

LAPORAN PRAKTIKUM ALGORITMA PEMOGRAMAN LANJUT



NAMA:Muhammad febriansyah

KELAS:B2 24 informatika

NIM:2409106090

TUGAS:laporan posttes 3

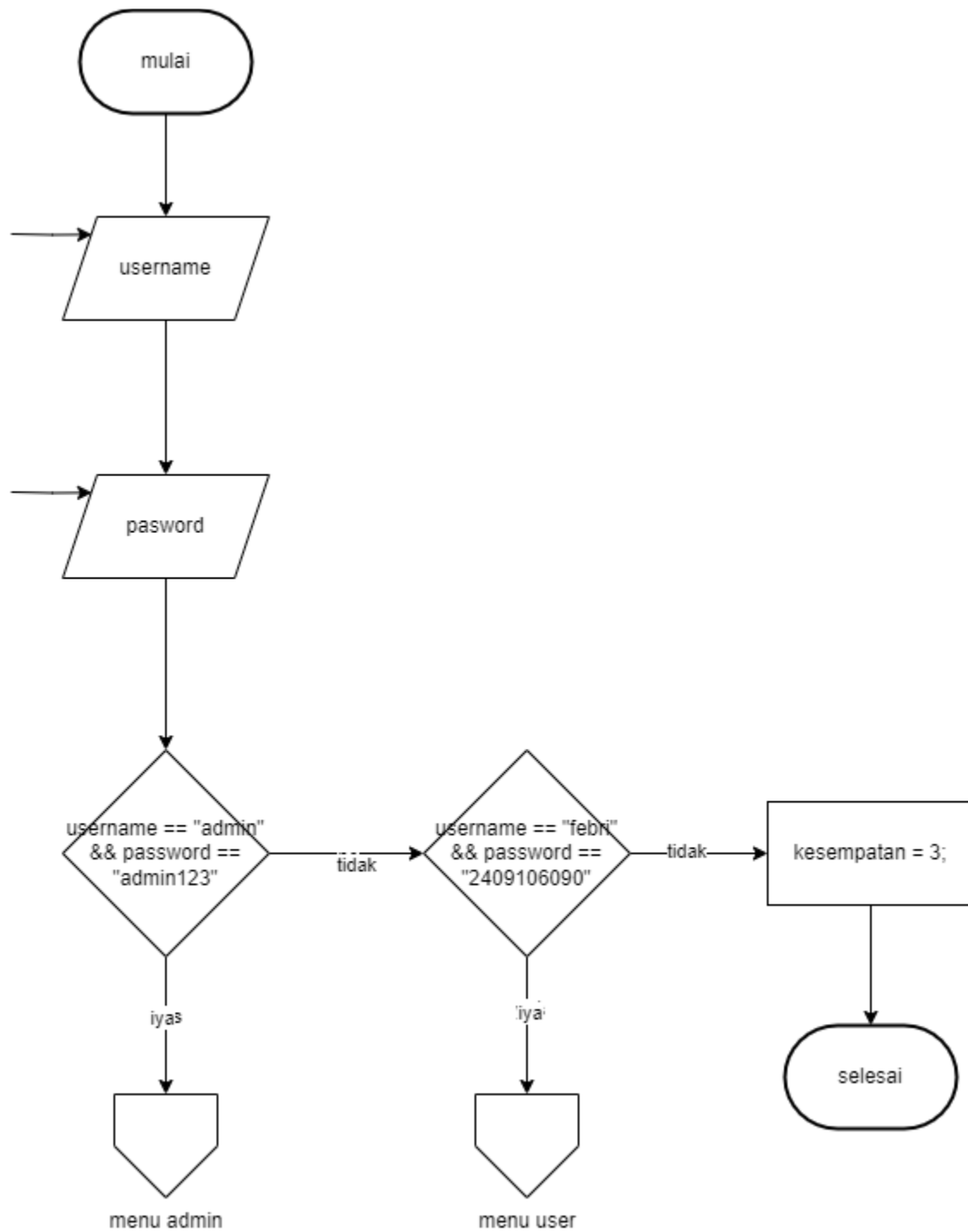
PROGRAM STUDI INFORMATIKA

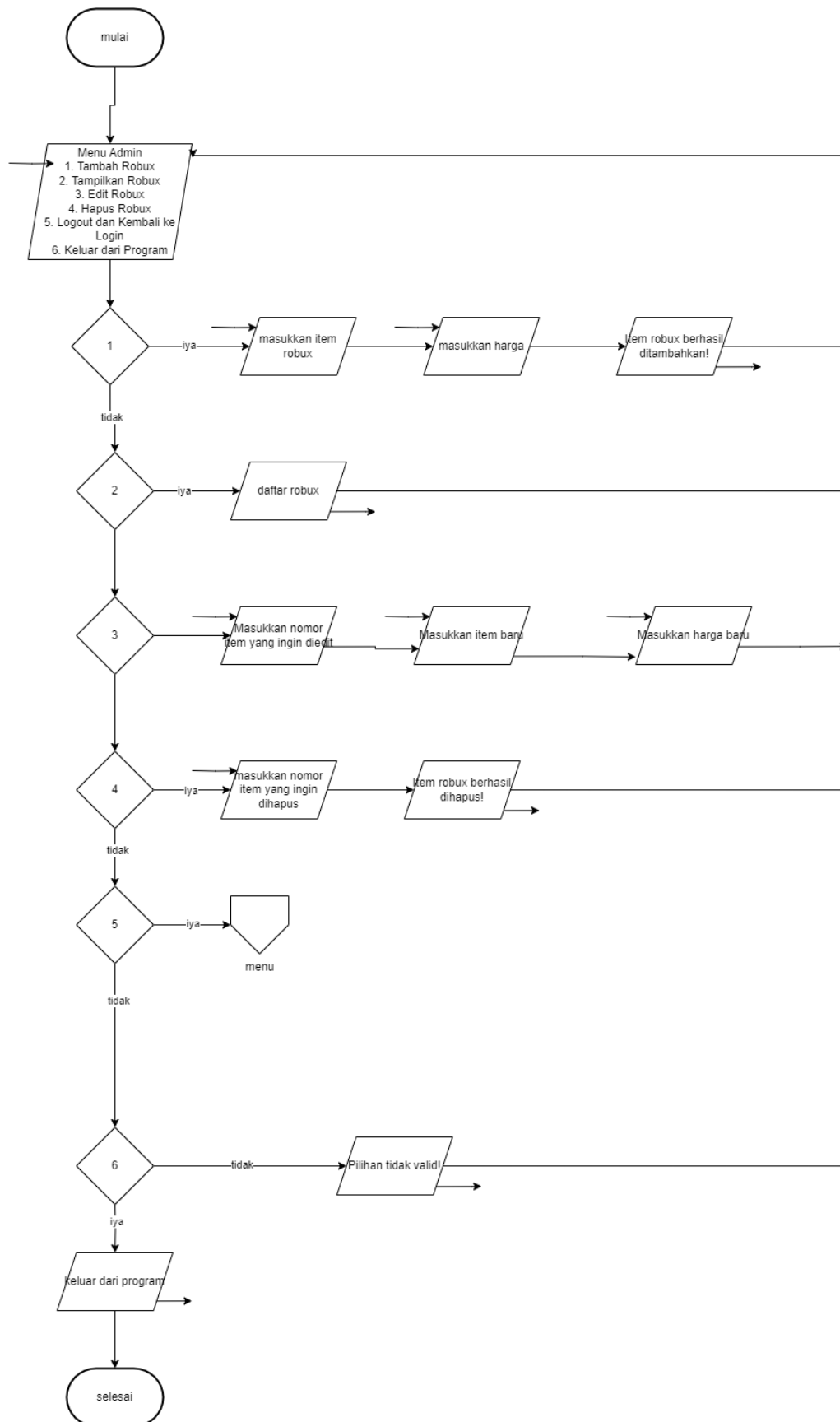
UNIVERSITAS MULAWARMAN

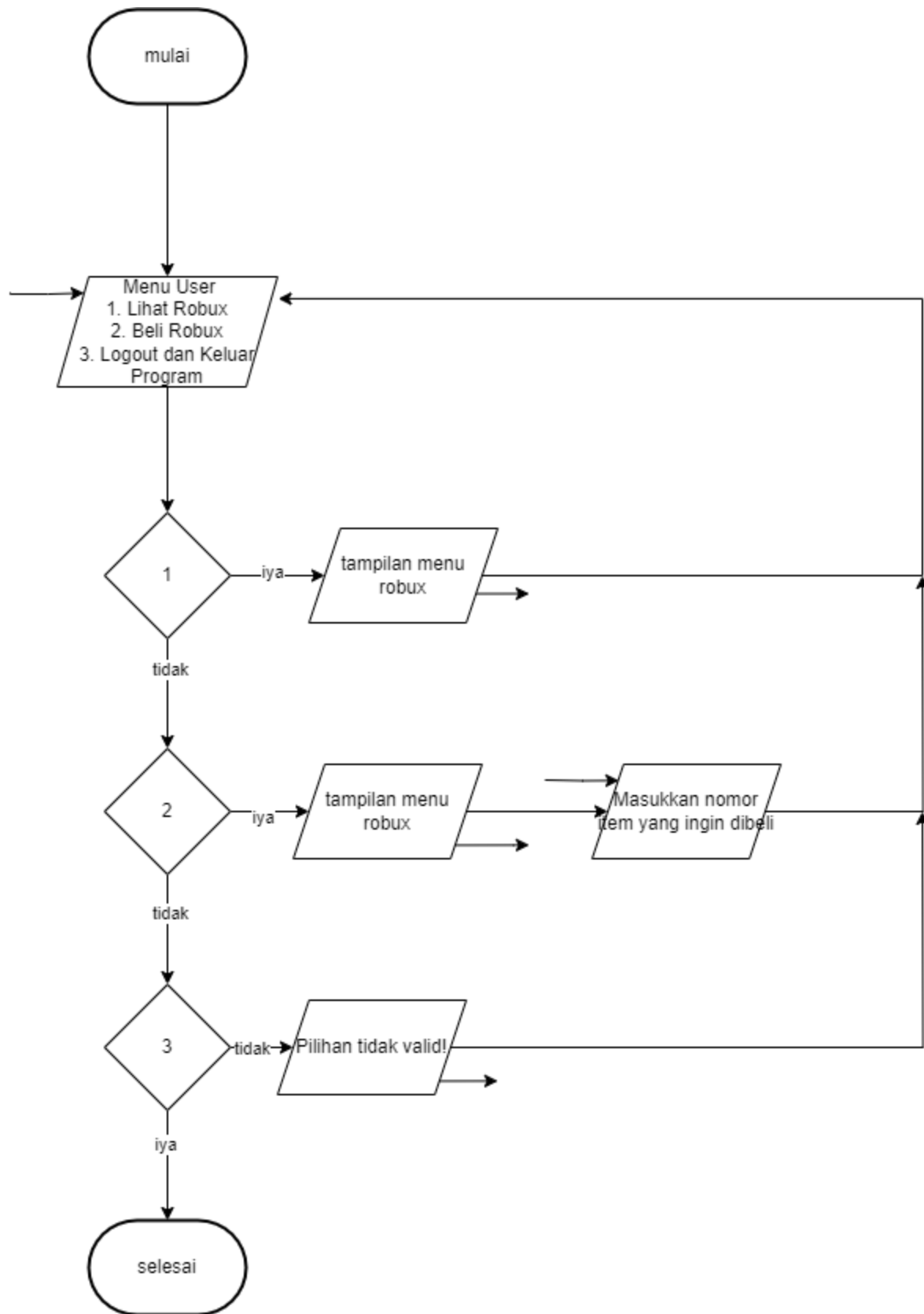
SAMARINDA

2025

1.FLOWCHART:







2.ANALISIS PROGRAM

Program ini adalah sistem manajemen penjualan Robux berbasis CLI (Command Line Interface). Program memiliki dua jenis pengguna:

Admin: Dapat menambahkan, menampilkan, mengedit, dan menghapus item Robux.

User: Hanya dapat melihat daftar item Robux dan melakukan pembelian

Sistem Login: Memastikan hanya pengguna yang memiliki kredensial yang benar dapat mengakses program.

A. Deklarasi Struktur dan Variabel

Struct RobuxItem digunakan untuk menyimpan informasi item Robux, yang terdiri dari:

nama: Nama item Robux.

harga: Harga per unit (tipe int).

stok: Jumlah stok tersedia.

Array robux[MAX_DATA] menyimpan semua item Robux dalam bentuk struct.

Variabel totalRobux menyimpan jumlah item yang tersedia.

Variabel login digunakan untuk autentikasi pengguna.

B. Sistem Login

Program meminta **username dan password**.

Jika salah, pengguna memiliki **3 kesempatan** untuk login.

Terdapat **dua jenis pengguna**:

Admin (admin, admin123): Bisa mengelola data.

User (febri, 2409106090): Hanya bisa melihat dan membeli.

Jika gagal login 3 kali, program keluar.

C. Menu Admin

1. Tambah Robux

Meminta input **nama, harga, dan stok**.

Menambah data ke dalam array robux.

Mengecek apakah array penuh sebelum menambah.

2. Tampilkan Robux

Menampilkan daftar item yang tersedia dalam format tabel.

3. Edit Robux

Memilih item berdasarkan nomor indeks.

Memungkinkan admin mengubah **nama, harga, dan stok**.

4. Hapus Robux

Memilih item berdasarkan nomor indeks.

Menggeser data untuk menghapus item.

5. Logout

Kembali ke halaman login.

6. Keluar Program

Menghentikan program.

D. Menu User

1. Lihat Robux

Sama seperti fitur admin, hanya bisa melihat daftar item.

2. Beli Robux

Memilih item berdasarkan nomor indeks.

Memasukkan jumlah yang ingin dibeli.

Jika stok mencukupi, stok akan dikurangi sesuai jumlah pembelian.

3. Logout dan Keluar

Keluar dari program.

3.source code

```
1  #include <iostream>
2  #include <iomanip>
3  #include <string>
4
5  using namespace std;
6
7  #define MAX_DATA 100
8
9  struct RobuxItem {
10     string nama;
11     int harga;
12     int stok;
13 };
14
15 RobuxItem robux[MAX_DATA];
16 int totalRobux = 0;
17
18 int main() {
19     string username, password;
20     int kesempatan;
21     bool loginSukses;
22     bool isAdmin;
23
24     while (true) {
25         kesempatan = 3;
26         loginSukses = false;
27         isAdmin = false;
28
29         while (kesempatan > 0) {
30             cout << "\nMasukkan Username: ";
31             cin >> username;
32             cout << "Masukkan Password: ";
33             cin >> password;
34
```

```

35         if (username == "admin" && password == "admin123") {
36             -----
37             isAdmin = true;
38             break;
39         } else if (username == "febri" && password == "2409106090") {
40             loginSukses = true;
41             isAdmin = false;
42             break;
43         } else {
44             kesempatan--;
45             cout << "Username atau Password salah! Kesempatan tersisa: " << kesempatan << "\n";
46         }
47     }
48
49     if (!loginSukses) {
50         cout << "Anda telah gagal login 3 kali. Program berhenti.\n";
51         return 0;
52     }
53
54     int pilihan;
55     if (isAdmin) {
56         do {
57             cout << "\n===== Menu Admin =====\n";
58             cout << "1. Tambah Robux\n";
59             cout << "2. Tampilkan Robux\n";
60             cout << "3. Edit Robux\n";
61             cout << "4. Hapus Robux\n";
62             cout << "5. Logout dan Kembali ke Login\n";
63             cout << "6. Keluar dari Program\n";
64             cout << "Pilihan Anda: ";
65             cin >> pilihan;
66

```

```

67         if (pilihan == 1) {
68             if (totalRobux >= MAX_DATA) {
69                 cout << "Data penuh! Tidak bisa menambahkan lagi.\n";
70             } else {
71                 RobuxItem robux[100] : ";
72                 cin >> robux[totalRobux].nama;
73                 cout << "Masukkan harga: ";
74                 cin >> robux[totalRobux].harga;
75                 cout << "Masukkan stok: ";
76                 cin >> robux[totalRobux].stok;
77                 totalRobux++;
78                 cout << "Item robux berhasil ditambahkan!\n";
79             }
80         } else if (pilihan == 2) {
81             if (totalRobux == 0) {
82                 cout << "\nTidak ada item robux tersedia.\n";
83             } else {
84                 cout << "\n===== \n";
85                 cout << "| No |      Item      |  Harga  |  Stok  |\n";
86                 cout << "===== \n";
87                 for (int i = 0; i < totalRobux; i++) {
88                     cout << "| " << setw(2) << i + 1 << " | "
89                         << setw(12) << robux[i].nama << " | "
90                         << setw(8) << robux[i].harga << " | "
91                         << setw(7) << robux[i].stok << " |\n";
92                 }
93                 cout << "===== \n";
94             }
95         } else if (pilihan == 3) {

```



```

98     } else {
99         int index;
00         cout << "Masukkan nomor item yang ingin diedit: ";
01         cin >> index;
02         if (index > 0 && index <= totalRobux) {
03             cout << "Masukkan item baru: ";
04             cin >> robux[index - 1].nama;
05             cout << "Masukkan harga baru: ";
06             cin >> robux[index - 1].harga;
07             cout << "Masukkan stok baru: ";
08             cin >> robux[index - 1].stok;
09             cout << "Item robux berhasil diperbarui!\n";
10         } else {
11             cout << "Nomor tidak valid!\n";
12         }
13     }
14 } else if (pilihan == 4) {
15     if (totalRobux == 0) {
16         cout << "Tidak ada item yang bisa dihapus!\n";
17     } else {
18         int index;
19         cout << "Masukkan nomor item yang ingin dihapus: ";
20         cin >> index;
21         if (index > 0 && index <= totalRobux) {
22             for (int i = index - 1; i < totalRobux - 1; i++) {
23                 robux[i] = robux[i + 1];
24             }
25             totalRobux--;
26             cout << "Item robux berhasil dihapus!\n";
27         }

```

4.GIT

1. Git add

Git add . berfungsi untuk menambahkan file file yang ada di dalam suatu folder atau repositori karena . sendiri berarti semua

```
PS C:\Users\Lenovo Gk\OneDrive\Documents\praktikum-apl> git add .
```

2.Git commit -m"bagus"

Git commit-m "bagus" berfungsi untuk mencommit pekerjaan kita sebelum di push ke github -m sendii berarti message karena saat mencommit kita harus meninggalkan pesan dan pesan yang ditinggalkan adalah "bagus":

```

PS C:\Users\Lenovo Gk\OneDrive\Documents\praktikum-apl> git commit -m "bagus"
[main 1147478] bagus
3 files changed, 736 insertions(+)
create mode 100644 post-test/post-test-apl-3/2409106090-muhammadfebriansyah-PT-3.cpp
create mode 100644 post-test/post-test-apl-3/2409106090-muhammadfebriansyah-PT-3.drawio
create mode 100644 post-test/post-test-apl-3/2409106090-muhammadfebriansyah-PT-3.exe

```

3. Git push origin main

Git push origin main berfungsi untuk meng upload file yang sudah kita add dan commit ke github

```
PS C:\Users\Lenovo Gk\OneDrive\Documents\praktikum-apl> git push origin main
```