

## DAFTAR LAMPIRAN

### Lampiran 1 – Daftar Riwayat Hidup

#### Data Pribadi

Nama : Sulistri Budi Laksono

Tempat, Tanggal Lahir : Bekasi, 26 April 2000

Jenis Kelamin : Laki - laki

Agama : Islam

Kewarganegaraan : Indonesia

Alamat : JL Balinda IV Blok C No 728 Kebalen, Babelan, Kab. Bekasi,  
Jawa barat 17610

Nomor Telepon / HP : 0895-2960-4280

E-mail : sulistribudilaksono@gmail.com

#### Riwayat Pendidikan

Periode (Tahun)			Sekolah/Institusi	Jurusan	Jenjang Pendidikan
2007	-	2012	SDN 03 Kebalen	-	SD
2013	-	2015	SMP Islam Almanar	-	SMP
2016	-	2018	SMK Mutiara 17 Agustus	TKJ (Teknik Komputer Jaringan)	SMK

## Lampiran 2 – Surat Balasan Penelitain Dari SMK Mutiara !7 Agustus



# YAYASAN MUTIARA 17 AGUSTUS SEKOLAH MENENGAH KEJURUAN MUTIARA 17 AGUSTUS

STATUS: TERAKREDITASI, BAP-S/M No.02.00/347/SK/BAP-SM/XI/2017

Jl. Perjuangan Kav.M 1 Harapan Baru, Telp.(021) 88967868 Po.Box 182  
Bekasi 17000

KECAMATAN BEKASI UTARA – KOTA BEKASI- JAWA BARAT

## SURAT KETERANGAN

003/SMK M17A/ VI/2022

### Yang bertanda tangan di bawah ini :

Nama : Paryono  
NIP : -  
Jabatan : Kepala Sekolah

Dengan ini menerangkan bahwa nama – nama dibawah ini telah melakukan penelitian tugas mata kuliah Seminar Proposal di SMK Mutiara 17 Agustus Kota Bekasi terhitung dari :

21 Maret 2022 sampai dengan saat ini.

No	Nama	NIM	Program Studi
1	Putra Dwi Andhika	20180801100	Teknik Informatika
2	S. Rimbayu Tia Pratika	20180801111	
3	Alfrizal Panji Mahendra	20180801070	
4	Sulistri Budi Laksono	20180801086	
5	Bogi Prasetyo	20180801034	

Demikian keterangan ini dibuat untuk diketahui dan dipergunakan sebagaimana mestinya.

### Lampiran 3 – Source Code Website Absensi (Autoload.php)

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

/*
| -----
| AUTO-LOADER
| -----
| This file specifies which systems should be loaded by default.
|
| In order to keep the framework as light-weight as possible only the
| absolute minimal resources are loaded by default. For example,
| the database is not connected to automatically since no assumption
| is made regarding whether you intend to use it. This file lets
| you globally define which systems you would like loaded with every
| request.
|
| -----
| Instructions
| -----
|
| These are the things you can load automatically:
|
| 1. Packages
| 2. Libraries
| 3. Drivers
| 4. Helper files
| 5. Custom config files
```

```

| 6. Language files
| 7. Models
|
| */
|
| /*
| -----
| Auto-load Packages
| -----
| Prototype:
|
| $autoload['packages'] = array(APPPATH.'third_party', '/usr/local/shared');
|
| */
$autoload['packages'] = array();
|
| /*
| -----
| Auto-load Libraries
| -----
|
| These are the classes located in system/libraries/ or your
| application/libraries/ directory, with the addition of the
| 'database' library, which is somewhat of a special case.
|
| Prototype:
|
| $autoload['libraries'] = array('database', 'email', 'session');
|
| You can also supply an alternative library name to be assigned

```

```
| in the controller:
|
|     $autoload['libraries'] = array('user_agent' => 'ua');
*/
$autoload['libraries'] = array('database', 'session');

/*
| -----
| Auto-load Drivers
| -----
| These classes are located in system/libraries/ or in your
| application/libraries/ directory, but are also placed inside their
| own subdirectory and they extend the CI_Driver_Library class. They
| offer multiple interchangeable driver options.
|
| Prototype:
|
|     $autoload['drivers'] = array('cache');
|
| You can also supply an alternative property name to be assigned in
| the controller:
|
|     $autoload['drivers'] = array('cache' => 'cch');
|
*/
$autoload['drivers'] = array();

/*
| -----
```

```

| Auto-load Helper Files
| -----
| Prototype:
|
|   $autoload['helper'] = array('url', 'file');
*/
$autoload['helper'] = array('url', 'form', 'sf_helper');

/*
| -----
| Auto-load Config files
| -----
| Prototype:
|
|   $autoload['config'] = array('config1', 'config2');
|
| NOTE: This item is intended for use ONLY if you have created custom
| config files. Otherwise, leave it blank.
|
*/
$autoload['config'] = array();

/*
| -----
| Auto-load Language files
| -----
| Prototype:
|
|   $autoload['language'] = array('lang1', 'lang2');

```

```

|
| NOTE: Do not include the "_lang" part of your file. For example
| "codeigniter_lang.php" would be referenced as array('codeigniter');
|
|
| */
$autoload['language'] = array();
|
|
| /*
| -----
| Auto-load Models
| -----
|
| Prototype:
|
| $autoload['model'] = array('first_model', 'second_model');
|
| You can also supply an alternative model name to be assigned
| in the controller:
|
| $autoload['model'] = array('first_model' => 'first');
|
| */
$autoload['model'] = array('M_user', 'M_notif');

```

#### Lampiran 4 – Source Code Website Absensi (config.php)

75

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');
|
|
| /*

```

```
|-----  
| Display Debug backtrace  
|-----  
|  
| If set to TRUE, a backtrace will be displayed along with php errors. If  
| error_reporting is disabled, the backtrace will not display, regardless  
| of this setting  
|  
*/  
defined('SHOW_DEBUG_BACKTRACE') or define('SHOW_DEBUG_BACKTRACE', TRUE);  
  
define('WEBNAME', 'Absensi siswa');  
$webname = 'thisisimyglobalvariable';  
  
/*  
|-----  
| File and Directory Modes  
|-----  
|  
| These prefs are used when checking and setting modes when working  
| with the file system. The defaults are fine on servers with proper  
| security, but you may wish (or even need) to change the values in  
| certain environments (Apache running a separate process for each  
| user, PHP under CGI with Apache suEXEC, etc.). Octal values should  
| always be used to set the mode correctly.  
|  
*/  
defined('FILE_READ_MODE') or define('FILE_READ_MODE', 0644);  
defined('FILE_WRITE_MODE') or define('FILE_WRITE_MODE', 0666);
```



```

defined('DIR_READ_MODE')    or define('DIR_READ_MODE', 0755);
defined('DIR_WRITE_MODE')  or define('DIR_WRITE_MODE', 0755);

/*
|-----|
| File Stream Modes
|-----|
|
| These modes are used when working with fopen()/popen()
|
*/

defined('FOPEN_READ')      or define('FOPEN_READ', 'rb');
defined('FOPEN_READ_WRITE') or define('FOPEN_READ_WRITE',
'r+b');

defined('FOPEN_WRITE_CREATE_DESTRUCTIVE') or
define('FOPEN_WRITE_CREATE_DESTRUCTIVE', 'wb'); // truncates existing file
data, use with care

defined('FOPEN_READ_WRITE_CREATE_DESTRUCTIVE') or
define('FOPEN_READ_WRITE_CREATE_DESTRUCTIVE', 'w+b'); // truncates existing
file data, use with care

defined('FOPEN_WRITE_CREATE') or
define('FOPEN_WRITE_CREATE', 'ab');

defined('FOPEN_READ_WRITE_CREATE') or
define('FOPEN_READ_WRITE_CREATE', 'a+b');

defined('FOPEN_WRITE_CREATE_STRICT') or
define('FOPEN_WRITE_CREATE_STRICT', 'xb');

defined('FOPEN_READ_WRITE_CREATE_STRICT') or
define('FOPEN_READ_WRITE_CREATE_STRICT', 'x+b');

/*
|-----|
| Exit Status Codes

```

```
|-----  
|  
| Used to indicate the conditions under which the script is exit()ing.  
| While there is no universal standard for error codes, there are some  
| broad conventions. Three such conventions are mentioned below, for  
| those who wish to make use of them. The CodeIgniter defaults were  
| chosen for the least overlap with these conventions, while still  
| leaving room for others to be defined in future versions and user  
| applications.  
|  
| The three main conventions used for determining exit status codes  
| are as follows:  
|  
| Standard C/C++ Library (stdlibc):  
|     http://www.gnu.org/software/libc/manual/html\_node/Exit-Status.html  
|     (This link also contains other GNU-specific conventions)  
| BSD sysexits.h:  
|     http://www.gsp.com/cgi-bin/man.cgi?section=3&topic=sysexits  
| Bash scripting:  
|     http://tldp.org/LDP/abs/html/exitcodes.html  
|  
|*/  
defined('EXIT_SUCCESS')      or define('EXIT_SUCCESS', 0); // no errors  
defined('EXIT_ERROR')        or define('EXIT_ERROR', 1); // generic error  
defined('EXIT_CONFIG')       or define('EXIT_CONFIG', 3); // configuration  
error  
defined('EXIT_UNKNOWN_FILE') or define('EXIT_UNKNOWN_FILE', 4); // file not  
found  
defined('EXIT_UNKNOWN_CLASS') or define('EXIT_UNKNOWN_CLASS', 5); // unknown  
class
```

```
defined('EXIT_UNKNOWN_METHOD') or define('EXIT_UNKNOWN_METHOD', 6); // unknown
class member

defined('EXIT_USER_INPUT')      or define('EXIT_USER_INPUT', 7); // invalid
user input

defined('EXIT_DATABASE')        or define('EXIT_DATABASE', 8); // database
error

defined('EXIT__AUTO_MIN')       or define('EXIT__AUTO_MIN', 9); // lowest
automatically-assigned error code

defined('EXIT__AUTO_MAX')       or define('EXIT__AUTO_MAX', 125); // highest
automatically-assigned error code
```