

Governance and Economic Policy: Administration and Trade



Disusun oleh Kelompok AshenVale:

Ahmad Mujahid (2509106059)

Yoga Pramudya Ananta(2509106049)

Aryasatya Rakha Phanyputra(2509106066)

Kelas (B1'25)

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA**

1. Konsep

Program simulasi manajemen kota berbasis teks. Konsep ini menggabungkan dua peran utama, Penguasa (Admin) dan Pedagang (User) di dalam satu sistem ekonomi sederhana. Dunia mengalami event acak yang memengaruhi fluktuasi harga barang dan kondisi ekonomi.

Pembeli Otomatis, Next Day Otomatis(Setiap 60 detik), Chatbot(opsional)
Waktu Real Time.

2. Role

1) *Penguasa/Admin* :

- Mengawasi perkembangan ekonomi seluruh kota.
- menentukan harga dasar barang di pasar.
- Memberikan bantuan modal(pinjaman) bagi masyarakat.
- Menentukan aturan pajak dan suku bunga pinjaman.

2) *Pedagang / User* :

- Membuka toko.
- Membeli barang.
- Menentukan harga jual barang sendiri (strategi pemain).
- Menjual barang.
- Mengelola modal usaha.
- Membayar pajak berdasarkan keuntungan.
- Dapat mengajukan pinjaman ke penguasa jika mengalami kekurangan modal.
- Berjuang agar tokonya tidak bangkrut.

3. Alur Pedagang / User

1) *Registrasi dan Login*

- Pedagang membuat akun
- Mendapat modal awal.
- Membuat nama toko.

2) *Kelola Toko*

- Beli barang dagang di pasar.
- Menjual Barang
- Lihat Stock
- Ubah harga barang
- Tarik barang dari penjualan
- Mengajukan pinjaman.

3) *Melihat Laporan*

- Melihat Laporan Harian : Sistem menampilkan ringkasan laporan: Event hari kemarin, Harga bahan, Jumlah pembeli , Barang terjual, Pendapatan, Pajak dan dana akhir
- Laporan Pembelian : Pembelian Pada Hari itu
- Laporan Pinjaman : Melihat Persetujuan Pinjaman

4) *Kebangkrutan*

- Jika dana user ≤ 0 , toko otomatis ditandai sebagai “bangkrut”.
- User masih bisa mengajukan pinjaman ke penguasa.

4. Alur Dewa / Admin

1) . *Login*

- Program memverifikasi identitas admin (username dan password sederhana).

2) *Manajemen Ekonomi Kota (CRUD)*

- (Membuat Kebijakan Kota) Create :
 - menetapkan pajak,
 - pinjaman,
 - menetapkan harga barang dasar yang berubah setiap hari tergantung event.
- Read :
 - Informasi toko : nama, kekayaan, status aktif/bangkrut
- Update : Mengubah tarif pajak, bunga pinjaman.
- Delete : Menghapus toko (misal toko tutup/bangkrut).

3) *Sistem Pajak & Pinjaman*

- menerima pajak otomatis dari hasil keuntungan pedagang setiap hari.
- Menetapkan tarif pajak (% dari keuntungan toko).
- Memberikan pinjaman kepada toko tertentu dengan bunga tertentu.
- Memantau status pinjaman: aktif, lunas, atau gagal bayar.
- Menetapkan kebijakan ekonomi berbasis waktu (misalnya, penghapusan pajak sementara).

`5. Fitur

1) *Next Day Feature*

- Setiap hari event berubah, harga bahan fluktuatif, dan pembeli otomatis datang, dan stock barang di pasar berubah.

2) *Event*

- Event memengaruhi harga barang di pasar, bukan harga jual(pemain menentukan harga jual barangnya sendiri).

3) *Sistem Pajak & Pinjaman*

- Admin dapat menentukan pajak, bunga, dan jumlah pinjaman untuk setiap toko.

4) *Manajemen Modal & Keuntungan*

- Setiap toko memiliki modal, pemasukan, dan pengeluaran.

5) *Kebangkrutan*

- Setiap toko dapat mengalami bangkrut. Sebagai sanksi admin dapat melarang user membuka toko dalam waktu yang telah ditentukan.

`6. Library

1) Os

- Membersihkan layar antar-menu.

2) Time

- Efek delay/animasi teks agar pertarungan terlihat lebih dinamis.

3) Random

- Menentukan fluktuasi pasar dengan memilih event secara random.
- Membuat BOT pembeli secara otomatis dan acak

4) Inquirer

- Membuat menu interaktif berbasis teks di terminal

30 detik per hari, 3,5 menit ragu”

Presentasi 10 menit

Pedagang 10 Beli bahan bahan/ barang → 2000 gold/10 barang(biaya)

Pedagang jual lagi barang tersebut → 300 gold/barang

Terjual → 5 barang → pendapatan 1500

Keuntungan = pendapatan - biaya

Keuntungan = 1500 – 1000 → 500

500 gold → 5%/minggu → otomatis berkurang 5 gold →pada minggu itu.

Pinjaman →5000 →

Bunga → 2%/minggu → mengurangi kekayaan toko

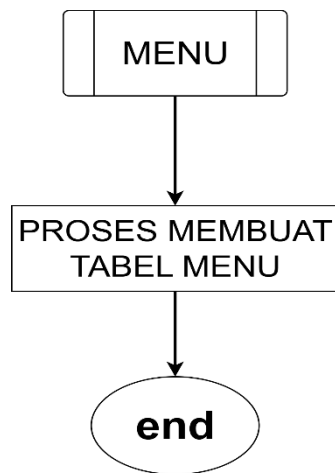
6. Flowchart

```
graph TD; Start([start]) --> Join(( )); 1((1)) --> Join; Join --> MENU[MENU]; MENU --> ANSWER[/ANSWER/]; ANSWER --> D1{ANSWER = 1}; D1 -- True --> LOGIN1[LOGIN]; LOGIN1 --> ADMIN[ADMIN]; ADMIN --> Join2(( )); D1 -- False --> D2{ANSWER = 2}; D2 -- True --> LOGIN2[LOGIN]; LOGIN2 --> USER[USER]; D2 -- False --> Join2; Join2 --> 2((1)); 2 --> End([end]);
```

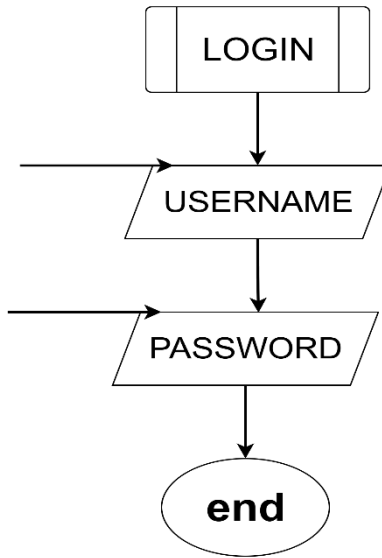
The flowchart illustrates the main system flow. It begins with a 'start' terminal, which leads to a join node. A connector labeled '1' also leads to this join node. From the join node, the flow proceeds to a 'MENU' process, then to an 'ANSWER' input/output symbol. This leads to a decision diamond 'ANSWER = 1'. If the answer is 'True', the flow goes to a 'LOGIN' process, then to an 'ADMIN' process, and finally to a second join node. If the answer is 'False', the flow goes to another decision diamond 'ANSWER = 2'. If this answer is 'True', the flow goes to a 'LOGIN' process, then to a 'USER' process, and finally to the second join node. If the answer is 'False', the flow also goes to the second join node. From the second join node, the flow proceeds to a connector labeled '1', which then leads to an 'end' terminal.

6.1 Flowchart Utama

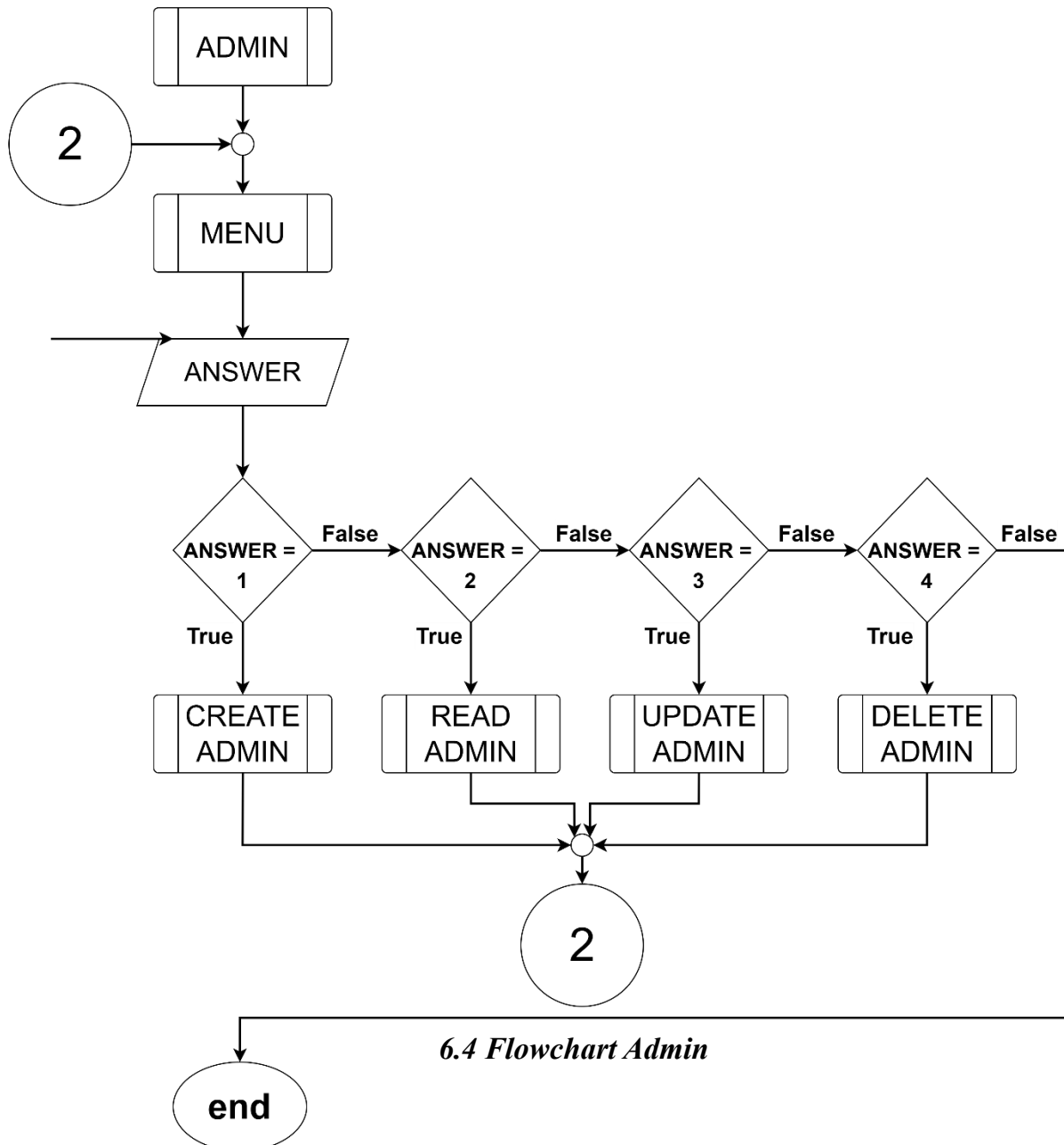
6.1 Flowchart Utama



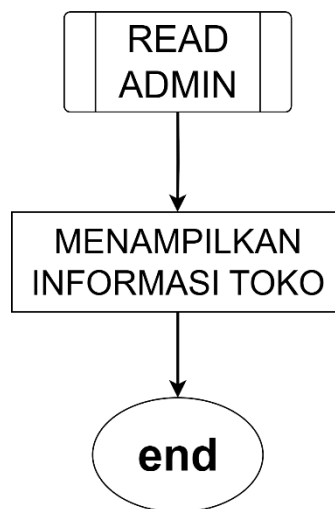
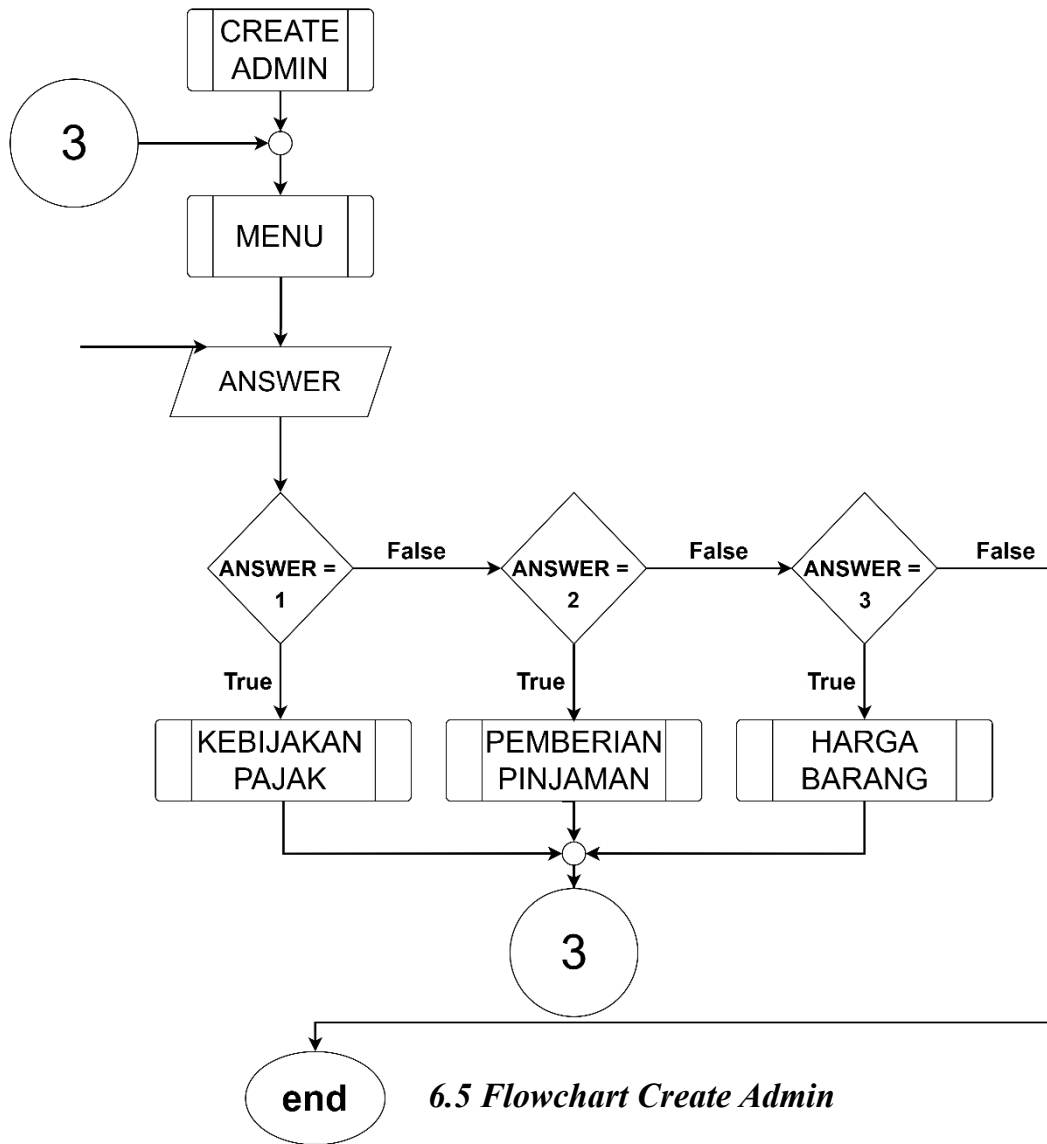
6.2 Flowchart fungsi menu

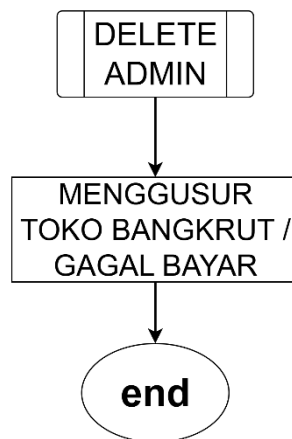
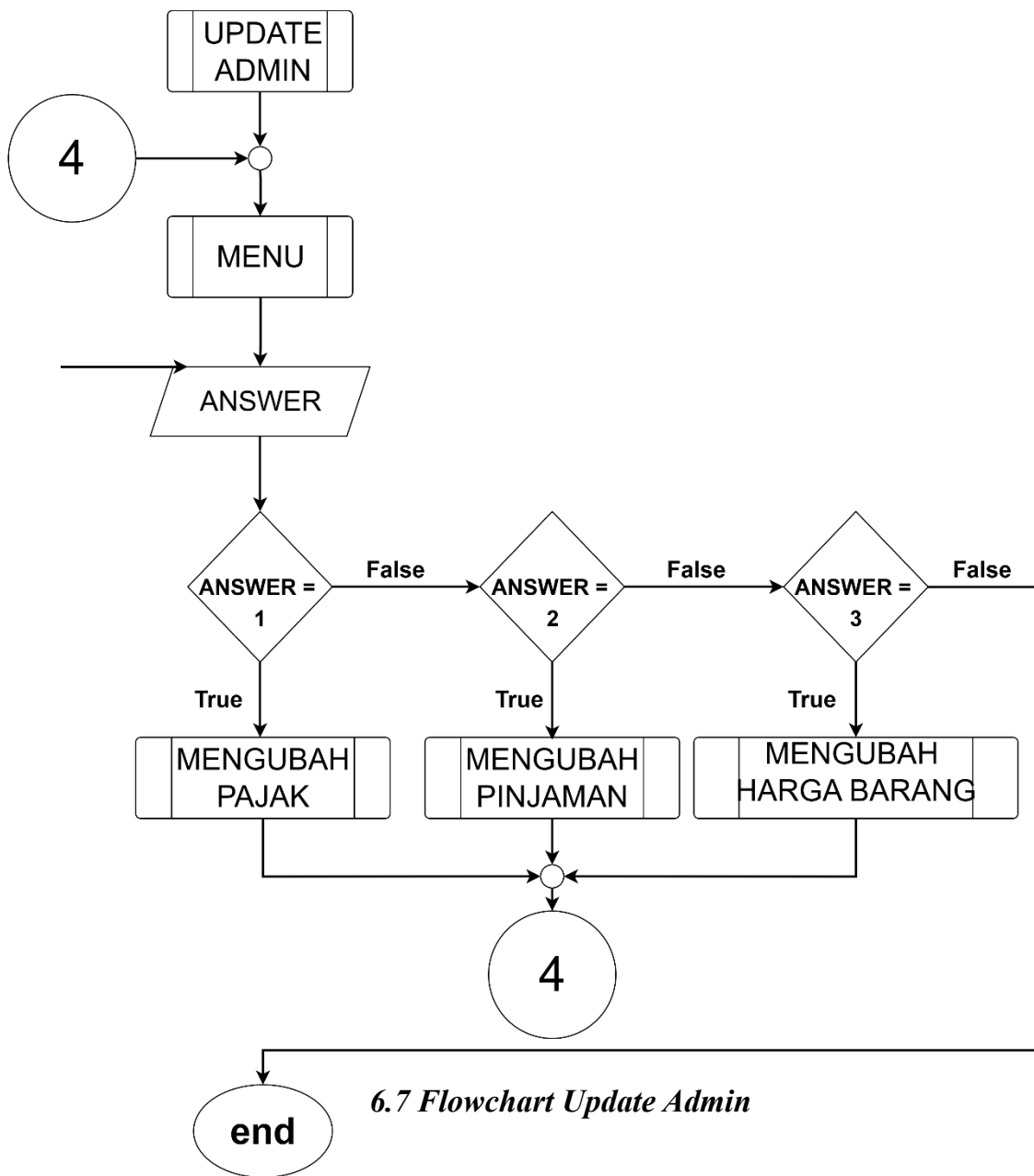


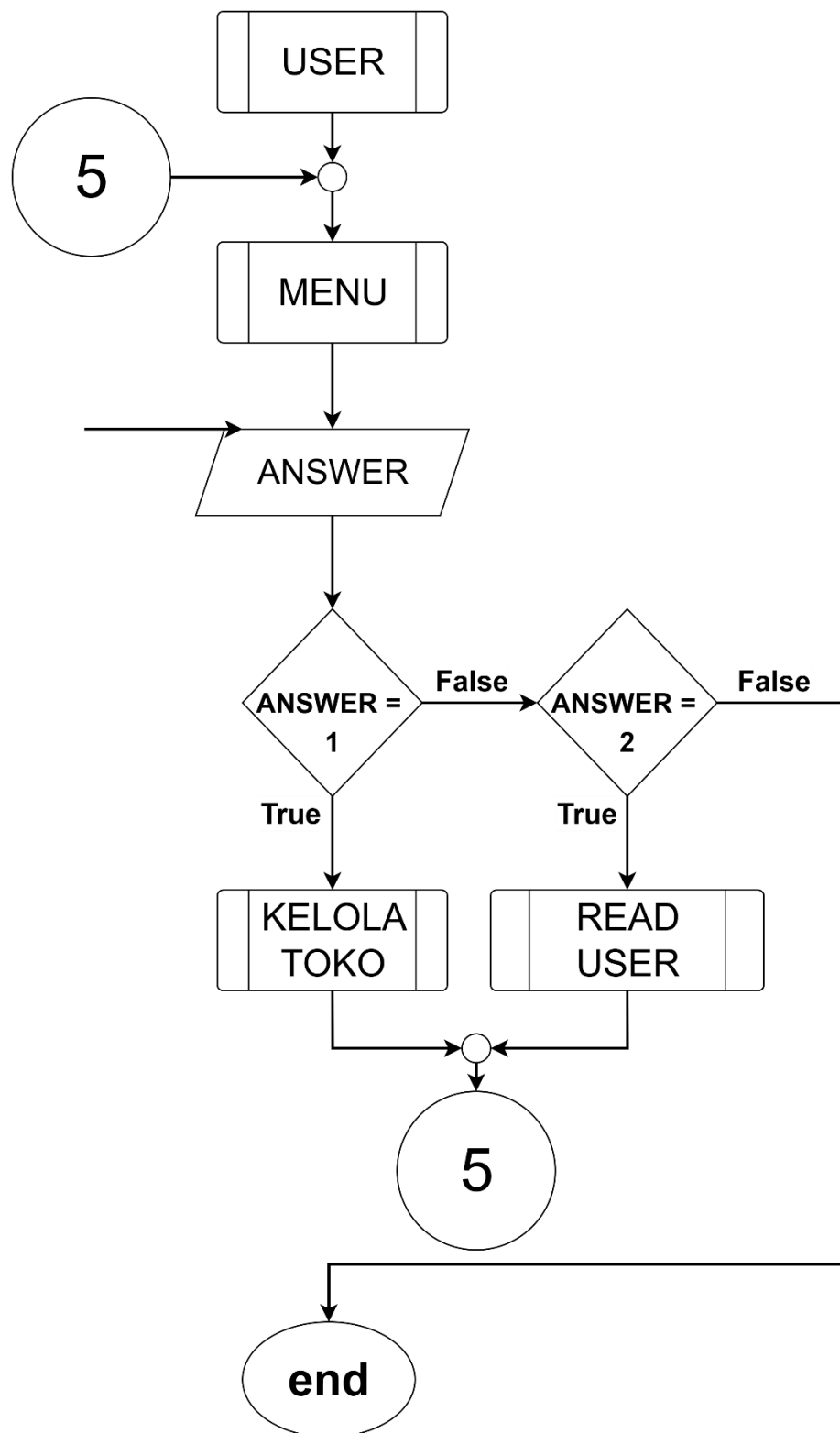
6.3 Flowchart fungsi login



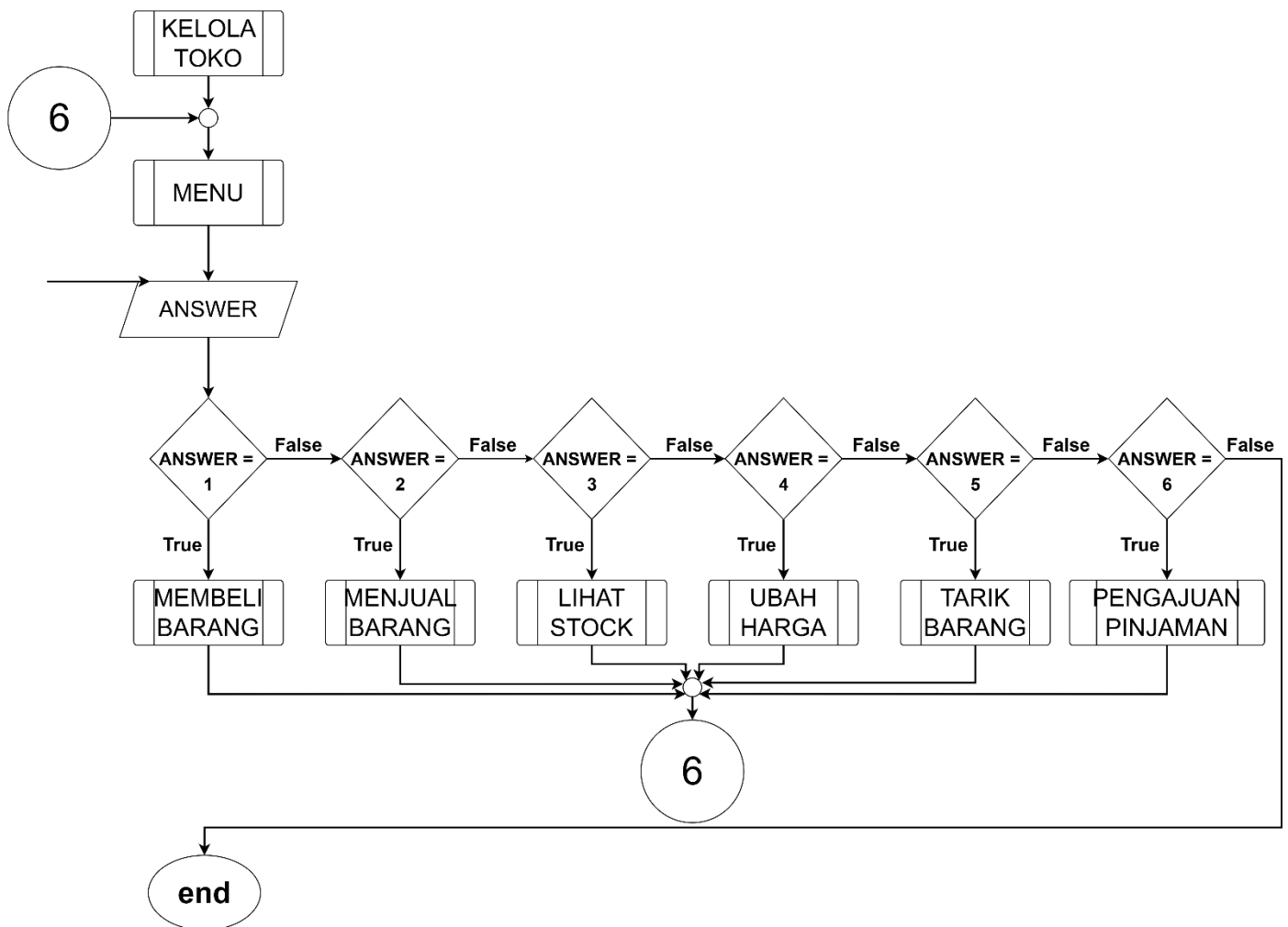
6.4 Flowchart Admin



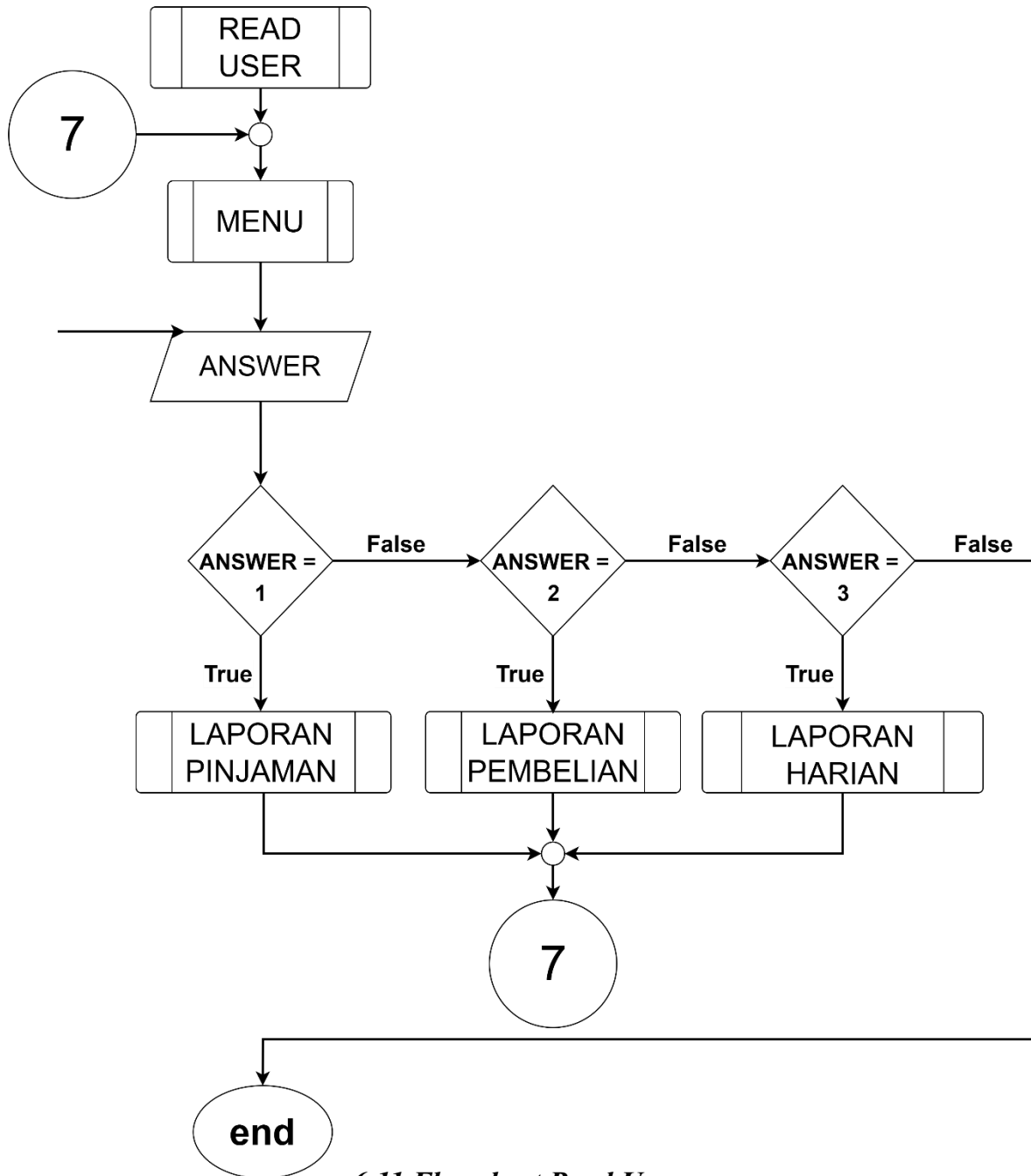




6.9 Flowchart User



6.10 Flowchart Kelola Toko User



6.11 Flowchart Read User