**Dokumentasi Coding Website**

Pada dokumentasi ini, akan dibahas mengenai coding / syntax di website. Terdapat 3 halaman pada website ini, yaitu halaman Master Barang, halaman Tambah Barang, dan halaman Ubah Barang. Halaman Default pada website ini adalah halaman Master Barang.

**Fitur Menampilkan Data di halaman Master Barang**

Secara default, data yang ditampilkan di halaman Master Barang, hanya produk yang memiliki status “Bisa Dijual”. Data yang sudah dipilih, ditampilkan ke dalam datatable.

User juga dapat mengganti data yang ditampilkan menjadi semua data, dengan drop down di bawah tombol “Add barang”.

Dibawah ini merupakan function refresh\_table dipanggil ketika halaman Master Barang pertama kali di load. Baris 4 digunakan untuk load function refresh\_barang di controller home. Baris 5 - 14 digunakan untuk menginisiasikan datatables beserta pengaturannya.

1. function refresh\_table()

2. {

3. var status\_filter = $("#status-filter").val();

4. $("#tabel").load("<?php echo base\_url('Home/refresh\_barang') ?>", { status\_filter : status\_filter }, function(){

5. $(".tabel-barang").DataTable({

6. info: true,

7. ordering: true,

8. paging: true,

9. pageLength: 25,

10. columnDefs: [

11. { type: 'num', targets: 0 } ,

12. { "orderable": false, "targets": 7 }

13. ]

14. });

15. });

16. }

Dibawah ini merupakan syntax / coding di function refresh\_barang di controller home. Baris 4 digunakan untuk memanggil barang\_model. Baris 6 - 11, digunakan untuk filter barang berdasarkan status “Bisa Dijual”. Baris 12 digunakan untuk memanggil function generate\_list\_barang di barang\_model. Baris 14 digunakan untuk memanggil view tabel\_barang. View tabel\_barang digunakan untuk mencetak data ke dalam datatable.

1. public function refresh\_barang()

2. {

3. header('Access-Control-Allow-Origin: \*');

4. $this->load->model('barang\_model');

5. $status\_filter = $this->input->post("status\_filter");

6. $where = "";

7. if ($status\_filter == 1 || $status\_filter == "1") {

8. $where = array(

9. "status\_id" => "S0001"

10. );

11. }

12. $data = $this->barang\_model->generate\_list\_barang(0, 0, $where);

13. $data["barang"] = $data["query"];

14. $this->load->view('content/tabel\_barang', $data);

15. }

Dibawah ini merupakan function generate\_list\_barang di barang\_model. Baris 3 digunakan untuk memilih kolom yang mau di select. Baris 4 digunakan untuk memilih tabel - tabel yang digunakan. Baris 5 - 7 digunakan ketika ada tambah syarat query / filter data.. Baris 8 - 9 digunakan untuk menentukan relasi antar tabel. Baris 18 digunakan untuk menjalankan query select. Baris 19 digunakan untuk mengecek apakah data yang dipilih kosong. Jika ada data maka, baris 20 dijalankan. Baris 20 digunakan untuk mengambil hasil query. Baris 22 dijalankan ketika hasil query kosong.

1. public function generate\_list\_barang($start = 0, $limit = 0, $where = "")

2. {

3. $this->db->select('p.id\_produk as id\_produk, p.nama\_produk as nama\_produk, p.harga as harga, p.kategori\_id as kategori\_id, k.nama\_kategori as nama\_kategori, p.status\_id as status\_id, s.nama\_status as nama\_status');

4. $this->db->from('produk p, kategori k, status s');

5. if ($where != "") {

6. $this->db->where($where);

7. }

8. $this->db->where("p.kategori\_id = k.id\_kategori");

9. $this->db->where("p.status\_id = s.id\_status");

10. ..$this->db->order\_by('id\_produk', 'DESC');

11. $data = [];

12. $tempdb = clone $this->db;

13. $num\_rows = $tempdb->count\_all\_results();

14. $data['count'] = $num\_rows;

15. if ($start > 0 && $limit > 0) {

16. $this->db->limit($limit, $start);

17. }

18. $query = $this->db->get();

19. if ($query->num\_rows() > 0) {

20. $data['query'] = $query->result\_array();

21. } else {

22. $data['query'] = 0;

23. }

24. return $data;

25. }

**Fitur Tambah Barang di halaman Tambah Barang**

Dibawah ini merupakan syntax / coding Javascript / Jquery, yang akan dipanggil ketika user menekan tombol “Tambah Barang”. Baris 2-6, digunakan untuk mengambil dari input user. Baris 7-12 digunakan untuk menghapus peringatan ketika data yang di input user ada yang tidak sesuai (seperti Nama Produk & Harga yang tidak di isi, atau Harga diisi bukan dengan angka. Baris 13 digunakan untuk mengirim data melalui AJAX ke function add\_barang di controller Home. Baris 14 sampai 33 digunakan untuk menerima hasil AJAX. Baris 15 digunakan untuk mengecek apakah data yang di input user sesuai. Baris 16 - 17 digunakan jika data yang user input sesuai. Baris 16 - 17 akan memberitahukan user jika user berhasil menambahkan barang, dan mengarahkan user kembali ke halaman Master barang. Baris 19 - 30 digunakan ketika data yang user masukkan tidak sesuai, dan akan ditampilkan pesan data mana yang tidak sesuai. Misal ketika user tidak mengisi nama produk, maka akan muncul pesan “Nama Produk Tidak Boleh Kosong”.

1. $("#insert-barang").click(function(){

2. var id\_produk = $("#id").val();

3. var nama\_produk = $("#nama").val();

4. var harga = $("#harga").val();

5. var kategori\_id = $("#kategori").find(":selected").val();

6. var status\_id = $("#status").find(":selected").val();

7. $("#error-id").html("");

8. $("#error-id").addClass("hidden");

9. $("#error-nama").html("");

10. $("#error-nama").addClass("hidden");

11. $("#error-harga").html("");

12. $("#error-harga").addClass("hidden");

//if (id\_produk != "" && nama\_produk != "") {

13 $.post( "<?= base\_url("home/add\_barang") ?>",

{ id\_produk : id\_produk, nama\_produk : nama\_produk, harga : harga, kategori\_id : kategori\_id, status\_id : status\_id })

14. .done(function( data ) {

15. if (data["cek\_validasi"] == 1 || data["cek\_validasi"] == "1") {

16. alert("tambah barang berhasil");

17. window.location.href = "<?= base\_url() ?>";

18. } else {

19. var data\_err = data["error"];

20. if (typeof(data\_err["id\_produk"]) != "undefined" && data\_err["id\_produk"] !== null) {

21. $("#error-id").removeClass("hidden");

22. $("#error-id").html(data\_err["id\_produk"]);

23. }

24. if (typeof(data\_err["nama\_produk"]) != "undefined" && data\_err["nama\_produk"] !== null) {

25. $("#error-nama").removeClass("hidden");

26. $("#error-nama").html(data\_err["nama\_produk"]);

27. }

28. if (typeof(data\_err["harga"]) != "undefined" && data\_err["harga"] !== null) {

29. $("#error-harga").removeClass("hidden");

30. $("#error-harga").html(data\_err["harga"]);

31. }

32. }

33. });

//}

34. });

Dibawah ini merupakan function add\_barang di controller Home. Baris 4 digunakan untuk memanggil model dengan nama barang\_model. Baris 6 - 9 digunakan untuk menetapkan syarat-syarat form validation beserta pesan ketika ada input user yang tidak sesuai, seperti ID Produk, Nama Produk, atau Harga tidak diisi atau Harga diisi selain dengan angka.. Baris 10 digunakan untuk mengecek apakah data yang di input user sudah sesuai.

Baris 11 - 21 dijalankan ketika data yang di input user sesuai. Baris 11 - 17 digunakan untuk mengambil data dikirimkan melalui AJAX. Baris 19 digunakan untuk memanggil function add\_barang di barang\_model. Baris 20 digunakan untuk mengembalikan bahwa validasi data berhasil.

Baris 23 - 26 dijalankan ketika ada data yang di input user tidak sesuai. Baris 23 - 26 akan mengembalikan ke AJAX bahwa validasi data gagal, dan mengembalikan data apa yang tidak sesuai.

1. public function add\_barang()

2. {

3. header('Access-Control-Allow-Origin: \*');

4. $this->load->model('barang\_model');

5. $data["cek\_validasi"] = 0;

6. $this->load->library('form\_validation');

7. $this->form\_validation->set\_rules('id\_produk', 'id\_produk', 'required',

array('required' => 'ID Produk tidak boleh kosong'));

8. $this->form\_validation->set\_rules('nama\_produk', 'nama\_produk', 'required',

array('required' => 'Nama Produk tidak boleh kosong'));

9. $this->form\_validation->set\_rules('harga', 'harga', 'required|numeric',

array('required' => 'Harga tidak boleh kosong', 'numeric' => 'Harga hanya boleh disi angka'));

10. if ($this->form\_validation->run() === TRUE)

{

11. $data = array(

12. 'id\_produk' => $this->input->post('id\_produk'),

13. 'nama\_produk' => $this->input->post('nama\_produk'),

14. 'harga' => $this->input->post('harga'),

15. 'kategori\_id' => $this->input->post('kategori\_id'),

16. 'status\_id' => $this->input->post('status\_id')

17. );

18. header("Content-Type: application/json");

19. $data["id\_barang"] = $this->barang\_model->add\_barang($data);

20. $data["cek\_validasi"] = 1;

21. echo json\_encode($data);

22. } else {

23. $data["cek\_validasi"] = 0;

24. $data["error"] = $this->form\_validation->error\_array();

25. header("Content-Type: application/json");

26. echo json\_encode($data);

27. }

28. }

Dibawah ini merupakan function add\_barang di barang\_model. Function add\_barang digunakan untuk menyimpan data barang ke dalam database.

public function add\_barang($data)

{

$this->db->insert('produk', $data);

$id\_barang = $this->db->insert\_id();

return $id\_barang;

}

**Fitur Ubah Barang di halaman Ubah Barang**

Dibawah ini merupakan syntax / coding Javascript / Jquery, yang akan dipanggil ketika user menekan tombol “Ubah Barang”. Baris 2-6, digunakan untuk mengambil ID dan data dari input user. Baris 7-12 digunakan untuk menghapus peringatan ketika data yang di input user ada yang tidak sesuai (seperti Nama Produk & Harga yang tidak di isi, atau Harga diisi bukan dengan angka. Baris 13 digunakan untuk mengirim data melalui AJAX ke function update\_barang di controller Home. Baris 14 sampai 33 digunakan untuk menerima hasil AJAX. Baris 15 digunakan untuk mengecek apakah data yang di input user sesuai. Baris 16 - 17 digunakan jika data yang user input sesuai. Baris 16 - 17 akan memberitahukan user jika user berhasil mengubah barang, dan mengarahkan user kembali ke halaman Master barang. Baris 19 - 30 digunakan ketika data yang user masukkan tidak sesuai, dan akan ditampilkan pesan data mana yang tidak sesuai. Misal ketika user tidak mengisi nama produk, maka akan muncul pesan “Nama Produk Tidak Boleh Kosong”.

1. $("#update-barang").click(function(){

2. var id\_produk = $("#id").val();

3. var nama\_produk = $("#nama").val();

4. var harga = $("#harga").val();

5. var kategori\_id = $("#kategori").find(":selected").val();

6. var status\_id = $("#status").find(":selected").val();

7. $("#error-id").html("");

8. $("#error-id").addClass("hidden");

9. $("#error-nama").html("");

10. $("#error-nama").addClass("hidden");

11. $("#error-harga").html("");

12. $("#error-harga").addClass("hidden");

//if (id\_produk != "" && nama\_produk != "") {

13. $.post( "<?= base\_url("home/update\_barang") ?>",

{ id\_produk : id\_produk, nama\_produk : nama\_produk, harga : harga, kategori\_id : kategori\_id, status\_id : status\_id })

14. .done(function( data ) {

15. if (data["cek\_validasi"] == 1 || data["cek\_validasi"] == "1") {

16. alert("ubah barang berhasil");

17. window.location.href = "<?= base\_url() ?>";

18. } else {

19. var data\_err = data["error"];

20. if (typeof(data\_err["id\_produk"]) != "undefined" && data\_err["id\_produk"] !== null) {

21. $("#error-id").removeClass("hidden");

22. $("#error-id").html(data\_err["id\_produk"]);

23. }

24. if (typeof(data\_err["nama\_produk"]) != "undefined" && data\_err["nama\_produk"] !== null) {

25. $("#error-nama").removeClass("hidden");

26. $("#error-nama").html(data\_err["nama\_produk"]);

27. }

28. if (typeof(data\_err["harga"]) != "undefined" && data\_err["harga"] !== null) {

29. $("#error-harga").removeClass("hidden");

30. $("#error-harga").html(data\_err["harga"]);

31. }

32. }

33. });

//}

34. });

Dibawah ini merupakan function update\_barang di controller Home. Baris 4 digunakan untuk memanggil model dengan nama barang\_model. Baris 6 - 9 digunakan untuk menetapkan syarat-syarat form validation beserta pesan ketika ada input user yang tidak sesuai, seperti ID Produk, Nama Produk, atau Harga tidak diisi atau Harga diisi selain dengan angka.. Baris 10 digunakan untuk mengecek apakah data yang di input user sudah sesuai.

Baris 11 - 21 dijalankan ketika data yang di input user sesuai. Baris 11 - 17 digunakan untuk mengambil data dikirimkan melalui AJAX. Baris 19 digunakan untuk memanggil function update\_barang di barang\_model. Baris 20 digunakan untuk mengembalikan bahwa validasi data berhasil.

Baris 23 - 26 dijalankan ketika ada data yang di input user tidak sesuai. Baris 23 - 26 akan mengembalikan ke AJAX bahwa validasi data gagal, dan mengembalikan data apa yang tidak sesuai.

1. public function update\_barang()

2. {

3. header('Access-Control-Allow-Origin: \*');

4. $this->load->model('barang\_model');

5. $data["cek\_validasi"] = 0;

6. $this->load->library('form\_validation');

7. /\* $this->form\_validation->set\_rules('id\_produk', 'id\_produk', 'required',

array('required' => 'ID Produk tidak boleh kosong')); \*/

8. $this->form\_validation->set\_rules('nama\_produk', 'nama\_produk', 'required',

array('required' => 'Nama Produk tidak boleh kosong'));

9. $this->form\_validation->set\_rules('harga', 'harga', 'required|numeric',

array('required' => 'Harga tidak boleh kosong', 'numeric' => 'Harga hanya boleh disi angka'));

10. if ($this->form\_validation->run() === TRUE)

{

11. $data = array(

12. 'nama\_produk' => $this->input->post('nama\_produk'),

13. 'harga' => $this->input->post('harga'),

14. 'kategori\_id' => $this->input->post('kategori\_id'),

15. 'status\_id' => $this->input->post('status\_id')

16. );

17. $id\_produk = $this->input->post('id\_produk');

18. header("Content-Type: application/json");

19. $data["id\_barang"] = $this->barang\_model->update\_barang($data, $id\_produk);

20. $data["cek\_validasi"] = 1;

21. echo json\_encode($data);

22. } else {

23. $data["cek\_validasi"] = 0;

24. $data["error"] = $this->form\_validation->error\_array();

25. header("Content-Type: application/json");

26. echo json\_encode($data);

27. }

28. }

Dibawah ini merupakan function update\_barang di barang\_model. Function update\_barang digunakan untuk mengubah data barang ke dalam database.

public function update\_barang($data, $id)

{

$this->db->where('id\_produk', $id);

$this->db->update('produk', $data);

}

**Fitur Delete Barang di Halaman Master Barang**

Ada 2 jenis delete barang di halaman Master Barang, yang pertama hanya mengubah status barang menjadi “Tidak Bisa Dijual”, yang kedua menghapus data tersebut dari database. Kedua opsi ini akan muncul di modal / pop up konfirmasi setelah user menekan tombol delete. Di modal konfirmasi, juga tersedia pilihan tidak jika user batal menghapus barang tersebut.

**Fitur “Delete” Barang (Hanya Ubah Status Jadi “Tidak Bisa Dijual”) di Halaman Master Barang**

Dibawah ini syntax / coding “delete” yang hanya mengubah status barang menjadi “Tidak Bisa Dijual”. syntax / coding “delete” ini dijalankan setelah user menekan tombol Ya (Ubah Status jadi Tidak Bisa Dijual) pada modal konfirmasi.

Baris 2 digunakan untuk mengambil ID Produk. Baris 5 digunakan untuk mengirim data ke function delete\_barang di controller home. Baris 7 digunakan untuk menampilkan pesan / alert bahwa delete barang berhasil. Baris 8 digunakan memanggil function refresh\_table yang akan memperbaharui data yang ditampilkan di halaman Master Barang. Baris 9 - 10 digunakan menutup modal konfirmasi apa user mau menghapus data barang tersebut.

1. $(document).on("click", "#delete-yes", function(){

2. var delete\_id = $("#delete-id").val();

3. if (delete\_id != undefined && delete\_id != 0) {

4. var id = delete\_id;

5. $.post( "<?= base\_url("home/delete\_barang") ?>",

{ id : id })

6. .done(function( data ) {

7. alert("delete barang berhasil");

8. refresh\_table();

9. $("#delete-confirmation").modal("hide");

10. $("#delete-confirmation").hide();

11. $("#delete-id").val(0);

12. });

13. }

14. });

Dibawah ini merupakan function delete\_barang di controller home, Function ini dipanggil melalui AJAX ketika user menekan tombol Ya (Ubah Status jadi Tidak Bisa Dijual) pada modal konfirmasi.

Baris 4 digunakan untuk memanggil modal bernama barang\_model. Baris 6 - 7 digunakan untuk mengecek ID Produk untuk barang yang mau dihapus tidak kosong. Baris 11 digunakan untuk memanggil function delete\_barang di barang\_model.

1. public function delete\_barang()

2. {

3. header('Access-Control-Allow-Origin: \*');

4. $this->load->model('barang\_model');

5. $this->load->library('form\_validation');

6. $this->form\_validation->set\_rules('id', 'id', 'required');

7. if ($this->form\_validation->run() === TRUE)

8. {

9. $id\_produk = $this->input->post('id');

10. header("Content-Type: application/json");

11. $data["id\_barang"] = $this->barang\_model->delete\_barang($id\_produk);

12. echo json\_encode($data);

13. } else {

14. http\_response\_code(404);

15. die();

16. }

17. }

Dibawah ini merupakan function delete\_barang di barang\_model. Pada function ini data barang tidak benar-benar di delete, hanya diubah status nya saja menjadi “Tidak Bisa Dijual”.

public function delete\_barang($id)

{

$data = array(

'status\_id' => "S0002"

);

$this->db->where('id\_produk', $id);

$this->db->update('produk', $data);

}

**Fitur Delete Barang (Delete dari Database) di Halaman Master Barang**

Dibawah ini syntax / coding “delete” yang menghapus data barang tersebut dari database. Syntax / coding “delete ini dijalankan setelah user menekan tombol Ya (Hapus dari database) pada modal konfirmasi.

Baris 2 digunakan untuk mengambil ID Produk. Baris 5 digunakan untuk mengirim data ke function delete\_barang\_2 di controller home. Baris 7 digunakan untuk menampilkan pesan / alert bahwa delete barang berhasil. Baris 8 digunakan memanggil function refresh\_table yang akan memperbaharui data yang ditampilkan di halaman Master Barang. Baris 9 - 10 digunakan menutup modal konfirmasi apa user mau menghapus data barang tersebut.

$(document).on("click", "#delete-other", function(){

var delete\_id = $("#delete-id").val();

if (delete\_id != undefined && delete\_id != 0) {

var id = delete\_id;

$.post( "<?= base\_url("home/delete\_barang\_2") ?>",

{ id : id })

.done(function( data ) {

alert("delete barang berhasil");

refresh\_table();

$("#delete-confirmation").modal("hide");

$("#delete-confirmation").hide();

$("#delete-id").val(0);

});

}

});

Dibawah ini merupakan function delete\_barang\_2 di controller home, Function ini dipanggil melalui AJAX ketika user menekan tombol Ya (Hapus dari database) pada modal konfirmasi.

Baris 4 digunakan untuk memanggil modal bernama barang\_model. Baris 6 - 7 digunakan untuk mengecek ID Produk untuk barang yang mau dihapus tidak kosong. Baris 11 digunakan untuk memanggil function delete\_barang\_2 di barang\_model.

1. public function delete\_barang\_2()

2. {

3. header('Access-Control-Allow-Origin: \*');

4. $this->load->model('barang\_model');

5. $this->load->library('form\_validation');

6. $this->form\_validation->set\_rules('id', 'id', 'required');

7. if ($this->form\_validation->run() === TRUE)

8. {

9. $id\_produk = $this->input->post('id');

10. header("Content-Type: application/json");

11. $data["id\_barang"] = $this->barang\_model->delete\_barang\_2($id\_produk);

12. echo json\_encode($data);

13. } else {

14. http\_response\_code(404);

15. die();

16. }

17. }

Dibawah ini merupakan function delete\_barang\_2 di barang\_model. Pada function ini data barang benar-benar di delete dari database.

public function delete\_barang\_2($id)

{

$this->db->where('id\_produk', $id);

$this->db->delete('produk');

}