



## REPORT CHAPTER 4

### FULL-STACK WEB DEVELOPMENT CLASS

This report is belong to student with name :

**Valdryan Ivandito**

( valdryan05@gmail.com )

Who passed Chapter 4 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>	99
<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>	98
<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>	96
<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>	98
<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>	98
<b>Correct exception handling</b> <i>Your program should handle exceptions properly as described in the project specification.</i>	95



Specific Acceptance Criteria	
<b>Menerapkan Pemrograman Berbasis Objek</b>	
Menampilkan hasil olahan data dari javascript ke HTML dengan DOM	✓
Class	✗
Object	✗
Function & method	✓
Encapsulation	✗
Abstraction	✗
Inheritance	✗
Constructor	✓
Properties	✓
Polymorphism	✗
Visibility Modifier	✓
Interface	✓
Primitive Data Type	✓
Variable	✓
Type Data Reference	✓
IF	✓
IF ELSE	✓
WHEN	✓
Looping	✗
Penulisan kode menggunakan IDE	✓



### General Acceptance Criteria

Push ke gitlab untuk direview	✓
Completed Tasks	✓

### Chapter 4 Overall Score (Scale 0 - 100)

**87**

### Notes From Facil

*Halaman sudah rapih, functionality berjalan baik,so far udah oke. Note-nya paling next approachnya coba menggunakan OOP ya. Yang saat ini masih procedural programming. Honorable mention, good job touchup untuk backsoundnya, so creative. Like it!!*

Sincerely,  
Binar Academy