**LAPORAN PRAKTIKUM**

**POSTTES 2**

**ALGORITMA PEMROGRAMAN LANJUT**

****

**Disusun oleh:**

**Nama Gilbert Finsen Zakaria Lian(2409106110)**

**Kelas (C‘2)**

**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

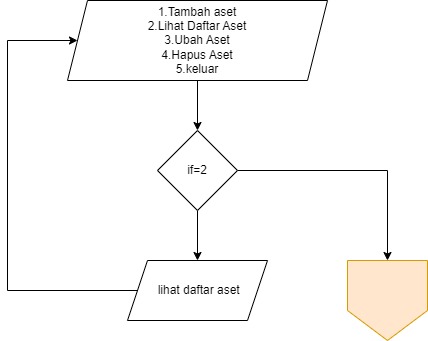
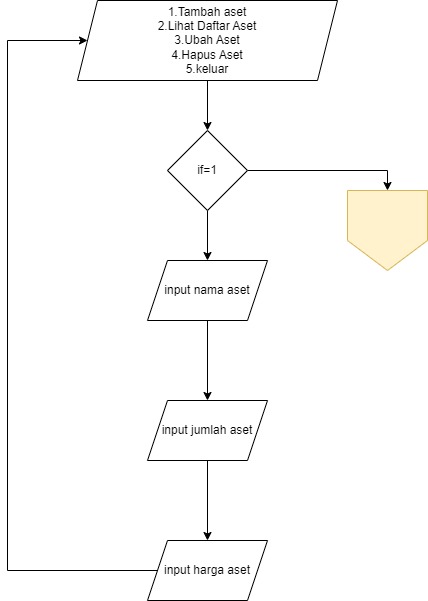
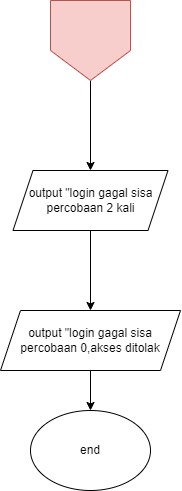
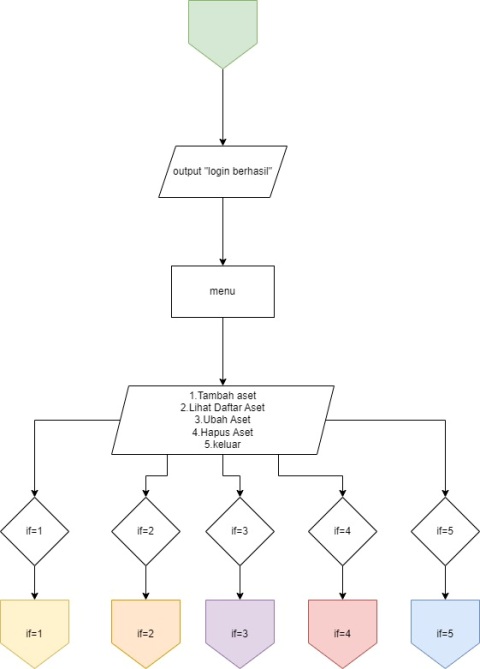
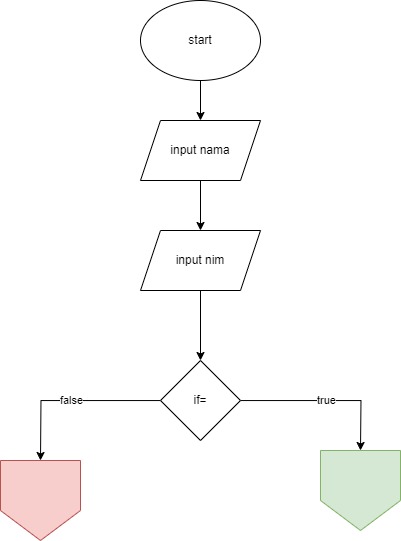
**2025**

# **1. Flowchart**

*(Pada bagian ini, Anda diminta untuk melampirkan flowchart yang menggambarkan alur logika program, mulai dari input, proses, hingga output. Pastikan flowchart dibuat dengan simbol yang sesuai untuk mempermudah pemahaman.)*

**Catatan:**

· Jika flowchart terlalu panjang + ribet pakai **on-page atau off-page reference** untuk memecah fiturnya biar ga burem karena kepanjangan.

********

# 

# 

# 

# 

# **2. Analisis Program**

mari kita analisis program manajemen inventaris aset perusahaan ini secara mendalam.

**2.1 Deskripsi Singkat Program**

Program ini adalah aplikasi konsol sederhana yang dirancang untuk membantu perusahaan dalam mengelola inventaris aset mereka. Fitur utama program ini meliputi:

Login: Memastikan hanya pengguna yang berwenang yang dapat mengakses program.

CRUD (Create, Read, Update, Delete): Memungkinkan pengguna untuk menambah, melihat, mengubah, dan menghapus data aset.

Tampilan Tabel: Menampilkan data aset dalam format tabel yang rapi untuk memudahkan pembacaan.

Tujuan utama program ini adalah untuk memberikan alat yang mudah digunakan untuk melacak dan mengelola aset perusahaan.

**2.2 Penjelasan Alur & Algoritma**

Berikut adalah penjelasan rinci tentang bagaimana program ini berjalan:

Login:

Program memulai dengan memanggil fungsi login().

Fungsi ini meminta pengguna untuk memasukkan nama dan NIM.

Program membandingkan input dengan nilai yang telah ditentukan ("Lian" dan "110").

Jika cocok, login berhasil dan fungsi mengembalikan true. Jika tidak, percobaan login dihitung, dan pengguna diberi kesempatan untuk mencoba lagi hingga 3 kali.

Jika login gagal setelah 3 percobaan, program menampilkan "Akses ditolak." dan keluar.

inisialisasi Data:

Setelah login berhasil, program menginisialisasi array untuk menyimpan data aset: namaAset, jumlahAset, dan hargaAset.

Variabel jumlahAsetTersimpan digunakan untuk melacak jumlah aset yang telah ditambahkan.

Menu Utama:

Program menampilkan menu dengan pilihan:

Tambah Aset (1)

Lihat Daftar Aset (2)

Ubah Aset (3)

Hapus Aset (4)

Keluar (5)

Pengguna memasukkan pilihan mereka.

Pemrosesan Pilihan:

Program menggunakan pernyataan switch untuk memproses pilihan pengguna.

Case 1 (Tambah Aset):

Program meminta pengguna memasukkan nama, jumlah, dan harga aset.

Data aset disimpan dalam array.

jumlahAsetTersimpan ditingkatkan.

Case 2 (Lihat Daftar Aset):

Program menampilkan data aset dalam format tabel yang rapi menggunakan setw() untuk mengatur lebar kolom.

Jika tidak ada aset yang tersimpan, pesan "Daftar aset kosong." ditampilkan.

Case 3 (Ubah Aset):

Program meminta pengguna memasukkan nomor aset yang ingin diubah.

Pengguna memasukkan data aset baru.

Data aset yang sesuai diperbarui dalam array.

Case 4 (Hapus Aset):

Program meminta pengguna memasukkan nomor aset yang ingin dihapus.

Data aset yang sesuai dihapus dari array dengan menggeser elemen-elemen setelahnya.

jumlahAsetTersimpan dikurangi.

Case 5 (Keluar):

Program menampilkan pesan "Terima kasih telah menggunakan program ini." dan keluar.

Default:

Jika pengguna memasukkan pilihan yang tidak valid, pesan "Pilihan tidak valid." ditampilkan.

Perulangan:s

Program mengulangi langkah 3 dan 4 hingga pengguna memilih opsi "Keluar".

Penjelasan Blok Kode:

#include <iostream>, #include <string>, #include <iomanip>: Menyertakan header yang diperlukan untuk input/output, string, dan manipulasi input/output.

using namespace std;: Menggunakan namespace standar untuk menghindari penulisan std:: berulang kali.

bool login(): Fungsi untuk melakukan proses login.

int main(): Fungsi utama program.

Array namaAset, jumlahAset, hargaAset: Array satu dimensi untuk menyimpan data aset.

int jumlahAsetTersimpan: Variabel untuk melacak jumlah aset yang tersimpan.

int pilihan: Variabel untuk menyimpan pilihan menu pengguna.

switch (pilihan): Pernyataan untuk memproses pilihan menu.

setw(): Fungsi untuk mengatur lebar kolom dalam tampilan tabel.

getline(): Fungsi untuk membaca input baris dari pengguna.

cin.ignore(): Fungsi untuk membersihkan buffer input.

Semoga analisis ini memberikan pemahaman yang jelas tentang program ini.

# **3. Source Code**

*(Pada bagian ini, sertakan hanya bagian kode yang berisi fitur penting dari program. Tidak perlu memasukkan seluruh kode, cukup bagian yang relevan dengan fungsionalitas utama.)*

1. Bagian Login

|  |
| --- |
| string nama, nim;  int percobaan = 0;  while (percobaan < 3) {  cout << "Nama: ";  getline(cin >> ws, nama); // Membaca nama dengan spasi  cout << "NIM: ";  getline(cin, nim);  if (nama == "Gilbert Finsen Zakaria Lian" && nim == "2409106110") {  cout << "Login berhasil!\n";  break; // Keluar dari loop jika login berhasil  } else {  percobaan++;  cout << "Login gagal. Sisa percobaan: " << 3 - percobaan << endl;  if (percobaan == 3) {  cout << "Akses ditolak." << endl;  return 0; // Hentikan program jika percobaan login gagal 3 kali  }  }  } |

Gambar 3.1

**Penjelasan:**

Program meminta nama dan NIM dari pengguna.

Pengguna diberikan **3 kesempatan** untuk login. Jika nama adalah "Lian" dan NIM adalah "110", maka login berhasil dan program lanjut ke menu.

Jika gagal login 3 kali berturut-turut, akses akan ditolak dan program dihentikan.

### **2. Data Aset**

|  |
| --- |
| const int MAX\_ASET = 100; string namaAset[MAX\_ASET]; int jumlahAset[MAX\_ASET]; double hargaAset[MAX\_ASET]; int jumlahAsetTersimpan = 0; |

**Gambar 3.2**

**Penjelasan:**

Tiga array disiapkan untuk menyimpan data aset :

namaAset[] untuk nama aset

jumlahAset[] untuk jumlah aset

hargaAset[] untuk harga aset

jumlahAsetTersimpan untuk menyimpan jumlah aset yang sudah disimpan kedalam array

**3. Menu Pilihan Pengguna**

|  |
| --- |
| **int pilihan; do {  cout << "\nMenu:\n";  cout << "1. Tambah Aset\n";  cout << "2. Lihat Daftar Aset\n";  cout << "3. Ubah Aset\n";  cout << "4. Hapus Aset\n";  cout << "5. Keluar\n";  cout << "Pilihan Anda: ";  cin >> pilihan;  cin.ignore(); *// Membersihkan newline setelah input pilihan*** |

**Gambar 3.3**

**Penjelasan:**

-Program menampilkan menu pilihan untuk menambah, melihat, mengubah, menghapus aset, atau keluar.

-Input pilihan pengguna disimpan dalam variabel pilihan dan dihapuskan karakter newline yang tersisa dari input sebelumnya dengan cin.ignore().

**4. Tambah Aset**

|  |
| --- |
| **if (pilihan == 1) { *// Tambah Aset*  if (jumlahAsetTersimpan < MAX\_ASET) {  string nama;    *// Memastikan nama aset tidak mengandung spasi*  cout << "Nama Aset (tanpa spasi): ";  cin >> nama; *// Input langsung tanpa spasi*   while (nama.find(' ') != string::npos) {  cout << "Nama aset tidak boleh mengandung spasi. Coba lagi.\n";  cin >> nama;  }   namaAset[jumlahAsetTersimpan] = nama;   cout << "Jumlah: ";  cin >> jumlahAset[jumlahAsetTersimpan]; *// Bisa negatif*   double harga;  *// Pastikan harga tidak negatif*  do {  cout << "Harga: ";  cin >> harga;  if (harga < 0) {  cout << "Harga tidak boleh negatif. Masukkan harga yang benar!\n";  }  } while (harga < 0); *// Pastikan harga tidak negatif*   hargaAset[jumlahAsetTersimpan] = harga;  jumlahAsetTersimpan++;  cout << "Aset berhasil ditambahkan." << endl;  } else {  cout << "Daftar aset penuh." << endl;  } }** |

**Gambar 3.4**

**Penjelasan:**

-Program meminta input nama aset dan memastikan nama aset tidak mengandung spasi.

-Jika nama aset mengandung spasi, program meminta input ulang.

-Program juga meminta input untuk jumlah dan harga aset. Jumlah bisa bernilai positif atau negatif.

-Harga harus valid (tidak negatif), jika harga negatif dimasukkan, program akan meminta input ulang.

-Aset yang valid ditambahkan ke dalam array dan jumlah aset bertambah.

**5. Lihat Daftar Aset**

|  |
| --- |
| **else if (pilihan == 2) { *// Lihat Daftar Aset*  if (jumlahAsetTersimpan > 0) {  cout << "\nDaftar Aset:\n";  cout << setw(5) << "No" << setw(20) << "Nama Aset" << setw(10) << "Jumlah" << setw(15) << "Harga" << endl;  cout << "----------------------------------------------------" << endl;  for (int i = 0; i < jumlahAsetTersimpan; i++) {  cout << setw(5) << i + 1 << setw(20) << namaAset[i] << setw(10) << jumlahAset[i] << setw(15) << hargaAset[i] << endl;  }  } else {  cout << "Daftar aset kosong." << endl;  } }** |

**Gambar 3.5**

**Penjelasan:**

-Program menampilkan daftar aset yang telah dimasukkan, termasuk nomor, nama, jumlah, dan harga.

-Jika tidak ada aset yang dimasukkan, program akan memberi pesan "Daftar aset kosong."

**6. Ubah Aset**

|  |
| --- |
| **else if (pilihan == 3) { *// Ubah Aset*  int nomor;  cout << "Masukkan nomor aset yang ingin diubah: ";  cin >> nomor;  if (nomor > 0 && nomor <= jumlahAsetTersimpan) {  nomor--; *// Sesuaikan dengan indeks array*  string nama;    *// Memastikan nama aset tidak mengandung spasi*  cout << "Nama Aset Baru : ";  cin >> nama;    while (nama.find(' ') != string::npos) {  cout << "Nama aset tidak boleh mengandung spasi. Coba lagi.\n";  cin >> nama;  }   namaAset[nomor] = nama;  cout << "Jumlah Baru: ";  cin >> jumlahAset[nomor]; *// Bisa negatif*   double harga;  *// Pastikan harga tidak negatif*  do {  cout << "Harga Baru: ";  cin >> harga;  if (harga < 0) {  cout << "Harga tidak boleh negatif. Masukkan harga yang benar!\n";  }  } while (harga < 0); *// Pastikan harga tidak negatif*   hargaAset[nomor] = harga;  cout << "Aset berhasil diubah." << endl;  } else {  cout << "Nomor aset tidak valid." << endl;  } }** |

**Gambar 3.6**

**Penjelasan:**

**-Program meminta input nomor aset yang ingin diubah.**

**-Jika nomor aset valid, program akan meminta input nama, jumlah, dan harga baru.**

**-Nama aset tidak boleh mengandung spasi, dan harga harus positif.**

### **7. Hapus Aset**

|  |
| --- |
| **else if (pilihan == 4) { *// Hapus Aset*  int nomor;  cout << "Masukkan nomor aset yang ingin dihapus: ";  cin >> nomor;  if (nomor > 0 && nomor <= jumlahAsetTersimpan) {  nomor--;   for (int i = nomor; i < jumlahAsetTersimpan - 1; i++) {  namaAset[i] = namaAset[i + 1];  jumlahAset[i] = jumlahAset[i + 1];  hargaAset[i] = hargaAset[i + 1];  }  jumlahAsetTersimpan--;  cout << "Aset berhasil dihapus." << endl;  } else {  cout << "Nomor aset tidak valid." << endl;  } }** |

**Gambar 3.7**

**Penjelasan:**

-Program meminta nomor aset yang ingin dihapus.

-Jika nomor valid, aset dihapus dengan cara menggeser elemen-elemen setelah aset yang dihapus untuk menutupi posisi kosong.

-Jumlah aset dikurangi satu.

**8. Keluar**

|  |
| --- |
| else if (pilihan == 5) { *// Keluar*  cout << "Terima kasih telah menggunakan program ini." << endl; } |

**Gambar 3.8**

**Penjelasan:**

Jika pengguna memilih Keluar, program akan mengucapkan terima kasih dan keluar dari menu.

### 

**Source Code:**

# **4. Uji Coba dan Hasil Output**

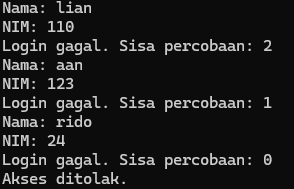
## **4.1 Uji Coba**

Skenario 1. Saya menginput Nama Gilbert Finsen Zakaria Lian dan Nim 2409106110 dan berhasil login.

Skenario 2. Saya menginput Nama Lian dan Nim 110 dan ternyata gagal login

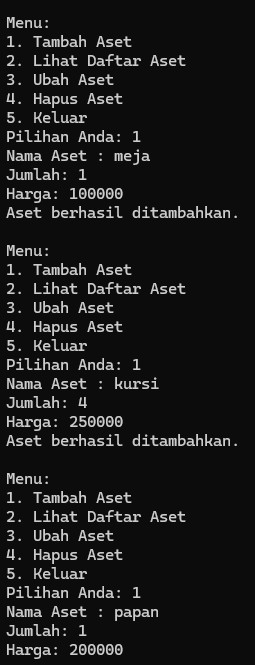


**Gambar 4.0 <login berhasil>**

****

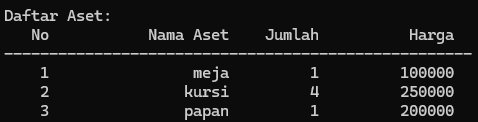
**Gambar 4.1 <login gagal>**

Skenario 3. Saya memilih pilihan 1 saya menginput meja,kursi,papan dengan jumlah 1,4,1,2 dan harga 100000,250000,200000 dan menambahkan aset berhasil



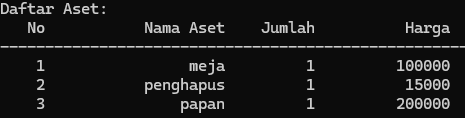
**Gambar 4.2 <Tambah Aset berhasil>**

Skenario 4.Saya ingin melihat daftar aset



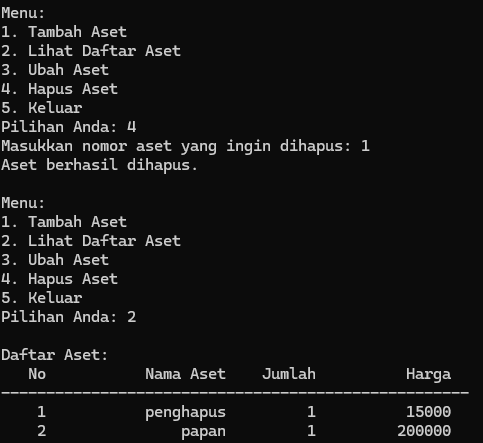
**Gambar 4.3 <Lihat Daftar Aset>**

Skenario 5.Saya ingin mengubah aset nomor 2 menjadi penghapus dengan jumlah 1 dan harga 15000 dan berhasil



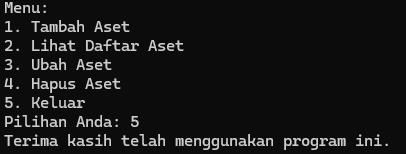
**Gambar 4.4<Ubah Aset>**

Skenario 6.Saya ingin menghapus aset nomor 1

**

**Gambar 4.5 <Hapus Aset>**

Skenario 7. Saya ingin keluar dari program



**Gambar 4.8 <Keluar Dari Program>**

**5.Sistem Github**