

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN
DASAR**



MANAJEMEN INVENTARIS TOKO BANGUNAN

Oleh:

Kelompok 2 A'24

NABILAH ALFA RAHMAH	2409106004
ANDI FACHRY ALAM T.	2409106006
WIDYA AYU ANGGRAINI	2409106011
NILUH FINCY GLORIA N. M.	2409106012

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan laporan dengan judul "Manajemen Inventaris Toko Bangunan". Laporan ini disusun sebagai salah satu bentuk pengembangan pengetahuan dan keterampilan dalam bidang pengelolaan data berbasis teknologi, sekaligus untuk memenuhi tugas mata kuliah terkait pemrograman dan sistem informasi.

Laporan ini berisi penjelasan mengenai perancangan dan implementasi program CRUD (Create, Read, Update, Delete) yang dirancang untuk mempermudah pengelolaan inventaris toko bangunan. Program ini memungkinkan pengguna untuk menambah, melihat, memperbarui, dan menghapus data barang secara efisien, dengan fitur pencatatan stok secara real-time yang membantu pengambilan keputusan terkait manajemen stok.

Selama proses penyusunan laporan dan pembuatan program, kami menghadapi berbagai kendala, seperti keterbatasan waktu, kendala teknis, dan kurangnya pengalaman dalam mengembangkan sistem yang terstruktur. Namun, berkat bimbingan dari Pak Awang dan Pak Bambang selaku dosen pengajar, asisten praktikum, serta dukungan dari rekan-rekan, kendala tersebut dapat diatasi. Untuk itu, kami mengucapkan terima kasih kepada semua pihak yang telah memberikan kontribusi, masukan, dan dukungan selama proses ini berlangsung.

Kami menyadari bahwa laporan ini masih memiliki kekurangan, baik dari segi isi maupun implementasi teknis. Oleh karena itu, kami terbuka terhadap kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi pembaca, khususnya dalam pengembangan sistem berbasis teknologi untuk manajemen inventaris.

Samarinda, 22 November 2024

Kelompok 2

TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut:

Bahasa Asing	Bahasa Indonesia	Penjelasan
<i>Function</i>	Fungsi	Bagian kode yang dapat dipanggil untuk menjalankan tugas tertentu.
<i>Variable</i>	Variabel	Tempat untuk menyimpan data dalam program.
<i>Loop</i>	Perulangan	Struktur untuk mengulang suatu blok kode.
<i>Condition</i>	Kondisi	Pernyataan untuk memutuskan alur program berdasarkan syarat tertentu.
<i>List</i>	Daftar	Tipe data untuk menyimpan kumpulan nilai secara berurutan.
<i>Dictionary</i>	Kamus	Tipe data untuk menyimpan pasangan key-value.
<i>String</i>	Teks	Tipe data untuk merepresentasikan teks atau karakter.
<i>Integer</i>	Bilangan bulat	Tipe data untuk angka tanpa pecahan.
<i>Boolean</i>	Logika	Tipe data dengan nilai benar (True) atau salah (False).
<i>File</i>	Berkas	Tempat untuk menyimpan data atau informasi dalam program.
<i>Output</i>	Keluaran	Data yang dihasilkan oleh program.
<i>Exception</i>	Pengecualian	Kesalahan atau kondisi tak terduga yang terjadi selama eksekusi.
<i>Debugging</i>	Penelusuran kesalahan	Proses menemukan dan memperbaiki kesalahan dalam kode.
<i>Error</i>	Kesalahan	Masalah pada program.

DAFTAR ISI

COVER

KATA PENGANTAR..... i

TAKARIR..... ii

DAFTAR ISI..... iii

DAFTAR GAMBAR..... iv

BAB I PENDAHULUAN..... 1

1.1 Latar Belakang 1

1.2 Kebutuhan Fungsional..... 1

1.3 Rumusan Masalah 5

1.4 Batasan Masalah..... 5

1.5 Tujuan..... 6

BAB II PERANCANGAN 8

2.1 Analisis Program 8

2.2 Flowchart..... 9

2.3 Konsep/Materi Praktikum yang dipakai 36

BAB III HASIL DAN PEMBAHASAN 40

3.1 Tampilan Program..... 40

3.2 Source Code 56

BAB IV PENUTUP 82

4.1 Kesimpulan..... 82

4.2 Saran..... 82

DAFTAR PUSTAKA 83

LAMPIRAN..... 84

DAFTAR GAMBAR

Gambar 1 Flowchart Menu Utama.....	9
Gambar 2 Flowchart Registrasi.....	11
Gambar 3 Flowchart Login	13
Gambar 4 Flowchart Menu Admin	14
Gambar 5 Flowchart Lihat Inventaris	16
Gambar 6 Flowchart Tambah Barang	17
Gambar 7 Flowchart Update Barang.....	18
Gambar 8 Flowchart Hapus Barang	20
Gambar 9 Flowchart Tampilkan User.....	22
Gambar 10 Flowchart Hapus Akun User	24
Gambar 11 Flowchart Lihat Riwayat Aktivitas	26
Gambar 12 Flowchart Menu User.....	28
Gambar 13 Flowchart Lihat Inventaris (User)	30
Gambar 14 Flowchart Tambah Barang (User).....	32
Gambar 15 Flowchart Update Barang (User)	34
Gambar 16 Flowchart hapus Barang (User).....	35
Gambar 17 Tampilan Menu	40
Gambar 18 Keluar	40
Gambar 19 Menu Pengguna.....	41
Gambar 20 Menu Admin	42
Gambar 21 Registrasi	43
Gambar 22 Login	44
Gambar 23 Lihat Inventaris (User)	45
Gambar 24 Tambah Barang (User)	46
Gambar 25 Update Barang (User).....	46
Gambar 26 Hapus Barang (User).....	47
Gambar 27 keluar menu user	48
Gambar 28 Login Admin	49
Gambar 29 Lihat Inventaris	50

Gambar 30 Tambah Barang	50
Gambar 31 Update Barang.....	51
Gambar 32 Hapus Barang	52
Gambar 33 Tambah Admin Baru	53
Gambar 34 Tampilkan User	54
Gambar 35 Hapus Akun User	55
Gambar 36 Lihat Riwayat Aktivitas	56
Gambar 37 Keluar Menu Admin	56
Gambar 38 Kartu Konsul Cover	84
Gambar 39 Kartu Konsul Lembar 1	85
Gambar 40 Kartu Konsul Lembar 2.....	86

BAB I

PENDAHULUAN

1.1 Latar Belakang

Program “Sistem Manajemen Inventaris Toko Bangunan” merupakan salah satu aspek penting dalam pengelolaan toko bangunan. Toko bangunan biasanya memiliki berbagai jenis barang dengan jumlah besar, mulai dari bahan bangunan seperti semen, pasir, dan batu bata hingga alat-alat seperti paku, cat, dan peralatan tukang. Pengelolaan inventaris yang baik diperlukan untuk memastikan ketersediaan barang, menghindari kelebihan atau kekurangan stok, serta meminimalkan kerugian. Tanpa sistem manajemen yang efisien, toko bangunan dapat mengalami kesulitan dalam memenuhi permintaan pelanggan, yang pada akhirnya akan berdampak pada kepuasan pelanggan dan reputasi usaha.

Manajemen inventaris yang baik tidak hanya meningkatkan efisiensi operasional tetapi juga berkontribusi pada pertumbuhan bisnis. Dengan pengelolaan yang terstruktur, toko bangunan dapat menghindari kerugian finansial akibat salah urus stok, seperti barang hilang atau kesalahan penghitungan. Selain itu, sistem manajemen inventaris yang baik juga memungkinkan toko untuk memberikan pelayanan yang lebih cepat dan akurat kepada pelanggan, meningkatkan daya saing di pasar. Oleh karena itu, penting bagi toko bangunan untuk terus mengembangkan strategi manajemen inventaris agar tetap relevan di era persaingan yang semakin ketat.

1.2 Kebutuhan Fungsional

1. Manajemen Pengguna

- Login Pengguna: Pengguna (baik admin maupun user) dapat login ke dalam sistem menggunakan username dan password yang valid.
- Registrasi Pengguna Baru: Pengguna baru dapat mendaftar untuk membuat akun. Secara default, akun yang dibuat adalah user biasa,

yang hanya dapat melihat inventaris umum, mengupdate, menambah dan menghapus inventaris user.

- **Pengelolaan Akun Admin:** Admin dapat menambahkan akun admin baru. Hanya admin yang memiliki hak akses untuk menambah admin lain ke dalam sistem.
- **Pembatasan Akses Berdasarkan Role:** Program membedakan hak akses antara admin (dengan kontrol penuh terhadap pengelolaan inventaris umum, inventaris user, melihat dan menghapus akun user, serta melihat Riwayat aktivitas) dan user biasa (hanya bisa melihat inventaris umum dan inventaris user, mengedit, menghapus dan menambah barang di inventaris user).

2. Manajemen Inventaris

- **Melihat Daftar Inventaris:** Pengguna (baik admin maupun user) dapat melihat daftar barang yang tersedia di inventaris umum dan user, termasuk nama barang, stok, dan harga.
- **Menambah Barang ke Inventaris:** Admin dapat menambah barang baru ke dalam inventaris umum dan inventaris user dengan memasukkan data barang seperti nama, stok, dan harga. User hanya dapat menambahkan barang baru ke dalam inventaris user dengan memasukkan data barang seperti nama, stok, dan harga.
- **Data Barang:** Admin dapat mengupdate barang dalam inventaris umum dan inventaris user dengan memasukkan data baru barang seperti stok, dan harga. User hanya dapat mengupdate barang dalam inventaris user dengan memasukkan data barang seperti nama, stok, dan harga.
- **Menghapus Barang dari Inventaris:** Admin dapat menghapus barang dalam inventaris umum dan inventaris user dengan memasukkan nama barang yang ingin dihapus. User hanya dapat menghapus barang dalam inventaris user dengan memasukkan nama barang yang ingin dihapus.
- **Penyimpanan dan Pemulihan Data Inventaris:** Data inventaris harus dapat disimpan dalam file CSV dan dimuat kembali setiap kali

program dijalankan, memastikan data tetap ada meskipun program ditutup atau restart.

3. Riwayat Aktivitas

- Pencatatan Riwayat Aktivitas: Setiap perubahan atau tindakan yang dilakukan dalam program (seperti menambah, mengubah, atau menghapus barang) harus tercatat dalam riwayat aktivitas dengan informasi waktu dan deskripsi aktivitas.
- Melihat Riwayat Aktivitas: Admin dapat melihat riwayat aktivitas untuk memantau setiap perubahan yang terjadi dalam sistem, memastikan akuntabilitas dan transparansi pengelolaan inventaris.

4. Keamanan dan Akses Pengguna

- Keamanan Data Pengguna: Program harus memastikan hanya pengguna yang terdaftar dengan username dan password yang benar yang bisa mengakses sistem.
- Verifikasi Login: Program harus memverifikasi setiap pengguna yang mencoba masuk ke sistem, membedakan akses antara admin dan user berdasarkan kredensial yang diberikan.
- Pembatasan Akses Berdasarkan Role: Admin memiliki hak akses penuh untuk mengakses inventaris umum dan inventaris user, sementara user hanya dapat melihat inventaris umum, dan mengakses penuh inventaris user.

5. Interaksi dengan Pengguna

- Antarmuka Pengguna Berbasis Teks: Program menyediakan antarmuka berbasis teks yang jelas dan mudah dimengerti oleh pengguna untuk memilih menu dan melakukan tindakan (seperti melihat inventaris, menambah barang, mengubah data barang, dan logout).

- **Feedback dan Notifikasi:** Program memberikan feedback atau notifikasi yang jelas kepada pengguna setelah melakukan tindakan, seperti pemberitahuan bahwa barang telah berhasil ditambahkan, diperbarui, atau dihapus.
- **Tampilan Inventaris yang Rapi:** Inventaris ditampilkan dalam format tabel yang rapi menggunakan alat seperti PrettyTable, sehingga pengguna dapat dengan mudah melihat detail barang seperti nama, harga, stok, dan siapa yang menambahkannya.

6. Penyimpanan Data

- **Penyimpanan Data Inventaris:** Semua data inventaris barang (nama, stok, harga) harus disimpan dalam file CSV, sehingga dapat dimuat kembali setiap kali program dijalankan tanpa kehilangan data.
- **Penyimpanan Data Pengguna:** Data pengguna (username, password, dan role) harus disimpan dalam file CSV agar bisa dimuat kembali saat pengguna login.
- **Penyimpanan Riwayat Aktivitas:** Semua aktivitas penting yang dilakukan dalam sistem, seperti perubahan inventaris dan login, harus dicatat dalam riwayat aktivitas dan disimpan dalam file CSV untuk tujuan audit.

7. Clear Terminal dan Tampilan Menu

- **Clear Terminal:** Program menyediakan fungsi untuk membersihkan tampilan terminal setelah setiap tindakan atau perubahan menu agar tampilan tetap rapi dan mudah dibaca.
- **Tampilan Menu yang Mudah Dinavigasi:** Menu utama harus jelas dan mudah dimengerti oleh pengguna. Pengguna dapat dengan mudah memilih opsi yang ingin diambil, baik itu admin ataupun user biasa.

8. Backup dan Pemulihan Data

- **Backup Data:** Program memungkinkan pengguna untuk membuat salinan cadangan data secara berkala, sehingga data tetap aman dan dapat dipulihkan jika terjadi kehilangan atau kerusakan file.
- **Pemulihan Data:** Program dapat memuat kembali data inventaris, pengguna, dan riwayat aktivitas yang telah disimpan sebelumnya, memastikan kelangsungan operasional toko meskipun program atau sistem restart.

1.3 Rumusan Masalah

1. Bagaimana merancang dan mengimplementasikan program CRUD (Create, Read, Update, Delete) untuk mendukung manajemen inventaris toko bangunan.
2. Apa saja kendala yang sering dihadapi dalam pengelolaan inventaris manual, dan bagaimana program CRUD dapat menjadi solusi yang efektif?
3. Bagaimana penerapan program CRUD dapat meningkatkan efisiensi dan akurasi dalam proses pengelolaan inventaris toko bangunan?
4. Bagaimana fitur dalam program CRUD, seperti pencatatan stok barang secara real-time, dapat membantu toko bangunan memprediksi kebutuhan stok di masa depan?
5. Apa saja kebutuhan pengguna yang harus dipenuhi oleh program CRUD agar dapat digunakan secara optimal dalam operasional toko bangunan?
6. Bagaimana cara memulihkan data inventaris tiap program dijalankan?

1.4 Batasan Masalah

1. Fokus pada Fungsi CRUD (Create, Read, Update, Delete): Sistem manajemen inventaris yang dikembangkan hanya mencakup fungsi dasar CRUD. Fungsi ini meliputi penambahan data barang (Create), menampilkan data inventaris (Read), memperbarui data barang (Update), dan menghapus data barang yang sudah tidak diperlukan (Delete). Fitur

tambahan seperti menampilkan daftar pengguna user, dan Riwayat aktivitas, bukan bagian dari ruang lingkup sistem ini.

2. Jenis Data yang Dikelola: Program hanya mengelola data inventaris yang berkaitan langsung dengan toko bangunan, seperti nama barang, jumlah stok, dan harga. Data lain seperti informasi pemasok, peringatan stok menipis tidak termasuk dalam sistem ini.
3. Pengguna Sistem: Sistem dirancang untuk digunakan oleh admin toko yang bertanggung jawab atas pengelolaan inventaris dan juga user toko. Sistem ini tidak menyamakan akses user dan admin.
4. Skalabilitas dan Lingkungan Implementasi: Program dibuat untuk mendukung pengelolaan inventaris dalam skala kecil hingga menengah. Sistem ini tidak dirancang untuk mendukung toko bangunan dengan cabang di lokasi berbeda atau integrasi ke sistem berbasis cloud. Selain itu, aplikasi hanya diimplementasikan dalam lingkungan lokal (standalone) tanpa konektivitas jaringan.
5. Keamanan dan Validasi Data: Sistem memiliki validasi dasar untuk memastikan integritas data (seperti memastikan jumlah stok tidak kosong), namun tidak mencakup fitur keamanan tingkat lanjut seperti peringatan stok menipis

1.5 Tujuan

1. Untuk merancang dan mengimplementasikan program CRUD (Create, Read, Update, Delete) untuk mendukung manajemen inventaris toko bangunan.
2. Untuk mengetahui kendala yang sering dihadapi dalam pengelolaan inventaris manual, dan bagaimana program CRUD dapat menjadi solusi yang efektif.
3. Untuk mengetahui penerapan program CRUD dapat meningkatkan efisiensi dan akurasi dalam proses pengelolaan inventaris toko bangunan.

4. Untuk mengetahui fitur dalam program CRUD, seperti pencatatan stok barang secara real-time, dapat membantu toko bangunan memprediksi kebutuhan stok di masa depan.
5. Untuk mengetahui apa saja kebutuhan pengguna yang harus dipenuhi oleh program CRUD agar dapat digunakan secara optimal dalam operasional toko bangunan.
6. Untuk mengetahui cara memulihkan data inventaris tiap program dijalankan.

BAB II

PERANCANGAN

2.1 Analisis Program

Program ini dirancang untuk mengelola inventaris barang dengan fitur manajemen pengguna dan kontrol akses. Program memiliki dua peran utama, yaitu Admin dan User, yang masing-masing memiliki akses berbeda.

1. Struktur Dasar Program

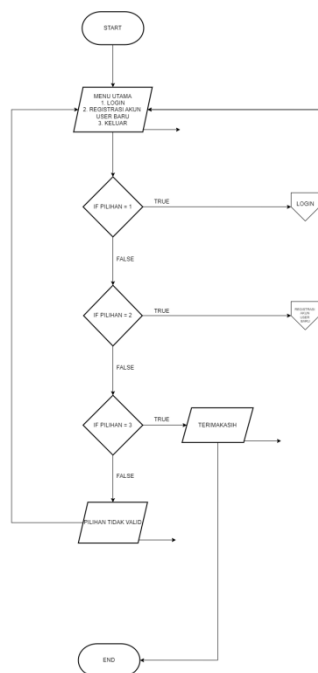
- CSV: Untuk membaca/menulis file CSV.
- Datetime: Untuk mencatat waktu di log aktivitas.
- OS dan platform: Kemungkinan digunakan untuk memastikan kompatibilitas sistem.
- PrettyTable: Untuk menampilkan tabel data dalam format yang rapi di terminal.
- Data Awal: inventory: Sebuah dictionary yang menyimpan data stok awal barang, harga, dan nama pengguna yang menambahkannya (default: admin). inventory_user: Sebuah dictionary kosong yang kemungkinan digunakan untuk menyimpan data inventaris milik pengguna tertentu.

2. Alur Program

- Pengelolaan Inventaris:
- Program memuat fungsi-fungsi untuk:
 - a. Menampilkan stok barang.
 - b. Menambah barang baru.
 - c. Menghapus barang tertentu.
 - d. Memperbarui stok atau harga barang.
- Manajemen Pengguna, terdapat fungsi untuk:
 - a. Menambahkan pengguna baru.
 - b. Memberikan akses khusus kepada pengguna tertentu untuk mengelola inventaris.

- c. Pencatatan Riwayat Aktivitas: Program mencatat aktivitas pengguna, seperti menambah atau mengubah data barang, dengan cap waktu menggunakan modul datetime.
- d. Output Tabel: Data ditampilkan dalam format tabel yang rapi menggunakan PrettyTable, membuatnya lebih mudah dibaca di terminal.

2.2 Flowchart



Gambar 1 Flowchart Menu Utama

1. Flowchart Menu Utama

Program menampilkan 3 pilihan utama:

- Registrasi Akun User Baru
- Login
- Keluar

Logika pilihan:

Jika Pilihan = 1 (Login):

- Program mengarahkan ke proses Login, di mana pengguna harus memasukkan kredensial mereka.
- Setelah login berhasil, program akan mengarahkan pengguna ke menu sesuai dengan perannya (Admin atau User).

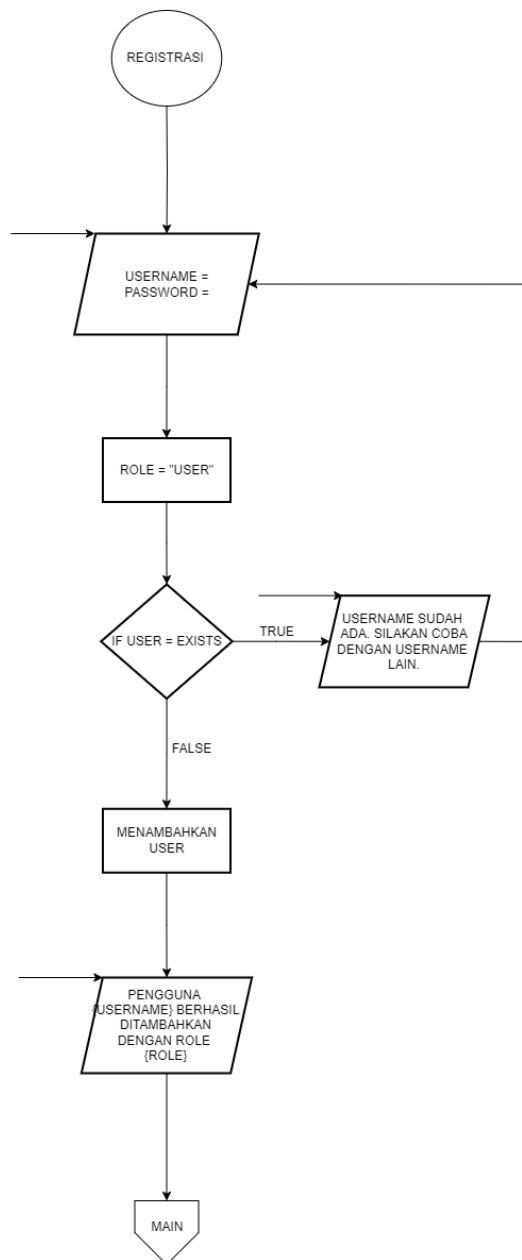
Jika Pilihan = 2 (Registrasi):

- Program mengarahkan ke proses Registrasi di mana pengguna baru dapat membuat akun.
- Setelah proses selesai, program kembali ke menu utama.

Jika Pilihan = 3 (Keluar):

- Program menampilkan pesan “Terima Kasih” dan proses berakhir
- Jika pengguna memasukkan pilihan yang tidak valid (selain 1, 2, atau 3), program akan menampilkan pesan “Pilihan Tidak Valid” dan kembali ke menu utama.
- Program akan berhenti jika pengguna memilih opsi 3 (Keluar).

2. Flowchart Registrasi



Gambar 2 Flowchart Registrasi

Proses Registrasi Dimulai:

- Pengguna memulai proses registrasi dengan memasukkan username dan password.

Penetapan Role:

- Setiap pengguna yang diregistrasi secara default diberikan role = “User”.

Validasi Username:

- Sistem memeriksa apakah username yang dimasukkan sudah ada di database: Jika username sudah ada: Sistem menampilkan pesan: “Username sudah ada, silakan coba dengan username lain.”
- Pengguna diminta memasukkan username baru.
- Jika username belum ada: Sistem melanjutkan proses untuk menambahkan pengguna baru ke database.

Penambahan User:

- Sistem menyimpan data pengguna baru (username, password, dan role) ke dalam database.

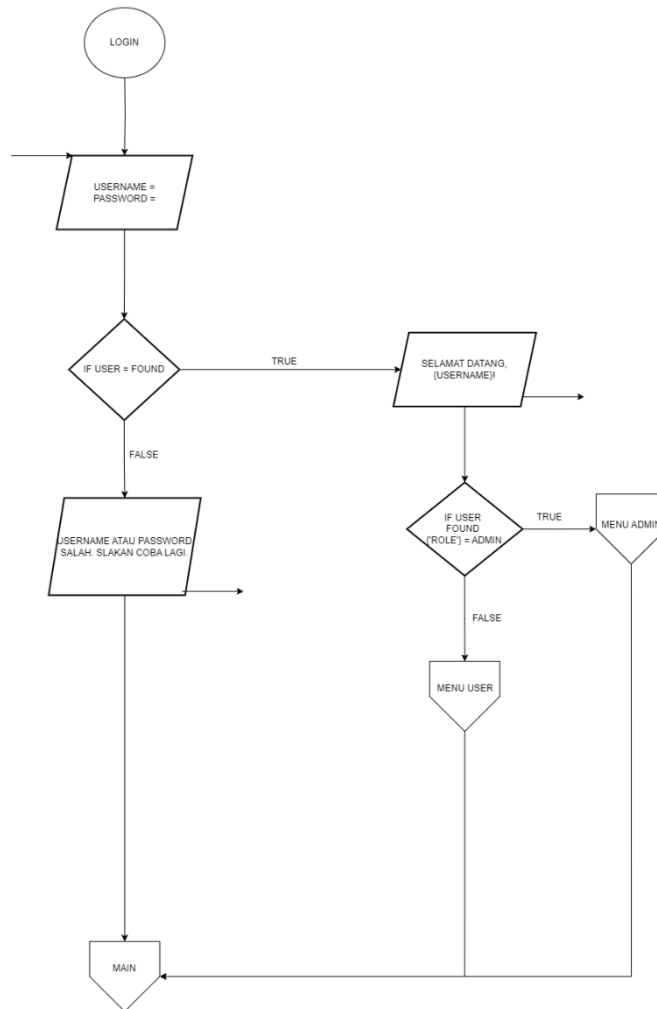
Konfirmasi Berhasil:

- Sistem menampilkan pesan: “Pengguna (username) berhasil ditambahkan dengan role (role)” untuk memberi tahu bahwa proses registrasi berhasil.

Kembali ke Menu Utama:

- Setelah registrasi selesai, alur diarahkan kembali ke menu utama.

3. Flowchart Login



Gambar 3 Flowchart Login

Login:

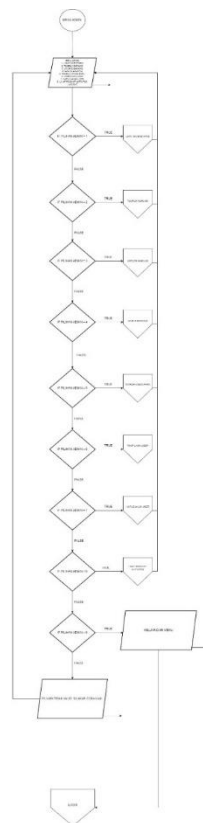
- Pengguna diminta untuk memasukkan username dan password.

Validasi Pengguna:

- Program memeriksa apakah username dan password yang dimasukkan ditemukan di database.
- Jika valid (user ditemukan): Program menampilkan pesan “Selamat datang, [username]”.
- Melanjutkan dengan mengecek peran pengguna (role).
Jika role = Admin, pengguna diarahkan ke Menu Admin.
Jika role = User, pengguna diarahkan ke Menu User.

- Pengalihan ke Menu Utama:

- #### 4. Flowchart Menu Admin



Masuk ke Menu Admin:

Program dimulai dengan menu utama yang menawarkan berbagai opsi yang dapat dipilih admin. Menu ini mencakup:

- Melihat inventaris.
- Menambahkan barang.
- Memperbarui barang.
- Menghapus barang.
- Menambahkan admin baru.
- Menampilkan pengguna.
- Menghapus akun pengguna.
- Melihat riwayat aktivitas.
- Keluar dari menu.

Proses Pemilihan Opsi Admin:

- Admin memilih opsi berdasarkan angka yang sesuai dari menu.

Setiap pilihan diverifikasi:

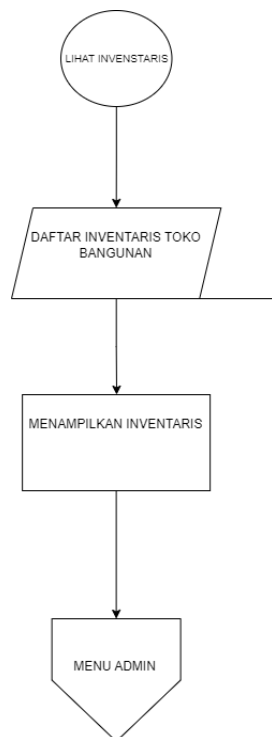
- Jika valid, proses yang terkait dengan pilihan tersebut dijalankan.
- Jika tidak valid, program memberikan pesan kesalahan “Pilihan Tidak Valid” dan kembali ke menu admin.

Deskripsi Setiap Fungsi:

- Lihat Inventaris: Admin dapat melihat daftar barang yang ada di inventaris.
- Tambah Barang: Admin menambahkan barang baru ke inventaris, biasanya dengan menginput data seperti nama barang, jumlah, dan detail lainnya.
- Update Barang: Admin dapat mengedit informasi barang yang sudah ada, seperti mengubah jumlah atau detail barang.
- Hapus Barang: Admin menghapus barang dari inventaris jika tidak lagi dibutuhkan.
- Tambah Admin Baru: Admin saat ini dapat mendaftarkan admin lain dengan memberikan data yang diperlukan.

- Tampilkan Pengguna: Daftar semua pengguna ditampilkan untuk dilihat oleh admin.
- Hapus Akun Pengguna: Admin memiliki opsi untuk menghapus akun pengguna tertentu.
- Lihat Riwayat Aktivitas: Admin dapat melihat log aktivitas, seperti data perubahan inventaris atau riwayat login.
- Keluar dari Menu: Admin keluar dari sistem atau kembali ke menu utama.

5. Flowchart Lihat Inventaris



Gambar 5 Flowchart Lihat Inventaris

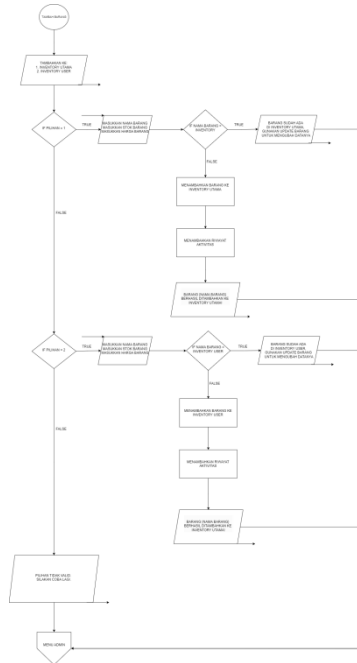
Mulai dengan Lihat Inventaris:

Sistem dimulai dari fitur “Lihat Inventaris”, di mana pengguna (user/admin) ingin melihat daftar inventaris toko bangunan.

Daftar Inventaris Toko Bangunan:

Sistem menampilkan daftar inventaris dari toko bangunan. Data ini memuat semua informasi terkait barang yang ada di toko.

6. Flowchart Tambah Barang



Gambar 6 Flowchart Tambah Barang

Tambah Barang

Awalnya, ada pilihan untuk menambahkan barang ke:

- Inventori utama
- Inventori user

Pilih Lokasi Tambah Barang

- Jika pilih 1 (Inventori utama):
 - Masukkan nama barang, stok, dan harga barang.
 - Dicek apakah nama barang sudah ada di inventori utama:
 - Kalau barang sudah ada: Sistem menyarankan untuk update data barang tersebut.
 - Kalau barang belum ada: Barang ditambahkan ke inventori utama, dan riwayat aktivitas juga diperbarui.
- Jika pilih 2 (Inventori user):
 - Masukkan nama barang, stok, dan harga barang
 - Dicek apakah nama barang sudah ada di inventori user:

- Kalau barang sudah ada: Sistem menyarankan untuk update data barang tersebut.
- Kalau barang belum ada: Barang ditambahkan ke inventori user, dan riwayat aktivitas juga diperbarui.

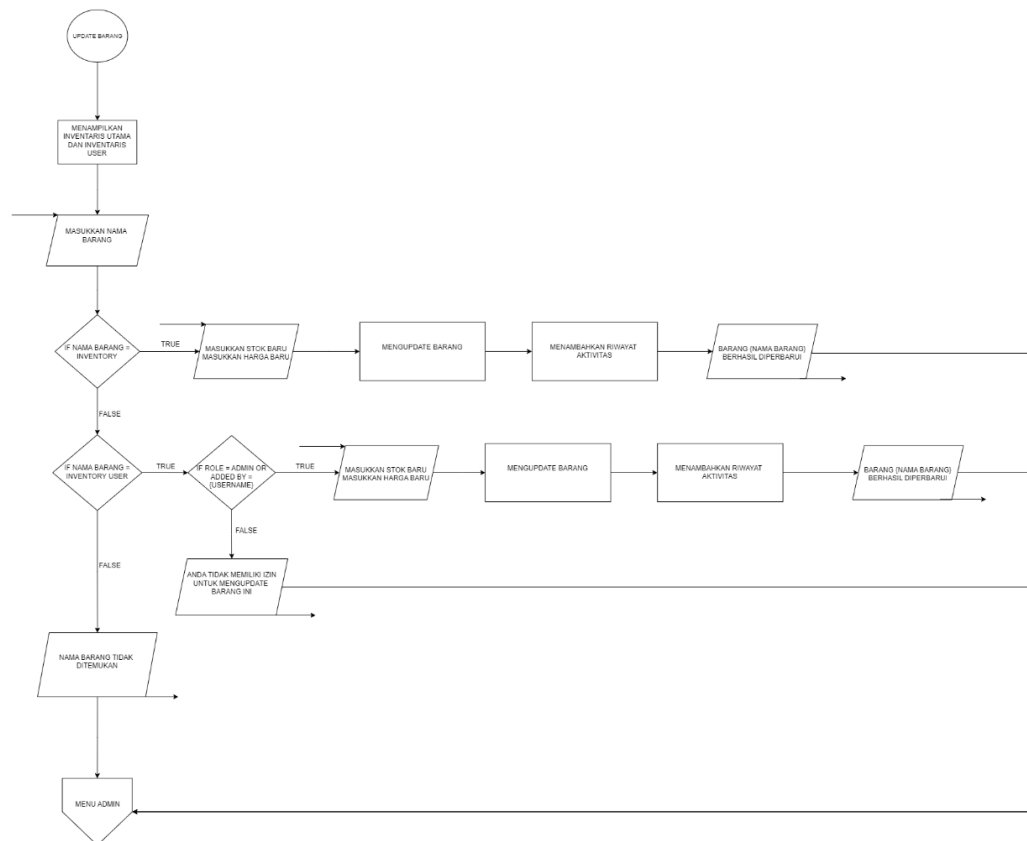
Jika Pilihan Tidak Valid

Kalau pilihan yang dimasukkan salah (tidak 1 atau 2), sistem akan memberi notifikasi dan meminta untuk mencoba lagi.

Selesai

Setelah proses selesai, admin bisa melanjutkan aktivitas lain.

7. Flowchart Update Barang



Gambar 7 Flowchart Update Barang

Memulai Proses Update Barang

- Alur dimulai dengan langkah untuk memperbarui data inventaris dan informasi user.

Memasukkan Nama Barang

- Sistem meminta input berupa nama barang yang ingin diperbarui.

Pengecekan Nama Barang di Inventaris

- Jika nama barang ditemukan:

Dilanjutkan dengan memasukkan stok baru dan/atau harga baru.\

- Jika nama barang tidak ditemukan:
- Sistem akan mengecek apakah user memiliki hak akses untuk menambahkan atau memperbarui data barang.

Validasi Hak Akses User

- Jika user memiliki hak akses (role admin atau sesuai):
- User diarahkan untuk memasukkan data baru (stok atau harga).
- Jika user tidak memiliki hak akses:
- Sistem memberi notifikasi bahwa user tidak diizinkan memperbarui barang ini.

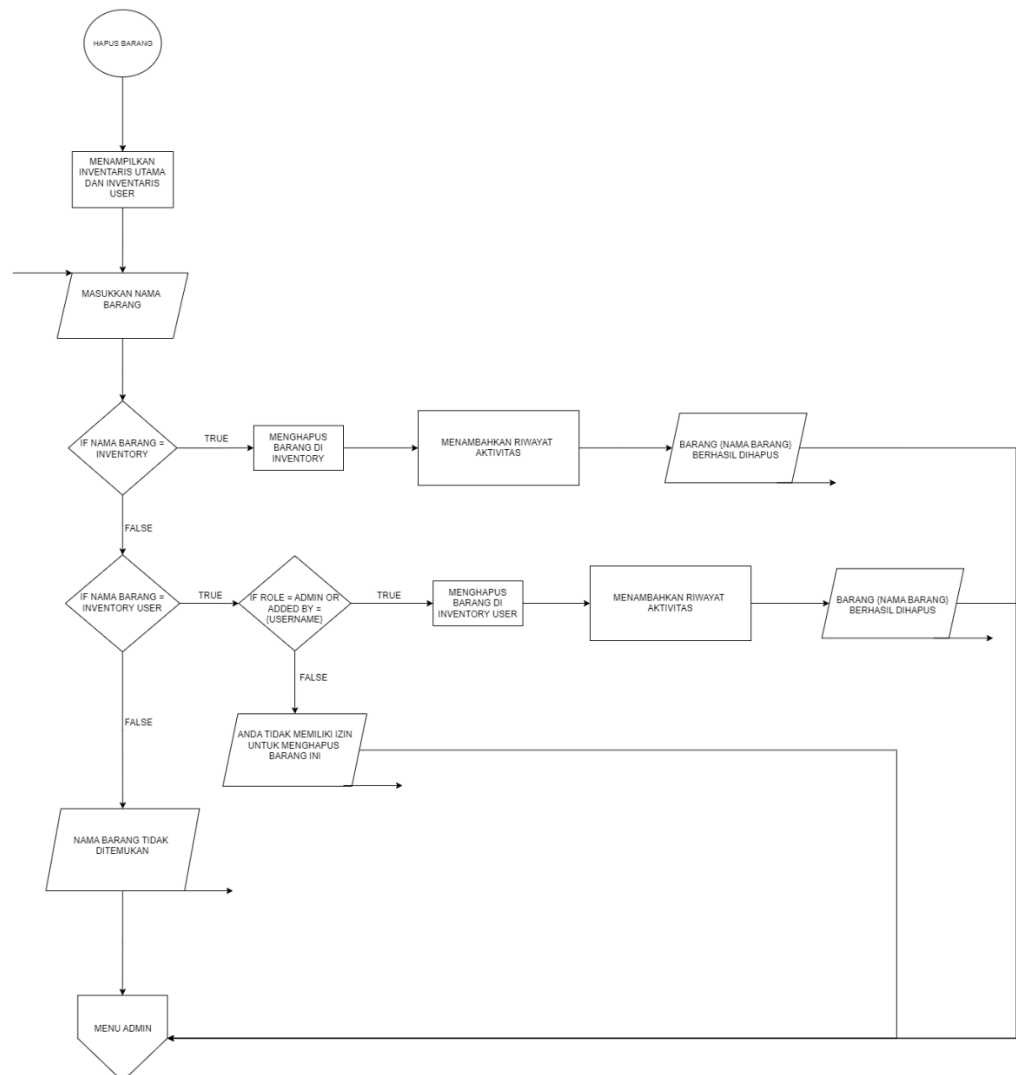
Update Barang

- Setelah data baru dimasukkan (stok dan/atau harga), sistem:
- Menambahkan riwayat aktivitas untuk mencatat perubahan.
- Memperbarui nama barang jika diperlukan.

Mengakhiri Proses

- Jika proses berhasil, pengguna diarahkan kembali ke menu admin untuk melanjutkan aktivitas lainnya.

8. Flowchart Hapus Barang



Gambar 8 Flowchart Hapus Barang

Memulai Proses Hapus Barang

- Alur dimulai dengan menampilkan inventaris utama dan inventaris user.

Memasukkan Nama Barang

- Sistem meminta input berupa nama barang yang akan dihapus.

Pengecekan Nama Barang di Inventaris Utama

- Jika nama barang ditemukan:
- Sistem melanjutkan dengan langkah untuk menghapus barang di inventaris.

- Jika nama barang tidak ditemukan:
- Sistem memeriksa apakah barang tersebut berada di inventaris user.

Pengecekan Nama Barang di Inventaris User

- Jika nama barang ditemukan:
- Sistem akan memvalidasi hak akses user.
- Jika nama barang tidak ditemukan:
- Sistem memberikan notifikasi bahwa nama barang tidak ditemukan.

Validasi Hak Akses User

- Jika user memiliki hak akses (role admin atau barang ditambahkan oleh user tersebut):
- Sistem menghapus barang di inventaris.
- Jika user tidak memiliki hak akses:
- Sistem memberikan notifikasi bahwa user tidak diizinkan untuk menghapus barang ini.

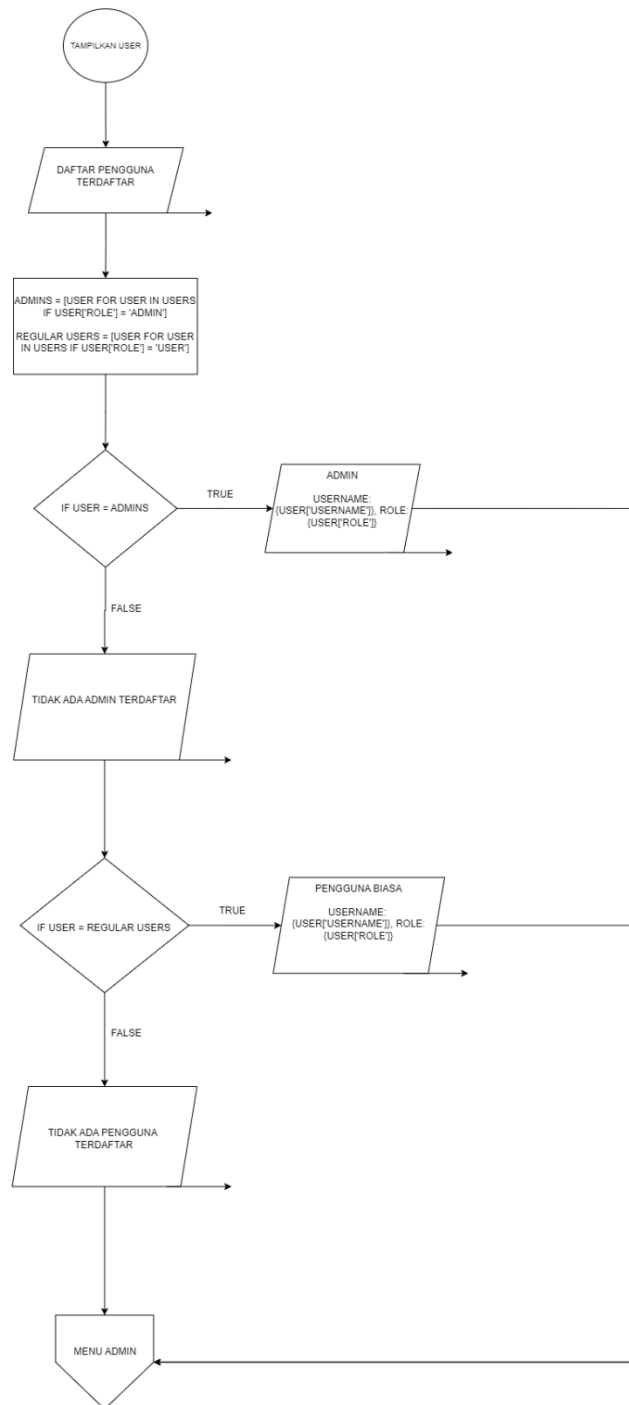
Menghapus Barang

- Setelah validasi selesai, sistem:
- Menghapus barang dari inventaris.
- Menambahkan riwayat aktivitas untuk mencatat penghapusan.
- Memberikan notifikasi bahwa barang berhasil dihapus.

Mengakhiri Proses

- Setelah proses selesai, pengguna diarahkan kembali ke menu admin.

9. Flowchart Tampilkan User



Gambar 9 Flowchart Tampilkan User

Tampilkan Pengguna

- Sistem menampilkan daftar pengguna yang terdaftar.

Pisahkan Pengguna Berdasarkan Peran

- Admins: Daftar pengguna dengan peran “Admin”.
- Regular Users: Daftar pengguna dengan peran “Pengguna Biasa”.

Periksa Apakah Pengguna Termasuk Admin

- Jika pengguna ditemukan dalam daftar Admin, tampilkan informasi pengguna dengan peran “Admin”.
- Jika Admin kosong, lanjutkan ke langkah berikutnya.

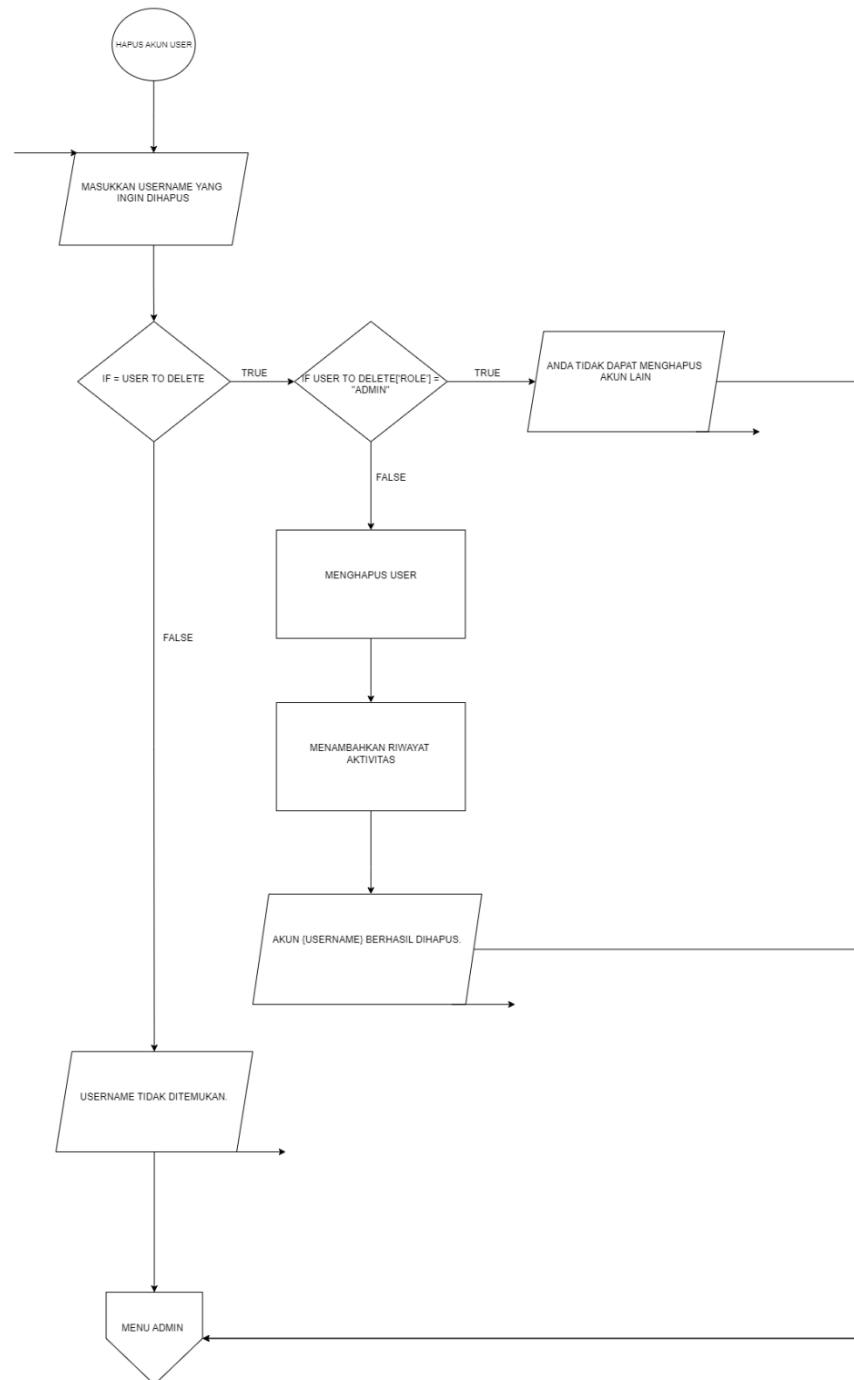
Periksa Apakah Pengguna Termasuk Regular Users

- Jika pengguna ditemukan dalam daftar Regular Users, tampilkan informasi pengguna dengan peran “Pengguna Biasa”.
- Jika Regular Users kosong, lanjutkan ke langkah berikutnya.
- Jika Tidak Ada Pengguna Terdaftar
- Jika tidak ada admin maupun pengguna biasa yang terdaftar, sistem mencatat bahwa tidak ada pengguna sama sekali.

Menu Admin

- Setelah memproses pengguna, sistem diarahkan ke menu khusus untuk admin.

10. Flowchart Hapus Akun User



Gambar 10 Flowchart Hapus Akun User

Mulai: Proses dimulai dari menu “Hapus Akun User”.

Memasukkan Username yang Ingin Dihapus:

- Admin diminta untuk memasukkan username dari akun yang akan dihapus.

Memeriksa Username:

- Jika username ditemukan di sistem, maka dilanjutkan ke langkah berikutnya.
- Jika tidak ditemukan, sistem akan memberikan pesan “Username Tidak Ditemukan” dan mengembalikan admin ke menu utama.

Memeriksa Hak Akses Akun:

- Jika akun yang akan dihapus memiliki peran “Admin”, maka muncul pesan “Anda Tidak Dapat Menghapus Akun Admin Lain”, dan proses kembali ke menu utama.
- Jika akun bukan “Admin”, proses dilanjutkan.

Menghapus Akun:

- Sistem menghapus akun yang sesuai dengan username yang dimasukkan.

Menambahkan Riwayat Aktivitas:

- Sistem mencatat aktivitas penghapusan akun ke dalam log riwayat aktivitas.

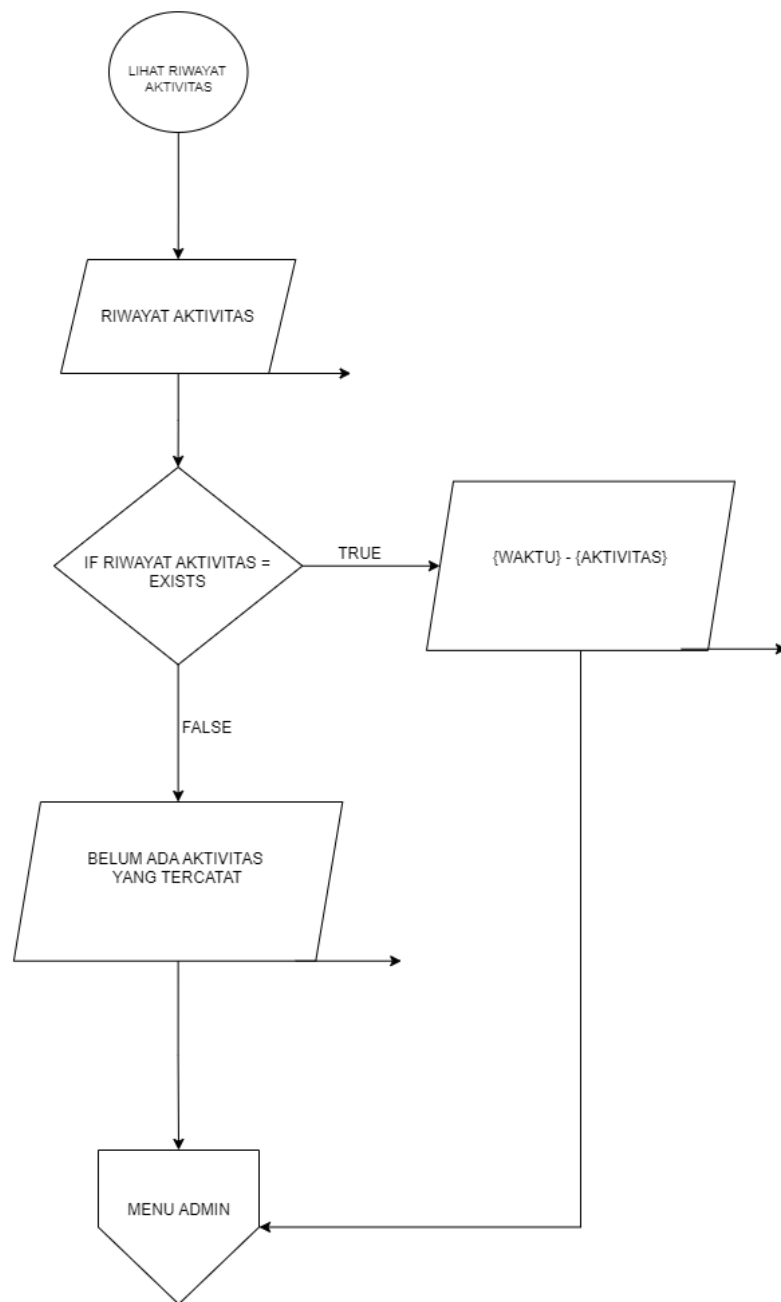
Konfirmasi Penghapusan:

- Sistem menampilkan pesan bahwa akun dengan username tertentu telah berhasil dihapus.

Kembali ke Menu Admin:

- Setelah proses selesai, admin dikembalikan ke menu utama untuk melakukan aktivitas lain.

11. Flowchart Lihat Riwayat Aktivitas



Gambar 11 Flowchart Lihat Riwayat Aktivitas

Lihat Riwayat Aktivitas:

Proses dimulai dengan perintah untuk melihat riwayat aktivitas. Ini bisa berupa tombol yang ditekan oleh pengguna atau perintah yang dijalankan secara otomatis.

Riwayat Aktivitas:

Sistem kemudian memeriksa apakah terdapat data riwayat aktivitas yang tersimpan. Data ini biasanya berisi informasi tentang waktu kejadian dan jenis aktivitas yang dilakukan.

Percabangan:

Sistem akan melakukan percabangan berdasarkan hasil pemeriksaan pada langkah sebelumnya:

- Jika Riwayat Aktivitas Ada (TRUE):

Sistem akan menampilkan daftar riwayat aktivitas yang telah tersimpan. Daftar ini biasanya berisi informasi waktu dan jenis aktivitas dalam format yang mudah dibaca oleh pengguna.

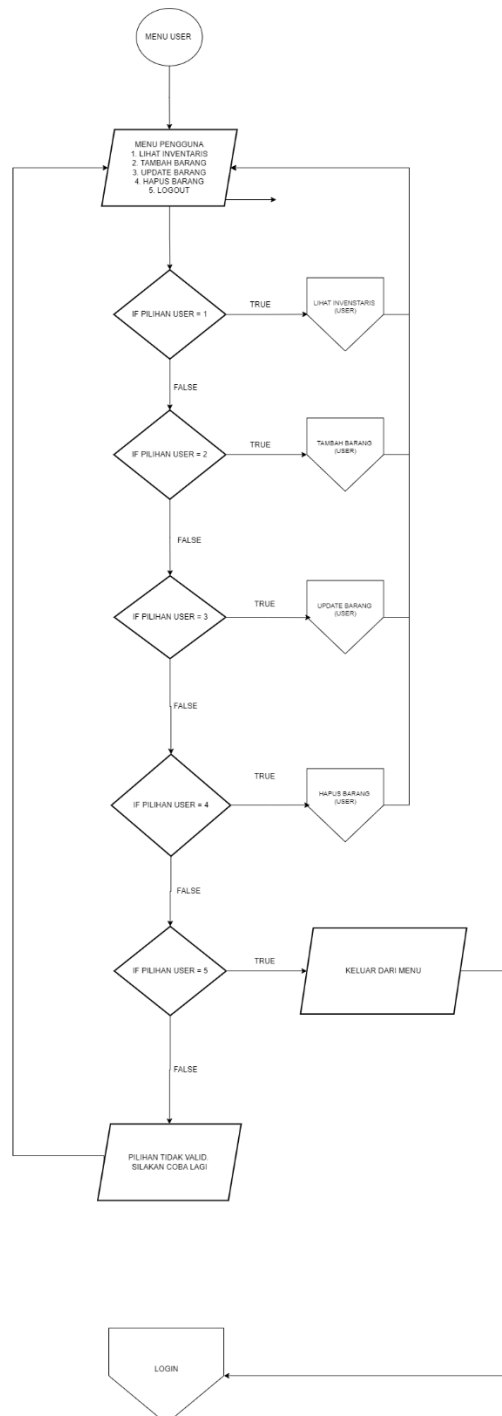
- Jika Riwayat Aktivitas Tidak Ada (FALSE):

Sistem akan menampilkan pesan bahwa “Belum ada aktivitas yang tercatat”. Ini menginformasikan kepada pengguna bahwa tidak ada data aktivitas yang dapat ditampilkan.

Menu Admin:

Setelah proses menampilkan riwayat aktivitas selesai, sistem akan kembali ke menu utama atau menu admin. Hal ini memungkinkan pengguna untuk melakukan tindakan lain yang terkait dengan sistem.

12. Flowchart Menu User



Gambar 12 Flowchart Menu User

Mulai dari Menu User:

Pengguna diberikan daftar opsi dalam menu:

- Lihat Inventaris

- Tambah Barang
- Update Barang
- Hapus Barang
- Logout

Pemilihan Menu oleh Pengguna:

- Sistem memeriksa input pilihan pengguna.

Logika Pilihan:

- Pilihan 1 (Lihat Inventaris):
- Jika pengguna memilih opsi ini, sistem akan menampilkan daftar inventaris.
- Pilihan 2 (Tambah Barang):
- Sistem membuka fitur untuk menambahkan barang baru ke dalam inventaris.
- Pilihan 3 (Update Barang):
- Sistem memberikan opsi untuk memperbarui informasi barang yang sudah ada.
- Pilihan 4 (Hapus Barang):
- Pengguna diarahkan ke fitur penghapusan barang dari inventaris.
- Pilihan 5 (Logout):
- Sistem akan keluar dari menu pengguna dan kembali ke halaman login.

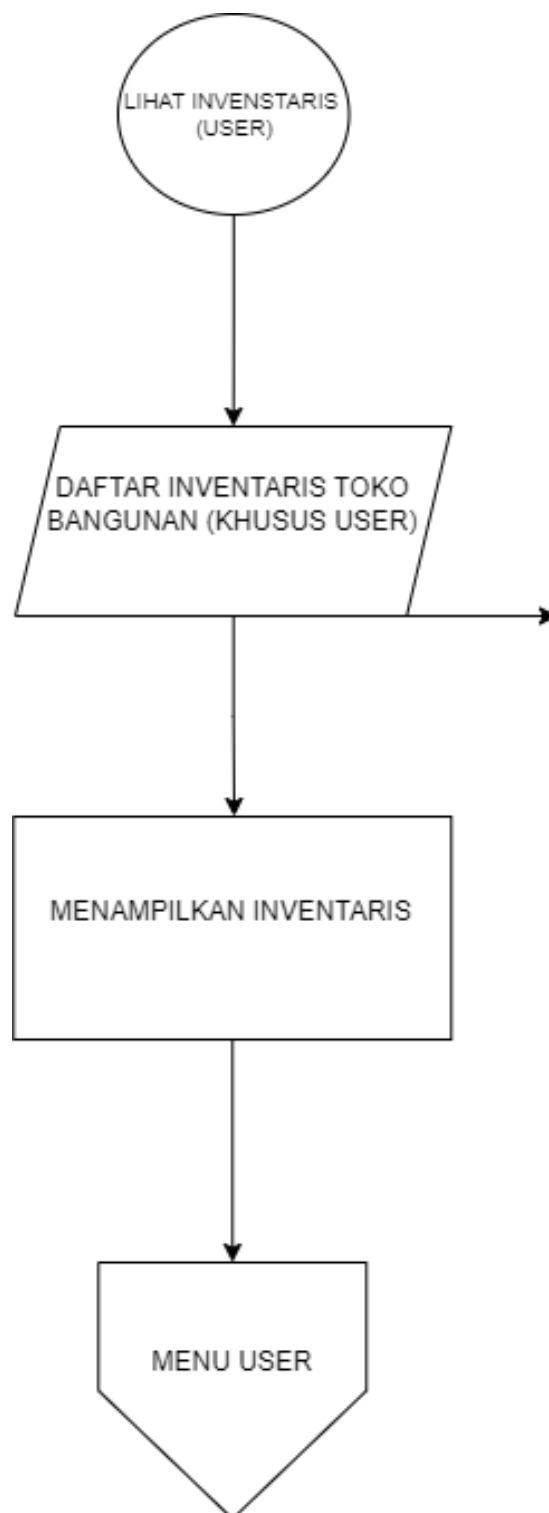
Validasi Input:

Jika input pengguna tidak sesuai dengan opsi (1–5), maka muncul pesan "Pilihan Tidak Valid, Silakan Coba Lagi". Pengguna diarahkan kembali ke menu untuk memilih ulang.

Kembali ke Login atau Menu:

Setelah proses logout (pilihan 5), pengguna diarahkan ke halaman login. Jika opsi lain selesai diproses, pengguna tetap berada di Menu Pengguna.

13. Flowchart Lihat Inventaris (User)



Gambar 13 Flowchart Lihat Inventaris (User)

Mulai (Lihat Inventaris - User):

- Proses dimulai dari user yang ingin melihat inventaris.

Daftar Inventaris Toko Bangunan:

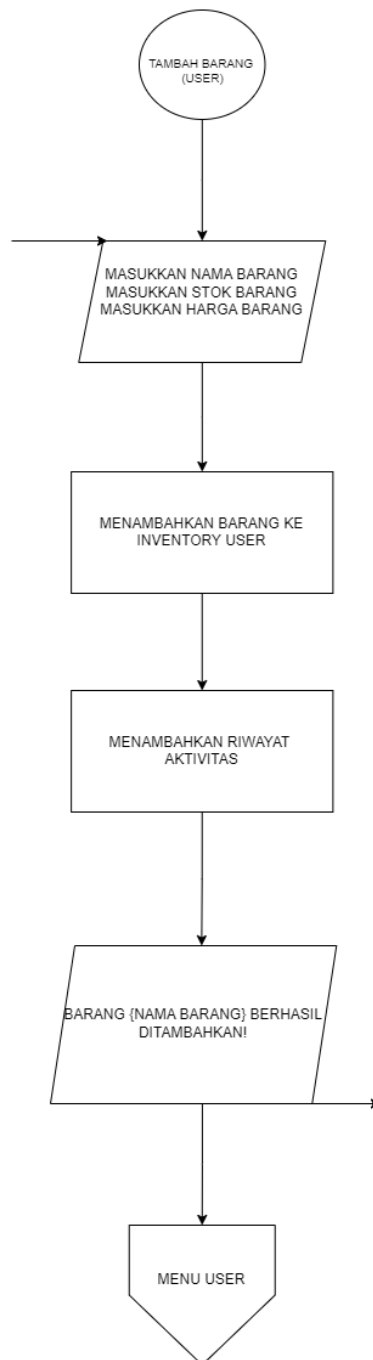
- Sistem menampilkan daftar inventaris toko bangunan yang hanya dapat diakses oleh user tertentu.

Sistem menampilkan rincian barang yang ada di inventaris, meliputi:

Kembali ke Menu User:

- Proses berakhir, dan user diarahkan kembali ke menu utama.

14. Flowchart Tambah Barang (User)



Gambar 14 Flowchart Tambah Barang (User)

Mulai dari Proses “Tambah Barang (User)”

- Proses dimulai ketika pengguna memilih fitur untuk menambahkan barang.

Input Data Barang

- Pengguna diminta untuk memasukkan informasi terkait barang, yaitu:
- Nama Barang: Nama yang akan digunakan untuk mengidentifikasi barang.
- Stok Barang: Jumlah barang yang tersedia.
- Harga Barang: Harga satuan dari barang tersebut.

Menambahkan Barang ke Inventory

- Data yang dimasukkan pengguna akan diproses dan ditambahkan ke dalam inventaris milik pengguna.

Menambahkan ke Riwayat Aktivitas

- Sistem mencatat tindakan ini ke dalam riwayat aktivitas pengguna sebagai log atau bukti bahwa barang telah berhasil ditambahkan.

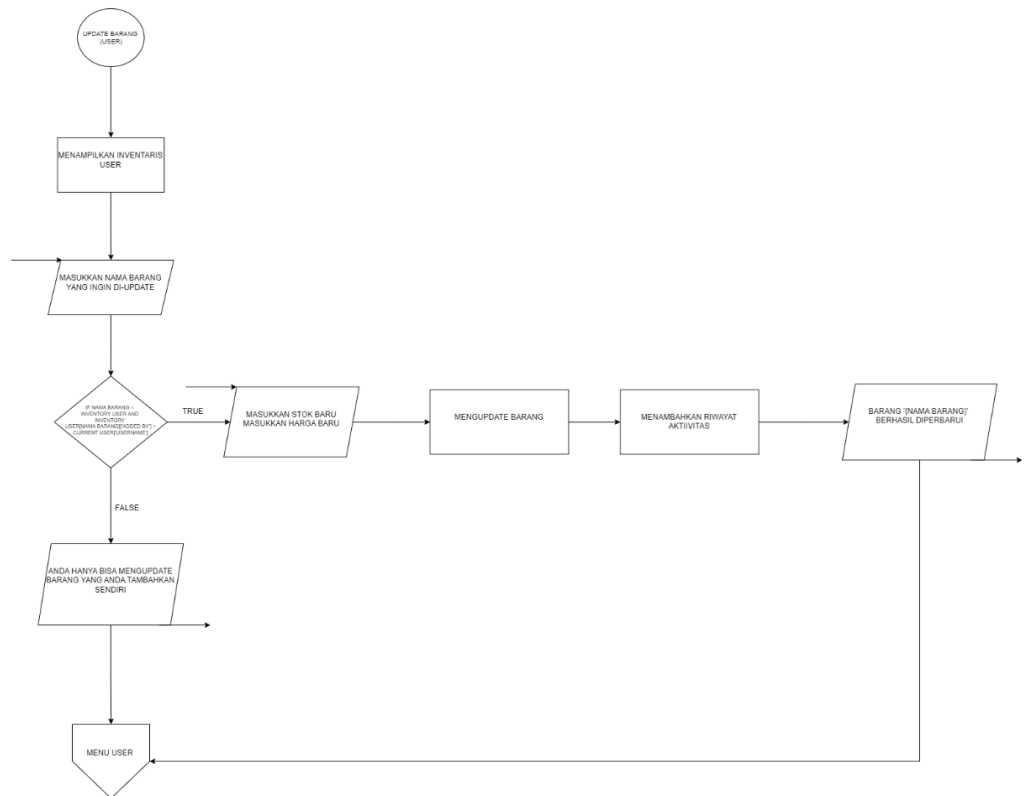
Konfirmasi Keberhasilan

- Sistem memberikan notifikasi kepada pengguna bahwa barang dengan nama tertentu berhasil ditambahkan.

Kembali ke Menu User

- Setelah proses selesai, pengguna diarahkan kembali ke menu utama untuk melakukan tindakan lainnya.

15. Flowchart Update Barang (User)



Gambar 15 Flowchart Update Barang (User)

Mulai (Start):

- Proses dimulai dari langkah “Update Barang User”.

Menampilkan Inventaris User:

- Sistem menampilkan daftar inventaris barang milik user.

Input Nama Barang:

- User diminta untuk memasukkan nama barang yang ingin diperbarui.

Pengecekan Barang:

- Sistem melakukan pengecekan apakah barang yang diinputkan sesuai dengan daftar inventaris user.

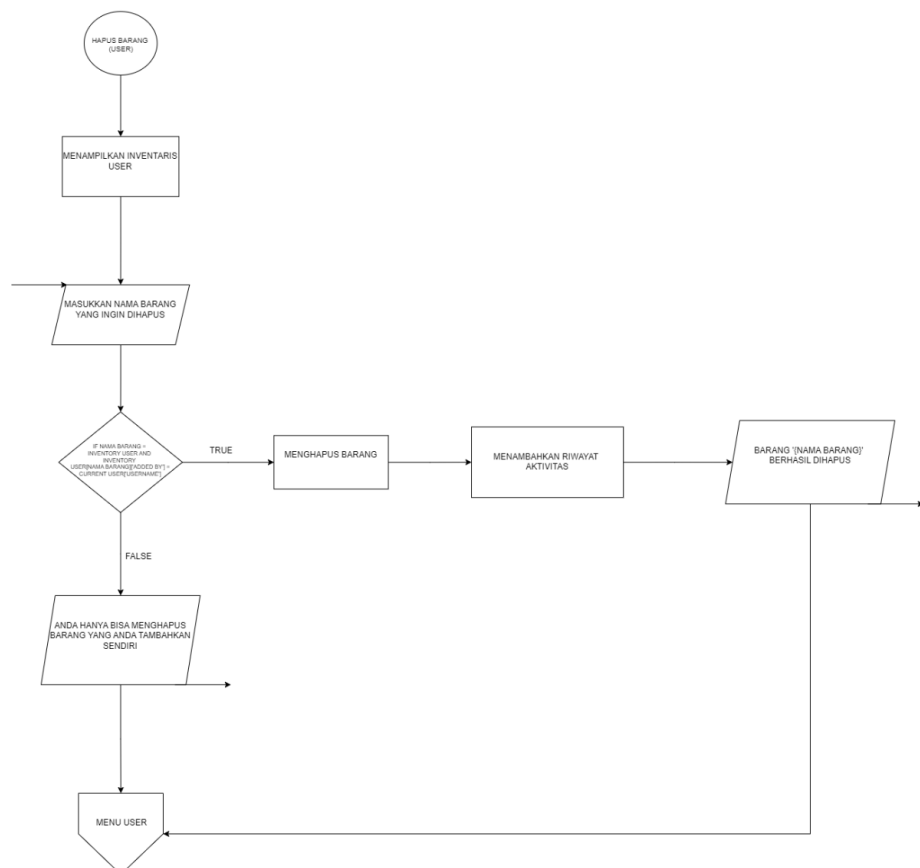
Jika benar (TRUE):

- Input Data Baru: User diminta memasukkan stok baru dan harga baru untuk barang tersebut.

Mengupdate Barang: Sistem memperbarui data barang sesuai dengan input user.

- Menambahkan Riwayat Aktivitas: Sistem mencatat perubahan yang dilakukan ke dalam riwayat aktivitas.
- Barang Berhasil Diperbarui: Proses pembaruan barang selesai.
- Jika salah (FALSE):
- Sistem memberikan pesan kesalahan: “Anda hanya bisa mengupdate barang yang ada di inventaris sendiri.”
- Proses diarahkan kembali ke Menu User.

16. Flowchart Hapus Barang (User)



Gambar 16 Flowchart hapus Barang (User)

Memulai Proses Hapus Barang (User)

- Proses dimulai dengan menampilkan daftar inventaris barang milik user.

Memasukkan Nama Barang yang Ingin Dihapus

- User diminta memasukkan nama barang yang ingin dihapus dari inventaris mereka.

Pengecekan Validitas Nama Barang

- Sistem memeriksa apakah nama barang yang dimasukkan ada di inventaris user:
- Jika valid (TRUE): Sistem melanjutkan proses untuk menghapus barang.
- Jika tidak valid (FALSE): Sistem memberikan notifikasi bahwa user hanya bisa menghapus barang yang ditambahkan sendiri.
- Proses Penghapusan Barang

Jika barang ditemukan dan valid:

- Barang akan dihapus dari inventaris user.
- Sistem mencatat tindakan ini dengan menambahkan riwayat aktivitas.
- Notifikasi Penghapusan Berhasil
- Setelah barang berhasil dihapus, sistem memberikan konfirmasi bahwa barang telah dihapus.
- Mengakhiri Proses
- Setelah semua langkah selesai, user diarahkan kembali ke menu user untuk melakukan aktivitas lain.

2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan persyaratan yang terdapat pada modul diantaranya:

1. Fungsi dasar: fungsi yang kami gunakan adalah fungsi print.

Fungsi print pada program merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga.

2. Fungsi *Number (int)*. Fungsi Number adalah tipe data number merepresentasikan nilai-nilai berupa angka. Karena python menggolongkan beberapa tipe data umum seperti, interger(bilangan bulat) dan floating point(bilangan decimal) ke dalam tipe data number.
3. Fungsi CRUD Aplikasi *create, read, update dan delete (CRUD)* merupakan operasi dasar yang umum digunakan pada pemrograman database.
4. Fungsi *list*. Fungsi list sering disebut juga array pada Bahasa pemrograman lain, list adalah jenis data campuran yang bisa memiliki komponen penyusun yang berbeda-beda. Dan dalam list kita bisa
5. Fungsi *Dictionary*. Fungsi Dictionary adalah sebuah tipe data yang anggotanya terdiri dari pasangan kunci : nilai (key :value) untuk merujuk pada isi variabelnya. Dictionary sifat kedua jenis data ini hanya berbeda dalam beberapa hal saja. Untuk mendeklarasikan sebuah dictionary, dalam python menggunakan atau memakai tanda { }.
6. Fungsi percabangan : didalam program kami menggunakan fungsi percabangan If Fungsi percabangan If digunakan saat terdapat satu pilihan keputusan. Pada saat menggunakan operator if relasi sama dengan (==) untuk membandingkan isi variabel. Sedangkan tanda titik-dua (:) adalah tanda untuk memulai blok kode if. Dan untuk penulisan blok if , harus diberikan indentasi tab atau spasi 2x.
7. Fungsi percabangan : pada fungsi percabangan di program kami bukan hanya menggunakan pfungsi If tapi juga menggunakan fungsi percabangan If/Else/Elif.Fungsi If / Else/ Elif digunakan untuk memuat kondisi/logika tambahan apabila kondisi pertama salah.
8. Fungsi *Def*.Fungsi def adalah sebuah blok kode yang terorganisir dengan baik sehingga suatu saat kita dapat menggunakan nya Kembali (reusable). Atau suatu cara untuk mendefinisikan sebuah method atau fungsi.
9. Fungsi *Return*. Tujuannya, agar sebuah function bisa mengembalikan nilai. Perintah return ada di baris 3, yakni return luas. Artinya, kembalikan

nilai yang tersimpan di dalam variabel luas kepada kode yang akan memanggil function tersebut.

10. Fungsi perulangan : fungsi perulangan yang kami gunakan adalah fungsi perulangan For. Fungsi perulangan For atau yang disebut juga counted loop (perulangan yang terhitung). Perulangan For I biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangan. Variabel I berfungsi untuk menampung indeks, dan fungsi range berfungsi untuk membuat list dengan range.
11. Fungsi *String (str)*. Selain angka, dalam python juga mampu melakukan manipulasi string. Yang dapat di ekspresikan dengan beberapa cara, penulisan nilai string pada python menggunakan tanda petik satu (') atau tanda petik dua("").
12. Fungsi *continue* dalam pemrograman digunakan untuk melewati satu iterasi tertentu dalam suatu loop (seperti loop for atau while) dan langsung melanjutkan ke iterasi berikutnya. Ketika program menemui perintah continue, sisa kode di dalam loop yang tersisa setelah perintah continue akan dilewati, dan kontrol program langsung menuju ke iterasi berikutnya.
13. *Error handling* adalah proses dalam pemrograman untuk menangani kesalahan atau situasi yang tidak terduga secara sistematis agar aplikasi tidak crash dan dapat tetap berjalan dengan baik.
14. Fungsi file eksternal dalam pemrograman merujuk pada penggunaan file yang berada di luar program untuk menyimpan dan membaca data. File eksternal dapat berupa berbagai format, seperti teks, CSV, JSON, XML, atau bahkan gambar dan video. Dengan menggunakan file eksternal, kita dapat menyimpan data secara permanen, berbagi data antar aplikasi, atau menyimpan konfigurasi program.
15. Fungsi *Input*. Fungsi Input adalah suatu algoritma harus memiliki nol atau lebih input. Artinya, suatu algoritma itu mungkin tidak memiliki input secara langsung dari pengguna tetapi dapat juga memiliki beberapa masukan. Algoritma tidak memiliki input secara langsung dari pengguna, maka semua data dapat diinisialisasikan atau dibandingkan dalam

algoritma. 14. 17. Fungsi output Fungsi output pada algoritma harus memiliki satu atau lebih output. Suatu algoritma yang tidak memiliki output adalah algoritma yang sia-sia karena algoritma dibuat untuk menghasilkan sesuatu yang diinginkan, yaitu hasil berupa keluaran.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu



Gambar 17 Tampilan Menu

Menampilkan menu utama program, dengan pesan “Selamat Datang: Toko Bangunan Jaya Selalu”. Pengguna diminta untuk memilih salah satu opsi dengan mengetikkan nomor (1, 2, atau 3).

- Input “1” (Login):
Program akan memulai proses login, meminta pengguna memasukkan username dan password.
- Input “2” (Registrasi):
Program akan mengarahkan ke fungsi registrasi untuk menambahkan user baru.



Gambar 18 Keluar

- Input “3” (Keluar):
Program akan mencetak pesan “Terima kasih” lalu berhenti.

- Input Tidak Valid

Jika pengguna memasukkan angka diluar 1-3 atau teks lain akan muncul pesan “Pilihan tidak valid.” dan program akan kembali meminta input dari pengguna.



Gambar 19 Menu Pengguna

Menampilkan menu pengguna program, dengan pesan “Menu Pengguna”. Pengguna diminta untuk memilih salah satu opsi dengan mengetikkan nomor (1, 2, 3, 4, atau 5).

- Input “1” (Lihat Inventaris):
Opsi untuk menampilkan daftar barang yang ada di inventaris.
- Input “2” (Tambah Barang):
Opsi untuk menambahkan item baru ke dalam inventaris pengguna.
- Input “3” (Update Barang):
Opsi untuk memperbarui informasi barang yang sudah ada di inventaris pengguna dan di barang yang pengguna tersebut tambahkan.
- Input “4” (Hapus Barang):
Opsi untuk menghapus barang tertentu dari inventaris pengguna dan di barang yang pengguna tersebut tambahkan.
- Input “5” (Logout):
Keluar dari menu ini dan kembali ke menu utama atau mengakhiri sesi pengguna.

- Jika pengguna memasukkan input tidak valid (misalnya “0” atau huruf) maka akan muncul pesan “Pilihan tidak valid.” dan program akan kembali meminta input dari pengguna.



Gambar 20 Menu Admin

Menampilkan menu admin program, dengan pesan “Menu Admin”. Admin diminta untuk memilih salah satu opsi dengan mengetikkan nomor (1, 2, 3, 4, 5, 6, 7, 8, atau 9).

- Input “1” (Lihat Inventaris):
Opsi untuk menampilkan daftar barang yang ada di inventaris.
- Input “2” (Tambah Barang):
Opsi untuk menambahkan item baru ke dalam inventaris utama atau inventaris pengguna.
- Input “3” (Update Barang):
Opsi untuk memperbarui informasi barang yang sudah ada di inventaris utama dan inventaris pengguna.
- Input “4” (Hapus Barang):
Opsi untuk menghapus barang tertentu dari inventaris utama dan inventaris pengguna .

- Input “5” (Tambah Admin Baru):
Opsinya untuk menambahkan akun admin baru ke sistem.
- Input “6” (Tampilkan User):
Opsinya untuk menampilkan daftar semua pengguna yang terdaftar dalam sistem, dipisahkan berdasarkan peran (admin dan *user* biasa).
- Input “7” (Hapus Akun User):
Opsinya untuk menghapus akun pengguna tertentu dari sistem.
- Input “8” (Lihat Riwayat Aktivitas):
Opsinya untuk menampilkan log aktivitas dalam sistem, seperti barang yang ditambahkan, dihapus, atau diperbarui, serta akun yang dibuat atau dihapus.
- Input “9” (*Logout*):
Opsinya untuk keluar dari menu admin dan kembali ke menu utama, atau mengakhiri sesi admin saat ini.

2. SS Lainnya

```

=== Register Akun User Baru ===
Masukkan username (5-20 karakter): nabila81
Masukkan password (8-20 karakter): 12345678
Pengguna nabila81 berhasil didaftarkan dengan role user!

Tekan Enter untuk melanjutkan...

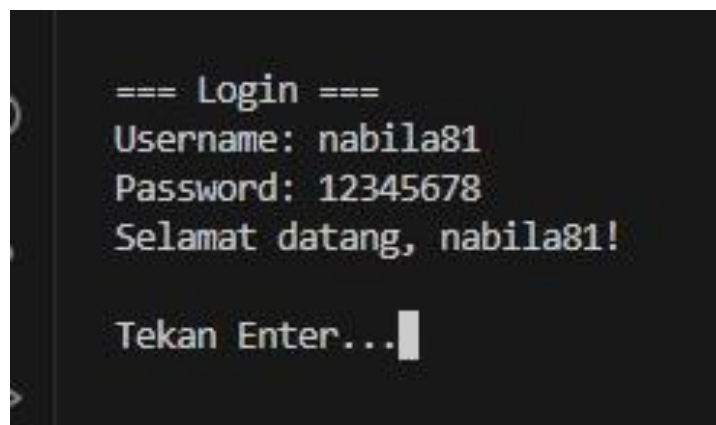
```

Gambar 21 Registrasi

Saat memilih opsi “2” di menu utama. Menampilkan pesan awal untuk memberi tahu pengguna bahwa mereka sedang berada dalam proses registrasi akun user baru.

- Pengguna diminta untuk memasukkan password sesuai dengan validasi berikut: Username harus memiliki panjang antara 5 hingga 20 karakter dan username belum digunakan oleh pengguna lain dalam sistem.
- Dalam contoh ini, username yang dimasukkan adalah "nabila81", dan validasi berhasil karena: Panjang username = 8 karakter (valid) dan username belum terdaftar dalam sistem.

- Pengguna diminta untuk memasukkan password sesuai dengan validasi berikut: Password harus memiliki panjang antara 8 hingga 20 karakter.
- Dalam contoh ini, password yang dimasukkan adalah "12345678", dan validasi berhasil karena: Panjang password = 8 karakter (valid).
- Setelah username dan password lolos validasi, pengguna berhasil didaftarkan ke sistem dengan role sebagai "user."
- Data pengguna baru (username, password, dan role) ditambahkan ke dalam database atau daftar user.
- Program meminta pengguna untuk menekan tombol Enter untuk kembali ke menu utama atau keluar dari proses registrasi.

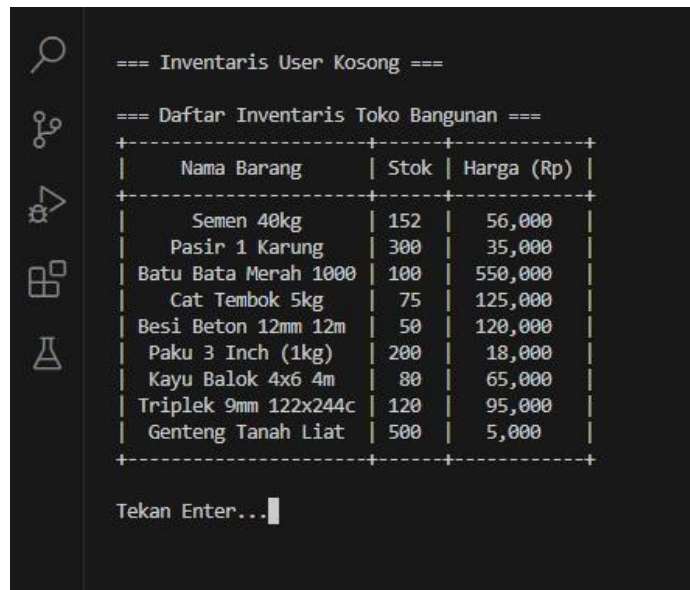


Gambar 22 Login

Saat memilih opsi "1" di menu utama. Menampilkan pesan awal untuk memberi tahu pengguna bahwa mereka sedang berada dalam proses login.

- Pengguna diminta untuk memasukkan username. Username yang dimasukkan adalah "nabila81", kemudian memeriksa apakah username ini terdaftar dalam daftar pengguna yang ada.
- Pengguna diminta untuk memasukkan password. Password yang dimasukkan adalah "12345678", kemudian memeriksa apakah password ini sesuai dengan username yang dimasukkan sebelumnya.

- Jika username dan password cocok dengan data yang ada di sistem, login berhasil dan akan ditampilkan pesan “Selamat datang, nabila81!”.
- Program meminta pengguna untuk menekan tombol Enter untuk keluar dari proses login.



```

=== Inventaris User Kosong ===

=== Daftar Inventaris Toko Bangunan ===
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
| Semen 40kg  | 152  | 56,000      |
| Pasir 1 Karung | 300  | 35,000      |
| Batu Bata Merah 1000 | 100  | 550,000     |
| Cat Tembok 5kg | 75   | 125,000     |
| Besi Beton 12mm 12m | 50   | 120,000     |
| Paku 3 Inch (1kg) | 200  | 18,000      |
| Kayu Balok 4x6 4m | 80   | 65,000      |
| Triplek 9mm 122x244c | 120  | 95,000      |
| Genteng Tanah Liat | 500  | 5,000       |
+-----+-----+-----+

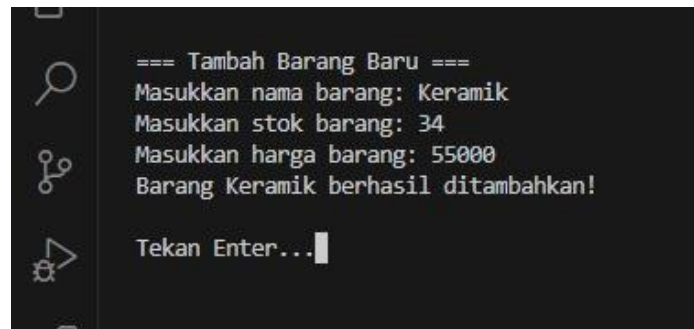
Tekan Enter...

```

Gambar 23 Lihat Inventaris (User)

Saat memilih opsi “1” di menu pengguna.

- Menampilkan pesan "=== Inventaris User Kosong ===" untuk memberi tahu pengguna bahwa tidak ada data yang dapat ditampilkan dari inventaris user.
- Menampilkan pesan "=== Daftar Inventaris Toko Bangunan ===" menandakan bahwa berikutnya adalah inventaris utama dari toko bangunan. Tabel tersebut memuat informasi detail barang yang tersedia dalam inventaris utama toko.
- Program meminta pengguna untuk menekan tombol Enter untuk kembali ke menu pengguna.



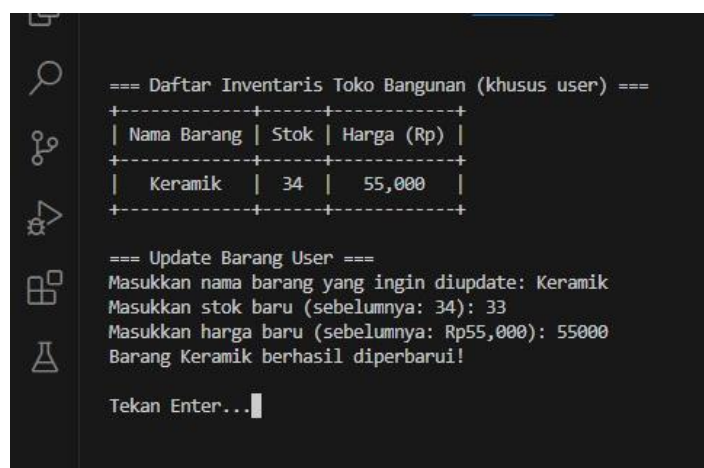
```
=== Tambah Barang Baru ===
Masukkan nama barang: Keramik
Masukkan stok barang: 34
Masukkan harga barang: 55000
Barang Keramik berhasil ditambahkan!

Tekan Enter...
```

Gambar 24 Tambah Barang (User)

Saat memilih opsi “2” di menu pengguna.

- Menampilkan pesan "=== Tambah Barang Baru ===".
- Pengguna diminta untuk memasukkan detail barang yang akan ditambahkan ke inventaris. Nama barang yang dimasukkan adalah "Keramik". Stok barang yang dimasukkan adalah 34. Harga yang dimasukkan adalah Rp55,000.
- Setelah semua data dimasukkan dan divalidasi, program menampilkan pesan bahwa barang berhasil ditambahkan ke inventaris user.
- Program meminta pengguna untuk menekan tombol Enter untuk kembali ke menu pengguna.



```
=== Daftar Inventaris Toko Bangunan (khusus user) ===
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
| Keramik    | 34   | 55,000     |
+-----+-----+-----+

=== Update Barang User ===
Masukkan nama barang yang ingin diupdate: Keramik
Masukkan stok baru (sebelumnya: 34): 33
Masukkan harga baru (sebelumnya: Rp55,000): 55000
Barang Keramik berhasil diperbarui!

Tekan Enter...
```

Gambar 25 Update Barang (User)

Saat memilih opsi “3” di menu pengguna.

- Menampilkan daftar inventaris yang dikelola khusus oleh user, bukan admin.
- Menampilkan pesan "=== Update Barang User ===".
- Pengguna diminta untuk memasukkan detail pembaruan barang:
Pengguna memasukkan nama barang "Keramik", yang sudah ada di inventaris user. Pengguna diminta untuk memasukkan stok baru untuk barang "Keramik", dalam hal ini, stok diubah dari 34 menjadi 33. Pengguna diminta untuk memasukkan harga baru untuk barang "Keramik", harga tetap Rp55,000.
- Setelah validasi data berhasil dan pembaruan disimpan, program menampilkan pesan bahwa barang berhasil diperbarui.
- Program meminta pengguna untuk menekan tombol Enter untuk kembali ke menu pengguna.

```

=== Daftar Inventaris Toko Bangunan (khusus user) ===
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
|  Keramik   |  33  |  55,000    |
+-----+-----+-----+

=== Hapus Barang User ===
Masukkan nama barang yang ingin dihapus: Keramik
Barang Keramik berhasil dihapus!

Tekan Enter...

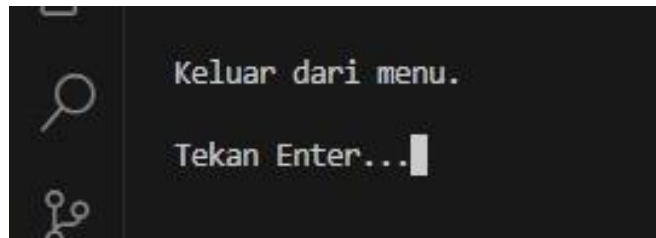
```

Gambar 26 Hapus Barang (User)

Saat memilih opsi “4” di menu pengguna.

- Menampilkan daftar inventaris yang dikelola khusus oleh user, bukan admin.
- Menampilkan pesan "=== Hapus Barang User ===".

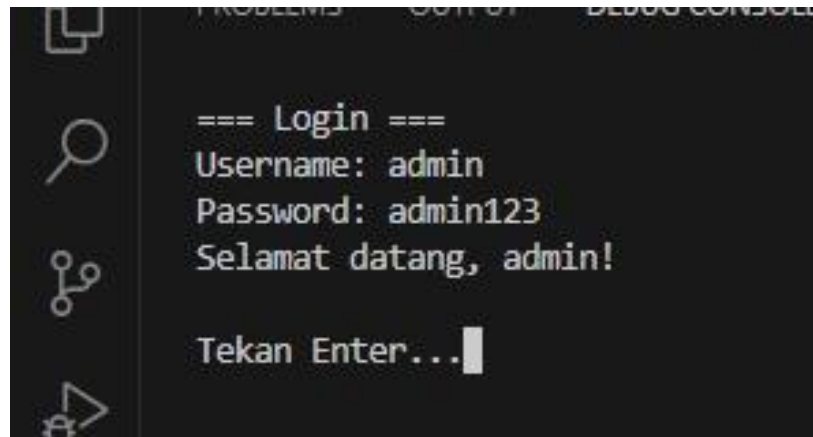
- Pengguna diminta untuk memasukkan nama barang yang ingin dihapus. Nama barang yang dimasukkan adalah "Keramik". Jika nama barang ditemukan dalam inventaris user, proses penghapusan akan dilanjutkan.
- Menampilkan bahwa barang "Keramik" telah berhasil dihapus dari inventaris user.
- Program meminta pengguna untuk menekan tombol Enter untuk kembali ke menu pengguna.



Gambar 27 keluar menu user

Saat memilih opsi “5” di menu pengguna.

- Menampilkan pesan “Keluar dari menu.” yang menandakan bahwa pengguna telah keluar dari menu pengguna.
- Program meminta pengguna untuk menekan tombol Enter untuk kembali ke menu utama



Gambar 28 Login Admin

Saat memilih opsi “1” di menu utama. Menampilkan pesan awal untuk memberi tahu pengguna bahwa mereka sedang berada dalam proses login.

- Pengguna diminta untuk memasukkan username. Username yang dimasukkan adalah "admin", kemudian memeriksa apakah username ini terdaftar dalam daftar pengguna yang ada.
- Pengguna diminta untuk memasukkan password. Password yang dimasukkan adalah "admin123", kemudian memeriksa apakah password ini sesuai dengan username yang dimasukkan sebelumnya.
- Jika username dan password cocok dengan data yang ada di sistem, login berhasil dan akan ditampilkan pesan “Selamat datang, admin!”.
- Program meminta pengguna untuk menekan tombol Enter untuk keluar dari proses login.

```

=== Daftar Inventaris Toko Bangunan ===
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
| Semen 40kg  | 152  | 56,000     |
| Pasir 1 Karung | 300  | 35,000     |
| Batu Bata Merah 1000 | 100  | 550,000    |
| Cat Tembok 5kg | 75   | 125,000    |
| Besi Beton 12mm 12m | 50   | 120,000    |
| Paku 3 Inch (1kg) | 200  | 18,000     |
| Kayu Balok 4x6 4m | 80   | 65,000     |
| Triplek 9mm 122x244c | 120  | 95,000     |
| Genteng Tanah Liat | 500  | 5,000      |
+-----+-----+-----+

=== Inventaris User Kosong ===

Tekan Enter...

```

Gambar 29 Lihat Inventaris

Saat memilih opsi “1” di menu admin.

- Menampilkan pesan "=== Daftar Inventaris Toko Bangunan ===" menandakan bahwa berikutnya adalah inventaris utama dari toko bangunan. Tabel tersebut memuat informasi detail barang yang tersedia dalam inventaris utama toko.
- Menampilkan pesan "=== Inventaris User Kosong ===" untuk memberi tahu pengguna bahwa tidak ada data yang dapat ditampilkan dari inventaris user.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.

```

=== Tambah Barang Baru ===
Tambahkan ke: 1. Inventory Utama | 2. Inventory User (Masukkan angka 1 atau 2) : 2
Masukkan nama barang: Keramik
Masukkan stok barang: 33
Masukkan harga barang: 55000
Barang 'Keramik' berhasil ditambahkan ke inventory user!

Tekan Enter...

```

Gambar 30 Tambah Barang

Saat memilih opsi “2” di menu admin.

- Menampilkan pesan "=== Tambah Barang Baru ===".
- Meminta admin untuk memilih menambahkan ke inventory utama atau inventory user. Admin memilih opsi 2, yang berarti barang akan ditambahkan ke inventaris user.
- Admin diminta untuk memasukkan detail barang yang akan ditambahkan ke inventaris. Nama barang yang dimasukkan adalah "Keramik". Stok barang yang dimasukkan adalah 33. Harga yang dimasukkan adalah Rp55,000.
- Setelah semua data dimasukkan dan divalidasi, program menampilkan pesan bahwa barang berhasil ditambahkan ke inventaris user.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.

```

===== Daftar Inventaris Toko Bangunan =====
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
| Semen 40kg  | 152  | 56,000      |
| Pasir 1 Karung | 300  | 35,000      |
| Batu Bata Merah 1000 | 100  | 550,000     |
| Cat Tembok 5kg | 75   | 125,000     |
| Besi Beton 12mm 12m | 50   | 120,000     |
| Paku 3 Inch (1kg) | 200  | 18,000      |
| Kayu Balok 4x6 4m | 80   | 65,000      |
| Triplek 9mm 122x244c | 120  | 95,000      |
| Genteng Tanah Liat | 500  | 5,000       |
+-----+-----+-----+

===== Inventaris User Kosong =====

===== Update Barang =====
Masukkan nama barang yang ingin diupdate: Semen 40kg
Masukkan stok baru (sebelumnya: 152): 151
Masukkan harga baru (sebelumnya: Rp56,000): 56000
Barang Semen 40kg berhasil diperbarui!

Tekan Enter...

```

Gambar 31 Update Barang

Saat memilih opsi “3” di menu admin.

- Menampilkan seluruh daftar inventaris.
- Menampilkan pesan "=== Update Barang User ===".

- Admin diminta untuk memasukkan detail pembaruan barang:
Pengguna memasukkan nama barang "Semen 40kg", yang sudah ada di inventaris user. Pengguna diminta untuk memasukkan stok baru untuk barang "Keramik", dalam hal ini, stok diubah dari 152 menjadi 151. Pengguna diminta untuk memasukkan harga baru untuk barang "Keramik", harga tetap Rp56,000.
- Setelah validasi data berhasil dan pembaruan disimpan, program menampilkan pesan bahwa barang berhasil diperbarui.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.

```

=== Daftar Inventaris Toko Bangunan ===
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
| Semen 40kg  | 152  | 56,000      |
| Pasir 1 Karung | 300  | 35,000      |
| Batu Bata Merah 1000 | 100  | 550,000     |
| Cat Tembok 5kg | 75   | 125,000     |
| Besi Beton 12mm 12m | 50   | 120,000     |
| Paku 3 Inch (1kg) | 200  | 18,000      |
| Kayu Balok 4x6 4m | 80   | 65,000      |
| Triplek 9mm 122x244c | 120  | 95,000      |
| Genteng Tanah Liat | 500  | 5,000       |
+-----+-----+-----+

=== Daftar Inventaris Toko Bangunan (khusus user) ===
+-----+-----+-----+
| Nama Barang | Stok | Harga (Rp) |
+-----+-----+-----+
| Keramik    | 33   | 55,000      |
+-----+-----+-----+

=== Hapus Barang ===
Masukkan nama barang yang ingin dihapus: Keramik
Barang Keramik berhasil dihapus!

Tekan Enter...

```

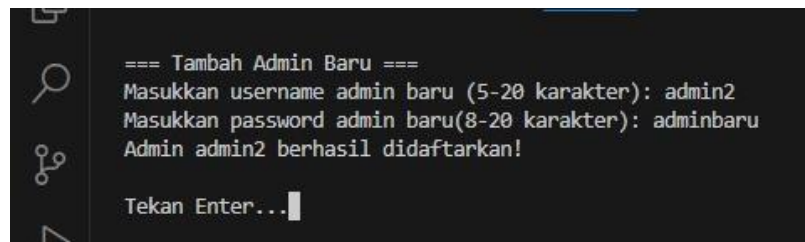
Gambar 32 Hapus Barang

Saat memilih opsi “4” di menu admin.

- Menampilkan seluruh daftar inventaris.
- Menampilkan pesan "=== Hapus Barang User ===".
- Admin diminta untuk memasukkan nama barang yang ingin dihapus. Nama barang yang dimasukkan adalah "Keramik". Jika nama barang

ditemukan dalam seluruh daftar inventaris, proses penghapusan akan dilanjutkan.

- Menampilkan bahwa barang "Keramik" telah berhasil dihapus.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.



Gambar 33 Tambah Admin Baru

Saat memilih opsi "5" di menu admin.

- Menampilkan pesan "=== Tambah Admin Baru ===".
- Admin diminta untuk memasukkan detail akun admin baru: Username yang dimasukkan adalah "admin2" (5-20 karakter dan tidak boleh kosong). Password yang dimasukkan adalah "adminbaru" (8-20 karakter dan tidak boleh kosong).
- Menampilkan bahwa akun admin dengan username "admin2" berhasil ditambahkan ke sistem dan disimpan dengan role sebagai "admin".
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.



```
=== Daftar Pengguna Terdaftar ===

-- Admin --
+-----+-----+
| Username | Role |
+-----+-----+
| admin    | admin |
| admin2   | admin |
+-----+-----+

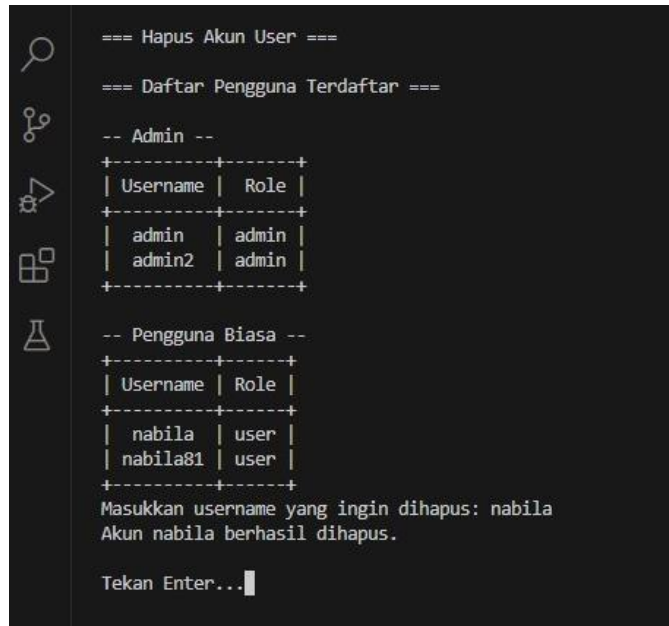
-- Pengguna Biasa --
+-----+-----+
| Username | Role |
+-----+-----+
| nabila   | user  |
| nabila81 | user  |
+-----+-----+

Tekan Enter...
```

Gambar 34 Tampilkan User

Saat memilih opsi “6” di menu admin.

- Menampilkan pesan "=== Daftar Pengguna Terdaftar ===".
- Label "-- Admin --" menunjukkan bahwa tabel di bawah ini berisi pengguna dengan peran admin. Tabel menunjukkan dua admin terdaftar.
- Label "-- Pengguna Biasa --" menunjukkan bahwa tabel ini berisi pengguna dengan peran user. Tabel menunjukkan dua pengguna biasa terdaftar.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.



```
=== Hapus Akun User ===

=== Daftar Pengguna Terdaftar ===

-- Admin --
+-----+-----+
| Username | Role |
+-----+-----+
| admin    | admin |
| admin2   | admin |
+-----+-----+

-- Pengguna Biasa --
+-----+-----+
| Username | Role |
+-----+-----+
| nabila   | user  |
| nabila81 | user  |
+-----+-----+

Masukkan username yang ingin dihapus: nabila
Akun nabila berhasil dihapus.

Tekan Enter...|
```

Gambar 35 Hapus Akun User

Saat memilih opsi “7” di menu admin.

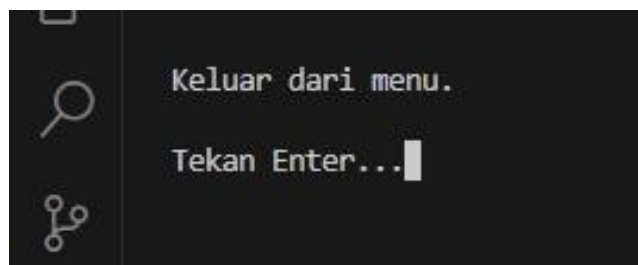
- Menampilkan pesan "=== Hapus Akun User ===".
- Menampilkan seluruh daftar pengguna.
- Admin diminta untuk memasukkan username akun yang ingin dihapus dari daftar pengguna. Dalam contoh ini, "nabila" dipilih sebagai username yang akan dihapus.
- Setelah akun "nabila" ditemukan, program berhasil menghapusnya dari daftar pengguna.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.



Gambar 36 Lihat Riwayat Aktivitas

Saat memilih opsi “8” di menu admin.

- Menampilkan pesan "=== Riwayat Aktivitas ===".
- Menampilkan timestamp atau waktu terjadinya aktivitas dalam format YYYY-MM-DD HH:MM:SS.
- Menampilkan deskripsi aktivitas yang terjadi di sistem, seperti: Penambahan atau pembaruan barang, penghapusan barang atau akun.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu admin.



Gambar 37 Keluar Menu Admin

Saat memilih opsi “9” di menu admin.

- Menampilkan pesan “Keluar dari menu.” yang menandakan bahwa admin telah keluar dari menu admin.
- Program meminta admin untuk menekan tombol Enter untuk kembali ke menu utama

3.2 Source Code

```
import csv # Untuk membaca dan menulis file dalam bentuk
```

```

csv
import datetime    # Untuk menambahkan cap waktu di riwayat
                    # aktivitas
import os          # untuk menjalankan perintah sistem operasi
                    # seperti clear dan cls
import platform    # untuk mendapatkan informasi tentang platform
                    # yang sedang digunakan.
from prettytable import PrettyTable # Untuk membuat tabel pada
                    # tampilan

# Data awal inventaris toko bangunan dalam dictionary
inventory = {
    "Semen 40kg": {"stok": 150, "harga": 52000, "added_by":
"admin"},
    "Pasir 1 Karung": {"stok": 300, "harga": 35000,
"added_by": "admin"},
    "Batu Bata Merah 1000": {"stok": 100, "harga": 550000,
"added_by": "admin"},
    "Cat Tembok 5kg": {"stok": 75, "harga": 125000,
"added_by": "admin"},
    "Besi Beton 12mm 12m": {"stok": 50, "harga": 120000,
"added_by": "admin"},
    "Paku 3 Inch (1kg)": {"stok": 200, "harga": 18000,
"added_by": "admin"},
    "Kayu Balok 4x6 4m": {"stok": 80, "harga": 65000,
"added_by": "admin"},
    "Triplek 9mm 122x244c": {"stok": 120, "harga": 95000,
"added_by": "admin"},
    "Genteng Tanah Liat": {"stok": 500, "harga": 5000,
"added_by": "admin"}

}

inventory_user = {}    # Inventaris khusus untuk user

# Variabel users dengan satu akun admin bawaan
users = [
    {"username": "admin", "password": "admin123", "role":
"admin"}
]

riwayat_aktivitas = [] # Menyimpan riwayat aktivitas

# Prosedur untuk menyimpan inventaris ke file CSV
def simpan_inventory(filename, inventory):

```

```

        try:
            with open(filename, "w", newline='', encoding='utf-8') as
file:
                fieldnames = ["item", "stok", "harga", "added_by"]
                writer = csv.DictWriter(file, fieldnames=fieldnames)
                writer.writeheader()

                # Menulis setiap barang ke file
                for item, data in inventory.items():
                    row = {"item" : item, "stok" : data['stok'],
"harga" : data["harga"], "added_by" : data["added_by"]}
                    writer.writerow(row)
        except Exception as e:
            print(f"Terjadi kesalahan saat menyimpan inventaris:
{e}")

# Fungsi untuk memuat inventaris dari file CSV
def muat_inventory(filename):
    inventory = {}
    try:
        with open(filename, "r", encoding='utf-8') as filename:
            csvdata = csv.DictReader(filename)
            for row in csvdata:
                try:
                    inventory[row["item"]] = {
                        "stok" : int(row["stok"]),
                        "harga" : int(row["harga"]),
                        "added_by" : row["added_by"]
                    }
                except (ValueError, KeyError) as e:
                    print(f"Kesalahan membaca baris: {row} -
{e}")
            except FileNotFoundError:
                print(f"File {filename} tidak ditemukan. Membuat data
inventaris baru.")
            except Exception as e:
                print(f"Kesalahan umum saat memuat inventaris: {e}")
            return inventory

# Prosedur untuk menyimpan inventaris user ke file CSV
def simpan_inventory_user(filename, inventory_user):
    try:
        with open(filename, "w", newline='', encoding='utf-8') as
file:
            fieldnames = ["item", "stok", "harga", "added_by"]

```



```

        writer = csv.DictWriter(file, fieldnames=fieldnames)
        writer.writeheader()

        # Menulis setiap barang user ke file
        for item, data in inventory_user.items():
            row = {"item" : item, "stok" : data["stok"],
                "harga" : data["harga"], "added_by" : data["added_by"]}
            writer.writerow(row)
        except Exception as e:
            print(f"Terjadi kesalahan saat menyimpan inventaris user: {e}")

# Fungsi untuk memuat inventaris user dari file CSV
def muat_inventory_user(filename):
    inventory_user = {}
    try:
        with open(filename, "r", encoding='utf-8') as filename:
            csvdata = csv.DictReader(filename)
            for row in csvdata:
                try:
                    inventory_user[row["item"]] = {
                        "stok" : int(row["stok"]),
                        "harga" : int(row["harga"]),
                        "added_by" : row["added_by"]
                    }
                except (ValueError, KeyError) as e:
                    print(f"Kesalahan membaca baris: {row} - {e}")
            except FileNotFoundError:
                print(f"File {filename} tidak ditemukan. Membuat data inventaris baru.")
            except Exception as e:
                print(f"Kesalahan umum saat memuat inventaris user: {e}")
    return inventory_user

# Prosedur untuk menyimpan data user ke file CSV
def simpan_users(filename, users):
    try:
        with open(filename, "w", newline='', encoding='utf-8') as file:
            fieldnames = ["username", "password", "role"]
            writer = csv.DictWriter(file, fieldnames=fieldnames)
            writer.writeheader()

            # Menulis setiap user ke file

```

```

        for user in users:
            writer.writerow(user)
    except Exception as e:
        print(f"Terjadi kesalahan saat menyimpan data pengguna: {e}")

# Fungsi untuk memuat data user dari file CSV
def muat_users(filename):
    users = []
    try:
        with open(filename, "r", encoding='utf-8') as filename:
            csvdata = csv.DictReader(filename)
            for row in csvdata:
                try:
                    users.append({
                        "username" : row["username"],
                        "password" : row["password"],
                        "role" : row["role"]
                    })
                except KeyError as e:
                    print(f"Kesalahan membaca data pengguna:{e}")
    except FileNotFoundError:
        print(f"File {filename} tidak ditemukan. Membuat data pengguna baru.")
    except Exception as e:
        print(f"Kesalahan umum saat memuat data pengguna:{e}")
    return users

# Prosedur untuk menyimpan riwayat aktivitas ke file CSV
def simpan_riwayat_aktivitas(filename, riwayat_aktivitas):
    try:
        with open(filename, "w", newline='', encoding='utf-8') as file:
            writer = csv.writer(file)
            writer.writerow(["waktu", "aktivitas"])

            # Menulis setiap riwayat aktivitas ke file
            for aktivitas in riwayat_aktivitas:
                writer.writerow(aktivitas)
    except Exception as e:
        print(f"Terjadi kesalahan saat menyimpan riwayat aktivitas:{e}")

# Fungsi untuk memuat riwayat aktivitas dari file CSV
def muat_riwayat_aktivitas(filename):

```

```

riwayat_aktivitas = []
try:
    with open(filename, "r", encoding='utf-8') as filename:
        csvdata = csv.reader(filename)
        next(csvdata) # Lewati header
        for row in csvdata:
            riwayat_aktivitas.append(row)
except FileNotFoundError:
    print(f"File {filename} tidak ditemukan. Membuat riwayat
    aktivitas baru.")
except Exception as e:
    print(f"Kesalahan umum saat memuat riwayat aktivitas:
    {e}")
return riwayat_aktivitas

# Memuat data awal dari file CSV
inventory = muat_inventory("inventory.csv")
inventory_user = muat_inventory_user("inventory_user.csv")
users = muat_users("users.csv")
riwayat_aktivitas =
muat_riwayat_aktivitas("riwayat_aktivitas.csv")

# Prosedur untuk menyimpan semua data ke file CSV
def simpan_data():
    simpan_inventory("inventory.csv", inventory)
    simpan_inventory_user("inventory_user.csv", inventory_user)
    simpan_users("users.csv", users)
    simpan_riwayat_aktivitas("riwayat_aktivitas.csv",
    riwayat_aktivitas)

# Prosedur untuk menclear terminal
def clear_terminal():
    if platform.system() == "Windows":
        os.system("cls") # Perintah untuk Windows
    else:
        os.system("clear") # Perintah untuk Linux dan macOS

# Prosedur untuk menambahkan cap waktu pada aktivitas
def tambah_riwayat(aktivitas):
    try:
        waktu = datetime.datetime.now().strftime("%Y-%m-%d
        %H:%M:%S")
        riwayat_aktivitas.append([waktu, aktivitas])
        simpan_riwayat_aktivitas("riwayat_aktivitas.csv",
        riwayat_aktivitas)

```



```

# Prosedur untuk registrasi akun user baru
def registrasi_pengguna(users):
    try:
        print("=== Register Akun User Baru ===")
        while True:
            # Input username dengan validasi
            username = input("Masukkan username (5-20 karakter):").strip()

            if not username:
                print("Error: Username tidak boleh kosong.Silakan coba lagi.")
                continue

            if len(username) < 5 or len(username) > 20:
                print("Error: Username harus memiliki panjang 5-20 karakter.")
                continue

            # Cek apakah username sudah ada
            user_exists = any(user['username'] == username for user in users)
            if user_exists:
                print("Username sudah ada. Silakan coba dengan username lain.")
                continue

            break # Username valid dan unik

        while True:
            # Input password dengan validasi
            password = input("Masukkan password (8-20 karakter):").strip()

            if not password:
                print("Error: Password tidak boleh kosong.Silakan coba lagi.")
                continue

            if len(password) < 8 or len(password) > 20:
                print("Error: Password harus memiliki panjang 8-20 karakter.")
                continue

            break # Password valid

```

```

        role = "user" # Semua registrasi pengguna biasa sebagai
"user"
        users.append({"username": username, "password": password,
"role": role})
        print(f"Pengguna {username} berhasil didaftarkan dengan
role {role}!")

        simpan_data() # Simpan data setelah registrasi

    except Exception as e:
        print(f"Terjadi kesalahan saat registrasi pengguna: {e}")

# Fungsi khusus untuk admin menambah akun admin lain
def tambah_admin(users, current_user):
    try:
        if current_user['role'] != "admin":
            print("Hanya admin yang bisa menambahkan admin
baru.")
            return

        print("=== Tambah Admin Baru ===")

        # Validasi username
        while True:
            username = input("Masukkan username admin baru (5-20
karakter): ").strip()

            if not username:
                print("Error: Username tidak boleh kosong.
Silakan coba lagi")
                continue

            if len(username) < 5 or len(username) > 20:
                print("Error: Username harus memiliki panjang 5-
20 karakter.")
                continue

            # Cek apakah username sudah ada
            user_exists = any(user['username'] == username for
user in users)
            if user_exists:
                print("Username sudah ada. Silakan coba dengan
username lain.")
                continue

```

```

        break # Username valid dan unik

    # Validasi password
    while True:
        password = input("Masukkan password admin baru(8-20
karakter): ").strip()

        if not password:
            print("Error: Password tidak boleh kosong.
Silakan coba lagi.")
            continue

        if len(password) < 8 or len(password) > 20:
            print("Error: Password harus memiliki panjang 8-
20 karakter.")
            continue

        break # Password valid

    # Menambahkan admin baru
    users.append({"username": username, "password": password,
"role": "admin"})
    tambah_riwayat(f"Admin baru '{username}' ditambahkan oleh
'{current_user['username']}'")
    print(f"Admin {username} berhasil didaftarkan!")

    simpan_data() # Simpan perubahan

except Exception as e:
    print(f"Terjadi kesalahan saat menambahkan admin: {e}")

# Fungsi untuk menampilkan inventaris
def tampilkan_inventaris(inventory):
    try:
        if not inventory:
            print("\n=== Inventaris Kosong ===")
            return # Tidak perlu lanjut jika inventaris kosong

        table = PrettyTable()
        table.field_names = ["Nama Barang", "Stok", "Harga (Rp)"]
        for nama_barang, detail in inventory.items():
            try:
                table.add_row([
                    nama_barang,

```

```

        detail['stok'],
        f"{detail['harga']:,}"
    ])
    except KeyError as ke:
        print(f>Data barang '{nama_barang}' tidak
lengkap: {ke}")
        continue

    print("\n=== Daftar Inventaris Toko Bangunan ===")
    print(table)
    except Exception as e:
        print(f"Terjadi kesalahan saat menampilkan inventaris:
{e}")

# Fungsi untuk menampilkan inventaris user
def tampilkan_inventaris_user(inventory):
    try:
        if not inventory:
            print("\n=== Inventaris User Kosong ===")
            return # Tidak perlu lanjut jika inventaris kosong

        table = PrettyTable() # Untuk menggunakan pretty table
        table.field_names = ["Nama Barang", "Stok", "Harga (Rp)"]
        for nama_barang, detail in inventory.items():
            try:
                table.add_row([
                    nama_barang,
                    detail['stok'],
                    f"{detail['harga']:,}"
                ])
            except KeyError as ke:
                print(f>Data barang '{nama_barang}' tidak
lengkap: {ke}")
                continue

        print("\n=== Daftar Inventaris Toko Bangunan (khusus
user) ===")
        print(table)
    except Exception as e:
        print(f"Terjadi kesalahan saat menampilkan inventaris
user: {e}")

# Fungsi untuk menangani pilihan user dalam menu
def menu_user(inventory_user, inventory, current_user):
    while True:

```



```

try:
    clear_terminal()
    print("""

    MENU PENGGUNA

    ☆☆☆

    1. Lihat Inventaris
    2. Tambah Barang
    3. Update Barang
    4. Hapus Barang
    5. Logout

""")

    pilihan_user = input("Pilih menu: ").strip()
    if pilihan_user == "1":
        clear_terminal()
        tampilkan_inventaris_user(inventory_user)
        tampilkan_inventaris(inventory)
        input("\nTekan Enter...")

    elif pilihan_user == "2":
        clear_terminal()
        tambah_barang_user(inventory_user, current_user)
        input("\nTekan Enter...")

    elif pilihan_user == "3":
        clear_terminal()
        update_barang_user(inventory_user, current_user)
        input("\nTekan Enter...")

    elif pilihan_user == "4":
        clear_terminal()
        hapus_barang_user(inventory_user, current_user)
        input("\nTekan Enter...")

    elif pilihan_user == "5":
        clear_terminal()
        print("Keluar dari menu.")
        input("\nTekan Enter...")
        return "Logout"

    else:
        print("Pilihan tidak valid. Silakan coba lagi.")
        input("\nTekan Enter...")

```

```

        except Exception as e:
            print(f"Terjadi kesalahan dalam menu: {e}")
            input("\nTekan enter untuk kembali ke menu...")

# Fungsi untuk menambah inventaris user
def tambah_barang_user(inventory_user, current_user):
    try:
        print("=== Tambah Barang Baru ===")
        nama_barang = input("Masukkan nama barang: ").strip()
        if not nama_barang:
            print("Nama barang tidak boleh kosong.")
            return

        try:
            stok = int(input("Masukkan stok barang: "))
            if stok < 0:
                print("Stok tidak boleh negatif. Silakan coba
lagi.")
                return

            harga = int(input("Masukkan harga barang: "))
            if harga < 0:
                print("Harga tidak boleh negatif. Silakan coba
lagi.")
                return
        except ValueError:
            print("Stok dan harga harus berupa angka. Silakan
coba lagi.")
            return

        # Menambahkan barang ke inventaris
        inventory_user[nama_barang] = {"stok": stok, "harga":
harga, "added_by": current_user['username']}

        # Menambah riwayat aktivitas
        tambah_riwayat(f"Barang baru '{nama_barang}' ditambahkan
oleh {current_user['username']} dengan stok {stok} dan harga
Rp{harga:,}")

        print(f"Barang {nama_barang} berhasil ditambahkan!")

        # Simpan data setelah penambahan
        simpan_data()

    except Exception as e:

```

```

        print(f"Terjadi kesalahan saat menambah barang: {e}")

# Fungsi untuk meng-update inventaris user
def update_barang_user(inventory_user, current_user):
    try:
        tampilkan_inventaris_user(inventory_user)
        print("\n=== Update Barang User ===")
        nama_barang = input("Masukkan nama barang yang ingin
diupdate: ").strip()

        if not nama_barang:
            print("Nama barang tidak boleh kosong.")
            return

        if nama_barang in inventory_user and
inventory_user[nama_barang]['added_by'] ==
current_user['username']:
            stok_lama = inventory_user[nama_barang]["stok"]
            harga_lama = inventory_user[nama_barang]["harga"]

            try:
                stok_baru = int(input(f"Masukkan stok baru
(sebelumnya: {stok_lama}): "))
                if stok_baru < 0:
                    print("Stok tidak boleh negatif. Silakan coba
lagi.")
                    return

                harga_baru = int(input(f"Masukkan harga baru
(sebelumnya: Rp{harga_lama:,}): "))
                if harga_baru < 0:
                    print("Harga tidak boleh negatif. Silakan
coba lagi.")
                    return

            except ValueError:
                print("Stok dan harga harus berupa angka. Silakan
coba lagi.")
                return

        # Update data barang
        inventory_user[nama_barang]["stok"] = stok_baru
        inventory_user[nama_barang]["harga"] = harga_baru
        tambah_riwayat(f"Barang '{nama_barang}' diperbarui
oleh {current_user['username']} dari stok {stok_lama} ke

```

```

{stok_baru} dan harga Rp{harga_lama:,} ke Rp{harga_baru:,}")
    print(f"Barang {nama_barang} berhasil diperbarui!")

    else:
        print("Anda hanya bisa mengupdate barang yang Anda
tambahkan sendiri.")

    # Simpan data setelah update
    simpan_data()

except Exception as e:
    print(f"Terjadi kesalahan saat memperbarui barang: {e}")

# Fungsi untuk menghapus inventaris user
def hapus_barang_user(inventory_user, current_user):
    try:
        tampilkan_inventaris_user(inventory_user)
        print("\n=== Hapus Barang User ===")
        nama_barang = input("Masukkan nama barang yang ingin
dihapus: ").strip()

        if not nama_barang:
            print("Nama barang tidak boleh kosong.")
            return

        if nama_barang in inventory_user and
inventory_user[nama_barang]['added_by'] ==
current_user['username']:
            stok_barang = inventory_user[nama_barang]["stok"]
            del inventory_user[nama_barang]
            tambah_riwayat(f"Barang '{nama_barang}' dihapus oleh
{current_user['username']} dengan stok {stok_barang} dihapus dari
inventaris.")
            print(f"Barang {nama_barang} berhasil dihapus!")
        else:
            print("Anda hanya bisa menghapus barang yang Anda
tambahkan sendiri.")
            simpan_data() # Memanggil fungsi simpan_data() untuk
menyimpan data setelah dihapus
    except Exception as e:
        print(f"Terjadi kesalahan saat menghapus barang: {e}")

# Prosedur untuk menangani pilihan admin dalam menu
def menu_admin(inventory, inventory_user, users, current_user):
    while True:

```

```

try:
    clear_terminal()
    print("""

    ACU ADMIN

    ☆☆☆

    1. Lihat Inventaris
    2. Tambah Barang
    3. Update Barang
    4. Hapus Barang
    5. Tambah Admin Baru
    6. Tampilkan User
    7. Hapus akun user
    8. Lihat Riwayat Aktivitas
    9. Logout

    """)

    pilihan_admin = input("Pilih menu: ").strip()

    if pilihan_admin == "1":
        clear_terminal()
        tampilkan_inventaris(inventory)
        tampilkan_inventaris_user(inventory_user)
        input("\nTekan Enter...")

    elif pilihan_admin == "2":
        clear_terminal()
        tambah_barang(inventory, inventory_user,
current_user)
        input("\nTekan Enter...")

    elif pilihan_admin == "3":
        clear_terminal()
        update_barang(inventory, inventory_user,
current_user)
        input("\nTekan Enter...")

    elif pilihan_admin == "4":
        clear_terminal()
        hapus_barang(inventory, inventory_user,
current_user)
        input("\nTekan Enter...")

    elif pilihan_admin == "5":
        clear_terminal()

```

```

        tambah_admin(users, current_user)
        input("\nTekan Enter...")

    elif pilihan_admin == "6":
        clear_terminal()
        tampilkan_users(users)
        input("\nTekan Enter...")

    elif pilihan_admin == "7":
        clear_terminal()
        hapus_user(users, current_user)
        input("\nTekan Enter...")

    elif pilihan_admin == "8":
        clear_terminal()
        tampilkan_riwayat_aktivitas()
        input("\nTekan Enter...")

    elif pilihan_admin == "9":
        clear_terminal()
        print("Keluar dari menu.")
        input("\nTekan Enter...")
        break

    else:
        print("Pilihan tidak valid. Silakan coba lagi.")
        input("\nTekan Enter...")
except Exception as e:
    print(f"Terjadi kesalahan dalam menu: {e}")
    input("Tekan enter untuk kembali ke menu...")

# Fungsi untuk menambah barang inventaris
def tambah_barang(inventory, inventory_user, current_user):
    if current_user['role'] != "admin":
        print("Hanya admin yang bisa menambahkan barang ke inventaris.")
        return

    print("=== Tambah Barang Baru ===")
    pilihan = input("Tambahkan ke: 1. Inventory Utama | 2. Inventory User (Masukkan angka 1 atau 2) : ").strip()

    try:
        if pilihan == "1": # Fungsi untuk menambahkan barang di inventory utama

```

```

        nama_barang = input("Masukkan nama barang: ").strip()
        if not nama_barang:
            print("Nama barang tidak boleh kosong.")
            return

        try:
            stok = int(input("Masukkan stok barang: "))
            if stok < 0:
                print("Stok tidak boleh negatif. Silakan coba
lagi.")
                return
        except ValueError:
            print("Stok harus berupa angka. Silakan coba
lagi.")
            return

        try:
            harga = int(input("Masukkan harga barang: "))
            if harga < 0:
                print("Harga tidak boleh negatif. Silakan
coba lagi.")
                return
        except ValueError:
            print("Harga harus berupa angka. Silakan coba
lagi.")
            return

        if nama_barang in inventory:
            print("Barang sudah ada di inventory utama,
gunakan update barang untuk mengubah datanya.")
        else:
            inventory[nama_barang] = {"stok": stok, "harga":
harga, "added_by": "admin"}
            tambah_riwayat(f"Admin
'{current_user['username']}' menambahkan barang '{nama_barang}'
ke inventory utama dengan stok {stok} dan harga Rp{harga:,}.")
            print(f"Barang '{nama_barang}' berhasil
ditambahkan ke inventory utama!")

        elif pilihan == "2": # Fungsi untuk menambahkan barang
di inventory user
            nama_barang = input("Masukkan nama barang: ").strip()
            try:
                stok = int(input("Masukkan stok barang: "))
                if stok < 0:

```

```

        print("Stok tidak boleh negatif. Silakan coba
lagi.")
        return
    except ValueError:
        print("Stok harus berupa angka. Silakan coba
lagi.")
        return

    try:
        harga = int(input("Masukkan harga barang: "))
        if harga < 0:
            print("Harga tidak boleh negatif. Silakan
coba lagi.")
            return
    except ValueError:
        print("Harga harus berupa angka. Silakan coba
lagi.")
        return

    if nama_barang in inventory_user:
        print("Barang sudah ada di inventory user,
gunakan update barang untuk mengubah datanya.")
    else:
        inventory_user[nama_barang] = {"stok": stok,
"harga": harga, "added_by": "admin"}
        tambah_riwayat(f"Admin
'{current_user['username']}' menambahkan barang '{nama_barang}'
ke inventory user dengan stok {stok} dan harga Rp{harga:,}.")
        print(f"Barang '{nama_barang}' berhasil
ditambahkan ke inventory user!")

    else:
        print("Pilihan tidak valid. Silakan coba lagi.")
    except ValueError:
        print("Input tidak valid. Pastikan semua input berupa
angka yang benar.")

    simpan_data() # Memanggil fungsi simpan_data() untuk
menyimpan data setelah ditambah

# Fungsi untuk mengupdate barang
def update_barang(inventory, inventory_user, current_user):
    tampilkan_inventaris(inventory)
    tampilkan_inventaris_user(inventory_user)
    print("\n=== Update Barang ===")

```



```

    nama_barang = input("Masukkan nama barang yang ingin
diupdate: ").strip()
    if not nama_barang:
        print("Nama barang tidak boleh kosong.")
        return

    try:
        if nama_barang in inventory: # Untuk mengupdate barang
di inventory utama
            stok_lama = inventory[nama_barang]["stok"]
            harga_lama = inventory[nama_barang]["harga"]

            try:
                stok_baru = int(input(f"Masukkan stok baru
(sebelumnya: {stok_lama}): "))
                if stok_baru < 0:
                    print("Stok tidak boleh negatif. Silakan coba
lagi.")
                    return

                harga_baru = int(input(f"Masukkan harga baru
(sebelumnya: Rp{harga_lama:,}): "))
                if harga_baru < 0:
                    print("Harga tidak boleh negatif. Silakan
coba lagi.")
                    return

            except ValueError:
                print("Stok dan harga harus berupa angka. Silakan
coba lagi.")
                return

            inventory[nama_barang]["stok"] = stok_baru
            inventory[nama_barang]["harga"] = harga_baru
            tambah_riwayat(f"Barang '{nama_barang}' diperbarui
oleh {current_user['username']} dari stok {stok_lama} ke
{stok_baru} dan harga Rp{harga_lama:,} ke Rp{harga_baru:,}")
            print(f"Barang {nama_barang} berhasil diperbarui!")

        elif nama_barang in inventory_user: # Untuk mengupdate
barang di inventory user
            if current_user['role'] == 'admin' or
inventory_user[nama_barang]["added_by"] ==
current_user["username"]:
                stok_lama = inventory_user[nama_barang]["stok"]

```

```

        harga_lama = inventory_user[nama_barang]["harga"]

        try:
            stok_baru = int(input(f"Masukkan stok baru
(sebelumnya: {stok_lama}): "))
            if stok_baru < 0:
                print("Stok tidak boleh negatif. Silakan
coba lagi.")
                return

            harga_baru = int(input(f"Masukkan harga baru
(sebelumnya: Rp{harga_lama:,}): "))
            if harga_baru < 0:
                print("Harga tidak boleh negatif. Silakan
coba lagi.")
                return

        except ValueError:
            print("Stok dan harga harus berupa angka.
Silakan coba lagi.")
            return

        inventory_user[nama_barang]["stok"] = stok_baru
        inventory_user[nama_barang]["harga"] = harga_baru
        tambah_riwayat(f"Barang '{nama_barang}'
diperbarui oleh {current_user['username']} dari stok {stok_lama}
ke {stok_baru} dan harga Rp{harga_lama:,} ke Rp{harga_baru:,}")
        print(f"Barang {nama_barang} berhasil
diperbarui!")

    else:
        print("Anda tidak memiliki izin untuk mengupdate
barang ini.")

    else:
        print("Nama barang tidak ditemukan.")
    except ValueError:
        print("Terjadi kesalahan saat mengupdate barang. Pastikan
stok dan harga berupa angka yang valid.")

    simpan_data() # Memanggil fungsi simpan_data() untuk
menyimpan data setelah di-update

# Fungsi untuk menghapus barang inventaris

```

```

def hapus_barang(inventory, inventory_user, current_user):
    tampilkan_inventaris(inventory)
    tampilkan_inventaris_user(inventory_user)
    print("\n=== Hapus Barang ===")
    nama_barang = input("Masukkan nama barang yang ingin dihapus: ").strip()
    if not nama_barang:
        print("Nama barang tidak boleh kosong.")
        return

    try:
        if nama_barang in inventory: # Untuk menghapus barang di inventory utama
            stok_barang = inventory[nama_barang]["stok"]
            del inventory[nama_barang]
            tambah_riwayat(f"Barang '{nama_barang}' dengan stok {stok_barang} dihapus dari inventaris.")
            print(f"Barang {nama_barang} berhasil dihapus!")

            elif nama_barang in inventory_user: # Untuk menghapus barang di inventory user
                if current_user['role'] == 'admin' or inventory_user[nama_barang]["added_by"] == current_user["username"]:
                    del inventory_user[nama_barang]
                    tambah_riwayat(f"Barang '{nama_barang}' dihapus oleh {current_user['username']} dari inventory pengguna.")
                    print(f"Barang {nama_barang} berhasil dihapus!")
                else:
                    print("Anda tidak memiliki izin untuk menghapus barang ini.")

            else:
                print("Nama barang tidak ditemukan.")
    except Exception as e:
        print(f"Terjadi kesalahan: {str(e)}")

    simpan_data() # Memanggil fungsi simpan_data() untuk menyimpan data setelah dihapus

# Fungsi untuk menghapus akun user
def hapus_user(users, current_user):
    if current_user['role'] != "admin":
        print("Hanya admin yang bisa menghapus akun user.")
    return

```

```

print("=== Hapus Akun User ===")
tampilkan_users(users)
username = input("Masukkan username yang ingin dihapus:
").strip()

if not username:
    print("Username tidak boleh kosong.")
    return

# Cek apakah username yang ingin dihapus ada di daftar user
user_to_delete = None
try:
    for user in users:
        if user['username'] == username:
            user_to_delete = user
            break

    if user_to_delete:
        if user_to_delete['role'] == "admin":
            print("Anda tidak dapat menghapus akun admin
lain.")
        else:
            users.remove(user_to_delete)
            tambah_riwayat(f"Akun '{username}' dihapus oleh
admin")
            print(f"Akun {username} berhasil dihapus.")
    else:
        print("Username tidak ditemukan.")
except Exception as e:
    print(f"Terjadi kesalahan saat menghapus user: {e}")

simpan_data() # Memanggil fungsi simpan_data() untuk
menyimpan data setelah dihapus

# Fungsi untuk menampilkan daftar pengguna dengan pemisahan peran
def tampilkan_users(users):
    print("\n=== Daftar Pengguna Terdaftar ===")

    # Pastikan list users tidak kosong
    if not users:
        print("Tidak ada pengguna terdaftar.")
        return

    try:

```

```

        # Filter pengguna berdasarkan peran
        admins = [user for user in users if user['role'] ==
'admin']
        regular_users = [user for user in users if user['role']
== 'user']

        # Buat tabel PrettyTable
        table = PrettyTable()
        table.field_names = ["Username", "Role"]

        # Tampilkan daftar admin
        if admins:
            print("\n-- Admin --")
            for user in admins:
                table.add_row([user['username'], user['role']])
            print(table)
        else:
            print("\n-- Tidak ada admin terdaftar --")

        # Tampilkan daftar pengguna biasa
        if regular_users:
            print("\n-- Pengguna Biasa --")
            table.clear_rows() # Menghapus baris sebelumnya
            for user in regular_users:
                table.add_row([user['username'], user['role']])
            print(table)
        else:
            print("\n-- Tidak ada pengguna biasa terdaftar --")
    except Exception as e:
        print(f"Terjadi kesalahan saat menampilkan daftar
pengguna: {e}")

# Prosedur untuk menampilkan riwayat aktivitas
def tampilkan_riwayat_aktivitas():
    print("\n=== Riwayat Aktivitas ===")
    try:
        if riwayat_aktivitas:
            table = PrettyTable()
            table.field_names = ["Waktu", "Aktivitas"]
            for waktu, aktivitas in riwayat_aktivitas:
                table.add_row([waktu, aktivitas])
            print(table)
        else:
            print("Belum ada aktivitas yang tercatat.")
    except Exception as e:

```

```

        print(f"Terjadi kesalahan saat menampilkan riwayat
aktivitas: {e}")

# Menu utama untuk menangani pilihan menu user atau registrasi
pengguna
def menu_utama():
    while True:
        try:
            pilihan = input("Pilih menu: \n1. Login\n2.
Registrasi Akun User Baru\n3. Keluar\nPilihan: ").strip()

            # Cek jika input kosong
            if not pilihan:
                print("Pilihan tidak boleh kosong. Silakan coba
lagi.")
                continue

            if pilihan == "1":
                clear_terminal()
                try:
                    current_user = login(users) # Memanggil
fungsi login

                    input("\nTekan Enter...")
                    if current_user:
                        # Jika user adalah admin
                        if current_user['role'] == 'admin':
                            menu_admin(inventory, inventory_user,
users, current_user) # Memanggil menu admin

                        # Jika user adalah user biasa
                        elif current_user['role'] == 'user':
                            menu_user(inventory_user, inventory,
current_user) # Memanggil menu user

                    else:
                        print("Login gagal. Username atau
password salah.")
                except Exception as e:
                    print(f"Terjadi kesalahan saat login: {e}")

            elif pilihan == "2":
                clear_terminal()
                try:
                    registrasi_pengguna(users) # Memanggil
fungsi registrasi untuk user

```

```

        except Exception as e:
            print(f"Terjadi kesalahan saat registrasi:
{e}")

            input("\nTekan Enter untuk melanjutkan...")

        elif pilihan == "3":
            clear_terminal()
            print("""
TERIMA KASIH

        TELAH MENGGUNAKAN SISTEM INI

        """)

            break

        else:
            print("Pilihan tidak valid. Silakan coba lagi.")
    except Exception as e:
        print(f"Terjadi kesalahan umum: {e}")

# Memanggil menu utama
menu_utama()

```

BAB IV

PENUTUP

4.1 Kesimpulan

Program CRUD untuk manajemen inventaris toko bangunan berhasil memberikan solusi yang efektif dan efisien dalam mengelola stok barang. Dengan fitur-fitur Create, Read, Update, and Delete, system ini memungkinkan pengelola toko untuk melakukan pencatatan barang masuk, memantau ketersediaan stok, memperbarui data barang, serta menghapus barang yang habis. Manajemen inventaris menjadi lebih terstruktur, akurat, dan mudah diakses, yang memudahkan pengelola toko dalam membuat keputusan yang tepat terkait pengadaan barang dan pengaturan stok. Dengan system ini, kesalahan dalam pencatatan manual dapat meminimalisir, dan proses transaksi menjadi lebih cepat.

4.2 Saran

- a. Integrasi dengan Sistem Pembayaran : agar lebih efisien, sebaiknya program ini diintegrasikan dengan system pembayaran atau kasir, sehingga stok barang dapat langsung terupdate secara otomatis setelah terjadi transaksi penjualan.
- b. Pengelolaan Persediaan yang Lebih Canggih : menambahkan peringatan jika stok menipis, atau habis. Hal ini dapat membantu pengelola toko dalam perencanaan pengadaan barang yang lebih baik.
- c. Peningkatan Fitur Laporan : menambahkan fitur laporan yang lebih detail, seperti laporan perkategori barang, laporan perbandingan antara barang yang laris dan barang yang kurang diminati. Hal ini dapat membantu pengelola toko dalam perencanaan pengadaan barang yang lebih baik.

DAFTAR PUSTAKA

Heizer, J., & Render, B. (2017). *Operations Management Sustainability and Supply Chain Management (12th ed)*. Pearson Education.

Krajewski, L. J., Ritzman, L. P., & Malhotra, M.K. (2013). *Operations Management: Processes and Supply Chains (10th ed)*. Pearson Education.

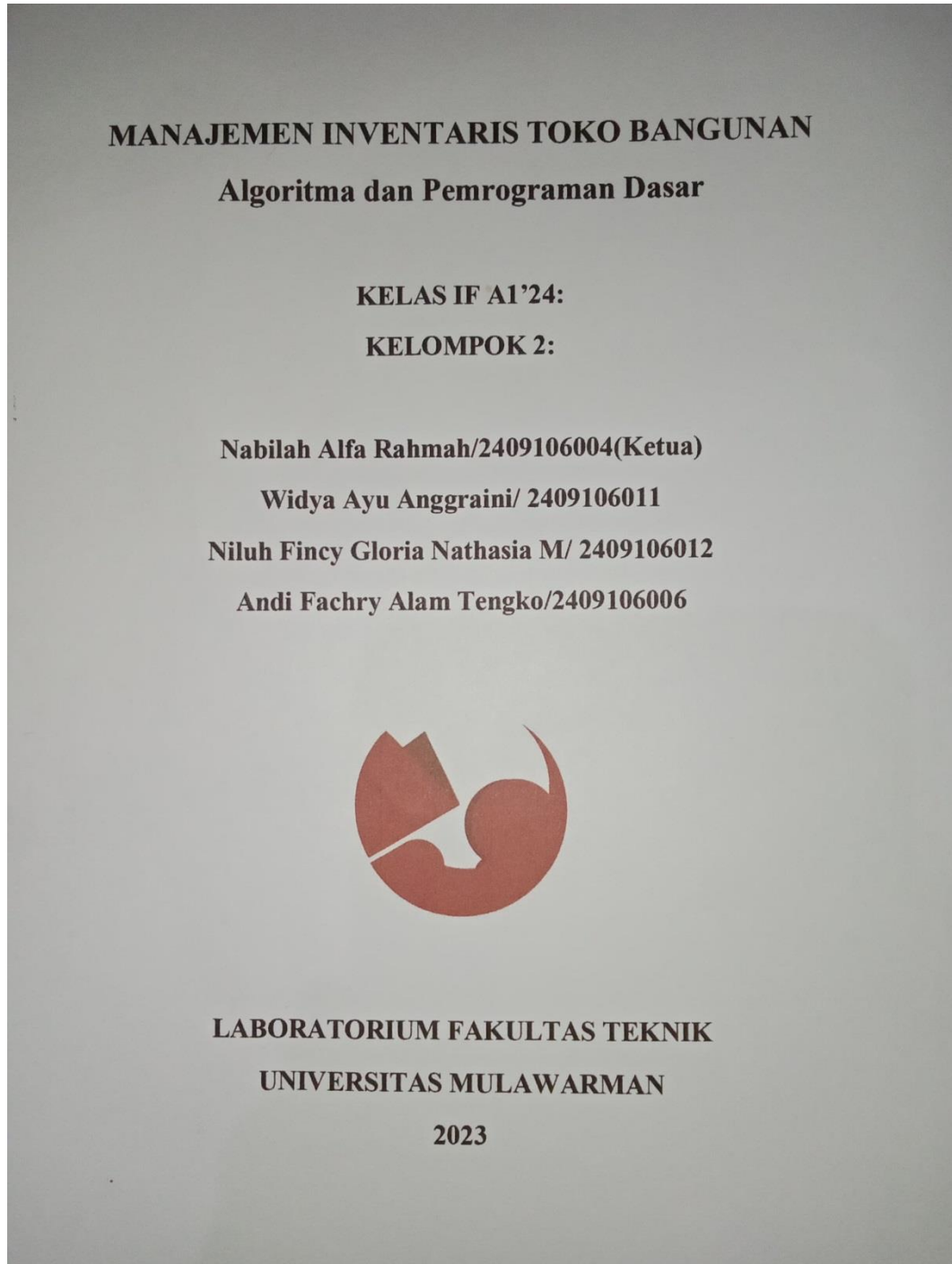
Van Rossum, G., & Drake, F. L. (2009). *The Python Language Reference (3rd ed)*. Python Software Foundation.

Lutz, M. (2013). *Learning Python (5th ed)*. O'Reilly Media

McKinney, W. (2017). *Python for Data Analysis: Data Wragling with Pandas, NumPy, and IPython (2nd ed)*. O'Reilly Media

Grayson, D. (2019). *Python Programming: An Introduction to Computer Science (3rd ed)*. Franklin, Beedle & Associate

LAMPIRAN



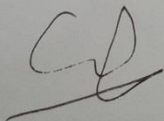
Gambar 38 Kartu Konsul Cover

Tanggal Konsultasi : Kamis . 31 Oktober 2024

Uraian / Pembahasan :

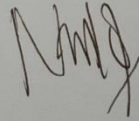
- Perbaiki flowchart (off -page)
- Menu user : - fitur CRUD (untuk barangnya sendiri)
- Menu Admin : - Bisa Menghapus akun user
- Menambahkan Error Handling untuk persamaan nama barang

Asisten Lab



Nama: Celio Arga

Ketua Kelompok



Nama: Nabilah Alfah Rahmah

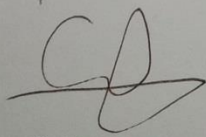
Gambar 39 Kartu Konsul Lembar 1

Tanggal Konsultasi : Rabu , 13 November 2024

Uraian / Pembahasan :

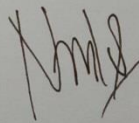
- Admin bisa CRUD punya user
- Menambah clear Terminal
- Menghapus Fungsi flowchart
- Start pada flowchart diganti menjadi on page
- flowchart dibuat lebih simpel (halaman baru)
- Menambahkan id user pada file eksternal

Asisten Lab



Nama: Celio Arga

Ketua Kelompok



Nama: Nabilah Alfah Rahmah

Gambar 40 Kartu Konsul Lembar 2