

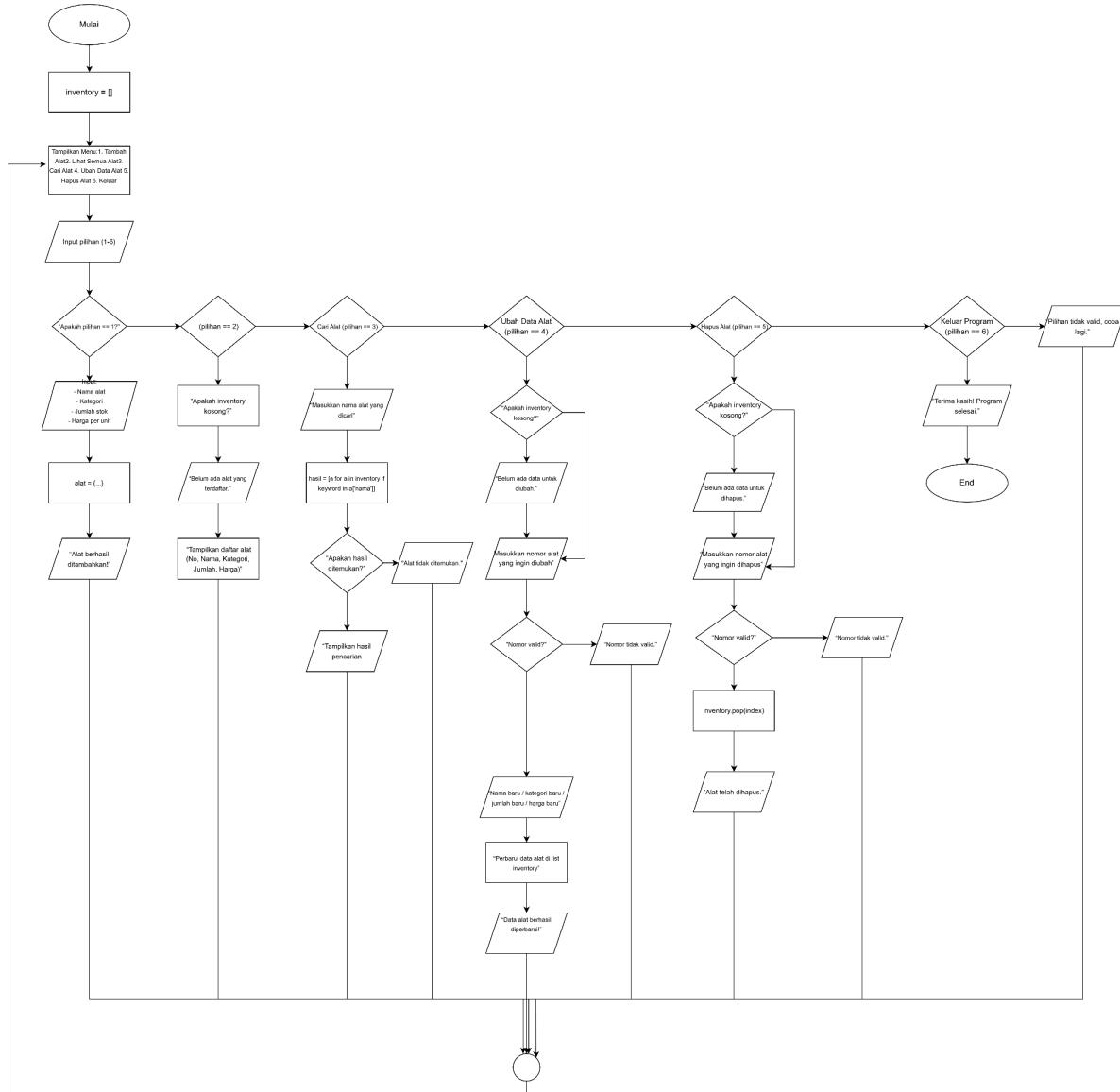
**LAPORAN PRAKTIKUM**  
**POSTTEST 8**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Fathur Rahman (2509106023)**  
**Kelas (A'25)**

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

## 1. Flowchart



**Gambar 1.1 Flowchart**

Flowchart Manajemen Inventory Alat Bangunan adalah diagram alur yang menggambarkan proses pengelolaan data alat bangunan secara sistematis dan terstruktur dalam sebuah program. Flowchart ini menunjukkan urutan langkah-langkah logika program, mulai dari memasukkan data alat, menyimpan informasi ke dalam sistem, menampilkan daftar alat, mencari, mengubah, hingga menghapus data alat yang tidak dibutuhkan lagi.

## 2. Deskripsi Singkat Program

Manajemen Inventory Alat Bangunan adalah sistem yang digunakan untuk mengatur, mencatat, dan memantau persediaan alat-alat bangunan seperti semen, kayu, cat, besi, paku, dan perlengkapan lainnya. Sistem ini membantu agar data alat tersimpan dengan rapi, mudah dicari, serta dapat diperbarui kapan saja jika ada perubahan stok atau harga.

## 3. Source Code

### Main.py



```
1  from auth import login
2  from data import login_user
3  from inventory import tambah_alat, cari_alat, ubah_data_alat, hapus_alat
4  from display import tampilan_semua_alat
5  from utils import clear_screen, input_int, input_float
6  def menu():
7      while True:
8          clear_screen()
9          print("\n== MENU MANAJEMEN INVENTORY ALAT BANGUNAN ==")
10         print("1. Tambah Alat")
11         print("2. Lihat Semua Alat")
12         print("3. Cari Alat")
13         print("4. Ubah Data Alat")
14         print("5. Hapus Alat")
15         print("6. Keluar")
16
17         pilihan = input("\nPilih menu (1-6): ").strip()
18
19         if pilihan == "1":
20             print("\n--- Tambah Alat Baru ---")
21             nama = input("Nama alat: ").strip()
22             kategori = input("Kategori: ").strip()
23             if not nama or not kategori:
24                 print("Nama dan kategori tidak boleh kosong!")
25                 input("\nTekan Enter untuk kembali...")
26                 continue
27             jumlah = input_int("Jumlah stok: ")
28             harga = input_float("Harga per unit: ")
29             tambah_alat(nama, kategori, jumlah, harga)
30
31         elif pilihan == "2":
32             tampilan_semua_alat()
33
34         elif pilihan == "3":
35             keyword = input("Masukkan nama alat yang dicari: ").strip()
36             if keyword:
37                 cari_alat(keyword)
38             else:
39                 print("Kata kunci tidak boleh kosong!")
40
41         elif pilihan == "4":
42             ubah_data_alat()
43
44         elif pilihan == "5":
45             hapus_alat()
46
47         elif pilihan == "6":
48             print(f"\n Terima kasih, {login_user}! Program selesai.")
49             break
50
51         else:
52             print("Pilihan tidak valid!")
53
54         input("\nTekan Enter untuk kembali ke menu...")
55
56 if __name__ == "__main__":
57     if login():
58         menu()
```

Gambar 3.1 SourceCode main.py

## Auth.py



```
1 from data import users, login_user
2
3 def login() -> bool:
4     global login_user
5     print("== SISTEM LOGIN ==")
6     percobaan = 0
7     while percobaan < 3:
8         username = input("Username: ")
9         password = input("Password: ")
10
11        if username in users and users[username] == password:
12            print(f"\nSelamat datang, {username}!")
13            login_user = username
14            return True
15        else:
16            print("Username atau password salah!")
17            percobaan += 1
18
19    print("\nTerlalu banyak percobaan. Program berhenti.")
20    return False
```

Gambar 3.2 SourceCode auth.py

## Data.py



```
1 users = {
2     "admin": "1234",
3     "kasir": "0000",
4     "manager": "abcd"
5 }
6
7 inventory = []
8 login_user = ""
```

Gambar 3.3 SourceCode data .py

## Display.py



```
 1 from prettytable import PrettyTable
 2 from data import inventory
 3
 4 def tampilan_semua_alat():
 5     if not inventory:
 6         print("Belum ada alat yang terdaftar.")
 7         return
 8
 9     table = PrettyTable()
10    table.field_names = ["No", "Nama", "Kategori", "Jumlah", "Harga (Rp)"]
11    for i, (nama, data) in enumerate(inventory.items(), start=1):
12        table.add_row([i, nama, data["kategori"], data["jumlah"], f"{data['harga']:.2f}"])
13
14    print(table)
15
16 def tampilan_hasil_pencarian(hasil: dict):
17    if not hasil:
18        print("Alat tidak ditemukan.")
19        return
20
21    table = PrettyTable()
22    table.field_names = ["Nama", "Kategori", "Jumlah", "Harga (Rp)"]
23    for nama, data in hasil.items():
24        table.add_row([nama, data["kategori"], data["jumlah"], f"{data['harga']:.2f}"])
25
26    print(f"\nDitemukan {len(hasil)} hasil:")
27    print(table)
```

*Gambar 3.4 SourceCode [display.py](#)*

## Inventory.py



```
● ● ●

1 from data import inventory
2 from display import tampilkan_semua_alat, tampilkan_hasil_pencarian
3
4
5 def tambah_alat(nama: str, kategori: str, jumlah: int, harga: float):
6     if nama in inventory:
7         print("Alat ini sudah ada di inventory.")
8     else:
9         inventory[nama] = {"kategori": kategori, "jumlah": jumlah, "harga": harga}
10        print(f"Alat '{nama}' berhasil ditambahkan!")
11
12 def cari_alat(keyword: str):
13     hasil = {n: d for n, d in inventory.items() if keyword.lower() in n.lower()}
14     tampilkan_hasil_pencarian(hasil)
15
16 def ubah_data_alat():
17     if not inventory:
18         print("Belum ada data untuk diubah.")
19         return
20
21     tampilkan_semua_alat()
22     nama_alat = input("\nMasukkan nama alat yang ingin diubah: ")
23
24     if nama_alat not in inventory:
25         print("Alat tidak ditemukan.")
26         return
27
28     data = inventory[nama_alat]
29
30     nama_baru = input("Nama baru (kosongkan jika tidak diubah): ").strip()
31     kategori_baru = input("Kategori baru (kosongkan jika tidak diubah): ").strip()
32     jumlah_baru = input("Jumlah baru (kosongkan jika tidak diubah): ").strip()
33     harga_baru = input("Harga baru (kosongkan jika tidak diubah): ").strip()
34
35     if nama_baru:
36         inventory[nama_baru] = inventory.pop(nama_alat)
37         nama_alat = nama_baru
38         data = inventory[nama_alat]
39
40     if kategori_baru:
41         data["kategori"] = kategori_baru
42     if jumlah_baru:
43         try:
44             data["jumlah"] = int(jumlah_baru)
45         except ValueError:
46             print("Jumlah harus berupa angka!")
47     if harga_baru:
48         try:
49             data["harga"] = float(harga_baru)
50         except ValueError:
51             print("Harga harus berupa angka!")
52
53     print("Data alat berhasil diperbarui!")
54
55 def hapus_alat():
56     if not inventory:
57         print("Belum ada data untuk dihapus.")
58         return
59
60     tampilkan_semua_alat()
61     nama_hapus = input("\nMasukkan nama alat yang ingin dihapus: ").strip()
62
63     if nama_hapus in inventory:
64         del inventory[nama_hapus]
65         print(f"Alat '{nama_hapus}' telah dihapus.")
66     else:
67         print("Nama alat tidak ditemukan.")
```

Gambar 3.5 SourceCode inventory.py

## Utils.py

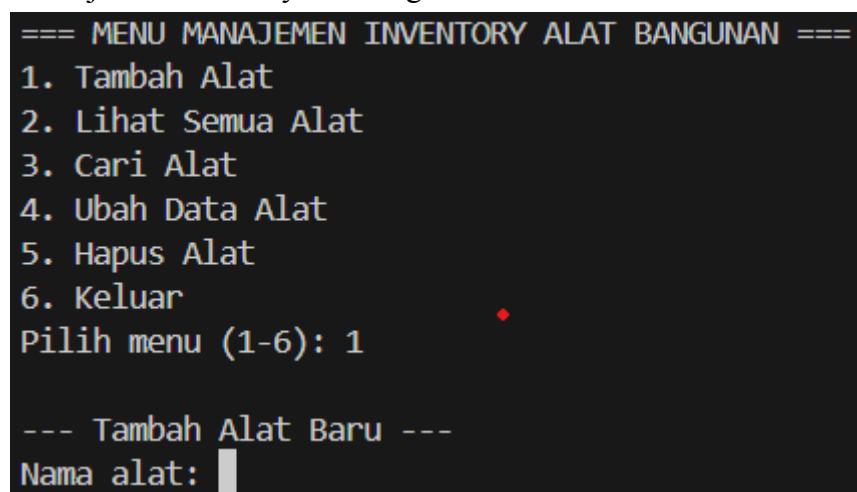


```
1 import os
2
3 def clear_screen():
4     os.system("cls" if os.name == "nt" else "clear")
5
6 def input_int(prompt: str) -> int:
7     while True:
8         try:
9             return int(input(prompt))
10        except ValueError:
11            print("Input harus berupa angka!")
12
13 def input_float(prompt: str) -> float:
14     while True:
15         try:
16             return float(input(prompt))
17        except ValueError:
18            print("Input harus berupa angka desimal/integer!")
```

Gambar 3.6 SourceCode utils.py

## 4. Hasil Output

- Menu manajemen inventory alat bangunan



```
==== MENU MANAJEMEN INVENTORY ALAT BANGUNAN ====
1. Tambah Alat
2. Lihat Semua Alat
3. Cari Alat
4. Ubah Data Alat
5. Hapus Alat
6. Keluar
Pilih menu (1-6): 1
--- Tambah Alat Baru ---
Nama alat: █
```

Gambar 4.1 Output menu manajemen inventory alat bangunan

- Bagian untuk menambahkan alat baru

```
--- Tambah Alat Baru ---
Nama alat: kayu
Kategori (contoh: cat, semen, kayu, besi): kayu
Jumlah stok: 50
Harga per unit: 50000
Alat 'kayu' berhasil ditambahkan!
```

*Gambar 4.2 Output penambahan alat baru*

- Daftar semua alat bangunan

Pilih menu (1-6): 2					
No	Nama	Kategori	Jumlah	Harga (Rp)	
1	cat	cat	20	50000.00	
2	semen	semen	30	90000.00	
3	kayu	kayu	50	60000.00	
4	besi	besi	60	80000.00	

Tekan Enter untuk kembali ke menu...

*Gambar 4.3 Output daftar semua alat bangunan*

- Cari daftar alat bangunan

Pilih menu (1-6): 3					
Masukkan nama alat yang dicari: cat					
Ditemukan 1 hasil:					
Nama	Kategori	Jumlah	Harga (Rp)		
cat	cat	20	50000.00		

Tekan Enter untuk kembali ke menu...

*Gambar 4.4 Output cari daftar alat bangunan*

- Daftar alat yang ingin di ubah

```
Pilih menu (1-6): 4
+---+-----+-----+-----+-----+
| No | Nama | Kategori | Jumlah | Harga (Rp) |
+---+-----+-----+-----+-----+
| 1 | cat | cat | 20 | 50000.00 |
| 2 | semen | semen | 30 | 90000.00 |
| 3 | kayu | kayu | 50 | 60000.00 |
| 4 | besi | besi | 60 | 80000.00 |
+---+-----+-----+-----+-----+
Masukkan nama alat yang ingin diubah: besi
Nama baru (kosongkan jika tidak diubah): paku
Kategori baru (kosongkan jika tidak diubah):
Jumlah baru (kosongkan jika tidak diubah): 90
Harga baru (kosongkan jika tidak diubah): 20000
Data alat berhasil diperbarui!

Tekan Enter untuk kembali ke menu...■
```

*Gambar 4.5 Output alat yang ingin di*

- Daftar alat yang ingin dihapus

```
Pilih menu (1-6): 5
+---+-----+-----+-----+-----+
| No | Nama | Kategori | Jumlah | Harga (Rp) |
+---+-----+-----+-----+-----+
| 1 | cat | cat | 20 | 50000.00 |
| 2 | semen | semen | 30 | 90000.00 |
| 3 | kayu | kayu | 50 | 60000.00 |
| 4 | paku | besi | 90 | 20000.00 |
+---+-----+-----+-----+-----+
Masukkan nama alat yang ingin dihapus: paku
Alat 'paku' telah dihapus.

Tekan Enter untuk kembali ke menu...■
```

*gambar 4.6 Output alat yang ingin di hapus*

- Program telah selesai

```
==== MENU MANAJEMEN INVENTORY ALAT BANGUNAN ====
1. Tambah Alat
2. Lihat Semua Alat
3. Cari Alat
4. Ubah Data Alat
5. Hapus Alat
6. Keluar
Pilih menu (1-6): 6
Terima kasih! Program selesai.
PS D:\praktikum-apd> █
```

*Gambar 4.7 Output program selesai*

## 5. GIT

### 5.1 GIT Add

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

*Gambar 5.1 GIT add*

Git add itu perintah untuk memasukkan file atau perubahan file kedalam daftar antrian.

### 5.2 GIT Commit

```
PS D:\praktikum-apd> git commit -m "Posttest 8"
[main ad9d16c] Posttest 8
19 files changed, 310 insertions(+), 45 deletions(-)
create mode 100644 angka.txt
delete mode 100644 data.txt
create mode 100644 faris.py
create mode 100644 import angka_random.py
create mode 100644 import random.py
create mode 100644 nilai.txt
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/__pycache__/auth.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/__pycache__/data.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/__pycache__/display.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/__pycache__/inventory.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/__pycache__/utils.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/auth.py
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/data.py
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/display.py
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/inventory.py
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/main.py
create mode 100644 post-test/post-test-apd-8/2509106023-Fathur Rahman-PT-8/utils.py
```

*Gambar 5.2 GIT commit*

Git commit itu digunakan untuk **menyimpan perubahan (save)** yang sudah kamu buat di repositori lokal (komputer kamu sendiri).

Ibaratnya seperti **menekan tombol “save” dengan catatan perubahannya** di proyekmu.

### 5.3 GIT Push

```
PS D:\praktikum-apd> git push
Enumerating objects: 28, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 16 threads
Compressing objects: 100% (23/23), done.
Writing objects: 100% (24/24), 8.55 KiB | 4.27 MiB/s, done.
Total 24 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/FathurRahmanA25/praktikum-apd.git
  78b0ff1..ad9d16c main -> main
PS D:\praktikum-apd>
```

*Gambar 5.3 GIT push*

Git push digunakan untuk **mengirim (upload)** perubahan yang sudah kamu *commit* ke **repositori online** seperti GitHub, GitLab, atau Bitbucket.