

TUGAS 5

Arsitektur Perangkat Lunak

SIMADA
(Sistem Manajemen Sumber Daya Wakanda)

untuk:

Auralea Alvinia S

Dipersiapkan oleh:

Kelompok 11

Ferdinand Gabe Tua Sinaga	13523051
Muhammad Aufa Farabi	13523023
Buege Mahara Putra	13523037
Ferdin Arsenarendra Purtadi	13523117
Muhammad Iqbal Haidar	13523111

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
JL. GANESA 10, BANDUNG 40132

2024

Daftar Isi

Daftar Isi	2
Daftar Gambar	3
1 Style/Pattern Arsitektur Acuan (Layered Architecture)	4
2 Model Arsitektur Perangkat Lunak	4
2.1 Logical View	4
2.2 Development View	5
2.3 Physical View	5

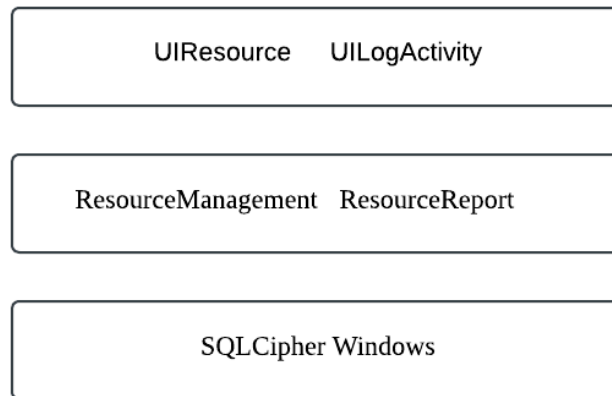
//

Daftar Gambar

<i>Gambar 1. Contoh Arsitektur Layered Architecture</i>	4
<i>Gambar 2.1. Logical View dengan Class Diagram</i>	4
<i>Gambar 2.2. Development View dengan Component Diagram</i>	6
<i>Gambar 2.3.1. Physical View dengan Deployment Diagram</i>	6
<i>Gambar 2.3.2. Artifact Manifest</i>	7

1 *Style/Pattern* Arsitektur Acuan (Layered Architecture)

Untuk Aplikasi SIMADA, *Style/Pattern* Arsitektur yang dijadikan acuan adalah Layered Architecture. style ini karena arsitektur ini memungkinkan pemisahan tanggung jawab yang jelas antar lapisan sehingga memudahkan pengelolaan dan pengembangan sistem. Dengan adanya pembagian lapisan, setiap komponen dapat diatur secara modular dan independen, yang meningkatkan fleksibilitas dalam pengembangan. Selain itu, Layered Architecture memungkinkan penggantian lapisan secara individual selama antarmuka tetap konsisten, yang berarti bahwa pemeliharaan dan peningkatan sistem dapat dilakukan tanpa mengganggu keseluruhan arsitektur.

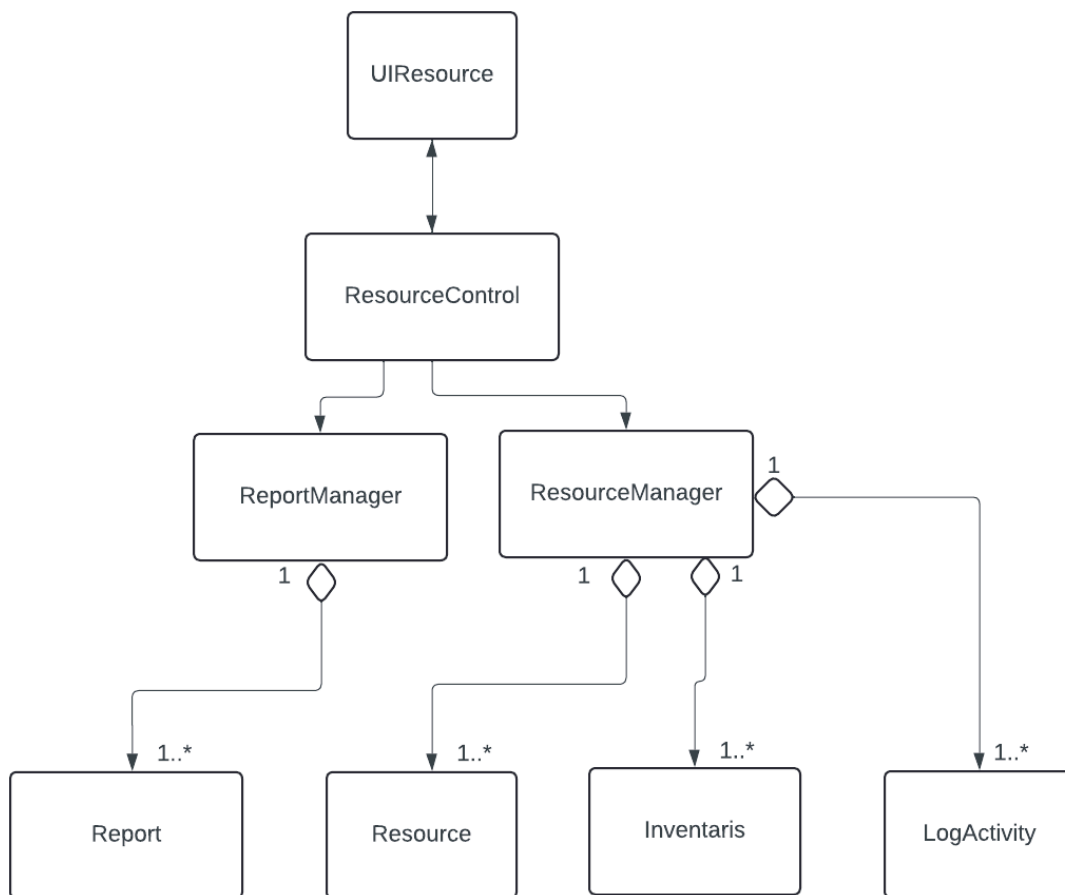


Gambar 1. Contoh Arsitektur Layered Architecture

2 Model Arsitektur Perangkat Lunak

2.1 *Logical View*

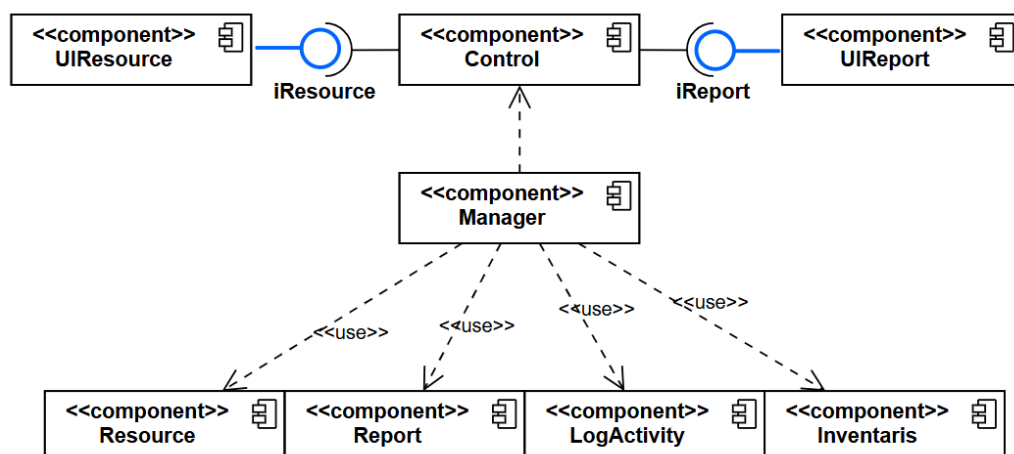
kami memakai model *Class Diagram* untuk *Logical View* karena *Class Diagram* dapat menunjukkan struktur sistem dengan jelas dan dapat menggambarkan relasi antar objek dari kelas masing-masing pada sistem kami.



Gambar 2.1. Logical View dengan Class Diagram

2.2 Development View

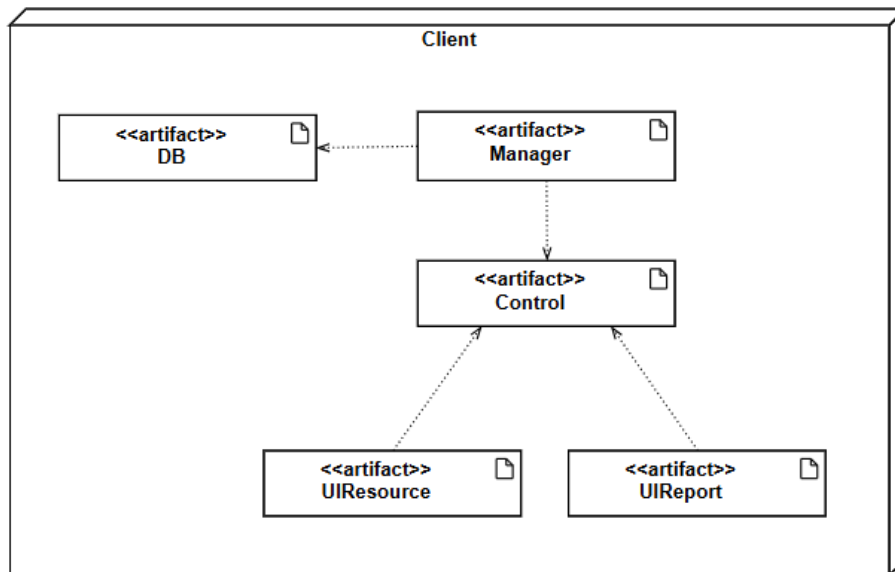
Kami memilih Component Diagram untuk menunjukkan implementasi dari komponen-komponen hasil dari dekomposisi software yang telah dibuat.



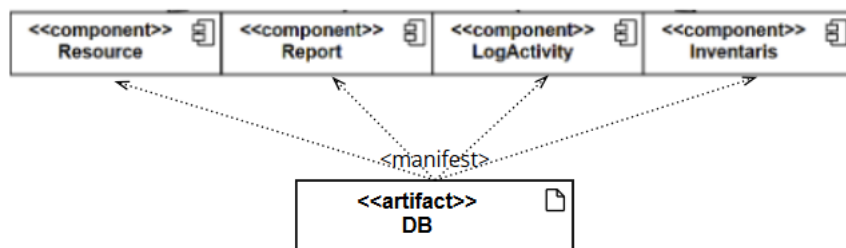
Gambar 2.2. Development View dengan Component Diagram

2.3 Physical View

Kami memilih *Deployment Diagram* karena cocok untuk menunjukkan bagaimana komponen perangkat lunak di *deploy* ke perangkat keras, memvisualisasikan interaksi antar *node*, dan memperjelas distribusi komponen serta kebutuhan infrastruktur sistem di lingkungan nyata.



Gambar 2.3.1. Physical View dengan Deployment Diagram



Gambar 2.3.2. Artifact Manifest

