

LAPORAN PRAKTIKUM

“Praktikum 7: Class Diagram (Kelas & Atribut)”

Diajukan untuk memenuhi salah satu praktikum Analisis Perancangan Perangkat Lunak yang
di ampu oleh:

Arfiani Nur Khusna, M.Sc.



Disusun Oleh:

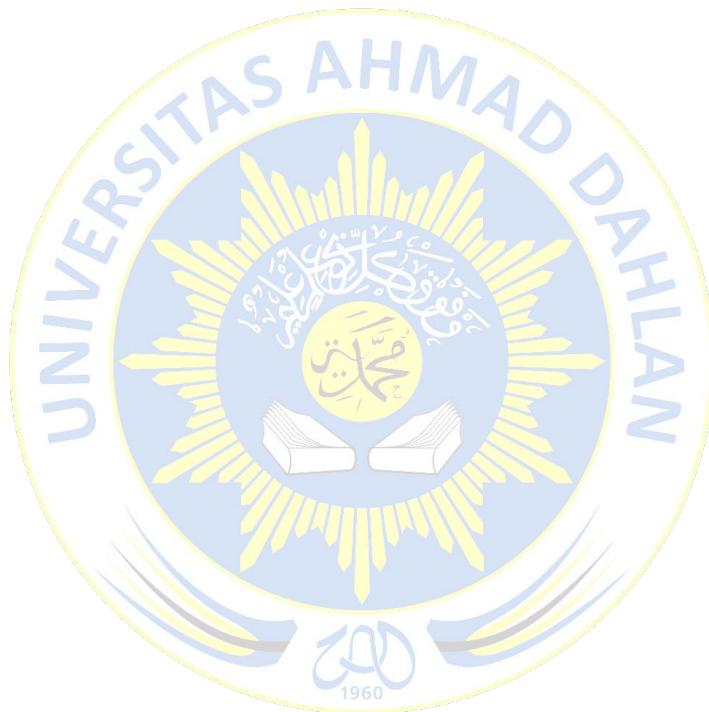
Mohammad Farid Hendianto 2200018401

A / Sabtu 16:45 Lab. Basis Data

PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

DAFTAR ISI

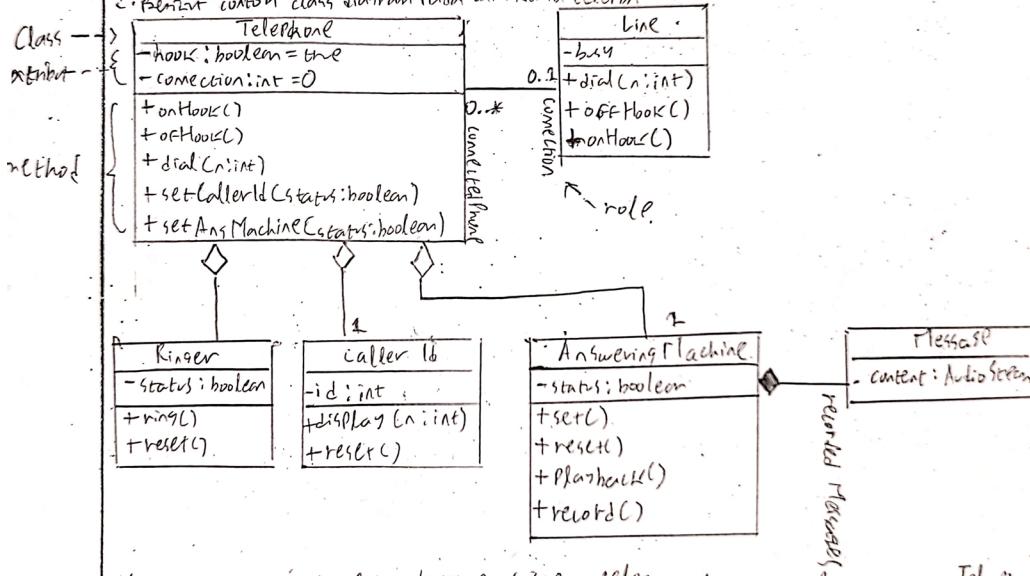
PRETEST	3
LANGKAH PRAKTIKUM.....	5
POST TEST	22



PRETEST

Nama: Mohammad Farid H, NIM: 2200018401	Asisten: Paraf Asisten:	Tanggal: 1. Juni 2024 Nilai:
--	----------------------------	---------------------------------

1. Ada banyak tools yang dapat digunakan untuk membuat class diagram.
- Visual Paradigm: Software UML tawarkan desain berbasis terminal, reverse engineering dan pembuatan node.
 - Visio: Berbentuk diagram dari software Microsoft. Include binanya dengan OFFICE
 - STARUML: Software open-source untuk pemodelan UML
 - Enterprise Architect: Tools pemodelan powerfull dari proses bisnis hingga class diagram
 - ArgotUML: Tools UML open source dengan interface sederhana standar UML
 - Drawio: Platform online gratis yang memungkinkan pembuatan class diagram dan diagram lainnya
 - LucidChart: Tools berbasis webuntuk membuat berbagai jenis diagram.
 - ActivitiUML: Tools UML lengkap, gratis untuk student.
2. Berikut contoh class diagram pada tema karya telepon



Class diagram ini menggambarkan sistem telepon dengan kelas `Telephone`. Sistem telepon dapat terhubung dengan `Line`, dan memiliki komponen `Ringer`, `caller Id` dan fungsi `Answering Machine`. Komponen `Answering Machine` menerima `Message`.

Lanjut ke halaman berikutnya

3.0 Class: Class diagram adalah sebuah diagram untuk memodelkan kelas-kelas dan relasinya seperti halnya di ERD (Entity Relation Diagram). Berbeda halnya dengan ERD yang hanya menggambarkan atribut, Class diagram menggambarkan kelas-kelas yang memiliki attribut, behavior dan state. Class diagraman seperti perwakilan yang terkait dengan tiga halaman

• Bagian atas: Berisi nama class (Contoh: kendaraan)

• Bagian tengah: Berisi daftar attribut yang dimiliki class (Contoh: Merk, warna, kelapa)

• Bagian bawah: Berisi daftar metode atau operasi yang dapat dilakukan oleh class (Contoh: Mulai, Berhenti, Mengalihdiri)

- Atribut: Atribut adalah karakteristik atau properti yang dimiliki oleh suatu class. Atribut menyimpan informasi tentang status objek yang diberikan dari class tersebut. Pada dasarnya dibuat dalam bentuk class diagram dengan format

[Visibility] name Atribut : type data C = nilai Awal]

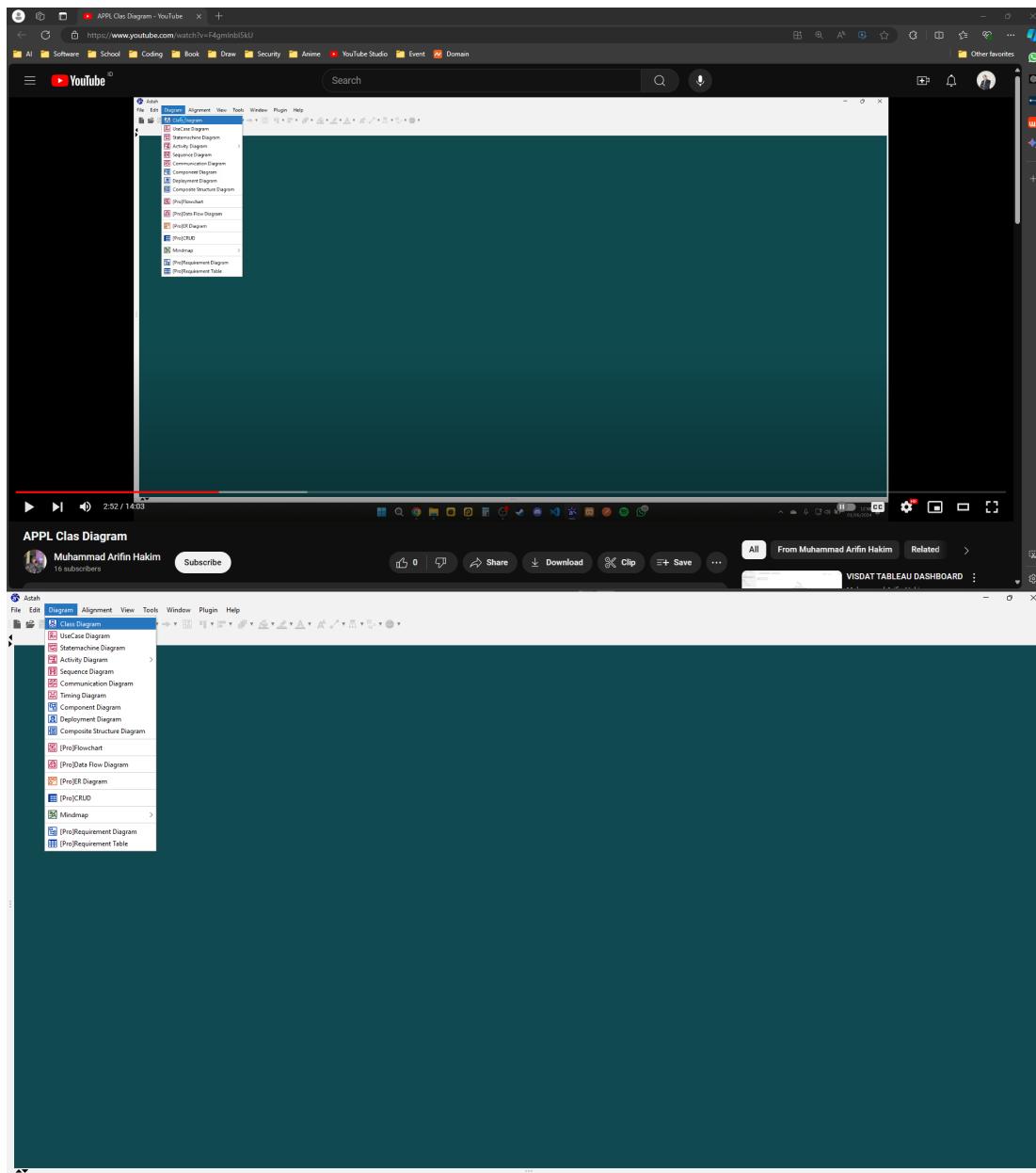
Visibility
 ↓
 + Public
 - Private
 ✘ Protected
 II Package

Mungkin anda akan bertanya ada berapa banyak tipe atribut yang tersedia? Terdapat dua tipe atribut yakni, misalkan atribut yang dapat diakses dari outside bahkan oleh objek lainnya, misalkan atribut yang tidak dapat diakses oleh objek lainnya kecuali itu sendiri, tetapi untuk pekerjaan sifatnya masih sama, contoh "Sifat".

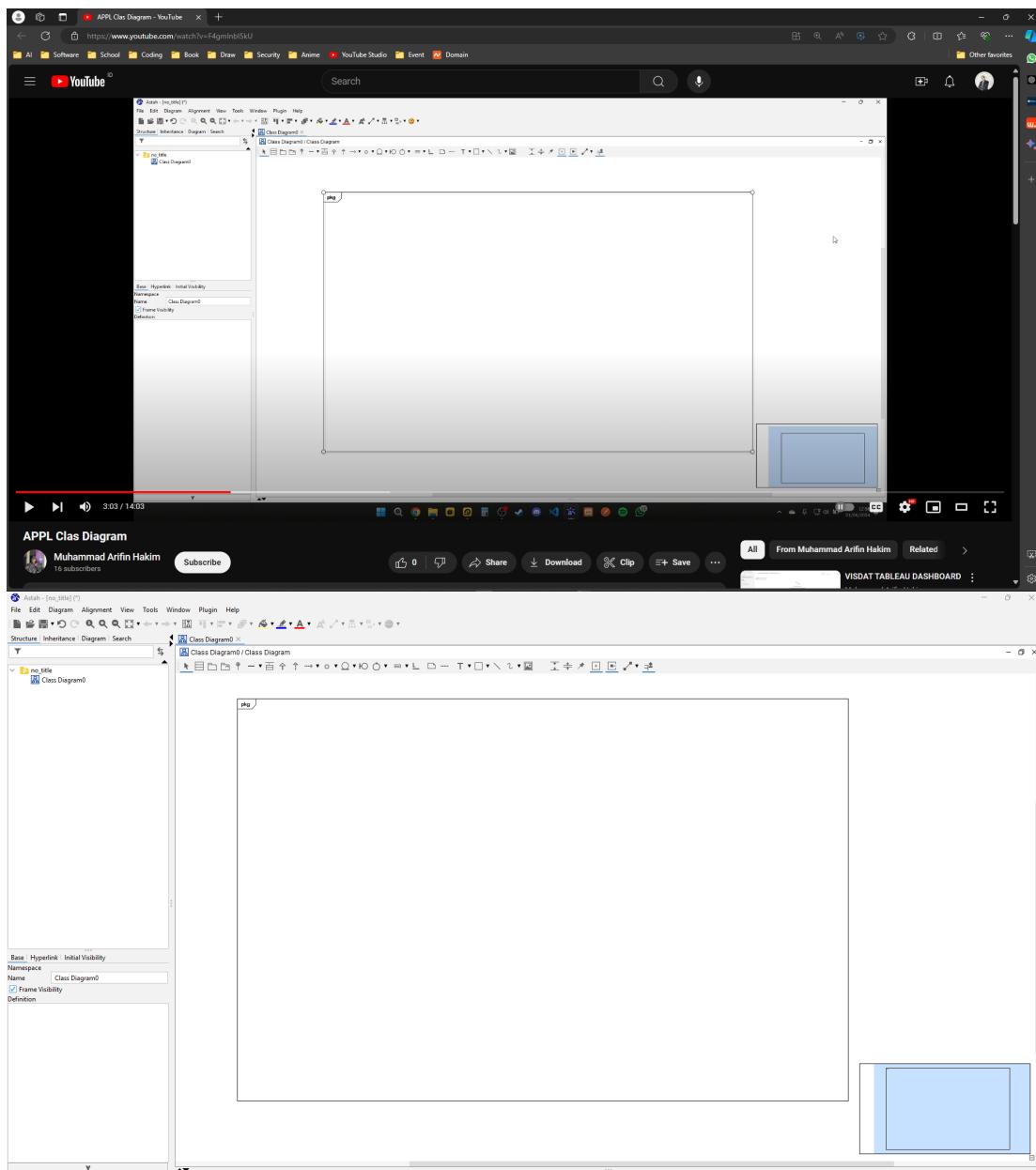
- Method (Operasi): Method adalah fungsi atau prosedur yang dapat dilakukan oleh suatu class, method mendefinisikan perilaku atau aksi yang bisa dilakukan oleh objek dari class tersebut. Pada dasarnya dibuat dalam bentuk class diagram misalnya diklasifikasikan konstruktur, query, atau update operasi. Biasanya pertama dari tambahan parameter atau informasi tambahan untuk melengkapi operasi tersebut.

LANGKAH PRAKTIKUM

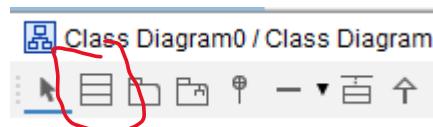
Membuka menu class Diagram. Jangan lupa buka AstahUML.



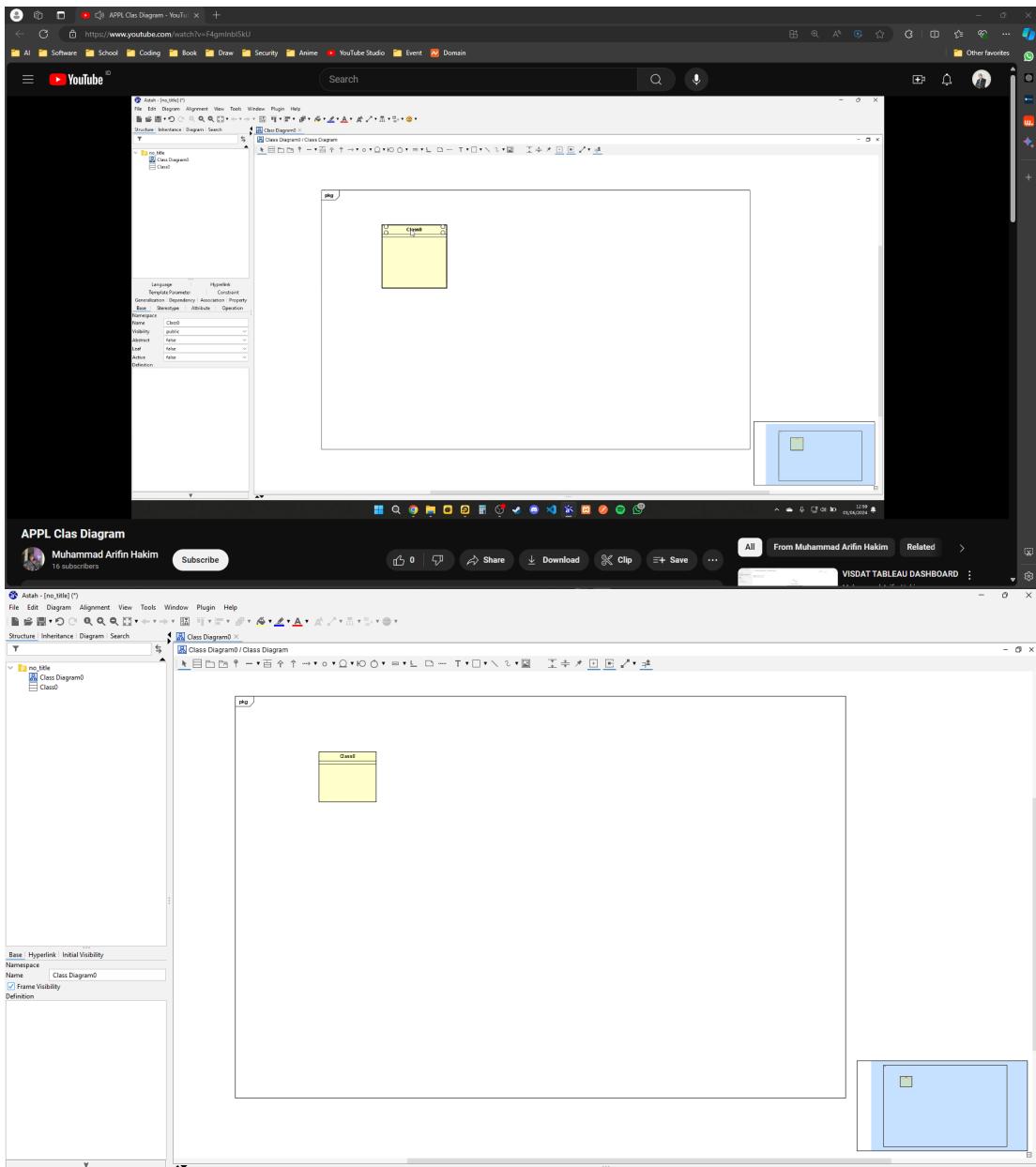
Mengubah ukuran board class diagram



Menambahkan class diagram



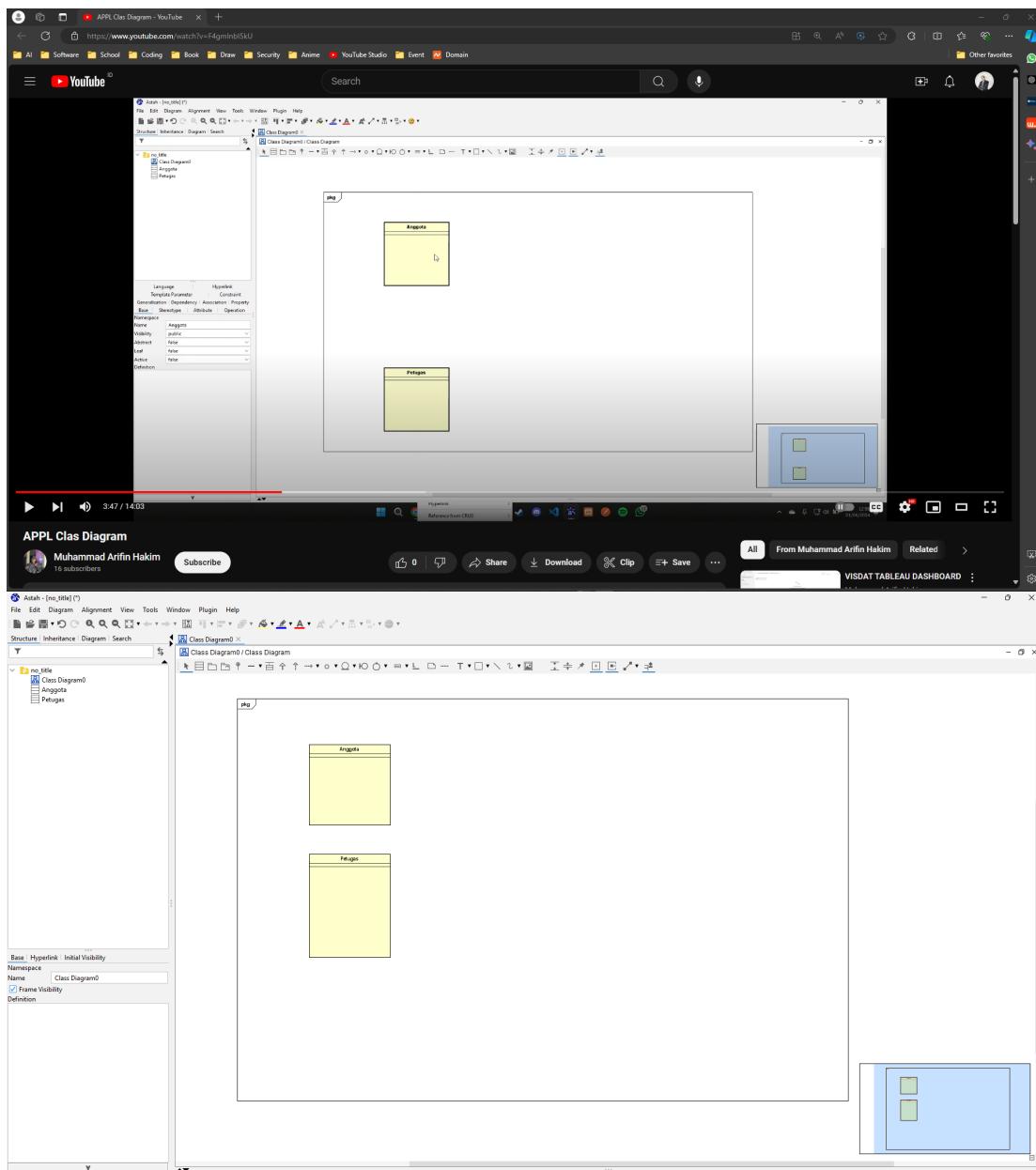
Klik pada kotak seperti tabel



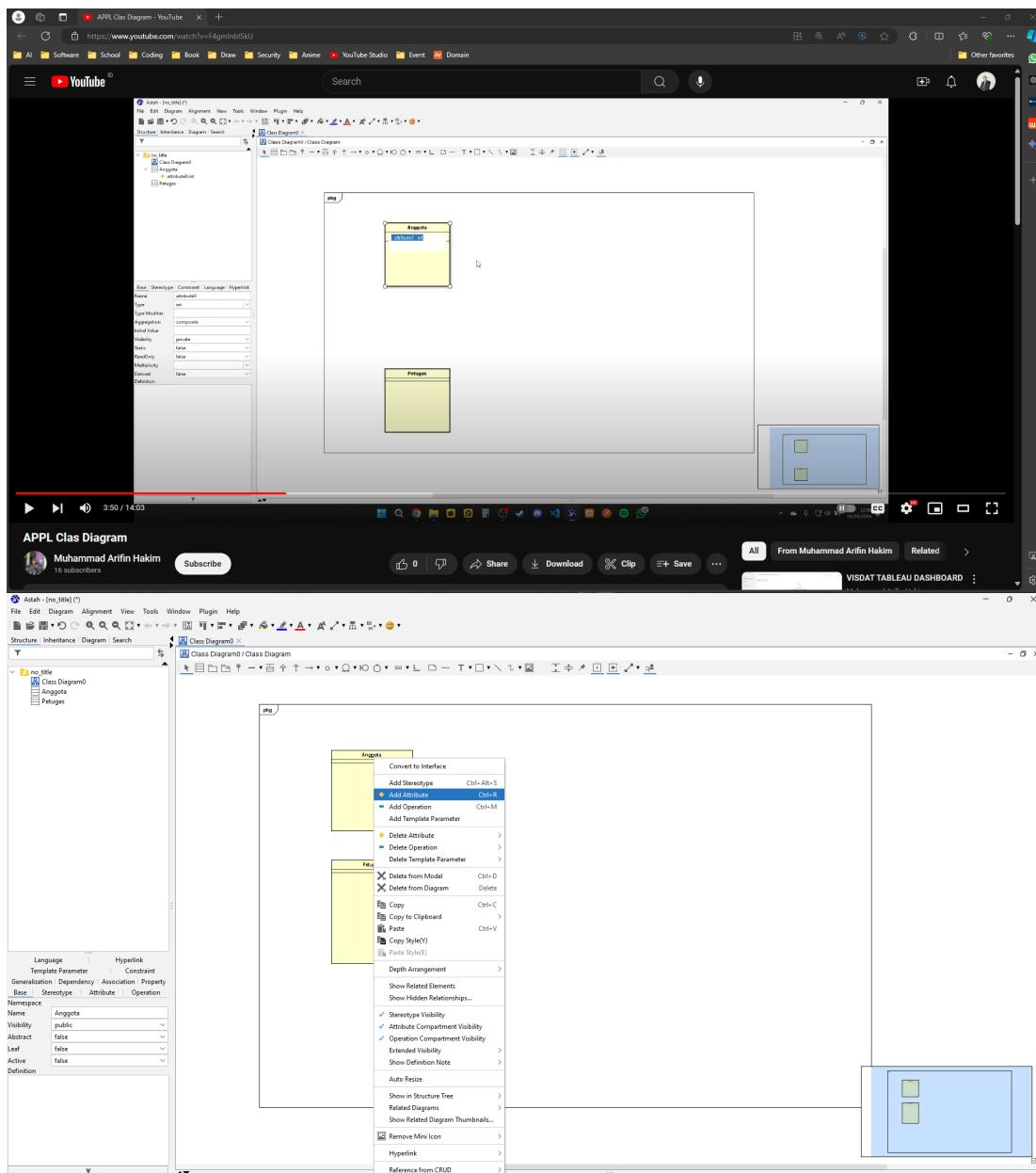
Hasil membuat objek Class di AstahUML

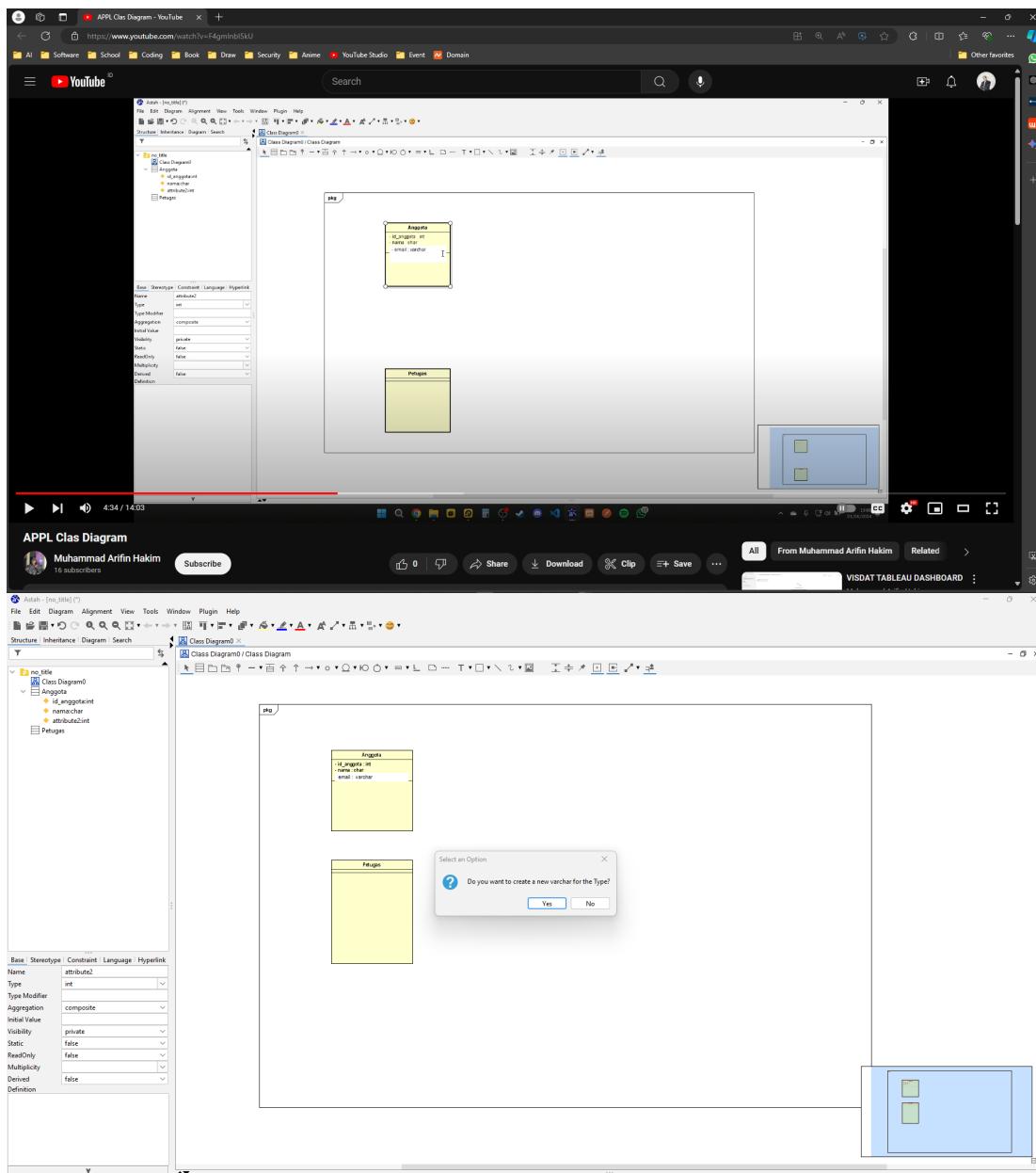
Setelah mengenal environment class diagram di AstahUML, kita akan mencoba menggambar class diagram, tentukan class diagram yang akan di buat yaitu di Langkah praktikum ini adalah “Perpustakaan Online”

Menentukan actor, yaitu Anggota dan petugas. Maka kita memerlukan 2 class yaitu Class Anggota dan Class Petugas



Setelah itu kita coba masukkan atributnya dengan klik kanan mouse, dan cari menu Add Attribute (Cara cepet tinggal select class dan CTRL+K)

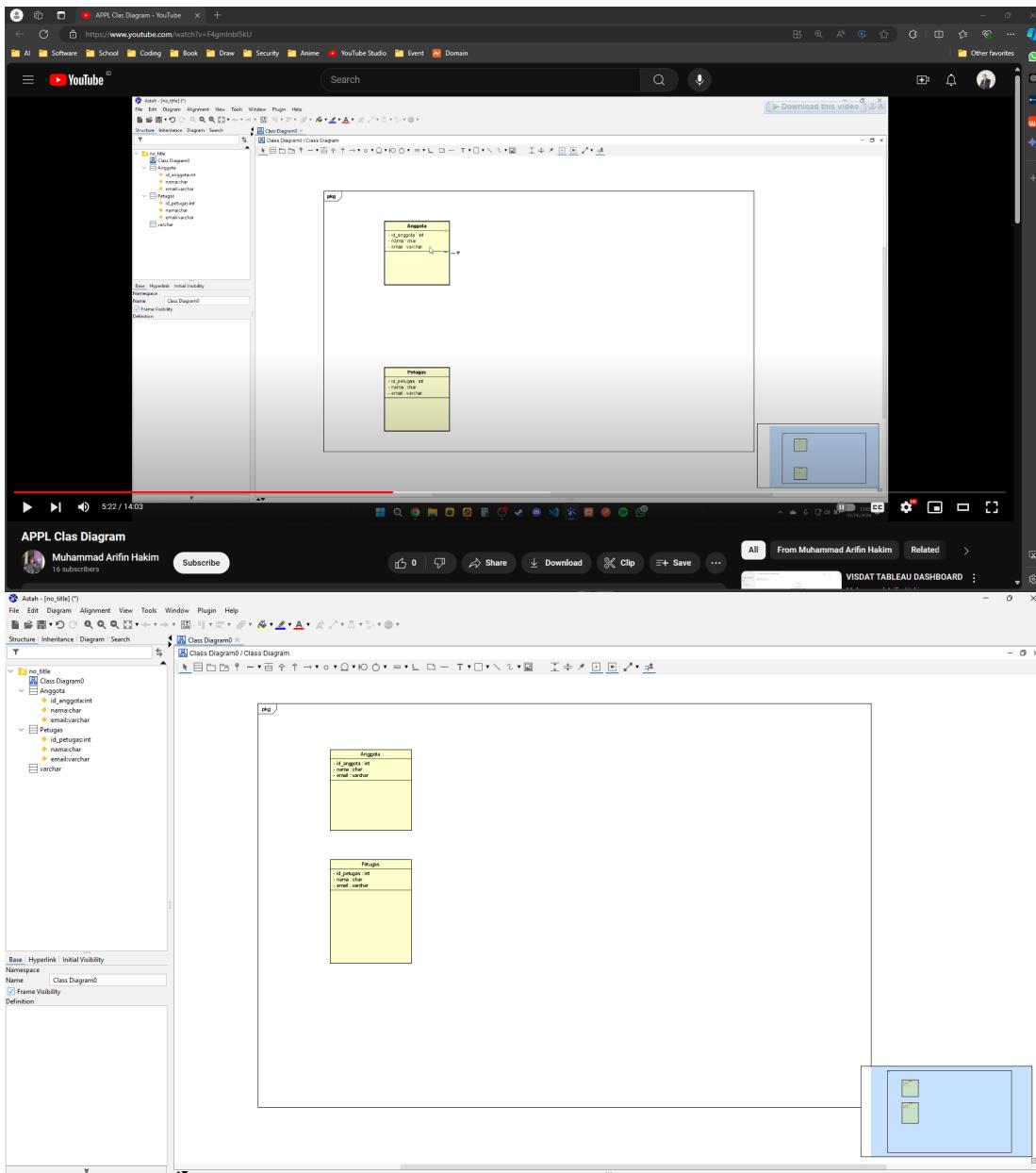




Setelah itu masukkan, apabila tidak ada type, tuliskan aja kemudian membuat type baru lalu klik “Yes”

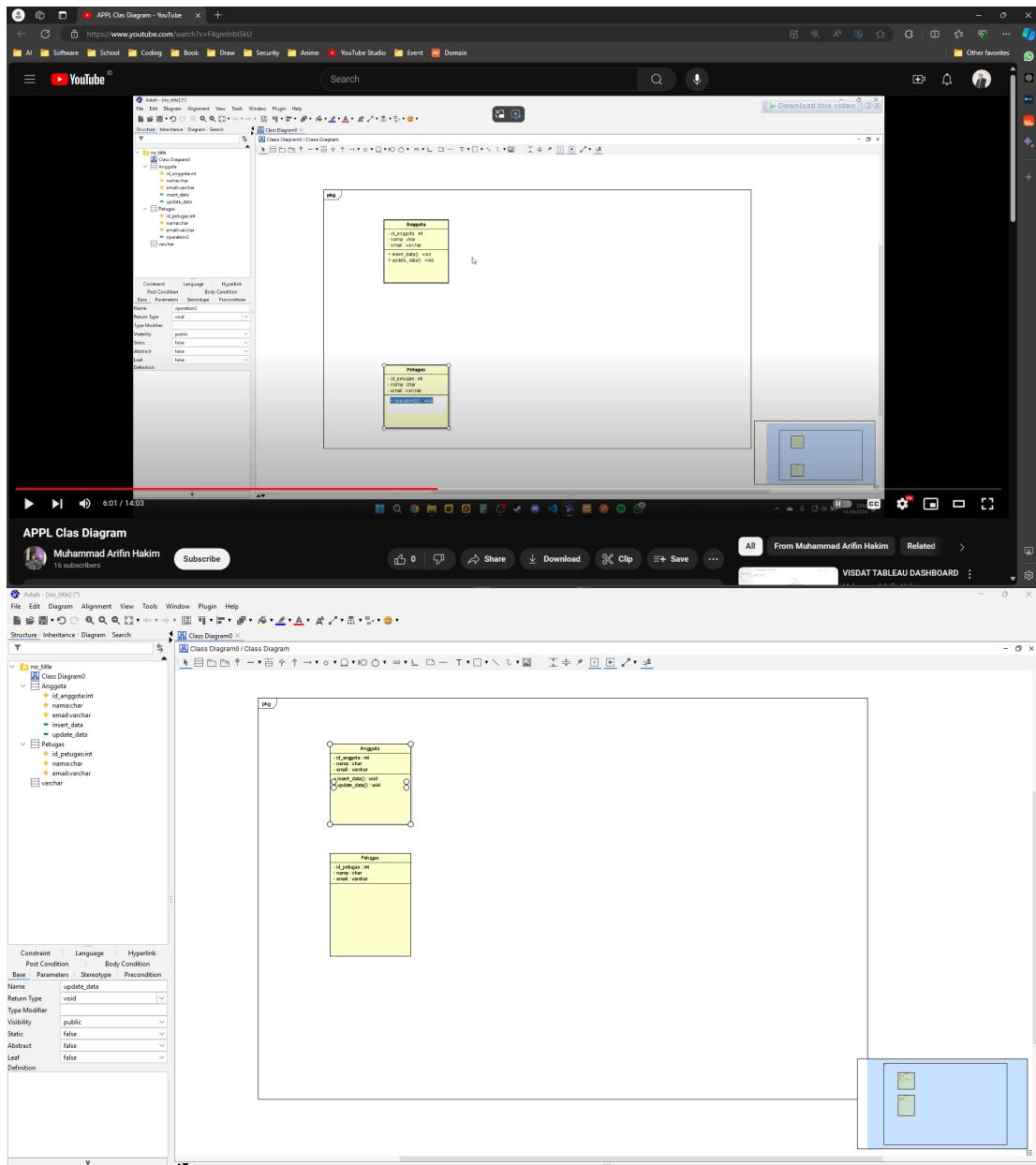
Menambahkan atribut petugas

Pastikan tiap kelas tidak ada id yang sama!

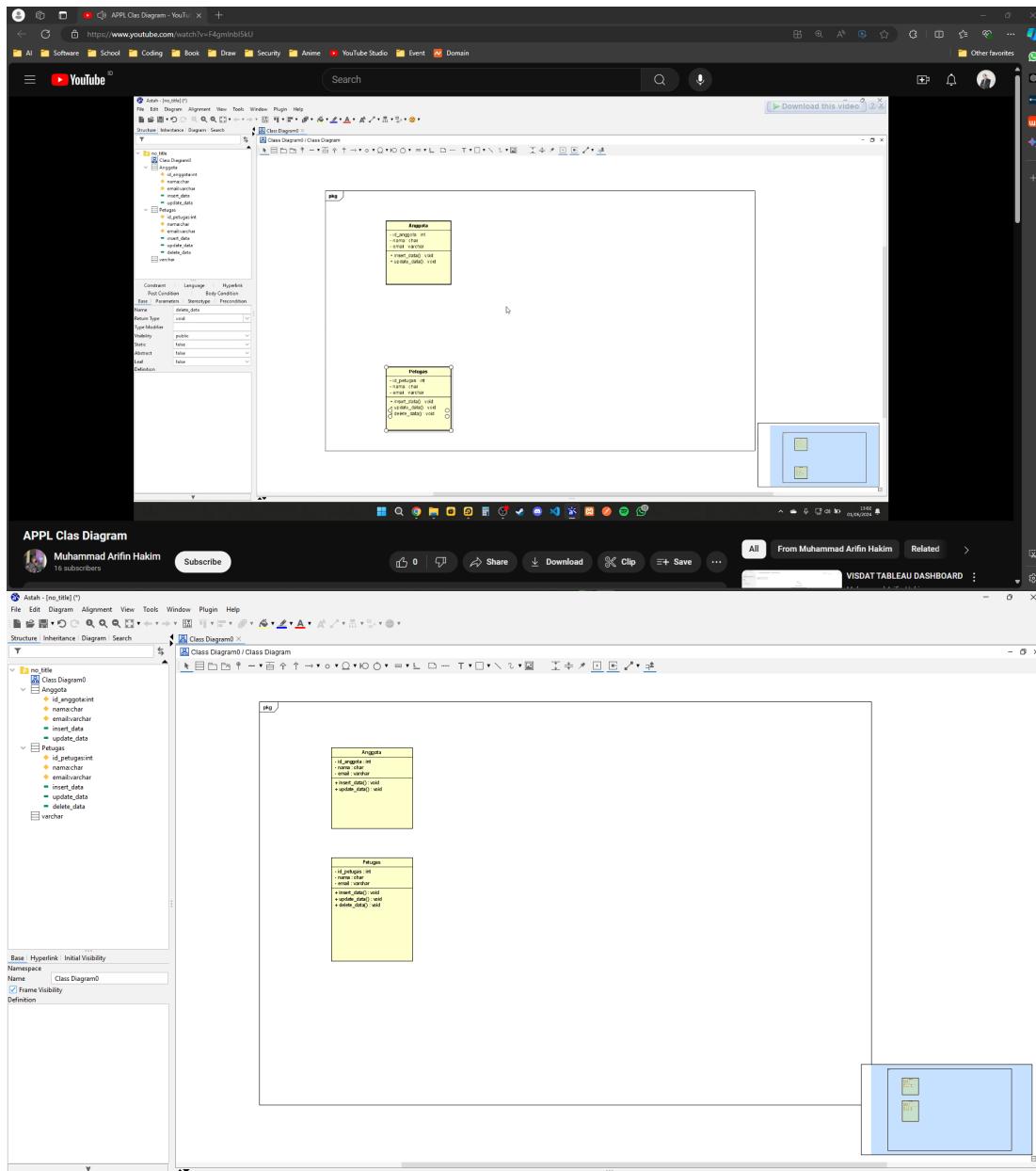


Langkah selanjutnya adalah memasukkan method (operasi apa saja di kelas masing-masing), caranya klik kanan, kemudian pilih menu Add Operation (Cara cepat adalah select class dan CTRL+M)

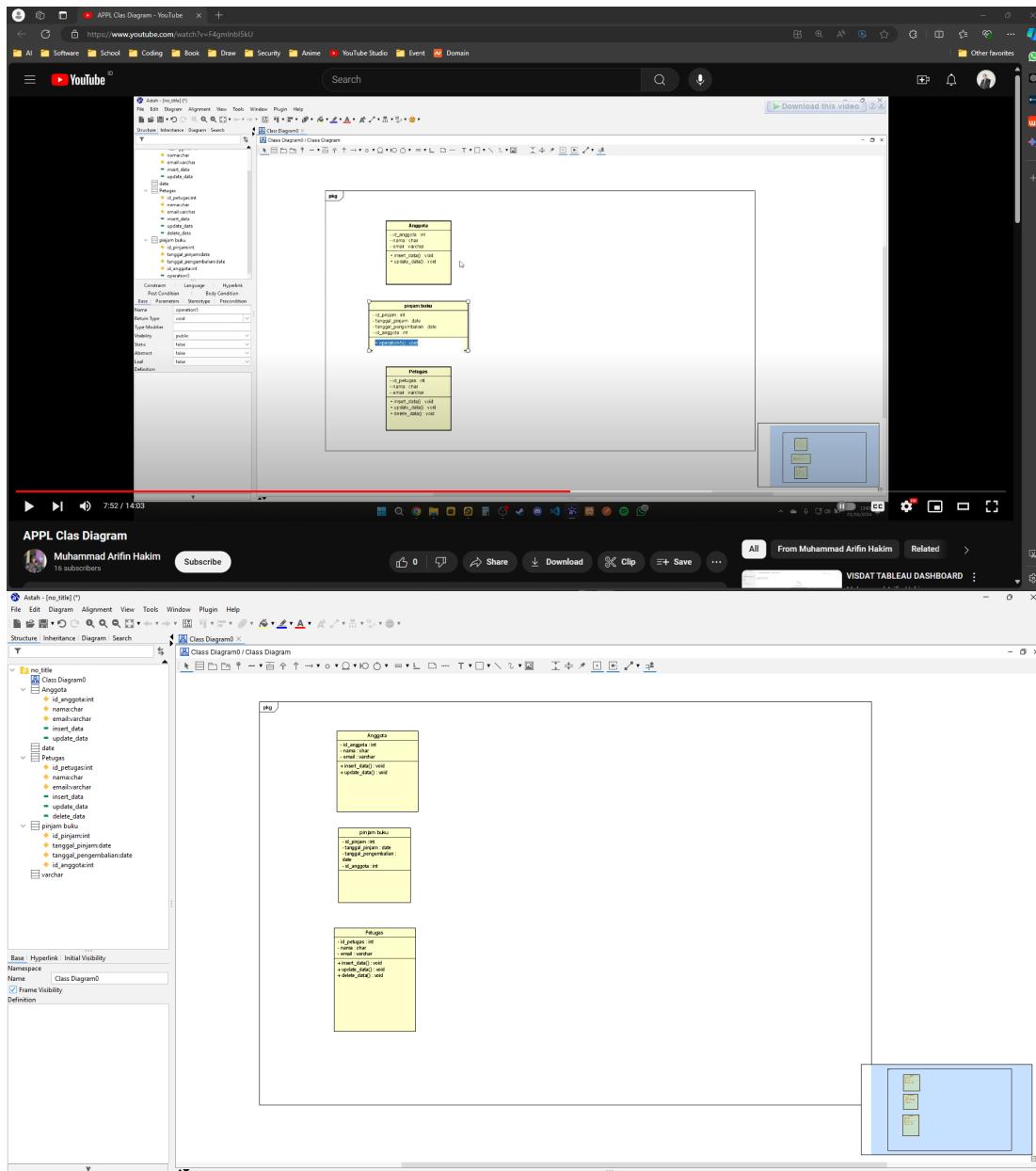
Kita coba masukka nmethod di kelas Anggota



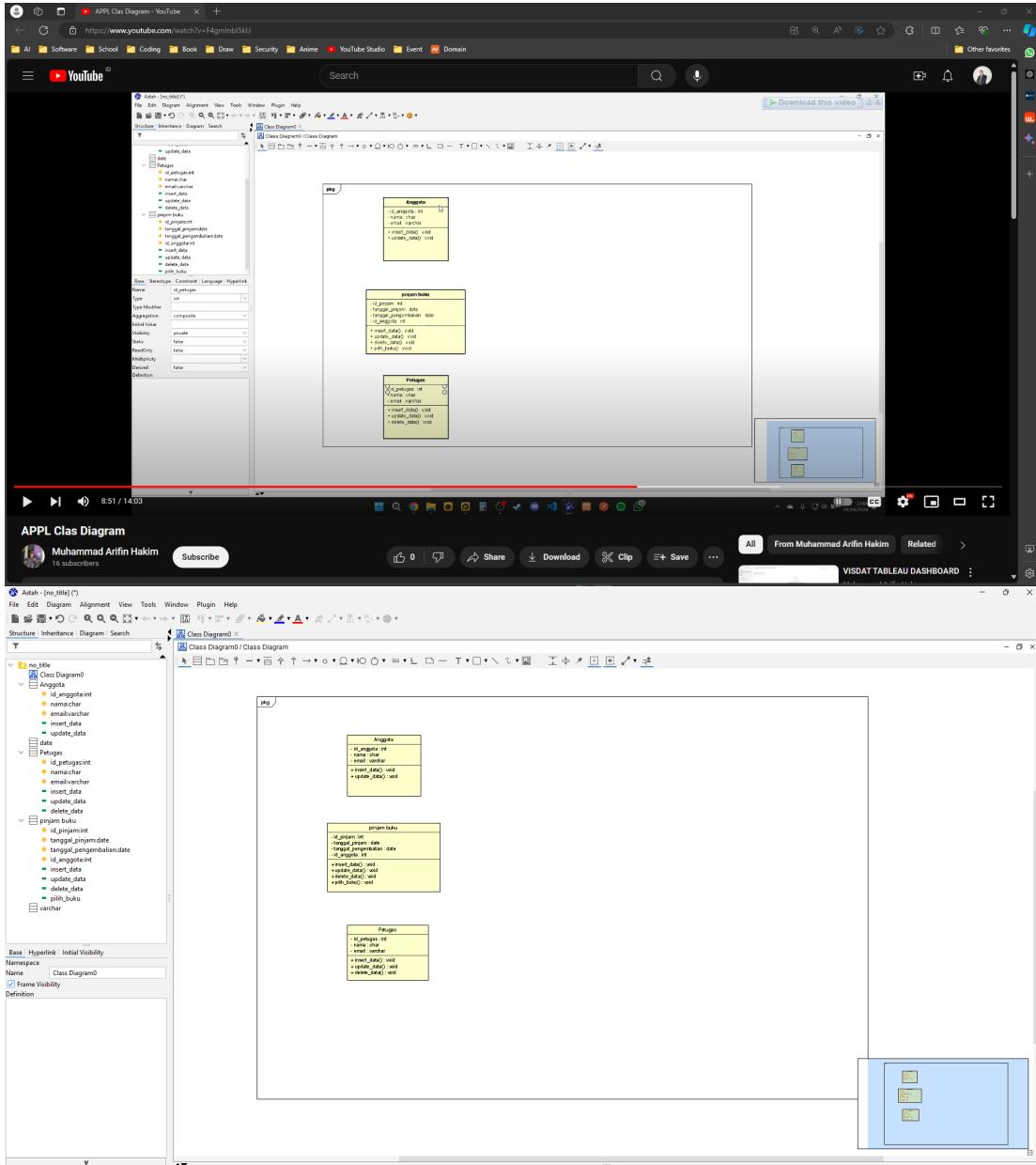
Kemudian masukkan ke petugas, kurang lebih sama dengan anggota! Perbedaannya kalo petugas itu bisa menghapus data.



Dari sini kita sudah membuat class anggota dan petugas. Karena kita membuat website perpustakaan, kita akan membuat class baru pinjam buku



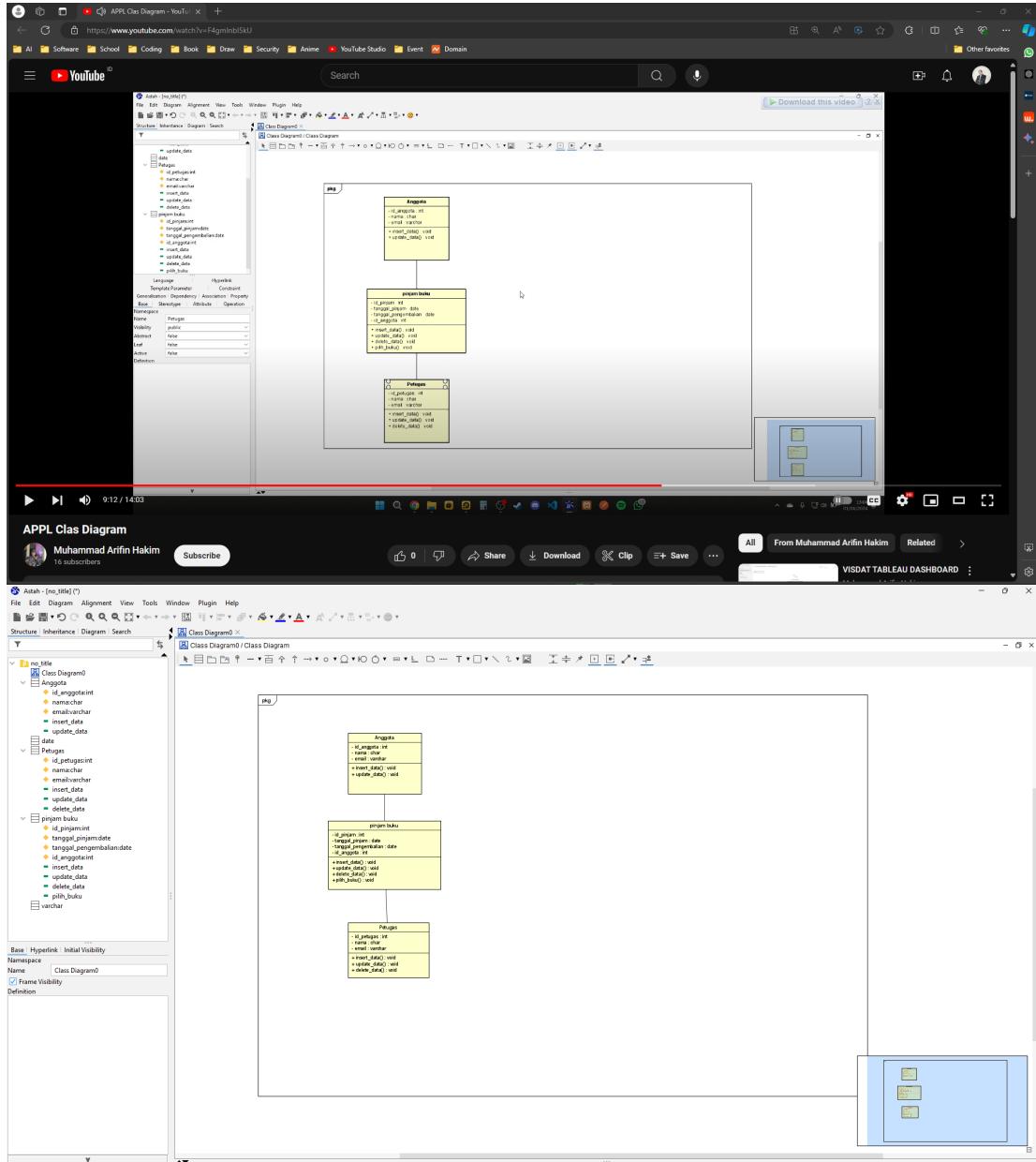
Kemudian kita tambahkan method di pinjam buku



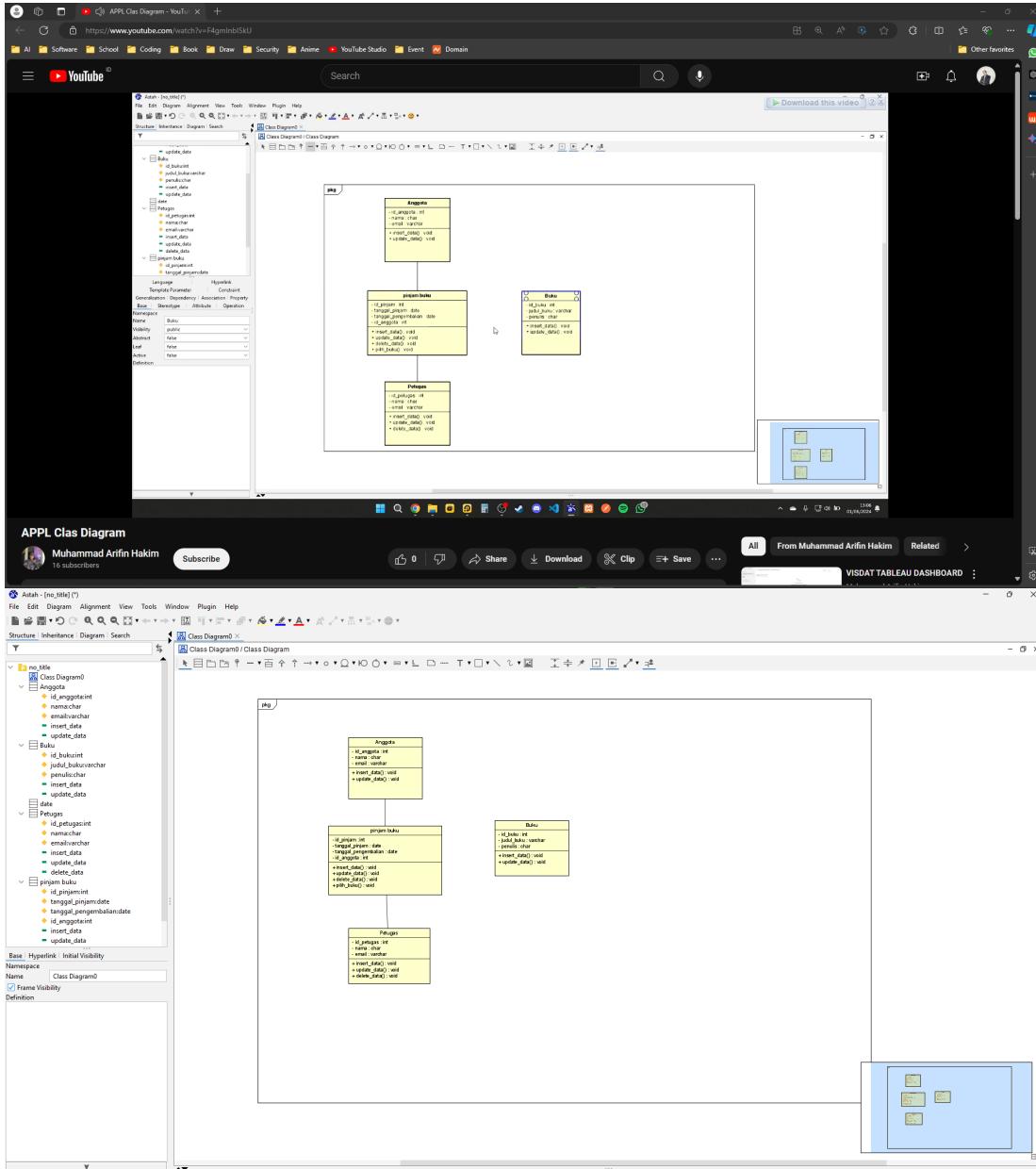
Setelah itu kita buat hubungan antara ketiga kelas, kita buat pertama anggota dan pinjam buku bersifat asosiasi. Begitu juga petugas dengan pinjam buku

agram0 / Class Diagram

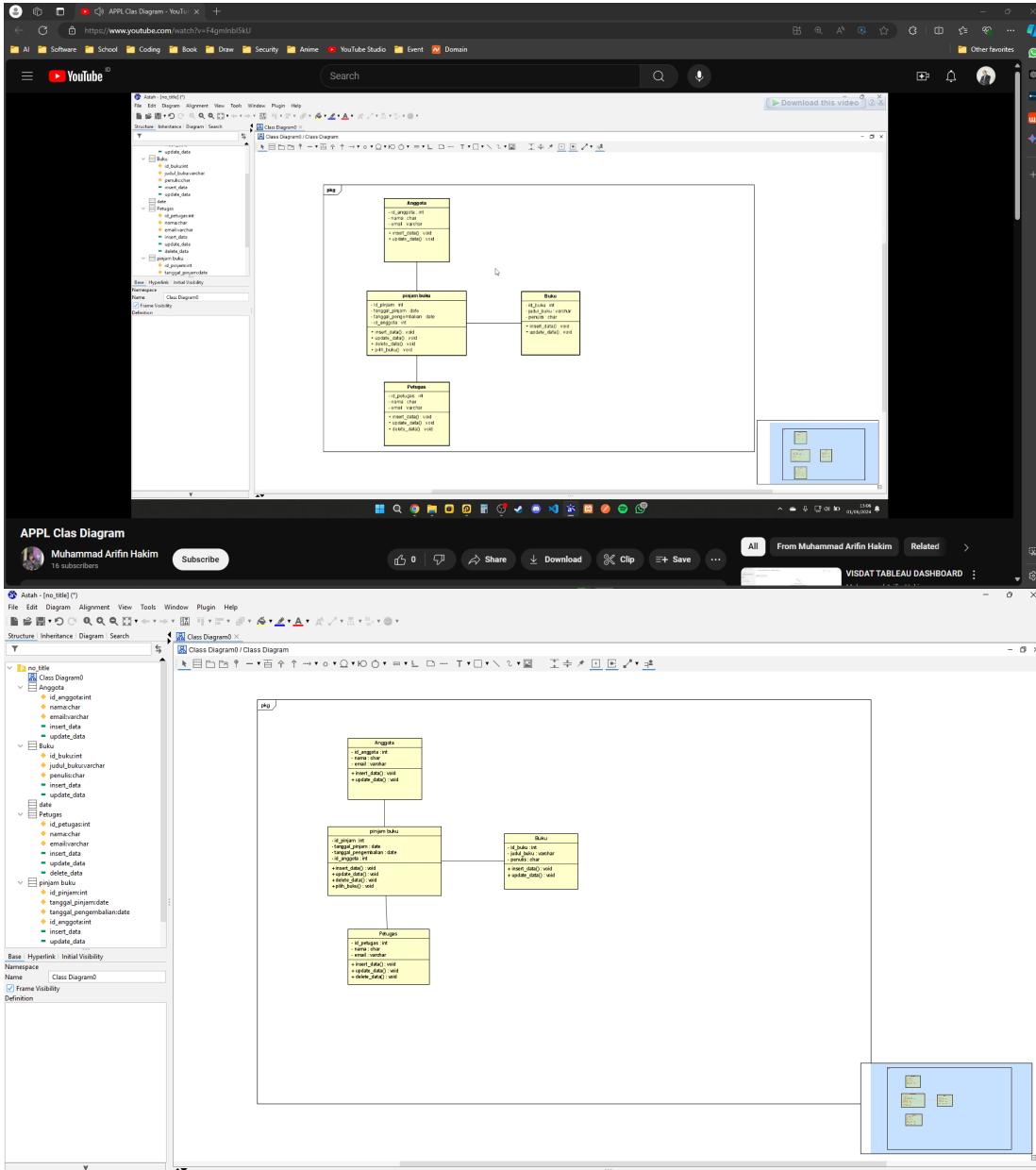




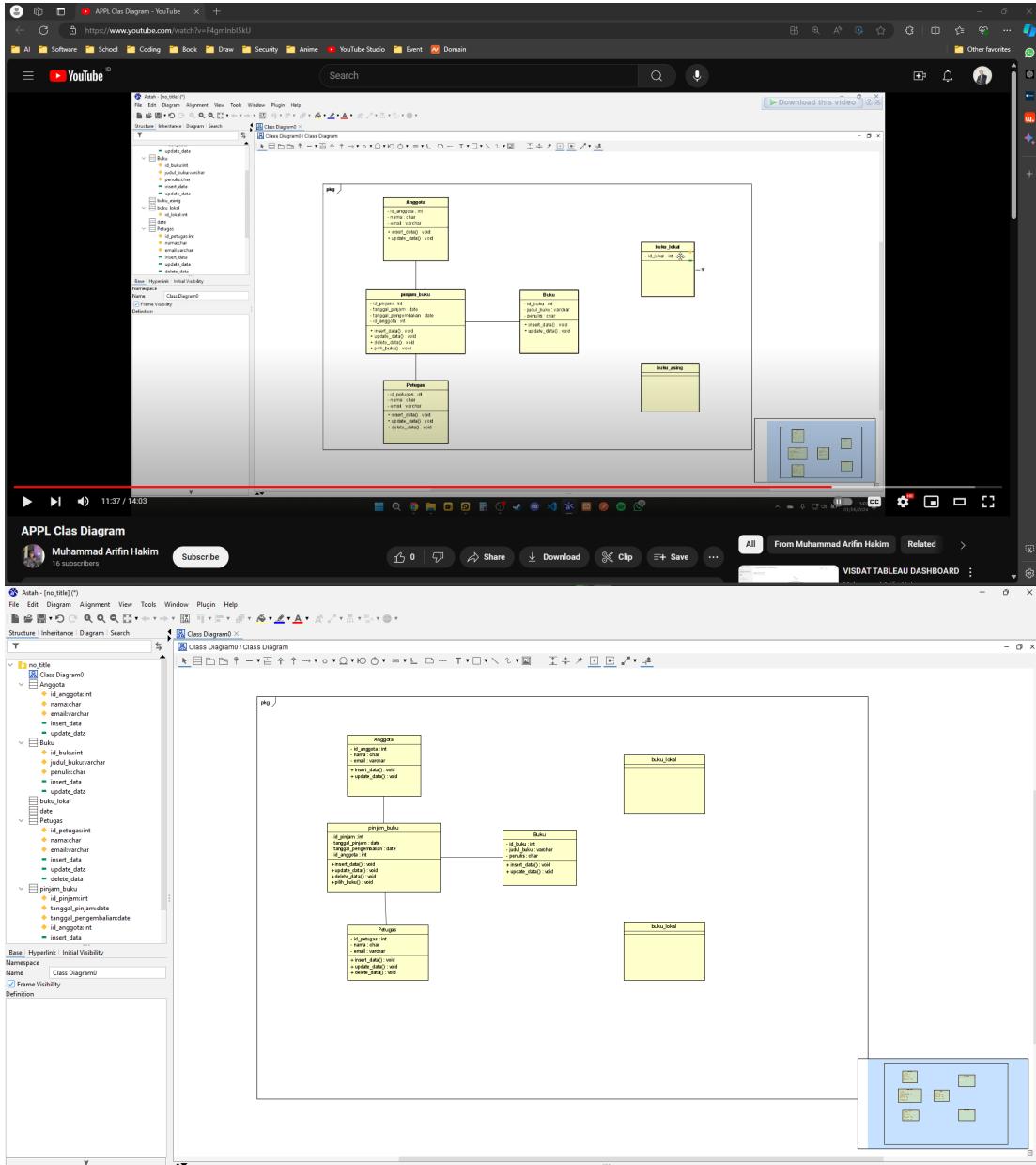
Karena pinjam buku operasi pilih buku, maka kita tambahkan class buku



Kemudian kita tambahkan hubungannya dari buku ke pinjam buku

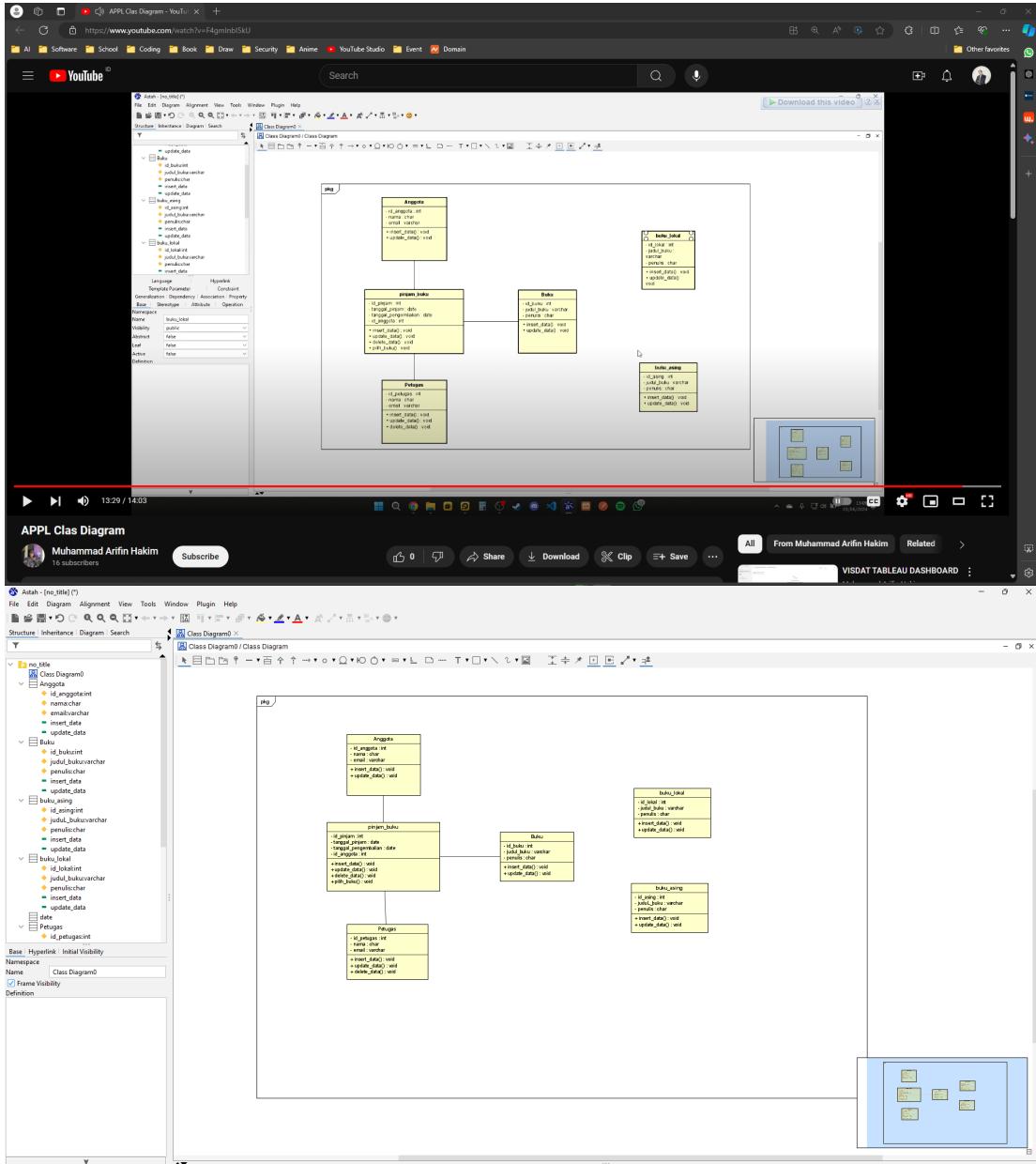


Kemudian kita akan mencoba 2 kelas baru untuk generalisasi



Ingin ya penulisan nama tidak boleh spasi, jika ada spasi gunakan “_”

Kemudian isikan atribut masing-masing

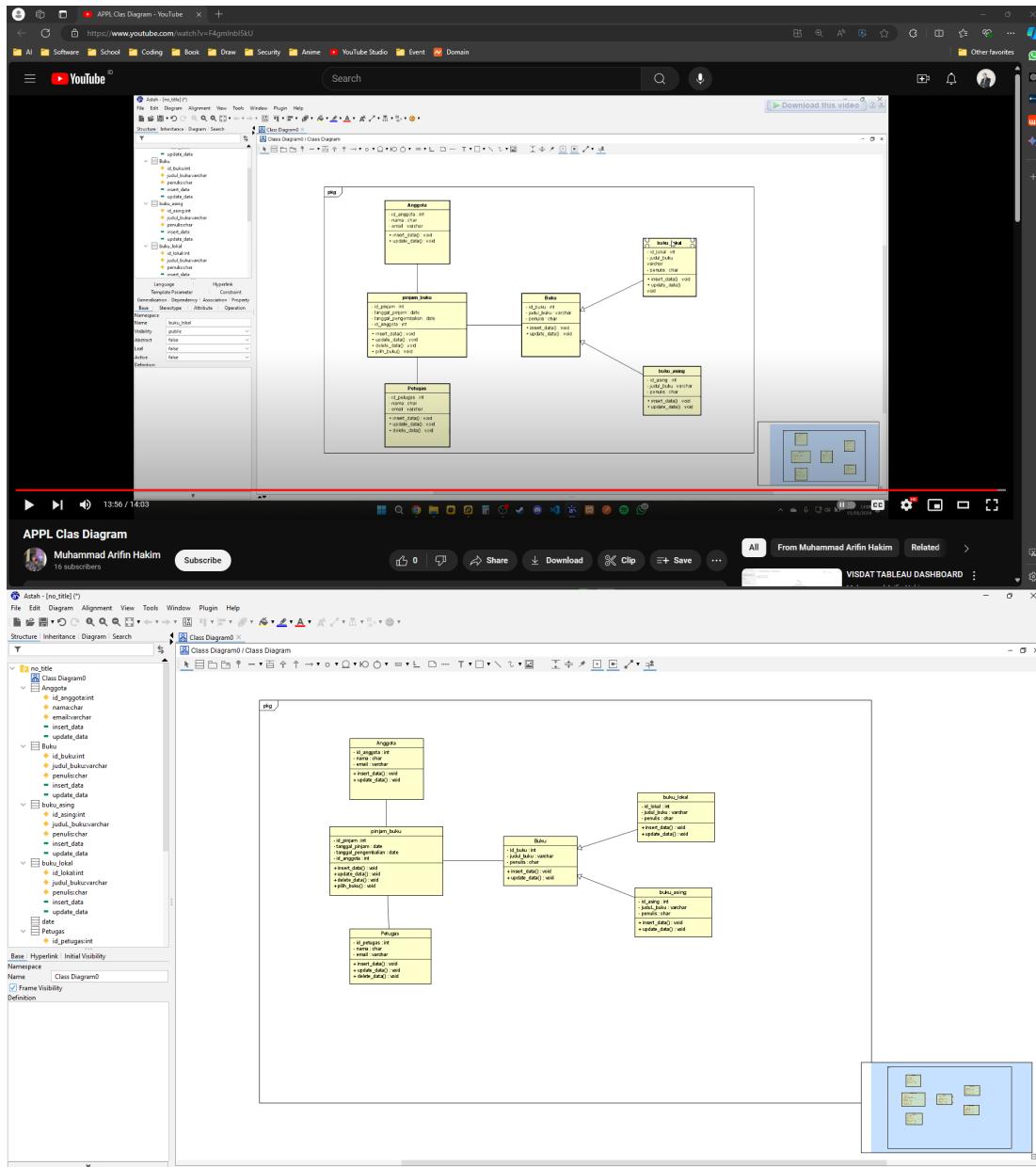


Tambahkan generalisasi yaitu garis yang ada panahnya

buku_asing itu subclass dari buku dan buku_lokal itu subclass dari buku



Generalisasi



Akhirnya class diagram “perpustakaan online sudah selesai dibuat.

POST TEST

Sebelum membuat class diagram, kita WAJIB menentukan alur kerja proses bisnis Konser berbasis web agar mengetahui objek yang kita gunakan dan proses apa yang dilakukan dalam aplikasi ini.

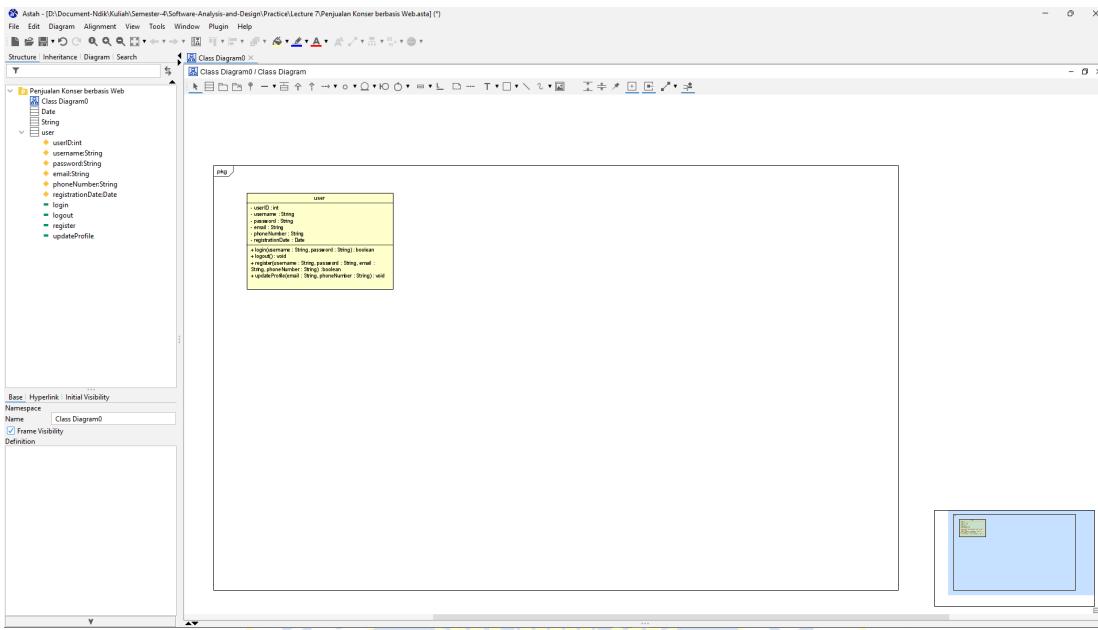
Aplikasi Penjualan Tiket Konser Berbasis Web dimulai dengan pengguna yang dapat melakukan registrasi atau login ke platform. Setelah masuk, pengguna dapat mencari konser yang diinginkan, melihat detail konser, dan memilih jenis tiket yang ingin dibeli. Setelah memilih tiket, pengguna dapat melanjutkan ke proses pembayaran, di mana mereka memberikan informasi pembayaran dan metode pembayaran yang diinginkan. Setelah pembayaran selesai, sistem akan menghasilkan tiket elektronik yang dapat diunduh atau dikirimkan melalui email kepada pengguna. Selain itu, pengguna juga dapat melihat riwayat pembelian tiket mereka dan memberikan ulasan tentang konser yang telah mereka hadiri. Di sisi admin, mereka dapat menambahkan, mengupdate, atau menghapus konser, serta melihat semua pesanan yang telah dilakukan oleh pengguna. Proses ini menciptakan lingkungan yang intuitif dan efisien untuk pengguna dalam menjelajahi dan membeli tiket konser secara online.

Setelah sudah mengetahui bagaimana alur kerja aktivitas, maka akan kita konversikan ke class diagram.

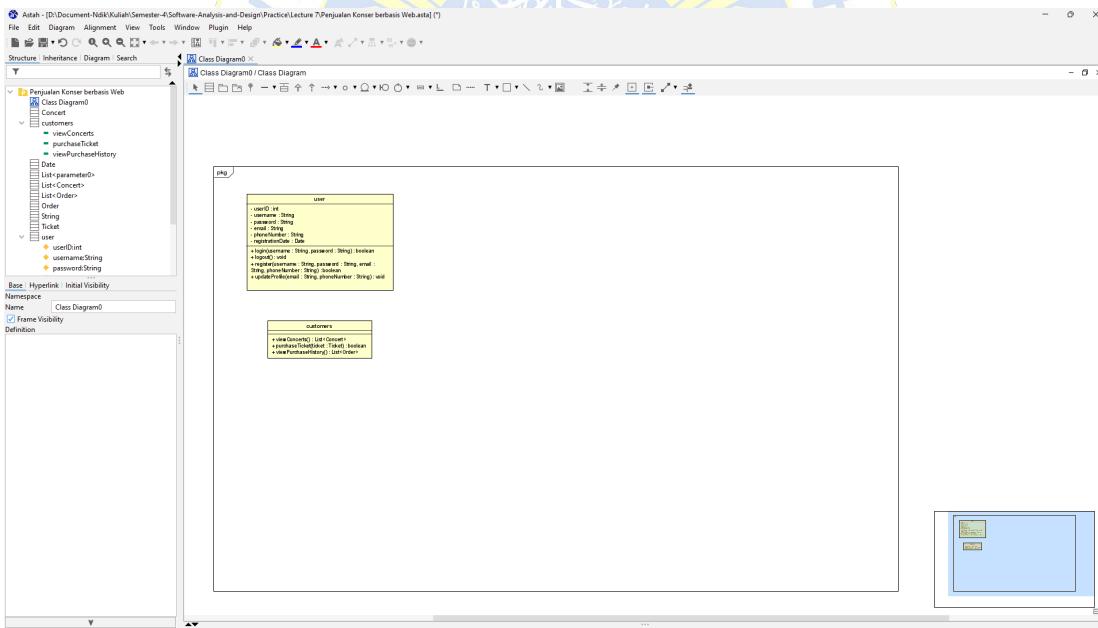
Sebelum itu, tentukan class yang akan digunakan

Daftar class

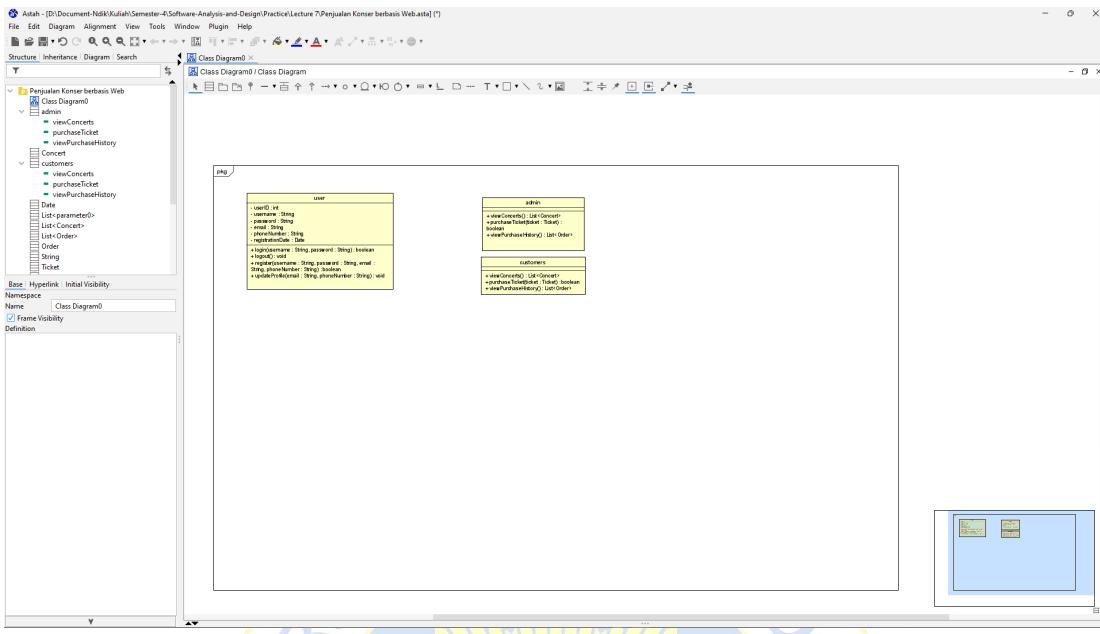
- User
- Admin
- Customer
- Order
- Payment
- Ticket
- RegularTicket
- VipTicket
- Review
- Venue
- Notification



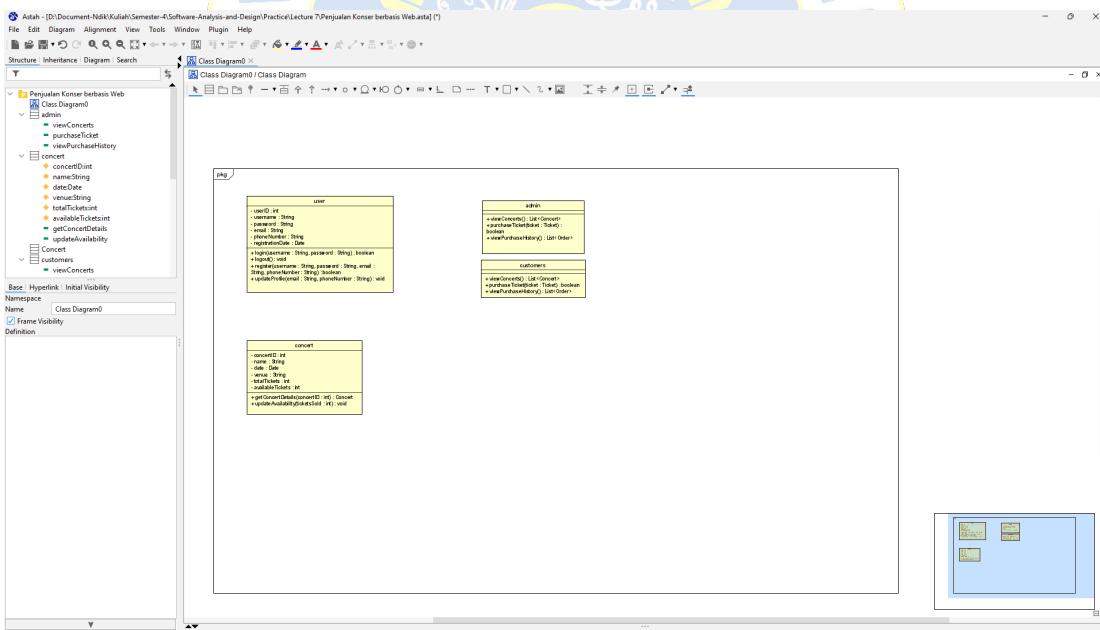
Membuat class user dan admin



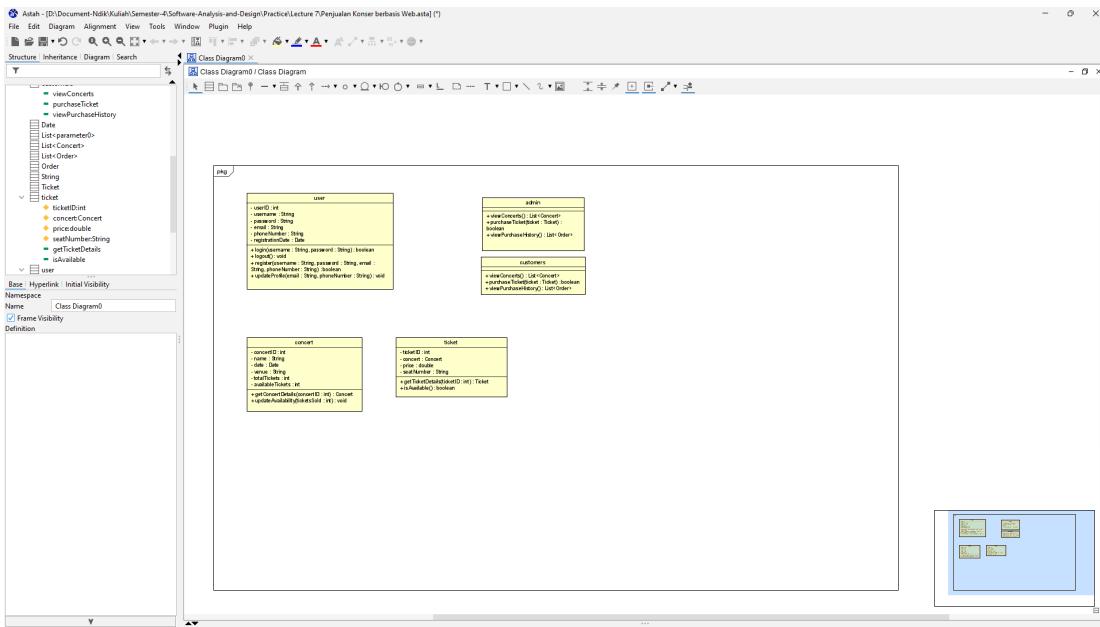
Membuat class customers, kita akan mencoba mengextend dari user. Begitu pula class admin, kita akan mengextend (melakukan generalisasi)



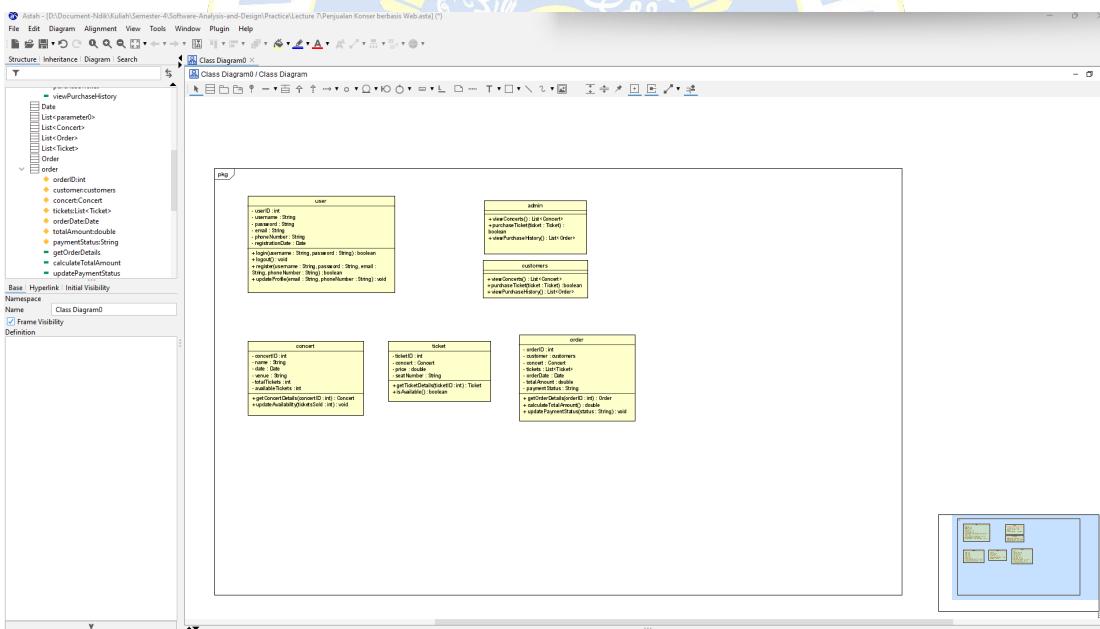
Membuat class concert. Disini akan menyimpan Concert seperti nama konser, jumlah tiket dan tanggalnya.



Pastinya konser kan butuh tiket, kita buat class tiket

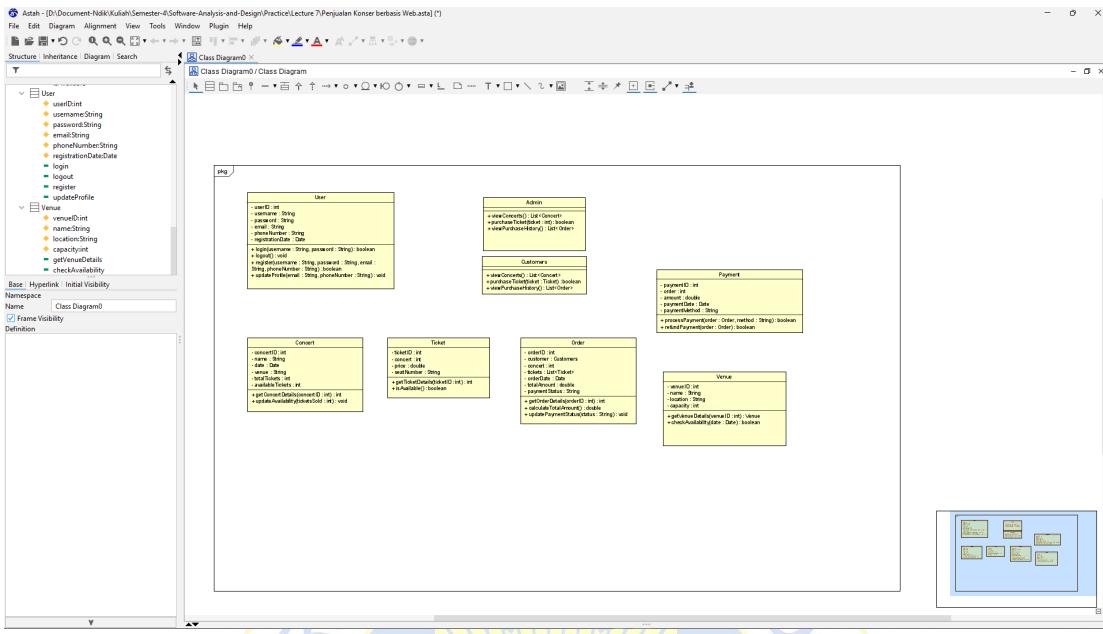


Saat melakukan pembelian order, kita membutuhkan relasi order. Nah disini menyimpan data oleh karena itu kita akan membuat class order

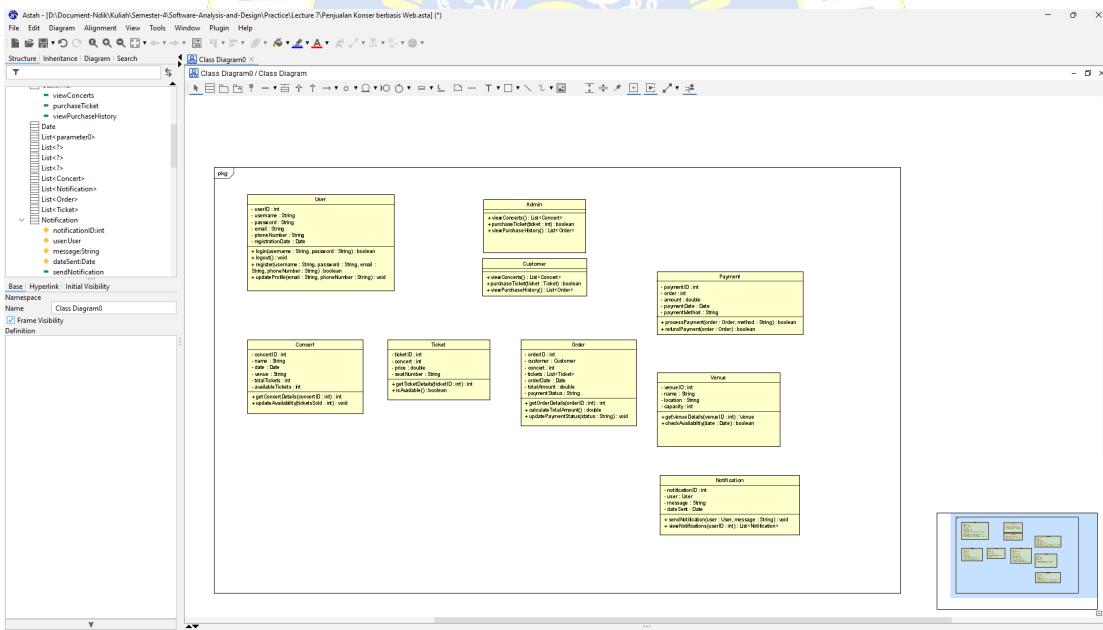


Setelah melakukan payment, maka diperlukan membayar dengan metode apa gitu. Dan id pembayarannya.

Setelah itu kita akan cek lokasi konser (Venue)

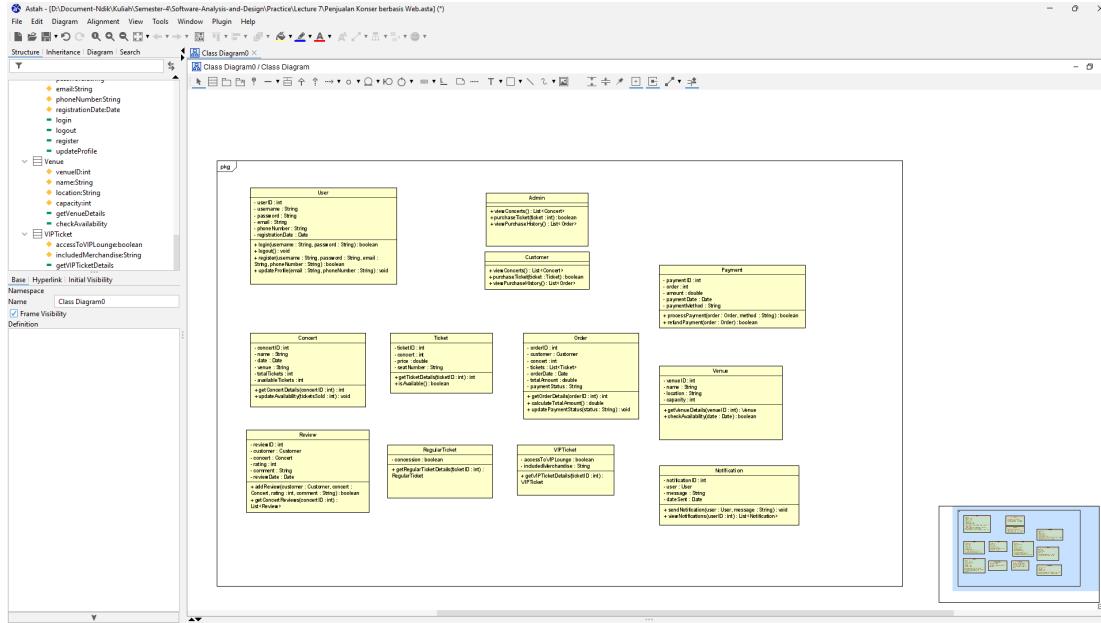


Pastinya kan di website bisa memberikan notifikasi bahwa konser kapan mulainya, sebagai pengingat Customers.



Setelah membuat notifikasi, Customer menonton sebuah konser. Setelah nonton, pasti ngereview apakah konsernya bagus atau tidak.

Oh iya kita tambahkan tiket itu kan gak hanya 1 jenis, ada yang reguler, VIP (khusus pejabat atau horang kaya) dsb.

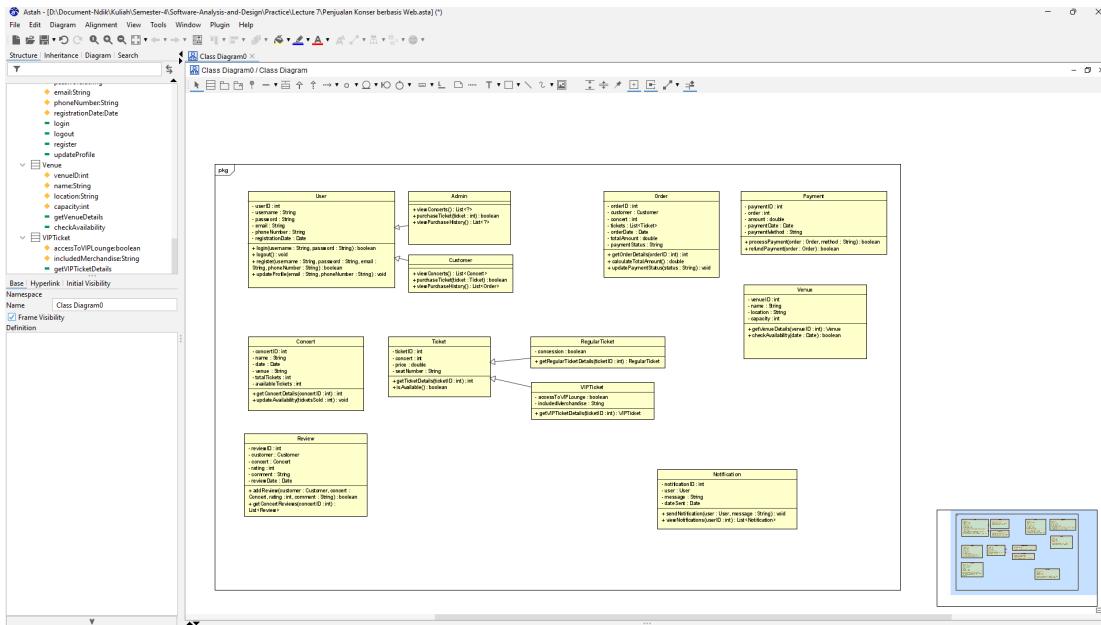


Setelah itu kita coba tentukan relasi masing masing

Generalisasi

Customer dan Admin meng-extend User.

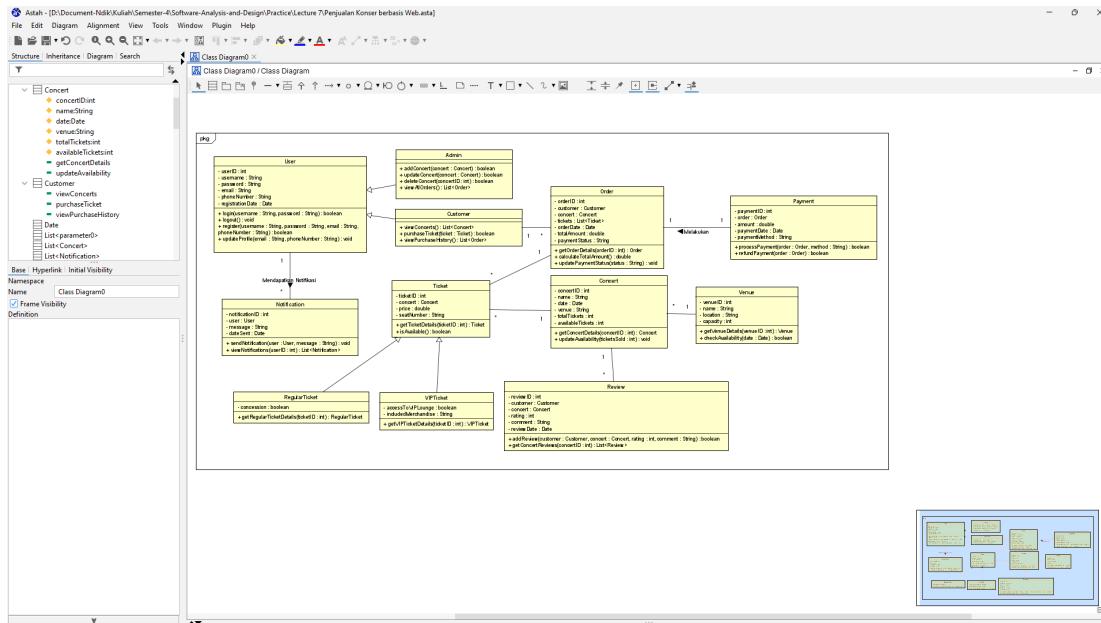
RegularTicket dan VIPTicket meng-extend Ticket.



Menentukan Hubungan semuanya ya

- User - Notification (One-to-Many): Seorang pengguna bisa menerima banyak notifikasi.
- Customer - Order (One-to-Many): Seorang pelanggan bisa membuat banyak pesanan.
- Order - Payment (One-to-One): Satu pesanan memiliki satu pembayaran.
- Concert - Ticket (One-to-Many): Satu konser memiliki banyak tiket.
- Concert - Review (One-to-Many): Satu konser bisa memiliki banyak ulasan.
- Venue - Concert (One-to-Many): Satu tempat bisa menyelenggarakan banyak konser.
- User – Concert (One-to-One): User hanya dapat memberikan satu komentar.

Berikut adalah class diagram lengkapnya



Penjelasan Class Diagram

Class diagram ini mencerminkan hubungan antar objek dalam aplikasi penjualan tiket konser berbasis web. Pertama, pengguna dapat melakukan akses sebagai pelanggan atau admin. Sebagai pelanggan, mereka dapat melihat konser, membeli tiket, dan melihat riwayat pembelian mereka. Admin memiliki kekuatan tambahan untuk mengelola konser, seperti menambah, mengubah, atau menghapus konser, serta melihat semua pesanan.

Setiap konser memiliki detail seperti nama, tanggal, tempat, dan jumlah tiket yang tersedia. Pengguna dapat memesan tiket untuk konser tertentu, yang kemudian menjadi bagian dari pesanan. Pesanan mencakup detail seperti pelanggan, konser yang dipesan, daftar tiket yang dibeli, tanggal pesanan, total biaya, dan status pembayaran.

Pembayaran diasosiasikan dengan pesanan, dengan kemampuan untuk memproses dan mengembalikan pembayaran. Ada juga kemungkinan untuk memberikan ulasan terhadap konser tertentu. Setiap ulasan terdiri dari peringkat dan komentar dari pelanggan yang terkait dengan konser. Melalui hubungan-hubungan ini, sistem memungkinkan pengguna untuk melakukan tindakan seperti pembelian tiket, mengelola konser, dan melakukan pembayaran dengan cukup terstruktur dan efisien.

Terdapat juga kelas abstrak Ticket, yang memiliki dua turunan: RegularTicket dan VPTicket. RegularTicket mewakili tiket biasa dengan opsi potongan harga, sementara VPTicket mewakili tiket VIP dengan akses ke area VIP dan barang dagangan tambahan.

Untuk mengakses hasil pembuatan class diagram, dapat melihat link github berikut:

<https://github.com/IRedDragonICY/Software-Analysis-and-Design/tree/main/Practice/Lecture%207>

