

LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR



INVENTARIS GUDANG ELEKTRONIK

Oleh:

Kelompok 5

Rafly Rahmat Wijaya A.S	2409106119
Reza Alameka	2409106107
Nazla Salsabila	2409106108
Desya Oktavia R.	2409106111

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024

KATA PENGANTAR

Bismillahirrohmanirrohim, Alhamdulillah puji syukur penulis panjatkan kehadirat Allah SWT, atas rahmat dan hidayahnya sehingga laporan proyek akhir “INVENTARIS GUDANG ELEKTRONIK” ini dapat diselesaikan sesuai waktu yang dijadwalkan. Laporan proyek akhir ini disusun untuk memenuhi penugasan proyek akhir praktikum pemrograman pada semester satu.

Sistem ini merupakan aplikasi untuk mengelola persediaan barang elektronik di gudang, yang dirancang bagi dua pengguna atau multiuser, dimana di dalam penggunaanya ada admin sebagai administrator dan user sebagai pegawai. Disini admin memiliki akses untuk menambah, melihat, mengupdate, menghapus dan membuat atau melihat riwayat peminjaman. Sementara user juga dapat menambah, dan melihat riwayat peminjaman.

Adapun kendala yang dialami saat merancang program yaitu komunikasi sesama anggota kelompok dan kesalahan atau eror saat menjalankan program. Oleh karena itu, pada kesempatan ini patutlah kiranya kami mengucapkan terima kasih kepada :

1. Kedua orang tua kami yang selalu memberikan doa serta dukungan.
2. Kepada dosen yaitu Bapak Muhammad Bambang Firdaus, S.Kom, M. Kom dan Bapak Awang Harsa Kridalaksana, S.Kom., M.Kom yang telah memberikan kami pengajaran dan bimbingan sehingga memudahkan kami dalam menyelesaikan program yang dibuat.
3. Kepada para asisten laboratorium mata kuliah algoritma pemrograman dasar. Terima kasih atas segala bimbingan, ajaran, dan ilmu-ilmu baru yang diberikan kepada kami selama praktikum dilaksanakan. Dengan segala kesibukan masing-masing masih bersedia membimbing dalam penyusunan proyek akhir.

Samarinda, 11 November 2024

Kelompok 5

TAKARIR

<i>Input</i>	Memasukan
<i>Output</i>	Mengeluarkan
<i>Real- Time Tracking</i>	Pelacakan waktu nyata
<i>Username</i>	Identitas pengguna
<i>Password</i>	Kata sandi
<i>Inventory</i>	Inventaris
<i>Ms. Excel</i>	Aplikasi pengolah data
<i>Print</i>	Mencetak pesan yang ditentukan ke layar
<i>Number</i>	Angka
<i>List</i>	Daftar atau kumpulan objek
<i>Append</i>	Menambah item diakhir list
<i>Item</i>	Barang
<i>Dictionary</i>	Menyimpan pasangan kunci-nilai
<i>Value</i>	Nilai
<i>Remove</i>	Menghapus
<i>Software</i>	Perangkat lunak, kumpulan instruksi
<i>Comma Separated Values</i>	Format file sederhana untuk menyimpan data tabular dalam spreadsheet atau database
<i>Exit</i>	Mengeluarkan dari aplikasi yang saat ini berjalan
<i>Flowchart</i>	Diagram atau bagan alur yang menggambarkan alur proses atau prosedur menggunakan simbol-simbol standar

<i>Import</i>	Proses mengimpor kode dari Pustaka ke dalam kode
<i>Break</i>	Keluar dari loop saat kondisi eksternal terpenuhi atau menghentikan perulangan
<i>Time.sleep</i>	Jumlah detik yang dibutuhkan untuk menghentikan kode
<i>Pwinput</i>	Modul Python lintas platform yang menampilkan ***** untuk memasukkan kata sandi
<i>Library</i>	Kumpulan kode yang telah dioptimalkan, sehingga membantu analyst dalam menghindari penulisan rutin dan fokus pada penciptaan solusi yang lebih inovatif
<i>Try-except</i>	Menangani atau mengontrol kesalahan pada alur program

DAFTAR ISI

KATA PENGANTAR	2
TAKARIR.....	3
DAFTAR ISI.....	5
DAFTAR GAMBAR	7
BAB I PENDAHULUAN	8
1.1 Latar Belakang	8
1.2 Kebutuhan Fungsional.....	9
1.3 Rumusan Masalah	10
1.4 Batasan Masalah.....	11
1.5 Tujuan.....	11
BAB II PERANCANGAN.....	12
2.1 Analisis Program	12
2.2 Flowchart.....	14
2.3 Konsep dan Materi Praktikum yang Dipakai	15
BAB III HASIL DAN PEMBAHASAN.....	20
3.1 Tampilan Program.....	20
1. Tampilan Menu	20
2. Tampilan login admin	20
3. Tampilan menu read alat.....	21
4. Tampilan menu create alat	22
5. Tampilan menu update alat	23
6. Tampilan berhasil update alat	23
7. Tampilan menu delete barang	24

8. Tampilan riwayat peminjaman.....	25
9. Tampilan register khusus user	25
10. Menu user.....	26
11. Tampilan register akun user.....	26
12. Peminjaman barang user	27
13. Keluar dari program	27
3.2 Source Code	27
BAB IV PENUTUP	45
4.1 Kesimpulan.....	45
4.2 Saran	45
DAFTAR PUSTAKA	46
LAMPIRAN.....	47

DAFTAR GAMBAR

Gambar 2.1 Flowchart	14
Gambar 3.1 Tampilan awal	20
Gambar 3.2 Tampilan login admin.....	20
Gambar 3.3 Tampilan menu read alat	21
Gambar 3.4 Tampilan menu create alat.....	22
Gambar 3.5 Tampilan menu update alat.....	23
Gambar 3.6 Tampilan berhasil update alat.....	23
Gambar 3.7 Tampilan menu delete barang	24
Gambar 3.8 Tampilan riwayat peminjaman	25
Gambar 3.9 Tampilan register khusus user	25
Gambar 3.10 Menu user	26
Gambar 3.11 Tampilan register akun user.....	26
Gambar 3.12 Peminjaman barang user.....	27
Gambar 3.13 Keluar dari program.....	2
Gambar 4.1 Halaman pertama kartu konsul	47
Gambar 4.2 Pemaparan konsultasi pertama	48
Gambar 4.3 Pemaparan konsultasi kedua.....	49

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suatu divisi mengatur semua barang yang terdaftar dari suatu instansi untuk digunakan sebagai pelaksanaan kegiatan-kegiatan dari instansi tersebut. Sistem informasi persediaan barang yang baik dapat meningkatkan produktifitas dalam pengolahan data persediaan barang menggunakan komputer untuk pencatatan data barang, yaitu dengan menggunakan software *Ms. Excel*. Namun, dalam pencatatan data masih mengalami kendala, yaitu terjadi ketidaksesuaian baik data-data hilang dan terjadi pengulangan data, sehingga dengan permasalahan tersebut membuat karyawan harus melakukan pengecekan ulang terhadap kondisi barang yang tersedia dan merevisi laporan *inventory* (persediaan barang).

Sistem *inventory* atau persediaan barang memiliki pengaruh besar terhadap suatu instansi ataupun perusahaan. Sistem *inventory* (persediaan barang) dapat membantu menyelesaikan adanya masalah pengolahan data barang dan memudahkan pelaporan data barang atau data alat yang tersedia. Dibutuhkan adanya suatu sistem yang dapat mengatur data persediaan barang dan alat dengan baik. Pengaturan perlengkapan barang berupa pencatatan, pelaporan serta persediaan barang yang digunakan untuk operasional kantor.

Dari uraian latar belakang tersebut terpikirlah sebuah ide untuk akhirnya memutuskan membuat sistem program yang berjudul Inventaris Gudang Elektronik. Program ini tentunya banyak menawarkan keuntungan dibandingkan metode manual, yaitu keunggulannya

1. Akurasi Data : Sistem ini akan terkomputerisasi dan tentunya mengurangi kesalahan manusia yang sering terjadi pada pencatatan manual.
2. Efisiensi : Otomatisasi proses input dan output barang atau alat, seperti penerimaan, penyimpanan, dan meningkatkan efisiensi operasional

3. *Real-time Tracking* : Kita dapat memantau stok barang secara langsung, mengetahui jumlah barang yang tersedia, dan mengidentifikasi barang yang stok nya sudah hampir habis.
4. Pengurangan Biaya : Mengurangi pemborosan dan kehilangan barang karena kontrol stok yang lebih ketat.
5. Peningkatan Keamanan : Sistem akses terkontrol, seperti username dan password, meningkatkan keamanan data inventaris.

1.2 Kebutuhan Fungsional

Pada program kami, kami memiliki masing masing fitur admin dan user, dengan kebutuhan fungsional sebagai berikut :

1. Fitur Register

Pada fitur ini berfungsi untuk meregister atau dikhususkan untuk mendaftarkan terlebih dahulu akun user baru dengan cara memasukan *username* dan *password* baru khusus user. Khusus admin tidak diwajibkan meregister akun karena akun admin sudah bersifat variabel statis. Akun user yang sudah terdaftar bisa dilihat riwayat di file eksternal *Comma Separated Values* (CSV).

2. Fitur Login

Setelah berhasil mendaftarkan akun user, fitur ini berfungsi sebagai tanda bahwa user atau admin bisa masuk ke dalam program untuk melanjutkan ke menu selanjutnya.

3. Fitur Lihat Alat

Fitur ini ada pada kedua menu untuk user dan admin dimana ketika admin atau user memilih opsi ini maka tampilan barang yang tersedia di pergudangan elektronik akan terlihat di tampilan program dalam bentuk tabel

4. Fitur Tambah Alat

Fitur ini hanya bisa diakses khusus admin saja, menambahkan barang atau alat memang kami jalankan untuk tugas admin yang berfungsi untuk menambahkan barang baru dalam tabel untuk ditampilkan pada user.

5. Fitur Update Alat

Sama seperti tambah alat, fitur ini hanya dijalankan khusus admin saja, yang fungsinya untuk mengedit atau mengubah barang persediaan yang diinginkan tapi tanpa mengubah id dari barang sebelumnya.

6. Fitur Hapus Alat

Fitur yang hanya dijalankan khusus untuk admin saja, fungsinya admin bisa menghapus kapan saja dan persediaan barang apa saja yang diinginkan dan barang yang sudah dihapus tidak ada lagi dalam tampilan.

7. Fitur Riwayat Peminjaman

Admin bisa melihat detail riwayat peminjaman dari user, fungsinya fitur ini yaitu sebagai pengingat bagi admin tentang barang barang yang dipinjam oleh user dan semua riwayat ini akan masuk juga di file eksternal. Riwayat yang terlihat nantinya berupa barang yang dipinjam tipe, stok, id, tanggal peminjaman dan pengembalian barang.

8. Fitur Peminjaman Barang

Pada fitur ini hanya disediakan khusus user untuk menjalankan aktivitas peminjaman barang yang fungsinya disini adalah user bisa melakukan peminjaman barang hanya dengan memasukkan nama, tipe, jumlah stok barang yang ingin dipinjam, id barang, dan jumlah hari peminjaman barang.

9. Fitur *Exit*/Keluar

Fungsinya untuk mengakhiri segala aktivitas dalam program, jika sudah selesai dengan pilihan pilihan menu pada program baik admin ataupun user bisa mengakhiri jalannya program, dan fitur ini disediakan untuk admin maupun user.

1.3 Rumusan Masalah

Berdasarkan uraian masalah diatas maka dapat diambil rumusan masalah sebagai berikut :

1. Bagaimana merancang program sistem inventaris gudang elektronik?
2. Bagaimana cara menyajikan data inventaris yang akurat dan tepat pada persediaan barang dan alat elektronik?

1.4 Batasan Masalah

Batasan masalah berguna untuk penjelasan program nantinya. Adapun batasan masalah yang digunakan yaitu :

1. Program kami tidak ada opsi pengembalian barang dalam menu user sehingga, tidak ada peringatan/tagihan dalam pengembalian barang
2. Barang yang disediakan khusus barang elektronik
3. Pada bagian *password* untuk registrasi dan login tidak diberikan batasan maksimal berapa karakter

1.5 Tujuan

1. Mempermudah Pengawasan dan Pengendalian
Menguraikan bagaimana program memungkinkan pengawasan stok barang, meminimalkan kesalahan, dan mempermudah pelacakan barang masuk/keluar.
2. Peningkatan Keandalan Data
Menunjukkan bagaimana program dapat menyediakan data *real-time* yang dapat diandalkan untuk mendukung pengambilan keputusan.
3. Evaluasi Sistem
Menyediakan alat untuk mengevaluasi efektivitas program, identifikasi kelemahan, dan area untuk perbaikan di masa depan.

BAB II

PERANCANGAN

2.1 Analisis Program

Program ini jika dijalankan akan masuk ke :

Menu register, pada tahap ini user diminta untuk mendaftarkan akun yang akan digunakan terlebih dahulu untuk bisa login. Sedangkan untuk admin bisa langsung login dengan akun dan kata sandi yang sudah ditentukan. Kemudian admin diminta untuk memilih menu atau fitur yang sudah tersedia khusus admin.

A. Menu Login Khusus Admin :

1. Menambahkan barang

Disini admin dapat menambahkan barang/alat, tipe alat, id alat beserta stok persediaan yang ingin ditambah.

2. Melihat barang

Disini admin dapat melihat barang/alat apa saja yang tersedia di dalam gudang elektronik.

3. Mengupdate barang

Disini admin dapat mengupdate/memperbarui/mengedit persediaan barang/alat yang tersedia namun tidak merubah id barang sebelumnya.

4. Menghapus barang

Disini admin dapat menghapus jenis jenis barang/alat yang ingin dihapus.

5. Melihat riwayat peminjaman barang

Disini admin bisa melihat berbagai riwayat peminjaman yang dilakukan oleh user atau karyawan mulai dari riwayat tanggal peminjaman barang, pengembalian barang, dan barang apa saja yang sudah dipinjam.

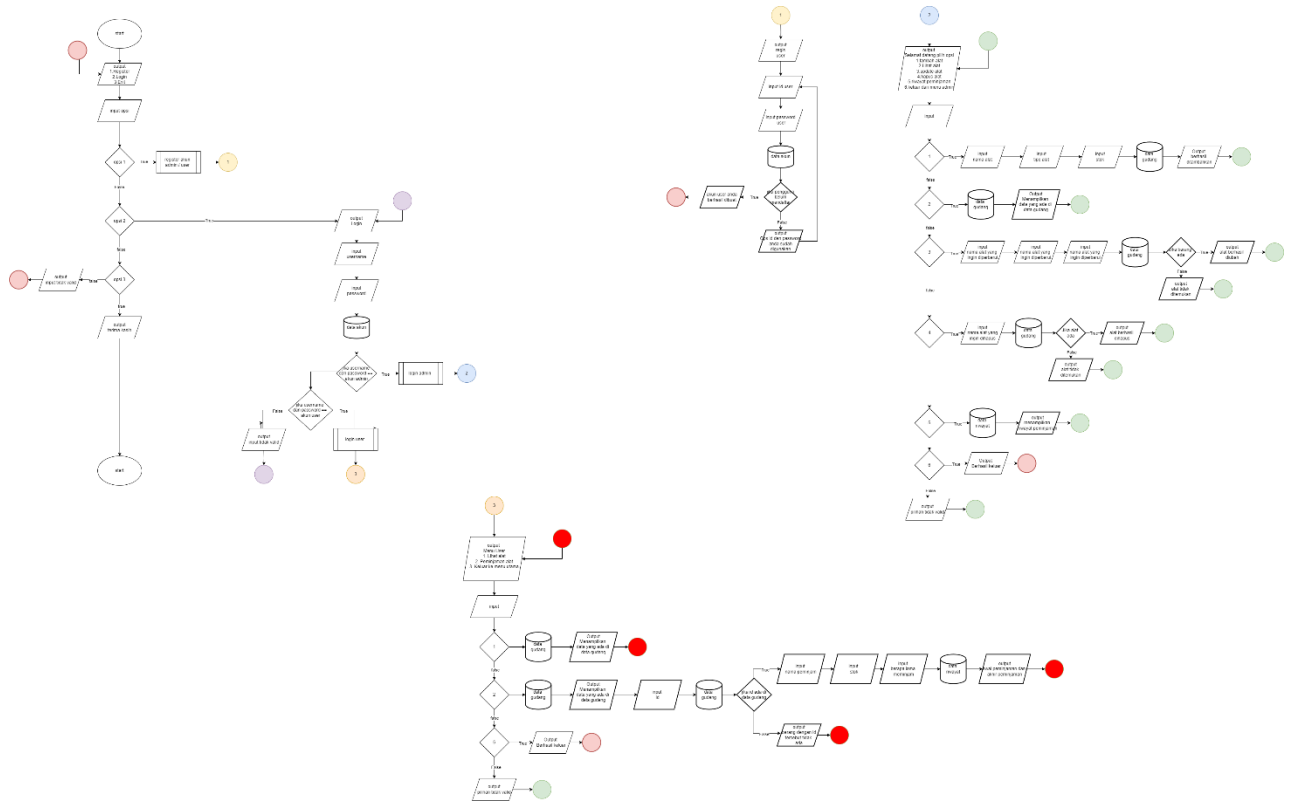
6. Keluar

Memilih opsi ini maka admin otomatis akan keluar dari program.

B. Menu Login Khusus User :

1. Untuk user disini akan diarahkan untuk melakukan register terlebih dahulu dengan cara membuat *username* dan *password* baru.
2. Setelah berhasil melakukan register untuk *username* dan *password* yang sudah terdaftar maka user sudah bisa ke menu *login*.
3. Sesudah *login* maka user kembali diarahkan untuk memilih menu.
4. Pada menu pertama yaitu melihat barang, user akan ditampilkan barang apa saja yang masih tersedia pada gudang, jenis barang, tipe barang, id barang dan stok barang.
5. Pada menu kedua yaitu menu untuk peminjaman barang, user bisa meminjam barang apa saja dengan hanya memasukkan id barang, kemudian menentukan jumlah hari peminjaman yang diinginkan.
6. Setelah sudah melakukan peminjaman barang, user bisa mengakhiri aktivitas di dalam program dengan memilih opsi keluar dari menu atau *Exit* maka program otomatis akan berhenti.

3.2 Flowchart



Gambar 2.1 Flowchart

3.3 Konsep/Materi Praktikum yang Dipakai

A. Algoritma

Algoritma adalah urutan langkah - langkah logis yang menyatakan suatu tugas dalam menyelesaikan suatu masalah yang disusun secara sistematis. Atau bisa juga diartikan dengan urutan aksi-aksi yang jelas dan tidak rancu untuk menyelesaikan suatu masalah.

B. Program

Program adalah sebuah algoritma yang ditulis dalam bahasa computer. Dimana suatu algoritma dikatakan efektif bila dijalankan oleh sebuah pemroses (processor). Instruksi yang terdapat pada algoritma dibaca dan kemudian dikerjakan oleh processor.

C. Python

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, *software* atau aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja. Karena sifatnya yang serba guna dan mudah digunakan, ia menjadi bahasa pemrograman yang paling banyak digunakan.

D. Flowchart

Flowchart (diagram alir) adalah sebuah penggambaran secara grafik dari Langkah pemecahan masalah yang harus diikuti oleh pemroses. Flowchart terdiri atas sekumpulan symbol, dimana masing-masing simbol menggambarkan suatu kegiatan tertentu. Flowchart diawali dengan penerimaan masukan (*input*), pemrosesan masukan, dan diakhiri dengan menampilkan hasilnya (*output*). Dimana

flowchart ini dapat digunakan sebagai alat bantu komunikasi dan untuk dokumentasi.

Adapun konsep yang kami gunakan dalam pembuatan laporan proyek akhir ini sesuai dengan persyaratan yang terdapat pada modul-modul diantaranya:

1. Fungsi dasar

Fungsi dasar yang kami pakai dalam program ini adalah fungsi *print*. Fungsi *print* disebut juga sebagai fungsi dasar, fungsi print program merupakan fungsi yang paling umum dipakai.

2. Fungsi *Number* (int)

Fungsi *Number* adalah tipe data number merepresentasikan nilai-nilai berupa angka. Karena python menggolongkan beberapa tipe data umum seperti, integer untuk bilangan bulat dan *floating-point* untuk bilangan desimal ke dalam tipe data *number*.

3. Fungsi CRUD

Fungsi *Create, Read, Update* dan *Delete* (CRUD) ini merupakan operasi dasar yang umum digunakan pada pemrograman database.

4. Fungsi *List*

Fungsi ini 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 ubah-ubah. Sebuah list dapat dibuat dengan menggunakan tanda kurung siku [].

5. Fungsi *Append (item)*

Fungsi ini adalah sebuah fungsi dalam python yang digunakan untuk menambahkan item dari belakang dan menambahkan objek-objek ke list.

6. 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 { }.

7. Fungsi percabangan

Didalam program ini 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. Pada fungsi percabangan di program ini kami bukan hanya menggunakan fungsi *If* tetapi juga menggunakan fungsi percabangan *If*, *Else*, dan *Elif*. Fungsi *If* / *Else* / *Elif* digunakan untuk memuat kondisi/logika tambahan apabila kondisi pertama salah.

8. 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*. Kemudian fungsi perulangan yang program kami gunakan lagi adalah fungsi perulangan *while* yaitu fungsi perulangan yang juga disebut *uncounted loop* (perulangan yang tak terhitung). Perulangan *while* adalah perulangan yang memiliki syarat dan tidak tentu berapa banyak perulangannya.

9. 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.

10. 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.

11. 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.

12. Fungsi *Import*

Fungsi *import* berfungsi untuk multi file maksudnya yaitu kita dapat memanggil atau mengakses file lain didalam satu module yang berbeda.

13. Fungsi *Remove ()*

Fungsi ini berfungsi untuk menghapus elemen yang dipilih, termasuk semua teks. Metode ini juga menghapus data dan peristiwa dari elemen yang dipilih.

14. Fungsi *Break*

Fungsi ini digunakan untuk menghentikan jalannya proses iterasi pada statement *for* atau *while*.

15. Fungsi *Time.sleep ()*

Fungsi ini berguna untuk menunda waktu eksekusi. Fungsi ini membawa parameter berupa angka yang menyatakan detik (lama) penundaan.

16. Fungsi *Pwinput*

Fungsi ini platform yang menampilkan ***** untuk memasukan kata sandi yang bertujuan untuk menjaga privasi.

17. Fungsi *PrettyTable*

Prettytable merupakan *library* atau pustaka dalam python yang digunakan untuk membuat atau mengeluarkan data dalam bentuk tabel.

18. Fungsi *Try Except*

Fungsi ini digunakan pada program kami untuk menangani kesalahan, *try-except* juga bisa digunakan sebagai kontrol alur program. Misalnya, kita dapat mencoba melakukan suatu aksi. Jika gagal, kita bisa melakukan aksi lain sebagai gantinya.

19. Fungsi File Eksternal

Pada program ini kami menggunakan file eksternal berupa *csv* yang fungsinya untuk memanggil fungsi dari file lain untuk memanggil fungsi dari file lain, kita dapat menggunakan instruksi “*from*” dan “*import*” di program utama.

20. Fungsi *Error handling*

Fungsi *error handling* dalam program kami ini adalah untuk mendeteksi, mengantisipasi, dan menyelesaikan kesalahan atau pengecualian yang mungkin terjadi selama eksekusi program. Tujuannya agar program tidak mogok atau berperilaku tidak terduga saat terjadi kesalahan.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Awal Pogram

```
Selamat Datang di E-GUDANG ELEKTRONIK
1.register
2.Login
3.Exit

Pilih opsi: 
```

Gambar 3.1 Tampilan awal

Saat masuk pada program, akan terlihat menu utama yang dapat dipilih dan digunakan untuk admin atau user.

2. Tampilan Login Admin

```
=== Login ===
Masukkan username: admin
Masukkan password: ****
Berhasil login sebagai admin
Loading..█
```

Gambar 3.2 Tampilan login admin

Saat masuk pada menu login untuk admin, admin memasukan *username* dan *password* yang sudah ditentukan. Pada program ini kami menggunakan fungsi *pwinput* yaitu platform yang menampilkan **** untuk memasukan kata sandi yang bertujuan untuk menjaga privasi.

3. Tampilan Pilihan Menu Lihat Khusus Admin

```
Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 2

+-----+-----+-----+-----+
| nama barang | tipe barang | stok barang | id |
+-----+-----+-----+-----+
| Samsung     | handphone  | 120         | YA |
| Lampu       | 5 Watt     | 200         | 70W |
| Kabel       | NYA        | 500         | 10X |
| Laptop      | Acer       | 90          | 30P |
+-----+-----+-----+-----+

Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 
```

Gambar 3.3 Tampilan menu read alat

Saat memilih opsi 2 untuk menu lihat alat maka program akan menampilkan nama, tipe, stok, dan id barang dalam bentuk tabel seperti pada gambar. Pada opsi ini program akan menampilkan spesifik barang yang tersedia di gudang dengan tampilan seperti gambar di atas.

4. Tampilan Pilihan Menu Tambah Khusus Admin

```
07. Keluar ke menu utama
Pilih opsi: 1
Masukkan nama barang: Televisi
Masukkan tipe barang: LG
Masukkan stok barang: 375
Masukkan id barang: 25C
Data Televisi berhasil ditambahkan dengan id:25C.

Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 2
+-----+-----+-----+-----+
| nama barang | tipe barang | stok barang | id |
+-----+-----+-----+-----+
| Samsung     | handphone  | 120         | YA |
| Lampu       | 5 Watt     | 200         | 70W |
| Kabel       | NYA        | 500         | 10X |
| Laptop      | Acer       | 90          | 30P |
| Televisi    | LG         | 375         | 25C |
+-----+-----+-----+-----+
```

Gambar 3.4 Tampilan menu create alat

Setelah sudah selesai melihat barang, admin memiliki fitur untuk menambahkan barang yang diinginkan. Sesudah menambahkan, barang akan masuk lagi dalam list persediaan barang baru seperti pada tampilan.

5. Tampilan Pilihan Menu Update Khusus Admin

nama barang	tipe barang	stok barang	id
Samsung	handphone	120	YA
Lampu	5 watt	200	70W
Kabel	NYA	500	10X
Laptop	Acer	90	30P
Televisi	LG	375	25C

Khusus ADMIN

1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama

Pilih opsi: 3

Masukkan id barang yang ingin diubah: YA

Barang ditemukan: Samsung, handphone, 120

Masukkan nama baru:

Gambar 3.5 Pilihan menu update alat

Pada tampilan ini admin bisa *mengupdate* atau mengubah barang sebelumnya dengan cara memasukan nama, tipe, dan stok barang baru untuk ditampilkan pada tabel dan barang terbaru akan muncul ditampilkan.

```
Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 3
Masukkan id barang yang ingin diubah: YA
Barang ditemukan: Samsung, handphone, 120
Masukkan nama baru: Setrika
Masukkan tipe baru: Philips
Masukkan stok baru: 250
Data barang dengan id:YA berhasil diperbarui.
Data YA berhasil diperbarui.
```

Gambar 3.6 Tampilan berhasil update alat

Barang sudah berhasil diperbarui dengan id yang sama pada barang sebelumnya. Id barang tidak berubah hanya saja nama, tipe, dan stok barang saja yang bisa berubah.

6. Tampilan Pilihan Menu Hapus Khusus Admin

nama barang	tipe barang	stok barang	id
Lampu	5 Watt	200	70W
Kabel	NYA	500	10X
Laptop	Acer	90	30P
Televisi	LG	375	25C
Samsung	handphone	120	YA

Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 4
Masukkan ID Barang: YA
Barang dengan ID YA berhasil dihapus.

Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 2

nama barang	tipe barang	stok barang	id
Lampu	5 Watt	200	70W
Kabel	NYA	500	10X
Laptop	Acer	90	30P
Televisi	LG	375	25C

Gambar 3.7 Tampilan menu delete barang

Jika admin memilih opsi 4, admin bisa menghapus barang apa yang diinginkan. Dengan cara menekan opsi 4 dan masukan id barang yang ingin dihapus maka secara otomatis barang akan terhapus dan barang tidak terdapat lagi ditampilkan layar.

7. Tampilan Melihat Riwayat Peminjaman Khusus Admin

```
Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 5
+-----+-----+-----+-----+-----+
| nama | id barang | stok | waktu awal minjam | waktu akhir peminjaman |
+-----+-----+-----+-----+-----+
| Kelompok 5 | 10X | 10 | 2024-11-17 | 2024-11-19 |
| Kelompok 5 | 10X | 5 | 2024-11-17 | 2024-11-20 |
| Reja | 30P | 7 | 2024-11-20 | 2025-02-18 |
+-----+-----+-----+-----+-----+

Khusus ADMIN
1. Tambah Alat
2. Lihat Alat
3. Update Alat
4. Hapus Alat
5. Riwayat Peminjaman
6. Keluar ke menu utama
Pilih opsi: 
```

Gambar 3.8 Tampilan riwayat

Jika memilih opsi 5 disini admin tentunya bisa melihat riwayat peminjaman seperti akun user. Riwayat yang ditampilkan seperti nama, id, stok barang tanggal yang ditampilkan waktu awal meminjam dan tanggal waktu akhir peminjaman sesuai ketentuan user.

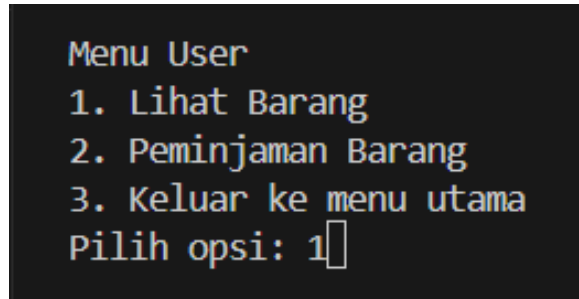
8. Tampilan Register User

```
==== Registrasi Akun User ====
Masukkan username baru: Kelompok lima
Masukkan password baru: ****
Akun Anda berhasil terdaftar dengan ID: Kelompok lima
Tekan enter untuk kembali ke menu utama
```

Gambar 3.9 Tampilan register khusus user

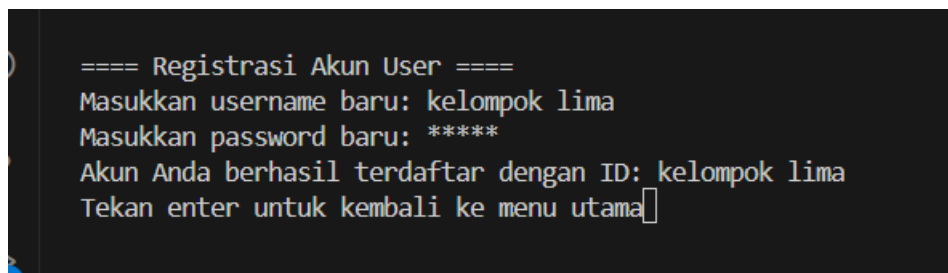
Khusus user sebelum login, user harus membuat akun atau register akun terlebih dahulu.

9. Tampilan Menu Khusus User



Gambar 3.10 Menu user

Pada tampilan khusus user, program kami hanya menyediakan 3 fitur saja seperti tertera pada gambar di atas.



Gambar 3.11 Tampilan registrasi akun user

Ini adalah tampilan akun user yang berhasil dan sudah melakukan registrasi atau pendaftaran akun agar bisa beralih ke menu *login*. Disini kami mengarahkan user menggunakan enter untuk kembali ke menu utama.

10. Tampilan Peminjaman Barang User

```
Menu User
1. Lihat Barang
2. Peminjaman Barang
3. Keluar ke menu utama
Pilih opsi: 2
+-----+-----+-----+-----+
| nama barang | tipe barang | stok barang | id |
+-----+-----+-----+-----+
| Lampu       | 5 Watt     | 200        | 70W |
| Kabel       | NYA        | 500        | 10X |
| Laptop      | Acer       | 90         | 30P |
| Televisi    | LG         | 375        | 25C |
+-----+-----+-----+-----+
Masukkan ID Barang: 30P
Masukkan nama peminjam: Kelompok lima
Berapa stok yang ingin dipinjam (maks 10): 8
Masukkan jumlah hari peminjaman: 17
Tanggal peminjaman: 2024-11-21
Tanggal pengembalian: 2024-12-08
```

Gambar 3.12 Peminjaman barang user

Jika user memilih opsi 2 otomatis program akan menampilkan barang yang tersedia untuk peminjaman, peminjaman dilakukan hanya dengan memasukan id barang, nama peminjaman sesuai akun yang terdaftar, stok yang ingin dipinjam (maks 10) dan masukan jumlah hari peminjaman yang diinginkan jika lebih dari 10 maka output yang keluar adalah “peminjaman barang tidak boleh melebihi 10 stok” dan program akan kembali ke menu user.

11. Tampilan keluar dari program

```
OUTPUT  DEBUG CONSOLE  PROBLEMS  TERMINAL  PORTS
Terima kasih telah mengunjungi E-GUDANG
Program akan keluar
Loading...█
```

Gambar 3.13 User keluar dari program

Jika memilih opsi 3 maka user akan otomatis akan keluar dari program.

3.1 Source Code

Register Akun User

```
import pwininput

import csv

# Membaca akun dari file CSV

def akun_baru():

    try:

        with open("akuns.csv", mode="r", newline="") as file:

            reader = csv.reader(file)

            return list(reader)

    except FileNotFoundError:

        return [["admin", "1234", "admin"]] # Data default jika file belum ada

# Menyimpan akun ke file CSV

def simpan_akun():

    with open("akuns.csv", mode="w", newline="") as file:

        writer = csv.writer(file)

        writer.writerows(akuns)

# Fungsi untuk registrasi akun

def register():

    print("\n==== Registrasi Akun User =====")

    username = input("Masukkan username baru: ")

    cek_akun = akun_baru() # Mengambil list akun dari file
```

```

akuna = False

if not username:

    print("username tidak boleh kosong")

    return

for akun in cek_akun:

    if akun[0] == username:

        akuna = True

        break

if akuna:

    print("Username telah dipakai, silakan coba lagi")

else:

    password = pwinput.pwinput("Masukkan password baru: ")

    if not password:

        print("password tidak boleh kosong")

        return

    akuns.append([username, password, "user"])

    simpan_akun() # Menyimpan akun yang baru didaftarkan ke file CSV

    print(f"Akun Anda berhasil terdaftar dengan ID: {username}")

# Contoh penggunaan

akuns = akun_baru() # Membaca akun yang sudah ada dari file saat program
dijalankan

if __name__ == '__main__':

    register()

```

Login Akun Admin atau User

```
from tools import loading

import pwinput

import os

from admin_tools import crud_admin

from user_tools import crud_user

import main

from auth import register


akuns = register.akun_baru()

# start login

def login():

    try:

        print("\n==== Login ====")

        username = input("Masukkan username: ")

        password = pwinput.pwinput("Masukkan password: ")

        for akun in akuns:

            if akun[0] == username and akun[1] == password and akun[2] ==

"admin":

                print("Berhasil login sebagai admin")

                loading.loading_efek()

                os.system('cls || clear')

                while True:
```

```
print("\nKhusus ADMIN\n1. Tambah Alat\n2. Lihat Alat\n3. Update  
Alat\n4. Hapus Alat\n5. Riwayat Peminjaman\n6. Keluar ke menu utama")
```

```
opsi = int(input("Pilih opsi: "))
```

```
if opsi == 1:
```

```
    crud_admin.tambah_barang()
```

```
elif opsi == 2:
```

```
    crud_admin.lihat_barang()
```

```
elif opsi == 3:
```

```
    crud_admin.update_barang()
```

```
elif opsi == 4:
```

```
    crud_admin.hapus_barang()
```

```
elif opsi == 5:
```

```
    crud_admin.riwayat_barang()
```

```
elif opsi == 6:
```

```
    print("Keluar dari menu admin")
```

```
    loading.loading_efek()
```

```
    os.system('cls || clear')
```

```
    main.menu()
```

```
    break
```

```
elif akun[0] == username and akun[1] == password and akun[2] == "user":
```

```
    print("Berhasil login sebagai user")
```

```
    loading.loading_efek()
```

```
    os.system('cls || clear')
```

```

while True:

    print("\nMenu User\n1. Lihat Barang\n2. Peminjaman Barang\n3.
Keluar ke menu utama")

    opsi = int(input("Pilih opsi: "))

    if opsi == 1:

        crud_user.lihat_barang()

    elif opsi == 2:

        crud_user.peminjaman_barang()

    elif opsi == 3:

        main.menu()

        break

    print("username atau password anda salah,silahkan coba lagi")

except ModuleNotFoundError:

    print("Modul tidak ketemu")

# end login

if __name__ == '__main__':

    login()

```


Khusus Admin

```
import csv

from prettytable import PrettyTable

import os

gudang = os.path.join(os.path.dirname(__file__), '..', 'gudang.csv')
riwayat = os.path.join(os.path.dirname(__file__), '..', 'riwayat.csv')

def tambah_barang():

    try:

        nama_barang = input("Masukkan nama barang: ").capitalize()

        tipe_barang = input("Masukkan tipe barang: ")

        stok_barang = int(input("Masukkan stok barang: "))

        id_barang = input("Masukkan id barang: ")

        if not nama_barang or not tipe_barang or not id_barang:

            raise ValueError("input valid tidak boleh kosong")

    except ValueError:

        print("input harus berupa angka")

    except KeyboardInterrupt:

        print("input dibatalkan.")
```

```

except Exception as e:

    print(f"Terjadi kesalahan: {e}")

else:

    with open(gudang, mode="a", newline="") as file:

        fieldnames = ["nama barang", "tipe barang", "stok barang", "id"]

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

        writer.writerow({

            "nama barang": nama_barang,

            "tipe barang": tipe_barang,

            "stok barang": stok_barang,

            "id"      : id_barang

        })

        print(f>Data {nama_barang} berhasil ditambahkan dengan
id:{id_barang}.")

def lihat_barang():

    try:

        with open(gudang, mode="r", newline="") as file:

            csvreader = csv.DictReader(file)

            data = list(csvreader)

            if data:

```

```

        table = PrettyTable()

        table.field_names = data[0].keys()

        for row in data:

            table.add_row(row.values())

            print(table)

        else:

            print("\nData kosong.")

    except FileNotFoundError:

        print("File tidak ditemukan, buat data baru terlebih dahulu.")

    except Exception as e:

        print(f"Terjadi kesalahan: {e}")

def update_barang():

    id_barang = input("Masukkan id barang yang ingin diubah: ")

    barang = False

    with open(gudang, mode="r", newline="") as file:

        read = csv.DictReader(file)

        for row in read:

            if row["id"] == id_barang:

                barang = True

```

```
print(f"Barang ditemukan: {row['nama barang']}, {row['tipe barang']},  
{row['stok barang']}")
```

```
nama_baru = input("Masukkan nama baru: ").capitalize()
```

```
tipe_baru = input("Masukkan tipe baru: ")
```

```
stok_baru = input("Masukkan stok baru: ")
```

```
row["nama barang"] = nama_baru
```

```
row["tipe_barang"] = tipe_baru
```

```
row["stok_barang"] = stok_baru
```

```
print(f>Data barang dengan id:{id_barang} berhasil diperbarui.")
```

```
break
```

```
if barang:
```

```
# Menyimpan data yang telah diperbarui ke dalam CSV
```

```
with open("gudang", mode="w", newline="", encoding="utf-8") as file:
```

```
fieldnames = ["nama barang", "tipe barang", "stok barang", "id"]
```

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

```
writer.writeheader()
```

```
writer.writerow(read)
```

```
print(f>Data {id_barang} berhasil diperbarui.")
```

```
else:
```

```
print(f"Barang dengan id:{id_barang} tidak ditemukan.")
```

```

def hapus_barang():

    try:

        hapus_barang = input("Masukkan ID Barang: ").strip()

        if not hapus_barang:

            raise ValueError("ID barang tidak boleh kosong.")

        barang_dihapus = True

    with open(gudang, 'r', newline="") as file:

        read = csv.DictReader(file)

        data_barang = [row for row in read]

    with open(gudang, 'w', newline="") as file:

        fieldnames = ["nama barang", "tipe barang", "stok barang", "id"]

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

        writer.writeheader()

        for row in data_barang:

            if row["id"] != hapus_barang:

                writer.writerow(row)

            else:

                barang_dihapus = True

    if barang_dihapus:

        print(f"Barang dengan ID {hapus_barang} berhasil dihapus.")

```

```

else:

    print(f"Barang dengan ID {hapus_barang} tidak ditemukan.")


except ValueError as e:

    print(f"error: {e}")


def riwayat_barang():

    try:

        with open(riwayat,'r',newline="",encoding='utf-8') as file:

            baca = csv.DictReader(file)

            data = list(baca)

            if data:

                table = PrettyTable()

                table.field_names = data[0].keys()

                for row in data:

                    table.add_row(row.values())

                print(table)

            else:

                print("\nData kosong.")

    except FileNotFoundError:

        print("File tidak ditemukan, buat data baru terlebih dahulu.")

    except Exception as e:

```

```
print(f"Terjadi kesalahan: {e}")
```

```
if __name__ == '__main__':
```

```
    tambah_barang()
```

```
    lihat_barang()
```

```
    update_barang()
```

```
    hapus_barang()
```

```
    riwayat_barang()
```

Khusus User

```
import csv

import os

from datetime import datetime, timedelta

from prettytable import PrettyTable

gudang = os.path.join(os.path.dirname(__file__), '..', 'gudang.csv')

def lihat_barang():

    try:

        with open(gudang, mode="r", newline="") as file:

            csvreader = csv.DictReader(file)

            data = list(csvreader)

            if data:

                table = PrettyTable()

                table.field_names = data[0].keys()

            else:

                print("\nData kosong.")

            for row in data:

                table.add_row(row.values())

            print(table)

    except FileNotFoundError:

        print("File tidak ditemukan, buat data baru terlebih dahulu.")
```



```

def peminjaman_barang():

    lihat_barang() # Pastikan fungsi ini didefinisikan

    id_barang = input("Masukkan ID Barang: ")

    barang_ditemukan = False

    riwayat_row = None

    # Membaca data dari file gudang

    with open(gudang, mode="r", newline=' ') as file:

        read = csv.DictReader(file)

        rows = list(read)

        for row in rows:

            if row["id"] == id_barang:

                barang_ditemukan = True

                nama_user = input("Masukkan nama peminjam: ").capitalize()

                try:

                    stok_user = int(input("Berapa stok yang ingin dipinjam (maks 10): "))

                    if 0 < stok_user <= 10:

                        jumlah_hari = int(input("Masukkan jumlah hari peminjaman: "))

                        tanggal_peminjaman = datetime.now()

                        print("Tanggal peminjaman:", tanggal_peminjaman.strftime("%Y-%m-%d"))

                        tanggal_pengembalian = tanggal_peminjaman +
timedelta(days=jumlah_hari)

```

```

        print("Tanggal pengembalian:",
tanggal_pengembalian.strftime("%Y-%m-%d"))

# Menambahkan data peminjaman ke dalam row baru untuk riwayat

        riwayat_row = {

            "nama": nama_user,

            "id barang": id_barang,

            "stok": stok_user,

            "waktu awal peminjaman": tanggal_peminjaman.strftime("%Y-
%m-%d"),

            "waktu akhir peminjaman":
tanggal_pengembalian.strftime("%Y-%m-%d")

        }

# Menambahkan data ke riwayat.csv jika barang ditemukan

        with open('riwayat.csv', 'a', newline="", encoding="utf-8") as
file_riwayat:

            fieldnames = ["nama", "id barang", "stok", "waktu awal
minjam", "waktu akhir peminjaman"]

            writer = csv.DictWriter(file_riwayat, fieldnames=fieldnames)

            writer.writerow(riwayat_row)

            print(f>Data barang dengan ID: {id_barang} berhasil dipinjam.")

            break

    else:

        print("Peminjaman barang tidak boleh melebihi 10 stok")

```

```

except ValueError:

    print("Input tidak valid, harus berupa angka")

if not barang_ditemukan:

    print(f"Barang dengan ID: {id_barang} tidak ditemukan.")

if __name__ == "__main__":

    lihat_barang()

    peminjaman_barang()

```

Loading

```

import time

import sys

def loading_efek():

    for i in range(3): # Melakukan iterasi sebanyak 3 kali

        sys.stdout.write("Loading" + "." * (i + 1)) # Tampilkan "Loading", titiknya
        bertambah

        sys.stdout.flush() # Memastikan bahwa output langsung ditampilkan

        time.sleep(0.5) # Delay selama 0.5 detik

        sys.stdout.write("\r") # Menghapus teks sebelumnya (menggantinya dengan
        baris baru)

    print("Kembali ke menu utama!") # Setelah selesai, tampilkan "Done!"

if __name__ == "__main__":

    loading_efek()

```

Main.py Kode Utama Program

```
import os

from tools import loading

from auth import login, register


# start menu

def menu():

    while True:

        print("\nSelamat Datang di E-GUDANG ELEKTRONIK\n 1.Register\n 2.Login\n 3.Exit ")

        try:

            opsi = int(input("\nPilih opsi: "))

            os.system('cls || clear')

            if opsi == 1:

                register.register()

                input("Tekan enter untuk kembali ke menu utama")

                loading.loading_efek()

                os.system('cls || clear')

            elif opsi == 2:

                login.login()

                loading.loading_efek()

                os.system('cls || clear')

            elif opsi == 3:
```

```

        print("Terima kasih telah mengunjungi E-GUDANG\nProgram akan
keluar")

        loading.loading_efek()

        os.system('cls || clear')

        break

    else:

        print("Opsi tidak valid. Silakan pilih 1,2 dan 3")

    except ValueError:

        print("Input tidak valid. Harap masukkan angka.")

    except KeyboardInterrupt:

        print("\nKeluar dari program secara paksa.")

        break

if __name__ == "__main__":

    menu()

```

BAB IV

PENUTUP

4.1 Kesimpulan

Program yang kami rancang ini untuk mengatasi permasalahan banyak pemilik jasa inventaris gudang elektronik di luar sana dalam mendata peminjaman yang dilakukan oleh orang banyak. Hal ini dapat mempermudah pemilik jasa inventaris untuk menjalankan pekerjaannya. Program ini dibuat juga untuk peningkatan keandalan data yang menunjukkan bagaimana program dapat menyediakan data *real-time* yang dapat diandalkan untuk mendukung pengambilan keputusan. Dibuat untuk evaluasi sistem dalam menyediakan alat untuk mengevaluasi efektivitas program, identifikasi kelemahan, dan area untuk perbaikan di masa depan.

4.2 Saran

Meskipun program ini masih terbilang sangat sederhana namun pada jalannya program sudah menerapkan beberapa materi dan modul yang berkaitan dengan praktikum python yang selama ini sudah kami jalani, saran nantinya rancangan program yang dibuat ini dapat bermanfaat bagi pengguna di masa yang akan mendatang dan untuk pengembangan system yang sudah ada selanjutnya. Serta diharapkan tampilan dalam program bisa lebih divariasikan lagi dan program akan lebih berkembang.

DAFTAR PUSTAKA

Noorizki, A. Z. (2023). Studi Independen: Rancang Bangun Sistem Pengajuan Pengadaan Inventaris Berbasis Website. *Studi Independen: Rancang Bangun Sistem Pengajuan Pengadaan Inventaris Berbasis Website. Undergraduate thesis, Universitas Dinamika.*, 50.

<https://blog.unmaha.ac.id/penanganan-error-dan-exception-di-python/>

<https://revou.co/panduan-teknis/try-except-python>

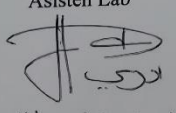
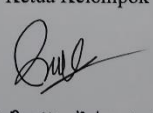
<https://rahmadya.com/2020/05/15/memanggil-fungsi-eksternal-pada-python/>

<https://dqlab.id/library-python-kenali-perbedaan-module-package-dan-library-pada-python>

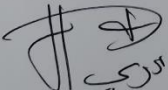

LAMPIRAN



Gambar 4.1 Halaman pertama kartu konsul

Tanggal Konsultasi : 31-Oktober-2024	
<p>Uraian / Pembahasan : User harus dijelaskan secara spesifik lagi sebagai Pegawai atau sebagai siapa Admin lebih baik ditetapkan saja usernamanya dan passwordnya, kalau admin ingin lebih dari satu, dibuatkan lebih lagi aksesnya.</p> <p>Batasin konsep programnya. Password dan Username sebaiknya dibuat Variabel Statis saja. Kalau ada peminjaman dari user sebaiknya stok barang berkurang. Admin mendata barangnya, karyawan / user cuman bisa melihat atau melakukan peminjaman saja. Di menu user bisa juga melihat riwayat penyimpanan barang seperti admin.</p> <p>Ingat!! Batasin konsep programnya biar tidak keterusan dan buat ribet.</p>	
<p>Asisten Lab</p>  <p>Nama: Adhiti Murni Ayo</p>	<p>Ketua Kelompok</p>  <p>Nama: Rafly Rahmat W.A.S</p>

Gambaar 4.2 Pemaparan konsultasi pertama

Tanggal Konsultasi : 14 - November - 2024	
<p>Uraian / Pembahasan : Untuk fitur khusus admin: Penomoran tidak masalah. Penambahan kode itu otomatis ketambah data baru dan id baru. Nambah nya dibagian tambah data ke file csv. Selain tambah data ke id / nomer nya.</p> <p>Untuk fitur khusus user: Menu tidak perlu lihat alat lagi jadi cuman ada dua tambah dan riwayat (saran).</p> <p>Bisa pakai try except kalau mau di pisah bisa, jadi bisa import dari file luar, tapi harus tau kalau ditanya. Saran dari mba adri lebih baik dan lebih enak nya dipisah saja agar lebih mudah cari eror nya dimana.</p> <p>Enaknya dibuat folder baru saja, misalnya nama nya login, karena kita menggunakan dua file itu dari satu folder yang sama, menghemat line, dan mempersingkat kode juga.</p> <p>Bagusin / Perbaiki di bagian Eror handling saja, tampilan hanya nilai plus.</p>	
Asisten Lab	Ketua Kelompok
	
Nama: Adhiti Manuk Allo	Nama: Rafly Rahmat W.A.S

Gambar 4.3 Pemaparan konsultasi kedua