



## REPORT CHAPTER 8

### FULL-STACK WEB DEVELOPMENT CLASS

This report is belong to student with name :

**Valdryan Ivandito**

( valdryan05@gmail.com )

Who passed Chapter 8 in Full-stack Web Development Class with the following result :

General Aspect (Scale 0 - 100)	
<b>Effective Class, Method, and Variable Names</b> <i>Names chosen for classes, methods, and variables should effectively convey the purpose and meaning of the named entity.</i>	90
<b>Effective Top-down Decomposition of Algorithms</b> <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>	90
<b>Code Layout Should Be Readable and Consistent</b> <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
<b>Effective source tree directory structure</b> <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
<b>Effective file organization</b> <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
<b>Correct exception handling</b> <i>Your program should handle exceptions properly as described in the project specification.</i>	90



Specific Acceptance Criteria	
<b>Menyusun Dok API</b>	
Mampu membuat dokumentasi API dengan CRUD menggunakan plugin swagger	✓
Menerapkan sistem monolith pada dashboard dengan menggunakan Design pattern MVC	✓
<b>Menerapkan SPA 1</b>	
Mampu menginisiasi dan menerapkan konsep component, state dan property pada react	✓
Membedakan project yang bersifat monolith atau terpisah	✓

General Acceptance Criteria	
Push ke gitlab untuk direview	✓
Completed Tasks	✓

Chapter 8 Overall Score (Scale 0 - 100)
<b>96</b>

Notes From Facil
<i>Sudah baik Valdyran, dari saya tidak ada catatan.</i>

Sincerely,  
Binar Academy