

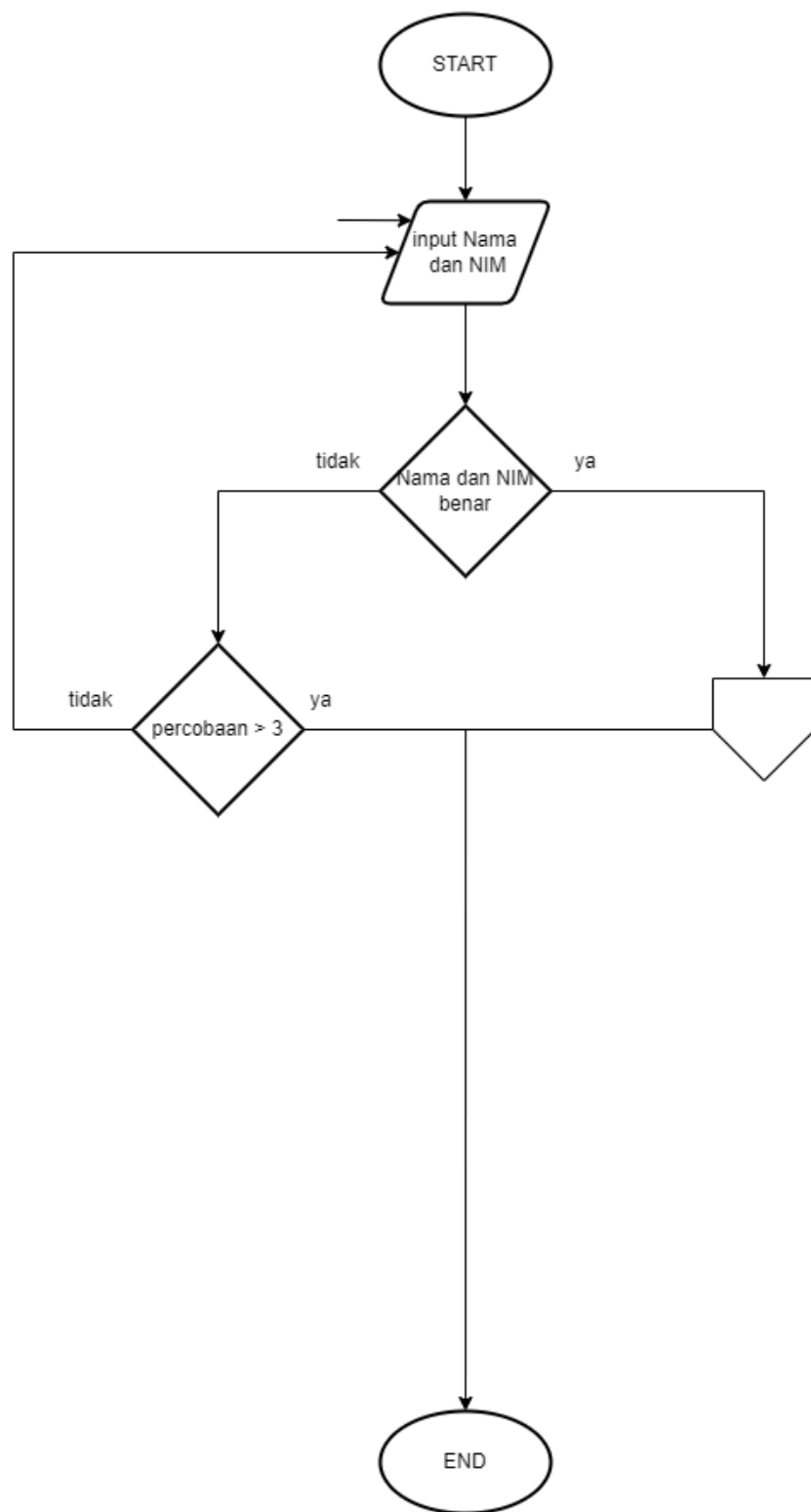
LAPORAN PRAKTIKUM
POSTTEST (2)
ALGORITMA PEMROGRAMAN LANJUT



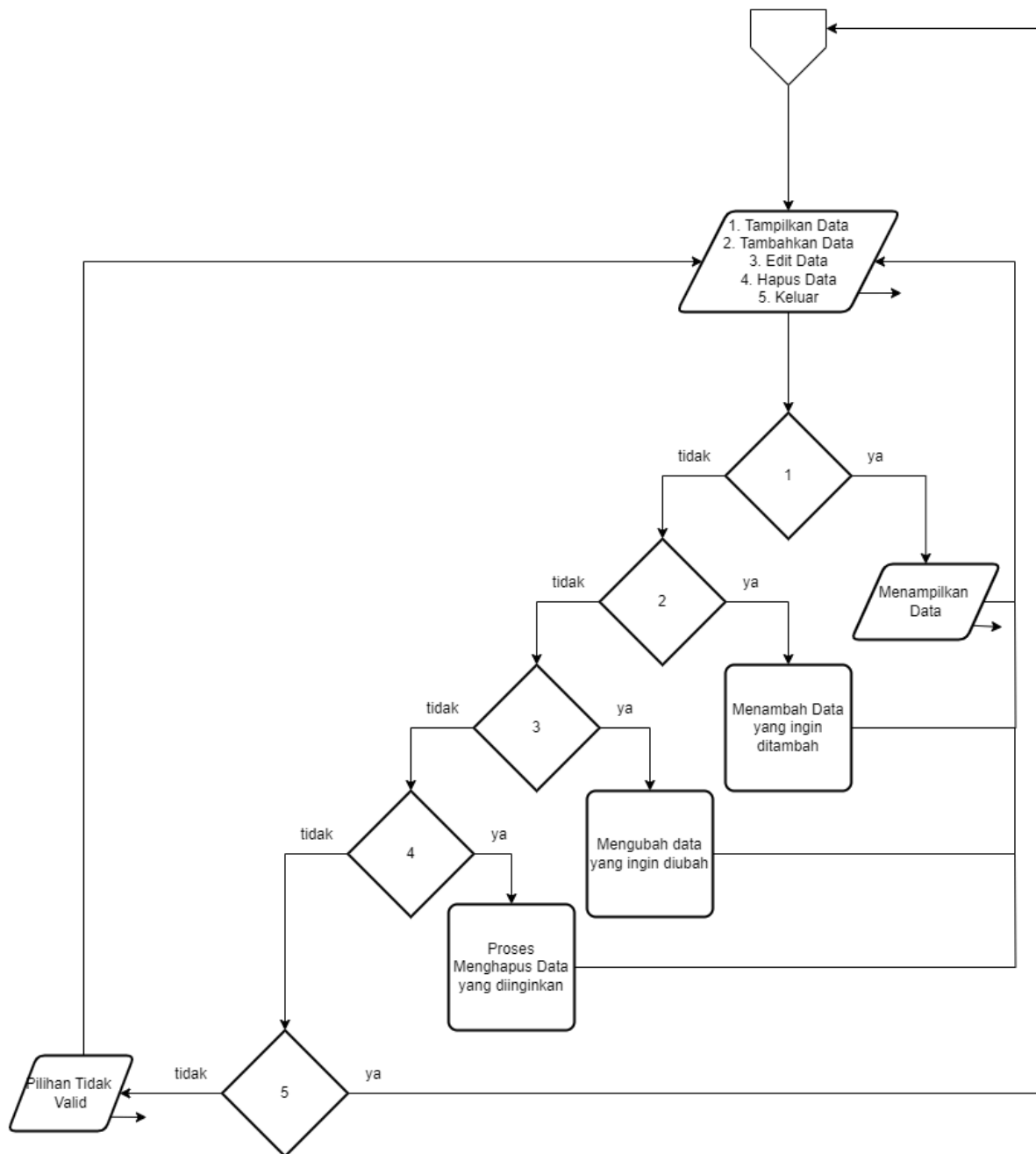
Disusun oleh:
Zulfikar Aryawinata (2409106020)
Kelas (A1'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 flowchart



Gambar 1.2 flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dibuat untuk membantu dalam mengatur pendataan HotWheels yang dijual, mulai dari pendataan nama HotWheels, Seri, Tahun Rilis, Harga, dan Stok barangnya.

2.2 Penjelasan Alur & Algoritma

1. Masukkan Nama dan NIM.
2. Program akan melakukan pengecekan, apakah Nama dan NIM nya sudah benar atau tidak. Jika sudah benar, maka program akan berlanjut ke menu utama. Jika tidak, user akan diberi 2 kali kesempatan lagi untuk mencoba. Jika masih gagal, program akan berhenti secara otomatis.
3. Di dalam menu utama, user akan diberi 5 pilihan. Yang pertama ada pilihan tampilkan data. Pada pilihan ini, user akan diberikan output berupa daftar HotWheels beserta keterangan lainnya seperti Seri, Tahun Rilis, dll.
4. Pada pilihan 2, user dapat menambah HotWheels baru ke dalam daftar, dan akan diminta untuk mengisi mulai dari nama HotWheels, tahun rilis, seri, harga, dan stok yang masuk.
5. Pada pilihan 3, user dapat mengubah informasi HotWheels dengan mengetikkan nama HotWheels yang sedang dijual. Informasi yang dapat diubah yaitu harga dan stok barang.
6. Pada pilihan 4, user dapat menghapus data HotWheels tertentu dengan mengetikkan nama HotWheels-nya apabila Hotwheels tersebut sudah tidak dijual lagi dan tidak ada restock barang.
7. Pilihan 5 berfungsi untuk menutup program.

3. Source Code

A. Fitur Menambah Data

Fitur ini digunakan untuk menambahkan data HotWheels baru.

Source-Code:

```
else if (pilihan == 2) {
    if (jumlah >= 100) {
        cout << "(X)Data penuh! Tidak bisa menambahkan lagi." << endl;
    } else {
        cout << endl;
        cout << "<=>|Masukkan Nama HotWheels : "; getline(cin,
nama_hotwheels[jumlah]);
        cout << "<=>|Masukkan Seri : "; getline(cin, seri[jumlah]);
        cout << "<=>|Masukkan Tahun Rilis : "; cin >>
tahun_rilis[jumlah];
        cout << "<=>|Masukkan Harga : "; cin >> harga[jumlah];
        cout << "<=>|Masukkan jumlah stok masuk :"; cin >> stok[jumlah];
        cout << endl;
        jumlah++;
        cout << "==Data Berhasil Ditambahkan!==" << endl;
        cin.ignore();
    }
}
```

B. Menu Ubah Data

Fitur ini digunakan untuk mengubah data dari HotWheels yang dijual.

Source-Code:

```
else if (pilihan == 3) {
    string nama_hotwheels_edit;
    cout << "Masukkan Nama HotWheels yang ingin diubah datanya : ";
    getline(cin, nama_hotwheels_edit);
    cout << endl;
    for (int i = 0; i < jumlah; i++) {
        if (nama_hotwheels[i] == nama_hotwheels_edit) {
            cout << "Data ditemukan. Masukkan data baru:" << endl;
            cout << "<=>|Masukkan Harga Baru : "; cin >> harga[i];
            cout << "<=>|Masukkan Stok Baru :"; cin >> stok[i];
            cout << endl;
            cout << "|==|Data berhasil diperbarui!";
            cout << endl;
            break;
        }
        else{
            cout << "(X)Data Tidak Ditemukan" << endl;
            cout << endl;
        }
    }
}
```

C. Menu Hapus Data

Fitur ini digunakan untuk menghapus data HotWheels yang sudah tidak dijual.

Source-Code:

```
else if (pilihan == 4) {
    string nama_hotwheels_hapus;
    cout << endl;
    cout << "<=>|Masukkan Nama HotWheels yang ingin dihapus : ";
    getline(cin, nama_hotwheels_hapus);
    for (int i = 0; i < jumlah; i++) {
        if (nama_hotwheels[i] == nama_hotwheels_hapus) {
            for (int j = i; j < jumlah - 1; j++) {
                nama_hotwheels[j] = nama_hotwheels[j + 1];
                seri[j] = seri[j + 1];
                tahun_rilis[j] = tahun_rilis[j + 1];
                harga[j] = harga[j + 1];
                stok[j] = stok[j + 1];
            }
            jumlah--;
            cout << endl;
            cout << "|==|Data Berhasil Dihapus|==" << endl;
            break;
        }
    }
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Menambah data HotWheels baru.

- Input:

User melakukan input nama HotWheels, seri-nya, tahun rilis, harga, dan jumlah stok.

- Ekspektasi:

Data berhasil disimpan oleh program.

2. Mengubah/Edit data HotWheels

- Input:

User diminta untuk menginput nama HotWheels yang ingin diubah datanya.

Jika nama sudah benar, user akan diminta untuk memasukkan inputan data baru untuk harga dan stok HotWheels.

- Ekspektasi:

Sistem menyimpan ulang data sehingga data ter-update.

Data tidak dapat diubah jika terdapat kesalahan dalam penulisan nama HotWheels.

4.2 Hasil Output

```
|2. Tambah Data
|3. Edit Data
|4. Hapus Data
=====
|5. Keluar
=====

Pilih menu (1-5) : 2

<=>|Masukkan Nama HotWheels : Ford Mustang
<=>|Masukkan Seri : Tuner
<=>|Masukkan Tahun Rilis : 2014
<=>|Masukkan Harga : 25000
<=>|Masukkan jumlah stok masuk :45

==Data Berhasil Ditambahkan!==

=====
|=== Manajemen Penjualan Hot Wheels ===|
|1. Tampilkan Data
|2. Tambah Data
|3. Edit Data
|4. Hapus Data
=====
|5. Keluar
=====

Pilih menu (1-5) : 1

Daftar HotWheels:
Nama          Seri          Tahun Rilis   Harga(Rp)  Stok
-----
Bone Shaker    Hot Rods      2006          50000      12
Twin Mill      Legends       1969          1200000    2
Deora II       Surf         2000          75000      13
Rip Rod        Concept       2014          25000      90
Rodger Dodger  Muscle       1974          200000     5
Night Shifter  Street       2018          25000      120
Ford Mustang   Tuner        2014          25000      45

△ 0
```

Gambar 4.2.1 Output Skenario 1


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

|4. Hapus Data |
=====|
|5. Keluar |
=====|

Pilih menu (1-5) : 3
Masukkan Nama Hotwheels yang ingin diubah datanya : Rip Rod

(X)Data Tidak Ditemukan

(X)Data Tidak Ditemukan

(X)Data Tidak Ditemukan

Data ditemukan. Masukkan data baru:
<=>|Masukkan Harga Baru : 30000
<=>|Masukkan Stok Baru :34

|==|Data berhasil diperbarui!

|=== Manajemen Penjualan Hot Wheels ===|
|1. Tampilkan Data |
|2. Tambah Data |
|3. Edit Data |
|4. Hapus Data |
=====|
|5. Keluar |
=====|

Pilih menu (1-5) : 1

Daftar Hotwheels:
Nama          Seri          Tahun Rilis  Harga(Rp)  Stok
-----
Bone Shaker   Hot Rods      2006        50000      12
Twin Mill     Legends       1969        1200000    2
Deora II      Surf          2000        75000      13
Rip Rod       Concept       2014        30000      34
Rodger Dodger Muscle        1974        200000     5
```

Gambar 4.2.2 Output Skenario 2

5. Langkah Langkah Git

```
PS C:\Praktikum\praktikum-apl> git add post-test/post-test-apl-2
PS C:\Praktikum\praktikum-apl> git add .
PS C:\Praktikum\praktikum-apl> git commit -m "Finish Post Test 2"
[main cd33b03] Finish Post Test 2
 2 files changed, 150 insertions(+)
 create mode 100644 post-test/post-test-apl-2/2409106020-ZulfikarAryawinata-PT-2.cpp
 create mode 100644 post-test/post-test-apl-2/2409106020-ZulfikarAryawinata-PT-2.exe
PS C:\Praktikum\praktikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 679.62 KiB | 8.60 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MamangolivE/praktikum-apl.git
   9e4a5d4..cd33b03  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Praktikum\praktikum-apl>
```

1. git add post-test/post-test-apl-2

Menambahkan folder post-test-apl-2 ke dalam staging area, agar perubahan dapat dikomit.

2. git add .

Menambahkan semua perubahan yang ada dalam proyek ke staging area (termasuk file baru, perubahan, atau penghapusan).

3. git commit -m "Finish Post Test 2"

Menyimpan perubahan yang telah ditambahkan ke dalam repository lokal dengan pesan "Finish Post Test 2".

4. git push -u origin main

Mengunggah perubahan dari repository lokal ke repository GitHub pada branch main.

Opsi -u (set upstream) memastikan bahwa branch main di lokal akan selalu terhubung dengan origin/main.