

Tugas Netdev Python #3

Pre-Task:

1. Buatlah *repository* github dengan format [pythonNetdev23_namakelompok]. Gunakan git untuk pemahaman konsep lebih baik (*kami nilai lebih baik*).
2. Undang seluruh anggota kelompok dan asisten menjadi *collaborator*. Berikut *email* asisten riset yang harus diundang:
 - metihervina@student.telkomuniversity.ac.id
 - shafira.r.febriyanti@gmail.com
 - novanpurba411@gmail.com
 - arigints@student.telkomuniversity.ac.id
 - melonna271202@gmail.com
 - salwa_aulia_27rpl@student.smktelkom-mlg.sch.id
 - rashidfajri@gmail.com
3. Setelah tugas diberikan buatlah *branch* sesuai jumlah tugas dan kerjakan tugas pada *branch* yang terpisah. Misal *branch* dengan nama “Tugas 1” khusus untuk mengerjakan tugas codingan tugas 1.
4. Kerjakan secara berkelompok dengan cara remote branch!
5. Buat *flowchart* cara kerja dari masing-masing kodingan!
6. Presentasikan hasil pengerjaan tugas kalian saat sesi presentasi demo minggu depan.

Tools:

1. [Python 3.10 or newer](#)
2. Text Editor: dibebaskan, tapi disarankan [Visual Studio code](#)
3. [Git](#)

Task 1:

Deskripsi Tugas

1. Menggunakan konsep OOP
2. Anda diminta untuk membuat sebuah program sederhana yang dapat melakukan transaksi pada sebuah rekening bank. Anda akan membuat sebuah class Bank yang memiliki beberapa method, yaitu:
 - a. `init(self, nama)`: method ini akan dijalankan ketika sebuah objek Bank dibuat. Method ini akan menginisialisasi saldo awal pada rekening (sebesar 1.000.000) dan nama pemilik rekening.
 - b. `Cek_saldo(self)`: method ini akan menampilkan saldo pada rekening.
 - c. `Transfer(self, tunai)`: method ini akan mentransfer sejumlah tunai dari rekening ke rekening lain. Method ini akan memanggil method `Kurang_saldo(self, tunai)` untuk mengurangi saldo pada rekening pengirim dan menampilkan pesan sukses jika saldo mencukupi, atau pesan gagal jika saldo tidak mencukupi.
 - d. `Setor(self, tunai)`: method ini akan melakukan setoran tunai ke rekening dan menampilkan pesan sukses beserta saldo terbaru setelah setoran.
 - e. `Tambah_saldo(self)`: method ini akan menambahkan jumlah tunai ke saldo pada rekening dan menampilkan pesan sukses beserta saldo terbaru.
 - f. `Kurang_saldo(self, tunai)`: method ini akan mengurangi saldo pada rekening dengan sejumlah tunai dan menampilkan pesan sukses beserta saldo terbaru jika saldo mencukupi, atau pesan gagal jika saldo tidak mencukupi.
3. Selanjutnya, Anda akan membuat sebuah objek Bank dengan nama "Robin" dan melakukan beberapa transaksi, yaitu:
 - a. Menampilkan saldo awal pada rekening "Robin".
 - b. Melakukan transfer sebesar 700.000 dari rekening "Robin" ke rekening lain.
 - c. Melakukan setoran tunai sebesar 100.000 ke rekening "Robin".
 - d. Melakukan transfer sebesar 500.000 dari rekening "Robin" ke rekening lain.