



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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:

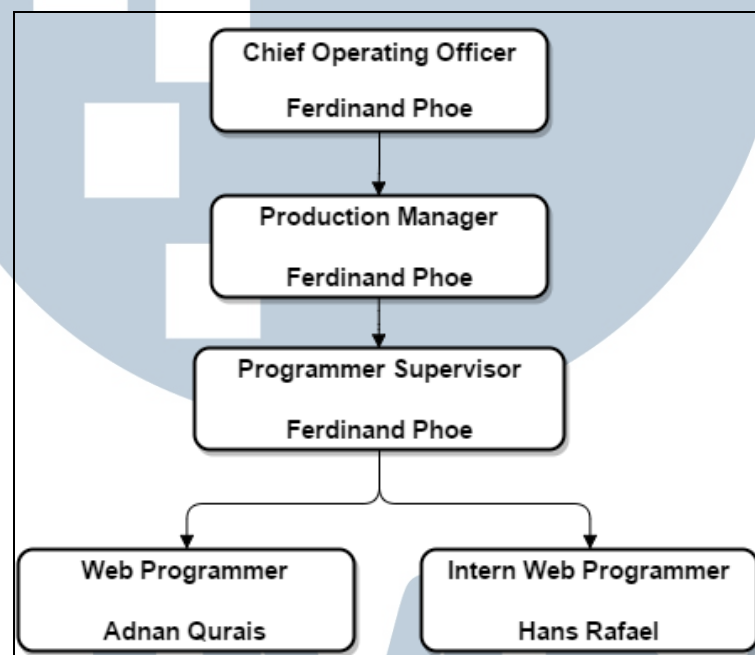
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kerja magang sebagai *web developer* yang dilakukan di Elven Digital Indonesia memiliki struktur koordinasi seperti pada Gambar 3.1.



Gambar 3.1 Struktur Koordinasi Magang

Secara umum, pengerjaan modul dikepalai oleh *Programmer Supervisor* yang kebetulan juga merupakan *Production Manager* dan *Chief Operating Officer*, *Programmer Supervisor* ini mengepaloi *Web Programmer* yang mengerjakan kode dan peserta magang yang bertugas membantu *Web Programmer*.

Selama pengerjaannya, kerja magang ini dibantu oleh *Web Programmer*. Setiap hasil pengerjaan dari fungsi-fungsi yang ada selalu dikomunikasikan kepada

Chief Operating Officer dan *Web Programmer* agar jika ada perubahan dalam desain dan proses bisnis, perubahan tersebut dapat segera diimplementasikan.

3.2 Tugas yang Dilakukan

Dalam membangun modul *promotion* pada situs *marketplace* Birds & Bees, terdapat 4 tahapan sebelum hasil akhir disampaikan kepada *Chief Operating Officer*, yaitu:

1. *Analysis Requirement*, yaitu menganalisis *requirement* untuk modul *promotion*. Analisa dilakukan dengan mencermati penyampaian proses bisnis oleh *Web Programmer*.
2. *Design*, yaitu merancang desain yang terdiri dari *sequence diagram*, dan *class diagram* yang berfungsi untuk mempermudah pembuatan kode modul *promotion* pada situs *marketplace* Birds & Bees.
3. implementasi, yaitu pembuatan kode program untuk modul *promotion*.
4. *Screenshot*, yaitu Tampilan *user interface* telah memenuhi *requirement* dan sesuai dengan proses bisnis yang ada, diawasi oleh *web Programmer*.

3.3 Uraian Pelaksanaan Kerja Magang

Hal – hal yang dilakukan selama pelaksanaan kerja magang terdapat pada Tabel 3.1. Uraian pekerjaan disusun berdasarkan hal yang dilakukan per minggunya. Berikut pula dijabarkan penjelasan dari uraian pekerjaan yang telah dilakukan selama 8 minggu.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Tabel 3.1 Uraian Pekerjaan yang Dilakukan Per Minggu

Minggu ke-	Uraian Pekerjaan
1	<ul style="list-style-type: none"> - Mempelajari penggunaan Laravel PHP <i>Framework</i>. - Membuat <i>home page</i> untuk <i>website</i> Clairemont
2	<ul style="list-style-type: none"> - Memasukkan data gambar untuk <i>website</i> seandy
3	<ul style="list-style-type: none"> - Mempelajari cara membuat <i>responsive website</i> - Membuat <i>landing page</i> untuk <i>website</i> vakanesia.com
4	<ul style="list-style-type: none"> - Mulai mengerjakan modul <i>promotion</i> pada <i>website</i> Birds & Bees
5	<ul style="list-style-type: none"> - Mengerjakan modul <i>promotion</i> pada <i>website</i> Birds & Bees
6	<ul style="list-style-type: none"> - Memindahkan modul <i>promotion</i> dari sandbox http://chronosh.com/birdnbees/ ke http://chronosh.com/birdsandbees/
7	<ul style="list-style-type: none"> - Mengerjakan <i>User Interface</i> untuk <i>website</i> vakanesia.com
8	<ul style="list-style-type: none"> - Melanjutkan pekerjaan <i>User Interface</i> untuk <i>website</i> vakanesia.com

Pada minggu pertama, mempelajari pemasangan dan penggunaan Laravel PHP Framework yang mirip dengan materi perkuliahan yaitu CodeIgniter *PHP Framework* yang juga menerapkan konsep MVC (*model-view-controller*). Lalu melakukan *slicing*, yaitu merubah data dari bentuk *Photoshop* menjadi *html website clairmont*.

Pada minggu kedua, diberikan tugas untuk memasukkan data gambar baju batik serta *caption* dari *list* barang yang terdapat pada *marketplace*.

Pada minggu ketiga, diberikan tugas untuk membuat *landing page website* vakanesia.com. Dan diajarkan juga cara membuat *website* menjadi *responsive* sesuai dengan media yang sedang dipakai oleh pengguna dengan memakai *@media screen*.

Pada minggu keempat, mulai mengerjakan modul *promotion* pada *website* Birds & Bees.

Pada minggu kelima, melakukan *briefing* lebih lanjut dengan *Chief Operating Officer* dan *Web Programmer* dari hasil pengerjaan selama satu minggu dan melanjutkan pengerjaan modul.

Pada minggu ketujuh, mengerjakan *slicing* untuk *home page* *vakanesia.com*.

Pada minggu kedelapan, melakukan *briefing* lebih lanjut dengan *Chief Operating Officer* dan *Web Programmer* dan melanjutkan pekerjaan *slicing* untuk *home page* *vakanesia.com*.

3.3.1 Proses Pelaksanaan

Analysis Requirement merupakan proses menganalisa *requirement* yang terdapat pada modul *promotion* berdasarkan hasil penjelasan proses bisnis yang terjadi pada modul *promotion website* Birds & Bees. Hasil dari proses ini adalah *use case*.

Design merupakan proses pembuatan desain yang akan digunakan. Pada proses ini dilakukan pembuatan *activity diagram* untuk menggambarkan aktivitas yang terjadi, *sequence diagram* untuk menggambarkan perpindahan data dan *class diagram* yang digunakan untuk menggambarkan *class* yang dibutuhkan.

Proses implementasi merupakan proses pembuatan kode program untuk setiap fungsi pada modul *promotion* sehingga sistem dapat berjalan sesuai dengan

requirement yang ada. Kode program yang dibuat untuk modul *promotion* terdiri dari *controller* dan *view*.

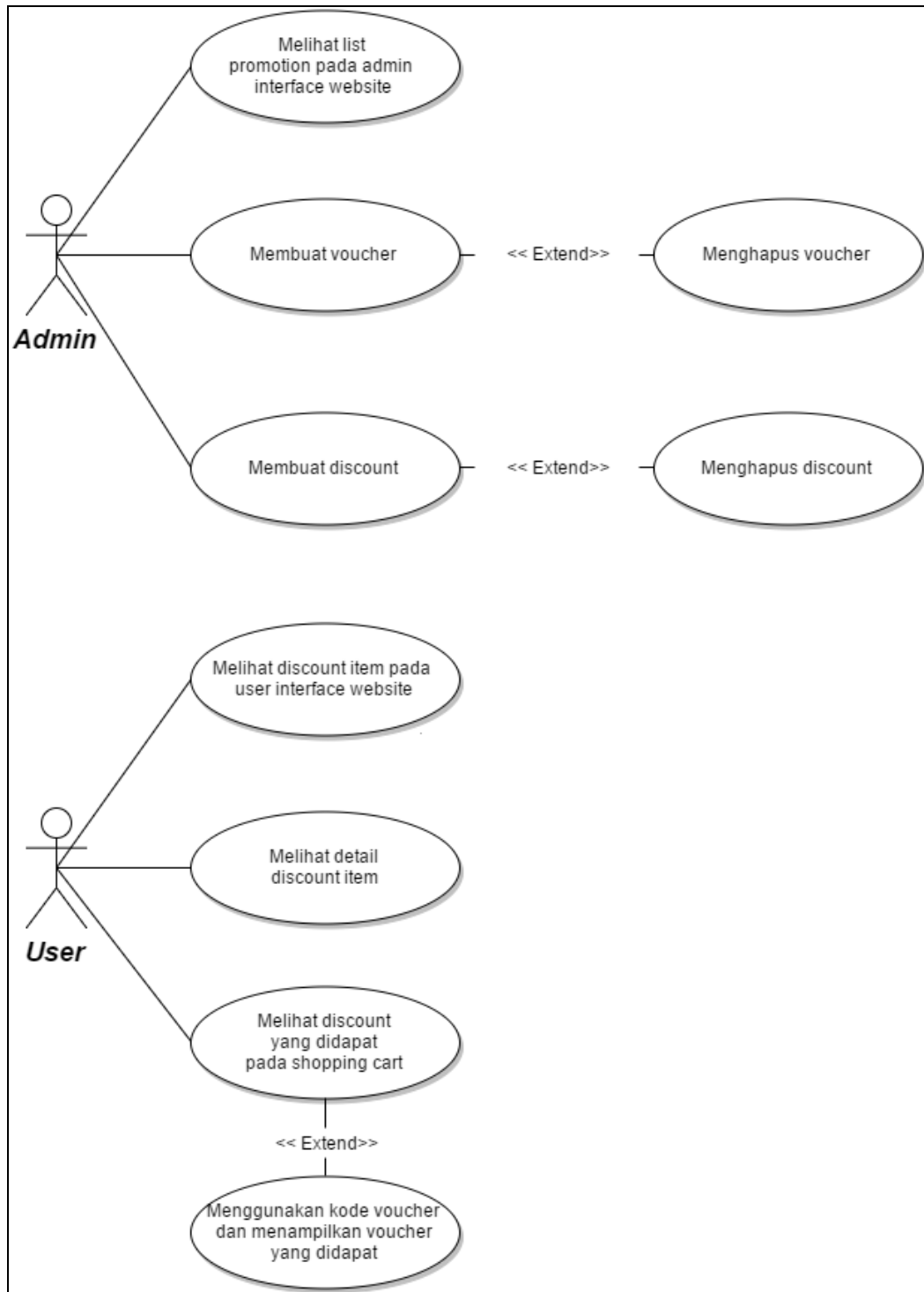
Screenshot merupakan *user interface* situs yang telah dibuat. Proses ini dilakukan dengan menunjukkan hasil setiap proses implementasi yang ada pada modul *promotion*.

3.3.2 Hasil Pelaksanaan

Hasil pelaksanaan kerja magang adalah *use case*, *activity diagram*, *sequence diagram*, *class diagram*, *database schema*, implementasi, dan *Screenshot*. Bagian implementasi berisi spesifikasi perangkat keras, spesifikasi perangkat lunak, konsep MVC yang digunakan, dan potongan kode program. Bagian *Screenshot* berisi *user interface* fungsionalitas modul *promotion* pada website Birds & Bees.

A. Use Case Diagram

Use Case Diagram digunakan untuk penggambaran sederhana dari interaksi yang terjadi di dalam sistem. Berikut digambarkan *use case diagram* yang digunakan untuk modul *promotion*. Pada Use Case Diagram digambarkan 2 buah aktor yaitu Admin yang mengerjakan proses melihat list promotion pada admin interface website, membuat dan menghapus voucher, dan membuat dan menghapus discount, dan User yang mengerjakan proses melihat discount item pada user interface website, melihat detail discount item, melihat discount yang didapat pada shopping cart, dan menggunakan kode voucher dan menampilkan voucher yang didapat.



Gambar 3.2 Menggambarkan *use case diagram* untuk modul *promotion*

Terdapat 8 buah tabel yang digunakan untuk menjelaskan *use case diagram* diatas, antara lain:

Tabel 3.2 Tabel melihat *list promotion*

Use Case Name	Melihat <i>list promotion</i> pada <i>admin interface website</i>
Actor	Admin
Description	Merupakan <i>event</i> dimana aktor dapat melihat <i>voucher</i> dan <i>discount</i>
Trigger	
Normal Flow of Event	1. Aktor melihat tampilan <i>website</i> admin
Pre-Condition	Melakukan <i>login</i> <i>admin</i> pada <i>chronosh.com/birdnbees/admin</i> , dan membuka bagian <i>promotion</i>
Post-condition	Aktor melihat tampilan <i>website admin</i> bagian <i>promotion</i>

Tabel 3.3 Tabel Membuat *voucher*

Use Case Name	Membuat <i>voucher</i>
Actor	Admin
Description	Merupakan <i>event</i> dimana aktor dapat membuat <i>voucher</i>
Trigger	
Normal Flow of Event	1. Aktor memilih <i>voucher</i> pada menu navigasi 2. Aktor menekan <i>add voucher</i> untuk menambahkan <i>voucher</i> baru 3. aktor mengisi form penambahan <i>voucher</i> . Tekan <i>submit</i> untuk melakukan penambahan, atau <i>back</i> untuk membatalkan penambahan
Pre-Condition	Melakukan <i>login</i> <i>admin</i> pada <i>chronosh.com/birdnbees/admin</i> , dan membuka bagian <i>voucher</i>
Post-condition	Aktor berhasil membuat <i>voucher</i>

Tabel 3.4 Tabel menghapus *voucher*

Use Case Name	Menghapus <i>voucher</i>
Actor	Admin
Description	Merupakan <i>event</i> dimana aktor dapat menghapus <i>voucher</i> yang telah dibuat
Trigger	
Normal Flow of Event	1. Aktor menekan <i>delete</i> pada kolom <i>action</i> untuk menghapus <i>voucher</i> atau <i>discount</i>
Pre-Condition	Melakukan <i>login</i> <i>admin</i> pada <i>chronosh.com/birdnbees/admin</i> , dan membuka bagian <i>discount</i>
Post-condition	Aktor berhasil menghapus <i>voucher</i>

Tabel 3.5 Tabel Membuat *discount*

Use Case Name	Membuat <i>discount</i>
Actor	Admin
Description	Merupakan <i>event</i> dimana aktor dapat membuat <i>discount</i>
Trigger	Aktor ingin membuat <i>discount</i>
Normal Flow of Event	<ol style="list-style-type: none"> 1. Aktor memilih <i>voucher</i> pada menu navigasi 2. Aktor menekan <i>add discount</i> untuk menambahkan <i>discount</i> baru 3. aktor mengisi form penambahan <i>discount</i>. Tekan <i>submit</i> untuk melakukan penambahan, atau <i>back</i> untuk membatalkan penambahan
Pre-Condition	Melakukan <i>login admin</i> pada <i>chronosh.com/birdnbees/admin</i> , dan membuka bagian <i>discount</i>
Post-condition	Aktor berhasil membuat <i>discount</i>

Tabel 3.6 Tabel menghapus *discount*

Use Case Name	Menghapus <i>discount</i>
Actor	Admin
Description	Merupakan <i>event</i> dimana aktor dapat menghapus <i>discount</i>
Trigger	Aktor ingin menghapus <i>discount</i>
Normal Flow of Event	4. Aktor Menekan <i>delete</i> pada kolom <i>action</i> untuk menghapus <i>discount</i>
Pre-Condition	Melakukan <i>login admin</i> pada <i>chronosh.com/birdnbees/admin</i> , dan membuka bagian <i>discount</i>
Post-condition	Aktor berhasil menghapus <i>discount</i>

Tabel 3.7 Tabel melihat *discount item* pada *user interface*

Use Case Name	Tabel melihat <i>discount item</i> pada <i>user interface</i>
Actor	<i>User</i>
Description	Merupakan <i>event</i> dimana aktor dapat melihat <i>list item</i> yang diberi <i>discount</i>
Trigger	
Normal Flow of Event	1. Aktor melihat tampilan <i>website</i> , dan memilih <i>list item</i> yang diberi <i>discount</i> pada menu navigasi
Pre-Condition	Aktor mengakses <i>website</i> <i>chronosh.com/birdnbees</i>
Post-condition	Aktor berhasil melihat <i>list item</i>

Tabel 3.8 Tabel melihat detail *discount item* pada *user interface*

Use Case Name	Tabel melihat detail <i>discount item</i> pada <i>user interface</i>
Actor	<i>User</i>
Description	Merupakan <i>event</i> dimana aktor dapat melihat <i>detail item</i> yang diberi <i>discount</i>
Trigger	Aktor memilih <i>item discount</i> dari <i>list item</i>
Normal Flow of Event	1. Aktor dapat memilih item untuk melihat <i>detail item</i> yang diberi <i>discount</i>
Pre-Condition	Aktor mengakses <i>website</i> chronosh.com/birdnbees
Post-condition	Aktor berhasil melihat <i>detail item</i> yang diberikan <i>discount</i>

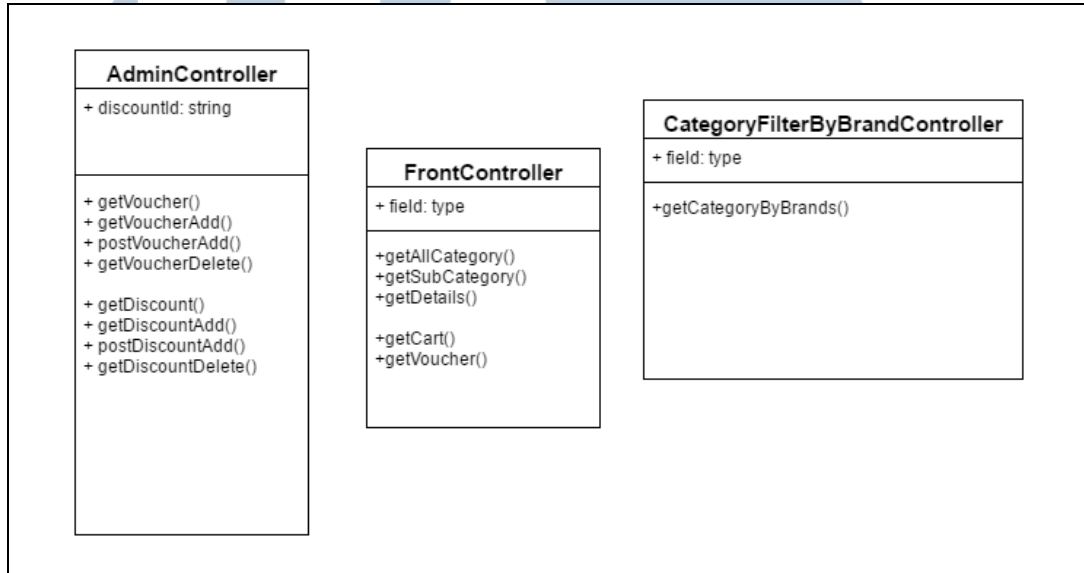
Tabel 3.9 Tabel melihat *discount* yang didapat pada *shopping cart*

Use Case Name	Melihat <i>discount</i> yang didapat pada <i>shopping cart</i>
Actor	<i>User</i>
Description	Merupakan <i>event</i> dimana aktor dapat melihat <i>discount</i> yang didapat pada <i>shopping cart</i> , dan menggunakan <i>voucher</i>
Trigger	
Normal Flow of Event	2. Aktor dapat masuk ke <i>shopping cart</i> untuk melihat <i>discount</i> yang didapat 3. (optional) Dapat memilih untuk memasukkan kode <i>voucher</i> , dan melihat <i>voucher</i> yang didapat
Pre-Condition	Aktor mengakses <i>website</i> chronosh.com/birdnbees
Post-condition	Aktor berhasil melihat <i>discount</i> pada <i>shopping cart</i> Aktor berhasil memasukkan kode <i>voucher</i> dan melihat <i>voucher</i> yang didapat



B. Class Diagram

Pembuatan *class diagram* digunakan untuk menggambarkan hubungan antara *class* yang dibutuhkan oleh modul *promotion*.

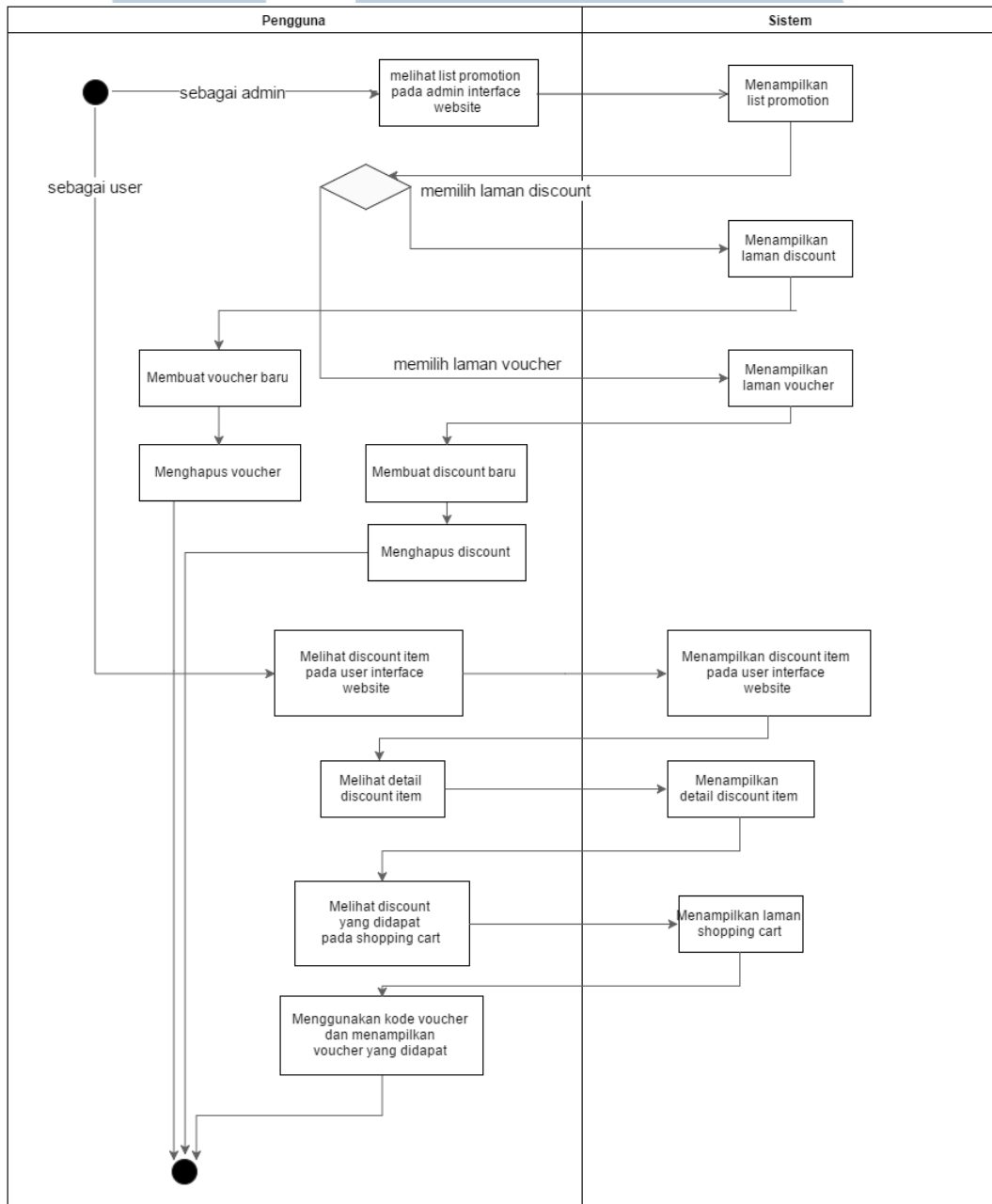


Gambar 3.3 Menggambarkan *class diagram* untuk modul *promotion*

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

C. Activity Diagram

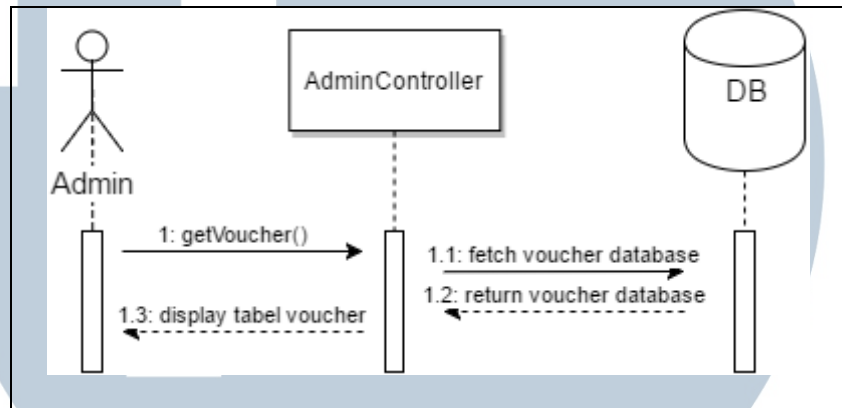
Activity diagram bertujuan untuk menggambarkan arus aktivitas yang terjadi antara pengguna dan sistem. Pada gambar 3.4 digambarkan *activity diagram* yang merupakan arus aktivitas admin dan user



Gambar 3.4 Menggambarkan *Activity diagram* modul *promotion* pada *website* Birds & Bees

D. Sequence Diagram

Sequence diagram dibuat bertujuan untuk menggambarkan interaksi antar objek yang satu dengan yang lainnya, dan perpindahan data dengan urutan yang sistematis.

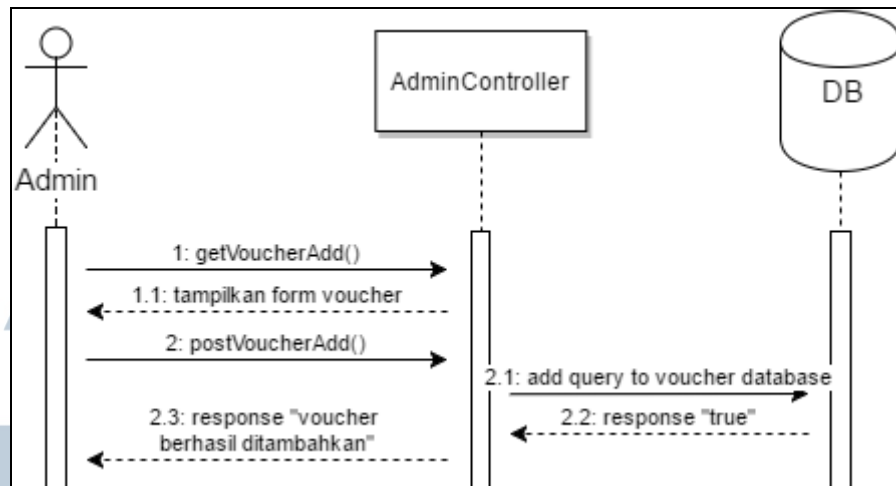


Gambar 3.5 Menggambarkan *sequence diagram* untuk melihat *list voucher* pada laman *admin*

Gambar merupakan *sequence diagram* untuk melihat *list voucher* pada laman *admin*. Skenario untuk *event* tersebut adalah:

1. *Admin* yang telah *login* akan menekan tombol *navigation* pada *sidebar* untuk masuk ke laman *voucher*.
2. *AdminController* akan mengambil data *voucher* dari *database* dan menampilkannya pada laman *admin*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



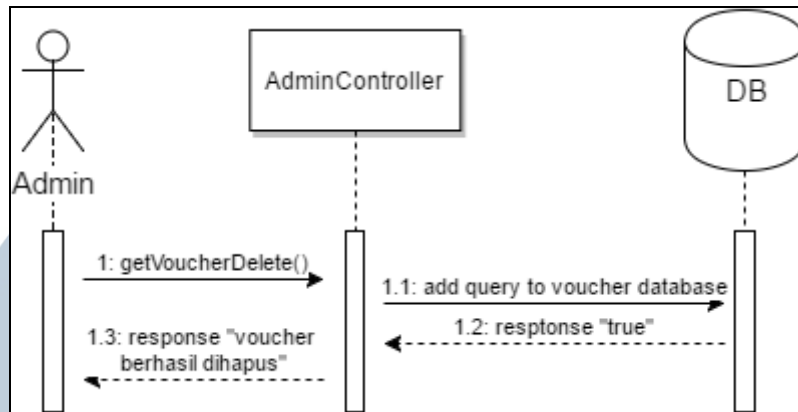
Gambar 3.6 Menggambarkan *sequence diagram* untuk membuat voucher baru

Gambar merupakan *sequence diagram* untuk membuat voucher baru.

Skenario untuk event tersebut adalah:

1. *Admin* menekan tombol *Add Voucher* untuk masuk ke laman *form* pembuatan *voucher* baru.
2. *AdminController* akan menampilkan form pembuatan *voucher* baru.
3. Setelah *admin* selesai mengisi form pembuatan *voucher* baru, menekan *submit* untuk melakukan penambahan *voucher* baru.
4. *AdminController* akan memasukkan data dari *form* yang telah diisi dan menambahkan *database* dan menampilkan bahwa *voucher* berhasil ditambahkan pada laman *admin*.

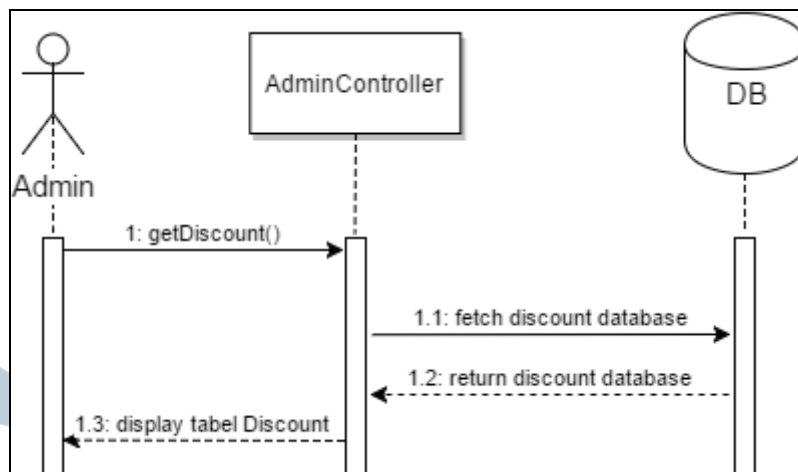
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.7 Menggambarkan *sequence diagram* untuk menghapus *voucher*

Gambar merupakan *sequence diagram* untuk menghapus *voucher*. Skenario untuk *event* tersebut adalah:

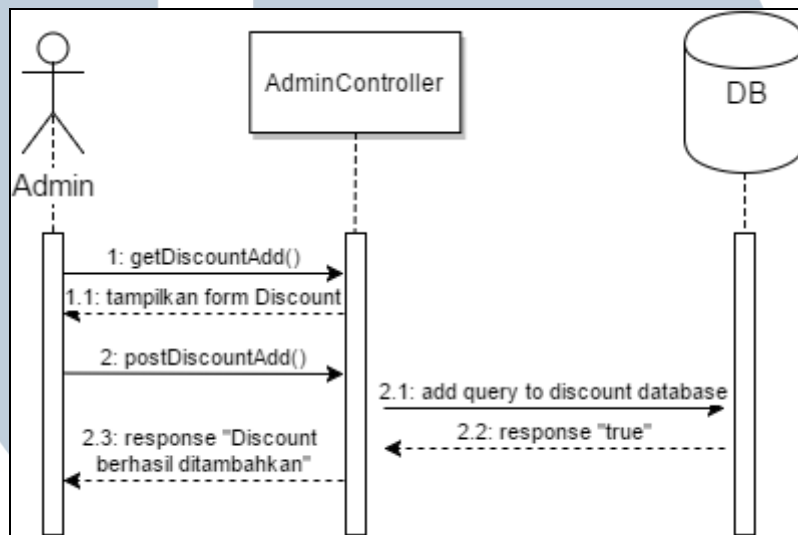
1. *Admin* menekan tombol *delete* pada kolom *action* untuk menghapus *voucher*.
2. *AdminController* akan menghapus data *voucher* dari *database* dan menampilkan *voucher* berhasil dihapus pada laman *admin*.



Gambar 3.8 Menggambarkan *sequence diagram* untuk melihat *list discount* pada laman *admin*

Gambar merupakan *sequence diagram* untuk melihat *list discount* pada laman *admin*. Skenario untuk *event* tersebut adalah:

1. *Admin* yang telah *login* akan menekan tombol *navigation* pada *sidebar* untuk masuk ke laman *discount*.
2. *AdminController* akan mengambil data *discount* dari *database* dan menampilkannya pada laman *admin*.

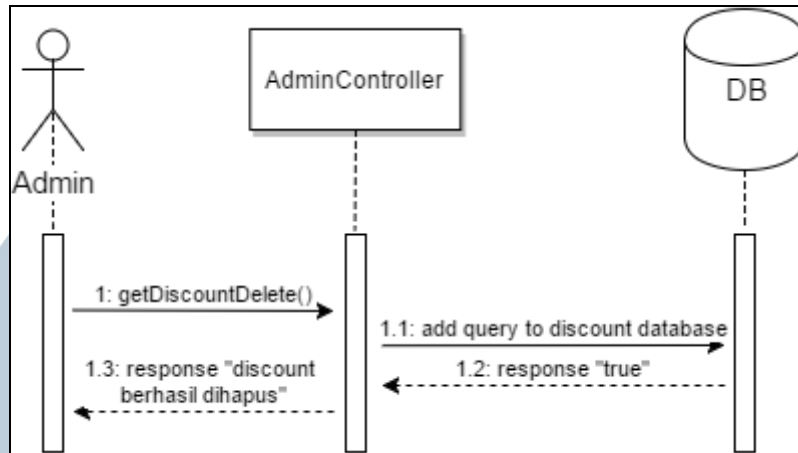


Gambar 3.9 Menggambarkan *sequence diagram* untuk membuat *discount* baru

Gambar merupakan *sequence diagram* untuk membuat *discount* baru.

Skenario untuk *event* tersebut adalah:

1. *Admin* tombol *Add Discount* untuk masuk ke laman *form* pembuatan *discount* baru.
2. *AdminController* akan menampilkan *form* pembuatan *discount* baru.
3. Setelah *admin* selesai mengisi *form* pembuatan *discount* baru, menekan *submit* untuk melakukan penambahan *discount* baru.
4. *AdminController* akan memasukkan data dari *form* yang telah diisi dan menambahkan *database* dan menampilkan bahwa *discount* berhasil ditambahkan pada laman *admin*.

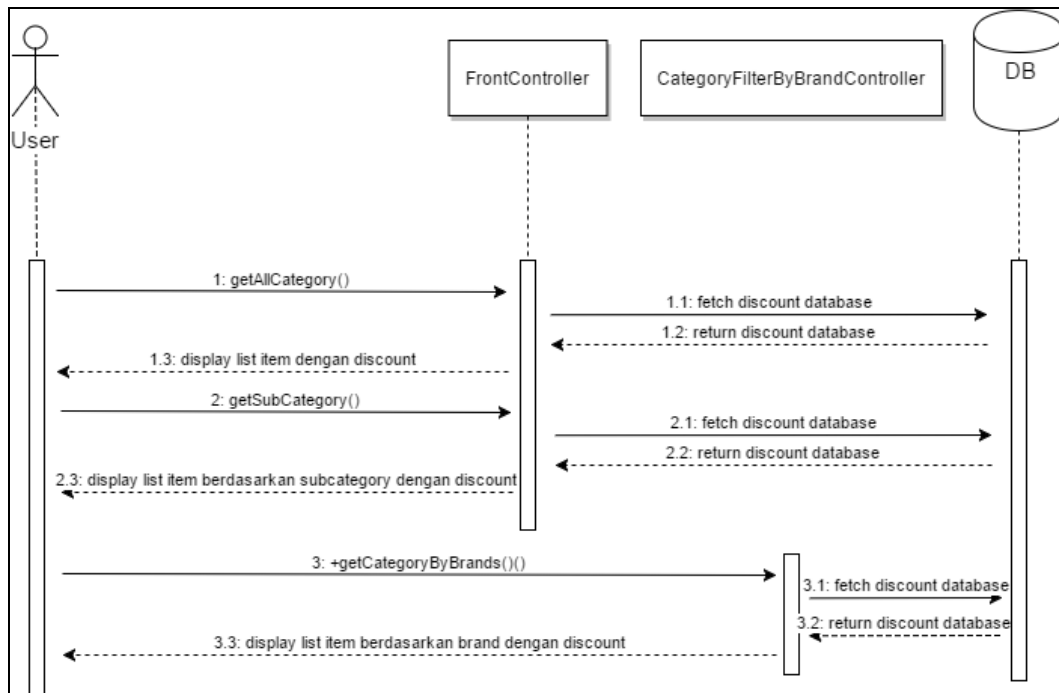


Gambar 3.10 Menggambarkan *sequence diagram* untuk menghapus *discount*

Gambar merupakan *sequence diagram* untuk menghapus *discount*. Skenario untuk *event* tersebut adalah:

1. Admin menekan tombol *delete* pada kolom *action* untuk menghapus *discount*.
2. AdminController akan menghapus data *discount* dari *database* dan menampilkan *discount* berhasil dihapus pada laman *admin*.

UMMN
 UNIVERSITAS
 MULTIMEDIA
 NUSANTARA

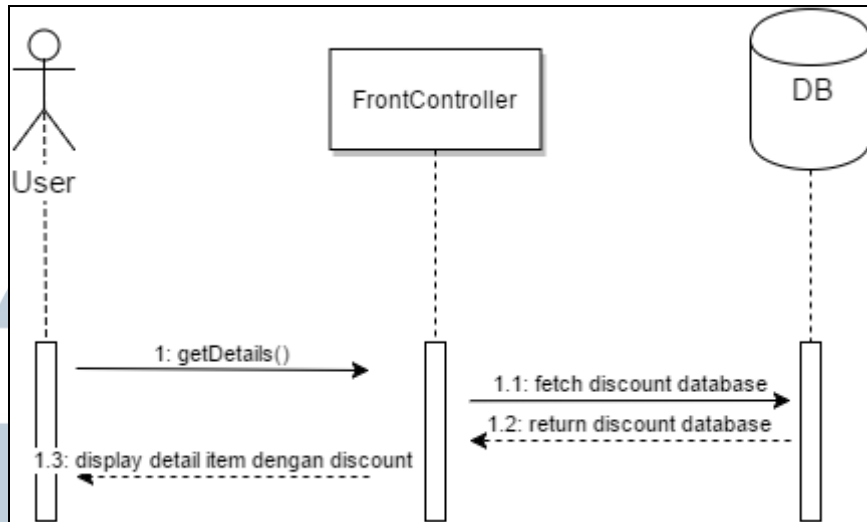


Gambar 3.11 Menggambarkan *sequence diagram* untuk melihat *discount item* pada *website*

Gambar merupakan *sequence diagram* untuk melihat *discount item* pada *website*.

Skenario untuk *event* tersebut adalah:

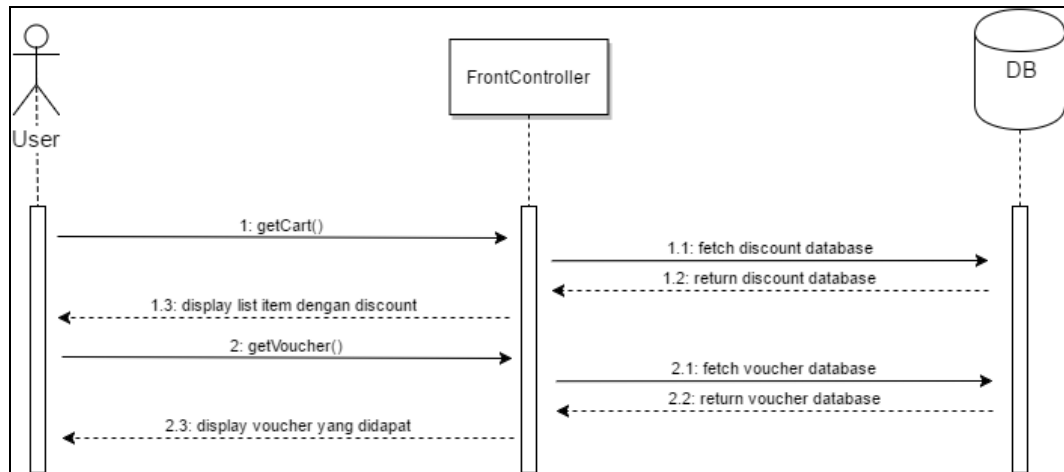
1. Pada laman utama *user* dapat memilih *category* pada menu *navigation* untuk mulai melihat *list item* yang diberikan *discount*.
2. *User* dapat menekan *subcategory* tertentu yang terdapat pada menu *navigation bar*, untuk melihat *list item* berdasarkan *category* yang diberikan *discount*.
3. *User* dapat menekan *category* tertentu yang terdapat pada menu *navigation bar*, untuk melihat *list item* berdasarkan *category* yang diberikan *discount*.
4. *User* dapat menekan *brand* tertentu yang terdapat pada menu *navigation bar*, untuk melihat *list item* berdasarkan *brand* yang diberikan *discount*.



Gambar 3.12 Menggambarkan *sequence diagram* untuk melihat *detail discount item*

Gambar merupakan *sequence diagram* untuk melihat *detail item*. Skenario untuk *event* tersebut adalah:

1. User dapat memilih *detail item* yang diberikan *discount* dari *list* tertentu.



Gambar 3.13 Menggambarkan *sequence diagram* untuk melihat *discount* yang didapat pada *shopping cart*, menggunakan *voucher*, dan menampilkan *voucher* yang didapat

Gambar merupakan *sequence diagram* untuk melihat *discount* yang didapat pada *shopping cart*, menggunakan *voucher*, dan menampilkan *voucher* yang didapat. Skenario untuk *event* tersebut adalah:

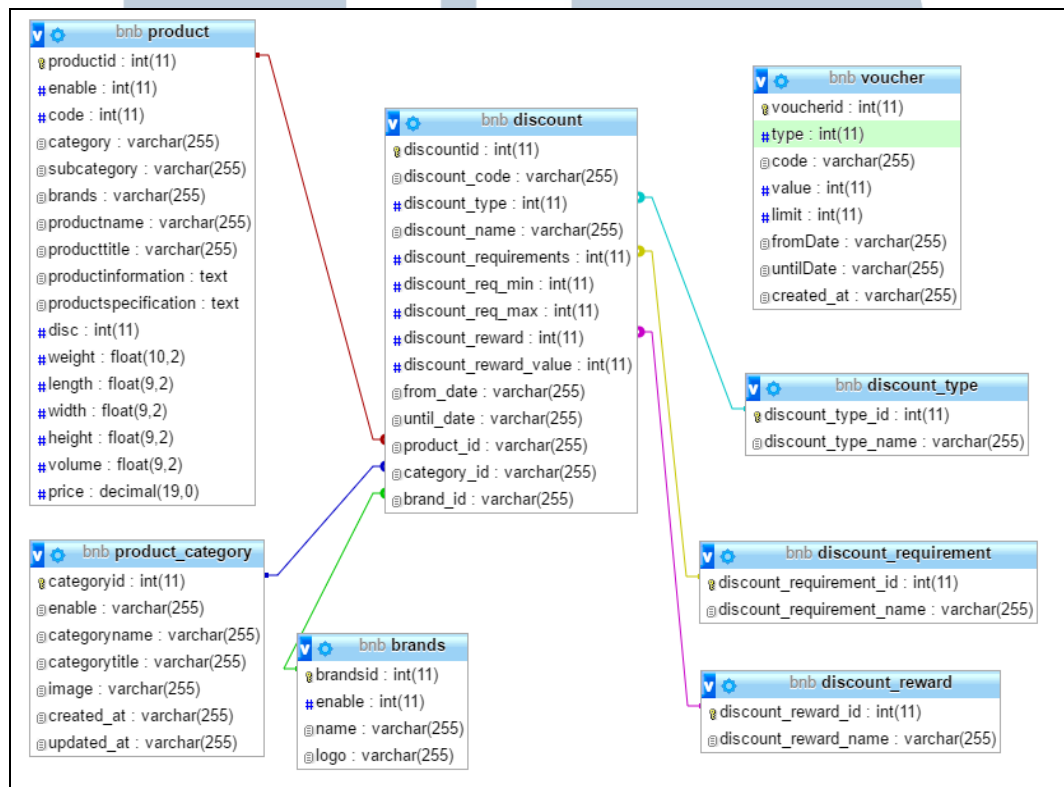
1. User dapat mengklik tombol *shopping cart* untuk melihat *item* yang telah dipilih beserta keterangan *discount* yang didapatkan.
2. User dapat memilih untuk memasukkan kode *voucher* pada kotak yang telah disediakan dan menekan *apply*.
3. FrontController akan mengecek keabsahan kode *voucher* tersebut.
4. User akan melihat *alert* bahwa telah mendapatkan *voucher*.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

E. Database Schema

Dalam pembuatan modul *promotion* website Birds & Bees terdapat 8 tabel *database* yang digunakan untuk menyimpan setiap data yang dibutuhkan untuk modul tersebut. Berikut digambarkan struktur dari masing-masing tabel beserta relasinya. Tabel *product*, *product_category*, dan *brands* merupakan foreign table.



Gambar 3.14 Gambaran Database Schema modul *promotion*

Tabel dibawah merupakan penjelasan masing-masing table yang digunakan.

1. Nama Tabel : brands

Fungsi : Menyimpan data *brand*

Primary Key : brandsid

Foreign Key : -

Tabel 3.10 Tabel struktur brands

Column	Type	Description
brandsid	int (11)	<i>Primary</i>
enable	int (11)	
name	varchar (255)	
logo	varchar (255)	

2. Nama Tabel : discount

Fungsi : Menyimpan data *discount*

Primary Key : discountid

Foreign Key : discount_type, discount_requirements,
discount_reward, product_id, category_id, dan brand_id

Tabel 3.11 Tabel struktur discount

Column	Type	Description
discountid	int (11)	<i>Primary</i>
discount_code	varchar (255)	
discount_type	int (11)	discount_type_id
discount_nama	varchar (255)	
discount_requirements	int (11)	discount_requirements_id
discount_req_min	int (11)	
discount_req_max	int (11)	
discount_reward	int (11)	
discount_reward_value	int (11)	
from_date	varchar (255)	
until_date	varchar (255)	
product_id	varchar (255)	productid
category_id	varchar (255)	categoryid
brand_id	varchar (255)	brandsid

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

3. Nama Tabel : discount_requirement
- Fungsi : Menyimpan data *requirement discount*
- Primary Key : discount_requirement_id
- Foreign Key : -

Tabel 3.12 Tabel struktur discount requirement

Column	Type	Description
discount_requirement_id	int (11)	<i>Primary</i>
discount_requirement_name	varchar (255)	

4. Nama Tabel : discount_reward
- Fungsi : Menyimpan data *reward discount*
- Primary Key : discount_reward_id
- Foreign Key : -

Tabel 3.13 Tabel struktur discount reward

Column	Type	Description
discount_reward_id	int (11)	<i>Primary</i>
discount_reward_name	varchar (255)	

5. Nama Tabel : discount_type
- Fungsi : Menyimpan data tipe discount
- Primary Key : discount_type_id
- Foreign Key : -

Tabel 3.14 Tabel struktur discount type

Column	Type	Description
discount_type_id	int (11)	<i>Primary</i>
discount_type_name	varchar (255)	

6. Nama Tabel : product
- Fungsi : Menyimpan data *product*
- Primary Key : productid
- Foreign Key : -

Tabel 3.15 Tabel struktur product

Column	Type	Description
productid	int (11)	<i>Primary</i>
enable	int (11)	
code	int (11)	
category	varchar (255)	
subcategory	varchar (255)	
brands	varchar (255)	
productname	varchar (255)	
producttitle	varchar (255)	
productinformation	text	
productspecification	text	
disc	int (11)	
weight	float (10,2)	
length	float (9,2)	
width	float (9,2)	
height	float (9,2)	
volume	float (9,2)	
price	decimal (19,0)	

7. Nama Tabel : product_category
- Fungsi : Menyimpan data *category product*
- Primary Key : categoryid
- Foreign Key : -

Tabel 3.16 Tabel struktur category

Column	Type	Description
categoryid(<i>Primary</i>)	int (11)	<i>Primary</i>
enable	varchar (255)	
categoryname	varchar (255)	
categorytitle	varchar (255)	
image	varchar (255)	
created_at	varchar (255)	
updated_at	varchar (255)	

8. Nama Tabel : voucher
- Fungsi : Menyimpan data *voucher*
- Primary Key : voucherid
- Foreign Key : -

Tabel 3.17 Tabel struktur voucher

Column	Type	Description
voucherid(<i>Primary</i>)	int (11)	<i>Primary</i>
type	int (11)	
code	varchar (255)	
value	int (11)	
limit	int (11)	
fromDate	varchar (255)	
untilDate	varchar (255)	
created_at	varchar (255)	

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA

F. Implementasi

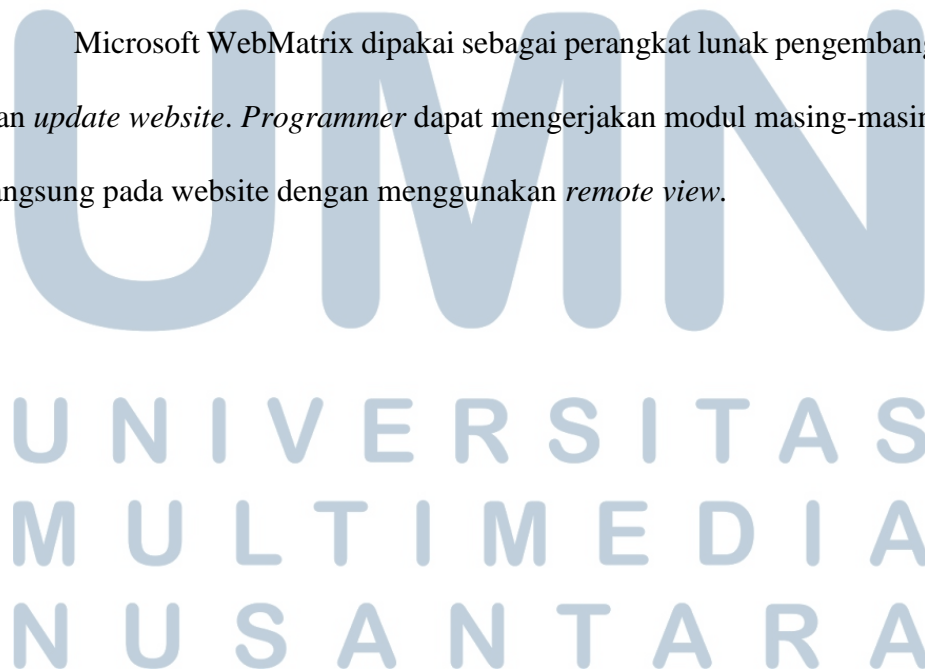
Dalam pengembangan modul *promotion website marketplace* Birds & Bees, digunakan perangkat lunak dan perangkat keras tertentu. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut.

1. Manufaktur : DELL
2. Model : Inspiron 3421
3. Prosesor : Intel® Core™ i3-3217U
4. Memori : 6GB RAM
5. Harddisk : 300GB

Sedangkan spesifikasi perangkat lunak yang digunakan dalam pengembangan aplikasi adalah sebagai berikut.

1. Sistem Operasi Windows 10 64-bit
2. Microsoft Webtrix v2
3. Google Chrome

Microsoft WebMatrix dipakai sebagai perangkat lunak pengembangan kode dan *update website*. Programmer dapat mengerjakan modul masing-masing secara langsung pada website dengan menggunakan *remote view*.



Publish Settings

Protocol:

Server:

Site path:

User name:

Password:

Destination URL:

Save password

Common Tasks

[Import publish settings](#)

Gambar 3.15 Menggambarkan *setting* yang dipakai untuk melakukan koneksi dengan *server* Birds & Bees

Laravel memiliki pengaturan *routes* khusus untuk menentukan bagaimana link yang dimasukkan dapat mengakses *controller* dengan *class* tertentu. File *routes.php* berisi *routes* beserta *controller*, fungsi yang akan dipanggil dan jenis HTTP Request Method yang digunakan.

File-file *controller* diletakkan pada *folder* *app*. *Controller* berguna untuk mengatur fungsi akan dijalankan dan *view* yang akan dipanggil sesuai dengan *routes*. *Controller-Controller* yang dibuat merupakan turunan dari class *Controller*.

```

routes.php* x voucher-add.blade.php voucher.blade.php product-category.blade.php FrontController.php*
105 Route::get('admin/voucher/view', 'AdminController@postvoucherview');
190 Route::post('admin/voucher/view', 'AdminController@postvoucherview');
191
192 /*Discount*/
193 Route::get('admin/discount', 'AdminController@getDiscount');
194 Route::get('admin/discount/add', 'AdminController@getDiscountAdd');
195 Route::post('admin/discount/add', 'AdminController@postDiscountAdd');
196 Route::get('admin/discount/delete/{cateid}', 'AdminController@getcategorydelete');
197
198 /*Slider*/
199 Route::get('admin/slider', 'AdminController@getslider');
200 Route::get('admin/slider/add', 'AdminController@getslideradd');
201 Route::post('admin/slider/add', 'AdminController@postslideradd');
202 Route::get('admin/slider/delete/{sliderid}', 'AdminController@getsliderdelete');
203 Route::get('admin/slider/view/{sliderid}', 'AdminController@getsliderview');
204 Route::post('admin/slider/view', 'AdminController@postsliderview');
205
206
207 /*Promotion Others */
208 Route::get('admin/others', 'AdminController@getothers');
209 Route::post('admin/others', 'AdminController@postothers');
210
211 /*Category Page*/
212 Route::get('admin/category', 'AdminController@getcategory');
213 Route::get('admin/category/add', 'AdminController@getcategoryadd');
214 Route::post('admin/category/add', 'AdminController@postcategoryadd');
215 Route::get('admin/category/delete/{cateid}', 'AdminController@getcategorydelete');
216 Route::get('admin/category/view/{cateid}', 'AdminController@getcategoryview');
217 Route::post('admin/category/view/{cateid}', 'AdminController@postcategoryview');
218
219
220 /*Subscribe*/
221 Route::get('admin/subscriber', 'AdminController@getsubscriber');

```

Gambar 3.16 Contoh routes.php

```

AdminController.php x routes.php* voucher-add.blade.php
1091
1092
1093 public function getvoucher()
1094 {
1095
1096     if(Session::get('sessionadmin')){
1097         //Voucher View
1098         $voucher = DB::table('voucher')->get();
1099         return view('admin/voucher',
1100             [
1101                 'voucher'=> $voucher,
1102             ]);
1103     }else{
1104         return redirect('admin/login');
1105     }
1106 }
1107
1108 }
1109
1110

```

Gambar 3.17 Contoh class AdminController

View pada Laravel diletakkan pada *folder* resources/view. *File view* pada Laravel selain mengatur tampilan dengan html juga memiliki keistimewaan “Blade Templates” yaitu dapat dimodifikasi sehingga dapat menerima kode PHP tanpa perlu menuliskan tag PHP untuk terbaca sebagai PHP.



```
AdminController.php routes.php* voucher-add.blade.php voucher.blade.php x product-  
1 @extends('admin/app')  
2 @section('content')  
3  
4 <!-- Content Wrapper. Contains page content -->  
5 <div class="content-wrapper">  
6 <!-- Content Header (Page header) -->  
7 <section class="content-header">  
8 <h1>  
9 Voucher  
10 <!--<small>advanced tables</small-->  
11 </h1>  
12 <ol class="breadcrumb">  
13 <li><a href="#"><i class="fa fa-dashboard"></i> Dashboard</a></li>  
14 <li class="active">Voucher</li>  
15 </ol>  
16 </section>  
17  
18 <!-- Main content -->  
19 <section class="content">  
20 <div class="row">  
21 <div class="col-xs-12">  
22 <div class="box">  
23 <div class="box-header">  
24 <!--<h3 class="box-title">Data Table With Full Features</h3-->  
25 </div><!-- /.box-header -->  
26 <div class="box-body">  
27 @if(Session::has('success-create'))  
28 <div class="alert alert-success alert-dismissable">  
29 <button type="button" class="close" data-dismiss="alert" aria-hid  
30 <h4> <i class="icon fa fa-check"></i> Alert!</h4>  
31 {{ Session::get('success-create') }}  
32 </div>
```

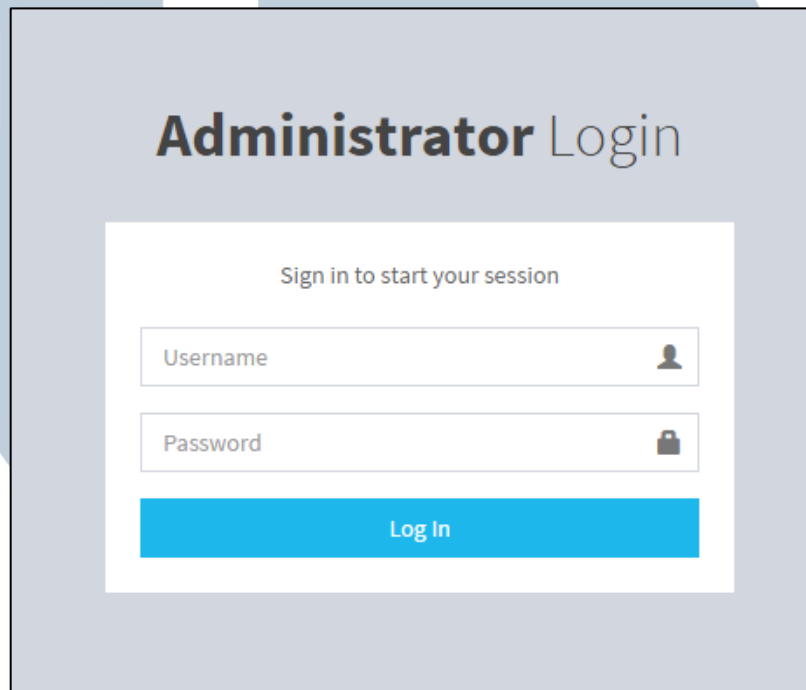
Gambar 3.18 Contoh View voucher.blade.php

G. Screenshot

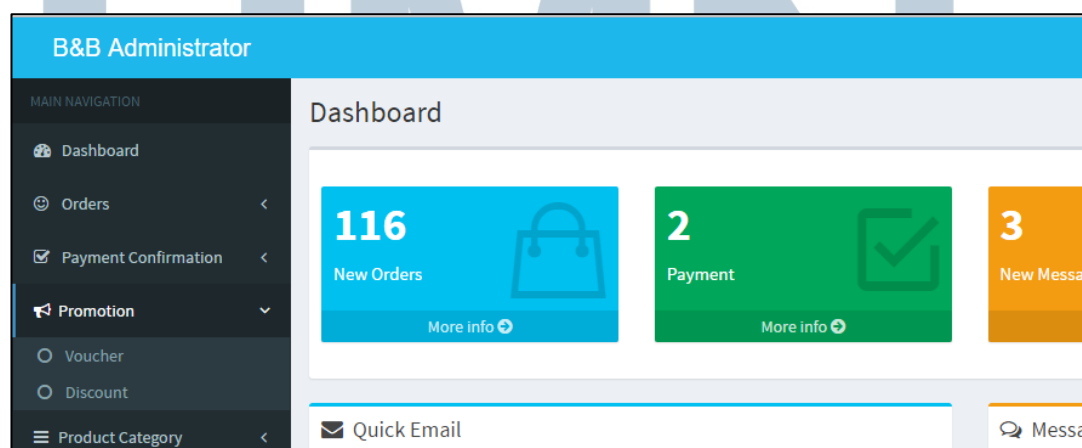
Hasil yang dapat dilaporkan dari *screenshot* adalah hasil pembuatan kode program pada proses implementasi dalam bentuk *User Interface* yang telah disetujui oleh *programmer*.

a. Laman Admin

Laman *Admin* dapat diakses dengan memasukkan `/admin/login` kedalam *website* *Birds & bees*, setelah memasukkan *username*, *password* dan mengklik tombol *log in*, *admin* dapat langsung mengakses laman *promotion voucher* dan *discount* pada *sidebar Promotion*.



Gambar 3.19 Screenshot Laman login admin



Gambar 3.20 Screenshot Laman admin

Setelah mengklik tombol *voucher*, *admin* akan masuk ke laman yang berisi *list voucher* yang telah dibuat sebelumnya. Pada laman ini *admin* dapat mengklik tombol *add voucher* untuk menambahkan *voucher* baru, atau mengklik tombol *delete* untuk menghapus suatu *voucher*.

ID	Code	Type	Limit	Value	Valid From	Valid Until	Created at	Action
3	95868	Percent (%)	1	20	07/04/2016	07/07/2016	2016-07-21 14:53:12	Delete
4	60754	Percent (%)	12	132	08/07/2016	08/16/2016	2016-08-09 09:56:11	Delete

Gambar 3.21 Screenshot Laman voucher

Dengan menekan tombol *Add Voucher*, *admin* akan masuk ke laman pengisian *form* untuk pembuatan *voucher* baru, pada laman ini terdapat kolom *type* untuk memilih bentuk potongan *voucher* apakah persen atau nominal, terdapat kolom *limit* untuk membatasi jumlah voucher yang akan dibuat oleh *admin*, kolom *value* diisi berdasarkan nominal atau nilai persentase yang akan dipotong dari total belanja *user*, dan *validity* yang merupakan range tanggal aktif *voucher* tersebut. Setelah menekan tombol *submit*, *admin* akan mendapatkan *alert* bahwa *voucher* baru telah ditambahkan. *Admin* juga dapat menghapus *voucher* yang diinginkan dengan menekan tombol *delete* yang tersedia pada kolom *action*.

Add Voucher

Type

Limit

Value

validity -

Gambar 3.22 Screenshot Laman form penambahan voucher

Alert!
Success added New Voucher

Show entries Search:

ID	Code	Type	Limit	Value	Valid From	Valid Until	Created at	Action
3	95868	Percent (%)	1	20	07/04/2016	07/07/2016	2016-07-21 14:53:12	Delete
4	60754	Percent (%)	12	132	08/07/2016	08/16/2016	2016-08-09 09:56:11	Delete
6	29576	Percent (%)	5	25	12/27/2016	12/30/2016	2016-12-22 16:56:36	Delete

Showing 1 to 3 of 3 entries

Gambar 3.23 Screenshot alert penambahan voucher

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

Alert! Success deleting data

+ Add Voucher

Show 10 entries Search:

ID	Code	Type	Limit	Value	Valid From	Valid Until	Created at	Action
3	95868	Percent (%)	1	20	07/04/2016	07/07/2016	2016-07-21 14:53:12	Delete
4	60754	Percent (%)	12	132	08/07/2016	08/16/2016	2016-08-09 09:56:11	Delete

Showing 1 to 2 of 2 entries Previous 1 Next

Gambar 3.24 Screenshot alert penghapusan voucher

Kurang lebih sama dengan voucher, Setelah mengklik tombol *discount*, admin akan masuk ke laman yang berisi *list discount* yang telah dibuat sebelumnya. Pada laman ini admin dapat mengklik tombol *add discount* untuk menambahkan *discount* baru, atau mengklik tombol *delete* untuk menghapus suatu *discount*.

Discount Dashboard > Discount

+ Add Discount

Show 10 entries Search:

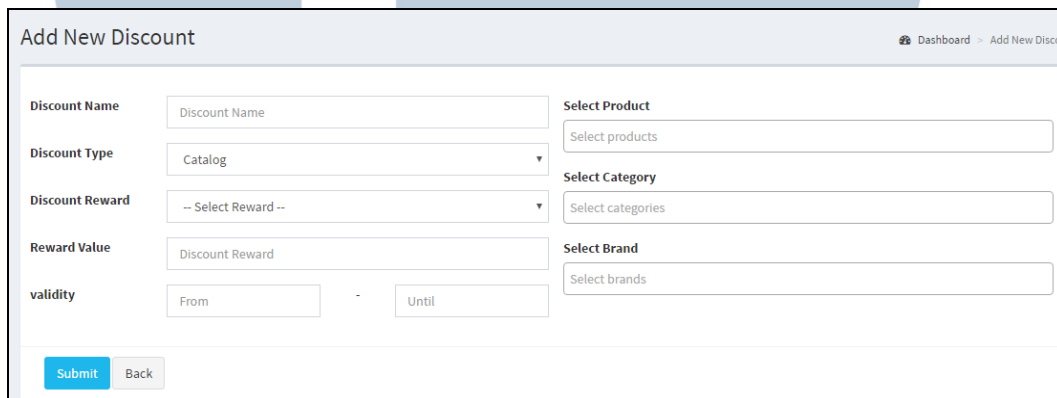
Id	Code	Type	Name	Requirement	Reward	Validation	Product	Category	Brand	Action
14	14015	Cart	hahaha	2 - 3 pcs	12 %	07/13/2018 - 10/28/2016	28, 29, 30	1, 2, 3	2, 3	Delete
18	46751	Catalog	diskon lebaran	-	25 %	07/19/2019 - 12/27/2016	28, 29, 30, 391, 394, 446	6	2	Delete
19	95493	Catalog	lalalala	-	50 %	08/15/2018 - 10/29/2017	29, 32	1	3	Delete
20	65977	Cart	diskon natal	200000 IDR - 400000 IDR	- 50000 IDR	08/09/2018 - 10/25/2017	28	1	3	Delete

Showing 1 to 4 of 4 entries Previous 1 Next

Gambar 3.25 Screenshot Laman discount

NUSANTARA

Dengan menekan tombol *Add discount*, *admin* akan masuk ke laman pengisian *form* untuk pembuatan *discount* baru, pada laman ini terdapat kolom *Discount Name* untuk mengisi nama *discount*, *discount type* untuk memilih jenis katalog, *discount reward* untuk memilih bentuk potongan *voucher* apakah persen atau nominal, kolom *Reward Value* diisi berdasarkan nominal atau nilai persentase yang akan dipotong dari harga *item*, dan *Validity* yang merupakan *range* tanggal aktif *discount* tersebut.



The screenshot shows a web form titled "Add New Discount". The form is organized into two columns. The left column contains: "Discount Name" (text input), "Discount Type" (dropdown menu with "Catalog" selected), "Discount Reward" (dropdown menu with "-- Select Reward --" selected), "Reward Value" (text input), and "validity" (two text inputs labeled "From" and "Until"). The right column contains: "Select Product" (dropdown menu with "Select products" selected), "Select Category" (dropdown menu with "Select categories" selected), and "Select Brand" (dropdown menu with "Select brands" selected). At the bottom left, there are two buttons: "Submit" (blue) and "Back" (grey). The top right corner shows a breadcrumb trail: "Dashboard > Add New Disco".

Gambar 3.26 Screenshot Laman form penambahan *discount*

Jenis katalog dapat dipilih antara *catalog discount* untuk *discount item* pada *website*, atau *cart discount* untuk *discount* berdasarkan item yang sudah ada pada *shopping cart*. Bila memilih *cart discount* akan menampilkan baris form baru yaitu *discount requirement* untuk menentukan prasyarat *discount* dengan *quantity* (kuantitas barang yang dibeli) atau *total shopping cart nominal* (total pembelian pada *shopping cart*), dan *requirement value* untuk menentukan *range* kuantitas atau nominal total pembelian yang akan mendapatkan *discount*.

Discount Type: Cart

Discount Requirement: -- Select Requirement --

Requirement Value: From [] - [] Until []

Gambar 3.27 Screenshot Laman baris form tambahan disaat *cart* dipilih sebagai *discount type*

Pada kolom sebelah kanan terdapat *Select Product*, *Select Category*, dan *Select Brand* yang digunakan untuk memilih produk, *category*, atau *brand* apa yang akan diberikan *discount*, pemilihannya dapat dilakukan langsung di tiga tempat tersebut dengan jumlah lebih dari 1. Setelah menekan tombol *submit*, *admin* akan mendapatkan *alert* bahwa *discount* baru telah ditambahkan. *Admin* juga dapat menghapus *discount* yang diinginkan dengan menekan tombol *delete* yang tersedia pada kolom *action*

Select Product

- Oops My Bathtime Friends - Forest
- Crocodile Creek Insulated Food Jar Butterfly

Select Category

- Nursing & Feeding

Select Brand

- ALZIPMAT
- 3 SPROUTS
- ALZIPMAT
- ANGLEBAY
- ATTIPAS
- AVENT
- BABA SLING

Gambar 3.28 Screenshot pemilihan product, *category*, dan *brand* yang ingin diberikan *discount*

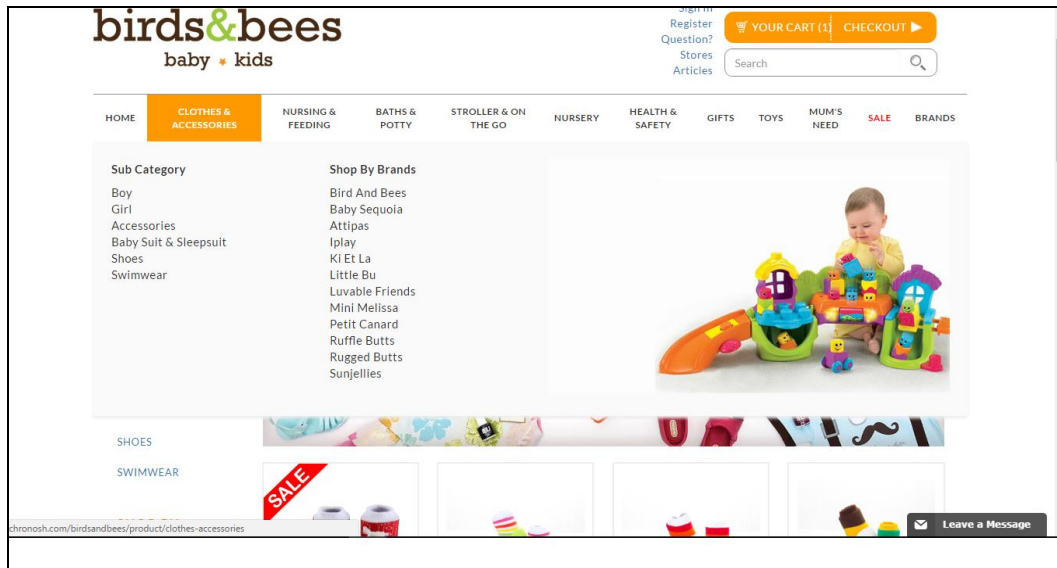
b. Laman *User*

Saat pertama kali *user* masuk ke *website* Birds & Bees akan masuk ke laman utama, dari laman utama dapat memilih salah satu *category item* yang terdapat pada menu *navigation*. Saat menu *navigation* dipilih, akan masuk ke laman *list item* berdasarkan *category*. Jika menekan salah satu menu *dropdown* yang terdapat pada menu *navigation* dengan tag *brand*, akan masuk ke laman *list item* berdasarkan *brand*. Jika menekan salah satu menu *dropdown* yang terdapat pada menu *navigation* dengan tag *subcategory*, akan masuk ke laman *list item* berdasarkan *subcategory*.

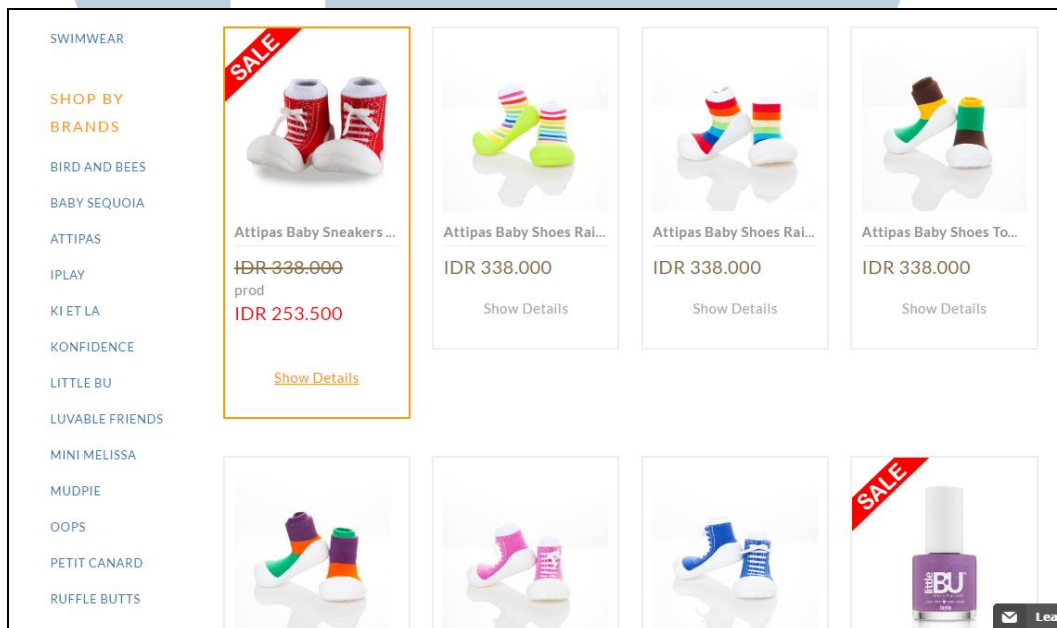


Gambar 3.29 Screenshot laman utama *website* Birds & Bees

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



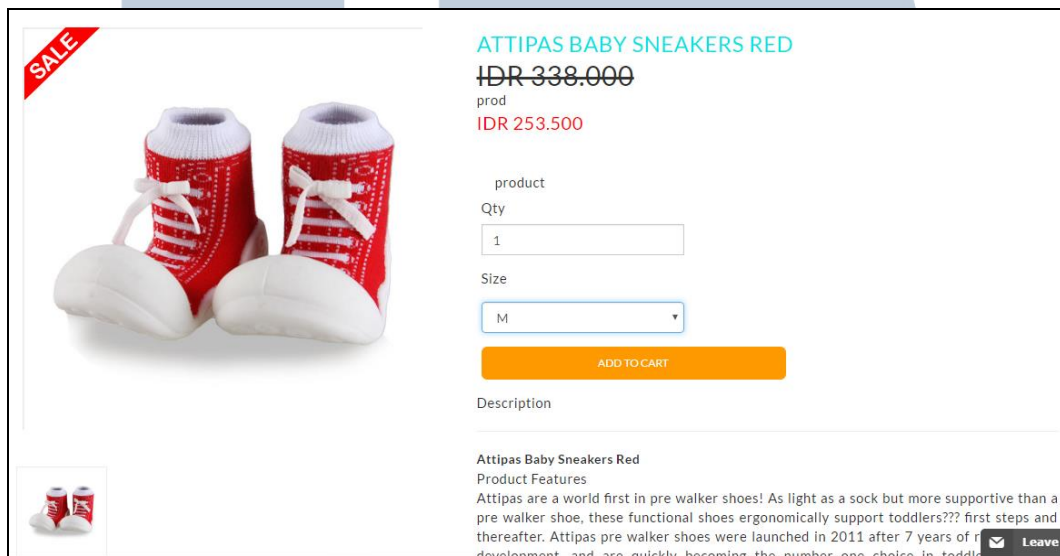
Gambar 3.30 Screenshot menu navigasi website Birds & Bees



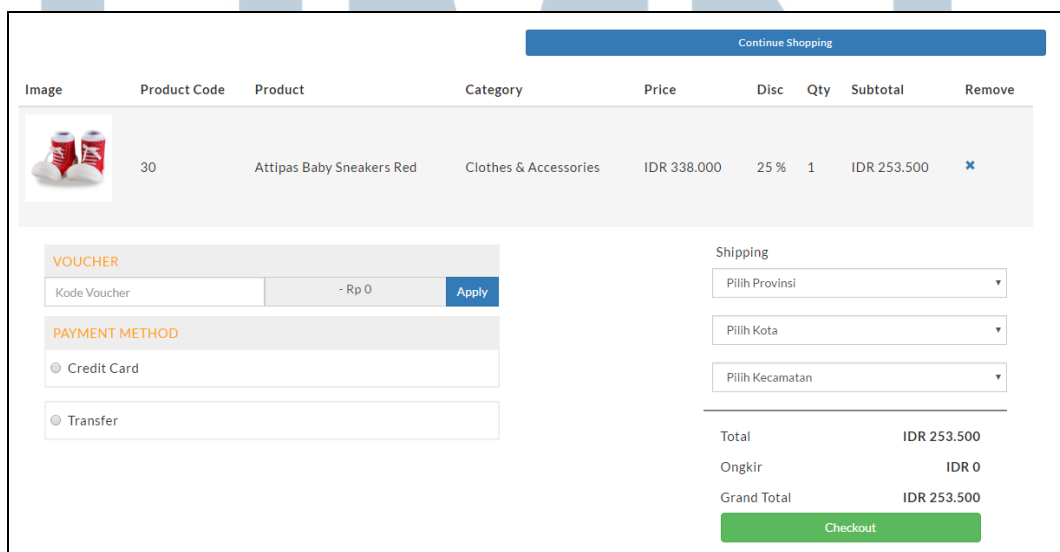
Gambar 3.31 Screenshot list item berdasarkan category clothes & accessories pada website Birds & Bees

Dengan memilih salah satu *item* yang terdapat pada *list*, *user* akan masuk ke laman *detail item*, dimana *user* dapat melihat *discount* yang didapat dan langsung memasukkannya ke *shopping cart* dengan menekan tombol *add to cart*. Setelah memilih *item* yang diberikan *discount*, maka

user dapat masuk ke laman *shopping cart* untuk melihat *discount* yang didapat, dan dapat memasukkan kode *voucher* untuk mendapatkan potongan harga dari total belanja. Setelah benar memasukkan kode *voucher*, akan muncul box alert dan memberitahukan bahwa user telah mendapatkan potongan harga.




Gambar 3.32 Screenshot contoh laman *detail item*



Gambar 3.33 Screenshot laman *shopping cart*

Selamat anda mendapatkan potongan harga dari kami !

Image	Product Code	Product	Category
	30	Attipas Baby Sneakers Red	Clothes & Ac

VOUCHER

95868	- Rp 50700 (20%)	Apply
-------	--------------------	--------------

PAYMENT METHOD

Credit Card

Transfer

Gambar 3.34 Screenshot input kode voucher

3.3.3 Kendala yang Ditemukan

Kendala yang ditemukan dalam pembuatan modul *Promotion* pada website Birds & Bees ini adalah sebagai berikut.

- Kurangnya informasi yang bisa diambil pada saat penafsiran *user requirement*, sehingga pada saat pengerjaan harus bertanya kembali kepada pembimbing magang.
- Sering terjadi perubahan desain dan logika dalam fungsionalitas modul *promotion* sehingga harus dilakukan pergantian logika pemrograman.

- Penggunaan Framework PHP Laravel yang belum pernah dipelajari sebelumnya

3.3.4 Solusi atas Kendala yang Ditemukan

Solusi atas kendala yang ditemukan ketika menjalankan kerja magang adalah sebagai berikut.

- Mendiskusikan dengan pembimbing magang langkah apa yang harus diambil untuk sementara waktu.
- Mendiskusikan dengan pembimbing magang mengenai kepastian desain dan alur logika dari fungsionalitas modul *promotion*.
- Mempelajari laravel dari website, dan sering bertanya kepada pembimbing magang.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA