

Tugas Class Diagram

Diajukan Untuk Memenuhi Tugas Mata Kuliah RPL: Desain dan Implementasi CII3J4

Dosen Pengampu: Ati Suci Dian Martha, S.Kom., M.T.



Disusun Oleh:

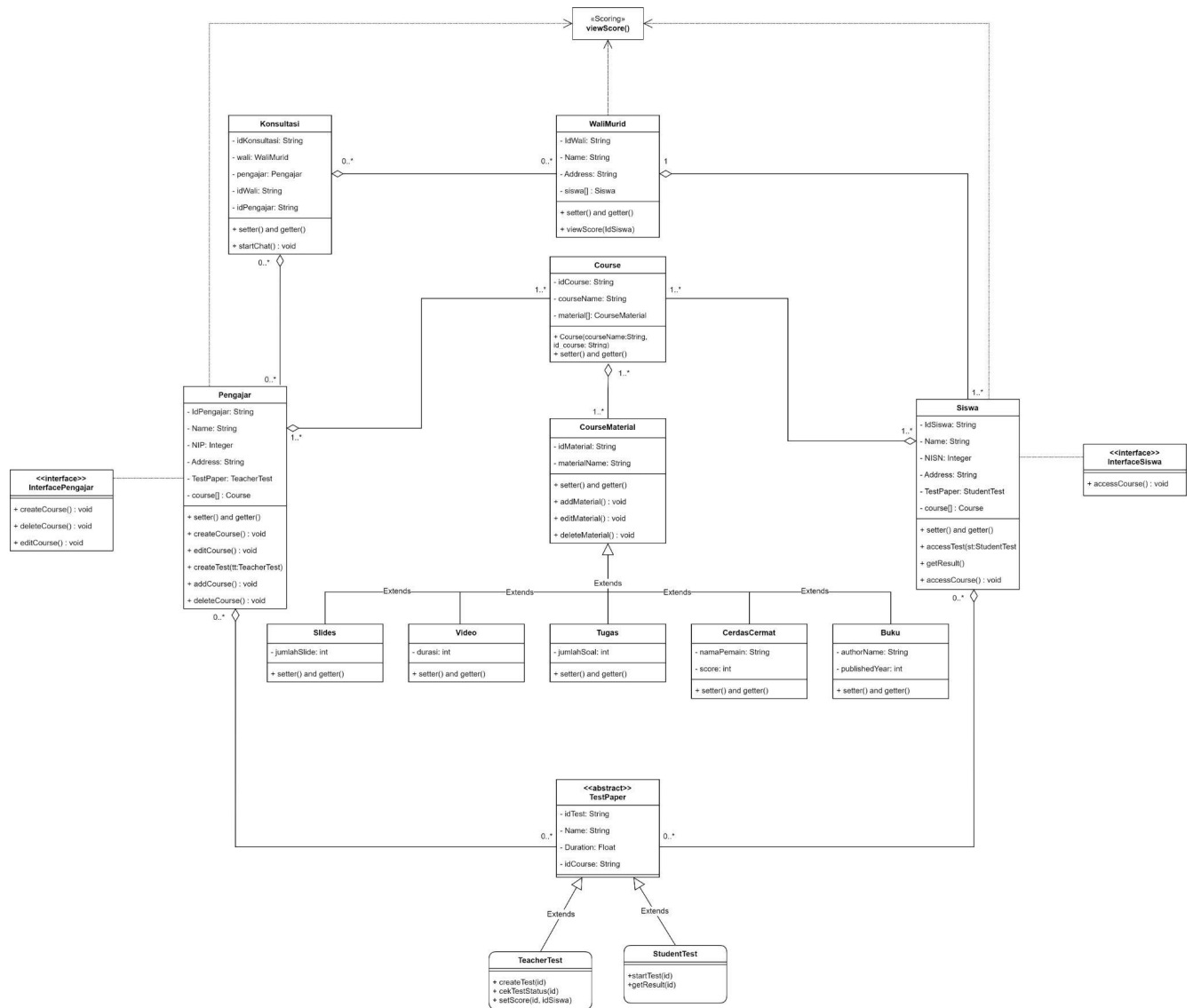
| | |
|-------------------------------|--------------|
| Ni Made Dwipadini Puspitarini | (1301194141) |
| Ahmad Julius Tarigan | (1301190345) |
| Afrizal Syahruluddin Yusuf | (1301194288) |
| Andi Muh. Syahrul Ulum | (1301190459) |
| Rizki Tri Setiawan | (1301194012) |

PROGRAM STUDI INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY

2021/2022



Class diagram diatas merupakan salah satu desain arsitektur yang kami rancang untuk membantu dalam pengerjaan tugas besar. Adapun class-class yang kami definisikan terdiri dari pengajar, siswa, wali murid, course, course material, konsultasi, abstract class test paper, dan beberapa interface. Selain itu, kami juga mencoba untuk mengimplementasikan prinsip SOLID pada perancangan class diagram tersebut dengan penjelasan sebagai berikut:

1. S (Single Responsibility Principle) : berdasarkan pengertian SRP, kami membangun class-class yang hanya memiliki satu responsibility saja. Maksud dari satu responsibility, bukan berarti setiap class hanya memiliki satu method saja, melainkan antara satu method dengan method lainnya memiliki hubungan korelasi yang kuat. Contohnya, pada method class Pengajar memiliki responsibility untuk mengelola course dan tidak digabung dengan konsultasi wali murid. Oleh karena itu, kami

membuat class baru yaitu Konsultasi yang memiliki responsibility untuk menghubungkan antara class Pengajar dengan class WaliMurid.

2. O (Open Closed Principle) : kami membangun abstract class TestPaper yang memiliki turunan atau extends berupa class TeacherTest dan StudentTest. Artinya, class TestPaper ini telah memenuhi sifat OCP yaitu open for extension.
3. L (Liskov Substitution Principle) : dimana prinsip ini menggunakan prinsip inheritances sehingga class child dapat bersifat seperti parent class misalnya memerlukan inputan yang sama dan memiliki hasil atau tipe keluaran yang sama. Pada kasus kami, LSP ditemukan pada class Teacher Test dan Student Test yang memiliki atribut dan fungsi dari Test Paper
4. I (Interface Segregation Principle) : kami membangun interface InterfacePengajar dan InterfaceSiswa untuk membedakan antara interface yang diimplementasikan oleh Pengajar dan Siswa. Hal ini dikarenakan Pengajar dan Siswa memiliki hak akses pengguna yang berbeda-beda.
5. D (Dependency Inversion Principle) : kami membangun abstract type Scoring dan membuat method yang ada didalamnya, dalam kasus ini kami menambahkan method viewScore(). Kemudian, kami mendefinisikan bahwa class Pengajar, Siswa, dan WaliMurid memiliki dependency pada abstract type Scoring.