

Penerapan Model View Controller dan Object Relational Mapping Pada Pengembangan Sistem Informasi Keanekaragaman Hayati di Taman Nasional Kutai Bontang

*1st Muhammad Romadinu Al Mukabir
Teknik Informatika
Universitas Mulawarman
Samarinda, Indonesia
rplkabir@gmail.com

2nd Edy Budiman
Teknik Informatika
Universitas Mulawarman
Samarinda, Indonesia
edybudiman.unmul@gmail.com

3rd Havaluddin
Teknik Informatika
Universitas Mulawarman
Samarinda, Indonesia
havaluddin@gmail.com

ABSTRAK- Pengembangan sistem informasi keanekaragaman hayati Taman Nasional Kutai Bontang merupakan bagian dari kajian study Bioinformatics. Pengembangan aplikasi yang berskala enterprise seperti sistem tersebut membutuhkan berbagai pendekatan model dan metode. Penelitian ini bertujuan untuk mengembangkan sistem informasi keanekaragaman hayati Taman Nasional Kutai Bontang sistem manajemen database dalam mendukung sistem pemantauan spesies, menjadi media penyimpangan data informasi secara digital. Metode pengembangan menggunakan pendekatan berorientasi objek dengan pola desain Model-view Controller dan tools Object Relational Mapping untuk menjembatani mengatasi masalah Impedance Mismatch ketidaksesuaian antara model relasional dengan model objek. Hasil penelitian merupakan produk perangkat lunak sistem informasi keanekaragaman hayati Taman Nasional Bontang sebagai Produk teknologi tepat guna dibidang Bioinformatics.

Kata Kunci : LARAVEL, framework, Object Relational Mapping, Object-relational Impedance Mismatch, Entity Relationship diagram.

I. PENDAHULUAN

Taman Nasional Kutai (TNK) merupakan salah satu taman nasional yang penting karena memiliki berbagai tipe vegetasi sebagai perwakilan ekosistem hutan hujan dataran rendah di Kalimantan Timur. seperti vegetasi hutan pantai/mangrove, hutan rawa air tawar, hutan kerangas, hutan genangan dataran rendah, hutan ulin/meranti/kapur dan hutan Dipterocarpaceae campuran bahkan merupakan perwakilan formasi ulin yang paling luas di Indonesia. Taman Nasional Kutai dikelilingi oleh berbagai bentuk pemanfaatan lahan diantaranya industri besar tambang, hutan tanaman dan hak perusahaan hutan alam (HPH), sehingga mengundang masyarakat pendatang yang menimbulkan tekanan terhadap hutan [1].

Sistem manajemen basis data yang banyak digunakan sekarang ini adalah basis data relasional. Sedangkan untuk pengembangan aplikasi yang berskala enterprise seperti BIS menerapkan konsep berorientasi objek. Dengan demikian, terdapat ketidaksesuaian antara basisdata relasional dengan pengembangan aplikasi yang menggunakan konsep berorientasi objek (*impedance mismatch*) [2]. Dalam mengatasi perbedaan tersebut, maka dibutuhkan teknologi

Object Relational Mapping untuk menjembatani perbedaan diatas [3], [4].

Teknologi *Object Relational Mapping* digunakan dalam membangun pemetaan relasi dalam pengembangan sistem informasi keanekaragaman hayati di Taman Nasional Kutai, Bontang agar data yang diolah menjadi lebih baik dan tidak terjadi ketidaksesuaian tersebut. Selain teknologi *Object Relational Mapping*, diterapkan pula pola desain *Model View Controller* sebagai media menampilkan hasil dari pemetaan relasi dari pengembangan tersebut.

II. TINJAUAN PUSTAKA

A. MODEL VIEW CONTROLLER

Model View Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) [5] untuk meng-enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah user interface. Definisi teknis dari arsitektur MVC menurut [6] dibagi menjadi tiga lapisan.

Model View Controller (MVC) mempunyai tiga bagian:

1. Model
Model merepresentasikan struktur data pada aplikasi web. Pada umumnya, model berisi fungsi-fungsi yang berhubungan dengan database, seperti pengambilan data, *update data*, *insert data*, *delete data*, dan lain sebagainya.
2. View
View merupakan informasi yang direpresentasikan kepada *user*. *View* biasanya berupa halaman web, dimana *client* dapat melihat informasi yang ditampilkan.
3. Controller
Controller memberikan pelayanan yang menjembatani *Model* dan *View*. *Controller* berisi fungsi-fungsi yang dapat membantu menjembatani *Model* dan *view*. *Controller* merupakan level tertinggi dari keseluruhan level MVC.

B. OBJECT RELATIONAL MAPPING

Object Relational Mapping, merupakan sebuah teknik pemrograman yang menghubungkan SQL dengan konsep pemrograman berorientasi objek. ORM mampu menjembatani perbedaan tipe data pada konsep pemrograman yang berorientasi objek dengan konsep RDBMS [4].

Object Relational Mapping digunakan untuk memetakan terhadap table-table pada basisdata relasional menggunakan suatu *class* pada aplikasi berbasis objek [7], [8].

C. ENTITY RELATIONSHIP DIAGRAM

Entity Relationship Diagram (ERD) menurut [9] adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi.

Lebih lanjut dalam kajian [10] Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan.

D. MODEL PROSES PERANGKAT LUNAK

Model proses pengembangan perangkat lunak dalam penelitian ini menggunakan pendekatan proses *waterfall*, merupakan salahsatu metode perancangan sistem yang akan dibuat dengan tahapan: *requirement* (analisis kebutuhan), *planning system design* (perencanaan desain sistem), *coding & testing*, *implementation* (penerapan), *maintenance* (pemeliharaan sistem) [11].

E. PENELITIAN TERDAHULU

Penelitian tentang keanekaragaman hayati sebelumnya telah dibahas oleh penelitian [12] yang mengkaji tentang spesies dan morfologi tanaman bambu di Kalimantan. Kajian [13] membuats sistem informasi Atlas Kayu Berbasis Digital dan Penelitian [14] merancang bangun Aplikasi Sistem Informasi Tanaman Obat Tradisional Berbasis Android.

Penelitian lainnya seperti yang dibahas oleh [15] dan [16] tentang BIS: Medicinal Plants from Tropical Rainforest Borneo Based on Traditional Knowledge Ethnic of Dayak. Kajian [17] Database: Taxonomy of plants Nomenclature for Borneo BIS. Penelitian [18] Borneo biodiversity: Exploring endemic tree species and wood characteristics. [19] tentang Ethnobotany Database dan [20] Feature selection for Malaysian medicinal plant leaf shape identification and classification.

III. HASIL DAN PEMBAHASAN

A. MODEL VIEW CONTROLLER

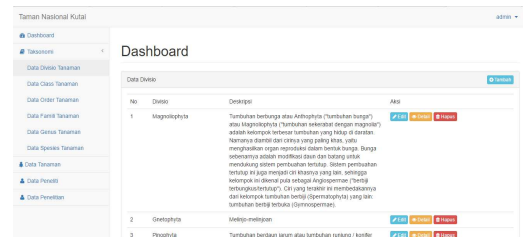
Model: merepresentasikan struktur data pada aplikasi web. Pada umumnya, model berisi fungsi-fungsi yang berhubungan dengan database, seperti pengambilan data, *update data*, *insert data*, *delete data*, dan lain sebagainya. Berikut merupakan contoh *script* untuk membuat model di *Laravel*:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
use App\Classes;
use App\Tumbuhan;
class divisio extends Model
- {
-     protected $fillable = [
-         'nama_divisio', 'deskripsi'
-     ];
-     public function kelas(){
-         return $this->hasMany(Classes::class);
-     }
-     public function Tumbuhan(){
-         return $this->hasMany(Tumbuhan::class);
-     }
- }
```

Gambar 1. Contoh script Model

Pada Gambar 1 merupakan contoh sederhana *script* yang menggambarkan dari struktur tabel *divisio*. Dimana *script* pada baris 10 sampai 12 merupakan *script* yang mendefinisikan *field-field* yang ada di tabel *divisio*.

View: merupakan informasi yang direpresentasikan kepada *user*. *View* biasanya berupa halaman web, dimana *client* dapat melihat informasi yang ditampilkan. Gambar 2 merupakan hasil dari *View*.



Gambar 2. Hasil dari View

Controller: memberikan pelayanan yang menjembatani *Model* dan *View*. *Controller* berisi fungsi-fungsi yang dapat membantu menjembatani *Model* dan *view*. *Controller* merupakan level tertinggi dari keseluruhan level MVC. Gambar 3 sebagai contoh dari *script Controller*:

```
public function index()
{
    DB::enableQueryLog();
    $tumbuhan = Tumbuhan::all();
    $time = DB::getQueryLog();
    $time = end($time);
    return view('tumbuhan.index', compact('tumbuhan', 'time'));
}
```

Gambar 3. Script Controller

Gambar 3 menjelaskan bahwa akan menampilkan data tumbuhan dan diurutkan berdasarkan *field created_at* dari yang terbesar ke terkecil dengan limit perhalaman sebanyak 3 data. *Variabel* tumbuhan akan di tampilkan di halaman *index* yang berada di folder *view* tumbuhan.

B. OBJECT RELATIONAL MAPPING

Relasi *One To One* merupakan relasi di database yang sangat mendasar. Dimana relasi ini menggambarkan hubungan antara tabel user dengan tabel peneliti. Dimana 1 data yang terdapat dalam tabel user hanya boleh dimiliki oleh 1 data di tabel admin.

Relasi *One To Many* merupakan relasi di database yang Dimana relasi ini menggambarkan hubungan antara tabel A dengan tabel B dengan ketentuan tabel B yang mengambil data dari tabel tersebut boleh memiliki lebih dari 1 data dari tabel A. Namun tabel A tidak bisa mengambil data dari tabel B. Contoh relasi yang terjadi adalah, tabel divisio berelasi dengan tabel class dimana tabel class memiliki *foreign key* dari tabel divisio sebagai penghubung antar tabel. Begitu juga dengan tabel-tabel seperti tabel class dengan tabel ordo, tabel ordo dengan tabel famili, tabel famili dengan tabel genus, dan tabel genus dengan tabel spesies. Tabel tumbuhan memiliki *foreign key* dari masing-masing tabel sebagai penghubung. Dimana dalam kasus ini, relasi dari keseluruhan relasi-relasi yang terbentuk antar tabel ini adalah *one to many*.

Pemetaan relasi tersebut berdasarkan konsep jenis. Dimana konsep ini menjelaskan Suatu takson yang merupakan suatu populasi terdiri atas individu-individu dengan sifat yang sama, sifat-sifat itu mirip dengan sifat semua keturunannya dan menempati suatu daerah distribusi tertentu pada saat bersamaan disebut suatu jenis (species). Beberapa jenis dengan persamaan sifat-sifat tertentu membentuk suatu takson yang menurut hirarki diberi kedudukan dan jenjang yang lebih tinggi yang disebut dengan istilah marga (genus). Setiap marga diberi nama seperti halnya setiap jenis. Demikian selanjutnya berturut-turut sejumlah marga dijadikan satu suku (familia), yang masing-masing diberi nama yang berbeda-beda pula. Beberapa suku dijadikan satu bangsa (ordo), beberapa bangsa menjadi kelas (classis) dan divisi (divisio).

Bahwa, setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan entitas B, dan begitu juga sebaliknya. Berikut gambar table yang berelasi.

Relasi *many to many* antara tabel tumbuhan dan peneliti. Dimana terdapat tabel bantu yaitu tabel peneliti_tumbuhan yang akan menampung hasil relasi dari tabel tumbuhan dan peneliti. Dengan kondisi satu tumbuhan dapat diteliti oleh banyak peneliti. Dan satu peneliti dapat meneliti banyak tumbuhan. Tabel bantu tersebut memiliki *field-field* seperti, id, peneliti_id, tumbuhan_id, judul_penelitian, hasil_penelitian dan kesimpulan.

Relasi *Has Many Throgh* adalah hubungan yang dapat memberikan jalan pintas yang mudah untuk mengakses relasi jarak jauh melalui relasi perantara. Relasi yang terjadi dalam kasus ini adalah antara tabel peneliti, peneliti_tumbuhan dan foto_penelitian.

Dalam memetakan relasi database di framework Laravel, maka di buatlah class dalam suatu model yang mendefinisikan masing-masing table. Didalam penelitian ini, terdapat 4 relasi yang terbentuk. Berikut relasi beserta *script* untuk memetakan relasi yang terjadi:

TABEL 1. SCRIPT UNTUK MEMETAKAN RELASI ORM

One To One	public function Admin(){ return \$this->hasOne(Admin::class); }
	public function User(){ return \$this->belongsTo(User::class); }
One To Many	public function Kelas(){ return \$this->hasMany(Kelas::class); }
	public function Divisio(){ return \$this->belongsTo(Divisio::class); }
Many To Many	public function peneliti(){ return \$this->belongsToMany('App\Peneliti', 'peneliti_tumbuhans', 'peneliti_id', 'tumbuhan_id')->withTimestamps(); }
	public function tumbuhans(){ return \$this->belongsToMany('App\Tumbuhan', 'peneliti_tumbuhans', 'peneliti_id', 'tumbuhan_id')->withTimestamps(); }
Has Many Through	public function Khasiat(){ return \$this->hasManyThrough('App\TumbuhanKhasiat', 'App\Tumbuhan', 'spesies_id', 'tumbuhan_id'); }

C. PENGUJIAN

Pengujian Sistem ini dilakukan untuk mendapatkan hasil dari waktu eksekusi *query* yang dilakukan untuk menampilkan data-data berdasarkan relasi yang terbentuk. Pengujian dilakukan terhadap masing-masing relasi yaitu, *one to one*, *one to many*, *many to many*, dan *has many through*.

TABEL 2. SCRIPT YANG AKAN DIUJI

Relasi	Script Query Builder Join	Script dengan ORM
One To One	Peneliti::join('users','users.id','=', 'penelitis.user_id')->select('users.email', 'users.username', 'penelitis.nama', 'penelitis.alamat', 'penelitis.jenis_kelamin', 'penelitis.tanggal_lahir')->get();	Peneliti::all();
One To Many	Tumbuhan::join('divisios', 'divisios.id', '=', 'tumbuhans.divisio_id')->join('kelas', 'kelas.id', '=', 'tumbuhans.kelas_id')->join('familis', 'familis.id', '=', 'tumbuhans.famili_id')->join('ordos', 'ordos.id', '=', 'tumbuhans.ordo_id')->join('spesies', 'spesies.id', '=', 'tumbuhans.spesies_id')->join('genera', 'genera.id', '=', 'tumbuhans.genus_id')->select('tumbuhans.nama_daerah', 'tumbuhans.nama_sinonim', 'tumbuhans.nama_simplisia', 'tumbuhans.foto', 'divisios.nama_divisio', 'kelas.nama_kelas', 'ordos.nama_ordo', 'familis.nama_famili', 'genera.nama_genus', 'spesies.nama_spesies')->get();	Tumbuhan::all();
Many To Many	DB::table('peneliti_tumbuhans')->join('tumbuhans', 'peneliti_tumbuhans.tumbuhan_id', '=', 'tumbuhans.id')->join('penelitis', 'peneliti_tumbuhans.peneliti_id', '=', 'penelitis.id')->select('penelitis.nama', 'tumbuhan	Peneliti::all();

	s.nama_daerah') ->get();	
Has Many Through	Spesies::join('tumbuhans','spesies.id','=', 'tumbuhans.spesies_id') >join('tumbuhan_khasiat','tumbuhans.id','=', 'tumbuhan_khasiat.tumbuhan_id') >select('tumbuhans.nama_daerah', 'tumbuhans.foto', 'tumbuhan_khasiat.khasiat', 'spesies.nama_spesies')) ->get();	Spesies::find(\$request->spesies_id);

Pengujian ini menggunakan *package Laravel Debugger*. Kemudian, akan dilakukan 2 jenis pengujian yaitu, pengujian system secara *offline* dan pengujian secara *online*. Versi *Laravel* yang digunakan adalah versi 5.5. Spesifikasi laptop yang digunakan untuk pengujian *offline* menggunakan laptop penulis sendiri. Spesifikasi dari laptop yang digunakan adalah sebagai berikut:

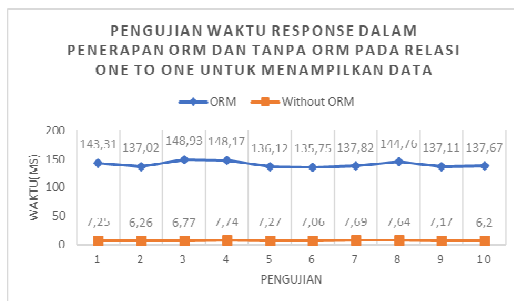
- RAM 4 GB
- Hardisk 500 GB
- Processor Intel i5
- VGA 128 MB
- Sistem operasi: Windows 10

Hasil Pengujian yang dilakukan secara *offline* adalah sebagai berikut:

a. Pengujian Relasi *One To One* offline

TABEL 3. PENGUJIAN RELASI ONE TO ONE OFFLINE

Pengujian	Query Builder Join	ORM
1	143.31 ms	7.25 ms
2	137.02 ms	6.26 ms
3	148.93 ms	6.77 ms
4	148.17 ms	7.74 ms
5	136.12 ms	7.27 ms
6	135.75 ms	7.06 ms
7	137.82 ms	7.69 ms
8	144.76 ms	7.64 ms
9	137.11 ms	7.17 ms
10	137.67 ms	6.2 ms
Rata-rata	140.666 ms	7.105 ms
Presentase (%)	95.19%	4.81%



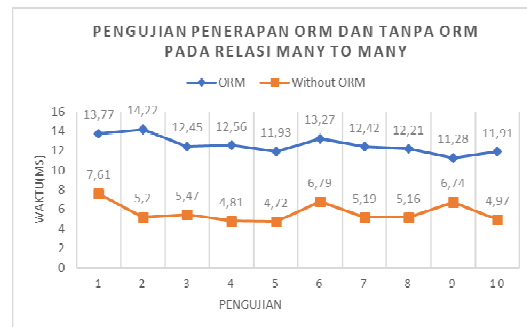
Gambar 4. Hasil pengujian relasi one to one offline

Pengujian diatas menjelaskan hasil rata-rata diatas dapat dilihat perbandingan antara penggunaan *Eloquent ORM* dan tanpa penggunaan *Eloquent ORM* bahwa telah terjadi perbedaan waktu sebesar 90,38% terhadap waktu eksekusi *query* untuk relasi *one to one* dengan jumlah tabel yang memiliki relasi sebanyak 2 tabel dan menampilkan semua data yang ada dengan ketentuan semua field ditampilkan. Didapatkan bahwa tanpa menggunakan ORM lebih cepat.

b. Pengujian Relasi *One To Many* offline

TABEL 4. PENGUJIAN RELASI ONE TO MANY OFFLINE

Pengujian	Query Builder Join	ORM
1	372.59 ms	68.02 ms
2	383.09 ms	15.83 ms
3	383.28 ms	66.14 ms
4	423.1 ms	14.51 ms
5	380.3 ms	14.84 ms
6	381.34 ms	5.94 ms
7	387.52 ms	6.1 ms
8	379.34 ms	6.76 ms
9	402.37 ms	6.58 ms
10	401.4 ms	6.3 ms
Rata-rata	389.433 ms	21.102 ms
Presentase%	94.86%	5.14%



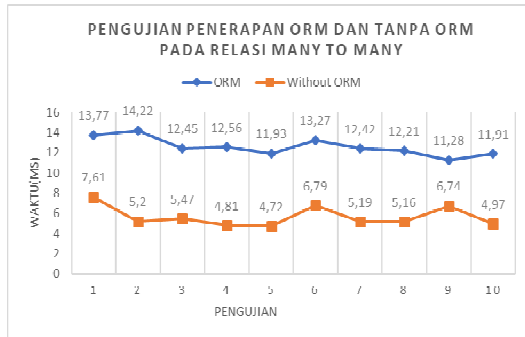
Gambar 5. Diagram hasil pengujian relasi one to many offline

Dari pengujian diatas dapat diketahui bahwa penerapan ORM memakan waktu lebih lama 89,72% dalam mengeksekusi relasi *one to many* dengan jumlah relasi sebanyak 7 tabel.

c. Pengujian Relasi *Many To Many* offline

TABEL 5. PENGUJIAN RELASI MANY TO MANY OFFLINE

Pengujian	Query Builder Join	ORM
1	13.77 ms	7.61 ms
2	14.22 ms	5.2 ms
3	12.45 ms	5.47 ms
4	12.56 ms	4.81 ms
5	11.93 ms	4.72 ms
6	13.27 ms	6.79 ms
7	12.42 ms	5.19 ms
8	12.21 ms	5.16 ms
9	11.28 ms	6.74 ms
10	11.91 ms	4.97 ms
Rata-rata	12.602 ms	5.666 ms
Presentase %	68.98%	31.02%



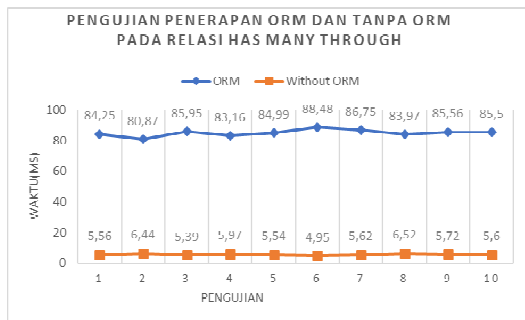
Gambar 6. Diagram hasil pengujian relasi many to many offline

Gambar 6 menjelaskan perbandingan antara penggunaan *Eloquent ORM* dan tanpa penggunaan *Eloquent ORM* bahwa telah terjadi peningkatan waktu sebesar 37,97% untuk relasi *many to many* dengan jumlah tabel yang berelasi sebanyak 3 tabel. penerapan ORM memakan waktu lebih lama.

d. Pengujian Relasi Has Many Through

TABEL 6. PENGUJIAN RELASI HAS MANY THROUGH OFFLINE

Pengujian	Query Builder Join	ORM
1	84.25 ms	5.56 ms
2	80.87 ms	6.44 ms
3	85.95 ms	5.39 ms
4	83.16 ms	5.97 ms
5	84.99 ms	5.54 ms
6	88.48 ms	4.95 ms
7	86.75 ms	5.62 ms
8	83.97 ms	6.52 ms
9	85.56 ms	5.72 ms
10	85.5 ms	5.6 ms
Rata-rata	84.948 ms	5.731 ms
presentase %	20.69%	1.40%



Gambar 7. Diagram hasil pengujian relasi has many through offline

hasil rata-rata diatas dapat dilihat perbandingan antara penggunaan *Eloquent ORM* dan tanpa penggunaan *Eloquent ORM* bahwa telah terjadi peningkatan eksekusi waktu sebesar 19,30% untuk relasi *many to many* dengan jumlah tabel yang berelasi sebanyak 3 tabel. Penerapan ORM memakan waktu lebih lama.

e. Pengujian Relasi One To One Online

TABEL 7. PENGUJIAN RELASI ONE TO ONE ONLINE

Pengujian	ORM	Query builder join
1	9710 ms	58.63 ms
2	6780 ms	250.75 ms
3	5680 ms	324.65 ms
4	7870 ms	40.65 ms
5	3550 ms	124.81 ms
6	4920 ms	119.59 ms
7	5.47 ms	140.65 ms
8	12190 ms	447 ms
9	2500 ms	574.75 ms
10	2750 ms	49.19 ms
Rata-rata	5595.547 ms	213.067 ms
presentase %	96.33%	3.67%

Perbandingan waktu penerapan ORM relasi one to one Dari pengujian diatas, dapat diketahui bahwa, dalam menampilkan keseluruhan data dan hanya sebagian field record, penerapan ORM justru lebih banyak memakan waktu sekitar 92.66%.

f. Pengujian Relasi One To Many Online

TABEL 8. PENGUJIAN RELASI ONE TO MANY ONLINE

Pengujian	ORM	Query builder join
1	22930 ms	115.6 ms
2	20160 ms	260.96 ms
3	22400 ms	328.45 ms
4	22400 ms	268.56 ms
5	21420 ms	252.57 ms
6	33460 ms	161.03 ms
7	42700 ms	367.03 ms
8	7700 ms	231.46 ms
9	29420 ms	411.13 ms
10	25960 ms	456.15 ms
Rata-rata	24855 ms	285.294 ms
presentase %	98.87%	1.13%

Dari pengujian diatas, dapat diketahui bahwa, dalam menampilkan keseluruhan data dan hanya sebagian *field record* dan relasi *one to many* terhadap konsep taksonomi tumbuhan, penerapan ORM justru lebih banyak memakan waktu sekitar 97,73%.

g. Pengujian Relasi Many To Many Online

TABEL 9. PENGUJIAN RELASI MANY TO MANY ONLINE

Pengujian	ORM	Query builder join
1	383.74 ms	257.24 ms
2	74.64 ms	57.46 ms
3	349.78 ms	79.4 ms
4	411.19 ms	259.61 ms
5	3630 ms	288.02 ms
6	799.72 ms	341.32 ms
7	228.31 ms	318.83 ms
8	429.79 ms	407.85 ms
9	146.54 ms	32.95 ms
10	432.81 ms	22.81 ms
Rata-rata	688.652 ms	206.549 ms
presentase %	76.93%	23.07%

Dari pengujian diatas didapatkan hasil pengujian yang menunjukkan bahwa dalam menampilkan data dengan relasi

many to many menggunakan penerapan ORM, memakan waktu lebih lama sekitar 53,85%.

Pengujian Relasi Has Many Through Online

TABEL 7. PENGUJIAN RELASI HAS MANY THROUGH ONLINE

Pengujian	ORM	Query builder join
1	6440 ms	329.33 ms
2	1490 ms	35.65 ms
3	317.42 ms	170.05 ms
4	439.23 ms	198.72 ms
5	3580 ms	189.52 ms
6	2920 ms	24.51 ms
7	3970 ms	285.56 ms
8	2650 ms	28.73 ms
9	1640 ms	24.83 ms
10	2640 ms	293.98 ms
Rata-rata	2608.665 ms	158.088 ms
presentase %	94.29%	5.71%

Dari pengujian diatas didapatkan hasil pengujian yang menunjukkan bahwa dalam menampilkan data dengan relasi has many through data memakan waktu lebih lama sekitar 88.57%.

D. PEMBAHASAN

Dalam implementasi penerapan model view controller dan object relational mapping ditemukan perbedaan waktu antara penerapan object relational mapping dan tanpa penerapan object relational mapping dengan menggunakan eloquent query builder. Perbedaan tersebut dapat dilihat dari pengujian yang dilakukan terhadap relasi-relasi yang terjadi didalam membangun sebuah sistem berbasis objek keanekaragaman hayati tumbuhan. Penerapan ORM dalam membangun sistem berbasis objek dapat digunakan ketika ingin menampilkan data dan relasi yang kompleks untuk menghindari kondisi mismatch impendensi. Namun, dengan menggunakan teknologi tersebut, sistem membutuhkan waktu dalam mengeksekusi query dalam menampilkan data dari tabel-tabel yang berelasi dalam jumlah yang banyak. Dikarenakan ORM akan mengeksekusi setiap relasi secara satu per satu dari table-tabel yang berelasi.

Contoh yang dapat diambil adalah dari relasi taksonomi tumbuhan. Dimana didalam relasi tersebut terdapat 7 tabel yang berelasi. Jadi, ORM akan mengeksekusi relasi antar table secara satu per satu agar menampilkan data sesuai dengan relasi tersebut. Berikut gambar yang menampilkan hasil dari eksekusi ORM.

IV. KESIMPULAN

Dari pengujian yang dilakukan dengan menggunakan Laravel Debugbar dapat disimpulkan bahwa Penerapan eloquent object relational mapping, dapat digunakan dalam membantu seorang pengembang sistem berbasis objek dalam mengatasi kondisi Mismatch Impedensi. Namun, dengan menggunakan penerapan tersebut, akan membuat sistem memakan waktu lebih lama dalam mengeksekusi dari masing-masing relasi. Untuk relasi one to one akan memakan waktu sekitar 95,19% secara offline dan 96,33% secara online. Untuk relasi one to many akan memakan waktu sekitar 94,86% secara offline dan 98,87% secara online. Untuk relasi many to many akan memakan waktu sekitar 68,97% secara offline dan 76,93% secara online. Untuk relasi

has many through akan memakan waktu sekitar 20.69% secara offline dan 94,29% secara online.

V. KESIMPULAN

Penulis menyadari dalam penyusunan skripsi ini masih jauh dari kata sempurna. untuk itu penulis ingin memberikan saran untuk perbaikan kedepannya:

1. Penulis berharap peneliti selanjutnya dapat meneliti seluruh relasi yang ada di Laravel, agar dapat lebih mengetahui segala bentuk relasi-relasi yang ada beserta perbandingan waktu antara penggunaan Eloquent object relational mapping dan tanpa menggunakan Eloquent object relational mapping.
2. Dalam membangun sistem ini, diharapkan peneliti atau pengembang, dapat menampilkan tidak hanya tanaman obat, tapi dapat menampilkan seluruh jenis tanaman yang ada di Kalimantan Timur.

DAFTAR PUSTAKA

- [1] Sawitri, R. and Adalina, Y., 2016. Kajian Usulan Zona Khusus Taman Nasional Kutai. Jurnal Penelitian Hutan dan Konservasi Alam, 13(2), pp.85-100.
- [2] Budiman, E., Jamil, M., Hairah, U. and Jati, H., 2017, August. Eloquent object relational mapping models for biodiversity information system. In Computer Applications and Information Processing Technology (CAIPT), 2017 4th International Conference on (pp. 1-5). IEEE.
- [3] Roebuck, K., 2011. Object-relational mapping (Orm): High-impact Strategies-What You Need to Know: Definitions, Adoptions. Impact, Benefits, Maturity, Vendors. Tebbo (June 2011).
- [4] Praditya, U.D., Saputra, R. and Noranita, B., 2014. Implementasi Object Relational Mapping pada Pengembangan E-Commerce Menggunakan Framework Yii. Journal of Informatics and Technology, 2(3), pp.113-124.
- [5] Bucanek, J., 2009. Model-view-controller pattern. Learn Objective-C for Java Developers, pp.353-402.
- [6] Simajuntak, P., 2016. Analisis Model View Controller (Mvc) Pada Bahasa Php. Journal Information System Development (ISD), 1(2).
- [7] Budiman, E., 2017. E-Government Data Profil dan Monografi Kelurahan Dadi Mulya Kecamatan Samarinda Ulu Kota Samarinda. STMÍK KHARISMA Makassar, 3 (1), 49-58.
- [8] Lw, C.W., Budiman, E. And Hairah, U., 2017. Manajemen Database Dan Sistem Informasi Keanekaragaman Hayati Tanaman Berkhasiat Obat Etnis Asli Kalimantan. Jurnal Rekayasa Teknologi Informasi, 1(1), Pp.73-80.
- [9] Haviluddin, A.T.H. and Rahmawati, D., 2016. Aplikasi Program PHP dan MySQL.
- [10] Asa, K., Budiman, E. and Pakpahan, H.S., 2017, September. Pengembangan Sistem Informasi Keanekaragaman Hayati Data Pohon Dan Endemik Pada Hutan Hujan Tropis Kalimantan. In Prosiding SAKTI (Seminar Ilmu Komputer dan Teknologi Informasi) (Vol. 2, No. 2, pp. 85-90).
- [11] Abdullah, A.A., Widiyans, J.A. and Budiman, E., 2017. Sistem Penjualan Pada Mitra Binaan Badak LNG. Jurnal Rekayasa Teknologi Informasi, 1(1), pp.8-17.
- [12] Nugraha, A.F., Budiman, E. and Hairah, U., 2017. Sistem Informasi Spesies Dan Morfologi Tanaman Bambu Di Kalimantan. Jurnal Rekayasa Teknologi Informasi, 1(1), pp.53-61.
- [13] Tupamahu, R., Budiman, E. and Pakpahan, H.S., 2017, September. Sistem Informasi Atlas Kayu Berbasis

- Digital. In *Prosiding SAKTI (Seminar Ilmu Komputer dan Teknologi Informasi)* (Vol. 2, No. 2, pp. 117-120).
- [14] Putri, N.K.S.C., Sudana, A.K.O. and Putra, I.K.G.D., 2014. Rancang Bangun Aplikasi Sistem Informasi Tanaman Obat Tradisional Berbasis Android. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*.
- [15] Budiman, E., Hairah, U., Tejawati, A., Darmawan, S. and Wahyuni, S., 2018. Biodiversity Information System of Medicinal Plants from Tropical Rainforest Borneo Based on Traditional Knowledge Ethnic of Dayak. *Advanced Science Letters*, 24(11), pp.8668-8673.
- [16] Dengen, N., Budiman, E., Widians, J.A., Wati, M., Hairah, U. and Ugiarto, M., 2018. Biodiversity Information System: Tropical Rainforest Borneo and Traditional Knowledge Ethnic of Dayak. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-9), pp.59-64.
- [17] Budiman, E. and Alam, S.N., 2017, November. Database: Taxonomy of plants Nomenclature for borneo biodiversity information system. In *Informatics and Computing (ICIC), 2017 Second International Conference on* (pp. 1-6). IEEE.
- [18] Hairah, U., Tejawati, A., Budiman, E. and Agus, F., 2017, October. Borneo biodiversity: Exploring endemic tree species and wood characteristics. In *Science in Information Technology (ICSITech), 2017 3rd International Conference on* (pp. 435-440). IEEE.
- [19] Haeruddin, H., Hairah, U., Budiman, E. and Johan, H., 2017. Ethnobotany Database: Exploring Diversity Medicinal Plants of Dayak Tribe Borneo. *Proceeding of the Electrical Engineering Computer Science and Informatics*, 4, pp.120-125.
- [20] Sainin, M.S. and Alfred, R., 2014, August. Feature selection for Malaysian medicinal plant leaf shape identification and classification. In *Computational Science and Technology (ICCST), 2014 International Conference on* (pp. 1-6). IEEE.