

LAPORAN PRAKTIKUM ALGORITMA PEMOGRAMAN LANJUT



NAMA:Muhammad febriansyah

KELAS:B2 24 informatika

NIM:2409106090

TUGAS:laporan posttes 6

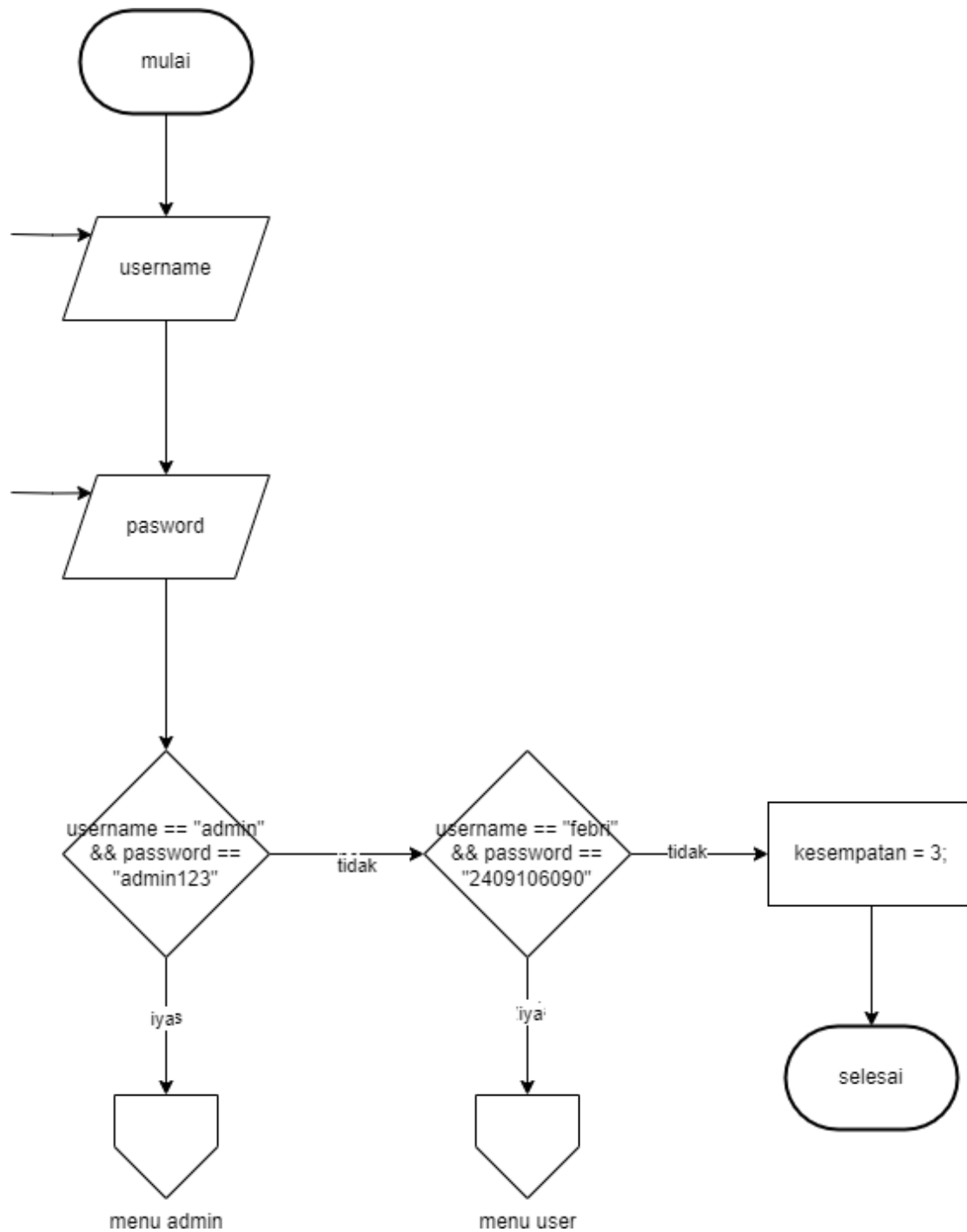
PROGRAM STUDI INFORMATIKA

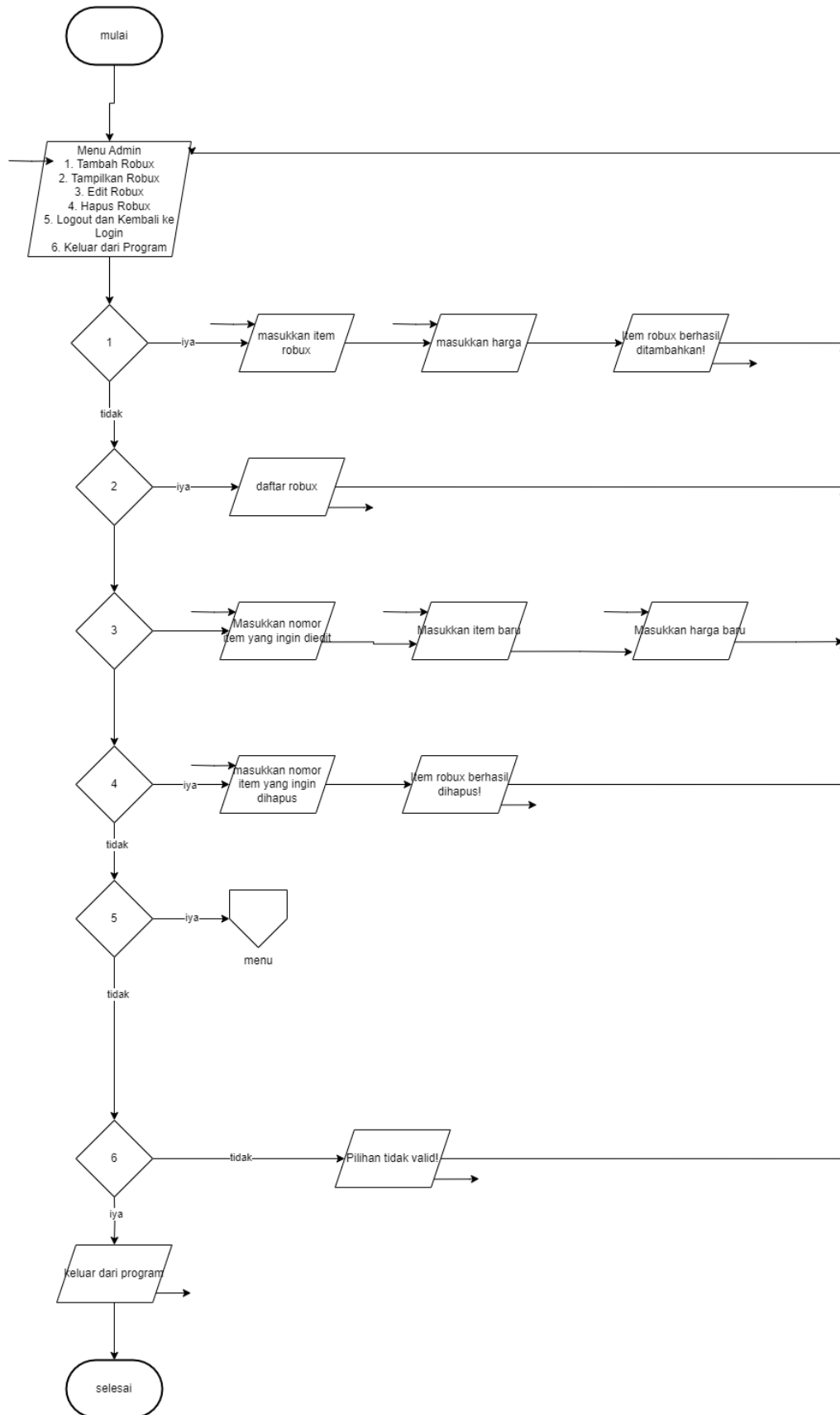
UNIVERSITAS MULAWARMAN

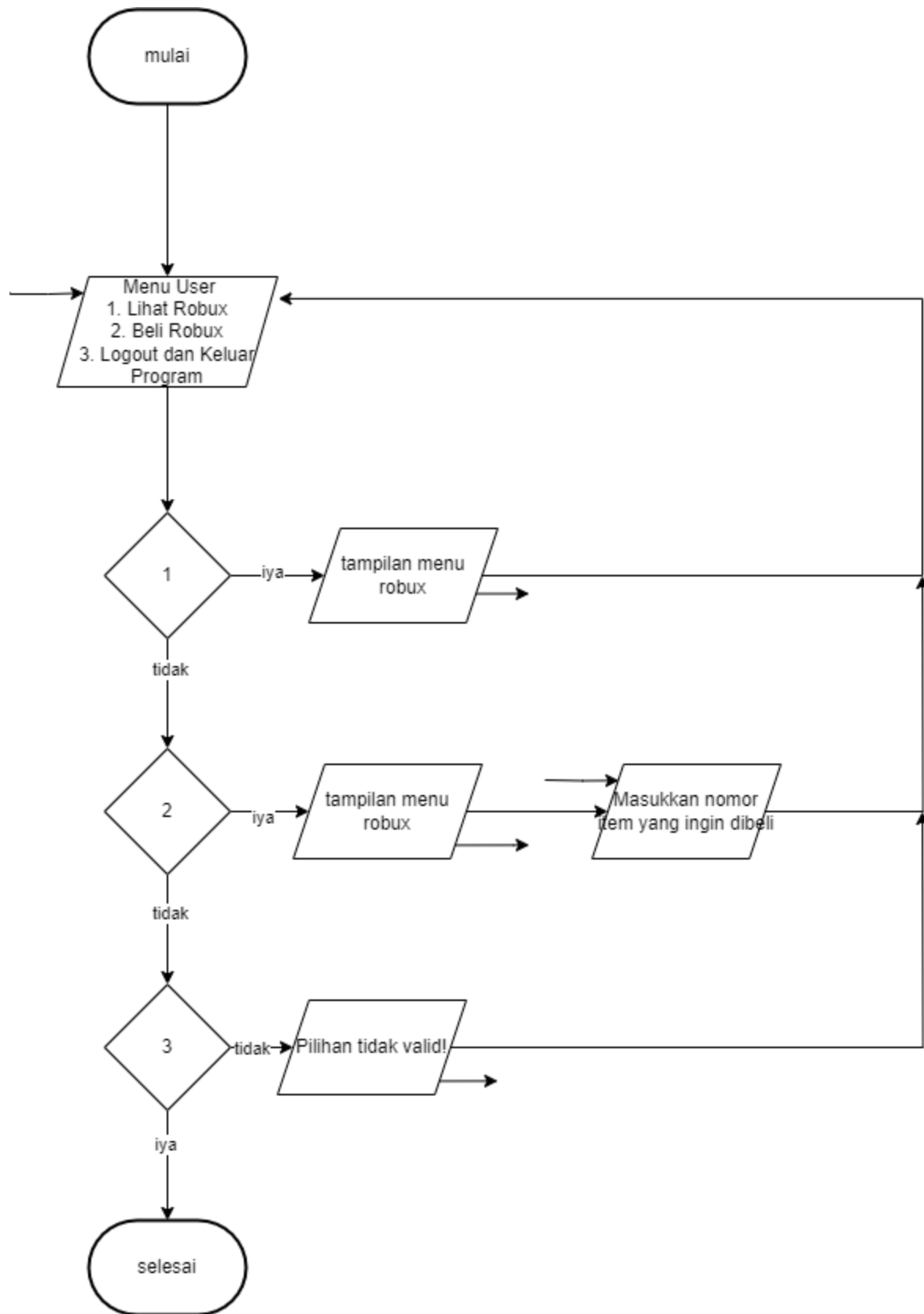
SAMARINDA

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

```
132 // ===== SORTING =====
133 void sortNamaDescending(RobuxItem data[], int total) {
134     for (int i = 0; i < total - 1; i++) {
135         for (int j = 0; j < total - i - 1; j++) {
136             if (data[j].nama < data[j + 1].nama) {
137                 swap(data[j], data[j + 1]);
138             }
139         }
140     }
141     cout << "Data berhasil diurutkan berdasarkan nama (Z-A).\n";
142 }
143
144 void sortHargaAscending(RobuxItem data[], int total) {
145     for (int i = 0; i < total - 1; i++) {
146         for (int j = 0; j < total - i - 1; j++) {
147             if (data[j].harga > data[j + 1].harga) {
148                 swap(data[j], data[j + 1]);
149             }
150         }
151     }
152     cout << "Data berhasil diurutkan berdasarkan harga (murah ke mahal).\n";
153 }
154
155 void sortStokDescending(RobuxItem data[], int total) {
156     for (int i = 0; i < total - 1; i++) {
157         for (int j = 0; j < total - i - 1; j++) {
158             if (data[j].stok < data[j + 1].stok) {
159                 swap(data[j], data[j + 1]);
160             }
161         }
162     }
163     cout << "Data berhasil diurutkan berdasarkan stok (terbanyak).\n";
```

1. Sorting Huruf (Descending) – Berdasarkan nama item Robux

- Menggunakan bubble sort
- Mengecek apakah `data[j].nama < data[j + 1].nama`, artinya jika nama sekarang lebih kecil secara alfabet, maka ditukar
- Hasil akhirnya: nama item dari Z → A

2. Sorting Angka (Ascending) – Berdasarkan harga

- Bubble sort juga digunakan
- Jika harga item sekarang lebih besar dari item berikutnya, maka ditukar
- Akhirnya: data akan terurut dari harga terkecil → terbesar

3. Metode Bebas (Ascending) – Berdasarkan stok

- Menggunakan insertion sort
- Cocok untuk data hampir terurut
- Menyisipkan setiap elemen ke posisi yang tepat dengan membandingkan jumlah stok

