



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1. Divisi dan Koordinasi

PT Wijaya Digital Ventures yang berlokasi di Jl. Sutera Boulevard Brooklyn Tower B/Lt. 09 Unit S Kav. 22 dan 26 Pakualam memiliki beberapa divisi utama untuk menunjang berjalannya proses bisnis yang dimilikinya. Divisi yang dimiliki oleh PT Wijaya Digital Ventures yaitu *Sales, HRD, Finance, IT*. Penulis melakukan praktik kerja magang pada divisi *IT* dari PT Wijaya Digital Ventures sebagai *Backend Programmer* dengan pekerjaan yang diterima sebagai berikut:

1. *Backend Engineer* untuk project *Promopick App* meliputi:
 - i. *Create authentication logic (register & login)* untuk *CMS*
 - ii. *CMS* untuk *coupon management* (*coupon create, read, update, delete*)
 - iii. *Coupon Summary* (*# of redeems, # of downloads, coupon history*)
2. *Create Rest API*
 - i. *JSON response* untuk menampilkan *Coupon*
 - ii. *Download* dan *redeem coupon*
 - iii. *Banner Management* (*create, read, update, delete*)
 - iv. *Search Function*

Divisi *IT* PT Wijaya Digital Ventures berada di bawah bimbingan Bapak Bambang Feri Hermansyah, S.Kom., M.Kom., selaku *IT* senior dan pembimbing

lapangan bagi penulis. Divisi *IT* pada bagian *Backend Programmer* memiliki peran untuk mengembangkan aplikasi sesuai dengan *Business logics* yang ada, melakukan pemeliharaan basis kode perusahaan, serta melakukan asistensi teknis terhadap tim *sales* PT Wijaya Digital Ventures, dan bekerja sama dengan divisi *IT* bagian *frontend application developer*.

3.2. Tugas yang Dilakukan

Penerimaan Mahasiswa sebagai salah satu anggota PT Wijaya Digital Ventures bertujuan untuk membantu proses pengembangan aplikasi *Promopick* PT Wijaya Digital Ventures. *Promopick* merupakan salah satu produk yang dimiliki oleh PT Wijaya Digital Ventures. Aplikasi *Promopick* bertujuan untuk membantu pemilik restoran untuk mendapatkan pelanggan pada jam-jam sepi pengunjung dengan cara memberikan diskon khusus pada jam-jam tertentu, sesuai dengan kerja sama yang telah di sepakati dengan *Promopick*, dengan cara menampilkannya di dalam aplikasi *Promopick*.

Bagian aplikasi yang harus dikembangkan untuk menunjang aplikasi *Promopick* adalah *CMS* atau *Coupon Management System*, sistem utama (*core system*) dari aplikasi *Promopick* yang bertujuan sebagai pusat kendali dari aplikasi *Promopick*. Aplikasi *CMS* berbasis web ini hanya boleh digunakan oleh karyawan PT Wijaya Digital Ventures yang memegang produk *Promopick*.

Website Aplikasi *CMS* ini memiliki beberapa proses bisnis yang kompleks yang digunakan untuk menunjang berjalannya aplikasi *Promopick*. Proses bisnis tersebut adalah sebagai berikut:

1. *Client Data*
2. *Banners*
3. *Summary*
4. *Others*

3.2.1. *Client Data*

Client Data merupakan tempat admin melakukan input terhadap data-data klien baru yang bergabung dengan aplikasi Promopick, seperti brand yang dimiliki, menu dari brand, harga, jam diskon, besaran diskon, tags, dan lain sebagainya.

3.2.2. *Banners*

Banners merupakan tempat admin untuk melakukan input *banner* untuk ditampilkan didalam aplikasi, menu *banners* berfungsi untuk mengatur, kapan *banner* akan tayang, kapan berakhir, berapa jumlah klik maksima, serta lain sebagainya.

3.2.3. *Summary*

Menu *Summary* merupakan menu yang hanya dapat muncul apabila *user* memiliki akses untuk dapat melihatnya. Menu ini bertujuan untuk memberikan gambaran statistik secara garis besar mengenai keseluruhan performa dari *Promopick*, seperti rasio *Coupon Download* dengan *Coupon Redemeed*, total transaksi per waktu, dan hal-hal lainnya yang membantu memberikan gambaran.

3.2.4. *Others*

Menu *Others* merupakan menu yang bertujuan untuk menampung fungsionalitas lain yang tidak terlalu sering digunakan. Salah satu contohnya adalah

untuk melakukan *input tag*, yang digunakan untuk menandai *Client*, dan *Brand*, sehingga dapat mempermudah melakukan pencarian internal.

3.3. Uraian Pelaksanaan Praktik Kerja Magang

Berikut merupakan *timeline* uraian kerja magang selama menjalani proses kerja magang di perusahaan PT Wijaya Digital Ventures untuk aplikasi *Promopick* yang dapat dilihat pada **Tabel 3.1**

Tabel 3.1 Uraian Praktik Kerja Magang Mingguan

Minggu ke-	Kegiatan yang dilakukan
1	<i>Create Authentication Logic</i> <ol style="list-style-type: none"> <i>Briefing</i> mengenai aplikasi <i>Promopick</i> <i>Create Auth Logic</i> Implementasi rancangan <i>Auth Logic</i> Pembuatan persyaratan yang dibutuhkan untuk melakukan <i>POST</i> data terhadap <i>Endpoint Auth</i> Membuat <i>Endpoint</i> untuk <i>Auth</i> Melakukan <i>testing</i> untuk memastikan kode berjalan dengan baik.
2	<i>CRUD for Coupon Mangement</i> <ol style="list-style-type: none"> Melakukan rapat dengan divisi <i>Promopick</i> Pembuatan <i>Coupon Category</i> Pembuatan <i>Coupon Start date End Date</i> Pembuatan <i>Coupon Location</i> Pembuatan <i>Coupon Insert Image</i> Pembuatan <i>Coupon Term and Conditions</i> <i>Link coupon</i> dengan <i>location</i> Pembuatan <i>Endpoint</i> untuk <i>Coupon</i> Melakukan pengecekan terhadap fungsionalitas untuk memastikan bahwa permintaan sesuai dengan hasil yang diberikan.
3	<i>Endpoint for Fetching Coupons (JSON)</i> <ol style="list-style-type: none"> Rapat dengan bagian <i>FrontEnd</i> untuk membahas struktur <i>JSON Payload</i> untuk <i>Fetching Coupon</i> Pembuatan persyaratan untuk melakukan <i>GET</i> data Pembuatan <i>response payload</i> untuk <i>Fetching Coupon</i>

	<ul style="list-style-type: none"> d. Pembuatan Endpoint untuk <i>Fetching Coupon</i> e. Melakukan pengecekan untuk memastikan <i>output</i> sesuai dengan kesepakatan awal perancangan struktur
4	<p><i>Endpoint for Coupon Download (JSON)</i></p> <ul style="list-style-type: none"> a. Rapat dengan bagian <i>FrontEnd</i> untuk membahas struktur JSON Payload untuk <i>Coupon Download</i> b. Rapat dengan bagian <i>FrontEnd</i> untuk membahas struktur JSON Payload untuk <i>Coupon Redeem</i> c. Pembuatan persyaratan untuk melakukan <i>POST</i> data bagi <i>Coupon Download</i> d. Pembuatan persyaratan untuk melakukan <i>POST</i> data bagi <i>Coupon Redeem</i> e. Pembuatan Endpoint untuk <i>Coupon Download</i>. f. Pembuatan Endpoint untuk <i>Coupon Redeem</i>. g. Melakukan pengecekan untuk memastikan <i>output</i> sesuai dengan kesepakatan awal perancangan struktur
5	<p><i>Endpoint for Coupon Redeem (JSON)</i></p> <ul style="list-style-type: none"> a. Rapat dengan divisi <i>Promopick</i> untuk membahas mengenai <i>Banner</i> b. Rapat dengan <i>Frontend</i> untuk membahas struktur pertukaran data yang dibutuhkan untuk pembuatan <i>Banner</i> c. Pembuatan <i>Banner Category</i> d. Pembuatan <i>Banner Start Date, End Date Input</i> e. Pembuatan <i>Banner Max Click</i> f. Pembuatan <i>Banner Price Per Click</i> g. Pembuatan <i>CRUD</i> untuk <i>Banner</i> h. Pembuatan persyaratan untuk melakukan <i>GET Banner</i> i. Pembuatan Endpoint untuk <i>GET Banner</i> h. Melakukan pengecekan untuk memastikan <i>output</i> sesuai dengan kesepakatan awal perancangan struktur
6	<p><i>CRUD for Banner Management</i></p> <ul style="list-style-type: none"> a. Rapat dengan divisi <i>Promopick</i> untuk membahas <i>Coupon Summary</i> b. Melakukan <i>query</i> yang dibutuhkan untuk menampilkan data-data yang diperlukan untuk melakukan visualisasi. c. Melakukan pengecekan untuk memastikan <i>output</i> sesuai dengan kesepakatan awal perancangan struktur
7	<p><i>Coupon Summary</i></p> <ul style="list-style-type: none"> a. Melakukan <i>Research</i> mandiri mengenai fitur <i>PostgreSQL Full Text Search</i> b. Melakukan penggabungan <i>query</i> dari beberapa table untuk melakukan <i>indexing</i> untuk melakukan <i>Full Text Search</i> menggunakan <i>Trigram</i> dan <i>Lexeme</i> c. Melakukan implementasi <i>PostgreSQL Full Text Search</i> pada <i>Endpoint Search</i> d. Melakukan <i>Research</i> mandiri mengenai <i>Solr</i>

8	<i>Search Endpoint (JSON)</i> <ul style="list-style-type: none"> a. Melakukan <i>Research</i> mandiri mengenai <i>Docker</i> b. Melanjutkan <i>Research</i> menggunakan <i>Solr</i> c. Melanjutkan <i>Research</i> menggunakan <i>Elasticsearch</i> d. Melanjutkan <i>Research</i> menggunakan <i>Kibana</i> e. Melanjutkan <i>Research</i> menggunakan <i>LogStash</i>
9	<i>Search Endpoint (JSON)</i> <ul style="list-style-type: none"> a. Melakukan <i>Indexing database</i> menggunakan <i>Elasticsearch</i> b. Rapat dengan bagian <i>FrontEnd</i> untuk membahas mengenai struktur <i>Response Payload</i> untuk di konsumsi <i>Frontend</i> c. Pembuatan persyaratan untuk melakukan <i>GET</i> data untuk <i>Endpoint Search</i> d. Pembuatan <i>Endpoint Search</i> e. Melakukan pengecekan untuk memastikan <i>output</i> sesuai dengan kesepakatan awal perancangan struktur

3.3.1. Minggu Pertama

Pada minggu pertama, kegiatan yang dilakukan adalah pembahasan mengenai, apa itu aplikasi *Promopick*, visi serta misi utama dari aplikasi *Promopick*. Setelah itu, melakukan pembahasan mengenai bagaimana cara melakukan implementasi dari *Authentication Logic* yang dibuat khusus untuk aplikasi *CMS* dan aplikasi *Promopick*. Pembahasan yang dilakukan adalah pembahasan mengenai bagaimana *user* dapat melakukan *access* terhadap server dan melakukan *request* data.

Dari segi *Administrator*, terdapat dua buah *role* yang akan di implementasikan. *Role* yang pertama adalah *role SuperAdmin*, role ini merupakan *role* yang memiliki kekuasaan penuh terhadap *CMS*, sama *role root* pada *System Unix*. *Role* yang selanjutnya adalah *role Admin*. *Role*

admin akan di berikan kepada staff yang ditugaskan untuk melakukan *maintenance* terhadap *client* yang ditugaskan kepadanya.

Selanjutnya, dari segi pengguna aplikasi, terdapat dua buah kategori yang akan diterapkan, yakni *Free member* dan *Premium Member*. *Free member* adalah anggota yang mendaftarkan dirinya pertama kali kedalam aplikasi (*Default*). Dimana *Premium Member* adalah anggota yang membeli keanggotaan terhadap aplikasi *Promopick*.

Tabel 3.2 menjelaskan mengenai *Endpoint* yang digunakan untuk mengakses hal-hal yang berhubungan dengan *Auth*, sedangkan pada **Tabel 3.3**, menjelaskan mengenai metode yang digunakan untuk melakukan *request data* serta *payload* yang dikirimkan. Setelah itu, dibahas pula mengenai *response* yang diberikan.

Tabel 3.2 Auth Endpoint

<i>API Endpoint</i>	Keterangan
<i>/api/v0/Auth</i>	<i>Endpoint untuk Auth</i>

Tabel 3.3 Auth Endpoint Details

<i>/register</i>	Method: POST Request Payload: { "username": "<username>", "password": "<password>", "email": "<email>", "gender": "<gender>" }
	Response Code: 201 400 500 201 Response:

	<pre> { "status": 201, "status_message": true, "data": [], "message": [], "error": [] } 400 Response (Invalid Param): { "status": 400, "status_message": false, "data": [], "message": "missing required parameter", "error": ["username", "email", "password"] } 400 Response (Duplicate Entry): { "status": 400, "status_message": false, "data": [], "message": "possible duplicate entry", "error": { "error_code": 2305, "error_message": "email already exists" } } 500 Response: { "status": 500, "status_message": false, "data": [], "message": "", "error": { "internal server error, please contact administrator" } } </pre>
/code_verify	<pre> Method: POST Request Payload: { "verification_code": "<verification_code>", "email": "<email>" } </pre>

	<p>Response Code: 200 401 500</p> <p>200 Response:</p> <pre>{ "status": 200, "status_message": true, "data": [], "message": "code verified", "error": [], }</pre> <p>401 Response:</p> <pre>{ "status": 401, "status_message": false, "data": [], "message": "invalid username / verification code", "error" : "Error during validating user. Please Contact Adminsitrator" }</pre> <p>500 Response:</p> <pre>{ "status": 500, "status_message": false, "data": [], "message": "internal server error, please contact administrator", "error" : "Error during validating user. Please Contact Adminsitrator" }</pre>
/resend_verification	<p>Method: POST</p> <p>Request Payload:</p> <pre>{ "email": "<email>" }</pre> <p>Response Code: 200 500</p> <p>200 Reponse:</p> <pre>{ "status": 200, "status_message": true, "data": [], "message": [], "error" : [] }</pre> <p>500 Response:</p> <pre>{ "status": 500, "status_message": true,</pre>

	<pre> "data": [], "message": "internal server error", "error" : "failed to resend code" } </pre>
/reset_password	<div> <div> Method: POST Request Payload: <pre> { "q": <reset token> <optional>, "email": "<email>" } </pre> </div> <div> Response Code: 200 403 500 200 Response: <pre> { "status": 200, "status_message": true, "data": [], "message": [], "error" : [] } </pre> 403 Response: <pre> { "status": 403, "status_message": false, "data": [], "message": "cannot reset password", "error" : "invalid code or time limit exceeded" } </pre> 500 Response: <pre> { "status": 500, "status_message": true, "data": [], "message": "internal server error", "error" : "failed to reset password" } </pre> </div> </div>
/login	<div> <div> Method: POST Request Payload: <pre> { "email": <email>, "password": <password> } </pre> </div> <div> Response Code: 200 400 401 500 200 Response: <pre> "status": 200, "status_message": true, "data": { </pre> </div> </div>

	<pre> "user_name": "Timothy", "user_email": "timothy@promopick.id", "user_id": 66, "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.e yJpc3MiOiIxOTIuMTY4LjEuNCIsImVudCI6M TU2OTEzODQ5OCwibmJmIjoXNTY5MTM4N Dk4LCJleHAiOiE1NjIxNDIwOTgsImRhdGEi OnsiaWQiOiY2LCJ1c2VyX25hbWUiOiJUaW1 vdGh5IiwidXNlcl9lbWVpbCI6InRpbW90aHku Y2hlYXBveUBnbWVpbC5jb20ifX0.1e5sffkM5 ZeiBNLbAJzymYPQVjutl8WaoS4LdLt1w84", "exp": 1569142098 }, "message": null, "error": null 400 Response: { "status": 400, "status_message": null, "data": null, "message": null, "error": { "missing_required_parameters":["email", "password"] } } 401 Response: { "status": 401, "status_message": false, "data": null, "message": "cannot login", "error": "invalid username / password" } 500 Reponse: { "status": 500, "status_message": true, "data": [], "message": "internal server error", "error" : "failed to resend code" } </pre>
--	--

3.3.2. Minggu Kedua

Pada minggu kedua bekerja, dilakukan rapat mingguan untuk membahas mengenai pembuatan *coupon management* untuk melakukan *CRUD Coupon*. Untuk dapat membuat *coupon* dari sebuah brand, alur yang harus dilakukan adalah sebagai berikut. Pertama *Client* harus terlebih dahulu didaftarkan. Tampilan pendaftaran *Client* dapat dilihat pada

Gambar

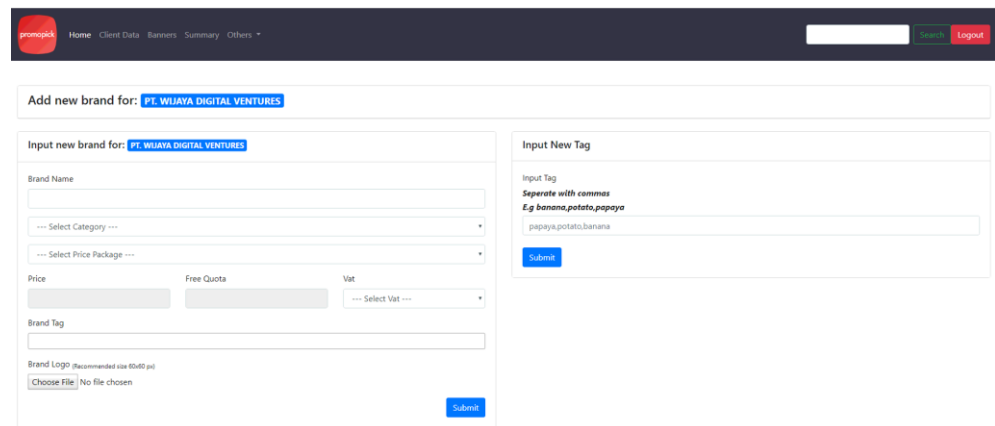
3.1

The screenshot shows a web application interface for 'Input Client Data'. At the top, there is a navigation bar with 'Home', 'Client Data', 'Banners', 'Summary', and 'Others'. Below this, the 'Input Client Data' form is displayed. It has a green header bar. The first section is 'Assign Admin' with a 'Select Admin' dropdown showing 'Super Admin (SuperAdmin)'. The second section is 'Client Company Data' with a 'Basic Data' sub-section. This sub-section contains several input fields: 'Client Company Name' (with a hint 'Eg Promopick'), 'Client Company Sub Name', 'Client Company Address' (with a hint 'Eg. Jalan Bungur 1 no 2'), 'Client Company Phone Number' (with a country code dropdown set to '+62' and a number input '899999999'), 'Client Company Email' (with the value 'admin@promopick.id'), and 'Client Company URL' (with the value 'www.promopick.id'). An 'Activate Windows' watermark is visible in the bottom right corner.

Gambar 3.1 *Input Client Data*

Langkah selanjutnya, setelah *Client* berhasil didaftarkan, adalah melakukan input brand yang menjadi bagian dari *Client* tersebut yang dapat dilihat pada **Gambar 3.2**, dimana data yang diperlukan oleh *Brand* akan

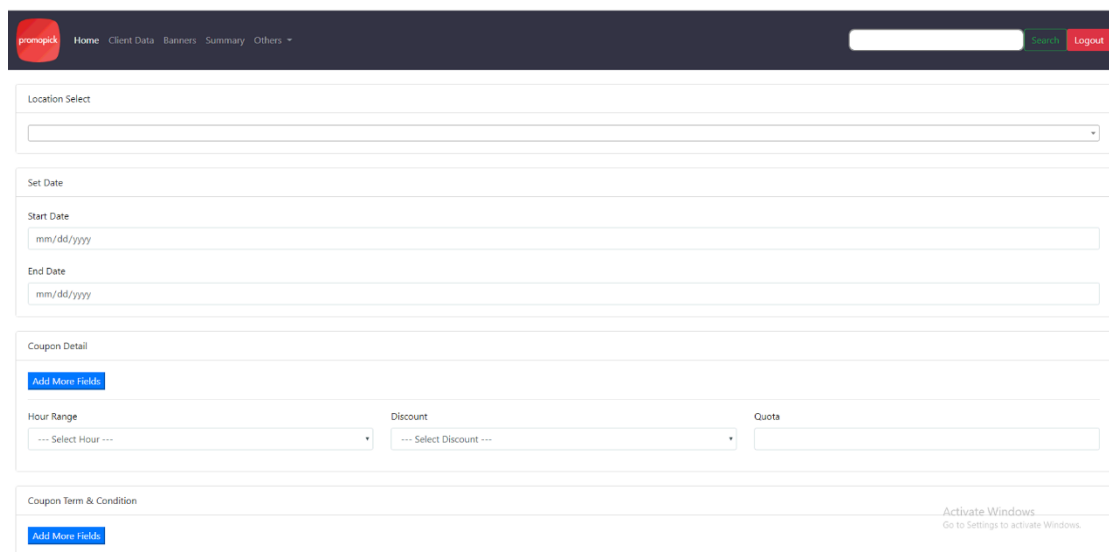
dimasukan.



The screenshot shows the 'Add new brand' form in the Promopick system. The form is titled 'Add new brand for: P.T. WIJAYA DIGITAL VENTURES'. It contains several input fields: 'Brand Name' (text input), 'Select Category' (dropdown), 'Select Price Package' (dropdown), 'Price' (text input), 'Free Quota' (text input), 'Vat' (dropdown), 'Brand Tag' (text input), and 'Brand Logo' (file upload). There is also a 'Submit' button at the bottom right. To the right of the main form, there is a section for 'Input New Tag' with a text input and a 'Submit' button.

Gambar 3.2 Input Brand

Setelah mengisi data yang diperlukan oleh *brand*, maka *Coupon* dapat dibuat dengan mengisi form yang dibutuhkan seperti pada **Gambar**



The screenshot shows the 'Input Coupon' form in the Promopick system. The form is divided into several sections: 'Location Select' (dropdown), 'Set Date' (with 'Start Date' and 'End Date' text inputs), 'Coupon Detail' (with 'Add More Fields' button, 'Hour Range' dropdown, 'Discount' dropdown, and 'Quota' text input), and 'Coupon Term & Condition' (with 'Add More Fields' button). There is also a 'Submit' button at the bottom right.

Gambar 3.3 Input Coupon

3.3.

3.3.3. Minggu Ketiga

Pada minggu ketiga, dilakukan pembuatan API untuk melakukan *fetching* data berdasarkan pekerjaan minggu kedua. *API Endpoint* ini berfungsi sebagai

tempat untuk mengambil seluruh kebutuhan data yang menyangkut *coupon*. Berkaitan dengan *Non Disclosure Agreement*, terdapat beberapa data yang tidak dapat ditampilkan di dalam laporan ini.

Tabel 3.4 menjelaskan mengenai *Endpoint* yang digunakan untuk mengakses hal-hal yang berhubungan dengan *Coupon*, sedangkan pada **Tabel 3.5**, menjelaskan mengenai metode yang digunakan untuk melakukan *request data* serta *payload* yang dikirimkan. Setelah itu, dibahas pula mengenai *response* yang diberikan.

Tabel 3.4 Coupon Endpoint

<i>API Endpoint</i>	Keterangan
<i>/api/v0/coupon</i>	<i>Endpoint untuk Coupon</i>

Tabel 3.5 Coupon Endpoint Detail

<i>/</i>	<i>Request Method: GET</i> <pre>{ "offset": 0, "limit": 100 }</pre> <i>Response Code: 200 / 500</i> <i>200 Response</i> <pre>{ "status": 200, "status_message": true, "data": [{ "coupon_id": <string>, "brand_name": <string>, "location_name": <string>, "short_location_name": <string>, </pre>
----------	--

	<pre> "long": <float>, "lat": <float>, "brand_logoUrl": "<string>", "details": [{ "hour_id": <int>, "hour_range": "18:00 - 19:00", "discount_value": 50 }, { "hour_id": <int>, "hour_range": "21:00 - 22:00", "discount_value": 50 }], "images": [{ "coupon_imgUrl": <string> }], "start_date": "YYYY-MM-DD", "end_date": null },] } 500 Response Code: { "status": 500, "status_message": true, "data": [], "message": "internal server error", "error": "failed to fetch coupon" } </pre>
/details	Untuk menampilkan detail <i>coupon</i>
/active	Untuk melakukan pengecek <i>active coupon</i> berdasarkan <i>user</i>
/delete	Untuk melakukan <i>delete coupon</i> oleh <i>user</i>
/history	Untuk melihat riwayat <i>coupon</i> dan transaksi yang pernah dilakukan oleh <i>user</i>

3.3.4. Minggu Keempat

Minggu keempat melanjutkan pekerjaan yang sebelumnya telah di bahas pada minggu ketiga, yakni penambahan *Endpoint* untuk *coupon redeem* dan *coupon download*.

Tabel 3.6 menjelaskan mengenai *Endpoint* yang digunakan untuk mengakses hal-hal yang berhubungan dengan *Coupon*, sedangkan pada **Tabel 3.7**, menjelaskan mengenai metode yang digunakan untuk melakukan *request data* serta *payload* yang dikirimkan. Setelah itu, dibahas pula mengenai *response* yang diberikan.

Tabel 3.6 Coupon Endpoint (2)

API Endpoint	Keterangan
/api/v0/coupon	Endpoint untuk Coupon

Tabel 3.7 Coupon Endpoint Detail (2)

/download	<i>Request Method: POST</i> { "user_id": <user_id>, "coupon_code": <coupon_code>, "nd": false, "hid": <numeric> } <i>Response Code: 200 / 400 / 500</i> <i>200 Reponse Code:</i> { "status": 200, "status_message": true, "data": null, "message": null, "error": null } <i>400 Response Code:</i>
-----------	--

	<pre> { "status": 400, "status_message": false, "data": null, "message": "Duplicate Download", "error": null } 500 Response: { "status": 500, "status_message": false, "data": [], "message": "internal server error, please contact administrator", "error": "Error during validating user. Please Contact Adminsitator" } </pre>
/redeem	<p><i>Request Method: POST</i></p> <pre> { "user_id": <user_id>, "coupon_code": <coupon_code>, "hid": <numeric>, "dis_id": <numeric>, "did": <numeric> } </pre> <p><i>Response Code: 200 / 400 / 500</i></p> <p><i>200 Response:</i></p> <pre> { "status": 200, "status_message": true, "data": [], "message": "coupon redeemed", "error": [] } </pre> <p><i>400 Response:</i></p> <pre> { "status": 400, "status_message": null, "data": null, "message": null, "error": { "missing_required_parameters": ["coupon_code", "user_id", "hid", </pre>

	<pre> "dis_id", "did"] } } } 500 Response: { "status": 500, "status_message": false, "data": [], "message": "internal server error, please contact administrator", "error": "Error during coupon redeem. Please Contact Adminsitrator" } </pre>
--	--

3.3.5. Minggu Kelima

Minggu kelima membahas mengenai *banner management system* untuk digunakan didalam aplikasi *Promopick*. *Banner Management System* bertujuan sebagai tempat utama bagi *admin* untuk melakukan *CRUD* terhadap *banner* dan mengatur fungsi-fungsi penting bagi *banner* seperti, *Price Per Click*, *Max Click Count*, *Start Date*, *End Date*, dan sebagainya yang dapat dilihat pada **Gambar**

3.4.

Tabel 3.7 menjelaskan mengenai *Endpoint* yang digunakan untuk mengakses hal-hal yang berhubungan dengan *Banner*, sedangkan pada **Tabel 3.8**, menjelaskan mengenai metode yang digunakan untuk melakukan *request data* serta *payload* yang dikirimkan. Setelah itu, dibahas pula mengenai *response* yang diberikan.

The screenshot shows the 'Input Banner Detail' form. At the top, there's a navigation bar with 'Home', 'Client Data', 'Banners', 'Summary', and 'Others'. Below this, the form has several sections: 'Banner on Front Page' with radio buttons for 'True' and 'False'; 'Banner Title Name' with a text input field; 'Banner Category' with a dropdown menu showing '--- SELECT CATEGORY ---'; 'Banner Type Category' with a dropdown menu showing '--- SELECT CATEGORY ---'; 'Start Date' and 'End Date' with date input fields showing 'mm/dd/yyyy'; 'Max Click Count' with a text input field; 'Price per Click' with a text input field; 'Banner Client' with a dropdown menu; and 'Banner Brand' with a dropdown menu. An 'Activate Windows' watermark is visible in the bottom right corner.

Gambar 3.4 *Input Coupon*

Tabel 3.7 *Banner Endpoint*

<i>API Endpoint</i>	<i>Keterangan</i>
<i>/api/v0/banner</i>	<i>Endpoint untuk Banner</i>

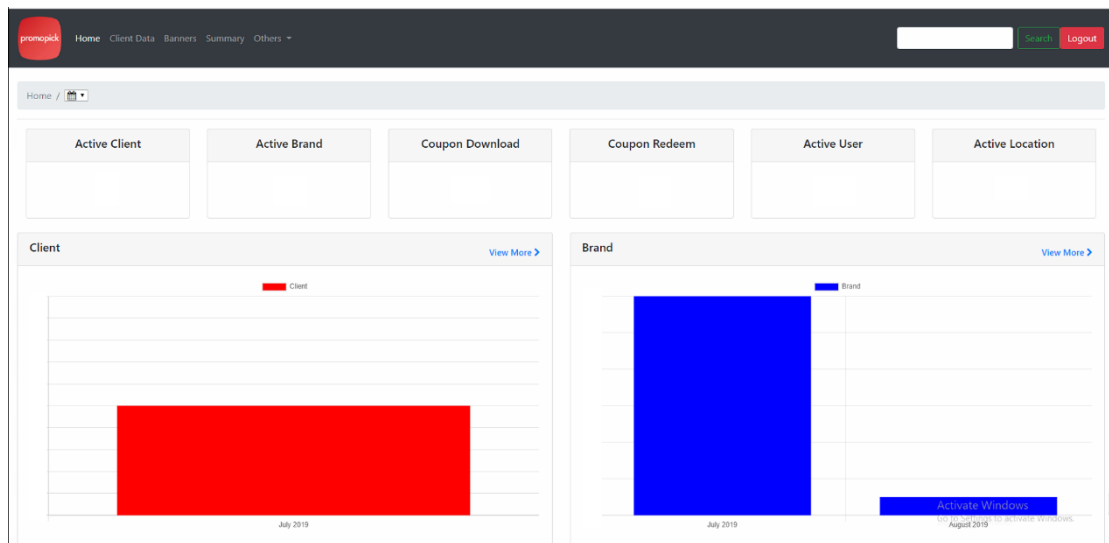
Tabel 3.8 *Banner Endpoint Detail*

<i>/</i>	<p><i>Request Method: GET</i></p> <p><i>Response Code : 200 500</i></p> <p><i>200 Response:</i></p> <pre>{ "status": 200, "status_message": "TRUE", "data": [{ "banner_id": <numeric>, "status_id": <numeric>, "banner_title": <string>, "banner_type_category": <numeric>, "banner_startDate": <string>, "banner_endDate": <string>, "banner_imgUrl": <string>, </pre>
----------	--

	<pre> "voucher_categoryId": <numeric>, "banner_clickCatId": <numeric>, "banner_urlValue": <string>, "banner_descriptions": null, "banner_createdDate": <timestamp>, "banner_customCode": null, "banner_isFrontBanner": <numeric> }] } 500 Response: { "status": 500, "status_message": true, "data": [], "message": "internal server error", "error": "failed to retrieve banner" }</pre>
--	---

3.3.6. Minggu Keenam

Minggu keenam membahas mengenai bagaimana menampilkan visualisasi dari data-data yang telah di kumpulkan guna memberikan gambaran yang lebih jelas mengenai apa yang terjadi antara interaksi pengguna dan aplikasi *Promopick* yang dapat dilihat pada **Gambar 3.5**.



Gambar 3.5 Coupon Summary, Data Disamarkan

Data-data yang telah dikumpulkan, diolah menjadi visualisasi, sehingga dapat memudahkan para *stakeholder*, untuk membuat strategi untuk arah aplikasi *Promopick*.

3.3.7. Minggu Ketujuh

Research mandiri dilakukan pada minggu ketujuh, guna menjawab kebutuhan atas perlunya *search engine* bagi aplikasi, baik untuk internal, ataupun untuk aplikasi *Promopick*.

Postgre Full Text Search merupakan sebuah fungsi yang disediakan oleh *PostgreSQL* untuk melakukan pencarian teks secara utuh, dengan menggunakan *query language*. (The PostgreSQL Global Development Group, 2019). Contoh *indexing* yang dihasilkan oleh *PostgreSQL Full Text Search* adalah seperti yang di gambarkan pada **Tabel 3.10**.

Tabel 3.9 TSVECTOR PostgreSQL Full Text Search

'americano':16B 'away':2A,19B 'barat':10 'bekasi':12 'coffee':3A,25B,34B 'digital':6 'drink':35B 'es':36B 'galaxy':14A 'hitam':31B 'indonesia':8 'jawa':9 'jelly':26B,29B 'kopi':17B,20B,28B,30B,32B,37B,38B 'kota':11 'latte':22B,39B 'minuman':40B 'mocha':23B 'no.3':15A 'pt':4 'ruko':13A 'susu':33B 'take':1A,18B 'ventures':7 'vietnam':21B 'white':24B,27B 'wijaya':5
--

Dan *query* yang digunakan untuk mendapatkan hasil di atas adalah seperti yang dijelaskan pada **Tabel 3.12**.

Tabel 3.10 Query PostgreSQL Full Text Search

```

select
    brnd.column,
    concat(
        setweight(to_tsvector('simple',coalesce(brnd.column)),
        'A') || ' ' ||
        to_tsvector('simple',coalesce(clnt."column")) || ' ' ||
        to_tsvector('simple',coalesce(cntry.column)) || ' ' ||
        to_tsvector('simple',coalesce(stts.column)) || ' ' ||
        to_tsvector('simple',coalesce(cty.column)) || ' ' ||
        setweight(to_tsvector('simple',
        coalesce(brnd_lc."column"), ' ')), 'A') || ' ' ||
        setweight(to_tsvector('simple',
        coalesce(brnd_mn."column"), ' ')), 'B') || ' ' ||
        setweight(to_tsvector('simple',
        coalesce(mstg.column), ' ')), 'B')
    )
    as brand_tags
from brand brnd
inner join client clnt on brnd."column" = clnt."column"
inner join brand_detail brnd_dt on brnd.column = brnd_dt.column
inner join brand_location brnd_lc on brnd_dt."column" =
brnd_lc."column"
inner join ms_country cntry on brnd_lc.column = cntry.column
inner join ms_states stts on brnd_lc.column = stts.column
inner join ms_city cty on brnd_lc.column = cty.column
inner join brand_menu brnd_mn on brnd_dt."column" =
brnd_mn.column
inner join brand_tag_list brnd_tgl on brnd_dt."column" =
brnd_tgl."column"
inner join ms_tag mstg on brnd_tgl.column = mstg.column

```

```

where brnd."column" = clnt."column" and
brnd.column = brnd_dt.column and brnd_dt."column" =
brnd_lc."column" and
cntry.column = brnd_lc.column and brnd_lc.column =
stts.column and cntry.column = stts.column and
brnd_lc.column = cty.column and stts.column = cty.column and
brnd_dt."column" = brnd_mn.column and
mstg.column = brnd_tgl.column and brnd_dt."column" =
brnd_tgl."column"
group by brnd.column, clnt."column", cntry.column, stts.column,
cty.column;

```

Namun *PostgreSQL* memakan waktu yang cukup lama untuk mengembalikan result dari *query* yang diberikan, sehingga, *PostgreSQL* menjadi pilihan terakhir apabila tidak terdapat aplikasi lain yang dapat memecahkan permasalahan yang ada.

Selanjutnya, dilakukan *research* mandiri kembali untuk menemukan jawaban atas masalah yang ada. Hasil dari *research* menemukan *Solr*, merupakan aplikasi yang dapat menjawab permasalahan yang ada. *Solr* merupakan program buatan *Apache Foundations* yang menggunakan aplikasi *Apache Lucene* sebagai dasarnya.

Pada dasarnya, *Solr* merupakan aplikasi *search engine*, dimana memiliki fitur yang serupa seperti *Postgre Full Text Search*, namun, yang membedakannya adalah, *Solr*, mengkususkan diri sebagai aplikasi mesin pencari. Sehingga performa *Full Text Search* yang diberikan lebih mumpuni dan stabil, jika dibandingkan dengan *Postgre Full Text Search*.

3.3.8 Minggu Kedelapan

Minggu kedelapan merupakan minggu *research* mandiri mengenai *Docker* dan *Elasticsearch*. *Elasticsearch* merupakan *search engine* yang menyerupai *Apache Solr*. Sama-sama berdasarkan *Apache Lucene*. *Elasticsearch* menawarkan interaksi *API* secara *RESTful*, dan mengeluarkan hasil pencarian dengan format *JSON* sebagai output *default*, dimana *Apache Solr* tidak menjadikan *JSON* sebagai *output* utama, karena *Solr* mendukung output dalam bentuk *JSON*, *XML*, *PHP*, *Python*, *Ruby*, *CSV*, *Velocity*, *XSLT*, *native Java*. *Elasticsearch* dipilih, dikarenakan dukungan *API* terhadap *client PHP*, dimana *Solr* hanya menawarkan dalam Bahasa *Java*.

Selain dari pada itu, aplikasi *Promopick* dibuat menggunakan *JSON* dan *Elasticsearch* dibuat untuk mendukung aplikasi berbasis *JSON*, sehingga banyak fitur yang tersedia yang sudah dibuat dalam bentuk *JSON*, sehingga dapat memudahkan pengerjaan pembuatan aplikasi.

Elasticsearch dapat berjalan di atas *Docker*, ataupun secara *native* pada computer server. Untuk dapat menggunakan *Elasticsearch* ada baiknya apabila disertakan instalasi program *Kibana*, sebuah *dashboard* bagi *Elasticsearch* yang membantu menggambarkan data yang kita miliki didalam *Elasticsearch*.

3.3.9. Minggu Kesembilan

Elasticsearch di pasang di server menggunakan *image* dari *docker*, lalu dengan menggunakan *script* yang tersedia dari website *docker*, dilakukan sedikit perubahan untuk menyesuaikan kebutuhan.

API Endpoint untuk *search* yang di berikan oleh *Elasticsearch* pada port 9200 memberikan *response* seperti pada **Tabel 3.13**.

Tabel 3.13 *Elasticsearch Result*

```
{
  "name": "es01",
  "cluster_name": "docker-cluster",
  "cluster_uuid": "PpNnhRGhQvyhZCHWvJouVg",
  "version": {
    "number": "7.3.1",
    "build_flavor": "default",
    "build_type": "docker",
    "build_hash": "4749ba6",
    "build_date": "2019-08-19T20:19:25.651794Z",
    "build_snapshot": false,
    "lucene_version": "8.1.0",
    "minimum_wire_compatibility_version": "6.8.0",
    "minimum_index_compatibility_version": "6.0.0-beta1"
  },
  "tagline": "You Know, for Search"
}
```

Setelah melakukan *indexing* data pada *Elasticsearch*, melakukan *GET request* kepada *Endpoint* yang di sediakan, untuk melihat hasil data yang telah di simpan di dalam *Elasticsearch*. Detail *endpoint* dapat dilihat pada **Tabel 3.14**

Tabel 3.114 Elasticsearch Result Untuk index promopick-search

/promopick-search/_search?q=*	<pre> { "took": 29, "timed_out": false, "_shards": { "total": 5, "successful": 5, "skipped": 0, "failed": 0 }, "hits": { "total": { "value": 80, "relation": "eq" }, "max_score": 1.0, "hits": [{ "_index": "promopick-search", "_type": "_doc", "_id": "mPnD1mwB4nYXSOMcPvxb", "_score": 1.0, "_source": { "brand_id": <int>, "brand_locationAddress": <string>, "brand_locationId": <int>, "brand_locationName": <string>, "brand_locationPoint": [<float>, <float>], "brand_locationShortName": <string>, "brand_logoUrl": <string>, "brand_name": <string>, "coupon_code": <string>, "coupon_image": [</pre>
-------------------------------	--

	<pre> <string>], "menu": ["Banana Bread", "Capuccino", "Chocolate Muffin", "Flat White", "Latte", "Long Black", "Macaron", "Piccolo Latte"], "tags": ["americano", "espresso", "kopi hitam", "kopi susu", "piccolo", "capuccino", "coffee", "drink", "es kopi", "kopi", "latte", "long black", "minuman"] },] } </pre>
--	--

3.4. Kendala

Berikut kendala yang didapatkan saat menjalani proses Praktik Kerja Lapangan di PT Wijaya Digital Ventures:

1. Terdapat beberapa istilah yang di berikan oleh *Subject Matter Expert* yang tidak diketahui, sehingga pada saat melakukan pengerjaan tugas yang diberikan, memerlukan waktu lebih untuk dapat mengerti terlebih dahulu.

3.5. Solusi

Berikut adalah solusi yang dapat dilakukan untuk mengatasi kendala yang dihadapi selama melakukan Praktik Kerja Lapangan di PT Wijaya Digital Ventures:

1. *Subject Matter Expert* memberikan pelajaran serta selalu bersedia menjelaskan kembali istilah-istilah yang tidak familiar, serta bersedia mendampingi proses pengerjaan aplikasi.