

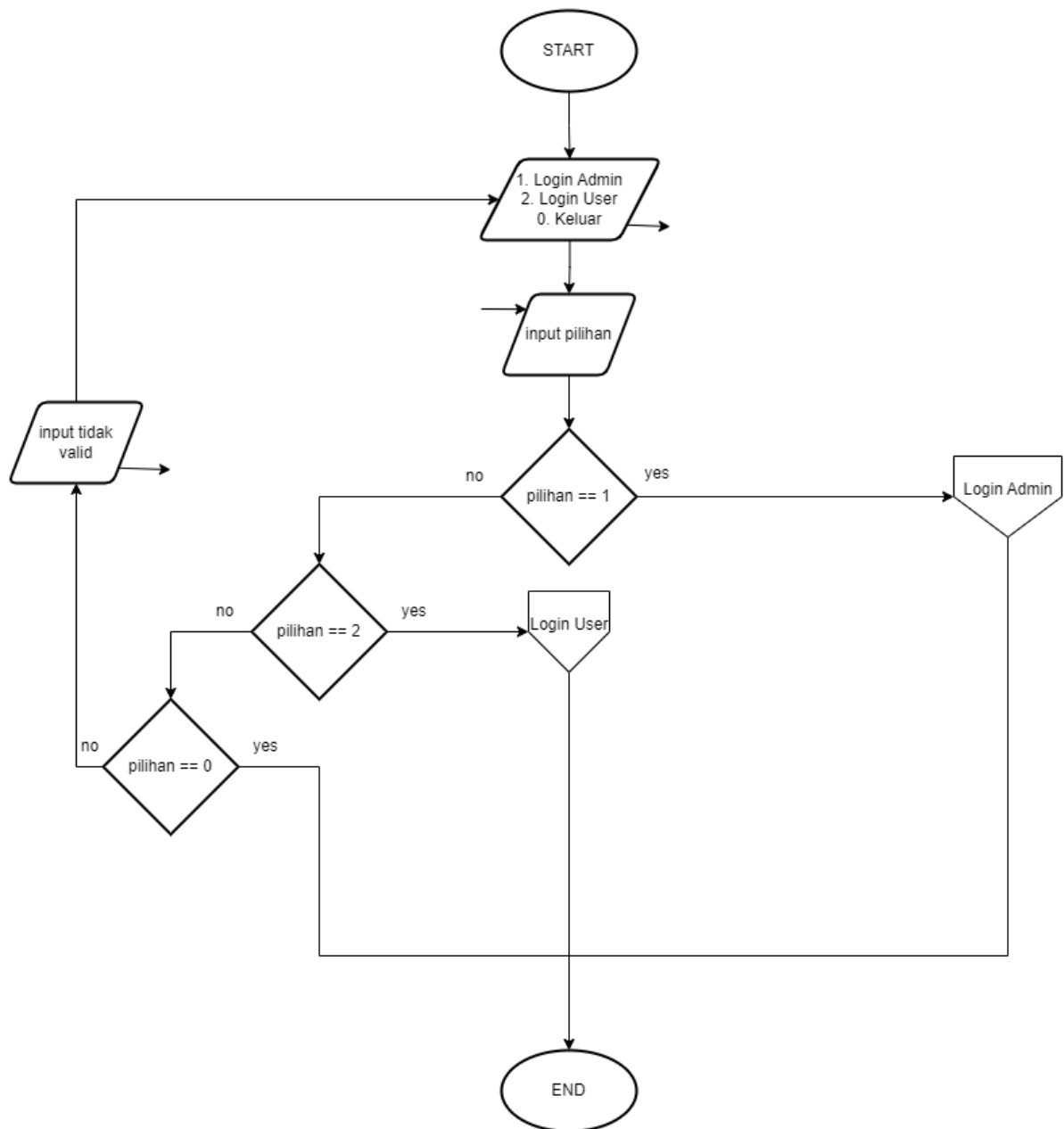
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



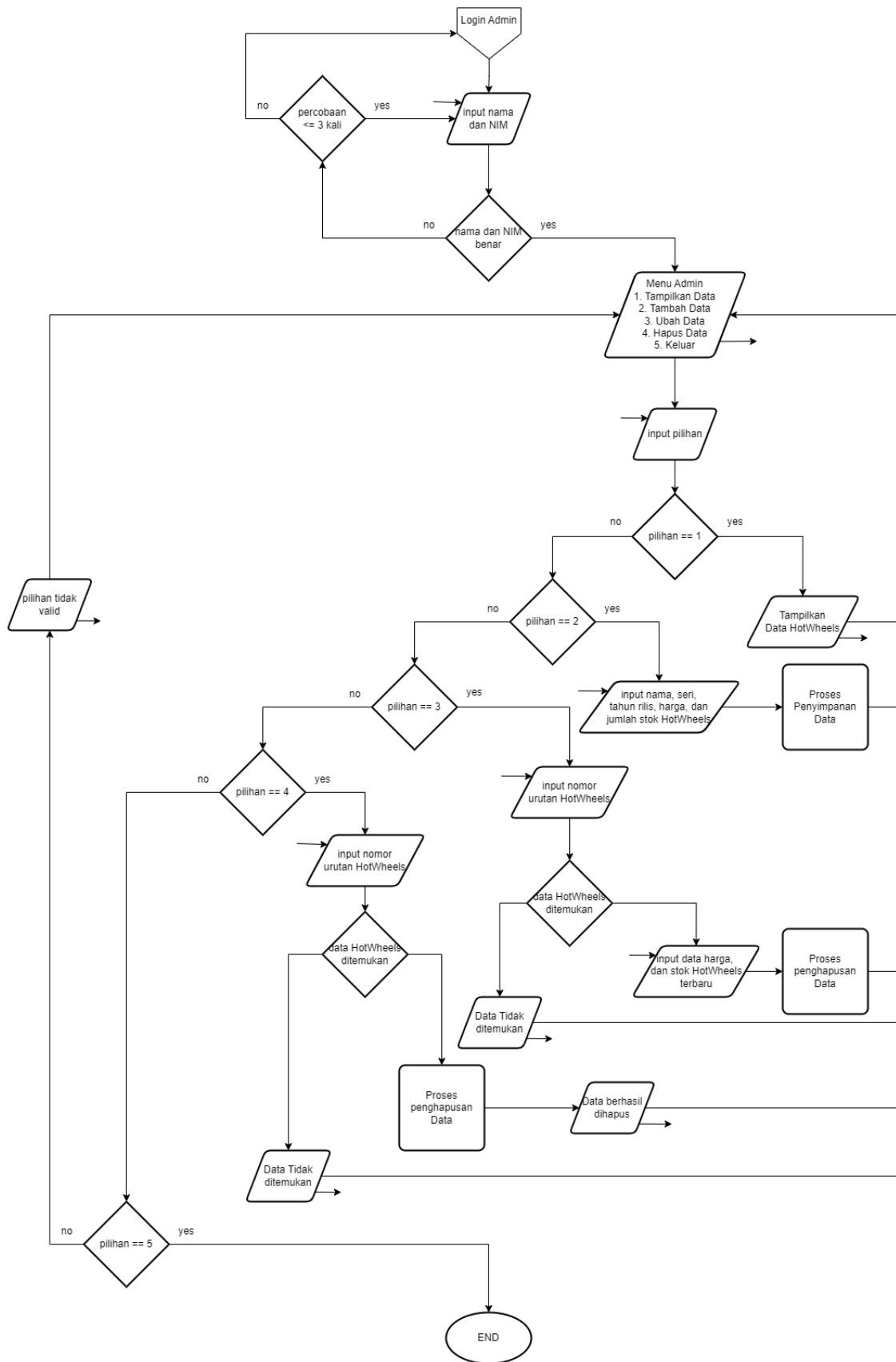
Disusun oleh:
Zulfikar Aryawinata (2409106020)
Kelas (A1'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

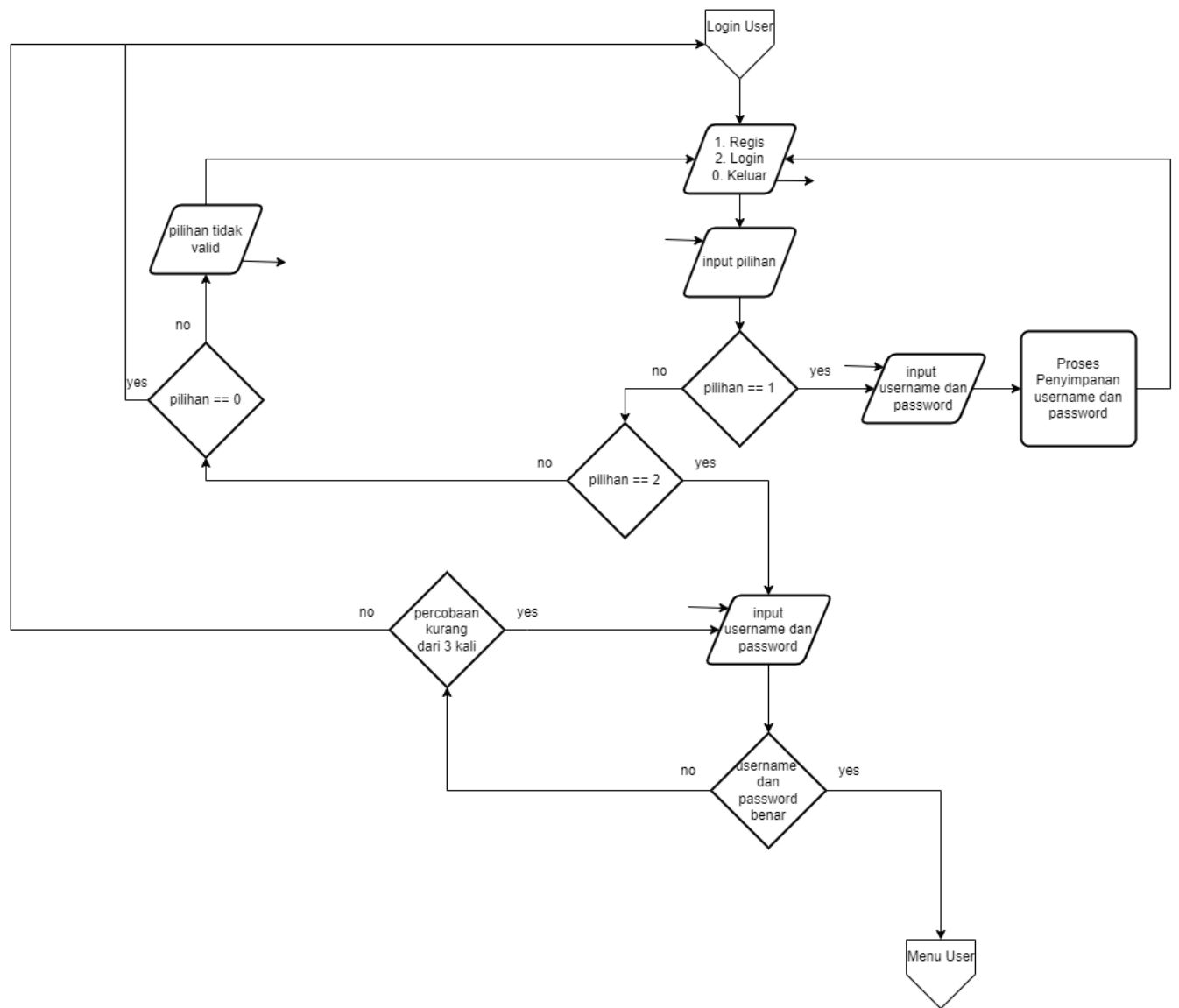
1. Flowchart



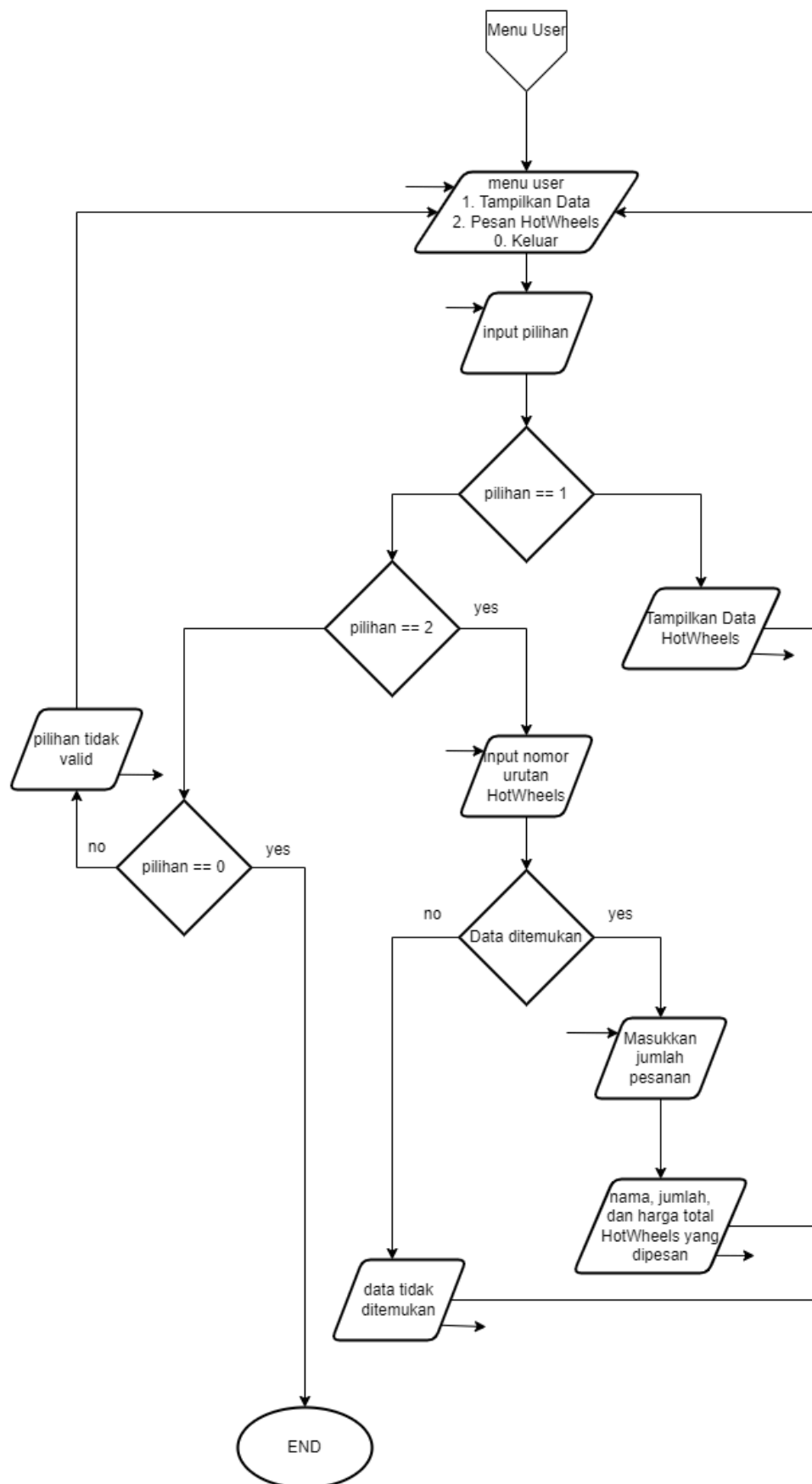
Gambar 1.1 flowchart



Gambar 1.2 flowchart



Gambar 1.3 flowchart



Gambar 1.3 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. Terdapat 2 pilihan cara Login, yang pertama sebagai admin, dan yang kedua sebagai user biasa.
2. Untuk login sebagai admin, Masukkan Nama dan NIM yang telah ditentukan.
3. 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, pengguna akan Kembali ke menu awal.
4. Di dalam menu utama, user akan diberi pilihan 0 sampai 5. Pilihan 0 adalah menu untuk menentukan urutan tampilan data HotWheels, dimana ada urutan nama HotWheels secara descending, urutan HotWheels berdasarkan dari harga tertinggi, dan urutan berdasarkan dari stok terbanyak.
5. Pilihan 1 adalah pilihan tampilkan data. Pada pilihan ini, user akan diberikan output berupa daftar HotWheels beserta keterangan lainnya seperti Seri, Tahun Rilis, dll.
6. 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.
7. Pada pilihan 3, user dapat mengubah informasi HotWheels dengan mengetikkan nama HotWheels yang sedang dijual. Informasi yang dapat diubah yaitu harga dan stok barang.
8. 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.
9. Pilihan 5 berfungsi untuk menutup program.

10. Pada menu login user terdapat 2 pilihan, yang pertama registrasi akun. User akan diminta menginput username dan password yang akan dipakai.
11. Pilihan kedua adalah login. Dimana user akan diminta menginput username dan password yang telah didaftarkan. Jika sudah benar, maka program akan berlanjut ke menu user. Jika tidak, user akan diberi 2 kali kesempatan lagi untuk mencoba. Jika masih gagal, pengguna akan Kembali ke menu login user.
12. Pada menu user. User memiliki 4 pilihan, yaitu melihat daftar HotWheels, memesan HotWheels, menentukan urutan tampilan data HotWheels, dan Menutup Program
13. Pada pilihan 2 yaitu memesan HotWheels. User akan diminta menginput nomor Hotwheels yang ada pada tabel yang telah diperlihatkan. Jika inputan tidak sesuai dengan jumlah HotWheels yang ada pada tabel, user akan diberi output “Data Tidak Ditemukan”. Jika inputan sesuai, maka user akan diperlihatkan Data HotWheels yang dipilih.
14. Selanjutnya user akan diminta menginput jumlah HotWheels yang ingin dipesan. Jika jumlah yang dipesan, akan muncul output “Jumlah Pesanan Melebihi Stok”. Jika tidak, maka user akan diperlihatkan output nama HotWheels, jumlah yang dipesan, serta total harga dari Hotwheels yang dipesan.
15. Pada pilihan ke-3, digunakan user untuk menutup program.

3. Source Code

Source-Code:

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

struct DataUser {
    string user[100] = {};
    string password[100] = {};
    string Nama[1] = {"Zulfikar Aryawinata"};
    string NIM[1] = {"2409106020"};
};
```

```

struct DataHotwheels {
    string nama_hotwheels[100] = {"Bone Shaker", "Twin Mill",
    "Deora II", "Rip Rod", "Rodger Dodger", "Night Shifter"};
    string seri[100] = {"Hot Rods", "Legends", "Surf", "Concept",
    "Muscle", "Street"};
    int tahun_rilis[100] = {2006, 1969, 2000, 2014, 1974, 2018};
    int harga[100] = {50000, 1200000, 75000, 25000, 200000,
    25000};
    int stok[100] = {12, 2, 13, 90, 5, 120};
    int jumlah = 6;
};

DataUser DU;
DataHotwheels DH;
int TipeSorting = 0;

int PartisiNama(string arr[], int low, int high, int indeks[]) {
    string pivot = arr[indeks[high]];
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (arr[indeks[j]] > pivot) { // Descending
            i++;
            swap(indeks[i], indeks[j]);
        }
    }
    swap(indeks[i + 1], indeks[high]);
    return i + 1;
}

void QuickSortNama(string arr[], int low, int high, int indeks[])
{
    if (low < high) {
        int pi = PartisiNama(arr, low, high, indeks);
        QuickSortNama(arr, low, pi - 1, indeks);
        QuickSortNama(arr, pi + 1, high, indeks);
    }
}

void BubbleSortHarga(int harga[], int indeks[], int n){ // secara ascending

```



```

    for (int i = 0; i < n - 1; i++){
        for (int j = 0; j < n - i - 1; j++){
            if (harga[j] > harga[j + 1]){
                swap(harga[j], harga[j + 1]);
                swap(indeks[j], indeks[j + 1]);
            }
        }
    }
}

void BubbleSortStok(int stok[], int indeks[], int n){ //secara descending
    for (int i = 0; i < n - 1; i++){
        for (int j = 0; j < n - i - 1; j++){
            if (stok[j] < stok[j + 1]){
                swap(stok[j], stok[j + 1]);
                swap(indeks[j], indeks[j + 1]);
            }
        }
    }
}

void tampilkanData(DataHotwheels* DH) {
    if (DH->jumlah == 0) {
        cout << "Belum ada data Hot Wheels." << endl;
        return;
    }

    int indeks[100];
    for (int i = 0; i < DH->jumlah; i++) indeks[i] = i;

    if (TipeSorting == 1) { // nama descending
        QuickSortNama(DH->nama_hotwheels, 0, DH->jumlah - 1,
indeks);
    } else if (TipeSorting == 2) { // harga ascending
        int temp[100];
        for (int i = 0; i < DH->jumlah; i++) temp[i] =
DH->harga[i];
        BubbleSortHarga(temp, indeks, DH->jumlah);
    } else if (TipeSorting == 3) { // stok descending
        int temp[100];
        for (int i = 0; i < DH->jumlah; i++) temp[i] =

```

```

DH->stok[i];
    BubbleSortStok(temp, indeks, DH->jumlah);
}

cout << "\nDaftar HotWheels:\n";
cout << left << setw(5) << "No."
    << setw(20) << "Nama"
    << setw(15) << "Seri"
    << setw(15) << "Tahun Rilis"
    << setw(10) << "Harga"
    << setw(10) << "Stok" << endl;
cout << string(70, '-') << endl;

for (int i = 0; i < DH->jumlah; i++) {
    int idx = (TipeSorting == 0) ? i : indeks[i];
    cout << left << setw(5) << i + 1
        << setw(20) << DH->nama_hotwheels[idx]
        << setw(15) << DH->seri[idx]
        << setw(15) << DH->tahun_rilis[idx]
        << setw(10) << DH->harga[idx]
        << setw(10) << DH->stok[idx] << endl;
}
}

void menuSorting(DataHotwheels* DH) {
    cout << "\n==== Menu Sorting HotWheels =====\n";
    cout << "1. Urutkan berdasarkan Nama (Z-A)\n";
    cout << "2. Urutkan berdasarkan Harga (terendah ke tertinggi)\n";
    cout << "3. Urutkan berdasarkan Stok (terbanyak ke tersedikit)\n";
    cout << "0. Kembali\n";
    cout << "Pilihan: ";
    int pilihan;
    cin >> pilihan;

    switch (pilihan) {
        case 1: TipeSorting = 1; break;
        case 2: TipeSorting = 2; break;
        case 3: TipeSorting = 3; break;
        case 0: return;
        default: cout << "(x) Pilihan tidak valid!" << endl;
    }
}

```

```

return;
}
cout << "\nData berhasil disorting!\n";
tampilkanData(DH);
}

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

void editData(DataHotwheels* DH) {
    tampilkanData(&DH);
    int urutan_hotwheels;
    cout << endl << "Masukkan Urutan HotWheels pada tabel yang
ingin diubah datanya : ";
    cin >> urutan_hotwheels;
    cout << endl;
    if (DH->jumlah >= urutan_hotwheels) {
        if (DH->jumlah == 0) {
            cout << endl;
            cout << "Belum ada data Hot Wheels." << endl;
        }
    }
}

```

```

    }
    else if (urutan_hotwheels < 1){
        cout << endl << "Nomor Tidak Valid!!!" << endl;
    }
    else {
        cout << "Data ditemukan. Masukkan data baru:" << endl;
        cout << "<=>|Masukkan Harga Baru : "; cin >>
        DH->harga[urutan_hotwheels - 1];
        cout << "<=>|Masukkan Stok Baru :"; cin >>
        DH->stok[urutan_hotwheels - 1];
        cout << endl;
        cout << "|==|Data berhasil diperbarui!";
        cout << endl;
    }
}
else{
    cout << "(X)Data Tidak Ditemukan" << endl;
    cout << endl;
}
}

void hapusData(DataHotwheels* DH) {
    tampilkanData(*&DH);
    int urutan;
    cout << endl;
    cout << "<=>|Masukkan Urutan HotWheels pada tabel yang ingin
dihapus : ";
    cin >> urutan;
    if (urutan > 0 && urutan <= DH->jumlah){
        for (int i = urutan - 1; i < DH->jumlah; i++){
            DH->nama_hotwheels[i] = DH->nama_hotwheels[i + 1];
            DH->seri[i] = DH->seri[i + 1];
            DH->tahun_rilis[i] = DH->tahun_rilis[i + 1];
            DH->harga[i] = DH->harga[i + 1];
            DH->stok[i] = DH->stok[i + 1];
            DH->jumlah--;
            cout << endl;
            cout << "|==|Data Berhasil Dihapus|==" << endl;
            break;
        }
    }
    else if (urutan < 1){

```

```

        cout << endl << "Nomor Tidak Valid!!!" << endl;
    }
    else{
        cout << endl;
        cout << "==Data Tidak Ditemukan==";
    }
}

void pesanHotwheels(DataHotwheels* DH) {
    tampilkanData(&DH);
    int pesan;
    int jumlah_pesan;
    cout << endl;
    cout << "Masukkan Urutan HotWheels yang ingin dipesan : ";
    cin >> pesan;
    if (pesan <= DH->jumlah){
        cout << endl;
        cout << "HotWheels yang ingin dipesan : " <<
DH->nama_hotwheels[pesan-1] << endl;
        cout << "Seri : " << DH->seri[pesan-1] << endl;
        cout << "Tahun Rilis : " << DH->tahun_rilis[pesan-1] <<
endl;
        cout << "Harga : " << DH->harga[pesan-1] << endl;
        cout << "Stok : " << DH->stok[pesan-1] << endl;
        cout << "Jumlah Pesanan : ";
        cin >> jumlah_pesan;
        if (jumlah_pesan <= DH->stok[pesan-1]){
            cout << endl;
            cout << "Pesanan : " << endl;
            cout << DH->nama_hotwheels[pesan-1] << endl;
            cout << jumlah_pesan << " Buah" << endl;
            cout << "Rp." << DH->harga[pesan-1]*jumlah_pesan <<
endl;
            DH->stok[pesan-1]-=jumlah_pesan;
        }
        else{
            cout << endl;
            cout << "Jumlah Pesanan Melebihi Stok.";
        }
    }
    else{
        cout << endl;
    }
}

```

```

        cout << "(x)Data Tidak Ditemukan";
        cout << endl;
    }
}

void loginAdmin(string &nama, string &nim, DataUser* DU) {
    cout << "_____ " << endl;
    cout << "|Masukkan Nama : ";
    getline(cin, nama);
    cout << "|Masukkan NIM : ";
    cin >> nim;
    cout << "===== " << endl;
}

void loginUser(string &username, string &pw, DataUser* DU) {
    cout << endl << "_____ " <<
endl;
    cout << "|Masukkan username : ";
    getline(cin, username);
    cout << "|Masukkan password : ";
    getline(cin, pw);
    cout << "===== " << endl;
}

int menuAdmin() {
    while (true) {
        cout << endl;
        cout << "_____ " <<
endl;
        cout << "|== Manajemen Penjualan Hot Wheels ==|" <<
endl;
        cout << "|0. Urutan Hotwheels                |" <<
endl;
        cout << "|1. Tampilkan Data                        |" <<
endl;
        cout << "|2. Tambah Data                          |" <<
endl;
        cout << "|3. Edit Data                            |" <<
endl;
        cout << "|4. Hapus Data                           |" <<
endl;
        cout << "===== " <<

```

```

endl;
    cout << "|5. Keluar Program" << endl;
endl;
    cout << "===== " << endl;
endl;
    cout << endl;
    cout << "Pilih menu (1-5) : ";
    int pilihan;
    cin >> pilihan;
    if (pilihan == 0) {
        menuSorting(&DH);
    }
    else if (pilihan == 1) {
        tampilkanData(&DH);
    }
    else if (pilihan == 2) {
        tambahData(&DH);
    }
    else if (pilihan == 3) {
        editData(&DH);
    }
    else if (pilihan == 4) {
        hapusData(&DH);
    }
    else if (pilihan == 5) {
        cout << endl << "=|Keluar Dari Program|=" << endl;
    }
    cout << endl;
    return 0;
}
}

int menuUser() {
    while (true) {
        cout << endl;
        cout << "===== " << endl;
endl;
        cout << "|=== Manajemen Penjualan Hot Wheels ===|" << endl;
endl;
        cout << "|1. Tampilkan Daftar HotWheels" << endl;
endl;
        cout << "|2. Pesan HotWheels" << endl;

```

```

endl;
    cout << "|2. Urutan Data HotWheels          |" <<
endl;
    cout << "===== " <<
endl;
    cout << "|0. Keluar Program          |" <<
endl;
    cout << "===== " <<
endl;

    cout << endl;
    cout << "Pilih menu (1/0) : ";
    int pilihan;
    cin >> pilihan;
    if (pilihan == 1) {
        tampilkanData(&DH);
    }
    else if (pilihan == 2) {
        pesanHotwheels(&DH);
    }
    else if (pilihan == 3) {
        menuSorting(&DH);
    }
    else if (pilihan == 0) {
        cout << endl << "Keluar Dari Program";
        return 0;
    }
}

}

int main() {
    int pilihan;
    cout << "====Menu Login====" << endl;
    cout << "|1. Login Admin      |" << endl;
    cout << "|2. Login User       |" << endl;
    cout << "===== " << endl;
    cout << "0. Close" << endl;
    cout << "Pilihan :";
    cin >> pilihan;
    cin.ignore();
    if (pilihan == 1) {
        string nama, nim;
        int percobaan = 3;

```



```

while (percobaan > 0) {
    loginAdmin(nama, nim, &DU);
    if (nim == DU.NIM[0] && nama == DU>Nama[0]) {
        cout << endl;
        cout << "===Selamat datang!==" << endl;
        menuAdmin();
        break;
    }
    else {
        percobaan--;
        cout << "!Login Gagal!" << endl; cout << "NIM atau
Nama salah!" << endl;
        cin.ignore();
    }
    if (percobaan == 0) {
        cout << endl;
        cout << "(X)Melewati Batas Percobaan. Program
Berhenti." << endl;
        return 0;
    }
}
}
else if (pilihan == 2) {
    while (true) {
        cout << "======" << endl;
        cout << "|1. Regis   |" << endl;
        cout << "|2. Login   |" << endl;
        cout << "======" << endl;
        cout << "|0. Keluar  |" << endl;
        cout << "======" << endl;
        int pilihan;
        cout << "Pilihan (1/2/0) : ";
        cin >> pilihan;
        cin.ignore();
        if (pilihan == 1) {
            int i;
            for (i = 0; i <= 100; i++){
                cout << endl <<
"
                cout << "|Masukkan username : ";
                getline(cin, DU.user[i]);
                cout << "|Masukkan password : ";

```

```

        getline(cin, DU.password[i]);
        cout <<
"===== " << endl;
        break;
    }
}
else if (pilihan == 2) {
    string username;
    string pw;
    int percobaan = 300;
    while (percobaan > 0){
        loginUser(username, pw, &DU);
        int i;
        for (i = 0; i <= 100; i++){
            if (username == DU.user[i] && pw ==
DU.password[i]){
                cout << endl;
                cout << "===Selamat datang!==" <<
endl;
                menuUSer();
                return 0;
            }
            else {
                percobaan--;
            }
            if (percobaan == 0){
                cout << endl;
                cout << "(X) Melewati batas
percobaan." << endl;
                break;
            }
        }
    }
}
else if (pilihan == 0) {
    cout << endl;
    cout << "===Keluar Dari Program===";
    return 0;
}
else {
    cout << endl << "(x)Pilihan Tidak Valid" << endl;
}

```

```

    }
}
else if (pilihan == 0){
    cout << endl;
    cout << "===Program Ditutup===";
    return 0;
}
}

```

4. Uji Coba dan Hasil Output

Menu Admin

```

=====Menu Login=====
|1. Login Admin      |
|2. Login User       |
=====
0. Close
Pilihan :1

-----
|Masukkan Nama : Zulfikar Aryawinata
|Masukkan NIM : 2409106020
|
=====

===Selamat datang!===

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

Pilih menu (1-5) : 1

Daftar HotWheels:
No.  Nama                Seri          Tahun Rilis  Harga(Rp)  Stok
-----
1    Bone Shaker          Hot Rods      2006         50000       12
2    Twin Mill             Legends      1969        1200000       2
3    Deora II              Surf         2000         75000       13
4    Rip Rod               Concept      2014         25000       90
5    Rodger Dodger         Muscle       1974        200000       5
6    Night Shifter         Street       2018         25000      120

```

Gambar 4.1

```

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

Pilih menu (1-5) : 2

<=>|Masukkan Nama HotWheels : Mustang
<=>|Masukkan Seri : Tuner
<=>|Masukkan Tahun Rilis : 2016
<=>|Masukkan Harga : 25000
<=>|Masukkan jumlah stok masuk :123

==Data Berhasil Ditambahkan!==

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

Pilih menu (1-5) : █

```

Gambar 4.2

Pilih menu (1-5) : 3

Daftar HotWheels:

No.	Nama	Seri	Tahun Rilis	Harga(Rp)	Stok
1	Bone Shaker	Hot Rods	2006	50000	12
2	Twin Mill	Legends	1969	1200000	2
3	Deora II	Surf	2000	75000	13
4	Rip Rod	Concept	2014	25000	90
5	Rodger Dodger	Muscle	1974	200000	5
6	Night Shifter	Street	2018	25000	120
7	Mustang	Tuner	2016	25000	123

Masukkan Urutan HotWheels pada tabel yang ingin diubah datanya : 7

Data ditemukan. Masukkan data baru:

<=>|Masukkan Harga Baru : 24000

<=>|Masukkan Stok Baru :100

|==|Data berhasil diperbarui!

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

Pilih menu (1-5) : █

Gambar 4.3

Pilih menu (1-5) : 4

Daftar HotWheels:

No.	Nama	Seri	Tahun Rilis	Harga(Rp)	Stok
1	Bone Shaker	Hot Rods	2006	50000	12
2	Twin Mill	Legends	1969	1200000	2
3	Deora II	Surf	2000	75000	13
4	Rip Rod	Concept	2014	25000	90
5	Rodger Dodger	Muscle	1974	200000	5
6	Night Shifter	Street	2018	25000	120
7	Mustang	Tuner	2016	24000	100

<=>|Masukkan Urutan HotWheels pada tabel yang ingin dihapus : 7

|==|Data Berhasil Dihapus|==|

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

Pilih menu (1-5) : 1

Daftar HotWheels:

No.	Nama	Seri	Tahun Rilis	Harga(Rp)	Stok
1	Bone Shaker	Hot Rods	2006	50000	12
2	Twin Mill	Legends	1969	1200000	2
3	Deora II	Surf	2000	75000	13
4	Rip Rod	Concept	2014	25000	90
5	Rodger Dodger	Muscle	1974	200000	5
6	Night Shifter	Street	2018	25000	120

Gambar 4.4

Pilih menu (1-5) : 0

===== Menu Sorting Hotwheels =====

1. Urutkan berdasarkan Nama (Z-A)
2. Urutkan berdasarkan Harga (terendah ke tertinggi)
3. Urutkan berdasarkan Stok (terbanyak ke tersedikit)
0. Kembali

Pilihan: 1

Data berhasil disorting!

Daftar Hotwheels:

No.	Nama	Seri	Tahun Rilis	Harga	Stok
1	Twin Mill	Legends	1969	1200000	2
2	Rodger Dodger	Muscle	1974	200000	5
3	Rip Rod	Concept	2014	25000	90
4	Night Shifter	Street	2018	25000	120
5	Deora II	Surf	2000	75000	13
6	Bone Shaker	Hot Rods	2006	50000	12

|=== Manajemen Penjualan Hot Wheels ===|

|0. Urutan Hotwheels|

|1. Tampilkan Data|

|2. Tambah Data|

|3. Edit Data|

|4. Hapus Data|

|5. Keluar Program|

Pilih menu (1-5) : █

Gambar 4.5

5. Langkah Langkah Git

```
PS C:\Praktikum> cd praktikum-apl
PS C:\Praktikum\praktikum-apl> git add post-test/post-test-apl-3
PS C:\Praktikum\praktikum-apl> git add .
PS C:\Praktikum\praktikum-apl> git commit -m "Finis Post Test 3"
[main 7f2b139] Finis Post Test 3
 8 files changed, 484 insertions(+)
 create mode 100644 Kelas/pertemuan-3/petemuan3.cpp
 create mode 100644 Kelas/pertemuan-3/petemuan3.exe
 create mode 100644 Kelas/pertemuan-4/pertemuan4.cpp
 create mode 100644 Kelas/pertemuan-4/pertemuan4.exe
 create mode 100644 post-test/post-test-apl-3/2409106020-ZulfikarAryawinata-PT-3.cpp
 create mode 100644 post-test/post-test-apl-3/2409106020-ZulfikarAryawinata-PT-3.exe
 create mode 100644 post-test/post-test-apl-3/tempCodeRunnerFile.cpp
PS C:\Praktikum\praktikum-apl> git push -u origin main
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 16 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (16/16), 1.16 MiB | 88.00 KiB/s, done.
Total 16 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/MamangolivE/praktikum-apl.git
   eec1d32..7f2b139  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Praktikum\praktikum-apl>
```

Gambar 5.1 langkah-langkah Git

1. git add post-test/post-test-apl-6

Menambahkan folder post-test-apl-6 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 6"

Menyimpan perubahan yang telah ditambahkan ke dalam repository lokal dengan pesan "Finish Post Test 6".

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.