

DAFTAR PUSTAKA

- Aji, Supriyanto. 2007. *Web Dengan HTML dan XML*. Graha Ilmu , Yogyakarta.
- Al Bahra, 2006. *Rekayasa Perangkat Lunak*, Andi Offset, Graha Ilmu, Yogyakarta.
- Arief M Rudyanto, 2011. *Pemograman Web Dinamis Menggunakan PHP dan MySQL*. Andi Offset, Yogyakarta.
- Edhy. Sutanta. 2004. *Sistem Basis Data*. Graha Ilmu, Yogyakarta.
- Jogiyanto, 2002. *Analisis Dan Desain Sistem Informasi*. Andi Offset, Yogyakarta.
- Kusrini, 2007. *Strategi Perancangan dan Pengelolaan Basis Data*. Andi Offset, Yogyakarta.
- London, 2007. *Analisa dan Desain Sistem Informasi*. Andi Offset, Yogyakarta.
- Yakub, 2008. *Sistem Basis Data Tutorial Konseptual*, Graha Ilmu, Yogyakarta.
- Yogi Wicaksono, 2008. *Perancangan Sistem Informasi Dan Aplikasinya*. Gramedia Pustaka Utama, Jakarta.

LISTING PROGRAM

```
<?php
// Version
define('VERSION', '2.0.3.1');

// Configuration
if (is_file('config.php')) {
    require_once('config.php');
}

// Install
if (!defined('DIR_APPLICATION')) {
    header('Location: install/index.php');
    exit;
}

// Startup
require_once(DIR_SYSTEM . 'startup.php');

// Registry
$registry = new Registry();

// Loader
$loader = new Loader($registry);
$registry->set('load', $loader);

// Config
$config = new Config();
$registry->set('config', $config);

// Database
$db = new DB(DB_DRIVER, DB_HOSTNAME, DB_USERNAME,
DB_PASSWORD, DB_DATABASE);
$registry->set('db', $db);

// Store
if (isset($_SERVER['HTTPS']) && (($_SERVER['HTTPS'] == 'on' ||
($_SERVER['HTTPS'] == '1')))) {
    $store_query = $db->query("SELECT * FROM " . DB_PREFIX . "store
WHERE REPLACE(`ssl`, 'www.', '') = " . $db->escape('https://' .
str_replace('www.', '', $_SERVER['HTTP_HOST'])) .
 rtrim(dirname($_SERVER['PHP_SELF']), '%\') . '/') . """);
} else {
```

```

        $store_query = $db->query("SELECT * FROM " . DB_PREFIX . "store
WHERE REPLACE(`url`, 'www.', '') = '' . $db->escape('http://' .
str_replace('www.', '', $_SERVER['HTTP_HOST'])) .
 rtrim(dirname($_SERVER['PHP_SELF']), '%\') . '/') . """);
    }

    if ($store_query->num_rows) {
        $config->set('config_store_id', $store_query->row['store_id']);
    } else {
        $config->set('config_store_id', 0);
    }

    // Settings
    $query = $db->query("SELECT * FROM `". DB_PREFIX . "setting` WHERE
store_id = '0' OR store_id = '' . (int)$config->get('config_store_id') . "" ORDER
BY store_id ASC");

    foreach ($query->rows as $result) {
        if (!$result['serialized']) {
            $config->set($result['key'], $result['value']);
        } else {
            $config->set($result['key'], unserialize($result['value']));
        }
    }

    if (!$store_query->num_rows) {
        $config->set('config_url', HTTP_SERVER);
        $config->set('config_ssl', HTTPS_SERVER);
    }

    // Url
    $url = new Url($config->get('config_url'), $config->get('config_secure') ?
$config->get('config_ssl') : $config->get('config_url'));
    $registry->set('url', $url);

    // Log
    $log = new Log($config->get('config_error_filename'));
    $registry->set('log', $log);

    function error_handler($errno, $errstr, $errfile, $errline) {
        global $log, $config;

        // error suppressed with @
        if (error_reporting() === 0) {
            return false;
        }
    }

```

```

switch ($errno) {
    case E_NOTICE:
    case E_USER_NOTICE:
        $error = 'Notice';
        break;
    case E_WARNING:
    case E_USER_WARNING:
        $error = 'Warning';
        break;
    case E_ERROR:
    case E_USER_ERROR:
        $error = 'Fatal Error';
        break;
    default:
        $error = 'Unknown';
        break;
}

if ($config->get('config_error_display')) {
    echo '<b>' . $error . '</b>: ' . $errstr . ' in <b>' . $errfile . '</b> on
line <b>' . $errline . '</b>';
}

if ($config->get('config_error_log')) {
    $log->write('PHP ' . $error . ': ' . $errstr . ' in ' . $errfile . ' on line ' .
$errline);
}

return true;
}

// Error Handler
set_error_handler('error_handler');

// Request
$request = new Request();
$registry->set('request', $request);

// Response
$response = new Response();
$response->addHeader('Content-Type: text/html; charset=utf-8');
$response->setCompression($config->get('config_compression'));
$registry->set('response', $response);

// Cache
$cache = new Cache('file');
$registry->set('cache', $cache);

```

```

// Session
$session = new Session();
$registry->set('session', $session);

// Language Detection
$languages = array();

$query = $db->query("SELECT * FROM `". DB_PREFIX . "language` WHERE
status = '1'");

foreach ($query->rows as $result) {
    $languages[$result['code']] = $result;
}

if (isset($session->data['language']) && array_key_exists($session->
data['language'], $languages)) {
    $code = $session->data['language'];
} elseif (isset($request->cookie['language']) && array_key_exists($request->
cookie['language'], $languages)) {
    $code = $request->cookie['language'];
} else {
    $detect = "";

    if (isset($request->server['HTTP_ACCEPT_LANGUAGE']) && $request->
server['HTTP_ACCEPT_LANGUAGE']) {
        $browser_languages = explode(',', $request->
server['HTTP_ACCEPT_LANGUAGE']);

        foreach ($browser_languages as $browser_language) {
            foreach ($languages as $key => $value) {
                if ($value['status']) {
                    $locale = explode(',', $value['locale']);

                    if (in_array($browser_language, $locale)) {
                        $detect = $key;
                        break 2;
                    }
                }
            }
        }
    }

    $code = $detect ? $detect : $config->get('config_language');
}

if (!isset($session->data['language']) || $session->data['language'] != $code) {

```

```

        $session->data['language'] = $code;
    }

    if (!isset($request->cookie['language']) || $request->cookie['language'] != $code) {
        setcookie('language', $code, time() + 60 * 60 * 24 * 30, '/', $request-
        >server['HTTP_HOST']);
    }

    $config->set('config_language_id', $languages[$code]['language_id']);
    $config->set('config_language', $languages[$code]['code']);

    // Language
    $language = new Language($languages[$code]['directory']);
    $language->load($languages[$code]['directory']);
    $registry->set('language', $language);

    // Document
    $registry->set('document', new Document());

    // Customer
    $customer = new Customer($registry);
    $registry->set('customer', $customer);

    // Customer Group
    if ($customer->isLoggedIn()) {
        $config->set('config_customer_group_id', $customer->getGroupId());
    } elseif (isset($session->data['customer']) && isset($session-
    >data['customer']['customer_group_id'])) {
        // For API calls
        $config->set('config_customer_group_id', $session-
        >data['customer']['customer_group_id']);
    } elseif (isset($session->data['guest']) && isset($session-
    >data['guest']['customer_group_id'])) {
        $config->set('config_customer_group_id', $session-
        >data['guest']['customer_group_id']);
    }

    // Tracking Code
    if (isset($request->get['tracking'])) {
        setcookie('tracking', $request->get['tracking'], time() + 3600 * 24 * 1000,
        '/');

        $db->query("UPDATE `". DB_PREFIX . "marketing` SET clicks =
        (clicks + 1) WHERE code = '" . $db->escape($request->get['tracking']) . "'");
    }

    // Affiliate

```

```

$registry->set('affiliate', new Affiliate($registry));

// Currency
$registry->set('currency', new Currency($registry));

// Tax
$registry->set('tax', new Tax($registry));

// Weight
$registry->set('weight', new Weight($registry));

// Length
$registry->set('length', new Length($registry));

// Cart
$registry->set('cart', new Cart($registry));

// Encryption
$registry->set('encryption', new Encryption($config->get('config_encryption')));

//OpenBay Pro
$registry->set('openbay', new Openbay($registry));

// Event
$event = new Event($registry);
$registry->set('event', $event);

$query = $db->query("SELECT * FROM ".DB_PREFIX."event");

foreach ($query->rows as $result) {
    $event->register($result['trigger'], $result['action']);
}

// Front Controller
$controller = new Front($registry);

// Maintenance Mode
$controller->addPreAction(new Action('common/maintenance'));

// SEO URL's
$controller->addPreAction(new Action('common/seo_url'));

// Router
if (isset($request->get['route'])) {
    $action = new Action($request->get['route']);
} else {
    $action = new Action('common/home');
}

```

```
}  
  
// Dispatch  
$controller->dispatch($action, new Action('error/not_found'));  
  
// Output  
$response->output();
```

