

LAPORAN PRAKTIKUM

POSTTEST (4)

ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:

Dewi astuti (2409106007)

Kelas (A1'24)

PROGRAM STUDI INFORMATIKA

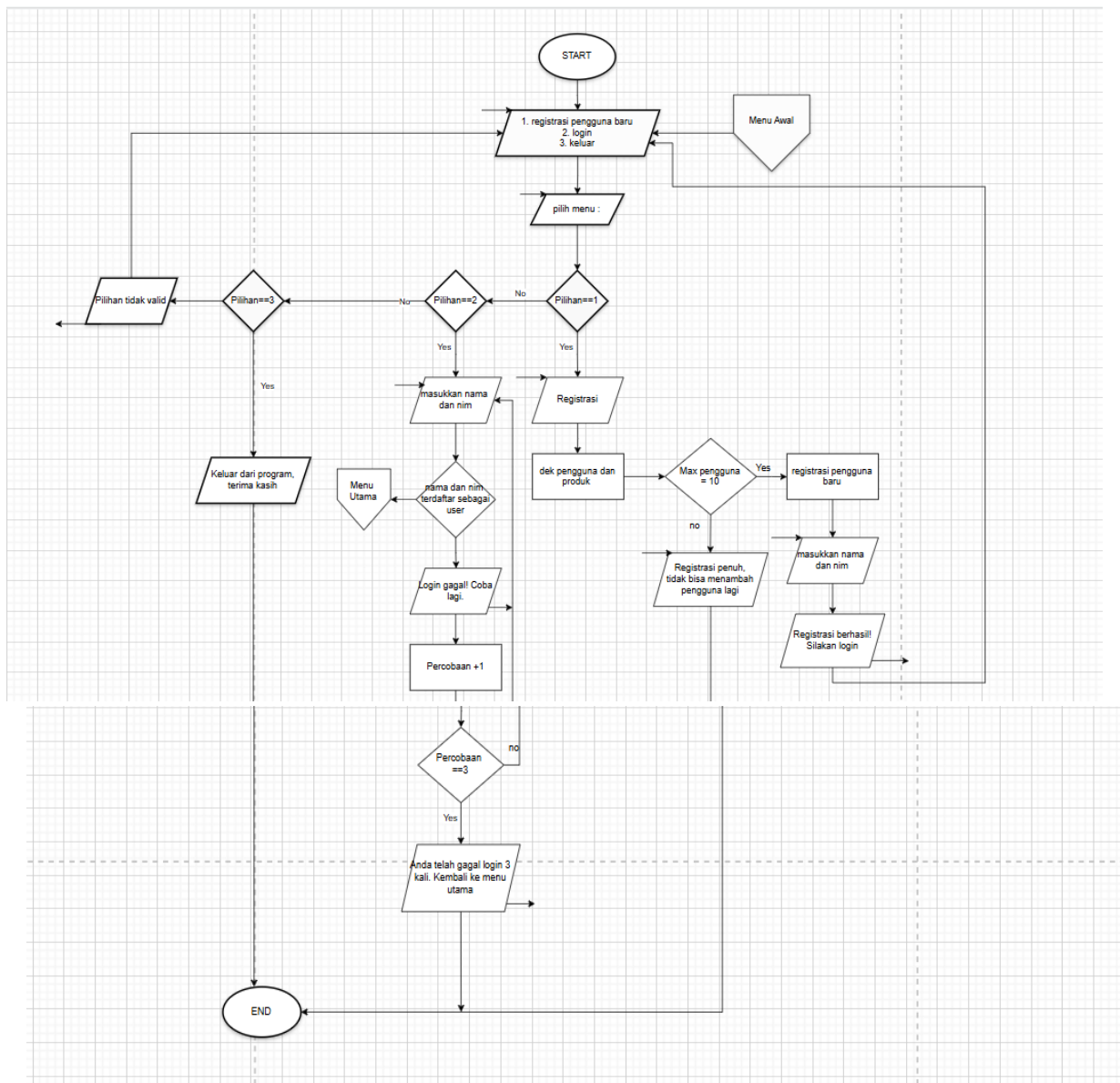
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

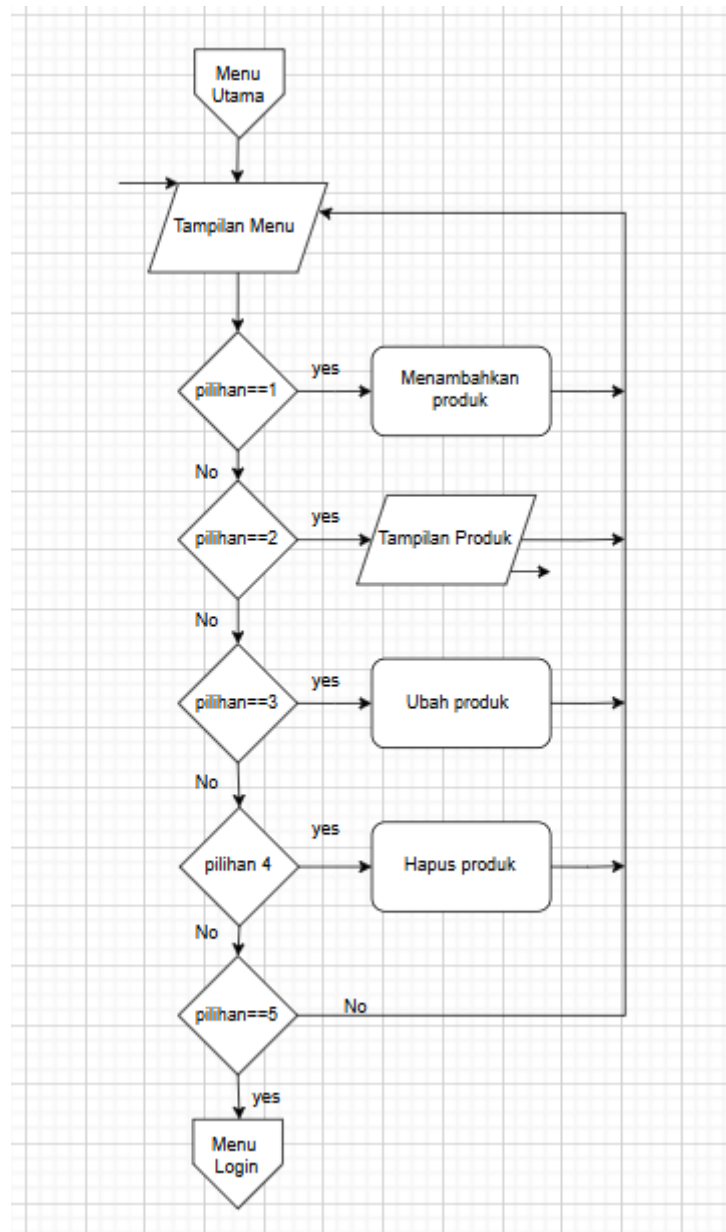
1. Flowchart

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
 - struct pengguna → Menyimpan informasi pengguna yang telah mendaftar.
 - struct 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 Struktur Data

Struct Produk menyimpan semua informasi penting tentang produk perangkat jaringan seperti kode, nama, tipe, harga, dan jumlah stok.

```
struct Profil {  
    string Nama;  
    string Nim;  
};  
  
struct Pengguna {  
    Profil profil;  
};  
  
struct Produk {  
    string kode;  
    string nama;  
    string tipe;  
    int harga;  
    int stok;  
};
```

3.2 Data Global

Array daftarProduk menyimpan daftar produk yang tersedia. Tiga produk awal langsung diisi. Variabel jumlahProduk menyimpan jumlah produk yang aktif.

```
Produk daftarProduk[MAX_PRODUK] = {  
    {"P001", "TP-LINK 2340", "Router", 450000, 2},  
    {"P002", "Cisco 2901", "Router", 3500000, 5},  
    {"P003", "MikroTik hAP AC2", "Access Point", 850000, 10}  
};  
int jumlahProduk = 3;
```

3.3 Fungsi Rekursif untuk Menampilkan Produk

Ini adalah fungsi rekursif yang menampilkan produk satu per satu dan menjumlahkan total stok. Fungsi ini akan terus memanggil dirinya sendiri hingga semua produk ditampilkan.

```
void tampilkanProduk(Produk arr[], int index, int n, int &total) {  
    if (index >= n) return;  
    cout << left << setw(5) << index  
        << setw(10) << arr[index].kode  
        << setw(20) << arr[index].nama  
        << setw(20) << arr[index].tipe  
        << setw(15) << arr[index].harga  
        << setw(10) << arr[index].stok << endl;  
    total += arr[index].stok;  
    tampilkanProduk(arr, index + 1, n, total);  
}
```

3.4 Fungsi Tambah Produk

Fungsi ini digunakan untuk menambahkan produk baru ke dalam daftar. Pengguna diminta mengisi data produk, lalu jumlahProduk ditambah 1.

```
void tambahProduk(Produk daftar[], int &jumlah) {  
    if (jumlah >= MAX_PRODUK) return;  
  
    cout << "Masukkan Kode Produk : ";  
    getline(cin, daftar[jumlah].kode);  
    cout << "Masukkan Nama Produk : ";  
    getline(cin, daftar[jumlah].nama);  
    cout << "Masukkan Tipe Produk : ";  
    getline(cin, daftar[jumlah].tipe);  
    cout << "Masukkan Harga Produk (Rp): ";  
    cin >> daftar[jumlah].harga;  
    cout << "Masukkan Stok Produk : ";  
    cin >> daftar[jumlah].stok;  
    cin.ignore();  
  
    jumlah++;  
}
```

```
}
```

3.5 Fungsi Login

Fungsi ini meminta nama dan NIM dari pengguna. Jika cocok dengan data pengguna yang terdaftar, maka login berhasil. Jika gagal 3 kali, login diblokir sementara.

```
int login(Pengguna daftar[], int jumlah) {  
    string Nama, Nim;  
    int attempts = 0;  
    while (attempts < 3) {  
        getline(cin, Nama);  
        getline(cin, Nim);  
        for (int i = 0; i < jumlah; i++) {  
            if (daftar[i].profil>Nama == Nama && daftar[i].profil.Nim == Nim) {  
                return i;  
            }  
        }  
        attempts++;  
    }  
    return -1;  
}
```

3.6 Fungsi Utama main()

Fungsi main() adalah pusat kendali program. Dimulai dari menu login (register atau login), lalu masuk ke menu utama jika berhasil login.a

```
int main() {  
    do {  
        cout << "Menu Login\n";  
        cout << "1. Register\n2. Login\n3. Keluar\n";  
        cin >> pilihan;  
        cin.ignore();  
    } while (pilihan != 3);  
}
```

```

    if (pilihan == 1) registrasi(...);
    else if (pilihan == 2) {
        loginIndex = login(...);
        if (loginIndex != -1) {
            do {
                cout << "Menu Utama\n1. Tambah\n2. Tampil\n3. Update\n4. Hapus\n5.
Logout\n";
                cin >> pilihan;
                ...
            } while (true);
        }
    } while (pilihan != 3);
}

```

4. Uji Coba dan Hasil Output

4. 1 Masuk Ke menu Login

```

=====
|   Menu Login   |
=====
| 1 | Register   |
| 2 | Login      |
| 3 | Keluar     |
=====
Pilih menu:

```

4.2 Masuk ke menu 1 yaitu register

```

Pilih menu: 1
Masukkan Nama: dewi astuti
Masukkan Nim: 2409106007
Registrasi berhasil! Silakan login.

```

4.3 setelah melakukan registrasi bisa masuk ke menu login


```

2. Login
3. Keluar
Pilih menu: 2
Masukkan Username: dewi astuti
Masukkan Password: 2409106007
Login berhasil! Selamat datang, dewi astuti.

```

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   Kode   Nama Produk   Tipe   Harga (Rp)   Stok
-----
0    P001    TP-LINK 2340  Router   450000       2
1    P002    Cisco 2901    Router   3500000      5
2    P003    MikroTik hAP AC2  Access Point  850000      10

Total seluruh stok produk: 17 unit

```

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

```

Pilih menu: 1
Masukkan Kode Produk : p008
Masukkan Nama Produk : NAS
Masukkan Tipe Produk : 23asnw
Masukkan Harga Produk (Rp): 11239100
Masukkan Stok Produk : 2
Produk berhasil ditambahkan!
-----

```

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

Daftar Produk:

No	Kode	Nama Produk	Tipe	Harga (Rp)	Stok
0	P001	TP-LINK 2340	Router	450000	2
1	P002	Cisco 2901	Router	3500000	5
2	P003	MikroTik hAP AC2	Access Point	850000	10
3	p008	NAS	23asnw	11239100	2

Total seluruh stok produk: 19 unit

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 Kode Produk baru : poo9
Masukkan Nama Produk baru : buah
Masukkan Tipe Produk baru : nans
Masukkan Harga Produk baru (Rp): 1234
Masukkan Stok Produk baru : 1
Produk berhasil diupdate!
```

4.8 Setelah itu kita pilih menu 4 yaitu hapus, setelah itu kita lihat tampilan produk apakah produk yang kita hapus sudah terhapus

```
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: 2

Daftar Produk:

No	Kode	Nama Produk	Tipe	Harga (Rp)	Stok
0	P001	TP-LINK 2340	Router	450000	2
1	P002	Cisco 2901	Router	3500000	5
2	P003	MikroTik hAP AC2	Access Point	850000	10

Total seluruh stok produk: 17 unit

4.10 lalu setelah itu kita pilih keluar untuk kembali kemenu login

```

=====
|           Menu Utama           |
=====
| 1 | Tambah Produk              |
| 2 | Tampilkan Produk           |
| 3 | Update Produk              |
| 4 | Hapus Produk               |
| 5 | Logout                     |
=====
Pilih menu: 5
Logout berhasil. Kembali ke menu utama.
=====
|           Menu Login           |
=====
| 1 | Register                   |
| 2 | Login                      |
| 3 | Keluar                     |
=====

```

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-test4>

```

5. GIT

5.1 Git Init (Inisiasi Repository Git), lalu git add ., lalu gunakan perintah git commit -m “Finish Commit Post-test 4.

```

PS D:\Praktikum-apl> git init
Reinitialized existing Git repository in D:/Praktikum-apl/.git/
PS D:\Praktikum-apl> git add .
warning: in the working copy of '.vscode/settings.json', LF will be replaced by CRLF the next time Git touches it

```

```

PS D:\Praktikum-apl> git commit -m "Finish Post-test 4"
[main 9dcb55a] Finish Post-test 4
4 files changed, 228 insertions(+), 1 deletion(-)
create mode 100644 kelas/pertemuan-4/Modul 4 - Konsep Lanjutan Fungsi.pdf
create mode 100644 post-test/post-test4/2409106007_DewiAstuti_PT_4.cpp
create mode 100644 post-test/post-test4/2409106007_DewiAstuti PT 4.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: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 832.65 KiB | 9.68 MiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/dewiasttt/praktikum-apl.git
88b0fb0..9dcb55a main -> main
PS D:\Praktikum-apl>

```

SELESAIII