

LAPORAN PRAKTIKUM
POSTTEST (2)
ALGORITMA PEMROGRAMAN LANJUT

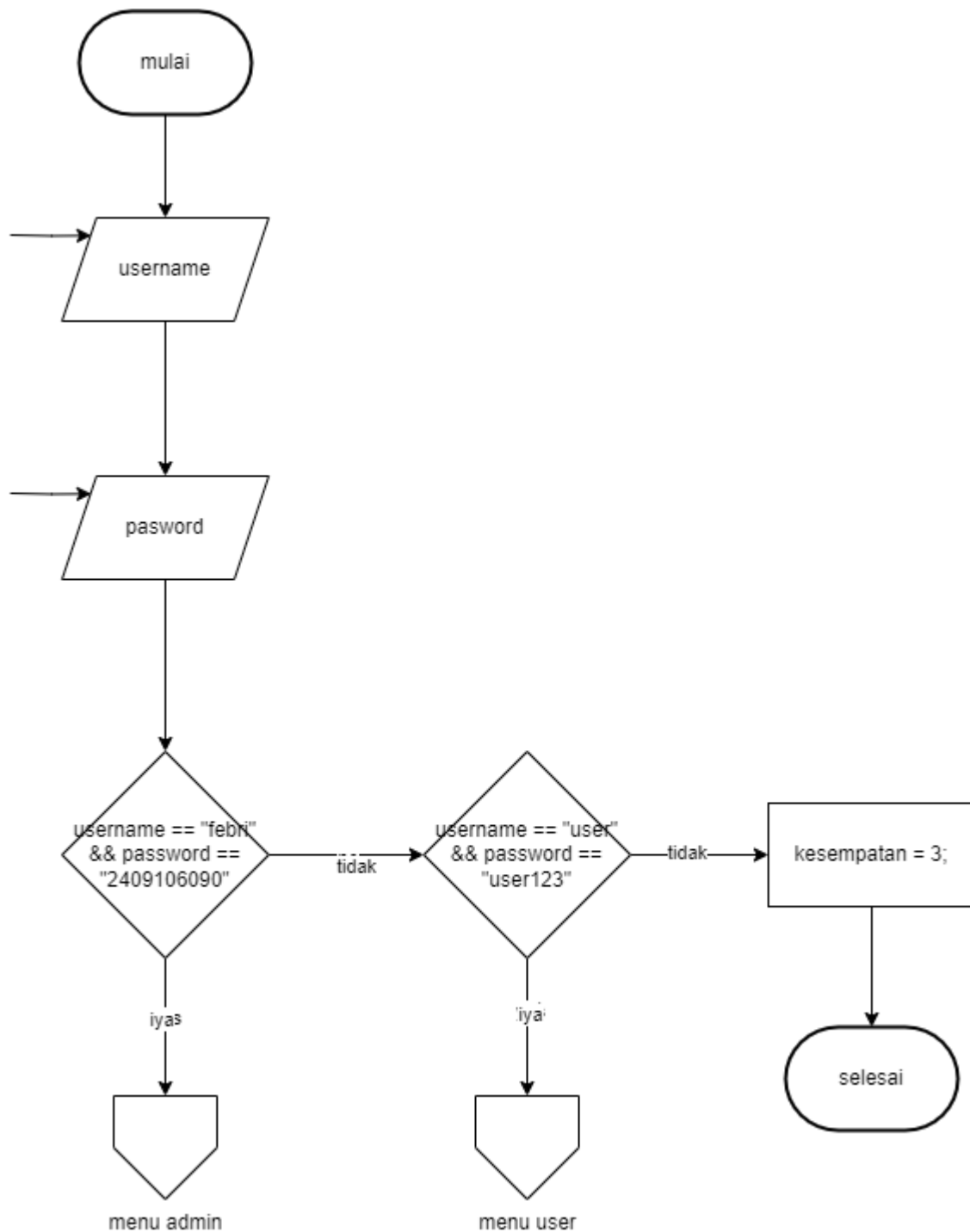


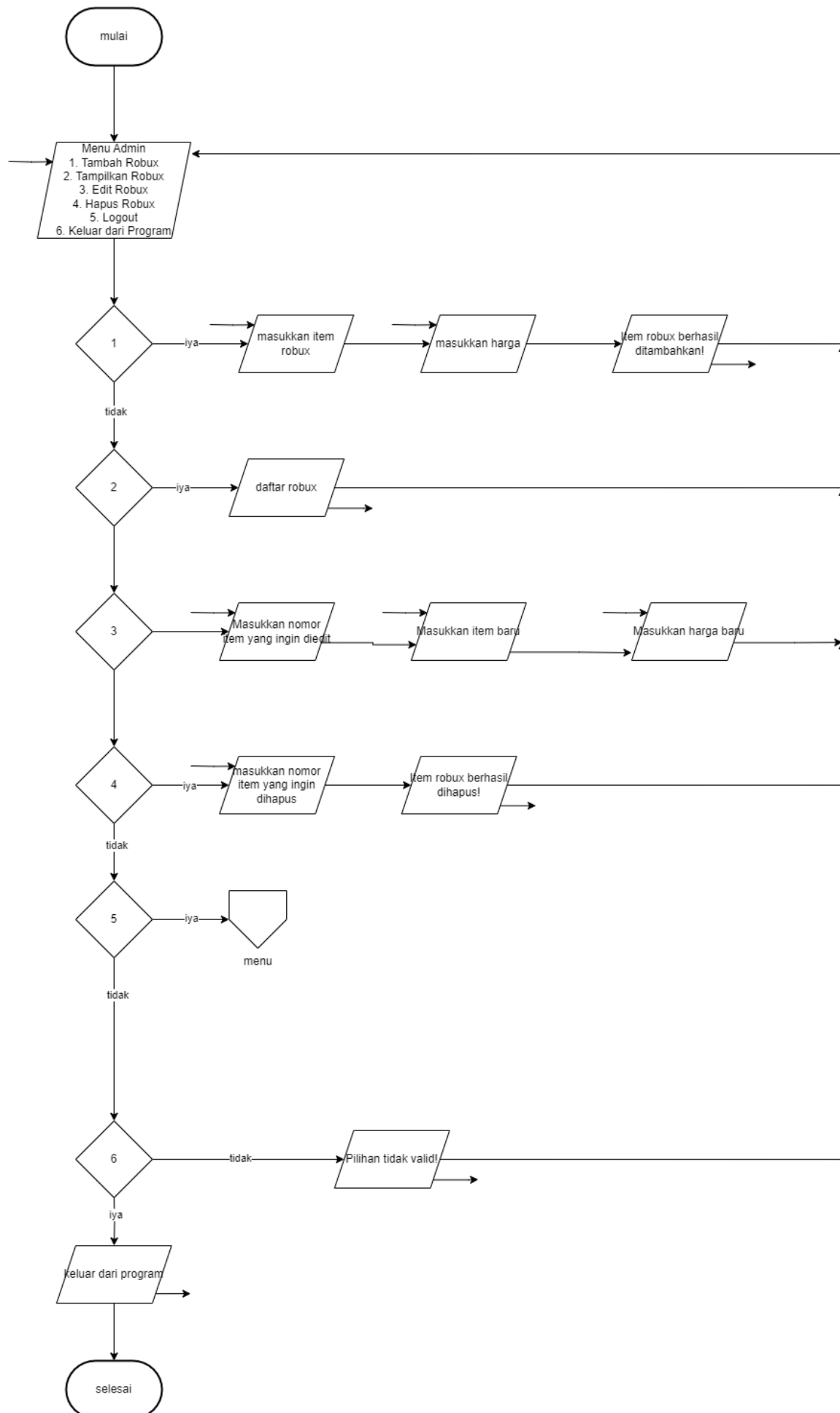
Disusun oleh:
Muhammad febriansyah
NIM: (2409106090)
Kelas (B2 24)

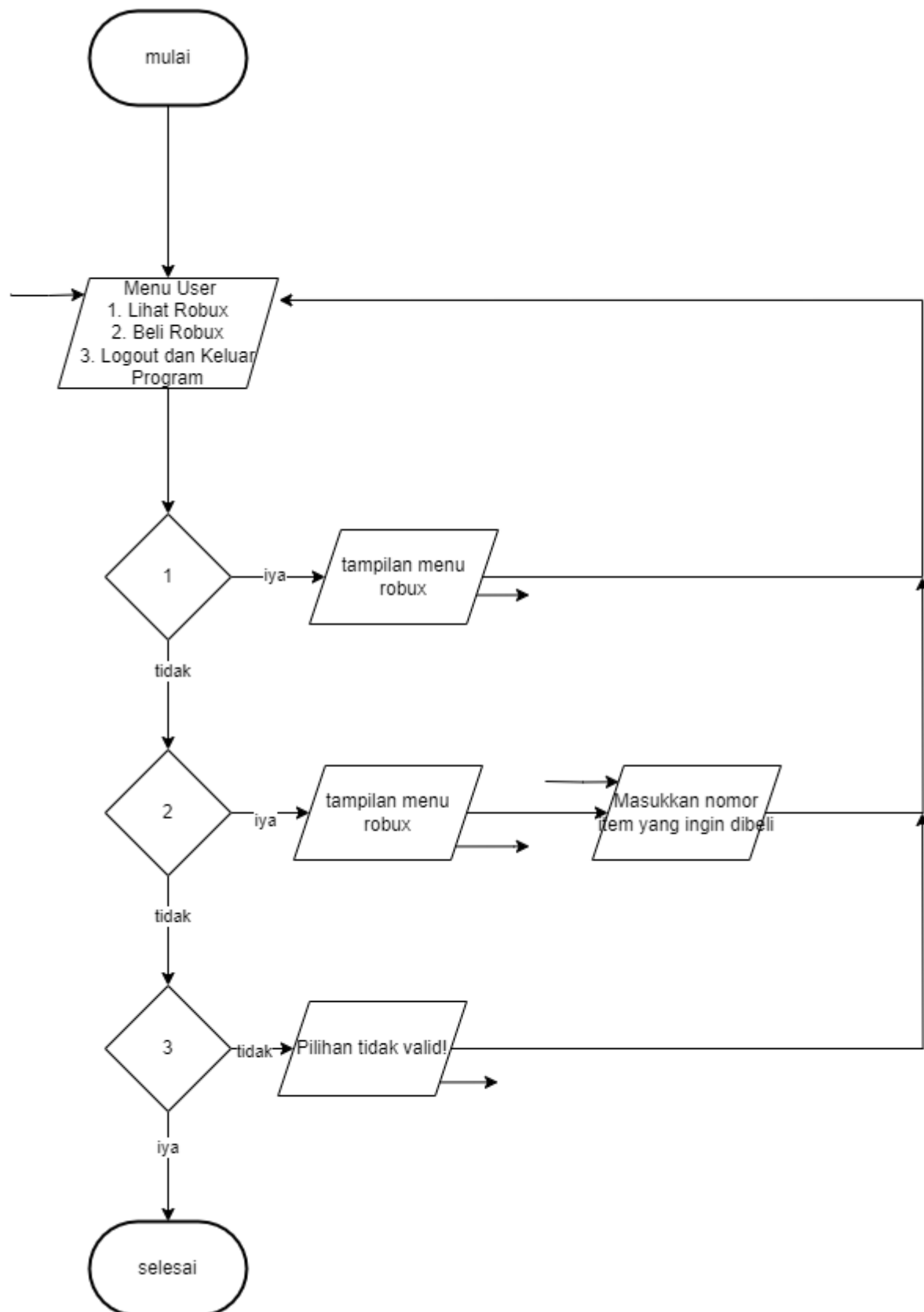
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

MENU UTAMA







2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini merupakan sistem penjualan robux yang memungkinkan user untuk melakukan login serta mengakses robux yang telah di jual oleh admin. Program menyediakan sistem CRUD seperti:

1. Tambah Robux
2. Tampilkan Robux
3. Edit Robux
4. Hapus Robux

Program juga memiliki fitur login dengan batas maksimum tiga kali percobaan sebelum akses diblokir. Jika login berhasil, pengguna atau admin dapat memilih menu yang sudah di sediakan atau keluar dari program, dan jika gagal pengguna tidak akan bisa login.

2.2 Penjelasan Alur & Algoritma

1. penjelasan alur program:

1. Memulai Program

Program dijalankan dan menampilkan halaman login.

User harus memasukkan username dan password.

Jika salah memasukkan login 3 kali, program akan berhenti otomatis.

2. Login sebagai Admin

Gunakan **Username: admin** dan **Password: admin123**.

Setelah berhasil login, akan muncul Menu Admin:

1. **Tambah Robux** Admin dapat menambahkan item Robux.
2. **Tampilkan Robux** Menampilkan daftar item Robux.
3. **Edit Robux** Mengedit data item Robux.
4. **Hapus Robux** Menghapus item Robux.
5. **Logout** Kembali ke halaman login.
6. **Keluar dari Program** Menutup program.

3. Login sebagai User

Gunakan **Username: user** dan **Password: user123**.

Setelah berhasil login, akan muncul **Menu User**:

1. **Lihat Robux** User bisa melihat daftar item Robux.
2. **Beli Robux** User bisa memilih item untuk dibeli.
3. **Logout dan Keluar** User keluar dari program.
4. **Logout dan Keluar Program**
 - Jika **Admin** memilih "Logout", maka akan kembali ke halaman login.
 - Jika **Admin** memilih "Keluar dari Program", maka program akan berhenti.
 - Jika **User** memilih "Logout dan Keluar", maka program juga akan berhenti.

2. penjelasan algoritma:

1. Login System

Mulai
Inisialisasi kesempatan = 3
Ulangi selama kesempatan > 0:
 Minta input username dan password
 Jika username dan password benar:
 Jika admin Masuk ke Menu Admin
 Jika user Masuk ke Menu User
 Jika salah:
 Kurangi kesempatan
 Tampilkan jumlah kesempatan tersisa
Jika kesempatan habis, hentikan program

2. Admin Menu CRUD

Tampilkan Menu Admin:

1. Tambah Robux
 - Minta input nama item dan harga
 - Simpan ke dalam array
 - Tambah total data
2. Tampilkan Robux
 - Cek apakah ada data robux
 - Jika ada, tampilkan dalam bentuk tabel
3. Edit Robux
 - Minta nomor item yang akan diedit
 - Minta input nama dan harga baru
 - Simpan perubahan
4. Hapus Robux
 - Minta nomor item yang akan dihapus

- Geser data dalam array untuk menghapus item
- Kurangi total data
- 5. Logout Kembali ke login
- 6. Keluar Program Hentikan program

3. User Menu (Melihat dan Membeli Robux)

Tampilkan Menu User:

1. Lihat Robux
 - Cek apakah ada data robux
 - Jika ada, tampilkan dalam bentuk tabel
2. Beli Robux
 - Minta nomor item yang ingin dibeli
 - Tampilkan konfirmasi pembelian
3. Logout dan Keluar → Hentikan program

3. Source Code

A.FITUR NAMBAH ROBUX

SOURCE CODE:

```

32  int tambahRobux() {
33      if (totalRobux >= MAX_DATA) {
34          cout << "Data penuh! Tidak bisa menambahkan lagi.\n";
35          return 0;
36      }
37      cout << "\nMasukkan item robux: ";
38      cin >> robux[totalRobux][0];
39      cout << "Masukkan harga: ";
40      cin >> robux[totalRobux][1];
41      totalRobux++;
42      cout << "Item robux berhasil ditambahkan!\n";
43      return 1;
44  }

```

B.FITUR HAPUS ROBUX

SOURCE CODE:

```

47 int hapusRobux() {
48     if (totalRobux == 0) {
49         cout << "Tidak ada item yang bisa dihapus!\n";
50         return 0;
51     }
52     int index;
53     tampilkanRobux();
54     cout << "Masukkan nomor item yang ingin dihapus: ";
55     cin >> index;
56     if (index > 0 && index <= totalRobux) {
57         for (int i = index - 1; i < totalRobux - 1; i++) {
58             robux[i][0] = robux[i + 1][0];
59             robux[i][1] = robux[i + 1][1];
60         }
61         totalRobux--;
62         cout << "Item robux berhasil dihapus!\n";
63         return 1;
64     } else {
65         cout << "Nomor tidak valid!\n";
66         return 0;
67     }
68 }

```

C.SISTEM TAMPILKAN ROBUX

SOURCE CODE:

```

14 v int tampilkanRobux() {
15 v     if (totalRobux == 0) {
16         cout << "\nTidak ada item robux tersedia.\n";
17         return 0;
18     }
19     cout << "\n===== \n";
20     cout << "| No |      Item      |      Harga      |";
21     cout << "\n===== \n";
22 v     for (int i = 0; i < totalRobux; i++) {
23         cout << "| " << setw(2) << i + 1 << " | "
24             << setw(12) << robux[i][0] << " | "
25             << setw(12) << robux[i][1] << " | \n";
26     }
27     cout << "===== \n";
28     return 1;
29 }

```

D.SISTEM EDIT ROBUX

SOURCE CODE:


```

71  int editRobux() {
72      if (totalRobux == 0) {
73          cout << "Tidak ada item yang bisa diedit!\n";
74          return 0;
75      }
76      int index;
77      tampilkanRobux();
78      cout << "Masukkan nomor item yang ingin diedit: ";
79      cin >> index;
80      if (index > 0 && index <= totalRobux) {
81          cout << "Masukkan item baru: ";
82          cin >> robux[index - 1][0];
83          cout << "Masukkan harga baru: ";
84          cin >> robux[index - 1][1];
85          cout << "Item robux berhasil diperbarui!\n";
86          return 1;
87      } else {
88          cout << "Nomor tidak valid!\n";
89          return 0;
90      }
91  }

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Login berhasil sebagai admin
2. Login berhasil sebagai user
3. Login gagal

4.2 Hasil Output

1. login berhasil sebagai admin

```

Masukkan Username: febri
Masukkan Password: 2409106090

===== Menu Admin =====
1. Tambah Robux
2. Tampilkan Robux
3. Edit Robux
4. Hapus Robux
5. Logout
6. Keluar dari Program
Pilihan Anda: 

```

2.login berhasil sebagai user

```
Masukkan Username: user
Masukkan Password: user123

===== Menu User =====
1. Lihat Robux
2. Beli Robux
3. Logout dan Keluar Program
Pilihan Anda: █
```

3.login gagal

```
Masukkan Username: 1
Masukkan Password: 1
Username atau Password salah! Kesempatan tersisa: 2

Masukkan Username: 1
Masukkan Password: 1
Username atau Password salah! Kesempatan tersisa: 1

Masukkan Username: 1
Masukkan Password: 1
Username atau Password salah! Kesempatan tersisa: 0
Anda telah gagal login 3 kali. Program berhenti.
```

5.GIT

5.1 GIT ADD

Git add digunakan untuk merubah status file, file yang belum di add maka akan memiliki status Untracked. dan akan berubah menjadi staged jika sudah di add ke repository tetapi belum di-commit.

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$ git add .
```

5.2GIT COMMIT

Git commit berfungsi untuk menyimpan perubahan file di repository.

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$ git commit -m "nguawur"
[main aeb2d0f] nguawur
2 files changed, 189 insertions(+)
create mode 100644 praktikum-apl/post-test/post-test-2/2409106081_BenyaminHaqie_PT_2.cpp
create mode 100644 praktikum-apl/post-test/post-test-2/2409106081_BenyaminHaqie_PT_2.exe
```

5.3GIT PUSH

git push -u origin main digunakan untuk mengupload perubahan yang ada pada repository lokal ke repository github pada branch main.

```
Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$ git push origin main
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 20 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 678.42 KiB | 3.85 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/haqi2/praktikum-apl.git
   551eb04..aeb2d0f  main -> main

Lenovo@lyev MINGW64 /c/APL/project-git/praktikum-apl/post-test/post-test-2 (main)
$
```