

EVALUASI PHP STANDARDS RECOMMENDATIONS PADA PROYEK SHARIF JUDGE

NICHOLAS KHRISNA SANDYAWAN—6181801060

1 Data Tugas Akhir

Pembimbing utama/tunggal: **Pascal Alfadian Nugroho**

Pembimbing pendamping: -

Kode Topik : **PAN5401**

Topik ini sudah dikerjakan selama : **2 semester**

Pengambilan pertama kali topik ini pada : Semester **54 - Genap 22/23**

Pengambilan pertama kali topik ini di kuliah : **Skripsi 1**

Tipe Laporan : **B** - Dokumen untuk reviewer pada presentasi dan **review Tugas Akhir 1**

2 Latar Belakang

Pengembangan aplikasi berbasis web dengan bahasa PHP cukup populer di kalangan pengembang web. Bahkan tersedia banyak *framework* yang dapat digunakan untuk memudahkan pengembangannya. Walaupun demikian, masih cukup sering dijumpai masalah dalam pengembangan. Salah satunya adalah penulisan kode program yang tidak konsisten karena belum ditentukan aturan atau standar penulisan tertentu. Hal ini membuat suatu proyek aplikasi web menjadi rumit dan sulit dipelihara, terutama jika melibatkan banyak pengembang.

Salah satu proyek aplikasi berbasis web yang digunakan di jurusan Informatika UNPAR adalah SharIF Judge. SharIF Judge merupakan aplikasi berbasis web yang dapat digunakan untuk menilai kode program dalam bahasa C, C++, Java, dan Python. Aplikasi ini ditulis dengan *framework* PHP CodeIgniter dan bagian *backend* dibuat dengan BASH. SharIF Judge yang dibahas pada tugas akhir ini adalah *fork* dari Sharif Judge yang dibuat oleh Mohammad Javad Naderi. Versi *fork* ini sudah dikembangkan sesuai kebutuhan Informatika UNPAR untuk digunakan dalam proses penilaian pada beberapa mata kuliah. Tidak menutup kemungkinan akan ada perbaikan atau penambahan fitur seiring berjalannya waktu. Hal ini dapat melibatkan lebih dari satu orang. Akan lebih baik jika ditentukan suatu aturan atau standar dalam pengembangannya.

PHP Standards Recommendation (PSR) adalah kumpulan standar penulisan PHP yang dibuat oleh PHP Framework Interop Group. Standar-standar ini berisi aturan dan rekomendasi penulisan kode program PHP yang dapat membantu pengembangan aplikasi agar lebih konsisten. Pada saat dokumen ini dibuat, terdapat 14 bab PSR yang berlaku untuk digunakan (*Accepted*) antara lain:

- PSR-01: Basic Coding Standard
- PSR-03: Logger Interface
- PSR-04: Autoloading Standard
- PSR-06: Caching Interface
- PSR-07: HTTP Message Interface
- PSR-11: Container Interface
- PSR-12: Extended Coding Style Guide
- PSR-13: Hypermedia Links

- PSR-14: Event Dispatcher
- PSR-15: HTTP Handlers
- PSR-16: Simple Cache
- PSR-17: HTTP Factories
- PSR-18: HTTP Client
- PSR-20: Clock

Standar yang digunakan hanya standar dari bab-bab yang berstatus “*Accepted*”. Bab dengan status lain yaitu “*Draft*”, “*Abandoned*”, dan “*Deprecated*” tidak akan digunakan sehingga tidak dicantumkan dalam dokumen tugas akhir.

Pada tugas akhir ini, kode-kode program PHP pada SharIF Judge akan diperiksa dan dievaluasi seberapa patuh SharIF Judge terhadap PSR berdasarkan jumlah standar yang sudah dipenuhi. Pemeriksaan PSR-12 dilakukan dengan bantuan *tools* PHP CS Fixer sementara pemeriksaan PSR lainnya dilakukan secara manual. Setelah itu, kode-kode yang belum memenuhi akan diberikan rekomendasi sesuai PSR berdasarkan hasil evaluasi tersebut.

3 Rumusan Masalah

- Bagaimana tingkat kepatuhan kode PHP pada SharIF Judge terhadap PSR?
- Rekomendasi perbaikan apa yang dapat diberikan pada kode PHP SharIF Judge untuk meningkatkan jumlah aturan PSR yang terpenuhi?

4 Tujuan

- Mengukur tingkat kepatuhan kode PHP pada SharIF Judge terhadap PSR.
- Membuat rekomendasi perbaikan pada kode PHP SharIF Judge agar meningkatkan jumlah PSR yang terpenuhi.

5 Detail Perkembangan Pengerjaan Tugas Akhir

Detail bagian pekerjaan tugas akhir sesuai dengan rencana kerja/laporan perkembangan terakhir :

1. Mempelajari SharIF Judge.

Status : Ada sejak rencana kerja tugas akhir.

Hasil : Studi tentang SharIF Judge sejauh ini adalah membedah *file-file* PHP yang menjadi lingkup evaluasi. Semua *file* yang berasal dari *framework* atau *library* tidak termasuk ke dalamnya. Hanya *file-file* dalam *folder application/* dan memiliki tanda “SharIF Judge” di dalamnya yang menjadi lingkup evaluasi. Eksperimen menjalankan SharIF Judge secara lokal sedang dalam tahap percobaan karena masih belum berhasil. Percobaan ini dilakukan berdampingan dengan pengecekan dan evaluasi PSR.

2. Melakukan studi literatur mengenai PSR dan PHP CS Fixer.

Status : Ada sejak rencana kerja rencana tugas akhir namun ada perubahan dari PHP Linter menjadi PHP CS Fixer.

Hasil : Berikut adalah rincian hasilnya.

- (a) Penulisan standar-standar PSR ini diterjemahkan dari bahasa Inggris ke bahasa Indonesia. Salah satu tantangan dalam menerjemahkan PSR ini adalah tidak setiap aturan ditulis dalam bentuk *list* yang mudah untuk dibaca. Banyak di antaranya merupakan bagian dari suatu paragraf yang berkelanjutan dengan konteksnya masing-masing sehingga penjelasan atau deskripsinya perlu dicantumkan pula dalam Landasan Teori agar tidak ada standar yang bermakna ambigu atau tidak jelas. Untuk saat ini, semua PSR dicantumkan dalam Landasan Teori kecuali dua bab yaitu PSR-07 dan PSR-12. Hal ini dikarenakan jumlah aturan yang harus diterjemahkan dan ditulis kembali cukup banyak sehingga dikhawatirkan akan memakan waktu yang lama. Untuk sementara, hal ini dicantumkan dalam Batasan Masalah. Rencananya kedua bab ini akan ditambahkan setelah selesai dilakukan pemeriksaan dan evaluasi kepatuhan.
- (b) Ada beberapa kata kunci yang harus ditaati dalam standar PSR berdasarkan dokumen Request For Comments (RFC) 2119, yaitu “MUST”, “MUST NOT”, “SHOULD”, “SHOULD NOT”, dan “MAY”. Ada juga beberapa kata kunci lain yang disertakan namun sangat jarang digunakan, yaitu “REQUIRED”, “SHALL”, “SHALL NOT”, “RECOMMENDED”, dan “OPTIONAL”. Setiap kata kunci ditulis dengan huruf kapital dan memiliki tingkat prioritas yang berbeda dalam hal keharusan pemenuhannya. Beberapa kata kunci memiliki tingkat yang sama, sebagai contoh “MUST”, “SHALL”, dan “REQUIRED” memiliki tingkat yang sama yaitu harus dipenuhi sesuai rekomendasi PSR. Maka dari itu, setiap kata kunci dikelompokkan berdasarkan tingkatan tersebut dan diterjemahkan ke bahasa Indonesia agar penulisan dokumen tugas akhir menjadi konsisten dan tidak mengubah makna dari standar yang asli. Misalnya, kata kunci “MUST” dan setingkatnya diterjemahkan menjadi “HARUS”. Begitu pula dengan lawan katanya “MUST NOT” diterjemahkan menjadi “TIDAK BOLEH”.
- (c) Setiap standar diberikan kode tertentu agar dapat digunakan sebagai acuan dari bab-bab lain dalam dokumen tugas akhir. Kode tersebut ditulis dengan format PSR-XXYY, di mana XX adalah nomor bab dan YY adalah nomor urut standar dalam bab tersebut. Sebagai contoh PSR-0402 berarti mengacu pada standar PSR nomor 2 dari bab PSR-04. Urutan tersebut didasarkan pada standar mana yang ditulis terlebih dahulu. Kode-kode ini tidak ada pada dokumentasi asli PSR dan hanya dibuat untuk kepentingan tugas akhir ini.
- (d) PHP Linter pada dasarnya digunakan untuk membantu melakukan pengecekan penulisan kode PHP secara keseluruhan dengan standar tertentu. Pada rencana kerja sebelumnya, PHP Linter yang digunakan adalah yang dibuat oleh pengguna GitHub dengan nama Brueggern. *Linter* ini dapat memeriksa berdasarkan PSR-02 (*Deprecated*) dan PSR-12. Fiturnya dapat menunjukkan kesalahan penulisan dan memberikan rekomendasi perbaikannya. Satu fitur lainnya yaitu perintah untuk memperbaiki kesalahan-kesalahan tersebut secara otomatis. Namun setelah dipelajari lebih lanjut, *linter* tersebut menggunakan PHP Coding Standards Fixer (PHP CS Fixer) yang dimodifikasi menjadi lebih sederhana untuk digunakan. Pada dokumentasinya, PHP CS Fixer memiliki *rule set* untuk PSR-01, PSR-02 (*Deprecated*), dan PSR-12. PHP CS Fixer dapat diatur untuk memberikan tanda bagian mana saja dari kode yang perlu diperbaiki dan juga memberikan perbaikan secara otomatis tergantung perintah yang diberikan. Dengan PHP CS Fixer, proses pemeriksaan menjadi lebih fleksibel karena perintah yang diberikan lebih beragam dibandingkan PHP Linter sebelumnya yang hanya memiliki dua perintah utama. Maka dari itu, *tool* yang digunakan adalah PHP CS Fixer.

3. Mengevaluasi tingkat kepatuhan SharIF Judge terhadap PSR.

Status : Ada sejak rencana kerja tugas akhir.

Hasil : Proses pemeriksaan dilakukan dengan cara mengecek terlebih dahulu menggunakan PHP CS Fixer lalu setelah itu diperiksa secara manual sesuai rekomendasi-rekomendasi PSR yang relevan. Dalam suatu *file*, tidak semua rekomendasi akan relevan untuk digunakan karena belum tentu

semua jenis perintah, aturan penulisan, dan fitur PHP yang diatur dalam PSR diimplementasikan dalam *file* tersebut. Berikut adalah contoh keluaran dari PHP CS Fixer pada potongan kode `application/controllers/Dashboard.php` dalam bentuk *diff* yang isinya menampilkan kode asli (bertanda “-”) dan rekomendasi perbaikannya (bertanda “+”).

```
@@ -4,66 +4,70 @@
* @file Dashboard.php
* @author Mohammad Javad Naderi <mjnaderi@gmail.com>
*/
- defined('BASEPATH') OR exit('No direct script access allowed');
+ defined('BASEPATH') or exit('No direct script access allowed');

class Dashboard extends CI_Controller
{
+   public function __construct()
+   {
+       parent::__construct();
+       if (! $this->db->table_exists('sessions')) {
+           redirect('install');
+       }
+       if (! $this->session->userdata('logged_in')) { // if not logged in
+           redirect('login');
+       }
+       $this->load->model('notifications_model')->helper('text');
+   }

-   public function __construct()
-   {
-       parent::__construct();
-       if ( ! $this->db->table_exists('sessions'))
-           redirect('install');
-       if ( ! $this->session->userdata('logged_in')) // if not logged in
-           redirect('login');
-       $this->load->model('notifications_model')->helper('text');
-   }
```

Jika diperiksa dengan seksama, kode asli (bertanda “-”) memiliki jarak indentasi satu kali *tab* dari sisi kiri awal penulisan yang mana seharusnya dimulai dengan empat spasi untuk setiap level indentasi seperti yang ada dalam PSR-12: “Kode HARUS menggunakan indentasi sebanyak 4 spasi untuk setiap level indentasi, dan TIDAK BOLEH menggunakan *tab-tab* untuk indentasi”. Hal ini mungkin tidak terlihat secara sekilas oleh mata sehingga harus diperiksa kembali setiap indentasinya. Hasil rekomendasi perbaikan yang diberikan keluaran (bertanda “+”) di atas menampilkan indentasi sepanjang empat spasi dan kode aslinya (bertanda “-”) sepanjang satu *tab*. Hal ini akan lebih jelas jika diperiksa secara langsung melalui terminal.

4. Mengimplementasikan rekomendasi PSR sesuai hasil evaluasi.

Status : Ada sejak rencana kerja tugas akhir.

Hasil : Proses implementasi berjalan beriringan dengan pemeriksaan. Dengan bantuan PHP CS Fixer, hasil keluaran dalam contoh potongan kode sebelumnya dapat diimplementasikan secara langsung untuk memenuhi PSR-12. Perintah yang diberikan dapat diubah untuk melakukan *fix* secara langsung. Dikarenakan sejauh ini baru dilakukan pemeriksaan menyeluruh secara otomatis, belum ada contoh

rekomendasi perbaikan berdasarkan PSR yang lain selain yang diberikan PHP CS Fixer. Pemeriksaan secara manual sedang dalam tahap pengerjaan sambil mengecek standar-standar PSR yang relevan untuk setiap *file* PHP.

5. **Menguji SharIF Judge yang sudah dievaluasi dan diperbaiki.**

Status : Ada sejak rencana kerja tugas akhir.

Hasil : Belum terlaksana karena SharIF Judge belum berhasil dijalankan dan evaluasi belum selesai dilakukan.

6. **Menulis dokumen tugas akhir.**

Status : Ada sejak rencana kerja tugas akhir.

Hasil : Dokumen tugas akhir yang sudah dibuat adalah Bab 1 Pendahuluan dan sebagian besar Bab 2 Landasan Teori, terutama PSR dan PHP CS Fixer. Tahap analisis, evaluasi, implementasi, dan pengujian (Bab 3 dan 4) sedang dalam proses sesuai yang dipaparkan pada poin-poin sebelumnya.

6 Pencapaian Rencana Kerja

Langkah-langkah kerja yang berhasil diselesaikan dalam Tugas Akhir 1 ini adalah sebagai berikut:

1. Membuat dokumen tugas akhir Bab 1 Pendahuluan dan sebagian besar Bab 2 Landasan Teori.
2. Melakukan studi literasi PSR dan PHP CS Fixer.
3. Menentukan strategi untuk mengevaluasi keseluruhan PHP SharIF Judge.
4. Mengevaluasi sebagian PHP pada SharIF Judge, khususnya yang dapat dilakukan dengan bantuan PHP CS Fixer.

Bandung, 06/12/2023

Nicholas Khrisna Sandyawan

Menyetujui,

Nama: Pascal Alfadian Nugroho
Pembimbing Tunggal