Program Mesin ATM Menggunakan Python dan Modul Eel untuk Antarmuka Pengguna Grafis

|  |  |
| --- | --- |
| Marchotridyo STEI Institut Teknologi Bandung Bandung, Indonesia acoxstpd@gmail.com | Willy Wilsen STEI Institut Teknologi Bandung Belitung Timur, Indonesia willywilsen.ww@gmail.com |
| Divya Maharani Lazuardi STEI Institut Teknologi Bandung Jakarta Timur, Indonesia divyamaharani21@gmail.com | Wervyan Shalannanda, S.T., M.T. KK Teknik Telekomunikasi, STEI Institut Teknologi Bandung Bandung, Indonesia wervyan@office.itb.ac.id |

*Abstract*—ATM (*Automated Teller Machine*) adalah salah satu mesin yang sering kita gunakan untuk keperluan kita sehari-hari. Alat ini memberikan sarana untuk melakukan transaksi finansial tanpa perlu melalui interaksi langsung dengan staf bank. ATM dapat menerima intruksi untuk mengambil uang, menyimpan uang, mentransfer uang, dan juga untuk mengecek saldo. Program yang ditampilkan pada *paper* ini menyimulasikan semua intruksi tersebut.

Keywords—ATM, mesin, intruksi, mengambil uang, menyimpan uang, mentransfer uang, simulasi

# Introduction

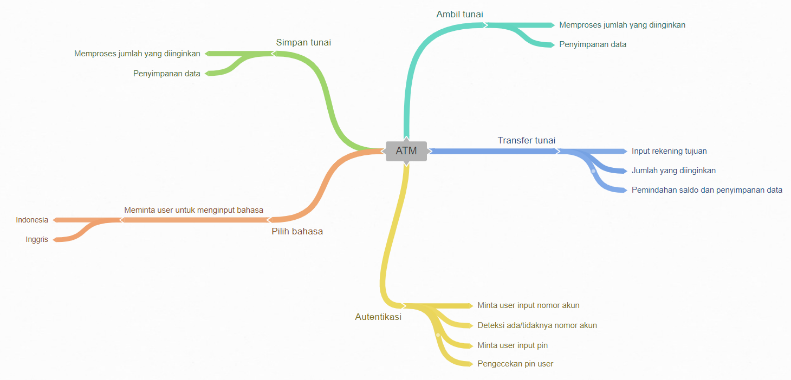
ATM (*Automated Teller Machine*) adalah suatu perangkat elektronik yang membantu konsumer untuk bertransaksi finansial tanpa perlu membutuhkan interaksi langsung dengan staf bank. Transaksi dapat berupa pengambilan uang, penyetoran uang, pentransferan uang, dan pengecekan akun pengguna.

Latar belakang kami memilih ATM sebagai pilihan tugas besar karena cara kerjanya yang cukup simpel dan visualisasi antarmukanya dapat dibuat dengan jelas. Operasi matematika yang dilakukan pada ATM hanya sekadar pertambahan dan pengurangan.

Video mengenai pembuatan ATM menggunakan Python pun sudah banyak bertebaran di internet. Namun, pada program ini lebih dikedepankan metode programming prosedural (tidak berorientasi objek) dan antarmukanya menggunakan Eel, suatu modul yang memungkinkan untuk membuat antarmuka ala Electron JS dengan memanfaatkan HTML/JS.

# System Overview

Dekomposisi sistem merupakan salah satu hal yang paling penting dalam perancangan suatu projek. Dekomposisi ini dilakukan dengan cara memecah suatu masalah yang besar, dalam kasus ini suatu mesin ATM, menjadi komponen-komponen penyusunnya yang lebih kecil, seperti operasi pengambilan uang, operasi penyetoran uang, operasi pentransferan uang, dan operasi pengecekan akun pengguna.

Garis besar dari dekomposisi sistem ATM ini ditunjukkan pada *mind-map* di bawah ini:

Kami bagi proses-proses pada ATM menjadi 5 bagian:

### Autentikasi: Menyangkut nomor akun pengguna dan verifikasi pin pengguna untuk diproses sebelum pengguna dapat bertransaksi.

### Pemilihan bahasa: Menyangkut pemrosesan input pengguna untuk mengatur bahasa antarmuka program.

### Penyimpanan tunai: Menyangkut proses penyimpanan tunai yang meliputi pengecekan input yang pengguna berikan diikuti dengan penyimpanan data.

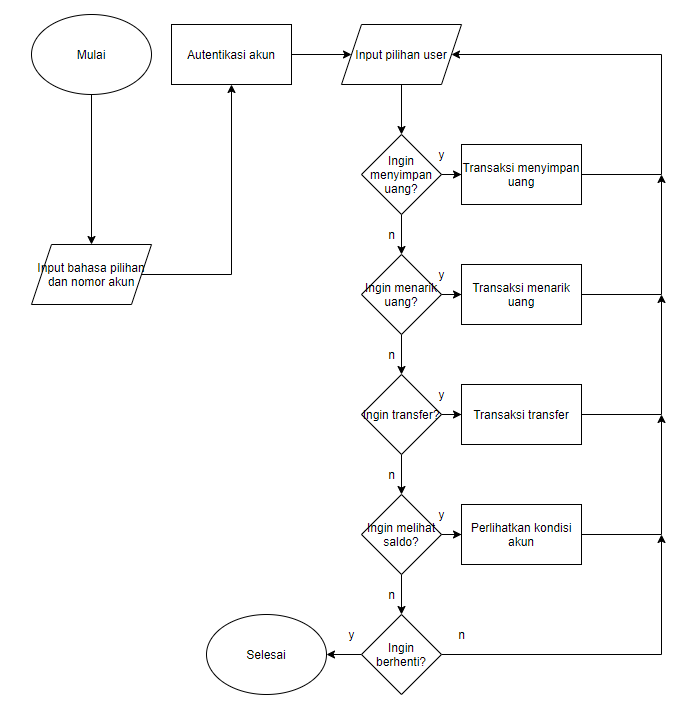
### Pengambilan tunai: Menyangkut proses pengambilan tunai yang meliputi pengecekan input yang pengguna berikan diikuti dengan penyimpanan data.

### Pentransferan tunai: Menyangkut proses transfer tunai yang meliputi pengecekan input yang pengguna berikan, pemutakhiran saldo kedua rekening, dan penyimpanan data.

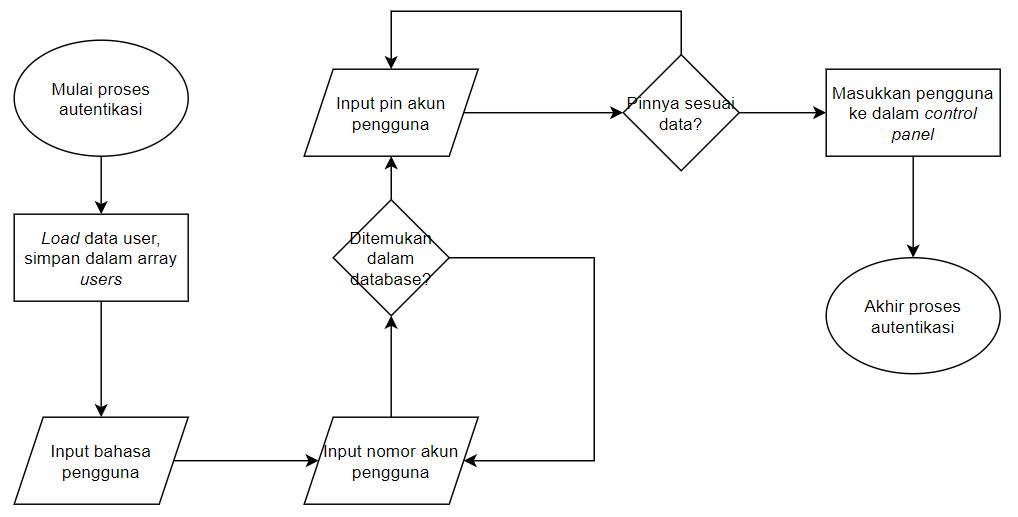
Cara kerja kelima poin yang telah disebutkan di atas akan dijelaskan secara mendetail pada bagian III.

# Proposed System

Secara umum, sistem bekerja mengikuti *flowchart* di bawah ini. Paragraf-paragraf selanjutnya akan menjelaskan secara detail masing-masing subproses yang dilakukan program saat berjalan.

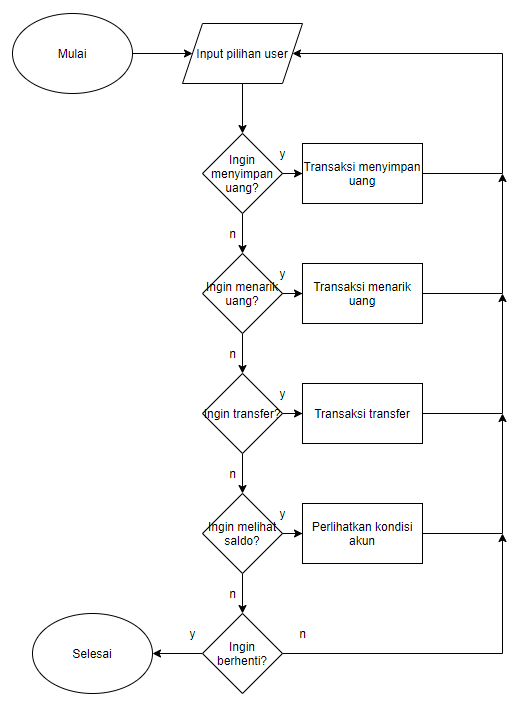


Sistem dimulai dari proses autentikasi. Program meminta pengguna untuk memasukkan nomor akun (sebagai pengganti sensor kartu) lalu sistem akan melakukan pengecekan pada *database*. Apabila nomor akun valid, pengguna akan diperintahkan untuk memasukkan nomor pinnya. Nomor pin yang valid akan membuat program membuka bagian *control panel* di mana pengguna dapat memilih transaksi yang ingin dilakukan.

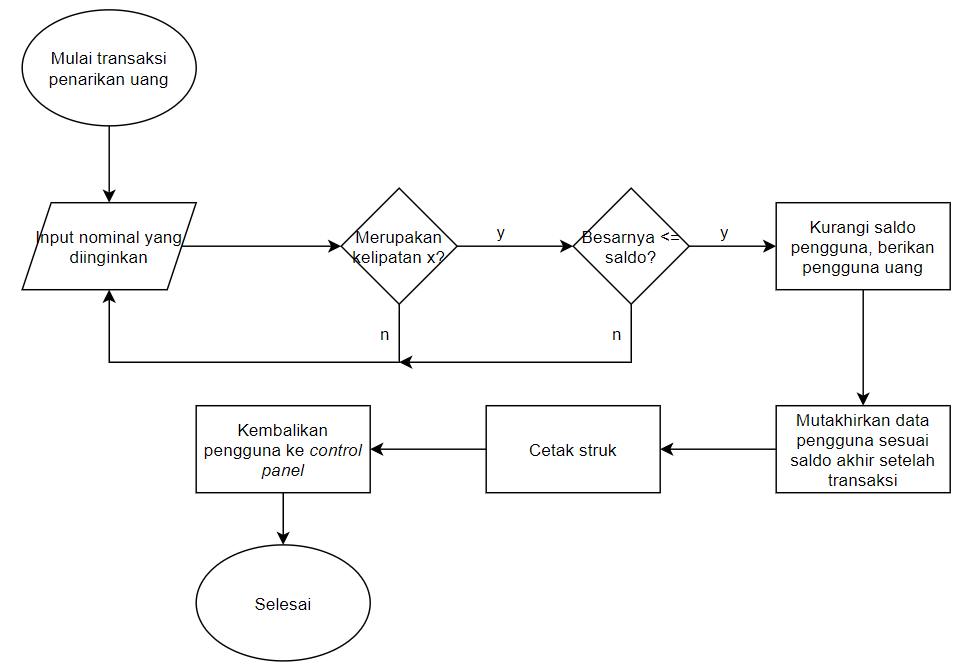


Lalu, pengguna akan melihat laman *control panel*. Di laman ini, pengguna dapat memilih transaksi sesuai yang pengguna inginkan. Transaksi yang dapat dipilih berupa

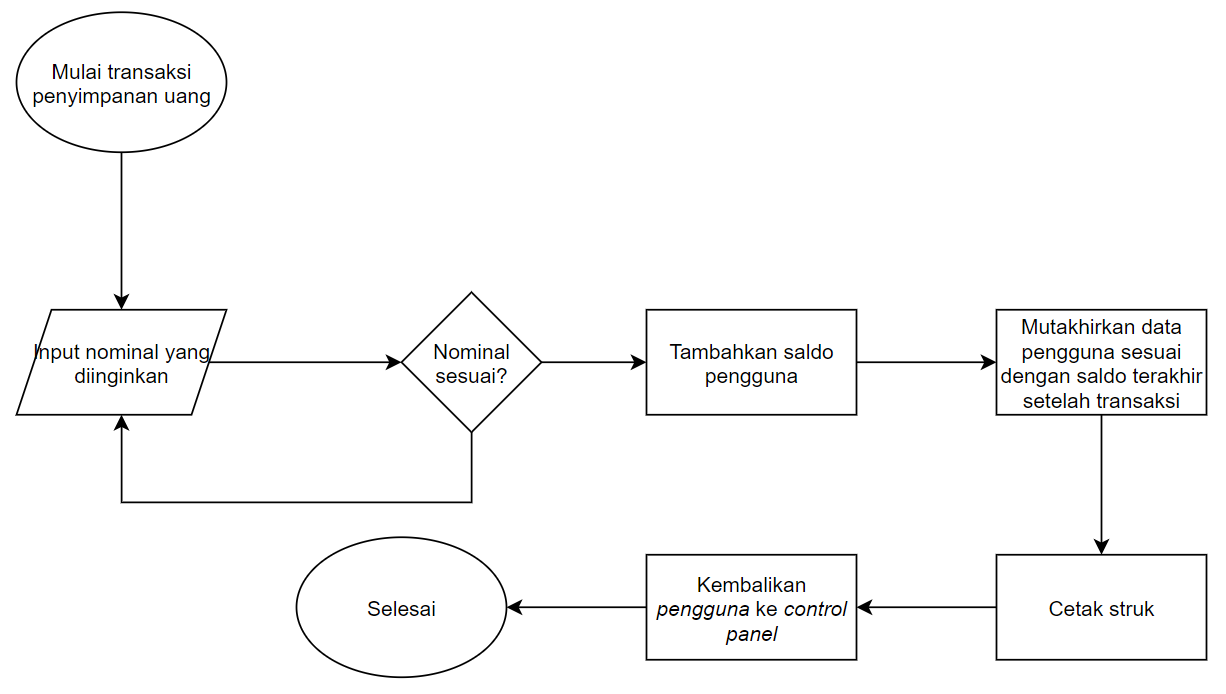
* menyimpan uang,
* menarik uang,
* mentransfer uang,
* melihat saldo, dan
* menghentikan transaksi.



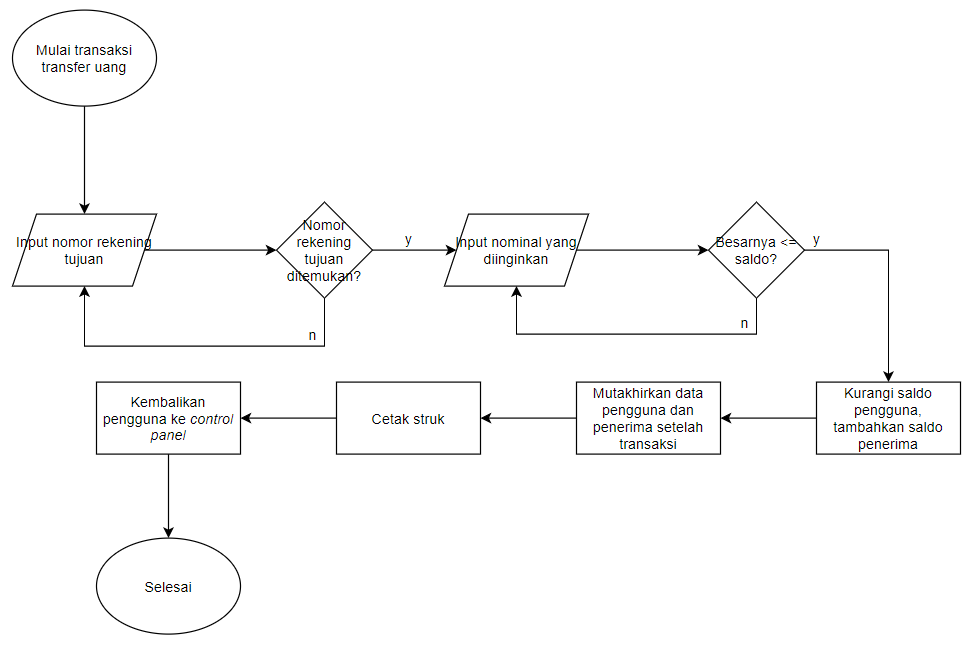
Proses penarikan uang cukup sederhana. Program meminta pengguna untuk menginput jumlah nominal yang pengguna inginkan. Apabila nominal uang memenuhi kriteria: (1) merupakan kelipatan dari nominal yang disediakan ATM dan (2) besarnya kurang atau sama dengan saldo yang dimiliki pengguna, pengguna akan diberikan uang yang ia inginkan. Saldo pengguna dikurangi sebesar penarikan yang ia lakukan. Program lalu merekam data pengguna setelah transaksi, lalu mencetak struk bagi pengguna. Pengguna dikembalikan ke laman *control panel*.



Sama halnya dengan penarikan uang, proses penyimpanan uang sederhana juga. Prosesnya tidak jauh berbeda dengan penarikan uang seperti ditunjukkan pada *flowchart* di bawah ini.

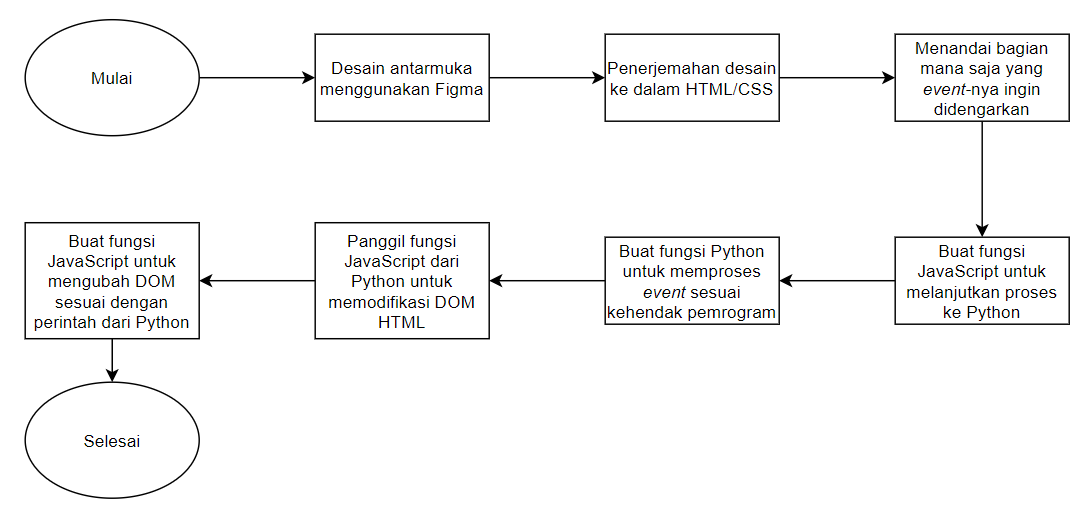


Proses pentransferan tidak jauh berbeda dari proses penarikan uang. Hal yang membedakan hanyalah pada proses ini diperlukan dua input pengguna: rekening tujuan dan nominal yang diinginkan. Saldo pengguna dikurangi sebanyak nominal yang ia inginkan dan saldo penerima ditambahkan dengan besar yang sama. Pemutakhiran *database* dilakukan, struk dicetak, lalu pengguna dikembalikan ke *control panel*.



Proses penunjukan saldo pengguna hanya berupa pencetakan struk (*flowchart* tidak diberikan) dan proses penghentian transaksi hanya berupa proses mengeluarkan pengguna dari program (*flowchart* tidak diberikan).

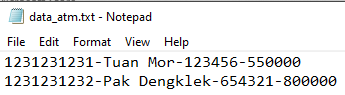
Antarmuka Pengguna Grafik disusun menggunakan bantuan modul Eel. Antarmuka berbasis HTML/CSS/JavaScript dengan Python yang berperan sebagai bagian *back-end*-nya. *Event-event* yang terjadi pada laman HTML ditangkap oleh JavaScript untuk lalu diteruskan ke Python. Begitu pula sebaliknya, manipulasi DOM (*Document Object Model*) dilakukan oleh JavaScript yang menerima perintah dari Python. Proses pembuatannya kurang lebih mengikuti *flowchart* di bawah ini.



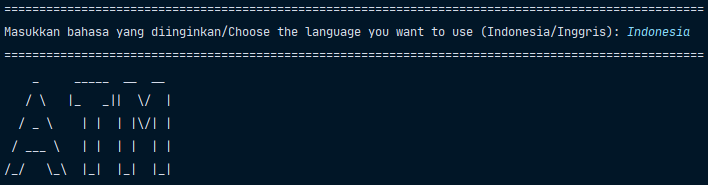
# Results and Discussion

Kami telah menghasilkan dua program simulasi ATM: berbasis terminal (tanpa antarmuka) dan versi yang memiliki antarmuka berbasis HTML/CSS/JS. Kedua program berjalan atas algoritma yang sama dan proses algoritmanya diproses dengan bahasa Python.

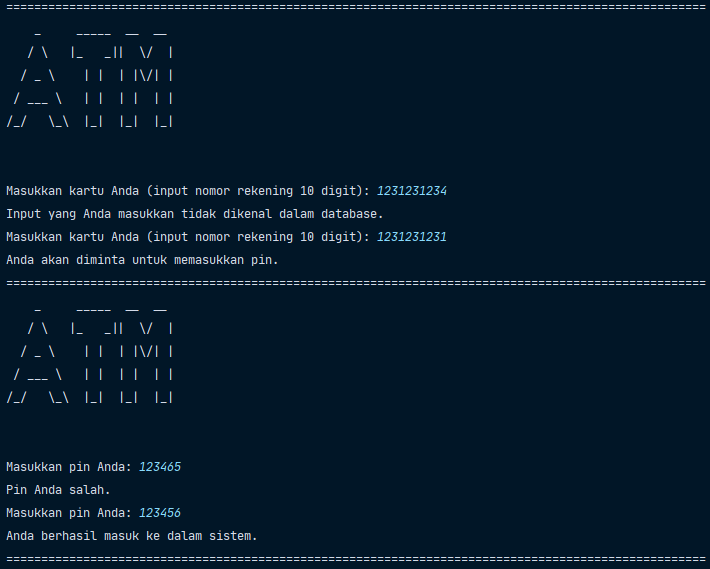
Pada kedua program, data pengguna disimpan dalam bentuk .*txt* yang mengikuti format REKENING-NAMA-PIN-SALDO seperti pada gambar di bawah. Data disimpan dalam bentuk *array*.



Digunakan modul *Art* dalam produk versi terminal untuk mempermudah pembuatan ASCII Art sebagai dekorasi output yang diberikan program.



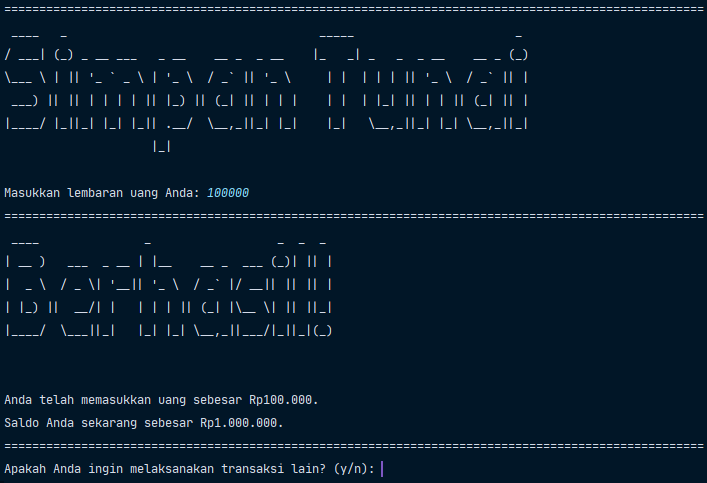
Sistem autentikasi ATM pada produk versi terminal dapat dilihat pada gambar di bawah ini.



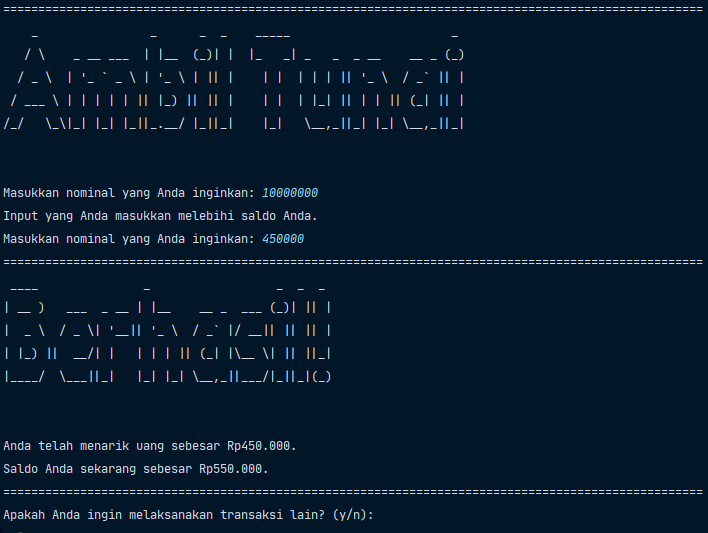
*Control panel*-nya berupa *prompt* bagi pengguna untuk memilih transaksi apa yang ingin pengguna lakukan.



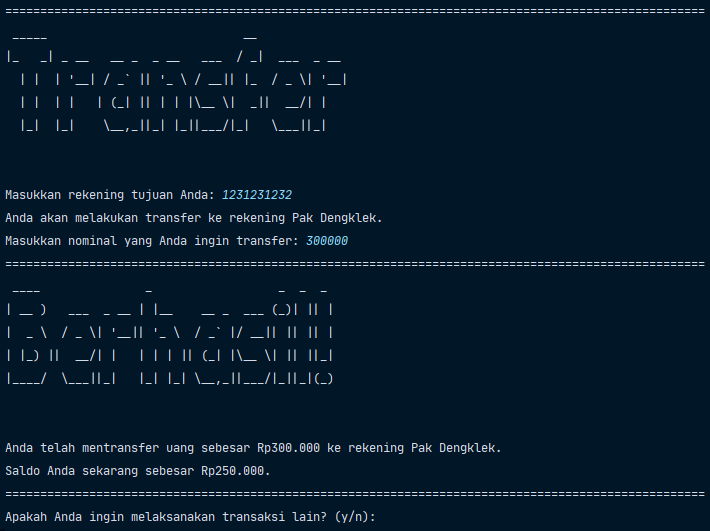
Bagian transaksi simpan tunai dapat dilihat pada gambar di bawah.



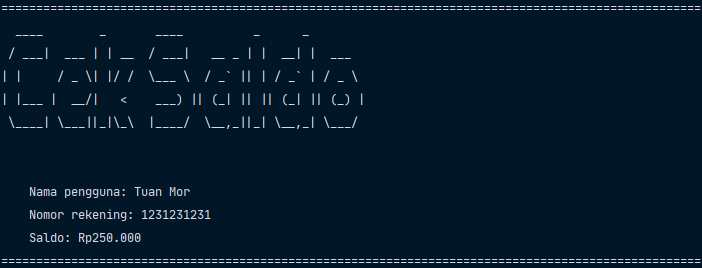
Hal yang serupa dapat dilihat dalam transaksi ambil tunai pada gambar di bawah.



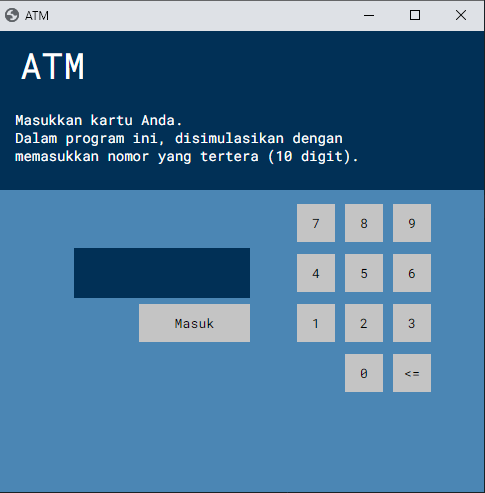
Untuk laman transfer tunai, pengguna diberitahu dengan jelas kepada siapa ia akan mentransferkan uangnya.

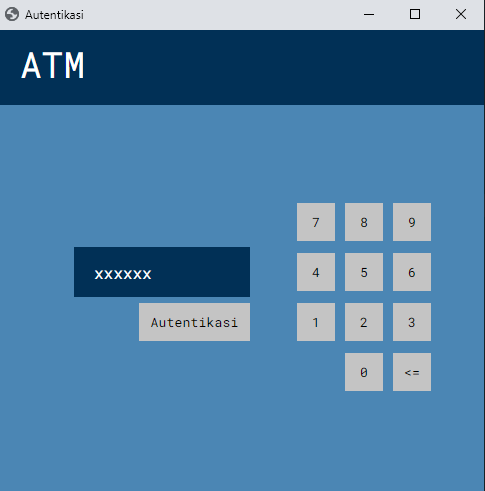


Laman cek saldo sangat sederhana seperti ditunjukkan pada gambar di bawah.

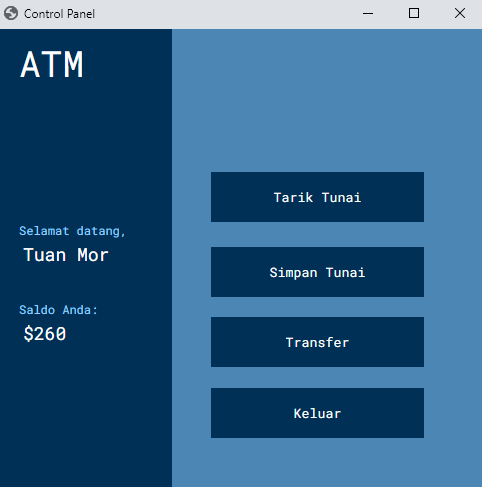


Pada program versi antarmuka, input dimasukkan menggunakan tombol, bukan keyboard. Berikut kedua laman ditunjukkan laman autentikasi. Sebagai catatan tambahan, antarmuka menggunakan satuan dollar untuk menghemat *space*.

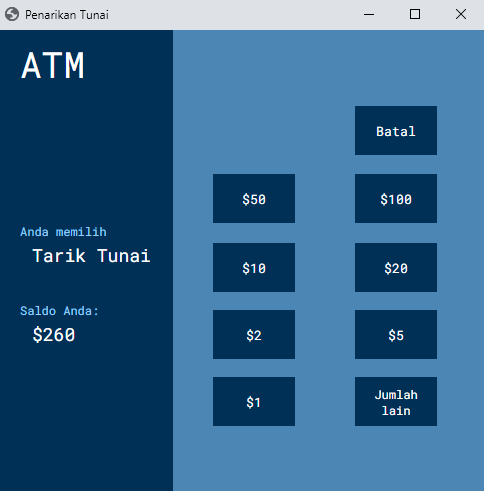


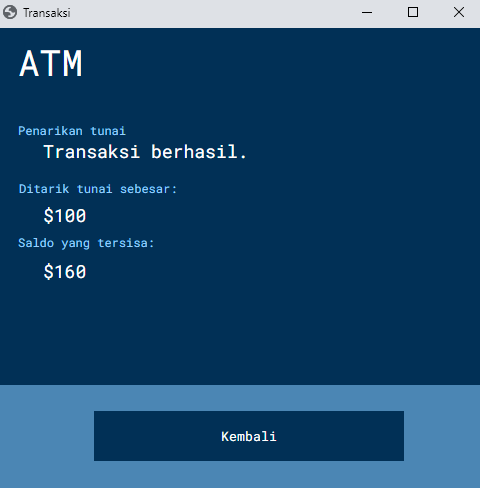


Laman *control panel*-nya seperti pada gambar di bawah.

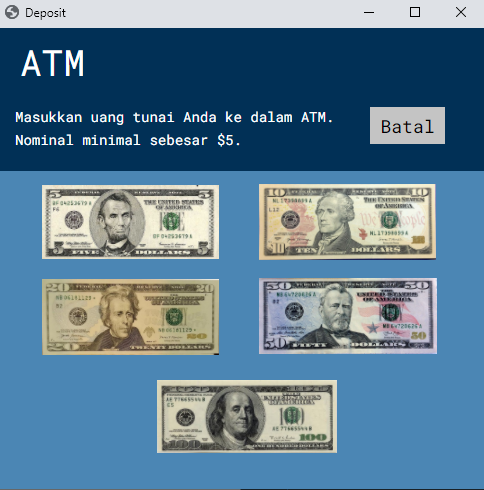


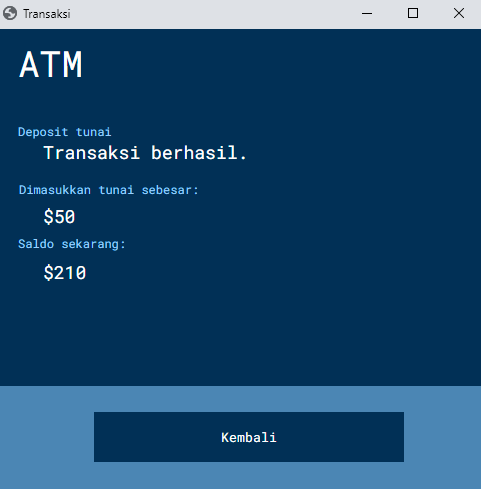
Laman tarik tunai dan struk yang dihasilkannya ditampilkan pada gambar di bawah.



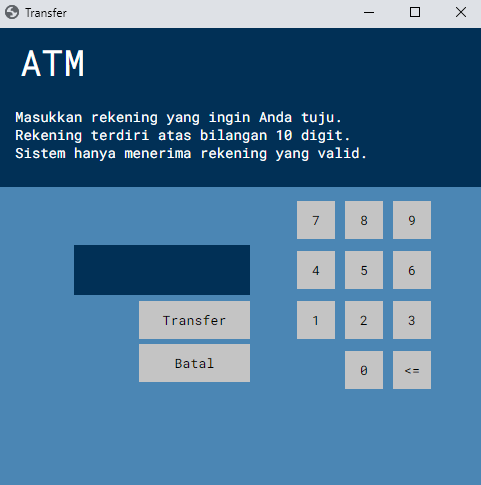


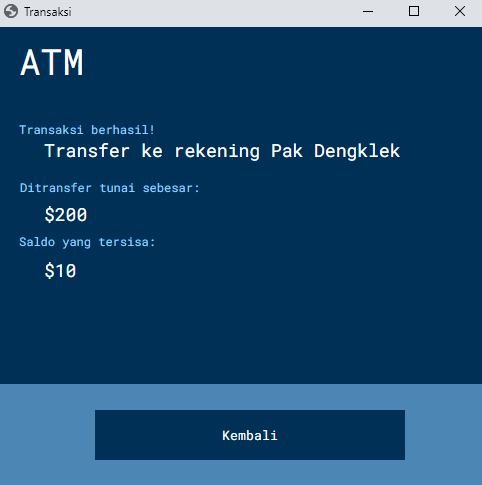
Berikutnya adalah laman penyimpanan tunai dan struk yang dicetaknya.





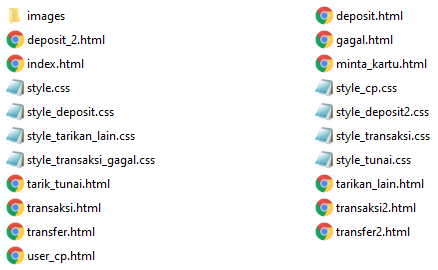
Halaman transfer mirip dengan halaman penarikan tunai, namun ada halaman permintaan rekening dan struk yang dicetak pun berbeda.





Penyimpanan data mungkin lebih baik menggunakan *database* seperti Pickle, Spreadsheet, ataupun JSON. Namun, karena ini merupakan simulasi yang sederhana, kami memilih untuk menggunakan .txt dengan bantuan *string slicing* untuk menyimpan dan membaca data pengguna.

Program versi terminal terdiri atas 376 baris kode, sedangkan program versi antarmuka memiliki banyak subprogram berupa laman HTML dan styling CSS yang jumlah baris kodenya sekitar 2000 baris kode.



# Conclusion

Cara kerja mesin ATM yang merupakan mesin untuk membantu manusia bertransaksi finansial tanpa harus berinteraksi langsung dengan staf bank dapat disimulasikan dengan sebuah program sederhana. Program ini menggunakan mayoritas hanya ekspresi aritmetika sederhana berupa penjumlahan dan pengurangan. Sistem *database* menggunakan *array* yang datanya disimpan dalam file *.txt* sederhana.

Terdapat alternatif antarmuka lain untuk Python selain yang sering digunakan khalayak ramai seperti Tkinter, PyQt5, PySide2, ataupun hal seperti Django dan Flask. Salah satunya adalah Eel yang memberikan suatu *framework* bagi developer untuk membuat GUI seperti menggunakan Electron JS menggunakan bahasa Python.

##### References

1. S. H. Willliams. “Eel”. Diakses November 2020 dari laman https://github.com/samuelhwilliams/Eel.
2. E. Matthes. *Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd Edition.* San Francisco: No Starch Press, 2019.

# Pembagian Tugas dalam Kelompok

1. Marchotridyo: Konsep, *pseudocode*, *coding* program terminal dan program dengan GUI, pembuatan laporan, pembuatan presentasi, dan video presentasi.
2. Willy Wilsen: Konsep, *pseudocode,* pemeriksaan program, laporan, dan presentasi.
3. Divya Maharani Lazuardi: Konsep, *pseudocode*, pemeriksaan program, laporan, dan presentasi.