, bukan dari 1 jadi kita perlu menghitung dengan indeks = ubah_status - 1, ini program mengubah nomor (yang dimulai dari 1) menjadi indeks list (yang dimulai dari 0). Lalu program akan mengganti status lama dengan status baru sesuai dengan yang diinput tadi, lalu menampilkan perubahan status berhasil.

```

# DELETE

        elif pilihan == "4":
            print("\n=== HAPUS ALAT BERAT ===")

            if alat_berat == []:
                print("Belum ada alat yang bisa dihapus.")
            else:
                no = 0
                for alat in alat_berat:
                    print(no + 1, ".", alat, "-",
status_alat[no])

                    no = no + 1
                hapus = int(input("Masukkan nomor alat yang
ingin dihapus: "))

                status_dihapus = input("Kenapa ingin menghapus
alat ini (Rusak/Ingin Diperbaiki): ")

                indeks = hapus - 1
                dihapus = alat_berat[indeks]
                del alat_berat[indeks]
                del merek[indeks]
                del tahun[indeks]
                del status_alat[indeks]

                print(f"Alat '{dihapus}' berhasil dihapus!")
                print(f"Karena alat tersebut {status_dihapus}")

```

Jika admin menginput menu 4 maka program akan menampilkan “hapus alat berat”. Program akan memeriksa jika data alat berat kosong, jika belum ada data maka program akan menampilkan belum ada alat yang bisa dihapus. Jika data tersedia, program akan menampilkan

seluruh alat berat beserta statusnya dalam daftar bernomor. Admin harus memasukkan nomor alat yang ingin dihapus, lalu program mengubah nomor itu menjadi indeks agar sesuai dengan data di dalam list. Setelah itu, program menghapus data alat berat yang dipilih. Jika sudah selesai program menampilkan pesan alat tersebut telah berhasil dihapus.

```
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 berenti. 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.

```
elif menu == "2":
    os.system("cls" if os.name == "nt" else "clear")
    print("=== MEMBUAT AKUN BARU ===")
    username = input("Masukkan username baru: ").strip()
    password = input("Masukkan password baru: ").strip()

    akses = ""
    for up in pengguna:
        if up[0] == username and up[1] == password:
            akses = up[2]
            break

    akses = "user"

    while akses == "user":
        print("\n=== MENU UNTUK USER ===")
        print("1. Tampilkan alat berat")
        print("2. Keluar")

        pilihan = input("Masukkan Nomor Pada Menu (1-2): ")
```

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 mengecek pada list pengguna untuk melihat apakah username dan password tersebut sudah ada atau belum, jika belum maka pengguna diharuskan untuk membuat akun dahulu. 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")
    table = PrettyTable()
    table.field_names = ["Nama", "Merek", "Tahun", "Status"]

    for i in range(len(alat_berat)):
        table.add_row([alat_berat[i], merek[i], tahun[i],
status_alat[i]])
    print(table)
elif pilihan == "2":
    break
else:
    print("Gunakan Urutan Nomor Pada Menu Yang Sudah
Disediakan")

```

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

```

=== AKSES PENGELOLA ALAT BERAT ===
1. Login
2. Register
3. Keluar
Pilih menu (1-3): 1
=== LOGIN ADMIN ===
Masukkan Username: alia
Masukkan Password: 015
Login berhasil sebagai admin

```

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

```

=== MENU PENGELOLA ALAT BERAT ===
1. Create = Tambah alat berat baru
2. Read = Tampilkan alat berat
3. Update = Mengubah status alat berat
4. Delete = Menghapus alat yang sudah tidak dipakai atau rusak
5. Keluar
Pilih menu untuk pengelola alat berat (1-5): █

```

Gambar 4.2 Output Menu Untuk Pengelola Alat Berat

```

Pilih menu untuk pengelola alat berat (1-5): 1

=== TAMBAH ALAT BARU ===
Masukkan nama alat: Crane
Masukkan merek alat: Terex
Masukkan tahun alat tersebut: 2024
Masukkan status alat (Siap Pakai/Rusak/Dalam Perbaikan): Siap Pakai
Alat: Crane, Merek: Terex, Tahun: 2024, Status: Siap Pakai Sudah Ditambahkan

```

Gambar 4.3 Menu 1 Menambahkan Alat Berat Baru

```

Pilih menu untuk pengelola alat berat (1-5): 2
+-----+-----+-----+-----+
| 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

```

Pilih menu untuk pengelola alat berat (1-5): 3

=== UBAH STATUS ALAT ===
1 . Excavator - Siap Pakai
2 . Bulldozer - Rusak
3 . Compactor - Dalam Perbaikan
4 . Crane - Siap Pakai
Masukkan nomor alat yang ingin diubah: 2
Masukkan status baru: Siap Pakai
Status alat 'Bulldozer' berhasil diubah

```

Gambar 4.5 Menu 3 Mengubah Status Alat Berat Baru

Pilih menu untuk pengelola alat berat (1-5): 2

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

Gambar 4.6 Output Setelah Mengubah Status Alat Berat

```
Pilih menu untuk pengelola alat berat (1-5): 4

=== HAPUS ALAT BERAT ===
1 . Excavator - Siap Pakai
2 . Bulldozer - Siap Pakai
3 . Compactor - Dalam Perbaikan
4 . Crane - Siap Pakai
Masukkan nomor alat yang ingin dihapus: 3
Kenapa ingin menghapus alat ini (Rusak/Ingin Diperbaiki): ingin diperbaiki
Alat 'Compactor' berhasil dihapus!
Karena alat tersebut ingin diperbaiki
```

Gambar 4.7 Memilih Menu 4 Untuk Menghapus Alat

Pilih menu untuk pengelola alat berat (1-5): 2

Nama	Merek	Tahun	Status
Excavator	Hitachi	2010	Siap Pakai
Bulldozer	Caterpillar	2020	Siap Pakai
Crane	Terex	2024	Siap Pakai

Gambar 4.8 Output Setelah Menghapus Alat Berat

```
Pilih menu (1-3): 2
=== MEMBUAT AKUN BARU ===
Masukkan username baru: arum
Masukkan password baru: 12

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

Gambar 4.9 Membuat Akun Baru dan Menampilkan Menu Untuk User

Masukkan Nomor Pada Menu (1-2): 1

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

Gambar 4.10 Tampilan Alat Berat 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.