

Analisis Pemanfaatan Laravel Debugger Dalam Mempermudah Penyelesaian Issue Pada Aplikasi Pusat Pengembangan Sumber Daya Manusia (PPSDM)

by John Doe

Submission date: 08-Jun-2021 09:50AM (UTC+0700)

Submission ID: 1602570730

File name: Pada_Aplikasi_Pusat_Pengembangan_Sumber_Daya_Manusia_PPSDM.pdf (1.04M)

Word count: 4200

Character count: 27599

Analisis Pemanfaatan Laravel Debugbar Dalam Mempermudah Penyelesaian Issue Pada Aplikasi Pusat Pengembangan Sumber Daya Manusia (PPSDM)

Abstract—Aplikasi PPSDM (Pusat Pengembangan Sumber Daya Manusia) merupakan sebuah portal aplikasi berbasis website dalam pengembangan sumber daya manusia di bidang PBJ (Pengadaan Barang/Jasa). Pada tahun 2019, Aplikasi ini memulai pengembangan versi aplikasi dengan melakukan perbaikan seperti UI/UX, perubahan *stack* dari Mongo DB ke MySQL menggunakan Galera serta ada penambahan fitur baru. Tim *development* mengembangkan aplikasi dengan mengerjakan *issue* yang diberikan dari *System Analyst*. Adapun *issue* tersebut berasal dari permintaan klien langsung seperti pengembangan fitur baru atau berasal dari munculnya *issue* seperti *bug* dan *error* yang ditemukan oleh pengguna setelah rilis versi aplikasi terbaru. Dalam pengerjaan *issue* dalam proyek PPSDM ini *developers* menggunakan salah satu *packages Laravel*, yaitu *Laravel Debugbar*. Sebuah *packages* sederhana yang menggunakan kelas PHP *DataCollector* yang mengumpulkan data-data tertentu menjadi sebuah informasi terkait halaman yang sedang diakses dalam aplikasi. Jenis *collector* tersebut antara lain *views*, *messages*, *exceptions*, *query*, *route*, *models*, *cache* dan *collector* lainnya. *Packages* ini akan menampilkan informasi berbentuk panel dengan *tab* sesuai dengan *collector* di bagian bawah layar setiap mengakses halaman di aplikasi. Adapun pemanfaatan *packages Laravel Debugbar* dalam kaitannya dengan pengerjaan *issue* proyek PPSDM yaitu memberikan kemudahan dengan menyediakan informasi untuk pencarian nama file *views*, *route*, dan *controller* serta dapat pula digunakan untuk melakukan *print out* sebuah *variable*. Pemanfaatan *packages* tersebut perlu dibuktikan dengan cara mengukur atau mengevaluasi menggunakan berdasarkan parameter pada model ISO/IEC 25010. Parameter tersebut ada 5 yaitu 3 subkarakteristik dari *function suitability* yaitu *functional completeness*, *functional correctness*, dan *functional appropriateness*. Sedangkan 2 karakteristik selanjutnya yaitu *performance efficiency* yaitu *time behavior* dan *resource utilization*. Adapun model ISO/IEC 25010 merupakan sebuah standar internasional terkait evaluasi kualitas sebuah perangkat lunak atau aplikasi.

Keywords—*Laravel Debugbar*, Analisis, ISO/IEC 25010

I. PENDAHULUAN

A. Latar Belakang

Dalam pengembangan aplikasi berbagai masalah muncul dan tidak bisa dideteksi diluar kendali dari tim proyek, pengguna atau klien diharapkan dapat memberikan *feedback* semaksimal mungkin untuk meningkatkan kualitas dari pengembangan aplikasi. Masalah-masalah tersebut biasa disebut dengan istilah *issue*.

Pada pengerjaan sebuah *issue* proyek pengembangan sebuah aplikasi, *developers* tentu harus dapat memahami dengan cepat maksud serta ekspektasi *issue* yang telah dikerjakan, memperkirakan alur pengerjaan serta penulisan kode, mengetahui *boundary* atau batasan kode yang akan digunakan, serta dapat mengimplementasikan kode tersebut dengan kualitas kode yang baik dan mengikuti standar proyek. Kemampuan atau *skills* yang dimiliki *developers* seiring

waktu akan bertambah dengan intensitas pengerjaan berbagai jenis *issue* yang diberikan. Semakin banyak *issue* yang dikerjakan maka tentu semakin tinggi tantangan yang akan datang serta menimbulkan persepsi dari *developers* bahwa ada kemungkinan kegagalan dalam mengerjakan *issue* tersebut. Namun dalam hal ini kegagalan tersebut bukanlah sesuatu yang dapat dihindari dalam proses pengerjaan sebuah *issue*. Beberapa penelitian mengungkapkan bahwa tantangan dalam mengerjakan sesuatu hal secara positif dapat meningkatkan motivasi dalam diri. Wilkie's (2016) dalam penelitiannya, menemukan bahwa siswa yang menerima tantangan lebih dalam *issue* matematika maka secara positif akan memberikan dampak dua sisi yaitu (kesenangan, ketertarikan) dan manfaat pembelajaran [1]. Selain itu dampak lain yang ditemukan adalah secara efektif meningkatkan motivasi siswa, performa pembelajaran, dan rasa kepuasan dalam pembelajaran [2]. Dengan kata lain *developers* memang akan menghadapi tantangan yang sulit, namun dampak yang didapatkan selama mengerjakan sebuah *issue* hingga menyelesaikannya tentunya akan bermanfaat untuk peningkatan kualitas dan kemampuan dalam bekerja.

Meningkatkan kemampuan dengan menambah intensitas dalam mengerjakan *issue* memang menjadi kewajiban bagi *developers*, karena dari penugasan tersebut *developers* akan mendapatkan lebih banyak pengalaman serta menyerap lebih banyak ilmu. Namun kemampuan yang dimiliki oleh setiap *developers* masih belum cukup. *Developers* perlu menggunakan *tools* yang relevan dan tepat agar pekerjaan membantu dalam bekerja. Ada berbagai macam jenis *tools* yang membantu dalam pengembangan sebuah perangkat lunak atau aplikasi dengan fungsi-fungsi tertentu seperti *tools communication team*, *database environment*, *code editor*, *task collaboration tools*, *version control systems* dan masih banyak *tools* lain. Salah satu *tools* yang digunakan dalam proyek yang peneliti lakukan penelitian yaitu aplikasi PPSDM (Pusat Pengembangan Sumber Daya manusia) bidang PBJ (Pengadaan Barang/Jasa) adalah *packages Laravel Debugbar*.

Laravel Debugbar merupakan sebuah *packages Laravel* yang membantu *developers* dengan mengumpulkan data-data menggunakan kelas PHP yaitu *DataCollector*[3] yang kemudian menjadi sebuah informasi yang relevan, valid, akurat dan lengkap yang dapat digunakan dalam proses *debugging* atau menulis kode. Cara kerja hingga keputusan yang dibuat oleh *developers* salah satunya berpengaruh karena informasi yang baik dan berkualitas[4]. Beberapa pemanfaatan *packages* ini dalam proses pengerjaan *issue* pada proyek PPSDM yaitu memberikan kemudahan dalam pencarian nama file *views*, *route* dan *controller* yang digunakan serta juga dapat melakukan *print out* sebuah *variable* sebagai pengganti *function* umum di *Laravel* yaitu *dump and die* atau *dd()*. *Packages* ini berbentuk sebuah panel dibagian bawah layar aplikasi, dengan masing-masing *tab* sesuai dengan *collectors* yang telah disediakan *packages* ini

- *MemoryCollector*, menampilkan penggunaan *memory* yang dihabiskan untuk menjalankan setiap halaman.
- *ExceptionsCollector*, menampilkan error *exception* (perubahan alur program dari kondisi normal ke kondisi tertentu). Ketika aplikasi sedang berjalan, error ini tidak akan menyebabkan *crash* atau aplikasi berhenti melainkan aplikasi tersebut akan berjalan tidak sesuai dengan ekspektasi, *exception* biasanya terjadi ketika adanya transaksi data ke *database*.

Packages *Laravel Debugbar* juga memiliki *collector* lain seperti *views*, *messages*, *exceptions*, *query*, *route*, *models*, *cache* dan masih banyak *collector* lainnya.

C. Pemanfaatan Packages *Laravel Debugbar* Dalam Pengerjaan *Issue* Proyek PPSDM

Issue yang dikerjakan oleh *developers* merupakan masalah yang ditemukan oleh klien atau pengguna pada saat menjalankan aplikasi setelah rilis versi terbaru aplikasi. Adapun *issue* yang dikerjakan oleh peneliti dibagikan menjadi pada saat bergabung dengan proyek PPSDM sebagai berikut.

- *Bug*, sebuah kesalahan dimana perangkat lunak tidak menjalankan seperti yang seharusnya dilakukan begitupun sebaliknya. Contoh seperti ikon pencil yang digunakan untuk menu *edit* tidak muncul.
- *Sonarqube*, masalah yang terjadi apabila kode sudah ditulis yang akan di-*merging* ke branch utama di Git melanggar *rules* yaitu terindikasi *bug*, *vulnerability*, dan *code smell*.
- *Error*, sebuah kesalahan pada aplikasi yang menyebabkan aplikasi tidak dapat menjalankan fungsinya. Contoh ketika mengakses halaman tertentu muncul error misal “*class not found*”.
- *Tasks*, termasuk juga kedalam jenis *issue* dimana tidak berkaitan dengan kesalahan dalam aplikasi, namun tugas ini perlu diselesaikan serta diimplementasikan kedalam proyek.

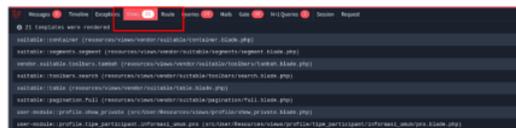
Salah satu tools yang digunakan dalam pengerjaan *issue* proyek PPSDM adalah *packages Laravel Debugbar*, berikut ini pemanfaatan *packages Laravel Debugbar* yang digunakan oleh peneliti dalam mengerjakan *issue-issue* tersebut.

- Pencarian nama file *views*, *route* dan *controller*. Tim PPSDM menggunakan *ActiveCollab* sebagai *tools* dalam mengatur manajemen proyek salah satunya adalah pembagian *issue*. Pada setiap *issue* yang diberikan kepada *developers* terdapat lampiran berupa *url* sebagai penanda bahwa halaman tersebut terdapat sebuah *issue* ataupun halaman yang akan ditambah fitur baru. Pada Gambar 3 merupakan contoh lampiran *url* pada *issue* yang diberikan.



Gambar 3. Contoh *issue* *ActiveCollab* proyek PPSDM

Setelah mengakses *url* tersebut, *developers* dapat memulai mengerjakan *issue* yang diberikan, langkah pertama yang harus dilakukan adalah mengetahui nama file *views* yang digunakan pada halaman tersebut. Dengan menggunakan *packages* *Laravel Debugbar* informasi tersebut dapat dengan mudah didapatkan yaitu pada *tab views*, *tab* tersebut akan menampilkan informasi semua nama-nama file *views* serta jumlah yang di-*load* pada halaman yang sedang diakses seperti pada Gambar 4. Sedangkan informasi mengenai nama *route* dan *controller* yang digunakan ada pada *tab Route*.



Gambar 4. Contoh *tab views* package *Laravel Debugbar*

- Melakukan *print out* sebuah *variable* dalam pengecekan output baris kode. Dalam menulis sebuah baris kode perlu dipastikan bahwa kode menghasilkan *output* yang benar sesuai ekspektasi yang diharapkan. Salah satu cara pengecekan baris kode menghasilkan *output* yang benar adalah dengan melakukan *print out* sebuah *variable* dari baris kode yang ditulis tersebut secara langsung. *Framework Laravel* menyediakan *function dump and die* atau *dd()* yang berfungsi untuk menghasilkan *output* dari *variable* namun memberhentikan eksekusi kode selanjutnya. Pada *packages Laravel Debugbar* juga dapat melakukan *print variable* tanpa memberhentikan eksekusi kode yang berjalan yaitu dengan memasukkan *variable* tersebut ke dalam *façade class* dari *Laravel Debugbar* atau menggunakan metode *non-static* dengan perintah pada baris ke-4 dan baris ke-9 seperti Gambar 5.

```

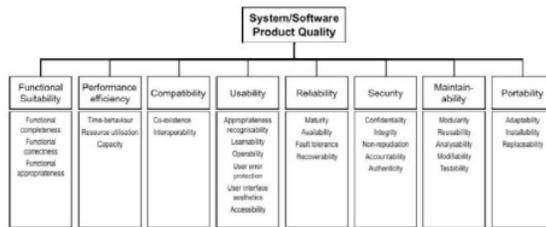
1  $user = auth()->user();
2
3  //Façade Laravel Debugbar/static method
4  Debugbar::nama_properti_atau_method('nama_variabel');
5  //contoh
6  Debugbar::info($user);
7
8  //non-static method
9  app('debugbar')->nama_properti_atau_method('nama_variabel');
10 //contoh
11 app('debugbar')->info($user);
12

```

Gambar 5. Notasi *print out* sebuah *variable*

D. Model ISO/IEC 25010

ISO/IEC merupakan standar dunia internasional dalam mengevaluasi kualitas sebuah perangkat lunak yang dikembangkan oleh *International Organization for Standardization and International Electronical Commission* disingkat dengan ISO/IEC. Adapun versi yang digunakan pada makalah ini adalah versi ISO/IEC 25010 edisi pertama tahun 2011, pengembangan dari versi ISO/IEC 9126 tahun 2001 yang telah direvisi secara teknis, menjadi standar internasional terbaru dan relevan untuk menguji software dan sistem komputer yang dikembangkan. ISO/IEC 25010 memiliki 8 karakteristik yang menjadi tolak ukur dalam mengevaluasi kualitas perangkat lunak berdasarkan dimensi *product quality* yaitu *functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability*, dan *portability* seperti pada Gambar 6 [5].



Gambar 6. Model product quality pada ISO/IEC 25010

E. Analisis Pemanfaatan Packages Laravel Debugbar dengan ISO/IEC 25010

Menurut Peter Salim dan Yenni Salim (2002), analisis merupakan proses pemecahan masalah yang diawali dengan membuat sebuah hipotesis kemudian melakukan pengamatan dan percobaan untuk membuktikan kebenaran dari hipotesis tersebut[7]. Adapun langkah yang digunakan untuk membuktikan hipotesis adalah dengan melakukan evaluasi pemanfaatan *packages Laravel Debugbar* pada saat pengerjaan *issue* proyek PPSDM dengan parameter pada Model ISO/IEC 25010. Parameter yang digunakan sebagai pada makalah ini yaitu 5 parameter, yaitu 3 subkarakteristik dari karakteristik *function suitability* ada *functional completeness, functional correctness, functional appropriateness*. Sedangkan karakteristik *performance efficiency* terdapat 2 subkarakteristik yaitu *time behavior*, dan *resource utilization* (Gambar 6). Terkait penjelasan lengkap dari 5 subkarakteristik dan 2 karakteristik tersebut sebagai berikut.

Function suitability adalah kemampuan sebuah sistem dalam menyediakan fungsi yang dapat memenuhi kebutuhan tertentu untuk digunakan di kondisi tertentu. Subkarakteristik yang digunakan pada makalah sebagai evaluasi *packages Laravel Debugbar* ada 3 sebagai berikut.

- **Functional Completeness**, kondisi dimana suatu sistem dapat mencakup semua *issue* yang ditentukan dan memenuhi tujuan dari pengguna.
- **Functional Correctness**, kondisi dimana suatu sistem menyediakan hasil yang benar dengan tingkat presisi yang diperlukan.
- **Functional Appropriateness**, kondisi dimana suatu sistem dapat memfasilitasi pencapaian *issue* dan tujuan tertentu.

Performance efficiency adalah kemampuan sebuah sistem dengan kinerja relatif terhadap jumlah sumber daya yang digunakan dalam kondisi yang ditentukan. Subkarakteristik yang digunakan pada makalah sebagai evaluasi *packages Laravel Debugbar* ada 2 sebagai berikut..

- **Time Behavior**, tingkat dimana respon dan waktu proses suatu produk atau sistem dalam memenuhi kebutuhan tertentu.
- **Resource Utilization**, tingkat dimana jumlah dan jenis sumber daya yang digunakan oleh suatu produk atau sistem ketika menjalankan fungsinya memenuhi persyaratan.

Adapun pemilihan 2 dari 8 karakteristik dari dimensi *product quality* pada ISO/IEC 25010 yaitu *function suitability* dan *performance efficiency*, karena 2 karakteristik tersebut paling cocok untuk dapat mengevaluasi pemanfaatan *packages Laravel Debugbar* yaitu pencarian nama file *views, route* dan *controller* serta melakukan *print out* sebuah *variable*.

III. METODOLOGI

Penyusunan makalah ini dibuat dengan metodologi yaitu, 1) evaluasi pemanfaatan *packages Laravel Debugbar* dalam pencarian nama file *views, route* dan *controller* serta penggunaan *packages* ini dalam melakukan *print out* sebuah *variable*. Parameter yang digunakan berdasarkan masing-masing subkarakteristik *function suitability* dan *performance efficiency* ISO/IEC 25010. 2) penarikan kesimpulan setelah evaluasi.

Skala penilaian dari evaluasi didapatkan dari asumsi peneliti berdasarkan fakta dilapangan penggunaan *packages Laravel Debugbar* dalam pengerjaan proyek PPSDM. Adapun skala penilaian adalah *Sangat Baik (efektif), Baik dan Kurang Baik (kurang efektif)*.

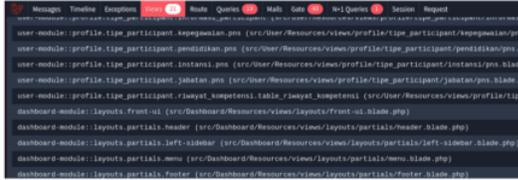
Adapun evaluasi yang sudah dilakukan untuk menilai *packages Laravel Debugbar* sebagai berikut.

A. Evaluasi packages Laravel Debugbar

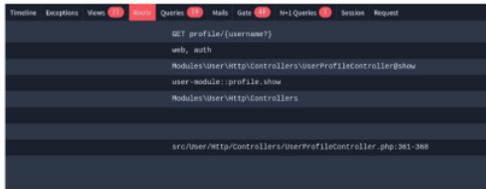
1) *Functional Completeness*, kemampuan *packages* untuk menyediakan kebutuhan tertentu dan memenuhi tujuan yang diinginkan dari *developers*.

a) Pencarian nama file *views* dan *controller*

Evaluasi berdasarkan *functional completeness, packages Laravel Debugbar* dapat menyediakan informasi yang dibutuhkan *developers* terkait nama file *views, route* dan *controller* yang digunakan (Gambar 7 dan Gambar 8). Hal ini mempermudah *developers* karena tidak perlu mencari secara manual informasi tersebut, belum lagi ada halaman yang menggunakan *blade extend* (penggunaan kode program lain pada kode program lainnya) pada Gambar 8.



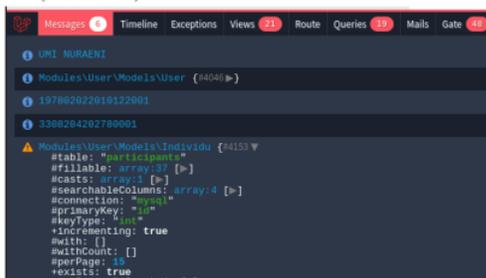
Gambar 7 Tampilan nama-nama file views yang mengimplementasikan blade extend pada halamannya



Gambar 8 Tampilan tab Route

b) Melakukan print out sebuah variable

Evaluasi berdasarkan *functional completeness*, packages *Laravel Debugbar* dapat melakukan print out sebuah variable dengan output pada tab messages, yang digunakan untuk melakukan pengecekan output sebuah variable dalam baris kode terdahulu atau menguji variable yang ingin diimplementasikan (Gambar 9).



Gambar 9 Tampilan print out variable

2) *Functional Correctness*, kemampuan packages untuk menyediakan hasil yang benar sesuai dengan tingkat presisi yang diperlukan.

a) Pencarian nama file views, route dan controller

Evaluasi berdasarkan *functional correctness*, packages *Laravel Debugbar* menyediakan hasil data yang benar mengenai letak baris kode dari method controller yang digunakan pada halaman tersebut (Gambar 10).

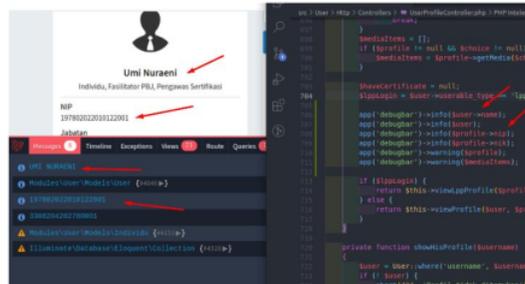


Gambar 10 Letak baris method controller sesuai berdasarkan informasi dari tab Route

b) Melakukan print out sebuah variable

Evaluasi berdasarkan *functional correctness*, packages *Laravel Debugbar* menampilkan hasil data yang benar

terkait print out variable pada tab messages (Gambar 11).

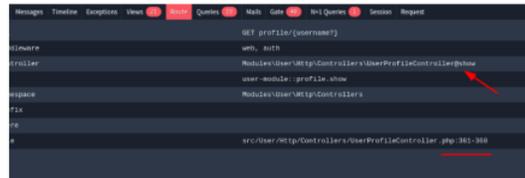


Gambar 11 Data valid pada saat print out variable

3) *Functional Appropriateness*, kemampuan packages untuk memfasilitasi pencapaian issue dan tujuan tertentu.

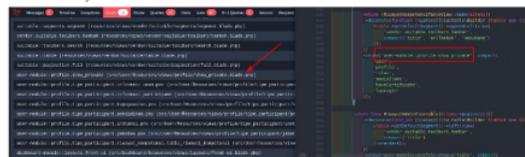
a) Pencarian nama file views, route dan controller

Evaluasi berdasarkan *functional appropriateness*, packages *Laravel Debugbar* selain menyediakan nama route dan controller yang digunakan pada halaman yang diakses, packages ini juga memfasilitasi developers berupa informasi berupa letak baris dari method controller yang digunakan pada halaman tersebut (Gambar 12). Selain itu juga packages ini menampilkan informasi spesifik seperti nama uri, jenis middleware, nama prefix, dan nama file.



Gambar 12 Informasi tambahan pada tab Route berupa letak baris kode method controller

Sedangkan pencarian nama file views berdasarkan *functional appropriateness*, *ViewsCollector* hanya menampilkan informasi nama file views yang di-load, tidak dapat memfasilitasi atau memberikan tanda satu nama file views yang menjadi base html yang digunakan pada setiap halaman yang diakses (Gambar 13).

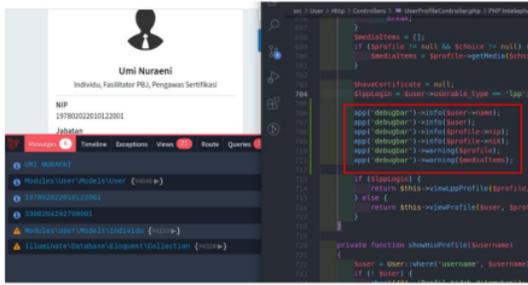


Gambar 13 Tidak memfasilitasi base html yang digunakan di halaman tersebut

b) Melakukan print out sebuah variable

Evaluasi berdasarkan *functional appropriateness*, packages *Laravel Debugbar* dapat melakukan multiple print out beberapa variable. Hal ini tentu bermanfaat agar developers dapat menguji beberapa variable

sekalius yang telah dibuat maupun melakukan pengecekan *output variable* baris kode terdahulu (Gambar 14).



Gambar 14 Multiple print out beberapa variables

4) *Time Behavior*, kemampuan *packages* dalam menampilkan respon dan waktu proses suatu produk atau sistem dalam memenuhi kebutuhan tertentu.

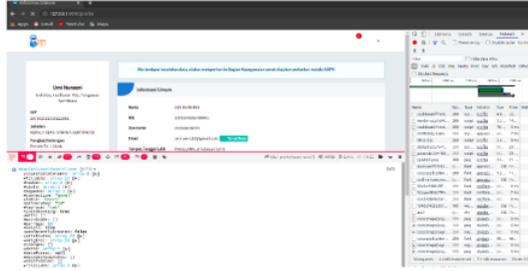
a) Pencarian nama file *views*, *route* dan *controller*

Evaluasi berdasarkan *time behavior*, *packages Laravel Debugbar* memang akan ada tambahan waktu respon dan waktu proses karena melakukan *request* data ke semua *collector*, namun hal ini lebih baik dari pada melakukan pencarian nama file *views*, *route* dan *controller* secara manual. Berikut ini perbandingan langkah pencarian nama file *views*, *route* dan *controller* menggunakan *packages Laravel Debugbar* dan manual atau tanpa *packages Laravel Debugbar*.

- *Packages Laravel Debugbar*: 1) buka *tab views* dan *tab route*. 2) akses halaman berdasarkan informasi dari 2 *tab* tersebut.
- Tanpa *packages Laravel Debugbar*: 1) buka file 'web.php'. 2) cari *route* yang menggunakan *uri* halaman yang sedang diakses. 3) buka *method controller* dari *route* yang sudah ditemukan. 4) buka file *views* dengan nama sesuai dengan dalam perintah `return view('nama_file')` pada *method controller*.

b) Melakukan *print out* sebuah *variable*

Evaluasi berdasarkan *time behavior*, *packages Laravel Debugbar* dalam melakukan *print out* sebuah *variable* waktu respon yang didapatkan yaitu 3,54s (Gambar 15), berbeda jauh dibandingkan *print out* menggunakan *die dump* atau *dd()* yaitu 1,71s (Gambar 16). Perbedaan ini terjadi karena *packages Laravel Debugbar* melakukan *request* data ke semua *collector* hingga menampilkan panel, sedangkan *print out* menggunakan *dd()* akan memberhentikan semua proses eksekusi setelah baris perintah *print out* dijalankan.



Gambar 15 Tampilan waktu respon print out sebuah variable menggunakan packages Laravel Debugbar

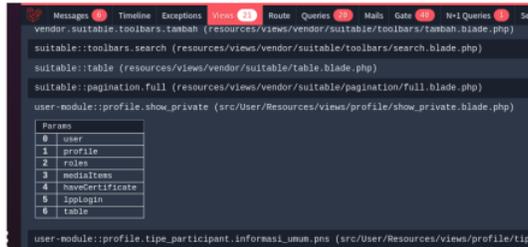


Gambar 16 Tampilan waktu respon print out sebuah variable menggunakan function dd()

5) *Resource Utilization*, kemampuan *packages Laravel Debugbar* dalam memanfaatkan jumlah dan jenis sumber daya tertentu ketika sedang menjalankan fungsinya memenuhi persyaratan.

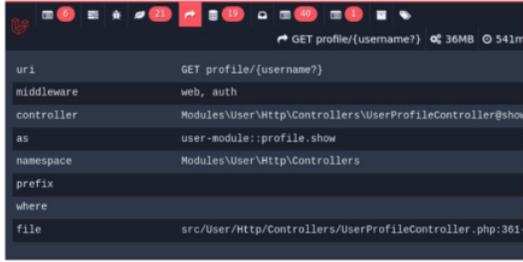
a) Pencarian nama file *views*, *route* dan *controller*

Evaluasi berdasarkan *resource utilization*, pada *packages Laravel Debugbar tab views* memberikan informasi tambahan berupa parameter yang dikirimkan pada masing-masing file *views* (Gambar 17).



Gambar 17 Parameter yang dikirimkan ke views

Sedangkan pada *tab Route*, *RouteCollector* bukan hanya menampilkan informasi nama *route* dan file *controller* yang digunakan, terdapat pembagian spesifik informasi seperti *uri*, jenis *middleware*, nama *prefix*, nama file, nama *method controller* serta letak barisnya (Gambar 18).



Gambar 18 Informasi spesifik pada tab Route

b) Melakukan *print out* sebuah *variable*

Evaluasi berdasarkan *resource utilization*, *packages Laravel Debugbar* tidak ada pemanfaatan informasi tertentu pada *tab messages print out* sebuah *variable*.

B. Hasil evaluasi *packages Laravel Debugbar*

1) Hasil evaluasi berdasarkan *functional completeness*

a) Pencarian nama file *views*, *route* dan *controller* menggunakan *packages Laravel Debugbar* dianggap **sangat baik** karena mampu memenuhi kebutuhan *developers* dengan menyediakan informasi terkait semua nama file *views*, nama *route* dan *controller* yang digunakan.

b) Melakukan *print out* sebuah *variable* dianggap **baik** karena dapat memenuhi kebutuhan *developers* dengan dapat melakukan perintah *print out* sebuah *variable*.

2) Hasil evaluasi berdasarkan *functional correctness*

a) Pencarian nama file *views*, *route* dan *controller* dianggap **sangat baik** karena memberikan hasil data yang valid dari nama file *views*, *route* dan *controller* yang dapat diakses.

b) Melakukan *print out* sebuah *variable* dianggap **baik** karena memberikan hasil data yang valid dari hasil *print out* *variable*.

3) Hasil evaluasi berdasarkan *functional appropriateness*

a) Pencarian nama *route* dan *controller* dianggap **sangat baik** karena memfasilitasi informasi tambahan dengan informasi spesifik letak baris kode dari *method controller*. Sedangkan *packages Laravel Debugbar* dianggap **kurang baik** untuk pencarian *views* karena tidak secara spesifik memberikan informasi terkait nama file *view* yang menjadi *base html* yang digunakan pada halaman yang sedang diakses.

b) Melakukan *print out* sebuah *variable* dianggap **baik** karena menyediakan *multiple print* beberapa *variable*. Sebenarnya ada juga *function* *Laravel* yang dapat melakukan hal yang sama yaitu *dump()* namun karena semua halaman di-load beserta *output print* justru membuat halaman tidak terstruktur berbeda dengan *packages Laravel Debugbar* yang menempatkan *output print* di *tab messages* sehingga lebih terstruktur dan rapi.

4) Hasil evaluasi berdasarkan *time behavior*

a) Pencarian nama file *views*, *route* dan *controller* dianggap **baik** meskipun waktu respon dan waktu proses akan bertambah dengan adanya *request* data ke semua *collector*, namun penggunaan *packages Laravel Debugbar* dalam pencarian nama file *views*, *route* dan *controller* lebih baik daripada melakukannya secara manual.

b) Melakukan *print out* sebuah *variable* dianggap **kurang baik** karena waktu respon *packages Laravel Debugbar* lebih tinggi dibandingkan *print out* sebuah *variables* menggunakan *function dd()*. Hal ini dapat terjadi karena waktu proses *request* data semua *collector* hingga akhirnya dapat menampilkan panel.

5) Hasil evaluasi berdasarkan *resource utilization*

a) Pencarian nama file *views* dianggap **baik** karena *packages Laravel Debugbar* menyediakan informasi tambahan berupa parameter yang dikirimkan ke nama file *views* pada *tab messages*. Sedangkan pencarian nama *route* dan *controller* dianggap **sangat baik** karena menyediakan informasi spesifik seperti nama *uri*, jenis *middleware*, nama *prefix*, dan letak baris kode dari *method controller*.

b) Melakukan *print out* sebuah *variable* dianggap **kurang baik** karena tidak ada pengolahan sumber daya dari *packages Laravel Debugbar* untuk menyediakan informasi baru pada *tab messages*.

IV. KESIMPULAN DAN SARAN

Berikut ini tabel yang sudah dirangkum dari hasil evaluasi penggunaan *packages Laravel Debugbar* dalam pengerjaan *issue* proyek PPSDM dengan parameter subkarakteristik dari *function suitability* dan *performance efficiency* model ISO/IEC 25010.

Parameter Evaluasi	Pemanfaatan Packages Laravel Debugbar dalam proyek PPSDM		
	Pencarian nama file views	Pencarian route & controller	Melakukan print out sebuah variable
Functional Completeness	Sangat Baik	Sangat Baik	Baik
Functional Correctness	Sangat Baik	Baik	Baik
Functional Appropriateness	Kurang baik	Sangat Baik	Baik
Time Behavior	Baik	Baik	Kurang baik
Resource Utilization	Baik	Baik	Kurang baik

Tabel 1 Hasil evaluasi pemanfaatan *packages Laravel Debugbar* dalam pengerjaan *issue* proyek PPSDM

A. Kesimpulan berdasarkan hipotesis

Adapun hasil kesimpulan yang diperoleh serta korelasinya dengan hipotesis adalah sebagai berikut.

1) Hipotesis pertama penelitian dapat diterima, subkarakteristik dari *function suitability* dan *performance efficiency* model ISO/IEC 25010 dapat menjadi tolak ukur evaluasi manfaat dari penggunaan *packages Laravel Debugbar* dalam pengerjaan proyek PPSDM. Hal ini dikarenakan semua evaluasi memiliki hasil mutlak dengan alasan yang jelas serta rasional.

2) Hipotesis kedua penelitian tidak sepenuhnya diterima, karena setelah dilakukan evaluasi terdapat parameter yang menunjukkan hasil kurang baik atau efektif, maka dengan itu berikut kesimpulan berdasarkan hipotesis kedua.

a) Pencarian nama file *views* menggunakan *packages Laravel Debugbar* berdasarkan hasil evaluasi 5 parameter, peneliti berasumsi bahwa *packages* ini dapat mempermudah

developers namun kurang efektif. *Packages* ini menyediakan informasi yang benar dan valid semua nama file *views* termasuk parameter yang dikirimkan dengan waktu proses menyesuaikan *collector* lain. Namun kekurangan utama pada *tab messages* yaitu tidak ada tanda atau *highlight* nama file *views* yang menjadi *base html*, hal ini sama dengan melakukan pencarian nama file *views* secara manual.

b) Pencarian nama *route* dan *controller* menggunakan *packages Laravel Debugbar* berdasarkan hasil evaluasi 5 parameter, peneliti berasumsi bahwa *packages* ini dapat mempermudah *developers* dan sangat efektif. *Packages* ini menyediakan informasi yang benar dan valid secara spesifik nama *route* dan *controller* yang digunakan dengan tambahan informasi letak baris *method controller* dan informasi pendukung lain yang digunakan dengan waktu proses yang menyesuaikan *collector* lain.

c) Melakukan *print out* sebuah *variable* dengan menggunakan *packages Laravel Debugbar* berdasarkan hasil evaluasi 5 parameter, peneliti berasumsi bahwa *packages* ini mempermudah *developers* namun kurang efektif. *Packages* ini dapat melakukan perintah *print out* sebuah *variable* dengan data yang valid dan benar, dengan *output* rapi dan terstruktur pada *tab messages* serta dapat juga *print out* beberapa *variable*. Namun kekurangan utama penggunaan *packages* ini dalam melakukan perintah tersebut adalah waktu proses yang lama karena menyesuaikan *collector* lain hanya untuk melakukan pengecekan *output* sebuah *variable*, hal ini lebih efektif jika menggunakan *function* umum *Laravel* yaitu *dump and die* atau *dd()*.

B. Saran

Untuk mendapatkan hasil penelitian yang lebih baik, berikut ini saran yang harus dilakukan untuk penelitian selanjutnya.

1) Memiliki skala penilaian yang jelas dan dapat diukur bukan hanya sekadar asumsi sehingga menghasilkan kesimpulan yang valid.

REFERENSI

- [1] K. Wilkie, "Rise or Resist: Exploring Senior Secondary Students' Reactions to Challenging Mathematics Tasks Incorporating Multiple Strategies," *Eurasia J. Math. Sci. Technol. Educ.*, vol. 12, pp. 2061–2083, 2016.
- [2] C.-Y. Hung, J. C.-Y. Sun, and P.-T. Yu, "The benefits of a challenge: student motivation and flow experience in tablet-PC-game-based learning," *Interact. Learn. Environ.*, vol. 23, pp. 172–190, 2015.
- [3] Symfony, "How to Create a custom Data Collector (Symfony Docs)," 2020. https://symfony.com/doc/current/profiler/data_collector.html (accessed Jun. 01, 2021).
- [4] V. Rivai, *Manajemen Sumber Daya Manusia untuk Perusahaan: Dari Teori Ke Praktek*. Jakarta: PT Rajagrafindo Persada, 2009.
- [5] ISO/IEC, "ISO / IEC 25010 : 2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models," 2011.
- [6] B. vd. Heuvel, "barryvdh/laravel-debugbar:Laravel Debugbar - Github," 2021. <https://github.com/barryvdh/laravel-debugbar> (accessed May 08, 2021).
- [7] P. Salim and Y. Salim, *Kamus bahasa Indonesia kontemporer*. Jakarta: Modern English Press, 2002.

Analisis Pemanfaatan Laravel Debugbar Dalam Mempermudah Penyelesaian Issue Pada Aplikasi Pusat Pengembangan Sumber Daya Manusia (PPSDM)

ORIGINALITY REPORT

6%

SIMILARITY INDEX

6%

INTERNET SOURCES

4%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1	eprints.uny.ac.id Internet Source	1%
2	Mahmood H. Hussein, Siew Hock Ow, Loh Sau Cheong, Meow-Keong Thong. "A Digital Game-Based Learning Method to Improve Students' Critical Thinking Skills in Elementary Science", IEEE Access, 2019 Publication	1%
3	www.lkpp.go.id Internet Source	1%
4	id.scribd.com Internet Source	1%
5	link.springer.com Internet Source	1%
6	Www.ejmste.com Internet Source	<1%
7	www.coursehero.com Internet Source	<1%

8

zombiedoc.com

Internet Source

<1 %

9

journal.student.uny.ac.id

Internet Source

<1 %

10

Maman Suryaman, Yuliarman Saragih, Agatha Elisabet S, Chia Chee Ming, Nguyen Thi Nhu Van, Abdul Aziz Hafeel. "Tailoring The Certified of Electric Engineering for Power Plant: Development Training Model by Using 3D VR Simulator Modification for Professional Engineer Case Study of Universitas Singaperbangsa Karawang", 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), 2020

Publication

<1 %

11

www.mdpi.com

Internet Source

<1 %

12

repository.uinsu.ac.id

Internet Source

<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On