**BAB IV**

**IMPLEMENTASI**

Implementasi adalah pelaksanaan rencana yang telah disusun sedemikian rupa dengan disertai dengan penerapan terhadap suatu sistem dalam sebuah pembuatan program. Pada bab ini akan menjelaskan tentang setiap bagian program yang terdapat pada masing-masing halaman website yang dibuat. Penjelasan segmen program masing-masing halaman dibagi sesuai dengan kegunaan dan fungsi nya, dibagi menjadi 3 yaitu user, merchant, dan admin. Ketiga bagian tersebut akan dijelaskan lebih terperinci agar dapat dipahami dengan mudah.

Gather.owl menggunakan bahasa pemrograman yang berbasis *framework.* CodeIgniter merupakan salah satu framework yang menggunakan bahasa pemrograman PHP: Hypertext Preprocessor(PHP) yang dideklarasikan secara mendasar dengan cara yang singkat dan mudah. Penyimpanan yang digunakan dalam website Gather.owl yaitu PHPMyAdmin. Website gather.owl menggunakan PHPMyAdmin karena basis dari pemrograman yang digunakan bahasa pemrograman PHP yang memiliki kelebihan tersendiri yaitu memiliki sistem penyimpanan data dalam bentuk MySQL.

Penggunaan CodeIgniter diperlukan beberapa proses yang harus dijalankan untuk dapat mengaktifkan CodeIgniter didalam program. Langkah-langkah ini diperlukan agar pada saat melakukan inisialisasi dan coding dapat berjalan dengan benar sesuai dengan framework yang digunakan. Ada 4 langkah yang harus dijalankan agar CodeIgniter dapat berjalan dalam program. Berikut langkah-langkah yang dilakukan.

1. Mengunduh seluruh berkas-berkas yang dibutuhkan untuk mengakses CodeIgniter dalam code program. Berkas-berkas yang dibutuhkan untuk mengaktifkan CodeIgniter dapat diunduh di website milik CodeIgniter. Berkas-berkas tersebut disediakan dengan ukuran yang kecil, yaitu 2.6 *Mega Byte*(MB). Berkas yang terunduh akan berbentuk berkas .*zip*.
2. Mengekstrak berkas yang ada di .zip dalam sebuah *folder.*
3. Mengunggah folder dan file CodeIgniter ke dalam server. Bagian file index.php yang akan berada di root server.
4. Membuka file aplikasi, kemudian pada bagian folder config, kemudian buka config.php dengan editor teks dan atur URL dasar dari program yang dibuat.

Setelah semua langkah dilakukan, ada beberapa hal yang dilakukan untuk dapat mengaktifkan media penyimpanan yang digunakan dalam CodeIgniter. Hal-hal tersebut dilakukan agar seluruh pengerjaan dari program dapat tersimpan dalam media penyimpanan sesuai dengan yang telah dibuat. Ada 4 langkah yang dilakukan untuk mengaktifkan media penyimpanan. Berikut langkah-langkah yang dilakukan.

1. Buka folder dari CodeIgniter pada bagian config, pada bagian *autoload.php* terdapat bagian$autoload['libraries'], ganti isi dari array() menjadi array('database'). Jika memerlukan *session* di dalam program, dapat menambahkan 'session' setelah penulisan 'database'. Jika ditambah dengan session, penulisan dalam editor teks tersebut maka menjadi array('database', 'session').
2. Buka bagian *database.php* dalam editor teks. Pada bagian $db['default'], isi 'hostname' yang digunakan dalam media penyimpanan. Media penyimpanan PHPMyAdmin menggunakan hostname 'localhost'.
3. Isi bagian 'username' yang akan digunakan dalam program. Nama username yang digunakan dalam media penyimpanan PHPMyAdmin adalah 'root'. Pada bagian 'password', isi dengan password yang digunakan dalam media penyimpanan. Dalam media penyimpanan untuk program Gather.owl, password akan dikosongkan karena pada keadaan standar bagian password adalah kosong.
4. Isi bagian 'database' dengan media penyimpanan yang akan digunakan dalam program. Nama *database* yang digunakan dalam media penyimpanan PHPMyAdmin untuk program gather.owl adalah 'db\_owl'.

Pada bab ini akan menjelaskan setiap potongan program yang tercantum dalam website gather.owl. Setiap akan masuk ke dalam website gather.owl, pengguna yang ingin mengakses website memerlukan akun agar dapat menikmati fitur-fitur yang ada dalam website. Oleh karena itu, terdapat sistem registrasi akun untuk setiap pengguna yang mengakses website. Berikut merupakan potongan program tahap registrasi secara umum.

Segmen Program 4.1 : Controller Registrasi User

1. $this->form\_validation
2. >set\_rules('regUsername','Username',
3. 'callback\_username\_check');
4. $this->form\_validation->set\_rules('regName', 'Nama',
5. 'required');
6. $this->form\_validation->set\_rules('regEmail','Email',
7. 'callback\_username\_check');
8. $this->form\_validation->set\_rules('regEmail','Email',
9. 'required|valid\_email');
10. $this->form\_validation
11. >set\_rules('regPass','Password',
12. 'required');
13. $this->form\_validation->set\_rules('regCPass','Password
14. Confirmation', 'required|matches[regPass]');
15. $this->form\_validation->set\_rules('regPhone','Nomor
16. HP','required');
17. $this->form\_validation->set\_rules('regPhone','Nomor
18. HP','callback\_username\_check');
19. if ($this->form\_validation->run() == FALSE){
20. $this->load->view('templates/header');
21. $this->load->view('register');
22. } else {
23. $this->User\_model->insertUser();
24. $id = $this->session->userdata('user');
25. redirect('Email/verifikasi/' . $id);
26. }

Segmen program 4.1 merupakan segmen program bentuk controller yang berfungsi untuk registrasi pengguna ke dalam website. Baris 1 hingga 18 menunjukkan potongan program dimana data yang dimasukkan oleh pengguna tersebut *valid* atau tidak. Selain memeriksa apakah valid atau tidak, bagian program tersebut memeriksa tempat dimana pengguna mengisi data yang disediakan telah diisi oleh pengguna atau masih kosong. Baris 19 hingga 26 merupakan bagian program yang berfungsi untuk memeriksa benar salah dari baris 1 hingga 18. Baris 20 dan 21 menunjukkan bahwa jika hasil dari pengecekan validasi adalah salah, maka pengguna tetap pada halaman register. Baris 23 menunjukkan jika hasil dari pengecekan validasi adalah benar, maka data yang dikirimkan akan terhubung ke model registrasi. Baris 25 menunjukkan bahwa pengguna akan menerima hasil data yang masuk setelah registrasi melalui *email*.

**Segmen Program 4.2 : Model Registrasi User**

1. $ctr = 1;
2. $query = $this->db->query("select \* from user");
3. $newId = $this->input->post('regName');
4. $user = $this->input->post('regUsername');
5. $name = $this->input->post('regName');
6. $email = $this->input->post('regEmail');
7. $phone = $this->input->post('regPhone');
8. $pass = $this->input->post('regPass');
9. $jk = $this->input->post('regJk');
10. $cekNewId = 'U' . substr(strtoupper($newId), 0, 1);
11. foreach ($query->result\_array() as $row) {
12. $cekId = substr(strtoupper($row['id\_user']), 0, 2);
13. if ($cekId == $cekNewId) {
14. $ctr++;
15. }
16. }
17. if ($ctr < 10) {
18. $generateId = $cekNewId . '000' . $ctr;
19. } else if ($ctr < 100) {
20. $generateId = $cekNewId . '00' . $ctr;
21. } else if ($ctr < 1000) {
22. $generateId = $cekNewId . '0' . $ctr;
23. } else {
24. $generateId = $cekNewId . $ctr;
25. }
26. $foto = $this->input->post('photoUser');
27. if ($foto == '') {
28. $foto = 'default.jpg';
29. }
30. $data = [
31. "id\_user" => $generateId,
32. "nama\_user" => $name,
33. "username\_user" => $user,
34. "pass\_user" => $pass,
35. "email\_user" => $email,
36. "trade\_link" => '',
37. "jenis\_kelamin" => $jk,
38. "foto" => $foto,
39. "phone" => $phone,
40. "saldo" => 0,
41. "status" => -1
42. ];
43. $this->db->insert('user', $data);
44. $this->session->set\_userdata(array('user' => $generateId));

Segmen program 4.2 adalah potongan program bentuk model untuk bagian registrasi dari website gather.owl. Bentuk model dari segmen program merupakan potongan program yang berisi *syntax-syntax* yang ditujukan untuk penghubung dari program dan database. Setiap bagian dari program akan mengirimkan setiap data yang masuk di website bagian registrasi ke dalam database *db\_owl*. Baris 2 hingga 9 menunjukkan bagian program dimana setiap data yang dimasukkan oleh pengguna di bagian registrasi akan ditampung dalam sebuah variabel. Baris 32 hingga 44 menunjukkan bagian program dimana setiap bagian dalam database diisi dengan setiap variabel yang telah menampung data dari pengguna. Baris 45 menunjukkan sintaks dimana dari baris 32 hingga 44 dimasukkan ke dalam database.

**Segmen Program 4.3 : Controller Login User**

1. $ada = false;
2. $query = $this->db->query("select \* from user");
3. $user = $this->input->post('LogUsername');
4. $pass = $this->input->post('LogPass');
5. foreach ($query->result\_array() as $row) {
6. if ($row['email\_user'] == $user && $row['pass\_user'] ==
7. $pass && $row['status'] != -1) {
8. $ada = true;
9. $id = $row['id\_user'];
10. }
11. if ($row['username\_user'] == $user && $row['pass\_user']
12. == $pass && $row['status'] != -1) {
13. $ada = true;
14. $id = $row['id\_user'];
15. }
16. }
17. if($ada == true) {
18. $data\_session = array('id\_user' => $id);
19. $this->session->set\_userdata($data\_session);
20. $this->User\_model->updateStatus(1);
21. redirect('MainMenu');
22. }else{
23. $this->session->set\_flashdata('flash','Wrong
24. Username/Password !!!');
25. redirect('Login');
26. }

Segmen program 4.3 merupakan potongan program bentuk controller untuk bagian login dari website gather.owl. Baris 2 menunjukkan potongan program yang berfungsi untuk mengambil semua data yang ada di table pengguna dalam database. Baris 3 dan 4 merupakan variabel yang menampung hasil *input* dari pengguna dalam website. Baris 5 hingga 16 menunjukkan bagian program yang berfungsi untuk mengkoreksi data pengguna yang masuk sama dengan data pengguna yang ada dalam tabel *user* yang diambil. Baris 17 hingga 26 menunjukkan bagian program yang berfungsi untuk memeriksa bahwa yang terjadi saat baris 5 hingga 16 benar atau salah. Jika data yang masuk adalah benar, maka website akan membawa pengguna ke tampilan utama. Jika data yang masuk adalah salah, maka website akan berada pada halaman yang sama disertai dengan pesan *error* bahwa data yang dimasukkan adalah salah.

**4.1 User**

Pada subbab in akan membahas tentang segmen program yang ditujukan kepada pengguna yang akan mengunjungi website gather owl. Website gather.owl mempunyai 10 fitur yang dapat dinikmati untuk setiap pengguna. Beberapa fitur-fitur telah terbagi menjadi 2 bagian, yaitu bagian model dan controller. Bagian model akan menghubungkan program dengan database dari website. Bagian controller akan berkaitan dengan proses berjalannya website yang akan dioperasikan oleh pengguna.

**Segmen Program 4.4 : Model Edit Profile User**

1. $id = $this->session->userdata('id\_user');
2. $name = $this->input->post('name');
3. $phone = $this->input->post('phone');
4. $pass = $this->input->post('pass');
5. $email = $this->input->post('email');
6. $trade = $this->input->post('trade');
7. $foto = $this->input->post('foto');
8. $foto = base64\_decode($foto);
9. $data = [
10. "nama\_user" => $name,
11. "pass\_user" => $pass,
12. "email\_user" => $email,
13. "trade\_link" => $trade,
14. "foto" => $foto,
15. "phone" => $phone
16. ];
17. $this->db->where('id\_user', $id);
18. $this->db->update('user', $data);

Segmen Program 4.4 merupakan segmen program dengan bentuk model yang berfungsi untuk mengubah profil dari pengguna yang telah disimpan di database. Profil pengguna dapat diubah sesuai dengan keinginan dari pengguna. Baris 1 hingga 8 menunjukkan variabel-variabel yang menampung data pembaruan yang dimasukkan oleh pengguna. Baris 9 hingga 16 merupakan segmen program yang bertujuan untuk mengumpulkan seluruh isi dari variabel pada baris 1 hingga 8 agar dapat masuk ke dalam database sesuai dengan urutan dari database yang dibuat. Membaharui isi dari tabel user di database dalam segmen program 4.4 terdapat pada baris 18, ditambah dengan pengecekan pengguna yang akan diubah isi dari data tabel dengan menggunakan sintaks yang ada pada baris 17.

**Segmen Program 4.5 : Model Search Item**

1. if(isset($\_SESSION['filter'])) {
2. $query = "select \* from item i
3. join merchant m on m.id\_merchant = i.id\_merchant
4. join game g on g.id\_game = i.id\_game
5. where i.nama\_item like '%" . $keyword . "%'
6. and i.jumlah\_item > 0
7. order by case when i.nama\_item like '" . $keyword . "%'
8. then 0
9. else 1 end ";
10. if($\_SESSION['filter'] == "newest") {
11. $query = $query . ", i.tanggal\_upload desc";
12. } else if($\_SESSION['filter'] == "oldest") {
13. $query = $query . ", i.tanggal\_upload asc";
14. } else if ($\_SESSION['filter'] == "cheapest") {
15. $query = $query . ", i.harga\_item asc";
16. } else {
17. $query = $query . ", i.harga\_item desc";
18. }
19. }else{
20. $query = "select \* from item i
21. join merchant m on m.id\_merchant = i.id\_merchant
22. join game g on g.id\_game = i.id\_game
23. where i.nama\_item like '%" . $keyword . "%'
24. order by case when i.nama\_item like '" . $keyword . "%'
25. then 0 else 1 end, i.nama\_item";
26. }
27. $res = $this->db->query($query);
28. return $res->result\_array();

Segmen program 4.5 merupakan potongan program dalam bentuk model yang berfungsi untuk fitur search dalam website gather owl. Fitur search dalam website gather owl mempunyai fitur tambahan yaitu fitur filter. Fitur filter dalam search berfungsi untuk menyaring barang-barang yang akan dicari sesuai dengan kriteria penyaringan yang dipilih oleh pengguna. Fitur pengecekan filter yang dipilih dalam segmen program 4.5 berada pada baris 1. Baris 2 hingga 9 menunjukkan sintaks berjenis *query* yang berfungsi untuk mengambil barang-barang yang ada di database sesuai dengan pengecekan filter yang dipilih serta *keyword* yang dimasukkan oleh pengguna. Baris 10 hingga 18 menunjukkan pengecekan dari filter yang dipilih oleh pengguna. Baris 19 hingga 26 menunjukkan pengecekan untuk search yang tidak menggunakan filter, sehingga menampilkan barang-barang yang sesuai dengan masukan dari pengguna. Seluruh barang yang telah terambil akan disimpan didalam penyimpanan program yaitu *array* yang ditunjukkan pada baris 28 pada segmen program 4.5.

**Segmen Program 4.6 : Controller Add Chart**

1. $idI = $this->input->post('idItem');
2. $this->Cart\_model->addCart($idI);

Segmen program 4.6 merupakan potongan program berbentuk controller yang berfungsi untuk menambahkan barang ke dalam daftar belanja milik pengguna. Setiap barang pilihan dari pengguna akan dimasukkan dalam daftar belanja sebelum masuk dalam tahap transaksi. Proses masuk keluarnya barang berjalan berdasarkan identitas dari barang tersebut yang disimpan dalam sebuah variabel seperti yang ditunjukkan pada baris pertama segmen program 4.6. Baris 2 (dua) menunjukkan sintaks yang berfungsi untuk memindahkan data melalui *parameter* menuju ke model *add chart*.

**Segmen Program 4.7 : Model Add Chart**

1. $ada = false;
2. $id = $this->session->userdata('id\_user');
3. $cekQuery = $this->db->query("select \* from USER\_CART");
4. $amount = 1;
5. foreach($cekQuery->result\_array() as $row) {
6. if($id == $row['id\_user'] && $idI == $row['id\_item'] &&
7. $row['status'] == 0) {
8. $ada = true;
9. $amount = $row['amount'];
10. }
11. }

**Segmen Program 4.7 : (lanjutan)**

1. if($ada){
2. $amount++;
3. $query = "update user\_cart set amount = $amount where
4. id\_user = '" . $id . "' and id\_item = '" . $idI . "' ";
5. $this->db->query($query);
6. } else{
7. $query = "insert into
8. user\_cart(id\_user,id\_item,amount,status)
9. values('" . $id . "' , '" . $idI . "' , $amount , 0)";
10. $this->db->query($query);
11. }

Segmen program 4.7 merupakan segmen program dalam bentuk model yang berfungsi untuk menambahkan barang ke dalam daftar belanja. Sebelum memasukkan barang ke dalam daftar belanja, program memeriksa pengguna yang sedang *login* dengan data nama user yang ada dalam database seperti yang ditunjukkan pada baris 5 (lima) hingga 11 (sebelas). Selain memeriksa pengguna yang sedang login, program juga memeriksa barang yang dipilih sudah ada didalam tabel user\_chart atau belum ada. Baris 3 (tiga) menunjukkan sintaks yang berfungsi untuk mengambil seluruh data yang ada di tabel user\_chart dalam database. Jika hasil dari baris 5 (lima) hingga 11 (sebelas) bahwa barang yang dipilih pengguna telah ada dalam tabel user\_chart, maka program akan membuat sintaks yang berfungsi untuk menambah jumlah barang yang dipilih pengguna seperti yang ditunjukkan pada baris 13 (tiga belas), serta membuat query yang berfungsi untuk mengubah jumlah barang pada baris dari tabel user\_chart. Jika hasil dari baris 5 (lima) hingga 11 (sebelas) bahwa barang yang dipilih pengguna belum ada dalam tabel user\_chart, maka program akan membuat query yang berfungsi untuk menambah baris yang berisi barang yang dipilih pengguna ke dalam tabel user\_chart.

**Segmen Program 4.8 : Controller Buy Item**

1. $id = $this->session->userdata('id\_user');
2. $cart = array();
3. $cart = $this->input->post('cart');
4. $cart = json\_decode($cart, true);
5. $this->Trans\_model->insertTrans();
6. $this->User\_model->updateSaldo("transaksi");
7. for ($i = 0; $i < count($cart); $i++) {
8. $this->Cart\_model->updateStatus2($cart[$i]['id']);
9. }

Segmen program 4.8 merupakan potongan program dalam bentuk controller yang berfungsi untuk pembelian item dalam website gather owl. Setelah semua barang yang ingin dibeli pengguna masuk ke dalam daftar belanja, proses transaksi dijalankan yang diawali dengan memasukkan seluruh isi dari daftar belanja ke dalam array yang ditunjukkan dalam baris 2 (dua) hingga 4 (empat), baris 2 (dua) menunjukkan inisialisasi array untuk daftar belanja serta 3 (tiga) dan 4 (empat) menunjukkan proses masuknya daftar belanja dalam array. Baris 5 (lima) merupakan bagian program yang memanggil *function* yang digunakan untuk proses pembayaran. Setelah pembayaran selesai dijalankan, program akan merubah isi dari baris saldo dalam tabel user dengan hasil selisih antara saldo awal dengan total harga dari barang yang dibeli pengguna. Baris 7 (tujuh) hingga 9 (sembilan) menunjukkan segmen program dimana program merubah isi dari baris status dalam tabel user\_chart dari setiap barang yang dibeli pengguna agar terhapus dari daftar belanja pengguna.

**Segmen Program 4.9 : Model Insert Transaksi**

1. $ctr = 1;
2. $query = $this->db->query("select \* from user");
3. $idUser = $this->session->userdata('id\_user');
4. foreach ($query->result\_array() as $row) {
5. if ($row['id\_user'] == $idUser) {
6. $newId = $row['nama\_user'];
7. }
8. }
9. $cart = array();
10. $cart = $this->input->post('cart');
11. $cart = json\_decode($cart, true);
12. $gross = $this->input->post('gross\_amount');
13. //$cashback = $gross \* 10 / 100;
14. $tgl = date("Y-m-d H:i:s");
15. $query = $this->db->query("select \* from transaksi");
16. $cekNewId = 'T' . substr(strtoupper($newId), 0, 1);
17. foreach ($query->result\_array() as $row) {
18. $cekId = substr(strtoupper($row['id\_transaksi']),0,2);
19. if ($cekId == $cekNewId) {
20. $ctr++;
21. }
22. }
23. if ($ctr < 10) {
24. $generateId = $cekNewId . '000' . $ctr;
25. } else if ($ctr < 100) {
26. $generateId = $cekNewId . '00' . $ctr;
27. } else if ($ctr < 1000) {
28. $generateId = $cekNewId . '0' . $ctr;

**Segmen Program 4.9 : (lanjutan)**

1. } else {
2. $generateId = $cekNewId . $ctr;
3. }
4. if (isset($\_SESSION['id\_promo'])) {
5. $query2 = $this->db->query("select \* from promo");
6. foreach ($query2->result\_array() as $row) {
7. if ($row['id\_promo'] == $\_SESSION['id\_promo']) {
8. $promo = $row['id\_promo'];
9. $potongan = $row['potongan'];
10. $maksimal = $row['maksimal'];
11. }
12. }
13. $this->session->unset\_userdata('id\_promo');
14. $cashback = $gross \* $potongan / 100;
15. if ($cashback > $maksimal) {
16. $cashback = $maksimal;
17. }
18. $this->session->set\_userdata(array('cashback’=>
19. $cashback));
20. } else {
21. $promo = '';
22. $cashback = '';
23. }
24. $data = [
25. "id\_transaksi" => $generateId,
26. "id\_user" => $idUser,
27. "id\_promo" => $promo,
28. "Gross\_Amount" => $gross,
29. "tanggal\_transaksi" => $tgl,
30. "cashback" => $cashback,
31. "status" => 0
32. ];
33. $this->db->insert('transaksi', $data);

Segmen program 4.9 merupakan segmen program bentuk model yang berfungsi untuk proses transaksi dalam website gather owl. Proses transaksi dalam segmen program 4.9 akan berjalan saat segmen program 4.8 berjalan. Segmen program 4.9 diawali dengan mencari pengguna yang sedang login yang ditunjukkan pada baris 2 (dua) hingga 8 (delapan). Baris 9 (sembilan) hingga 11 (sebelas) merupakan bagian program yang sama dengan segmen program 4.8 pada baris 2 (dua) hingga 4 (empat) yang berguna untuk memasukkan daftar belanja ke dalam suatu array. Proses pembelian barang dapat menggunakan promo. Program akan memeriksa promo yang dipakai oleh pengguna yang ditunjukkan pada baris 32 (tiga puluh dua) hingga 51 (lima puluh satu) dalam segmen program 4.9. Setelah seluruh promo dan barang sudah diperiksa, program akan menyusun data-data yang akan dikirim ke dalam tabel transaksi sesuai dengan baris dalam tabel, seperti yang ditunjukkan pada baris 52 (lima puluh dua) hingga 60 (enam puluh). Baris 61 (enam puluh satu) menunjukkan program mengirim data transaksi yang sudah dibuat pada baris 52 (lima puluh dua) hingga 60 (enam puluh) ke dalam tabel transaksi.

**Segmen Program 4.10 : Model Insert Transaksi Item**

1. $idmerchant = array();
2. for ($i = 0; $i < count($cart); $i++) {
3. echo "masuk" . $i;
4. // KIRIM EMAIL KE MERCHANT
5. $query2 = $this->db->query("select \* from item");
6. foreach ($query2->result\_array() as $row2) {
7. if ($row2['id\_item'] == $cart[$i]['id']) {
8. if (!in\_array($row2['id\_merchant'],
9. $idmerchant)) {
10. array\_push($idmerchant,
11. $row2['id\_merchant']);
12. }
13. }
14. }
15. $data = [
16. "id\_transaksi" => $generateId,
17. "id\_item" => $cart[$i]['id'],
18. "jumlah" => $cart[$i]['quantity'],
19. "subtotal" => $cart[$i]['subtotal'],
20. "keterangan" => "\*Upload your proof(s) and press the
21. update button"
22. ];
23. $this->db->insert('transaksi\_item', $data);
24. }

Pada segmen program 4.9, program hanya mengirimkan *header* dari proses transaksi yang dijalankan dan setiap barang yang dibeli belum masuk ke dalam database. Program akan memasukkan setiap barang yang dibeli ke dalam tabel transaksi item, yang ditunjukkan dalam segmen program 4.10. Baris pertama menunjukkan inisialisasi untuk menyimpan id dari *merchant*. Setiap terjadi transaksi, merchant akan diberi konfirmasi melalui email bahwa barang yang dijual telah dibeli oleh pengguna. Pesan konfirmasi tersebut di dalam program ditunjukkan pada baris 2(dua) hingga 14 (empat belas). Setelah program mengirim pesan konfirmasi, program akan menyusun data untuk memasukkan setiap barang yang dibeli oleh pengguna yang ditunjukkan pada baris 15 (lima belas) hingga 22 (dua puluh dua). Baris 23 (dua puluh tiga) menunjukkan program mengirim data barang yang sudah disusun pada baris 15 (lima belas) hingga 22 (dua puluh dua) ke dalam tabel transaksi item.

**Segmen Program 4.11 : Controller Insert Channel**

1. $this->Channel\_model->insertChannel();
2. $this->ChannelUser\_model->insertChannelUser();
3. redirect('Community/refreshAccItem');

Segmen program 4.11 merupakan bagian program bentuk controller dimana pengguna membuat *channel* pada website gather owl. Fitur channel merupakan bagian dari halaman *community*. Baris 1 akan memanggil function yang digunakan sebagai proses pembuatan channel dalam website. Baris 2 akan memanggil function dengan fungsi yang berbeda, yaitu function yang digunakan untuk pengguna yang dapat mengakses channel berdasarkan pemilihan dari pembuat channel. Baris 1 akan lebih lanjut dijelaskan dalam segmen program 4.12 dan baris 2 akan lebih lanjut dijelaskan dalam segmen program 4.13.

**Segmen Program 4.12 : Model Insert Channel**

1. $ctr = 1;
2. $query = $this->db->query("select \* from channel");
3. $newId = $this->input->post('nama\_channel');
4. $cekNewId = 'C' . substr(strtoupper($newId), 0, 1);
5. foreach ($query->result\_array() as $row) {
6. $cekId = substr(strtoupper($row['id\_channel']), 0, 2);
7. if ($cekId == $cekNewId) {
8. $ctr++;
9. }
10. }
11. if ($ctr < 10) {
12. $generateId = $cekNewId . '000' . $ctr;
13. } else if ($ctr < 100) {
14. $generateId = $cekNewId . '00' . $ctr;
15. } else if ($ctr < 1000) {
16. $generateId = $cekNewId . '0' . $ctr;
17. } else{
18. $generateId = $cekNewId . $ctr;
19. }
20. $foto = $this->input->post('foto');
21. $foto = base64\_decode($foto);
22. $data = [
23. "id\_channel" => $generateId,
24. "nama\_channel" => $this->input->post('nama\_channel'),
25. "foto\_channel" => $foto
26. ];
27. $this->db->insert('channel', $data);

Segmen program 4.12 merupakan segmen program model yang berjalan pada saat pengguna menyusun channel. Baris 2 (dua) hingga baris 10 (sepuluh) merupakan bagian program yang berfungsi untuk menghasilkan id yang bersifat unik, setiap id dibentuk dengan struktur yang berbeda, yang akan dimasukkan ke baris id dalam tabel channel. Baris 11 (sebelas) hingga 19 (sembilan belas) menunjukkan bagian program dimana program menghasilkan nomor urut id untuk id channel.

**Segmen Program 4.13 : Model Insert Channel User**

1. $idUser = $this->session->userdata('id\_user');
2. if (isset($\_SESSION['newChannel'])) {
3. $idChannel = $this->session->userdata('newChannel');
4. $this->session->unset\_userdata('newChannel');
5. $jenis = 2;
6. } else {
7. $idChannel = $this->input->post('idChannel');
8. $jenis = -1;
9. }
10. $ada = false;
11. $query = $this->db->query("select \* from channel\_user");
12. foreach ($query->result\_array() as $row) {
13. if ($idChannel == $row['id\_channel'] && $idUser ==
14. $row['id\_user']) {
15. $ada = true;
16. }
17. }
18. if (!$ada) {
19. $data = [
20. "id\_user" => $idUser,
21. "id\_channel" => $idChannel,
22. "jenis" => $jenis
23. ];
24. $this->db->insert('channel\_user', $data);
25. }

**Segmen Program 4.14 : Controller Insert Team**

1. $this->Team\_model->insertTeam();
2. $this->TeamMember\_model->insertTeamMember();
3. redirect('Community/refreshAccItem');

**Segmen Program 4.15 : Model Insert Team**

1. $idUser = $this->session->userdata('id\_user');
2. $ctr = 1;
3. $query = $this->db->query("select \* from team");
4. $newId = $this->input->post('nama\_team');
5. $cekNewId = 'T' . substr(strtoupper($newId), 0, 1);
6. foreach ($query->result\_array() as $row) {
7. $cekId = substr(strtoupper($row['id\_team']), 0, 2);
8. if ($cekId == $cekNewId) {
9. $ctr++;
10. }
11. }
12. if ($ctr < 10) {
13. $generateId = $cekNewId . '000' . $ctr;
14. } else if ($ctr < 100) {
15. $generateId = $cekNewId . '00' . $ctr;
16. } else if ($ctr < 1000) {
17. $generateId = $cekNewId . '0' . $ctr;
18. } else {
19. $generateId = $cekNewId . $ctr;
20. }
21. $foto = $this->input->post('foto');
22. $foto = base64\_decode($foto);
23. $tgl = date("Y-m-d H:i:s");
24. $data = [
25. "id\_team" => $generateId,
26. "id\_user" => $this->input->post('id\_user'),
27. "nama\_team" => $this->input->post('nama\_team'),
28. "tanggal\_pembuatan" => $tgl,
29. "foto\_team" => $foto
30. ];
31. $this->db->insert('team', $data);
32. $data = [
33. "id\_user" => $idUser,
34. "id\_team" => $generateId
35. ];
36. $this->db->insert('team\_members', $data);
37. $this->session->set\_userdata(array('insertTeam' => "true"));

**Segmen Program 4.16 : Model Insert Team Members**

1. $idUser = $this->session->userdata('id\_user');
2. if(isset($\_SESSION['insertTeam'])) {
3. $idTeam = $this->session->userdata('idTeam');
4. } else {
5. $idTeam = $this->input->post('idTeam');
6. }
7. $ada = false;
8. $query = $this->db->query("select \* from team\_members");
9. foreach ($query->result\_array() as $row) {
10. if ($idTeam == $row['id\_team'] && $idUser ==
11. $row['id\_user']){
12. $ada = true;
13. }
14. }
15. if (!$ada) {
16. $data = [
17. "id\_user" => $idUser,
18. "id\_team" => $idTeam
19. ];
20. $this->db->insert('team\_members', $data);
21. }
22. $this->session->unset\_userdata('insertTeam');