LAPORAN PRAKTIKUM

POSTTEST (3)

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Dewi astuti (2409106007)

Kelas (A1’24)

PROGRAM STUDI INFORMATIKA

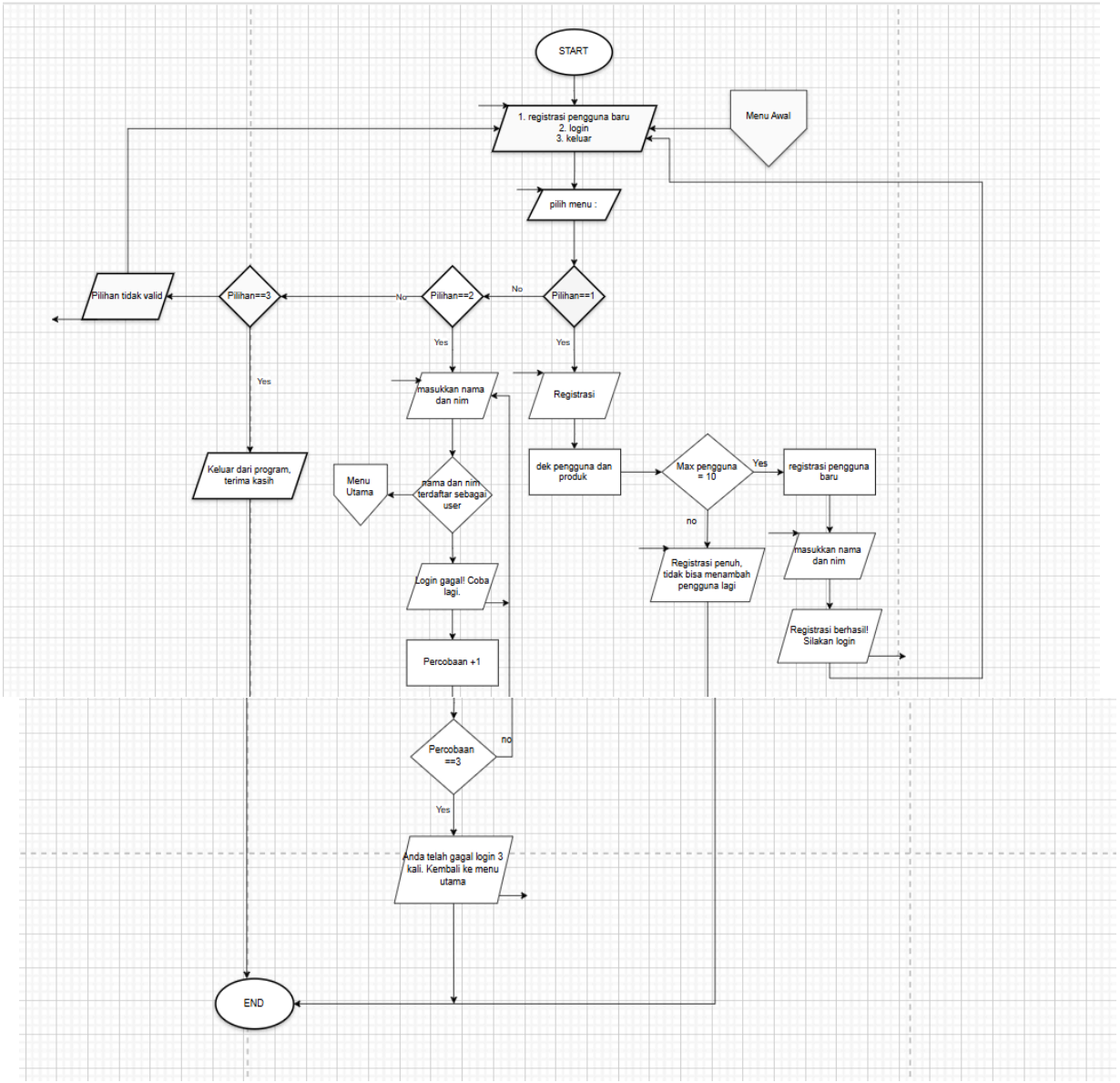
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

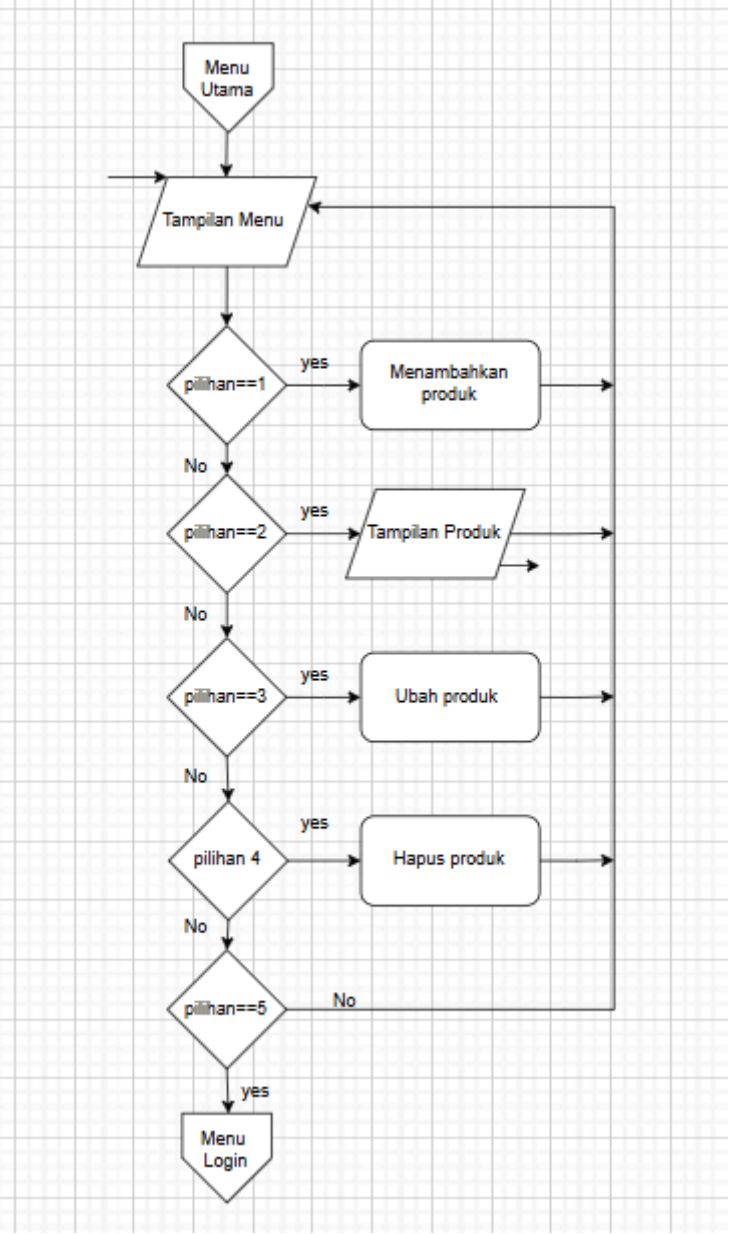
1. Flowchart

1.1 Menu Login



gambar 1.1 Menu Login

1.2 Menu Utama



gambar 1.2 Menu Utama

2. Analisis Program

Program Manajemen Produk Perangkat Jaringan ini dibuat menggunakan bahasa C++ dengan konsep Array of Struct. Program ini dirancang untuk membantu dalam pengelolaan data perangkat jaringan, seperti router, access point, dan switch. Program ini memiliki dua fitur utama, yaitu manajemen pengguna dan manajemen produk perangkat jaringan. Pengguna harus melakukan registrasi dan login terlebih dahulu sebelum dapat mengakses

fitur manajemen produk. Setelah login, pengguna dapat menambah, melihat, memperbarui, dan menghapus produk dalam sistem. Fungsi & Manfaat Utamanya yaitu ;

1. Manajemen Pengguna (Register & Login)
    - Pengguna dapat mendaftarkan akun dengan memasukkan Nama dan NIM
    - Sistem memastikan hanya pengguna yang telah terdaftar yang bisa mengakses fitur manajemen produk.
    - Batas percobaan login sebanyak 3 kali untuk mencegah akses yang tidak sah.
2. Manajemen Produk Perangkat Jaringan (CRUD - Create, Read, Update, Delete)
    - Menambah perangkat jaringan baru ke dalam daftar, seperti router atau switch.
    - Menampilkan daftar perangkat dalam tabel yang rapi dengan informasi nama produk, tipe, harga, dan stok.
    - Memperbarui informasi produk, termasuk harga dan jumlah stok.
    - Menghapus produk dari daftar berdasarkan indeks.
3. Struktur Data yang Digunakan
    - struck pengguna → Menyimpan informasi pengguna yang telah mendaftar.
    - struck produk → Menyimpan informasi perangkat jaringan, seperti nama, tipe, harga, dan stok.
4. Keunggulan Program
    - Mudah digunakan, dengan tampilan menu yang sederhana dan sistem input yang jelas.
    - Menggunakan Array of Struct, sehingga memudahkan pengelolaan data pengguna dan produk.
    - Memastikan keamanan login dengan sistem validasi yang membatasi percobaan login.

3. Source Code

3. 1 Struct untuk Menyimpan Data Pengguna dan Produk\

- struct Pengguna menyimpan data user yang mendaftar, dengan Nama dan NIM sebagai kredensial login.
- struct Produk menyimpan informasi produk perangkat jaringan, termasuk nama, tipe, harga, dan jumlah stok.

```
struct Pengguna {
    string Nama;
    string Nim;
};

struct Produk {
    string nama;
    string tipe;
    int harga;
    int stok;
};
```

3.2 Array untuk Menyimpan Data Pengguna dan Produk

- jumlahPengguna = 0; → Awalnya tidak ada pengguna yang terdaftar.
- daftarProduk[MAX_PRODUK] → Array berisi produk perangkat jaringan awal.
- jumlahProduk = sizeof(daftarProduk) / sizeof(daftarProduk[0]) → Menghitung jumlah produk awal secara otomatis.

```
Pengguna daftarPengguna[MAX_PENGGUNA];
int jumlahPengguna = 0;

Produk daftarProduk[MAX_PRODUK] = {
    {"TP-LINK 2340", "Router", 450000, 2},
    {"Cisco 2901", "Router", 3500000, 5},
    {"MikroTik hAP AC2", "Access Point", 850000, 10}
};

int jumlahProduk = sizeof(daftarProduk) / sizeof(daftarProduk[0]);
// Menentukan jumlah produk awal
```

3. 3 Fitur Registrasi Pengguna

- Cek jumlahPengguna >= MAX_PENGGUNA → Mencegah penambahan user jika kapasitas penuh.
- Menggunakan getline(cin, Nama/Nim); → Memungkinkan input nama lengkap (dengan spasi).

- jumlahPengguna++ → Menambah jumlah pengguna setelah registrasi berhasil.

```
if (pilihan == 1) {
    if (jumlahPengguna >= MAX_PENGGUNA) {
        cout << "Registrasi penuh, tidak bisa menambah pengguna
lagi!\n";
        continue;
    }
    cout << "Masukkan Nama: ";
    getline(cin, daftarPengguna[jumlahPengguna].Nama);
    cout << "Masukkan Nim: ";
    getline(cin, daftarPengguna[jumlahPengguna].Nim);
    jumlahPengguna++;
    cout << "Registrasi berhasil! Silakan login.\n";
}
```

3.4 Fitur Login Pengguna

- maksimal 3 kali percobaan login.
- Pencarian dalam array → for loop mencari user berdasarkan Nama dan NIM.
- loginIndex = i; → Jika ditemukan, user dianggap berhasil login.
- Jika loginIndex == -1 setelah 3 percobaan → User dikembalikan ke menu utama.

```
if (pilihan == 2) {
    string Nama, Nim;
    int attempts = 0;
    while (attempts < 3) {
        cout << "Masukkan Username: ";
        getline(cin, Nama);
        cout << "Masukkan Password: ";
        getline(cin, Nim);

        for (int i = 0; i < jumlahPengguna; i++) {
            if (daftarPengguna[i].Nama == Nama &&
daftarPengguna[i].Nim == Nim) {
                loginIndex = i;
                cout << "Login berhasil! Selamat datang, " <<
daftarPengguna[i].Nama << ".\n";
                break;
            }
        }
        if (loginIndex != -1) break;
        attempts++;
        cout << "Login gagal! Coba lagi.\n";
    }
```

```
    if (loginIndex == -1) {
        cout << "Anda telah gagal login 3 kali. Kembali ke menu
utama.\n";
        continue;
  }
}
```

3. 5 Menambah Produk Baru (CREATE - CRUD)

- Looping while (attempts < 3) → Memberikan
  Cek jumlahProduk >= MAX_PRODUK → Menghindari penambahan data melebihi kapasitas array.
- Gunakan getline() untuk string → Agar input tipe produk tidak terpotong.
- Tambahkan cin.ignore(); setelah input angka → Mencegah error saat membaca string berikutnya.
- jumlahProduk++ → Menambah jumlah produk setelah berhasil ditambahkan.

```
if (pilihan == 1) { // Tambah Produk
    if (jumlahProduk >= MAX_PRODUK) {
        cout << "Data produk penuh, tidak bisa menambahkan
lagi!\n";
        continue;
    }

    cout << "Masukkan Nama Produk: ";
    getline(cin, daftarProduk[jumlahProduk].nama);
    cout << "Masukkan Tipe Produk: ";
    getline(cin, daftarProduk[jumlahProduk].tipe);
    cout << "Masukkan Harga Produk (Rp): ";
    cin >> daftarProduk[jumlahProduk].harga;
    cout << "Masukkan Stok Produk: ";
    cin >> daftarProduk[jumlahProduk].stok;
    cin.ignore();

    jumlahProduk++;
    cout << "Produk berhasil ditambahkan!\n";
}
```

3. 6 Menampilkan Daftar Produk (READ - CRUD)

- Gunakan setw() → Agar tampilan data produk lebih rapi dan mudah dibaca.
- Loop for → Menampilkan semua produk yang tersimpan dalam array.

```
if (pilihan == 2) { // Tampilkan Produk
    cout << "\nDaftar Produk:\n";
    cout << left << setw(5) << "No"
```

```
         << setw(20) << "Nama Produk"
         << setw(20) << "Tipe"
         << setw(15) << "Harga (Rp)"
         << setw(10) << "Stok" << endl;
    cout << string(70, '-') << endl;

    for (int i = 0; i < jumlahProduk; i++) {
        cout << left << setw(5) << i
             << setw(20) << daftarProduk[i].nama
             << setw(20) << daftarProduk[i].tipe
             << setw(15) << daftarProduk[i].harga
             << setw(10) << daftarProduk[i].stok << endl;
    }
}
```

3.7 Mengupdate Produk (UPDATE - CRUD)

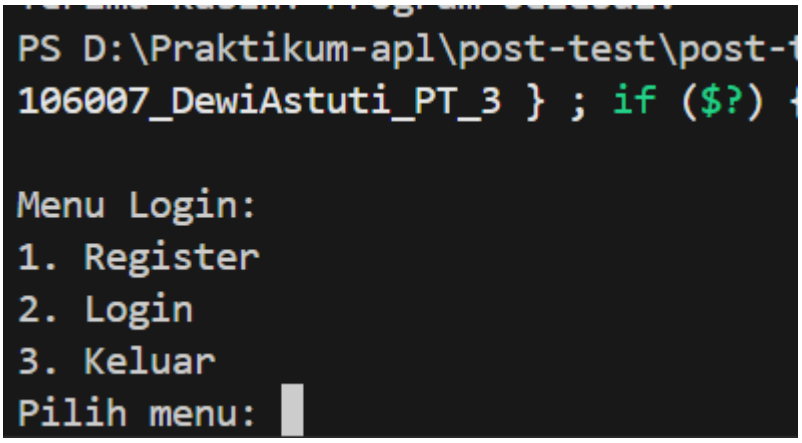
```
if (pilihan == 3) { // Update Produk
    cout << "Masukkan nomor produk yang ingin diupdate: ";
    int index;
    cin >> index;
    cin.ignore();

    if (index >= 0 && index < jumlahProduk) {
        cout << "Masukkan Nama Produk baru: ";
        getline(cin, daftarProduk[index].nama);
        cout << "Masukkan Tipe Produk baru: ";
        getline(cin, daftarProduk[index].tipe);
        cout << "Masukkan Harga Produk baru (Rp): ";
        cin >> daftarProduk[index].harga;
        cout << "Masukkan Stok Produk baru: ";
        cin >> daftarProduk[index].stok;
        cout << "Produk berhasil diupdate!\n";
    } else {
        cout << "Index tidak valid!\n";
    }
}
```

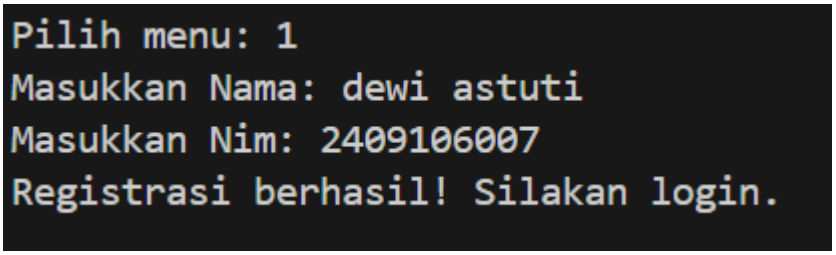
- Cek apakah index valid → Agar tidak terjadi error saat mengakses array.
- Menggunakan getline() untuk input string → Menghindari error ketika mengedit nama atau tipe produk.

4. Uji Coba dan Hasil Output

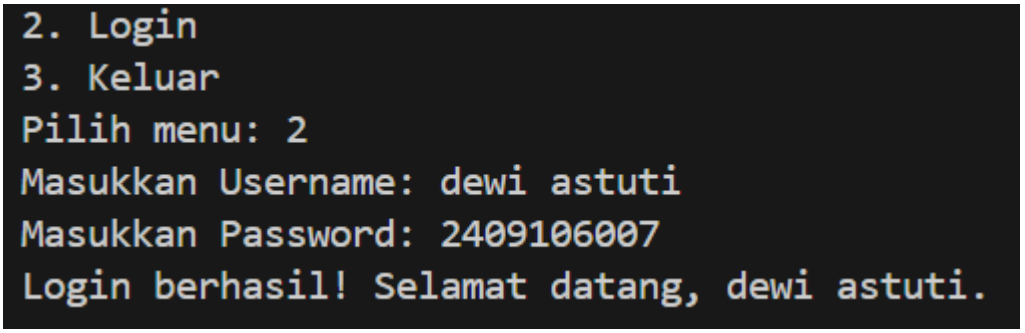
4. 1 Masuk Ke menu Login



4.2 Masuk ke menu 1 yaitu register



4.3 setelah melakukan registrasi bisa masuk ke menu login



4.4 setelah login kita akan masuk ke menu utama lalu pilih menu 2

```
Menu Utama:
1. Tambah Produk
2. Tampilkan Produk
3. Update Produk
4. Hapus Produk
5. Logout
Pilih menu: 2

Daftar Produk:
No   Nama Produk         Tipe                Harga (Rp)     Stok
-------------------------------------------------------------------
0    TP-LINK 2340        Router              450000         2
1    Cisco 2901          Router              3500000        5
2    MikroTik hAP AC2    Access Point        850000         10
```

4.5 setelah melihat tampilan produk kita lalu kita pilih no 1 tambah produk

```
Pilih menu: 1
Masukkan Nama Produk: NAS (Network Attached storage)
Masukkan Tipe Produk: cloud storage
Masukkan Harga Produk (Rp): 2000000
Masukkan Stok Produk: 3
Produk berhasil ditambahkan!
```

4. 6 lalu kita akan melihat produk yang sudah ditambahkan, kita akan memilih menu 2

```
Daftar Produk:
No   Nama Produk         Tipe                Harga (Rp)     Stok
-------------------------------------------------------------------
0    TP-LINK 2340        Router              450000         2
1    Cisco 2901          Router              3500000        5
2    MikroTik hAP AC2    Access Point        850000         10
3    NAS (Network Attached storage)cloud storage       2000000        3
```

4.7 Setelah itu kita lihat, lalu kita pilih menu ke 3 yaitu update

```
Pilih menu: 3
Masukkan nomor produk yang ingin diupdate: 3
Masukkan Nama Produk baru: NAS
Masukkan Tipe Produk baru: Storage Cloud
Masukkan Harga Produk baru (Rp): 2500000
Masukkan Stok Produk baru: 4
Produk berhasil diupdate!
```

4.8 Setelah itu kita pilih menu 4 yaitu hapus

```
Pilih menu: 4
Masukkan nomor produk yang ingin dihapus: 3
Produk berhasil dihapus!

Menu Utama:
1. Tambah Produk
2. Tampilkan Produk
3. Update Produk
4. Hapus Produk
5. Logout
Pilih menu:
```

4.9 setelah itu kita lihat tampilan produk apakah produk yang kita hapus sudah terhapus

```
Daftar Produk:
No   Nama Produk         Tipe              Harga (Rp)     Stok
-----------------------------------------------------------------
0    TP-LINK 2340        Router            450000         2
1    Cisco 2901          Router            3500000        5
2    MikroTik hAP AC2    Access Point      850000         10
```

4.10 lalu setelah itu kita pilih keluar untuk kembali kemenu login

```
5. Logout
Pilih menu: 5
Logout berhasil. Kembali ke menu utama.

Menu Login:
1. Register
2. Login
3. Keluar
Pilih menu:
```

4.11 Setelah itu pilih menu ke 3 lalu keluar

```
Menu Login:
1. Register
2. Login
3. Keluar
Pilih menu: 3
Terima kasih! Program selesai.
PS D:\Praktikum-apl\post-test\post-test3>
```

5. GIT

5.1 Git Init (Inisiasi Repository Git), lalu git add . , lalu gunakan perintah git commit -m “Finish Commit Post-test 3.

```
PS D:\Praktikum-apl> git init
Reinitialized existing Git repository in D:/Praktikum-apl/.git/
PS D:\Praktikum-apl> git add .
PS D:\Praktikum-apl> git commit -m "Finish Commit Post-Test 3"
[main 956cb96] Finish Commit Post-Test 3
 5 files changed, 176 insertions(+), 1 deletion(-)
 create mode 100644 kelas/pertemuan-3/Modul 3 - Struct.pdf
 create mode 100644 post-test/post-test3/2409106007_DewiAstuti_PT_3.cpp
 create mode 100644 post-test/post-test3/2409106007_DewiAstuti_PT_3.exe
```

5.2 Lalu ketik ‘git push -u origin main’, untuk mengupload semua file tadi ke cloud github.

```
PS D:\Praktikum-apl> git push origin main
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 850.62 KiB | 4.97 MiB/s, done.
Total 12 (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/dewiastt/praktikum-apl.git
   77a79b5..956cb96  main -> main
PS D:\Praktikum-apl>
```