**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 (tiga) 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 (empat) 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 (dua koma enam) *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 (empat) 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|min\_length[8]');
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 pertama hingga 18 (delapan belas) 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 (sembilan belas) hingga 26 (dua puluh enam) merupakan bagian program yang berfungsi untuk memeriksa benar salah dari baris pertama hingga 18 (delapan belas). Baris 20 (dua puluh) dan 21 (dua puluh satu) menunjukkan bahwa jika hasil dari pengecekan validasi adalah salah, maka pengguna tetap pada halaman register. Baris 23 (dua puluh tiga) menunjukkan jika hasil dari pengecekan validasi adalah benar, maka data yang dikirimkan akan terhubung ke model registrasi. Baris 25 (dua puluh lima) 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,

**Segmen Program 4.2 : ( Lanjutan)**

1. "foto" => $foto,
2. "phone" => $phone,
3. "saldo" => 0,
4. "status" => -1
5. ];
6. $this->db->insert('user', $data);
7. $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 (dua) hingga 9 (sembilan) menunjukkan bagian program dimana setiap data yang dimasukkan oleh pengguna di bagian registrasi akan ditampung dalam sebuah variabel. Baris 32 (tiga puluh dua) hingga 44 (empat puluh empat) menunjukkan bagian program dimana setiap bagian dalam database diisi dengan setiap variabel yang telah menampung data dari pengguna. Baris 45 (empat puluh lima) menunjukkan sintaks dimana dari baris 32 (tiga puluh dua) hingga 44 (empat puluh empat) 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{

**Segmen Program 4.3 : (Lanjutan)**

1. $this->session->set\_flashdata('flash','Wrong
2. Username/Password !!!');
3. redirect('Login');
4. }

Segmen program 4.3 merupakan potongan program bentuk controller untuk bagian login dari website gather.owl. Baris 2 (dua) menunjukkan potongan program yang berfungsi untuk mengambil semua data yang ada di table pengguna dalam database. Baris 3 (tiga) dan 4 (empat) merupakan variabel yang menampung hasil *input* dari pengguna dalam website. Baris 5 (lima) hingga 16 (enam belas) menunjukkan bagian program yang berfungsi untuk mengkoreksi data pengguna yang masuk sama dengan data pengguna yang ada dalam tabel *user* yang diambil. Baris 17 (tujuh belas) hingga 26 (dua puluh enam) menunjukkan bagian program yang berfungsi untuk memeriksa bahwa yang terjadi saat baris 5 (lima) hingga 16 (enam belas) 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 15 (lima belas) fitur yang dapat dinikmati untuk setiap pengguna. Beberapa fitur-fitur telah terbagi menjadi 2 (dua) 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');

**Segmen Program 4.4 : (Lanjutan)**

1. $foto = $this->input->post('foto');
2. $foto = base64\_decode($foto);
3. $data = [
4. "nama\_user" => $name,
5. "pass\_user" => $pass,
6. "email\_user" => $email,
7. "trade\_link" => $trade,
8. "foto" => $foto,
9. "phone" => $phone
10. ];
11. $this->db->where('id\_user', $id);
12. $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 pertama hingga 8 (delapan) menunjukkan variabel-variabel yang menampung data pembaruan yang dimasukkan oleh pengguna. Baris 9 (sembilan) hingga 16 (enam belas) merupakan segmen program yang bertujuan untuk mengumpulkan seluruh isi dari variabel pada baris pertama hingga 8 (delapan) 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 (delapan belas), ditambah dengan pengecekan pengguna yang akan diubah isi dari data tabel dengan menggunakan sintaks yang ada pada baris 17 (tujuh belas).

**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{

**Segmen Program 4.5 : (Lanjutan)**

1. $query = "select \* from item i
2. join merchant m on m.id\_merchant = i.id\_merchant
3. join game g on g.id\_game = i.id\_game
4. where i.nama\_item like '%" . $keyword . "%'
5. order by case when i.nama\_item like '" . $keyword . "%'
6. then 0 else 1 end, i.nama\_item";
7. }
8. $res = $this->db->query($query);
9. 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 pertama. Baris 2 (dua) hingga 9 (sembilan) 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 (sepuluh) hingga 18 (delapan belas) menunjukkan pengecekan dari filter yang dipilih oleh pengguna. Baris 19 (sembilan belas) hingga 26 (dua puluh enam) 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 (dua puluh delapan) 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 = $row['amount'];
10. }
11. }
12. if($ada){
13. $amount++;
14. $query = "update user\_cart set amount = $amount where
15. id\_user = '" . $id . "' and id\_item = '" . $idI . "' ";
16. $this->db->query($query);
17. } else{
18. $query = "insert into
19. user\_cart(id\_user,id\_item,amount,status)
20. values('" . $id . "' , '" . $idI . "' , $amount , 0)";
21. $this->db->query($query);
22. }

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']);}

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. }

**Segmen Program 4.9 : (Lanjutan)**

1. }
2. $cart = array();
3. $cart = $this->input->post('cart');
4. $cart = json\_decode($cart, true);
5. $gross = $this->input->post('gross\_amount');
6. //$cashback = $gross \* 10 / 100;
7. $tgl = date("Y-m-d H:i:s");
8. $query = $this->db->query("select \* from transaksi");
9. $cekNewId = 'T' . substr(strtoupper($newId), 0, 1);
10. foreach ($query->result\_array() as $row) {
11. $cekId = substr(strtoupper($row['id\_transaksi']),0,2);
12. if ($cekId == $cekNewId) {
13. $ctr++;
14. }
15. }
16. if ($ctr < 10) {
17. $generateId = $cekNewId . '000' . $ctr;
18. } else if ($ctr < 100) {
19. $generateId = $cekNewId . '00' . $ctr;
20. } else if ($ctr < 1000) {
21. $generateId = $cekNewId . '0' . $ctr;
22. } else {
23. $generateId = $cekNewId . $ctr;
24. }
25. if (isset($\_SESSION['id\_promo'])) {
26. $query2 = $this->db->query("select \* from promo");
27. foreach ($query2->result\_array() as $row) {
28. if ($row['id\_promo'] == $\_SESSION['id\_promo']) {
29. $promo = $row['id\_promo'];
30. $potongan = $row['potongan'];
31. $maksimal = $row['maksimal'];
32. }
33. }
34. $this->session->unset\_userdata('id\_promo');
35. $cashback = $gross \* $potongan / 100;
36. if ($cashback > $maksimal) {
37. $cashback = $maksimal;
38. }
39. $this->session->set\_userdata(array('cashback’=>
40. $cashback));
41. } else {
42. $promo = '';
43. $cashback = '';
44. }
45. $data = [
46. "id\_transaksi" => $generateId,
47. "id\_user" => $idUser,
48. "id\_promo" => $promo,
49. "Gross\_Amount" => $gross,
50. "tanggal\_transaksi" => $tgl,
51. "cashback" => $cashback,
52. "status" => 0
53. ];
54. $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 pertama akan memanggil function yang digunakan sebagai proses pembuatan channel dalam website. Baris 2 (dua) akan memanggil function dengan fungsi yang berbeda, yaitu function yang digunakan untuk pengguna yang dapat mengakses channel berdasarkan pemilihan dari pembuat channel serta keinginan dari pengguna lain. Baris pertama akan lebih lanjut dijelaskan dalam segmen program 4.12 dan baris 2 (dua) 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) {

**Segmen Program 4.12 : (Lanjutan)**

1. $cekId = substr(strtoupper($row['id\_channel']), 0, 2);
2. if ($cekId == $cekNewId) {
3. $ctr++;
4. }
5. }
6. if ($ctr < 10) {
7. $generateId = $cekNewId . '000' . $ctr;
8. } else if ($ctr < 100) {
9. $generateId = $cekNewId . '00' . $ctr;
10. } else if ($ctr < 1000) {
11. $generateId = $cekNewId . '0' . $ctr;
12. } else{
13. $generateId = $cekNewId . $ctr;
14. }
15. $foto = $this->input->post('foto');
16. $foto = base64\_decode($foto);
17. $data = [
18. "id\_channel" => $generateId,
19. "nama\_channel" => $this->input->post('nama\_channel'),
20. "foto\_channel" => $foto
21. ];
22. $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. Baris 22 (dua puluh dua) hingga 26 (dua puluh enam) menunjukkan potongan program yang berfungsi untuk menyusun data yang akan masuk ke dalam tabel channel sesuai dengan baris yang ada dalam tabel.

**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");

**Segmen Program 4.13 : (Lanjutan)**

1. foreach ($query->result\_array() as $row) {
2. if ($idChannel == $row['id\_channel'] && $idUser ==
3. $row['id\_user']) {
4. $ada = true;
5. }
6. }
7. if (!$ada) {
8. $data = [
9. "id\_user" => $idUser,
10. "id\_channel" => $idChannel,
11. "jenis" => $jenis
12. ];
13. $this->db->insert('channel\_user', $data);
14. }

Segmen program 4.13 merupakan segmen program model yang berjalan pada saat pengguna memasukkan pengguna-pengguna lain yang berkeinginan untuk bergabung dengan channel. Baris 2 (dua) hingga baris 9 (sembilan) merupakan bagian program yang berguna untuk menentukan jenis pengguna dalam program. Jika pengguna adalah pembuat channel, maka di dalam tabel pada baris jenis akan bernilai 2 (dua). Jika pengguna adalah pengguna lain yang ingin bergabung, maka di dalam tabel pada baris jenis akan bernilai -1 (minus satu). Sebelum program memasukkan pengguna-pengguna lain, program akan memeriksa pengguna didalam tabel channel user sudah bergabung ke dalam channel atau belum bergabung, pemeriksaan ini ditunjukkan dalam segmen program 4.13 pada baris 11 (sebelas) hingga 17 (tujuh belas). Jikalau hasil dari bagian program baris 11 (sebelas) hingga 17 (tujuh belas) menyatakan bahwa belum tergabung, maka data pengguna dapat masuk ke dalam tabel channel user yang ditunjukkan pada baris 18 (delapan belas) hingga 25 (dua puluh lima).

**Segmen Program 4.14 : Controller Insert Team**

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

Segmen program 4.14 merupakan segmen program controller yang berguna untuk menambah *team* dalam website gather.owl. Baris pertama pada segmen program 4.14 akan memanggil function yang digunakan sebagai proses pembuatan team ke dalam tabel team. Baris 2 (dua) akan memanggil function dengan fungsi yang berbeda dengan baris pertama, yaitu function yang digunakan untuk pengguna yang berkeinginan untuk bergabung dengan team. Baris pertama akan lebih lanjut dijelaskan dalam segmen program 4.15 dan baris 2 (dua) akan lebih lanjut dijelaskan dalam segmen program 4.16.

**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 4.15 merupakan segmen program model yang berjalan pada saat pengguna menyusun team baru ke dalam database. Baris 3 (tiga) hingga baris 11 (sebelas) merupakan bagian program yang berfungsi untuk menghasilkan id yang unik yang akan dimasukkan ke baris id dalam tabel team. Baris 12 (dua belas) hingga 20 (dua puluh) menunjukkan potongan program dimana program menghasilkan nomor urut untuk baris id team dalam tabel team. Baris 21 (dua puluh satu) hingga 30 (tiga puluh) menunjukkan potongan program yang berfungsi untuk menyusun data sesuai dengan baris yang ada dalam tabel team serta inisialisasi variabel untuk menyimpan data yang akan dikirim ke tabel team. Program akan mengirim data dari baris 21 (dua puluh satu) hingga 30 (tiga puluh) ke dalam tabel team menggunakan sintaks yang ada pada baris 31 (tiga puluh satu). Baris 32 (tiga puluh dua) hingga 35 (tiga puluh lima) merupakan bagian program dimana program menyusun data yang akan dikirim ke tabel team\_members, program akan mengirim ke dalam tabel team\_members dengan sintaks yang ditunjukkan pada baris 36 (tiga puluh enam).

**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');

Segmen program 4.16 merupakan segmen program model yang berjalan pada saat pengguna memasukkan pengguna-pengguna lain yang berkeinginan untuk bergabung dengan team serta pembuat dari team. Baris 2 (dua) hingga 7 (tujuh) merupakan bagian program yang berguna untuk menentukan team yang dipilih oleh pengguna dalam program. Sebelum program memasukkan pengguna serta pembuat dalam tabel team\_members, program akan memeriksa pengguna didalam tabel team\_members sudah bergabung ke dalam team atau belum bergabung, program pemeriksaan dalam segmen program 4.16 ditunjukkan dalam pada baris 7 (tujuh) hingga 14 (empat belas). Jikalau hasil dari bagian program baris 7 (tujuh) hingga 14 (empat belas) menyatakan bahwa belum tergabung, maka data pengguna dapat masuk ke dalam tabel team\_members yang ditunjukkan pada baris 15 (lima belas) hingga 21 (dua puluh satu).

**Segmen Program 4.17 : Controller Insert Tournament**

1. $this->Tournament\_model->insertTournament();
2. $slot = $this->input->post('jumlah\_slot');
3. $pembagian = $slot / 2;
4. $this->Pertandingan\_model-
5. >insertPertandingan('final');
6. if ($pembagian != 1) {
7. $cekKiri = $pembagian / 2;
8. $cekKanan = $pembagian - $cekKiri;
9. $this->Pertandingan\_model-
10. >insertPertandingan('semifinalkiri');
11. $this->Pertandingan\_model-
12. >insertPertandingan('semifinalkanan');
13. $this->Pertandingan\_model-
14. >insertPertandingan('third');
15. $kiri = 1;
16. $kanan = 1;
17. $ctr = 0;
18. $ada = false;
19. while ($kiri \* 2 <= $cekKiri) {
20. $kiri \*= 2;
21. if ($ctr == 0) {
22. $this->Pertandingan\_model-
23. >insertPertandingan('quarterfinalkiri1');
24. $this->Pertandingan\_model-
25. >insertPertandingan('quarterfinalkiri2');
26. } else if ($ctr > 0) {
27. $ada = true;
28. }
29. $ctr++;
30. }
31. if ($ada == true) {
32. $ctr -= 1;
33. $ctr2 = 4;
34. for ($i = $ctr; $i >= 1; $i--) {
35. for ($j = 1; $j <= $ctr2; $j++) {

**Segmen Program 4.17 : (Lanjutan)**

1. $this->Pertandingan\_model-
2. >insertPertandingan('round'
3. . $i . 'kiri' . $j);
4. }
5. $ctr2 \*= 2;
6. }
7. }
8. $ctr = 0;
9. $ada = false;
10. while ($kanan \* 2 <= $cekKanan) {
11. $kanan \*= 2;
12. if ($ctr == 0) {
13. $this->Pertandingan\_model
14. >insertPertandingan('quarterfinalkanan1');
15. $this->Pertandingan\_model
16. >insertPertandingan('quarterfinalkanan2');
17. } else if ($ctr > 0) {
18. $ada = true;
19. }
20. $ctr++;
21. }
22. if ($ada == true) {
23. $ctr -= 1;
24. $ctr2 = 4;
25. for ($i = $ctr; $i >= 1; $i--) {
26. for ($j = 1; $j <= $ctr2; $j++) {
27. $this->Pertandingan\_model-
28. >insertPertandingan('round'
29. . $i . 'kanan' . $j);
30. }
31. $ctr2 \*= 2;
32. }
33. }
34. }

Segmen program 4.17 berisi algoritma untuk menentukan nama, letak, dan banyaknya pertandingan dalam turnamen. Sebelum menghitung banyaknya pertandingan yang dibutuhkan, Pada baris pertama program memanggil model *insert* turnamen yang ditunjukkan pada segmen program 4.18 untuk menambah turnamen yang idnya akan digunakan dalam penambahan pertandingan turnamen. Pada baris 2 (dua), program mengambil input slot dari pengguna yang kemudian digunakan untuk menentukan jumlah pertandingan pada bagian kiri dan kanan turnamen sesuai pada baris 3 (tiga) hingga baris 7 (tujuh). Selanjutnya pada baris 8 (delapan) hingga 68 (enam puluh delapan), program menambahkan pertandingan – pertandingan yang diperlukan sesuai slot yang sudah didapatkan.

**Segmen Program 4.18 : Model Insert Tournament**

1. date\_default\_timezone\_set('Asia/Jakarta');
2. $idUser = $this->session->userdata('id\_user');
3. $ctr = 1;
4. $query = $this->db->query("select \* from
5. tournament");
6. $newId = $this->input->post('nama\_turnament');
7. $cekNewId = 'O'.substr(strtoupper($newId),0,1);
8. foreach ($query->result\_array() as $row) {
9. $cekId=substr(strtoupper($row['id\_turnament']),0,
10. 2);
11. if ($cekId == $cekNewId) {
12. $ctr++;
13. }
14. }
15. if ($ctr < 10) {
16. $generateId = $cekNewId.'000'.$ctr;
17. } else if ($ctr < 100) {
18. $generateId = $cekNewId.'00'.$ctr;
19. } else if ($ctr < 1000) {
20. $generateId = $cekNewId.'0'.$ctr;
21. } else {
22. $generateId = $cekNewId.$ctr;
23. }
24. $tgl = $this->input->post('tanggal\_mulai');
25. $tgl = strtotime($tgl);
26. $tgl = date("Y-m-d H:i", $tgl);
27. $jenis = $this->input->post('jenis\_pemain');
28. if ($jenis == "Team") {
29. $jenis = 0;
30. } else {
31. $jenis = 1;
32. }
33. $this->session->set\_userdata(array('idTurnament' =>
34. $generateId));
35. $data = [
36. "id\_turnament" => $generateId,
37. "id\_channel" => $this->input->post('id\_channel'),
38. "id\_game" => $this->input->post('id\_game'),
39. "nama\_turnament" => $this->input-
40. >post('nama\_turnament'),
41. "jumlah\_pemain" => "0",
42. "tanggal\_mulai" => $tgl,
43. "jumlah\_slot" => $this->input-
44. >post('jumlah\_slot'),
45. "jenis\_pemain" => $jenis,

**Segmen Program 4.18 : (lanjutan)**

1. "status" => 0
2. ];
3. $this->db->insert('tournament', $data);

Segmen Program 4.18 merupakan bagian program yang digunakan untuk menambahkan turnamen yang dibuat oleh pengguna dalam suatu channel. Bagian program ini menerima input dari pengguna dan membuat id baru sesuai dengan format id turnamen yang kemudian ditambahkan ke database. Pada baris pertama, program menentukan standar zona waktu yaitu zona waktu Jakarta atau WIB (Waktu Indonesia Barat). Pada baris ketiga hingga 31 (tiga puluh satu), program membuat variabel yang nantinycontrollera digunakan untuk mengisi kolom yang tidak diisi oleh input pengguna. Selanjutnya pada baris 33 (tiga puluh tiga) hingga baris 46 (empat puluh enam), program mengambil mengisi kolom dengan input user dan variabel yang telah dibuat dan kemudian ditambahkan ke database.

**Segmen Program 4.19 : Model Insert Tournament Standings**

1. $data = [
2. "id\_turnament" => $generateId,
3. ];
4. $this->db->insert('tournament\_standing', $data);

Segmen Program 4.19 merupakan bagian program yang berfungsi untuk menambahkan klasemen turnamen setelah menambahkan turnamen ke database. Bagian program ini hanya mengambil id turnamen yang sudah dibuat pada segmen program 4.18. Program hanya mengambil id turnamen walaupun pada tabel klasemen terdapat kolom juara 1 (satu), 2 (dua) dan 3 (tiga) karena memang disediakan untuk pengisian juara setelah turnamen selesai. Cara ini memudahkan validasi jika terdapat turnamen yang pengisian kolom juaranya tepisah.

**Segmen Program 4.20 : Model Insert Pertandingan**

1. $idUser = $this->session->userdata('id\_user');
2. $idTurnament = $this->session-
3. >userdata('idTurnament');
4. $ctr = 1;
5. $query = $this->db->query("select \* from

**Segmen Program 4.20 : (Lanjutan)**

1. pertandingan");
2. $newId = $bagian;
3. $cekNewId = 'B'.substr(strtoupper($newId),0,1);
4. foreach ($query->result\_array() as $row) {
5. $cekId=substr(strtoupper($row['id\_match']),0,2);
6. if ($cekId == $cekNewId) {
7. $ctr++;
8. }
9. }
10. if ($ctr < 10) {
11. $generateId = $cekNewId . '000' . $ctr;
12. } else if ($ctr < 100) {
13. $generateId = $cekNewId . '00' . $ctr;
14. } else if ($ctr < 1000) {
15. $generateId = $cekNewId . '0' . $ctr;
16. } else {
17. $generateId = $cekNewId . $ctr;
18. }
19. $data = [
20. "id\_match" => $generateId,
21. "id\_turnament" => $idTurnament,
22. "waktu\_mulai" => '',
23. "bagian" => $bagian,
24. "status" => 0,
25. "skor\_1" => '',
26. "skor\_2" => '',
27. "team\_1" => '',
28. "team\_2" => ''
29. ];
30. $this->db->insert('pertandingan', $data);

Segmen Program 4.20 digunakan untuk menambahkan pertandingan turnamen yang diperlukan ke dalam database. Bagian program ini juga hanya memerlukan id turnamen yang telah dibuat pada segmen program 4.18. Pada baris 5 (lima) hingga 23 (dua puluh tiga), program membuat id baru dengan memeriksa semua id pertandingan di dalam database untuk menghindari kesamaan id. Pada baris 24 (dua puluh empat) hingga 35 (tiga puluh lima), program menambahkan pertandingan ke dalam database dengan beberapa kolom kosong yang disediakan untuk penngisian selama turnamen berlangsung.

**Segmen Program 4.21 : Model Join Pertandingan**

1. $idTurnament = $this->input->post('id\_turnament');
2. $idTeam = $this->input->post('id\_team');

**Segmen Program 4.21 : (Lanjutan)**

1. $slot = $this->input->post('jumlah\_slot');
2. $join = false;
3. $query = $this->db->query("select \* from pertandingan");
4. foreach ($query->result\_array() as $row) {
5. if ($row['id\_turnament'] == $idTurnament) {
6. if($row['team\_1']==$idTeam||$row['team\_2']==$idTeam)
7. {
8. $join = true;
9. }
10. }
11. }
12. if (!$join) {
13. $randomBagian = array();
14. if ($slot == 2) {
15. array\_push($randomBagian, "finalteam1");
16. array\_push($randomBagian, "finalteam2");
17. } else {
18. $ctr = $slot / 4;
19. for ($i = 1; $i <= $ctr; $i++) {
20. for ($j = 1; $j <= 2; $j++) {
21. if ($slot == 4) {
22. array\_push($randomBagian,
23. "semifinalkiriteam".$j);
24. array\_push($randomBagian,
25. "semifinalkananteam".$j);
26. } else if ($slot == 8) {
27. array\_push($randomBagian,
28. "quarterfinalkiri" .$i . "team" . $j);
29. array\_push($randomBagian,
30. "quarterfinalkanan". $i . "team" . $j);
31. } else if ($slot == 16 || $slot == 32) {
32. array\_push($randomBagian, "round1kiri" .
33. $i."team" . $j);
34. array\_push($randomBagian, "round1kanan"
35. . $i."team" . $j);
36. }
37. }
38. }
39. }
40. $query = $this->db->query("select \* from pertandingan");
41. foreach ($query->result\_array() as $row) {
42. if ($row['id\_turnament'] == $idTurnament) {
43. for ($i = 0; $i < $slot; $i++) {
44. if($randomBagian[$i]==$row['bagian'].
45. "team1") {
46. if ($row['team\_1'] != "") {
47. $randomBagian[$i] = "";
48. }
49. }
50. if($randomBagian[$i]==$row['bagian'].
51. "team2") {
52. if ($row['team\_2'] != "") {
53. $randomBagian[$i] = "";
54. }
55. }

**Segmen Program 4.21 : (Lanjutan)**

1. }
2. }
3. }
4. $ada = false;
5. $bagian = "";
6. while ($ada == false) {
7. $rand = rand(0, $slot - 1);
8. if ($randomBagian[$rand] != "") {
9. $bagian = $randomBagian[$rand];
10. $ada = true;
11. }
12. }
13. $tim = substr($bagian, -5);
14. $idx = strpos($bagian, "team");
15. $bagian2 = substr($bagian, 0, $idx);
16. if ($tim == "team1") {
17. echo "masuk1";
18. echo $bagian2;
19. $query = "update pertandingan set team\_1 = '" .
20. $idTeam ."' where bagian = '" . $bagian2 . "' and
21. id\_turnament='" .
22. $idTurnament . "' ";
23. $this->db->query($query);
24. } else if ($tim == "team2") {
25. echo "masuk2";
26. echo $bagian2;
27. $query = "update pertandingan set team\_2 = '" .
28. $idTeam ."' where bagian = '" . $bagian2 . "' and
29. id\_turnament='" .$idTurnament . "' ";
30. $this->db->query($query);
31. }
32. //TAMBAH JUMLAH PEMAIN
33. $query = $this->db->query("select \* from tournament");
34. foreach ($query->result\_array() as $row) {
35. if ($row['id\_turnament'] == $idTurnament) {
36. $jml = $row['jumlah\_pemain'];
37. }
38. }
39. $jml += 1;
40. $data = [
41. "jumlah\_pemain" => $jml
42. ];
43. $this->db->where('id\_turnament', $idTurnament);
44. $this->db->update('tournament', $data);
45. //UPDATE STATUS FULL (1)
46. $query = $this->db->query("select \* from tournament");
47. foreach ($query->result\_array() as $row) {
48. if ($row['jumlah\_pemain'] == $row['jumlah\_slot'] &&
49. $row['id\_turnament'] == $idTurnament) {
50. $data = [
51. "status" => 1
52. ];
53. $this->db->where('id\_turnament', $idTurnament);
54. $this->db->update('tournament', $data);
55. }

**Segmen Program 4.21 : (Lanjutan)**

1. }
2. redirect('Community/refreshTournament');
3. } else {
4. echo "joined";
5. }

Segmen program 4.21 digunakan agar user bisa mengikuti turnamen. Terlihat pada baris pertama hingga ke 3 (tiga), program mengambil input id turnamen, id team, dan jumlah slot. Kemudian pada baris ke 4 (empat) hingga baris ke 14 (empat belas), program melakukan pengecekan apakah pengguna sudah mengikuti pertandingan atau belum. Jika belum, maka program akan mengacak letak pengguna di dalam bagan turnamen. Terlihat pada baris ke 15 (lima belas) hingga baris ke 40 (empat puluh), program membuat array yang berisi bagan – bagan dalam turnamen. Baris 70 (tujuh puluh) hingga 88 (delapan puluh delapan) menunjukkan bagian program yang berfungsi untuk pembagian team dalam pertandingan serta merubah nama team dalam tabel. Baris 90 (sembilan puluh) hingga 101 (seratus satu) menunjukkan bagian program yang berfungsi untuk menambah jumlah pemain dalam tabel. Baris 103 (seratus tiga) hingga 113 (seratus tiga belas) merupakan bagian program yang berguna untuk mengganti status pertandingan dalam tabel.

**4.2 Merchant**

Pada subbab ini akan membahas tentang segmen program yang ditujukan kepada penjual atau pemilik toko yang menjualkan barang-barang ke dalam website gather.owl. Website gather.owl mempunyai 7 fitur yang dapat dinikmati untuk setiap pembuat toko. Beberapa fitur-fitur, seperti insert merchant, edit merchant, insert item, remove item, edit item, insert *subscriber*, edit *banner* telah terbagi menjadi 2 bagian yang sama dengan user, yaitu bagian model dan controller. Bagian model menghubungkan dari program dengan database db\_owl. Bagian controller berkaitan dengan proses berjalannya website yang dioperasikan oleh pengguna.

**Segmen Program 4.22 : Controller Insert Merchant**

1. $this->Merchant\_model->insertMerchant();

Segmen program 4.22 merupakan segmen program bentuk controller yang berfungsi untuk memasukkan toko yang dibuat oleh penjual dalam website gather.owl. Program memanggil model dari merchant (penjual) dengan fungsi yang bernama insertMerchant yang ditunjukkan pada baris pertama. Setelah controller memanggil fungsi tersebut, maka program melakukan pengecekan dan penambahan di dalam tabel merchant melalui fungsi tersebut. Fungsi insertMerchant akan dijelaskan setiap bagian program pada segmen program 4.23.

**Segmen Program 4.23 : Model Insert Merchant**

1. $id = $this->session->userdata('id\_user');
2. $ctr = 1;
3. $query = $this->db->query("select \* from merchant");
4. $newId = $this->input->post('name');
5. $foto = $this->input->post('foto');
6. $foto = base64\_decode($foto);
7. $cekNewId = 'M' . substr(strtoupper($newId), 0, 1);
8. foreach ($query->result\_array() as $row) {
9. $cekId = substr(strtoupper($row['id\_merchant']), 0, 2);
10. if ($cekId == $cekNewId) {
11. $ctr++;
12. }
13. }
14. if ($ctr < 10) {
15. $generateId = $cekNewId . '000' . $ctr;
16. } else if ($ctr < 100) {
17. $generateId = $cekNewId . '00' . $ctr;
18. } else if ($ctr < 1000) {
19. $generateId = $cekNewId . '0' . $ctr;
20. } else {
21. $generateId = $cekNewId . $ctr;
22. }
23. $data = [
24. "id\_merchant" => $generateId,
25. "id\_user" => $id,
26. "nama\_merchant" => $this->input->post('name'),
27. "bio" => $this->input->post('desc'),
28. "foto\_profil" => $foto
29. ];
30. $this->db->insert('merchant', $data);

Segmen Program 4.23 merupakan segmen program model yang digunakan untuk menambahkan merchant yang diperlukan ke dalam database. Bagian program 4.23 memerlukan id merchant yang bersifat unik. Baris 7 (tujuh) hingga 22 (dua puluh dua) menunjukkan program membuat id baru dengan memeriksa semua id merchant yang ada di dalam tabel merchant untuk menghindari kesamaan id. Baris 23 (dua puluh tiga) hingga 29 (dua puluh sembilan) menunjukkan program menyusun data merchant untuk ditambahkan ke dalam tabel merchant sesuai dengan baris yang ada di dalam tabel, serta pada baris 30 (tiga puluh) menunjukkan program mengirim seluruh data yang ada di baris 23 (dua puluh tiga) hingga 29 (dua puluh sembilan) ke dalam database.

**Segmen Program 4.24 : Controller Edit Merchant**

1. $this->Merchant\_model->editMerchant();

Segmen program 4.24 merupakan segmen program bentuk controller yang berfungsi untuk merubah data toko dalam website gather.owl. Program memanggil model dari merchant dengan fungsi yang bernama editMerchant yang ditunjukkan pada baris pertama. Setelah controller memanggil fungsi tersebut, maka program melakukan pengecekan dan perubahan di dalam tabel merchant melalui fungsi tersebut. Fungsi editMerchant akan dijelaskan setiap bagian program pada segmen program 4.25.

**Segmen Program 4.25 : Model Edit Merchant**

1. $id = $this->input->post('id');
2. $foto = $this->input->post('foto');
3. $foto = base64\_decode($foto);
4. $data = [
5. "nama\_merchant" => $this->input->post('name'),
6. "bio" => $this->input->post('desc'),
7. "foto\_profil" => $foto
8. ];
9. $this->db->where('id\_merchant', $id);
10. $this->db->update('merchant', $data);

Segmen Program 4.25 merupakan segmen program model yang digunakan untuk merubah data merchant yang ada di dalam database. Baris pertama dan baris 2 (dua) menunjukkan program menyimpan data yang di masukkan oleh pembuat toko dalam sebuah variabel, data yang dimasukkan oleh pembuat antara lain id dan foto. Baris 4 (empat) hingga 8 (delapan) menunjukkan program menyusun data merchant yang berisi masukan dari pembuat untuk dirubah ke dalam tabel merchant. Baris 9 (sembilan) merupakan bagian program yang berfungsi untuk menentukan merchant yang akan diubah datanya sesuai dengan masukan dari pembuat. Baris 10 (sepuluh) mengirim seluruh data yang ada di baris 4 (empat) hingga 8 (delapan) ke dalam database.

**Segmen Program 4.26 : Controller Insert Subscribers**

1. $this->Subs\_model->insertSubs();

Segmen program 4.26 merupakan segmen program bentuk controller yang berfungsi untuk memasukkan data pembeli yang telah berlangganan dengan toko dalam website gather.owl. Program memanggil model dari subs (subscriber) dengan fungsi yang bernama insertSubs yang ditunjukkan pada baris pertama. Setelah controller memanggil fungsi tersebut, maka program akan melakukan pengecekan dan penambahan di dalam tabel subscribers melalui fungsi tersebut. Fungsi insertSubs akan dijelaskan setiap bagian program pada segmen program 4.27.

**Segmen Program 4.27 : Model Insert Subscribers**

1. $ctr = 1;
2. $query = $this->db->query("select \* from subscribers");
3. $cekNewId = 'SU';
4. foreach ($query->result\_array() as $row) {
5. $cekId = substr(strtoupper($row['id\_sub']), 0, 2);
6. if ($cekId == $cekNewId) {
7. $ctr++;
8. }
9. }
10. if ($ctr < 10) {
11. $generateId = $cekNewId . '000' . $ctr;
12. } else if ($ctr < 100) {
13. $generateId = $cekNewId . '00' . $ctr;
14. } else if ($ctr < 1000) {
15. $generateId = $cekNewId . '0' . $ctr;
16. } else {
17. $generateId = $cekNewId . $ctr;
18. }
19. $tglAwal = date("Y-m-d H:i:s");
20. $tglAkhir = date('Y-m-d H:i:s', strtotime('+31 days',
21. strtotime($tglAwal)));
22. $foto = $this->input->post('foto');
23. $foto = base64\_decode($foto);
24. $data = [
25. "id\_sub" => $generateId,
26. "id\_merchant" => $this->input->post('id\_merchant'),
27. "banner" => $foto,
28. "tgl\_akhir" => $tglAkhir,
29. "status" => 1

**Segmen Program 4.27 : (Lanjutan)**

1. ];
2. $this->db->insert('subscribers', $data);

Segmen Program 4.27 merupakan segmen program model yang digunakan untuk menambahkan data pengguna yang berkeinginan untuk berlangganan dengan toko ke dalam database. Bagian program 4.27 memerlukan id sub yang bersifat unik. Baris pertama hingga 18 (delapan belas) menunjukkan program pembuatan id baru dengan memeriksa semua id sub yang ada di dalam tabel subscribers untuk menghindari kesamaan id. Sebelum program menyusun data yang diperlukan, setiap data yang merupakan masukan dari pengguna akan disimpan dalam varibel secara terpisah yang ditunjukkan pada baris 19 (sembilan belas) hingga 23 (dua puluh tiga). Baris 24 (dua puluh empat) hingga 30 (tiga puluh) menunjukkan program menyusun data pelanggan untuk ditambahkan ke dalam tabel subscribers sesuai dengan baris yang ada di dalam tabel. Baris 31 (tiga puluh satu) menunjukkan program mengirim seluruh data yang ada pada 24 (dua puluh empat) hingga 30 (tiga puluh) ke dalam database.

**Segmen Program 4.28 : Controller Edit Banner**

1. $this->Subs\_model->editBanner();

Segmen program 4.28 merupakan segmen program bentuk controller yang berfungsi untuk merubah data *banner* yang dimiliki oleh penjual dalam website gather.owl. Program memanggil model dari subs (subscribers) dengan fungsi yang bernama editBanner yang ditunjukkan pada baris pertama. Setelah controller memanggil fungsi tersebut, maka program melakukan pengecekan dan penambahan di dalam tabel subscribers melalui fungsi tersebut. Fungsi editBanner akan dijelaskan setiap bagian program pada segmen program 4.29.

**Segmen Program 4.29 : Model Edit Banner**

1. $idSub = $this->input->post('id\_sub');
2. $foto = $this->input->post('foto');
3. $foto = base64\_decode($foto);
4. $data = [
5. "banner" => $foto
6. ];

**Segmen Program 4.29 : (Lanjutan)**

1. $this->db->where('id\_sub', $idSub);
2. $this->db->update('subscribers', $data);

Segmen Program 4.29 merupakan segmen program model yang digunakan untuk merubah data banner yang ada di dalam tabel subscribers. Baris pertama dan baris 2 (dua) menunjukkan program menyimpan data yang di masukkan oleh pembuat toko dalam sebuah variabel, data yang dimasukkan antara lain id dan foto banner. Baris 4 (empat) hingga 6 (enam) menunjukkan program menyusun data perubahan banner untuk dirubah ke dalam tabel subscribers. Baris 7 (tujuh) merupakan bagian program yang berfungsi untuk menentukan pelanggan yang akan diubah datanya. Baris 8 (delapan) mengirim seluruh data yang ada di baris 4 (empat) hingga 6 (enam) ke dalam database.

**Segmen Program 4.30 : Controller Update Status Transaksi Item Merchant**

1. $this->TransItem\_model->changeStatus($status);

Segmen program 4.30 merupakan segmen program bentuk controller yang berfungsi untuk merubah data status yang berada dalam transaksi\_item dalam website gather.owl. Program memanggil model dari TransItem (transaksi\_item) dengan fungsi yang bernama changeStatus yang ditunjukkan pada baris pertama. Setelah controller memanggil fungsi tersebut, maka program melakukan pengecekan dan perubahan pada bagian status di dalam tabel transaksi\_item melalui fungsi tersebut. Fungsi changeStatus akan dijelaskan setiap bagian program pada segmen program 4.31.

**Segmen Program 4.31 : Model** **Update Status Transaksi Item Merchant**

1. $idTrans = $this->input->post('id\_transaksi');
2. $idItem = $this->input->post('id\_item');
3. $ket = $this->input->post('keterangan');
4. $harga = $this->input->post('harga');
5. $foto = $this->input->post('foto');
6. $foto = base64\_decode($foto);
7. if ($status == 1) {
8. $data = [
9. "status" => $status,
10. "keterangan" => "Awaiting admin's approval",
11. "foto" => $foto

**Segmen Program 4.31 : (Lanjutan)**

1. ];
2. } else if ($status == -1) {
3. $data = [
4. "status" => $status,
5. "keterangan" => "Cancel by User : " . $ket
6. ];
7. $query = $this->db->query("select \* from transaksi");
8. foreach ($query->result\_array() as $row) {
9. if ($row['id\_transaksi'] == $idTrans) {
10. $query2 = $this->db->query("select \* from user");
11. foreach ($query2->result\_array() as $row2) {
12. if ($row2['id\_user'] == $row['id\_user']) {
13. $cashback = 0;
14. $hargaM = $harga;
15. if ($row['cashback'] != 0) {
16. $promo = $row['Gross\_Amount'];
17. $promo = $promo / $row['cashback'];
18. $potongan = $harga \* $promo / 100;
19. $grandtotal = $row['Gross\_Amount'];
20. $grandtotal -= $harga;
21. $cachback = $row['cashback'];
22. $cachback -= $potongan;
23. $data2 = [
24. "Gross\_Amount" => $grandtotal,
25. "cashback" => $cachback
26. ];
27. $this->db->where('id\_transaksi',
28. $row['id\_transaksi']);
29. $this->db->update('transaksi',
30. $data2);
31. $harga -= $potongan;
32. } else {
33. $grandtotal = $row['Gross\_Amount'];
34. $grandtotal -= $harga;
35. $data2 = [
36. "Gross\_Amount" => $grandtotal
37. ];
38. $this->db->where('id\_transaksi',
39. $row['id\_transaksi']);
40. $this->db->update('transaksi',
41. $data2);
42. }
43. $saldo = $row2['saldo'];
44. $saldo += $harga;
45. //UPDATE SALDO USER YANG BELI
46. $data3 = [
47. "saldo" => $saldo
48. ];
49. $this->db->where('id\_user',
50. $row2['id\_user']);
51. $this->db->update('user',$data3);
52. }
53. }
54. }
55. }

**Segmen Program 4.31 : (Lanjutan)**

1. }

Segmen Program 4.31 merupakan segmen program model yang digunakan

untuk merubah data banner yang ada di dalam tabel subscribers. Baris pertama dan baris 2 (dua) menunjukkan program menyimpan data yang di masukkan oleh pembuat toko dalam sebuah variabel, data yang dimasukkan antara lain id dan foto banner. Baris 4 (empat) hingga 6 (enam) menunjukkan program menyusun data perubahan banner untuk dirubah ke dalam tabel subscribers. Baris 7 (tujuh) merupakan bagian program yang berfungsi untuk menentukan pelanggan yang akan diubah datanya. Baris 8 (delapan) mengirim seluruh data yang ada di baris 4 (empat) hingga 6 (enam) ke dalam database.

**Segmen Program 4.32 : Controller Insert Item**

1. $this->Item\_model->insertItem();

Segmen program 4.32 berfungsi dalam penambahan tabel item. Program memanggil model item dengan fungsi yang bernama insertItem. Controller segmen 4.32 akan dipanggil jika merchant ingin menambahkan item miliknya. Setelah controller memanggil fungsi tersebut, maka akan dilakukan pengecekan dan penambahan di dalam tabel item.

**Segmen Program 4.33 : Model Insert Item**

1. $ctr = 1;
2. $query = $this->db->query("select \* from item");
3. $newId = $this->input->post('name');
4. $cekNewId = 'I'.substr(strtoupper($newId), 0, 1);
5. foreach ($query->result\_array() as $row) {
6. $cekId = substr(strtoupper($row['id\_item']), 0, 2);
7. if ($cekId == ekNewId) {
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. $tgl = date("Y-m-d H:i:s");
21. $foto = $this->input->post('foto');

**Segmen Program 4.33 : (Lanjutan)**

1. $foto = base64\_decode($foto);
2. $harga = $this->input->post('price');
3. //BIAYA ADMIN
4. $admin = 1000;
5. $harga += $admin;
6. $data = [
7. "id\_item" => $generateId,
8. "id\_merchant" => $this->input->post('id\_merchant'),
9. "id\_game" => $this->input->post('id\_game'),
10. "nama\_item" => $this->input->post('name'),
11. "desc\_item" => $this->input->post('desc'),
12. "harga\_item" => $harga,
13. "tanggal\_upload" => $tgl,
14. "jumlah\_item" => $this->input->post('stok'),
15. "foto\_item" => $foto
16. ];
17. $this->db->insert('item', $data);

Segmen program 4.33 berfungsi dalam penambahan isi tabel item. Terlihat pada baris pertama hingga ke 19 (sembilan belas), program membuat id baru untuk item yang akan ditambahkan. Kemudian pada baris ke 20 (dua puluh) hingga ke 26 (dua puluh enam), program mengambil foto yang telah diinputkan dan tanggal pada saat item ditambahkan. Pada baris ke 24 (dua puluh empat) hingga ke 26 (dua puluh enam), program menambahkan 1000 (seribu) ke harga item sebagai biaya administrasi. Selanjutnya pada baris ke 28 (dua puluh delapan) hingga baris ke 38 (tiga puluh delapan), program mengumpulkan dan menyimpan seluruh data yang dibutuhkan ke dalam variabel yang bernama data dan menambahkannya ke dalam tabel item.

**Segmen Program 4.34 : Controller Remove Item**

1. $this->Item\_model->removeItem();

Segmen program 4.34 berfungsi dalam penghapusan isi tabel item. Program memanggil model item dengan fungsi yang bernama remove Item. Controller segemn 4.34 akan dipanggil jika merchant ingin menghapus item miliknya. Setelah controller memanggil fungsi tersebut, maka akan dilakukan penghapusan item milik merchant di dalam tabel item.

**Segmen Program 4.35 : Model Remove Item**

1. $idI = $this->input->post('idItem');
2. $query = "delete from item where id\_item = '". $idI. "'
3. ";
4. $this->db->query($query);

Segmen program 4.35 merupakan bagian program yang bertugas untuk menghapus data di dalam tabel item. Model segmen program 4.35 akan dipanggil ketika merchant ingin menghapus item miliknya. Pada baris pertama, program mengambil id item yang akan dihapus. Kemudian pada baris ke 2 (dua) dan ke 3 (tiga), program membuat kalimat untuk proses penghapusan data pada tabel dengan id yang sudah diambilnya tadi. Pada baris ke 4 (empat), kalimat tersebut dieksekusi untuk menghapus item.

**Segmen Program 4.36 : Controller Edit Item**

1. $this->Item\_model->editItem();

Segmen program 4.36 berfungsi dalam perubahan isi tabel item. Program memanggil model item dengan fungsi yang bernama edit Item. Controller segmen program 4.36 akan dipanggil jika merchant ingin mengganti detail item miliknya. Setelah controller memanggil fungsi tersebut, maka akan dilakukan perubahan detail item milik merchant di dalam tabel item.

**Segmen Program 4.37 : Model Edit Item**

1. $id = $this->input->post('id');
2. $data = [
3. "id\_game" => $this->input->post('id\_game'),
4. "nama\_item" => $this->input->post('name'),
5. "desc\_item" => $this->input->post('desc'),
6. "harga\_item" => $this->input->post('price'),
7. "jumlah\_item" => $this->input->post('stok')
8. ];
9. $this->db->where('id\_item', $id);
10. $this->db->update('item', $data);

Segmen program 4.34 merupakan bagian program yang bertugas untuk mengubah detail item pada tabel item. Pada baris pertama, program mengambil id item yang detailnya diubah oleh merchant. Kemudian pada baris ke 2 (dua) hingga ke 8 (delapan), program mengumpulkan dan menyiapkan data detail item untuk mengubah detail item yang lama menjadi baru. Selanjutnya, pada baris ke 9 (sembilan) hingga ke 10 (sepuluh), program melakukan proses perubahan dengan id item yang sudah disiapkan diambil sebelumnya.

**4.3 Admin**

Pada subbab ini membahas tentang segmen program yang ditujukan kepada admin dari website gather.owl. Website gather.owl mempunyai 7 fitur yang dapat dilakukan oleh admin website gather.owl. Beberapa fitur-fitur, seperti insert sub, *change* status transaksi item, insert promo, check status transaksi telah terbagi menjadi 2 bagian sama dengan user dan merchant, yaitu bagian model dan controller. Bagian model menghubungkan dan mengirim data dari program admin dengan database. Bagian controller berkaitan dengan proses berjalannya website admin yang dioperasikan oleh admin.

**Segmen Program 4.38 : Controller Admin Insert Sub**

1. $this->load->model('Sub\_model');
2. $data['merchant'] = $this->Merchant\_model
3. ->getAllMerchant();
4. $this->form\_validation->set\_rules('photoBanner',
5. 'Banner','required');
6. $this->form\_validation->set\_rules('tglAkhir', 'Tanggal
7. Akhir', 'required');
8. if ($this->form\_validation->run() == FALSE) {
9. $this->load->view('templates/header');
10. $this->load->view('templates/navbar');
11. $this->load->view('templates/sidebar');
12. $this->load->view('sub/insertSub', $data);
13. } else
14. $this->Sub\_model->insertSub();
15. $this->session->set\_flashdata('flash','Success
16. Insert Subscription !!!');
17. redirect('sub/listSub');
18. }

Segmen program 4.38 berfungsi untuk mengisi tabel *Subscription.* Terlihat pada baris pertama dan ke 2 (dua), program memanggil model subsciption dan model merchant. Selanjutnya pada baris ke 4 (empat) hingga ke 12 (dua belas), program melakukan pengecekan input pengguna. Selanjutnya pada baris ke 13 (tiga belas) sampai dengan ke 16 (enam belas), program melakukan penambahan kedalam tabel subscription dengan memanggil model yang bernama *insertSub* lalu menampilkan tabel subscription.

**Segmen Program 4.39 : Model Admin Insert Sub**

1. $ctr = 1;
2. $query= $this->db->query("select \* from subscribers");
3. $cekNewId = 'SU';
4. foreach ($query->result\_array() as $row) {
5. $cekId = substr(strtoupper($row['id\_sub']), 0, 2);
6. if ($cekId == $cekNewId) {
7. $ctr++;
8. }
9. }
10. if ($ctr < 10) {
11. $generateId = $cekNewId . '000' . $ctr;
12. } else if ($ctr < 100) {
13. $generateId = $cekNewId . '00' . $ctr;
14. } else if ($ctr < 1000) {
15. $generateId = $cekNewId . '0' . $ctr;
16. } else {
17. $generateId = $cekNewId . $ctr;
18. }
19. $tglAkhir = $this->input->post('tglAkhir');
20. $tglAkhir = date('Y-m-d H:i:s', strtotime($tglAkhir));
21. $foto = $this->input->post('photoBanner');
22. $foto = base64\_decode($foto);
23. $data = [
24. "id\_sub" => $generateId,
25. "id\_merchant" => $this->input->post('idMerchant'),
26. "banner" => $foto,
27. "tgl\_akhir" => $tglAkhir
28. ];
29. $this->db->insert('subscribers', $data);

Segmen program 4.39 bertujuan untuk mengisi tabel subscription. Pada baris pertama hingga ke 18 (delapan belas), program membuat id baru. Kemudian pada baris ke 19 (sembilan belas) hingga baris ke 22 (dua puluh dua), program mengambil tanggal bulan depan dan foto dari input pengguna. Selanjutnya pada baris ke 23 (dua puluh tiga) hingga 29 (dua puluh sembilan), program menambahkan data ke dalam tabel subscription.

**Segmen Program 4.40 : Controller Admin Insert Promo**

1. $this->form\_validation->set\_rules('kodepromo', 'Kode
2. Promo', 'required');
3. $this->form\_validation->set\_rules('potongan', 'Potongan',
4. 'required');
5. $this->form\_validation->set\_rules('potongan', 'Potongan',

**Segmen Program 4.40 : (Lanjutan)**

1. 'numeric');
2. $this->form\_validation->set\_rules('maksimal', 'Maksimal',
3. 'required');
4. $this->form\_validation->set\_rules('tglKadaluwarsa',
5. 'Tanggal Kadaluwarsa', 'required');
6. if ($this->form\_validation->run() == FALSE) {
7. $this->load->view('templates/header');
8. $this->load->view('templates/navbar');
9. $this->load->view('templates/sidebar');
10. $this->load->view('promo/insertPromo');
11. } else {
12. $this->Promo\_model->insertPromo();
13. $this->session->set\_flashdata('flash', 'Success Insert
14. Promo !!!');
15. redirect('Promo');
16. }

Segmen program 4.40 bertujuan untuk memanggil seluruh model yang diperlukan dalam pengisian tabel promo. Pada baris pertama hingga ke 10 (sepuluh), program melakukan pengecekan input pengguna. Jika validasi input pengguna salah, maka baris ke 12 (dua belas) hingga ke 15 (lima belas) akan dieksekusi dan akan dikembalikan ke halaman semula. Sebaliknya, data input pengguna akan ditambahkan kedalam tabel promo dengan memanggil segmen program 4.41.

**Segmen Program 4.41 : Model Admin Insert Promo**

1. $ctr = 1;
2. $query = $this->db->query("select \* from promo");
3. $newId = $this->input->post('kodepromo');
4. $cekNewId = 'V' . substr(strtoupper($newId), 0, 1);
5. foreach ($query->result\_array() as $row) {
6. $cekId = substr(strtoupper($row['id\_promo']),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. }

**Segmen Program 4.41 : (Lanjutan)**

1. $tgl = $this->input->post('tglKadaluwarsa');
2. $tglAwal = substr($tgl, 0, 10);
3. $tglAkhir = substr($tgl, 14);
4. $newTglAwal = date("Y-m-d H:i:s",strtotime($tglAwal));
5. $newTglAkhir=date("Y-m-d H:i:s",strtotime($tglAkhir));
6. // $tgl = date("Y-m-d H:i:s");
7. $data = [
8. "id\_promo" => $generateId,
9. "kodepromo" => $this->input->post('kodepromo'),
10. "tanggal\_aktif" => $newTglAwal,
11. "tanggal\_kadaluwarsa" => $newTglAkhir,
12. "potongan" => $this->input->post('potongan'),
13. "maksimal" => $this->input->post('maksimal')
14. ];
15. $this->db->insert('promo', $data);

Segmen program 4.41 bertujuan untuk pengisian tabel promo. Pada baris pertama hingga ke 19 (sembilan belas), program membuat id baru. Kemudian baris ke 20 (dua puluh) hingga ke 24 (dua puluh empat), program mengambil input tanggal pengguna dan mengganti format sesuai tabel. Pada baris ke 26 (dua puluh enam) hingga 34 (tiga puluh empat), program menambahkan data ke dalam tabel promo.

**Segmen Program 4.42 : Controller Admin Change Status Transaksi Item**

1. $this->TransItem\_model->changeStatus($status);
2. redirect("Transaksi");

Segmen program 4.42 bertujuan untuk mengganti status transaksi item. Program memanggil model transaksi item dengan judul fungsi *changeStatus*. Pemanggilan tersebut akan dilakukan jika admin ingin mengganti status dari pending menjadi sukses atau sebaliknya. Setelah pemanggilan, admin akan dikembalikan ke halaman transaksi.

**Segmen Program 4.43 : Model Admin Change Status Transaksi Item**

1. $idTrans = $this->input->post('id\_transaksi');
2. $idItem = $this->input->post('id\_item');
3. $harga = $this->input->post('harga');
4. $jumlah = $this->input->post('jumlah');
5. if ($status == 2) {
6. $data = [

**Segmen Program 4.43 : (Lanjutan)**

1. "status" => $status,
2. "keterangan" => "Success"
3. ];
4. //TAMBAH SALDO MERCHANT
5. $query2 = $this->db->query("select \* from item");
6. foreach ($query2->result\_array() as $row2) {
7. if ($row2['id\_item'] == $idItem) {
8. //UPDATE AMOUNT ITEM
9. $amount = $row2['jumlah\_item'];
10. $amount -= $jumlah;
11. $data2 = [
12. "jumlah\_item" => $amount
13. ];
14. //echo $amount;
15. $this->db->where('id\_item', $row2['id\_item']);
16. $this->db->update('item', $data2);
17. $query3 = $this->db->query("select \* from
18. merchant");
19. foreach ($query3->result\_array() as $row3) {
20. if ($row3['id\_merchant'] ==
21. $row2['id\_merchant']) {
22. $query4 = $this->db->query("select \*
23. from user");
24. foreach ($query4->result\_array() as
25. $row4) {
26. if ($row4['id\_user'] ==
27. $row3['id\_user']) {
28. //ADMIN
29. $admin = 1000;
30. $harga -= $admin;
31. $saldo = $row4['saldo'];
32. $saldo += $harga;
33. // echo $saldo;
34. $queryUpdate = "update user
35. set saldo = '" . $saldo . "'
36. where id\_user = '" .
37. $row4['id\_user'] . "' ";
38. $this->db
39. ->query($queryUpdate);
40. }
41. }
42. }
43. }
44. }
45. }
46. } else if ($status == -2) {
47. $data = [
48. "status" => $status,
49. "keterangan" => "Cancel by Admin"
50. ];

**Segmen Program 4.43 : (Lanjutan)**

1. $query=$this->db->query("select \* from
2. transaksi");
3. foreach ($query->result\_array() as $row) {
4. if ($row['id\_transaksi'] == $idTrans) {
5. $query2 = $this->db->query("select \* from
6. user");
7. foreach($query2->result\_array() as $row2){
8. if ($row2['id\_user'] ==
9. $row['id\_user']) {
10. $cashback = 0;
11. $hargaM = $harga;
12. if ($row['cashback'] != 0) {
13. $promo = $row['Gross\_Amount'];
14. $promo = $promo /
15. $row['cashback'];
16. $potongan = $harga\*$promo/100;
17. $grandtotal =
18. $row['Gross\_Amount'];
19. $grandtotal -= $harga;
20. $cachback = $row['cashback'];
21. $cachback -= $potongan;
22. //UPDATE GRANDTOTAL DAN
23. CASHBACK DI TRANSAKSI
24. $data2 = [
25. "Gross\_Amount" =>
26. $grandtotal,
27. "cashback" => $cachback
28. ];
29. $this->db->where
30. ('id\_transaksi',
31. $row['id\_transaksi']);
32. $this->db->update('transaksi',
33. $data2);
34. //UPDATE SALDO USER YANG BELI
35. $harga -= $potongan;
36. } else {
37. $grandtotal =
38. $row['Gross\_Amount'];
39. $grandtotal -= $harga;
40. //UPDATE GRANDTOTAL DAN
41. CASHBACK DI TRANSAKSI
42. $data2 = [
43. "Gross\_Amount" =>
44. $grandtotal
45. ];
46. $this->db
47. ->where('id\_transaksi',
48. $row['id\_transaksi']);
49. $this->db->update('transaksi',
50. $data2);

**Segmen Program 4.43 : (Lanjutan)**

1. }
2. $saldo = $row2['saldo'];
3. $saldo += $harga;
4. //UPDATE SALDO USER YANG BELI
5. $data3 = [
6. "saldo" => $saldo
7. ];
8. $this->db->where('id\_user',
9. $row2['id\_user']);
10. $this->db->update('user', $data3);
11. }
12. }
13. }
14. }
15. }
16. $this->db->where('id\_transaksi', $idTrans);
17. $this->db->where('id\_item', $idItem);
18. $this->db->update('transaksi\_item', $data);

Segmen program 4.43 berfungsi untuk mengganti status dan melakukan perubahan dalam tabel sesuai dengan status yang dirubah. Pada baris pertama hingga ke 4 (empat), program mengambil seluruh input yang tersedia. Pada baris ke 5 (lima) hingga ke 43 (empat puluh tiga), program akan menambahkan saldo admin dan saldo merchant jika status yang diinput adalah 2(dua). Sebaliknya pada baris ke 45 (empat puluh lima) hingga baris ke 95 (sembilan puluh lima), program akan mengembalikan saldo kepada pembeli. Kemudian pada baris ke 98 (sembilan puluh delapan), program memperbarui tabel item transaksi.

**Segmen Program 4.44 : Model Admin Check Status Transaksi**

1. $change = true;
2. $query= $this->db->query("select \* from transaksi\_item
3. where id\_transaksi = '" . $idTransaksi . "' ");
4. foreach ($query->result\_array() as $row) {
5. if ($row['status'] == 0 || $row['status'] == 1) {
6. $change = false;
7. }
8. }
9. if ($change) {
10. $status = -1;
11. $query= $this->db->query("select \* from
12. transaksi\_item where id\_transaksi =
13. '" . $idTransaksi . "' ");
14. foreach ($query->result\_array() as $row2) {
15. if ($row2['status'] == 2) {

**Segmen Program 4.44 : (Lanjutan)**

1. $status = 1;
2. }
3. }
4. $data = [
5. "status" => $status
6. ];
7. $this->db->where('id\_transaksi', $idTransaksi);
8. $this->db->update('transaksi', $data);
9. if ($status == -1) {
10. //echo "masuk";
11. $total = 0;
12. $query= $this->db->query("select \* from
13. transaksi\_item where id\_transaksi =
14. '" . $idTransaksi . "' ");
15. foreach ($query->result\_array() as $row3) {
16. $total += $row3['subtotal'];
17. }
18. $query = $this->db->query("select \* from
19. transaksi");
20. foreach ($query->result\_array() as $row3) {
21. if ($idTransaksi == $row3['id\_transaksi'])
22. {
23. $query = $this->db->query("select \*
24. from promo");
25. foreach ($query->result\_array() as
26. $row4) {
27. if ($row3['id\_promo'] ==
28. $row4['id\_promo']) {
29. $potongan = $row3['potongan'];
30. }
31. }
32. }
33. }
34. $cashback = $total \* $potongan / 100;
35. $data = [
36. "Gross\_Amount" => $total,
37. "cashback" => $cashback
38. ];
39. $this->db->where('id\_transaksi',$idTransaksi);
40. $this->db->update('transaksi',$data);
41. }
42. }

Segmen program 4.44 berfungsi untuk mengecek seluruh status item dalam transaksi. Pada baris pertama hingga ke 6 (enam), program mengecek apakah masih ada transaksi yang belum selesai atau sudah selesai. Jika semua item dibatalkan maka status transaksi akan diganti menjadi -1 (minus satu) yang berarti transaksi dibatalkan sesuai dengan baris ke 8 (delapan). Tetapi jika ada salah satu saja yang status transaksi itemnya sukses, maka status transaksi akan diganti sukses sesuai dengan baris ke 10 (sepuluh) hingga ke 16 (enam belas). Setelah mengganti status, program akan mengembalikan uang pembeli dan mengambil cashback jika transaksi dibatalkan atau tidak berhasil sesuai dengan baris ke 22 (dua puluh dua) hingga baris ke 45 (empat puluh lima). Pada baris ke 47 (empat puluh tujuh), program memperbarui data tabel transaksi.