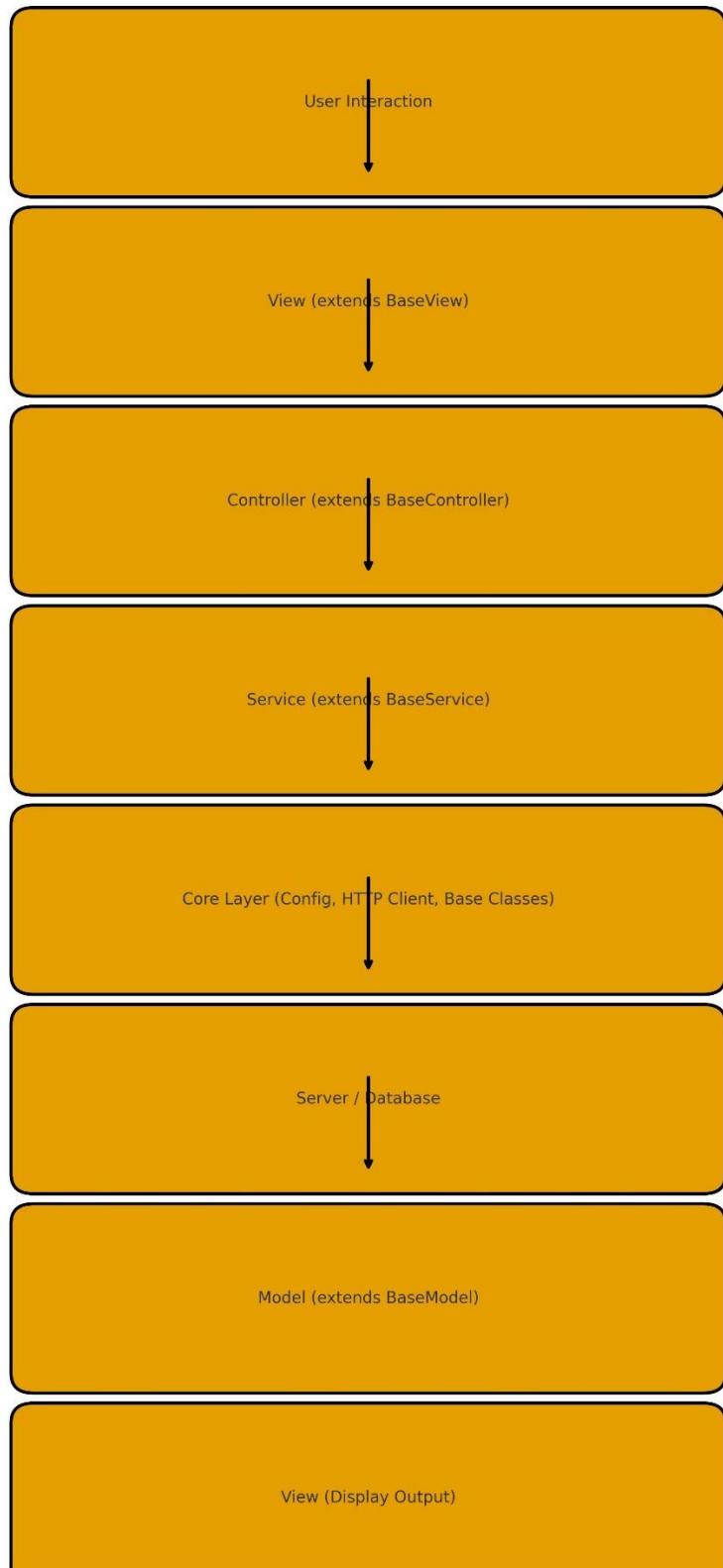
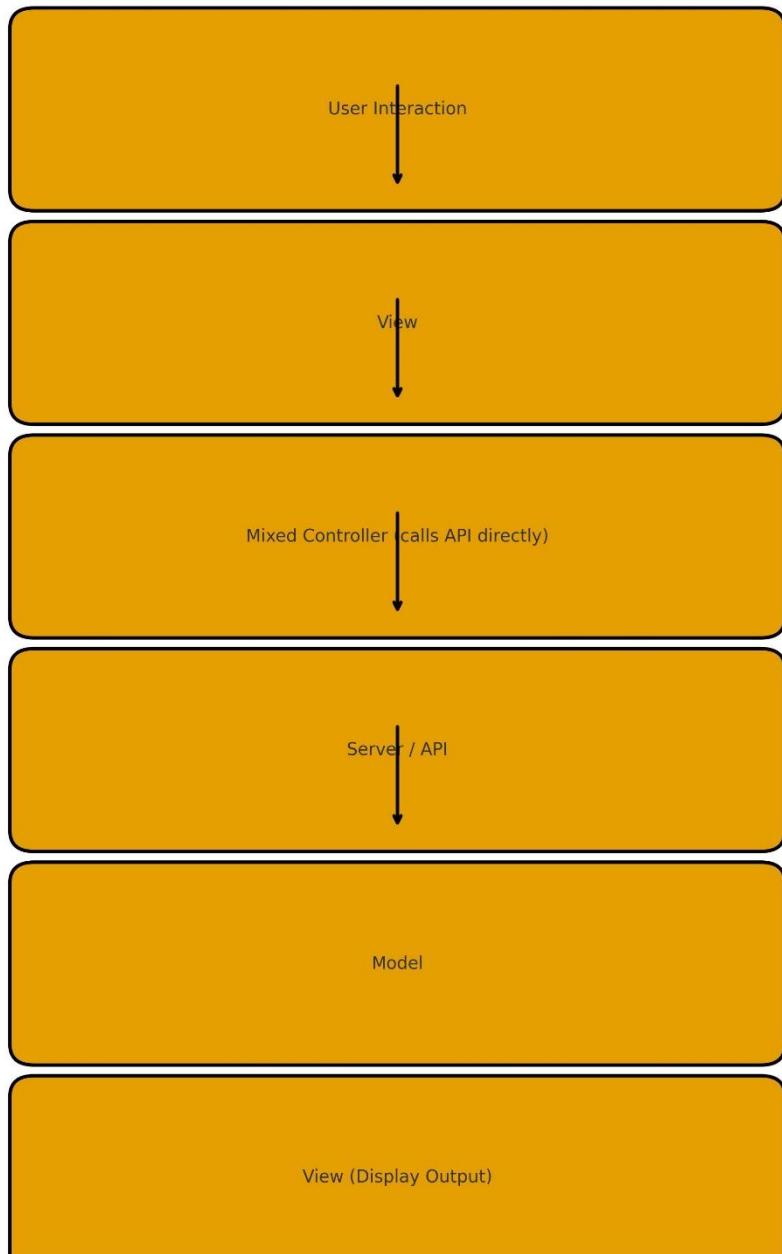


Flowchart Setelah Restruktur (Dengan Base Classes)



Flowchart Sebelum Restuktur



PERUBAHAN PALING BESAR SETELAH RESTRUKTUR (DENGAN BASE CLASS)

1. Aplikasi kini mempunyai “aturan inti” yang mengikat semua modul

Kami menambahkan **empat Base Class** di folder core/:

- BaseController
- BaseView
- BaseService
- BaseModel

Keempatnya berfungsi sebagai:

Fondasi utama

yang mengatur bagaimana *setiap file* di seluruh modul harus bekerja.

Ini membuat semua modul sekarang **konsisten, rapi, dan ikut aturan arsitektur yang sama**.

2. Semua modul sekarang diwariskan dari Base Class

Contoh:

```
class TransaksiController(BaseController):
```

```
    def __init__(self):  
        super().__init__()
```

Dampaknya:

- Tidak ada controller liar yang berbeda dari modul lain
- Semua modul berjalan dengan **format dan aturan yang sama**
- Perubahan aturan di BaseClass langsung turun ke semua modul

Contoh benefit:

Jika nanti Anda ingin menambahkan debug logging ke semua controller, cukup modifikasi 1 file: BaseController.
Semua modul otomatis ikut upgrade.

3. Pemisahan Tugas Menjadi Sangat Jelas

Dengan base class:

Layer	Tugas	Tidak Boleh
View	UI & event	Memanggil API
Controller	Mengatur alur	Simpan IP / logika API
Service	Logika bisnis + API	Sentuh UI
Model	Struktur data	Akses server

Ini membuat aplikasi:

- jauh lebih stabil
 - mudah diperbaiki
 - tidak mudah kacau
 - cocok untuk tim besar
-

4. CORE sekarang menjadi pusat kekuatan aplikasi

Folder core/ berisi:

- konfigurasi global
- Base Classes
- http client
- logger
- db helper
- device utils

Sehingga:

kalau IP server berubah,
cukup ubah satu file → seluruh aplikasi ikut menyesuaikan.

Tidak ada lagi:

- IP tertanam di controller
 - URL tercecer di 20 file berbeda
-

5. Modul benar-benar bersih & mandiri

Contoh modul penjualan/:

controllers/

models/

services/

views/

widgets/

Dengan BaseClass:

Sebuah modul selalu memiliki controller, service, model, dan view yang **selaras** dan **koheren**.

3. PERBEDAAN SEBELUM vs SESUDAH

Aspek	Sebelum	Sesudah
Struktur	Campur, tidak teratur	Modular per fitur
Konsistensi	Berbeda-beda	Seragam karena BaseClass
IP Server	Hardcoded di banyak file	Centralized di core/config_utils
Controller	Berantakan	Terkontrol via BaseController
Service	Tidak terstandarisasi	Semua ikut BaseService
Model	Bebas bentuk	Tertata via BaseModel
View	Bercampur logika	Bersih → event saja
Skalabilitas	Rendah	Sangat tinggi
Kemudahan Dev	Sulit	Mudah & cepat
Maintainability	Tinggi resiko error	Sangat aman
Clean Code	Tidak terjamin	Dijamin oleh BaseClass

Sebelum dilakukan restruktur, aplikasi tidak memiliki standar umum. Setiap file berdiri sendiri dan sulit diprediksi.

Setelah restruktur, kami menambahkan empat Base Class utama di dalam folder Core.

BaseClass ini menjadi landasan arsitektur, sehingga seluruh controller, view, service, dan model di setiap modul wajib mengikuti aturan yang sama. Dengan cara ini, semua modul menjadi konsisten, modular, mudah dipelihara, dan sangat scalable.

Jika aturan di BaseClass diperbarui, seluruh modul aplikasi akan otomatis ikut meningkat kualitasnya.