

Implementasi Laravel Eloquent ORM pada Pengembangan Aplikasi E-Commerce Balelabs Billing

by Naufal Hanan

Submission date: 27-Nov-2021 12:51AM (UTC+0700)

Submission ID: 1712706260

File name: paper_non_author.docx (513.46K)

Word count: 3088

Character count: 20724

Implementasi Laravel Eloquent ORM pada Pengembangan Aplikasi E-Commerce Balelabs Billing

Abstract—Pesatnya perkembangan teknologi informasi mendesak banyak perusahaan untuk beralih dari model bisnis konvensional ke model bisnis modern. *E-commerce* adalah salah satu model bisnis yang berkembang pesat seiring dengan gaya hidup masyarakat millennial. *E-commerce* memberikan berbagai manfaat baik bagi konsumen maupun perusahaan itu sendiri. Dengan *e-commerce*, proses bisnis yang terjadi antara perusahaan dan konsumen menjadi lebih efisien. Hal inilah yang mendorong PT Bale Lab Indonesia (Balelabs) untuk mengembangkan sebuah aplikasi *e-commerce*. Sebelumnya, Balelabs masih menerapkan model bisnis konvensional dalam memasarkan produk mereka sehingga proses bisnis yang terjadi antara perusahaan dan konsumen menjadi kurang efisien. Guna meningkatkan proses bisnis di perusahaan, Balelabs mengembangkan aplikasi *e-commerce berbasis web* dengan memanfaatkan Eloquent ORM Laravel dalam pengembangannya. Dari hasil pengembangan aplikasi ini, proses bisnis yang terjadi antara konsumen dan Balelabs menjadi lebih efektif. Hal ini dikarenakan konsumen balelabs tidak perlu lagi mendatangi kantor balelabs saat hendak melakukan transaksi.

Keywords—Teknologi Informasi, *E-commerce*, Laravel, Eloquent ORM, PT Bale Lab Indonesia

I. PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi yang cepat khususnya pada bidang teknologi informasi sangat berperan dalam berbagai aspek kehidupan. Segala kegiatan yang dilakukan sehari-hari pasti tidak lepas dari sentuhan teknologi. Pengembangan teknologi informasi sendiri memiliki berbagai tujuan, antara lain: untuk memecahkan suatu masalah, mengefisienkan pekerjaan, dan menciptakan hal baru. PT Bale Lab Indonesia (Balelabs) adalah salah satu perusahaan di Yogyakarta yang bergerak dalam bidang teknologi informasi. Balelabs memberikan layanan pada pengembangan perangkat lunak dan *IT Consultant* dengan berfokus pada jasa pembuatan, pengembangan dan optimasi *website* serta aplikasi berbasis *mobile*.

PT Bale Lab Indonesia telah berdiri sejak awal tahun 2020 di Yogyakarta. Perusahaan ini didirikan oleh empat orang, yakni: Bapak Wahyu Tri Utomo, Bapak Amarudin Murtadlo, Bapak Verry Bustomi Taslim dan Ibu Zusni Adisya. Balelabs menerapkan sistem kerja secara *remote/work from home* untuk divisi pengembangannya. Sampai saat ini, Balelabs telah memiliki lebih dari 20 karyawan yang tersebar di berbagai daerah di Indonesia. Balelabs juga telah bekerjasama dengan banyak mitra untuk mengembangkan berbagai proyek.

Selama ini, Balelabs masih menggunakan metode transaksi konvensional untuk setiap transaksi yang didapat. Adapun yang dimaksud metode konvensional adalah metode transaksi yang mengharuskan perusahaan dan konsumen untuk berkomunikasi secara langsung pada transaksi yang akan dilakukan. Metode konvensional memiliki beberapa kelemahan, antara lain: kurangnya efisiensi saat pemesanan aplikasi, pendataan transaksi yang masih dilakukan secara manual dan keterbatasan waktu oleh konsumen ketika ingin membeli aplikasi. Jika hal ini tetap dibiarkan, proses bisnis akan menjadi lebih lambat dan perusahaan menjadi tidak dapat berkembang. Selain itu, dari hasil wawancara dengan *Product Owner* Balelabs, dibutuhkan sebuah sistem yang dapat menjembatani transaksi antara konsumen dan Balelabs agar proses bisnis menjadi lebih efisien. Beberapa perusahaan sejenis seperti Domainsia, IDwebhost, rumahweb memiliki sebuah sistem yang belum dimiliki. Perusahaan – perusahaan tersebut telah mengembangkan sistem yang memudahkan konsumen mereka melakukan transaksi pada platform yang dikembangkan. Dengan sistem ini, proses bisnis dan jangkauan pasar menjadi lebih luas.

Sebagai upaya dalam mengembangkan proses bisnis, dibuatlah sebuah aplikasi *e-commerce* berbasis web bernama Balelabs Billing. *E-commerce* dipilih karena telah digunakan di beberapa perusahaan sejenis dan terbukti mampu untuk meningkatkan proses bisnis perusahaan. Aplikasi ini dikembangkan sebagai media pembelian dan manajemen paket aplikasi untuk konsumen balelabs. Dengan Balelabs Billing, konsumen tidak perlu mendatangi kantor Balelabs saat hendak memesan sebuah aplikasi. Hal ini memberikan efisiensi transaksi antara konsumen dan perusahaan sehingga proses bisnis menjadi lebih cepat dan optimal. Selain itu, Balelabs juga membutuhkan sebuah sistem dengan masa pengembangan yang cepat, dan pemeliharaan yang mudah. Sebagai salah satu bentuk upaya untuk memenuhi kebutuhan ini, pengembangan sistem memanfaatkan Laravel Eloquent ORM sebagai model *query* yang digunakan. Dengan memanfaatkan Eloquent, pengembangan sistem menjadi lebih cepat dan efisien karena baris kode yang digunakan lebih sedikit dan sederhana. Hal ini juga membuat perawatan sistem menjadi lebih efektif karena kode lebih mudah dibaca sehingga developer lain dapat dengan mudah memahami alur sistem.

II. DASAR TEORI

A. *E-Commerce*

E-Commerce adalah model bisnis yang berfokus pada skala ekspansi yang cepat. *e-commerce* meningkatkan

efektivitas operasi dan prospek perusahaan terhadap pertumbuhan produktivitas. Beberapa manfaat yang didapatkan dari model bisnis *e-commerce*, antara lain: memperluas jangkauan pasar, menggunakan teknik modern dengan media elektronik alih-alih teknik tradisional dengan kertas, dan pengurangan jumlah gudang atau biaya persediaan [1]. Berbagai perusahaan telah menerapkan model bisnis ini untuk meningkatkan produktivitas perusahaan.

B. Laravel

Laravel merupakan salah satu framework PHP yang digunakan dalam pengembangan aplikasi berbasis web. Laravel membantu memaksimalkan penggunaan PHP dalam pengembangan *website* [2]. Hal ini dikarenakan Laravel menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu [3]. Laravel telah dikembangkan sejak tahun 2011 oleh Taylor Otwell dan telah mengalami perkembangan yang pesat. Framework ini berfokus pada kejelasan dan kerapian sehingga menghasilkan fungsionalitas aplikasi yang optimal. Hal inilah yang membuat laravel dapat digunakan dalam mengembangkan proyek dari skala kecil hingga besar. Laravel juga menggunakan konsep Model View Controller (MVC) sehingga proses pengembangan aplikasi menjadi lebih terstruktur.

C. Eloquent ORM

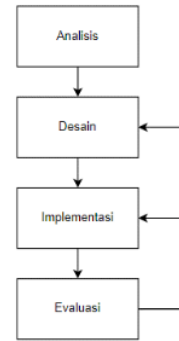
Laravel memberikan sebuah fitur yang sangat memudahkan pengembangnya dalam menuliskan *query*. Fitur tersebut adalah Eloquent ORM (*Object Relational Mapping*) [4]. Laravel Eloquent menyediakan berbagai fungsi untuk mengeksekusi suatu *query* dengan penggunaan yang mudah. Selain memberikan kemudahan dalam menuliskan *query*, Eloquent juga memberikan pencegahan terhadap serangan seperti SQL Injection. Hal ini disebabkan karena Eloquent menggunakan *PDO parameter binding* agar pengguna tidak dapat mengirim suatu *input* yang dapat mengubah maksud dari suatu *query* [5].

D. Laravel Artisan

Laravel Artisan adalah *Command-line Utility* (CLI) yang dimiliki oleh *framework* Laravel. Artisan menyediakan banyak perintah yang sangat memudahkan developer dalam mengembangkan sebuah aplikasi. Penggunaan artisan sangat membantu pengerjaan banyak hal yang bila dikerjakan secara manual akan memakan banyak waktu [6].

III. METODOLOGI

Metodologi yang digunakan dalam pengembangan aplikasi Balelabs Billing adalah Simple Scrum. Metode ini berfokus pada fase pengembangan yang iteratif. Dengan menggunakan Scrum, pengembangan aplikasi menjadi lebih adaptif terhadap kebutuhan sistem yang berubah. Alur metodologi pengembangan dapat dilihat pada gambar 1.



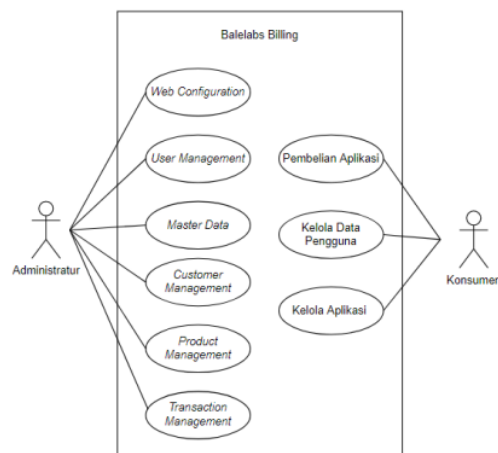
Gambar 1 Metodologi pengembangan aplikasi Balelabs Billing

A. Analisis

Pada tahap awal pengembangan, dilakukan pengumpulan data. Pengumpulan data dilakukan dengan melakukan wawancara pada pengguna dan observasi pada sistem sejenis. Data yang telah dikumpulkan kemudian dianalisis untuk mendapatkan kebutuhan sistem.

B. Desain

Berdasarkan analisis yang telah dilakukan, pengembangan dilanjutkan dengan membuat desain *use case diagram*, antarmuka, dan desain basis data untuk aplikasi. Desain dilakukan untuk memastikan masukan dan keluaran sesuai dengan kebutuhan pengguna. Gambar 2 merupakan hasil desain kebutuhan sistem dalam bentuk *use case diagram*.



Gambar 2 Use case diagram aplikasi Balelabs Billing

- a. *Web configuration* digunakan untuk mengelola *layout* dari *landing page* halaman publik pada aplikasi Balelabs Billing. Selain mengelola *layout*, menu ini juga digunakan untuk mengkonfigurasi email untuk notifikasi

- bagi konsumen. Sub menu *web configuration* meliputi *web setting, mail content, testimony, inquiry, dan FAQ*.
- User management* digunakan untuk mengelola pengguna dari halaman admin Balelabs Billing. Administrator dapat menambahkan pengguna baru dan mengatur hak akses mereka. Sub menu *user configuration* meliputi *permission, role, dan user*.
 - Master data* digunakan untuk mengelola data-data yang dibutuhkan terkait transaksi yang dilakukan oleh konsumen. Sub menu *master data* meliputi *GL account, dan bank*.
 - Customer management* digunakan untuk mengelola akun konsumen dan perusahaan konsumen. Sub menu *customer management* meliputi *company, dan customer*.
 - Product management* digunakan untuk mengelola setiap produk yang akan dijual pada konsumen balelabs. Sub menu *product management* meliputi *package category, package, package addon, dan package discount*.
 - Transaction management* digunakan untuk mengelola setiap transaksi yang telah dilakukan oleh konsumen balelabs.
 - Pembelian aplikasi adalah fungsionalitas yang diberikan pada konsumen agar dapat melakukan pembelian paket aplikasi di Balelabs Billing.
 - Kelola data pengguna memungkinkan konsumen untuk memperbarui data diri atau pun data perusahaan mereka.
 - Kelola aplikasi digunakan oleh konsumen untuk mengelola aplikasi yang telah dibeli di Balelabs Billing. Konsumen dapat menghentikan, ataupun memperbaharui masa langganan aplikasi mereka.

C. Implementasi

Pengembangan kemudian dilanjutkan dengan mengimplementasikan tahapan desain kedalam sistem/pengkodean. Digunakan beberapa teknologi sebagai sarana untuk mengimplementasikan sistem, seperti: Laravel, PostgreSQL, Bootstrap, Ajax, JQuery, dan Datatable. Teknologi ini digunakan atas hasil diskusi oleh seluruh pemegang kepentingan,

D. Evaluasi

Saat sebuah kebutuhan telah selesai diimplementasikan, kebutuhan tersebut kemudian dievaluasi agar sesuai dengan keluaran yang diharapkan. Apabila keluaran masih belum sesuai, fase pengembangan akan kembali pada tahap yang diperlukan. Sebagai contoh, apabila terdapat *bug/error* pada sistem, fase akan kembali pada tahap implementasi. Namun apabila terdapat kebutuhan yang berubah sehingga memerlukan pembaharuan desain, fase akan kembali pada tahap desain.

IV. PEMBAHASAN DAN HASIL

A. Pembahasan

Untuk mengimplementasikan Eloquent ORM pada proyek Laravel, hal pertama yang harus dilakukan adalah membuat model. Model sendiri adalah sebuah *class* yang mempresentasikan struktur dan logika dari basis data. Salah satu cara untuk mendefinisikan model adalah dengan menggunakan perintah Artisan **make:model**.

```
php artisan make:model SalesInvoiceLine
```

Gambar 3 Perintah artisan untuk membuat model

Gambar 3 merupakan contoh perintah yang dapat digunakan untuk membuat model menggunakan Laravel Artisan. *SalesInvoiceLine* pada perintah tersebut merupakan nama model yang akan dibuat.

Setelah perintah dieksekusi, sistem akan membuat *file* baru bernama *SalesInvoiceLine.php*. Untuk menghubungkan model pada basis data terkait, terdapat beberapa variabel yang harus ditambahkan, yakni nama dan *primary key* tabel. Gambar 4 merupakan struktur kode pada *file* *SalesInvoiceLine.php*, yang telah ditambahkan variabel.

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class SalesInvoiceLine extends Model
{
    protected $primaryKey = 'id';
    protected $table = 'sales_invoices_lines';
}
```

Gambar 4 Struktur kode *file* *SalesInvoicesLine.php*

Gambar 4 memperlihatkan bahwa *class* *SalesInvoiceLine* adalah kelas turunan dari *Model*. Hal ini memberikan *class* *SalesInvoiceLine* kemampuan untuk menggunakan fungsi-fungsi yang ada pada *class* *Model* dari Laravel.

Setelah model didefinisikan, *controller* kemudian dapat menggunakan *class* model tersebut untuk mengambil data dari basis data menggunakan Laravel Eloquent ORM. Untuk menggunakan sebuah model, *controller* harus mendeklarasikan *class* model pada awal kode. Gambar 5 memperlihatkan pendeklarasian model *SalesInvoiceLine* pada *controller*.

```
use App\Models\SalesInvoiceLine;
```

Gambar 5 Pendeklarasian model pada *controller*

Laravel Eloquent menyederhanakan penggunaan *query* sehingga kode menjadi lebih mudah dipelihara oleh developer. Eloquent menyediakan berbagai fungsi yang dapat digunakan untuk berinteraksi dengan basis data. Gambar 6 merupakan salah satu fungsi yang disediakan oleh Laravel Eloquent.

```
$items = SalesInvoiceLine::all();
```

Gambar 6 Fungsi Eloquent untuk memanggil seluruh data pada tabel

Gambar 6 merupakan salah satu fungsi yang disediakan Eloquent untuk mengambil seluruh data pada suatu tabel. Model *SalesInvoiceLine* yang telah dideklarasikan pada *controller* memanggil fungsi **all()** untuk mengambil seluruh

data dari tabel `sales_invoice_lines`. Data ini kemudian disimpan pada variabel “\$items” dalam bentuk object `collection`.

Terdapat beberapa model `query` yang bisa digunakan dalam mengembangkan sebuah aplikasi. Pada aplikasi web berbasis PHP dengan framework Laravel sendiri, terdapat dua model `query` yang dapat digunakan, yaitu: Eloquent ORM, dan raw SQL. Pada pengembangan aplikasi Balelabs Billing, Eloquent ORM dipilih sebagai model `query` yang digunakan.

Eloquent dipilih karena memberikan beberapa kelebihan, seperti *readability* dan *maintainability*. Secara *default*, penggunaan Eloquent juga memberikan keamanan yang lebih baik daripada model raw SQL sehingga aplikasi menjadi lebih aman terhadap serangan seperti SQL Injection [5]. Gambar 7 merupakan contoh `query` dengan model Eloquent, sedangkan Gambar 8 merupakan contoh `query` dengan model raw SQL.

```
$items = SalesInvoiceLine::find(2)
    →with('package', 'addon')
    →get();
```

Gambar 7 `Query` data dengan menggunakan Laravel Eloquent

```
$items = DB::select('select * from sales_invoices_lines
left join mgt_packages on
sales_invoices_lines.package_id = mgt_packages.id
left join mgt_package_addons on
sales_invoices_lines.addon_id = mgt_package_addons.id
where sales_invoice_lines.id = 2');
```

Gambar 8 `Query` data dengan menggunakan raw SQL

Gambar 7 dan 8 merupakan `query` yang digunakan untuk memanggil data dari tabel `sales_invoices_lines` yang memiliki id 2. `Query` tersebut juga memanggil dua tabel relasi, yakni: tabel `mgt_packages` dan tabel `mgt_package_addons`. Pada gambar 7, relasi tabel telah dimodelkan pada `class` model sehingga Eloquent hanya perlu memanggil model relasi tersebut.

Dapat dilihat pada gambar 7 bahwa kode yang ditulis lebih sederhana dan lebih mudah dibaca dibandingkan dengan kode pada gambar 8. Dengan kemudahan pembacaan kode, anggota tim dapat memahami kode yang ditulis dengan lebih baik. Hal ini juga membuat pemeliharaan sistem menjadi lebih efisien.

Perlu diketahui bahwa penggunaan Eloquent tidak selalu lebih baik dari raw SQL. Pada aspek kecepatan pengambilan data, model raw SQL masih lebih baik dari pada Laravel Eloquent. Hal ini disebabkan karena pada eloquent, setiap parameter yang diberikan akan dikeluarkan dalam bentuk objek sehingga proses pengambilan data menjadi lebih lambat. Sedangkan pada raw SQL, pengambilan data langsung dilakukan pada basis data dan dikeluarkan dalam bentuk array [7]. Tabel 1 memperlihatkan kecepatan respon data antara Laravel Eloquent dan raw SQL.

Tabel 1 Rata-rata waktu respon antara model `query` Laravel Eloquent dan raw SQL dalam detik

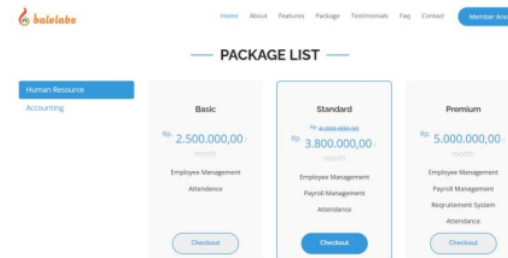
Join	Laravel Eloquent	Raw SQL
0	0.0029	0.0018
1	0.0065	0.0036
2	0.010	0.0045
Rata-rata	0.0064	0.0033

Pada tabel 1 dapat dilihat bahwa kecepatan pengambilan data pada raw SQL lebih unggul dibandingkan dengan Laravel Eloquent. Kecepatan pengambilan data semakin melambat seiring dengan bertambahnya jumlah `join` yang digunakan. Untuk rata-rata sendiri, raw SQL hampir dua kali lebih cepat dibandingkan dengan Laravel Eloquent. Oleh karena itu, pemilihan model `query` harus tetap disesuaikan dengan kebutuhan sistem.

B. Hasil Implementasi

1) Halaman Pemilihan Paket

Halaman pemilihan paket dikembangkan sebagai sarana konsumen untuk memilih setiap paket yang telah terdaftar pada aplikasi. Gambar 9 menampilkan paket semua paket yang terdaftar pada kategori Human Resource.



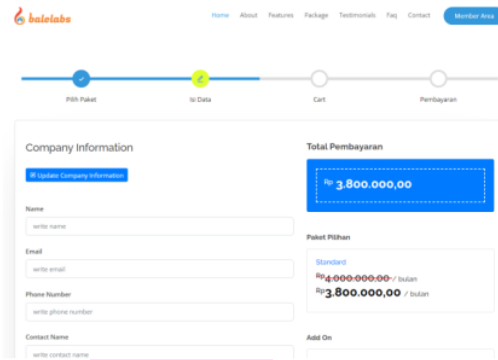
Gambar 9 Halaman pemilihan paket aplikasi

Pada halaman ini, ditampilkan setiap paket untuk setiap kategori yang ada. Saat ini terdapat dua kategori paket yang tersedia, yakni Human Resource dan Accounting. Terdapat tiga opsi paket yang diberikan pada kategori Human Resource, yaitu paket basic, standard, dan premium. Setiap opsi paket memiliki harga yang berbeda-beda tergantung dari jumlah `addon` yang ditawarkan.

Dengan adanya halaman ini, konsumen dapat mempertimbangkan paket yang akan dibeli dengan melihat harga dan `addon` yang ditawarkan. Setelah konsumen telah memutuskan paket yang akan dibeli, konsumen dapat menekan tombol `checkout` untuk masuk ke halaman berikutnya.

2) Halaman Formulir Pembelian

Halaman ini digunakan untuk konsumen mengisi data perusahaan yang dibutuhkan saat pembelian aplikasi sesuai pada gambar 10.

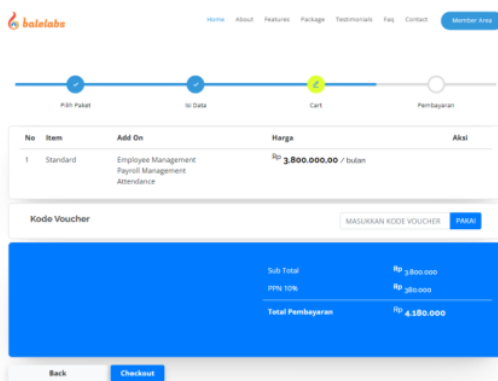


Gambar 10 Halaman formulir pembelian

Pada halaman ini, konsumen perlu memasukkan beberapa informasi, meliputi: nama perusahaan, email, nomor telepon, nama kontak, alamat, dan deskripsi perusahaan. Halaman ini juga menampilkan total harga dari paket yang telah dipilih. Setelah semua informasi dimasukkan, konsumen dapat melanjutkan ke halaman berikutnya dengan menekan tombol next.

3) Halaman Checkout

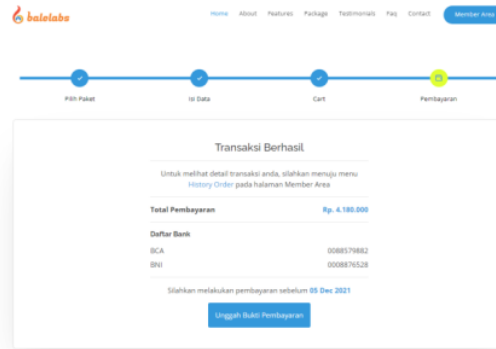
Halaman checkout menampilkan informasi mengenai total pembayaran yang harus dibayar oleh konsumen. Selain informasi mengenai total pembayaran, konsumen juga dapat memasukkan kode voucher yang mereka miliki pada kolom voucher. Tampilan halaman checkout dapat dilihat pada gambar 11.



Gambar 11 Halaman checkout

4) Halaman Informasi Pembayaran

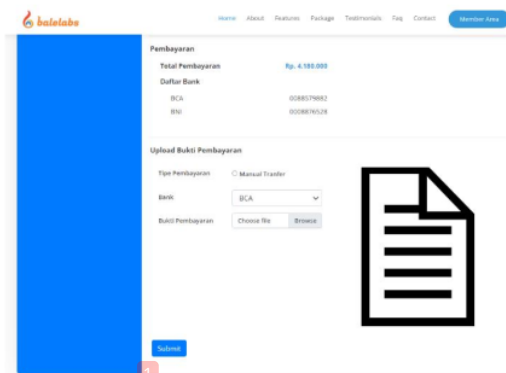
Ketika konsumen telah selesai melakukan checkout, halaman kemudian dilanjutkan dengan menampilkan informasi pembayaran yang harus dilakukan. Pada halaman ini, disajikan informasi mengenai total pembayaran, daftar bank yang dapat digunakan untuk pembayaran, dan batas akhir waktu pembayaran. Pengunjung kemudian dapat mengunggah bukti pembayaran dengan menekan tombol unggah bukti pembayaran. Tampilan halaman informasi pembayaran dapat dilihat pada gambar 12.



Gambar 12 Halaman informasi pembayaran

5) Halaman Unggah Bukti Pembayaran

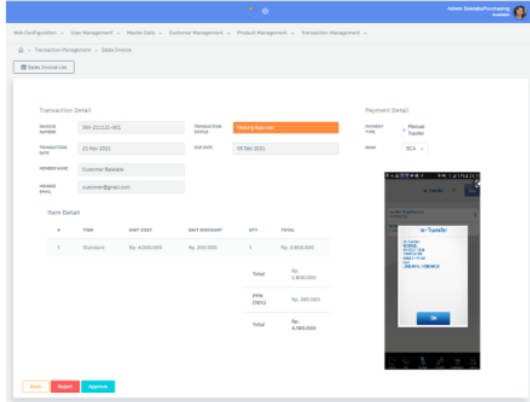
Konsumen yang telah melakukan pembelian paket aplikasi kemudian dapat mengunggah bukti pembayaran pada halaman unggah bukti pembayaran. Pada halaman ini, diberikan informasi mengenai total pembayaran yang harus dibayar dan daftar alamat bank yang tersedia. Konsumen dapat mengisi detail pembayaran pada bagian kedua pada halaman ini. Detail pembayaran yang harus dilengkapi meliputi tipe pembayaran, nama bank, dan gambar bukti pembayaran. Gambar 13 menampilkan halaman unggah bukti pembayaran.



Gambar 13 Halaman unggah bukti pembayaran

6) Halaman Verifikasi Pembayaran

Halaman verifikasi pembayaran digunakan oleh administrator untuk memverifikasi setiap pembayaran yang telah dilakukan oleh konsumen. Pada halaman ini, administrator dapat melihat dan mengecek detail transaksi yang telah dilakukan oleh konsumen. Detail transaksi yang ditampilkan meliputi nomor faktur, tanggal transaksi, nama konsumen, email konsumen, status transaksi, tenggat waktu pembayaran, paket aplikasi yang dibeli, dan bukti pembayaran. Tampilan dari halaman verifikasi pembayaran dapat dilihat pada gambar 14.



Gambar 14 Halaman verifikasi pembayaran

Administrator kemudian dapat memverifikasi bukti pembayaran berdasarkan data yang ada. Apabila data yang diberikan masih belum memenuhi ketentuan, administrator dapat menekan tombol reject untuk membatalkan transaksi. Sebaliknya, jika data transaksi yang dilakukan dirasa sudah memenuhi persyaratan, administrator dapat menekan tombol approve untuk menyetujui bukti pembayaran. Saat transaksi sudah disetujui oleh administrator, sistem akan men-generate aplikasi untuk konsumen sesuai dengan pembelian yang dilakukan.

7) Hasil Pengujian

2 Sistem yang telah dikembangkan kemudian diuji dengan menggunakan metode *black box testing*. Hasil dari pengujian dapat dilihat pada tabel 2.

Tabel 2 Hasil pengujian menggunakan metode *black box testing*

No.	Fungsionalitas	Hasil yang diharapkan	Hasil pengujian
1.	Menekan <i>checkout</i> pada halaman pemilihan paket saat pengguna tidak memiliki paket dalam keranjang	Masuk pada halaman pengisian formulir dengan paket yang telah dipilih	Masuk pada halaman pengisian formulir dengan paket yang telah dipilih
2.	Menekan <i>checkout</i> pada halaman pemilihan paket saat pengguna memiliki paket dalam keranjang	Muncul <i>pop-up</i> yang memberitahu pengguna bahwa terdapat aplikasi pada keranjang	Muncul <i>pop-up</i> yang memberitahu pengguna bahwa terdapat aplikasi pada keranjang
3.	Menggunakan voucher yang masih tersedia	Memperbarui total pembayaran berdasarkan	Memperbarui total pembayaran berdasarkan

	pada halaman <i>checkout</i>	potongan voucher yang digunakan	potongan voucher yang digunakan
4.	Menggunakan voucher yang tidak tersedia pada halaman <i>checkout</i>	Muncul keterangan bahwa voucher telah kadaluwarsa/ tidak tersedia	Muncul keterangan bahwa voucher telah kadaluwarsa/ tidak tersedia
5.	Mengunggah bukti pembayaran pada halaman unggah bukti pembayaran	Tombol submit menghilang dan status transaksi berubah menjadi menunggu verifikasi pembayaran	Tombol submit menghilang dan status transaksi berubah menjadi menunggu verifikasi pembayaran
6.	Menyetujui pembayaran pada halaman verifikasi pembayaran	Tombol <i>reject</i> dan <i>approve</i> menghilang. Status transaksi berubah menjadi telah disetujui	Tombol <i>reject</i> dan <i>approve</i> menghilang. Status transaksi berubah menjadi telah disetujui

V. KESIMPULAN

Model bisnis yang masih konvensional menyebabkan proses bisnis pada PT Bale Lab Indonesia menjadi kurang efisien. Hal ini mengakibatkan kurang maksimalnya keuntungan yang didapat oleh perusahaan, dan ketertinggalan teknologi terhadap kompetitor lain. Untuk mengatasi hal ini, dibuatlah aplikasi *e-commerce* bernama Balelabs Billing. Balelabs Billing dikembangkan untuk menjembatani transaksi antara konsumen dan perusahaan sehingga proses bisnis menjadi lebih efisien. Pada pengembangannya, Balelabs Billing memanfaatkan model query dari Laravel, yakni Eloquent ORM. Penggunaan Laravel Eloquent ditujukan untuk memenuhi kebutuhan *non-fungsional*, yakni kebutuhan akan masa pengembangan yang cepat, dan perawatan sistem yang mudah. Dari pengembangan sistem ini, dapat disimpulkan bahwa:

- Model bisnis *e-commerce* dipilih sebagai solusi untuk meningkatkan proses bisnis di PT Bale Lab Indonesia karena memberikan berbagai manfaat. Manfaat yang didapat antara lain: efisiensi transaksi, pengelolaan data transaksi yang lebih baik, dan memperluas jangkauan pasar.
- Eloquent ORM dapat digunakan ketika proyek membutuhkan struktur kode yang sederhana, *maintainability* yang mudah, dan keamanan bawaan yang baik.
- Eloquent ORM tidak direkomendasikan untuk digunakan pada aplikasi yang membutuhkan waktu pemuatan/performa yang cepat.

REFERENSI

- [1] T. Gajewska, D. Zimon, G. Kaczor and P. Madzik, "The Impact of The Level of Customer Satisfaction on The Quality of E-commerce Services," *International Journal of Productivity and Performance Management*, Vol. 69 No. 4, pp. 666-684. <https://doi.org/10.1108/IJPPM-01-2019-0018>, 2020.
- [2] Iwana, "Laravel Adalah," 6 February 2020. [Online]. Available: <https://jeriatno.medium.com/laravel-adalah-4f876fc71a72>.
- [3] K. NFA, "Laravel — Pengertian, Kelebihan, Kekurangan dan Cara Install Laravel," 4 September 2019. [Online]. Available: <https://medium.com/@kevinnfa0107/laravel-pengertian-kelebihan-kekurangan-dan-cara-install-laravel-224a79550a91>.
- [4] M. B. Alhaq and A. Sujarwo, "UTILISASI PENGOLAHAN PEMROSESAN DATA UNTUK MENINGKATKAN PERFORMA APLIKASI," *Automata Vol. 2 No. 1 2021*, vol. 2, no. 1, p. 330, 2021.
- [5] A. Pratama, P. Sukarno and A. A. Wardana, "Analisis dan Perbandingan Pencegahan SQL Injection pada Framework CodeIgniter dengan Escaping Query dan Framework Laravel dengan Eloquent ORM," *e-Proceeding of Engineering : Vol.7, No.1 April 2020*, vol. 7, p. 2245, 2020.
- [6] S. E. Wahyudi, "Artisan Console," 9 October 2019. [Online]. Available: <http://informatika.uc.ac.id/id/2019/10/laravel-artisan/>.
- [7] H. Budaraju, "Laravel Eloquent ORM Vs Raw SQL ?," 20 December 2017. [Online]. Available: <https://medium.com/@harshavardhanbudaraju/laravel-eloquent-orm-vs-raw-sql-d576276ee848>.

Implementasi Laravel Eloquent ORM pada Pengembangan Aplikasi E-Commerce Balelabs Billing

ORIGINALITY REPORT

9%

SIMILARITY INDEX

8%

INTERNET SOURCES

3%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1	repository.dinamika.ac.id Internet Source	2%
2	eprints.ums.ac.id Internet Source	1%
3	text-id.123dok.com Internet Source	1%
4	ejurnal.stmik-budidarma.ac.id Internet Source	<1%
5	doku.pub Internet Source	<1%
6	libraryproceeding.telkomuniversity.ac.id Internet Source	<1%
7	core.ac.uk Internet Source	<1%
8	journals.itb.ac.id Internet Source	<1%
9	jurnal.untan.ac.id Internet Source	<1%

10	www.coursehero.com Internet Source	<1 %
11	123dok.com Internet Source	<1 %
12	ninaherlinaibrahim.blogspot.com Internet Source	<1 %
13	slideplayer.info Internet Source	<1 %
14	delitama29.blogspot.com Internet Source	<1 %
15	docplayer.info Internet Source	<1 %
16	repository.its.ac.id Internet Source	<1 %
17	www.scribd.com Internet Source	<1 %
18	repository.uin-suska.ac.id Internet Source	<1 %
19	Pahrul Irfan, Kartarina Kartarina, Cristofer Satria. "SISTEM INFORMASI PEMASARAN PRODUK DESA BERBASIS WEB", ILKOM Jurnal Ilmiah, 2019 Publication	<1 %
20	jtiik.ub.ac.id Internet Source	<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On