



REPORT CHAPTER 7

FULL-STACK WEB DEVELOPMENT CLASS

This report is belong to student with name :

Valdryan Ivandito

(valdryan05@gmail.com)

Who passed Chapter 7 in Full-stack Web Development Class with the following result :

General Aspect (Scale 0 - 100)	
Effective Class, Method, and Variable Names <i>Names chosen for classes, methods, and variables should effectively convey the purpose and meaning of the named entity.</i>	90
Effective Top-down Decomposition of Algorithms <i>Code duplication should be avoided by factoring out common code into separate routines.</i> <i>Routines should be highly cohesive. Each routine should perform a single task or a small number of highly related tasks. Routines that perform multiple tasks should call different subroutines to perform each subtask. Routines should be relatively short in most cases. [Rule of Thumb: Many routines will be less than 20 lines. Almost all routines will be less than 50 lines. Routines longer than 100 lines should be rare].</i>	85
Code Layout Should Be Readable and Consistent <i>The layout of your code should be readable and consistent. This means things like placement of curly braces, code indentation, wrapping of long lines, layout of parameter lists, etc.</i>	90
Effective source tree directory structure <i>The source code for your project should be effectively organized into subdirectories. Something along the lines of that discussed in class would be appropriate.</i>	90
Effective file organization <i>Your source code should be effectively organized into multiple files. Each class should be placed in a separate file (or two separate files, .h and .cpp). Lumping all of your code in one or two files is not acceptable.</i>	90
Correct exception handling <i>Your program should handle exceptions properly as described in the project specification.</i>	90

**Specific Acceptance Criteria****Merancang Arsitektur dalam Web API**

Mampu menerapkan Design Pattern MVC pada monolith dashboard	✓
Mampu menerapkan Design Pattern MCR untuk pembuatan JSON API	✓
Mampu mengimplementasikan template engine ke dashboard monolithic	✓
Menggunakan Asynchronous untuk proses-proses dengan antrian lebih dari 5	✓
Mampu melakukan proteksi terhadap suatu page yang diperlukan untuk login terlebih dahulu (Monolith)	✓
Mampu melakukan proteksi terhadap suatu endpoint dan melakukan pengecekan terhadap token (RESTful API)	✓

General Acceptance Criteria

Push ke gitlab untuk direview	✓
Completed Tasks	✗

Chapter 7 Overall Score (Scale 0 - 100)**85****Notes From Facil**

Halamannya sudah berjalan ya Valdryan, next utk bikin multiplayer gausah pakai socket.io utk yang sederhana karena memang perlu pemahaman lebih kalau mau pakai module itu. Config by database aja. But overall, nice try yaa!!!



Binar Academy

PT. Lentera Bangsa Benderang
The Breeze BSD, Blok L30
Jl. Grand Boulevard, BSD Green Office Park
BSD City, Tangerang 15345

*Sincerely,
Binar Academy*