

BAB 6 IMPLEMENTASI SISTEM

Tahapan implementasi yang dilakukan terdiri dari penjelasan tentang spesifikasi lingkungan pengembangan sistem, batasan-batasan implementasi, implementasi algoritma, implementasi basis data dan implementasi antarmuka pengguna.

6.1 Spesifikasi Sistem

Spesifikasi sistem merupakan lingkungan implementasi dalam pembuatan sistem ini yang terdiri dari spesifikasi perangkat keras dan perangkat lunak.

6.1.1 Spesifikasi Perangkat Keras

Dalam penelitian ini ini penulis menggunakan sebuah komputer karena sistem yang dibangun tidak berkomunikasi dengan perangkat keras yang lain. Spesifikasi perangkat keras yang digunakan dalam pembuatan sistem *Customer Relationship Management (CRM) self-service reservation Everyday Smart Hotel Malang* dapat dilihat pada Tabel 6.1

Tabel 6.1 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>System Model</i>	Hp Pavilion G Series Notebook PC
<i>Processor</i>	Intel Core CPU N2920 @122GHZ
<i>Memory</i>	500GB HDD
<i>Memory RAM</i>	8GB

6.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembuatan sistem *Customer Relationship Management (CRM) self-service reservation Everyday Smart Hotel Malang* dapat dilihat pada Tabel 6.2

Tabel 6.2 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 62-bit
<i>Pemrograman Language</i>	PHP
<i>Framework</i>	Codeigniter
<i>Text Editor</i>	Sublime Text
<i>Web Service</i>	Xampp v3.2.1

Tabel 6.2 Spesifikasi Perangkat Lunak (lanjutan)

Nama Komponen	Spesifikasi
<i>Database</i>	MySQL
<i>Browser</i>	Google Chrome

6.2 Batasan Implementasi

Berikut ini merupakan batasan implementasi yang digunakan dalam pembuatan sistem *Customer Relationship Management (CRM) Self Service Reservation Everyday Smart Hotel Malang*.

1. Untuk mengakses sistem ini pengguna harus memiliki apache *web-server* yang sedang aktif
2. Basis data yang digunakan adalah MySQL

6.3 Implementasi Class

Setiap kelas yang telah dirancang pada tahap perancangan sistem direalisasikan pada sebuah file sistem dengan eksistensi *.php. Penjelasan terkait kelas dan file sistem yang digunakan pada sistem akan dijelaskan pada Tabel 6.3

Tabel 6.3 Implementasi Class

No	Package	Nama Kelas	Nama File Sistem
1	<i>Controller</i>	Reservation	reservation.php
2	<i>Models</i>	Customer_model	customer_model.php
3	<i>Models</i>	Room_model	room_model.php
4	<i>View</i>	File HTML	v_index.php
5	<i>View</i>	File HTML	v_formwizard.php
6	<i>View</i>	File HTML	v_success.php
7	<i>View</i>	File HTML	v_formlaundry.php
8	<i>View</i>	File HTML	history.php

6.4 Implementasi Algoritma

Di dalam implementasi algoritma ini hanya mencantumkan proses pemesanan kamar, *ballroom*, *laundry* dan konfirmasi pemesanan yang telah dicantumkan algoritmanya pada sub-bab perancangan komponen sehingga tidak semua fungsi dicantumkan.

6.4.1 Algoritma Pemesanan Kamar

Fungsi `room()` terdapat pada *controller* reservation yang berfungsi untuk melakukan pemesanan kamar dan menampilkan kamar yang tersedia pada jadwal tertentu beserta harganya. Pada Tabel 6.4 merupakan implementasi algoritma dari fungsi `room()` dalam *controller* reservation.

Tabel 6.4 Implementasi Algoritma Fungsi room

1	<code>public function room(){</code>
2	<code> \$data['old'] = \$this->input->post();</code>
3	<code> \$data['roomlist'] = \$this->m_room->get_empty_reservation(\$data['old']['checkin'], \$data['old']['checkout']);</code>
4	<code> if(\$data['old']['checkout'] < \$data['old']['checkin'] \$data['old']['checkout'] == \$data['old']['checkin']){</code>
5	<code> \$this->session->set_flashdata('error', '1');</code>
6	<code> redirect(site_url());</code>
7	<code> }else{</code>
8	<code> foreach(\$data['typelist'] as \$key => \$type){</code>
9	<code> \$data['count'][\$type['id']] = 0;</code>
	<code> foreach (\$data['roomlist'] as \$room) {</code>
10	<code> if(\$type['id'] == \$room['room_type_id']){</code>
11	<code> \$data['count'][\$type['id']]++;</code>
12	<code> }</code>
13	<code> }</code>
14	<code> }</code>
	<code> \$date1=date_create(\$data['old']['checkin']);</code>
15	<code> \$date2=date_create(\$data['old']['checkout']);</code>
16	<code> \$diff=date_diff(\$date1,\$date2);</code>
17	<code> \$data['old']['day'] = \$diff->format('%a');</code>
18	<code>\$this->load->view('home/v_header');</code>
19	<code> \$this->load->view('book_room/v_formwizard', \$data);</code>
20	<code> \$this->load->view('home/v_footer');</code>
21	<code> }</code>
22	<code>}</code>

6.4.2 Algoritma Pemesanan Ballroom

Fungsi `ballroom()` terdapat pada *controller* reservation yang berfungsi untuk melakukan pemesanan *ballroom* dan menampilkan *ballroom* yang tersedia pada jadwal tertentu beserta harganya. Pada Tabel 6.5 merupakan implementasi algoritma dari fungsi `ballroom()` dalam *controller* reservation.

Tabel 6.5 Implementasi Algoritma Fungsi *ballroom*

1	<code>public function ballroom(){</code>
2	<code> \$data['old'] = \$this->input->post();</code>
3	<code> \$data['roomlist'] = \$this->m_ballroom->get_empty_reservation(\$data['old']['checkin'], \$data['old']['checkout']);</code>
4	<code> \$data['typelist'] = \$this->m_ballroom->get_type();</code>
5	<code> \$data['count'] = array();</code>
6	<code> foreach(\$data['typelist'] as \$key => \$type){</code>
7	<code> \$data['count'][\$type['id']] =0;</code>
8	<code> foreach (\$data['roomlist'] as \$room) {</code>
9	<code> if(\$type['id'] == \$room['ballroom_type_id']){</code>
10	<code> \$data['count'][\$type['id']]++;</code>
11	<code> }</code>
12	<code> }</code>
13	<code> }</code>
14	<code> \$date1=date_create(\$data['old']['checkin']);</code>
15	<code> \$date2=date_create(\$data['old']['checkout']);</code>
16	<code> \$diff=date_diff(\$date1,\$date2);</code>
17	<code> \$data['old']['day'] = \$diff->format('%a');</code>
18	<code> \$this->load->view('home/v_header');</code>
19	<code> \$this->load->view('book_room/v_formwizard', \$data);</code>
20	<code> \$this->load->view('home/v_footer');</code>
21	<code> }</code>

6.4.3 Algoritma Pemesanan *Laundry*

Fungsi *laundry()* terdapat pada *controller* reservation yang berfungsi untuk melakukan pemesanan *laundry* dan menampilkan jenis *laundry* yang tersedia pada Everyday Smart Hotel Malang beserta harganya. Pada Tabel 6.6 merupakan implementasi algoritma dari fungsi *laundry()* dalam *controller* reservation.

Tabel 6.6 Implementasi Algoritma Fungsi *laundry*

1	<code>public function laundry(\$id=null){</code>
2	<code> if (\$id == null) {</code>
3	<code> \$data['error_laundry'] = "Sorry, Customer ID \$id does not exist!";</code>
4	<code> \$this->load->view('home/v_header');</code>
5	<code> \$this->load->view('home/v_index', \$data);</code>
6	<code> \$this->load->view('home/v_footer');</code>

Tabel 6.6 Implementasi Algoritma Fungsi laundry

7	}else{
8	\$data['profile'] = \$this->m_customer->get(\$id);
9	\$data['room'] = \$this->m_room->get_reservation_user(\$id);
10	\$data['laundry'] = \$this->m_laundry->get_type();
11	\$today = date("Y-m-d");
12	if (\$today < \$data['room']['check_in'] \$today >
	\$data['room']['check_out']) {
13	redirect('reservation/laundry');
14	}
15	\$this->load->view('home/v_header');
16	\$this->load->view('book_laundry/v_form_laundry', \$data);
17	\$this->load->view('home/v_footer');
18	}
19	}

6.4.4 Algoritma Konfirmasi Pemesanan

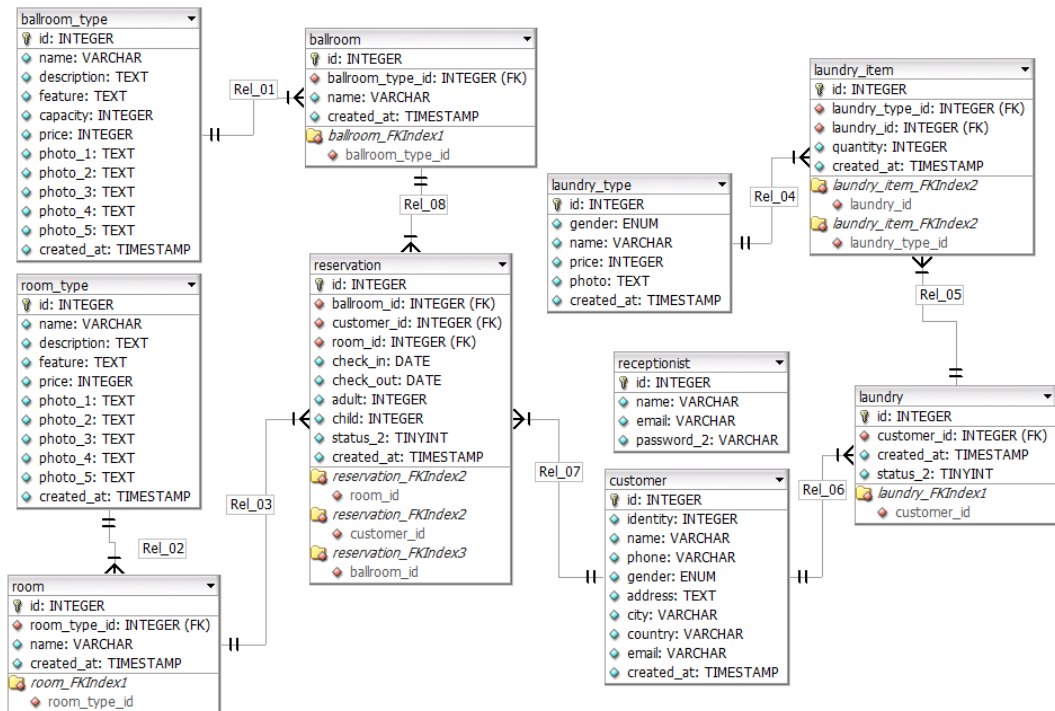
Fungsi confirm() terdapat pada *controller* reservation yang dimiliki *receptionist* yang berfungsi untuk melakukan konfirmasi pemesanan yang telah dipesan oleh pelanggan. Pada Tabel 6.7 merupakan implementasi algoritma dari fungsi confirm() dalam *controller* reservation.

Tabel 6.7 Implementasi Algoritma Fungsi Konfirmasi Pemesanan

1	public function confirm(\$type, \$id) {
2	\$data = array(
3	'status' => 1
4);
5	if(\$type == 'ballroom'){
6	\$this->m_ballroom->update_reservation(\$id, \$data);
7	redirect(site_url()."/admin/reservation/history_ballroom");
8	}else{
9	\$this->m_room->update_reservation(\$id, \$data);
10	redirect(site_url()."/admin/reservation/history_room");
11	}
12	}

6.5 Implementasi Basis Data

Implementasi basis data dilakukan berdasarkan perancangan sistem yang diterapkan ke dalam basis data MySQL. Skema basis data yang telah diimplementasikan dapat dilihat pada Gambar 6.1



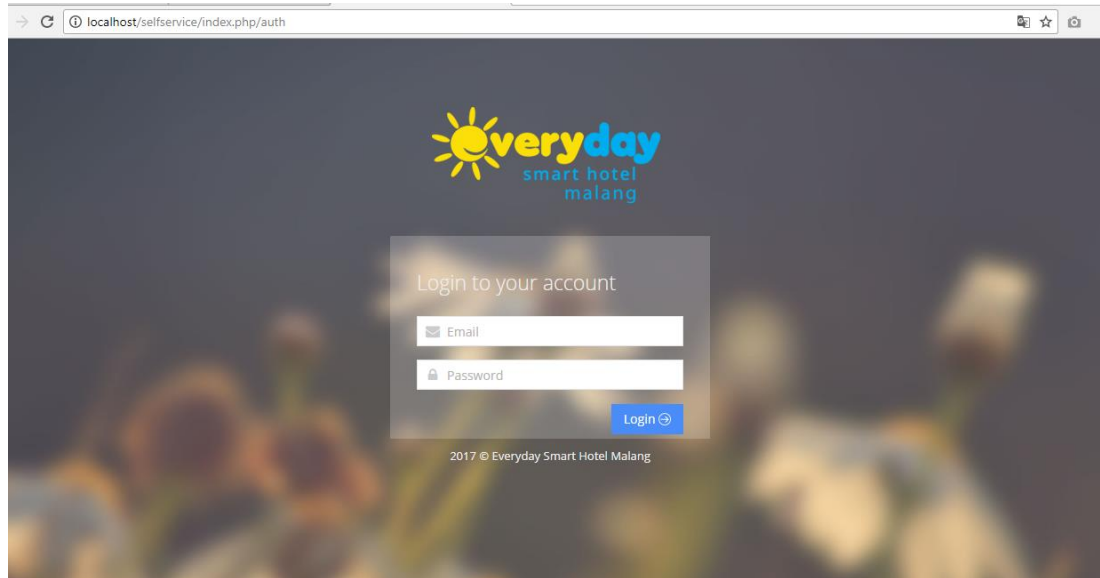
Gambar 6.1 Implementasi Basis Data

Gambar 6.1 menunjukkan hasil implementasi dari perancangan basis data sistem *Customer Relationship Management (CRM) Self-service reservation* yang telah dibuat ke dalam basis data MySQL. Melihat fasilitas dan melakukan pemesanan kamar memiliki tabel *room* dan *room type*, sedangkan untuk melihat fasilitas dan melakukan pemesanan *ballroom* memiliki tabel *ballroom_type* dan *ballroom*. Kemudian untuk melakukan pemesanan *laundry* beserta harganya pada tabel *laundry_item* dan *laundry*, sedangkan untuk *laundry type* digunakan untuk membedakan jenis *laundry*. Selanjutnya pada tabel *customer* digunakan untuk pendataan pelanggan yang telah melakukan pemesanan dan terdaftar pada Everyday Smart Hotel Malang yang terhubung dengan tabel *reservation*. Kemudian pada tabel *receptionist* digunakan untuk hak akses masuk kedalam sistem yang dilakukan oleh *receptionist*.

6.6 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna dibuat berdasarkan perancangan yang diterapkan ke dalam bahasa PHP sebagai view pada *Framework Codeigniter*. Berikut adalah hasil implementasi antarmuka pengguna.

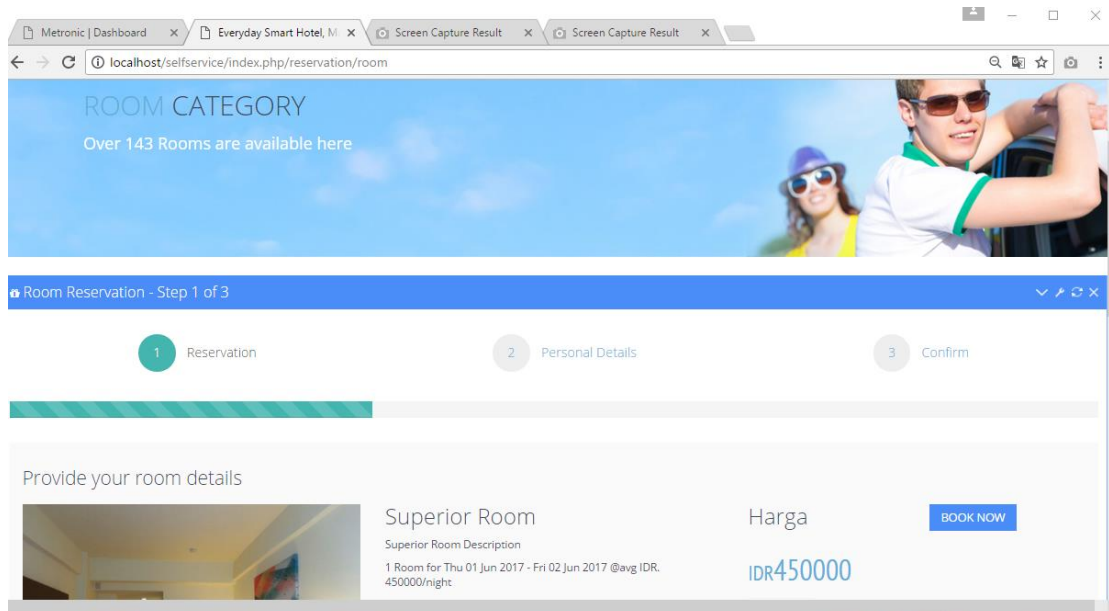
6.6.1 Implementasi Antarmuka Halaman Login



Gambar 6.2 Impementasi Antarmuka Halaman Login

Gambar 6.2 menunjukkan hasil implementasi rancangan antarmuka halaman login. Halaman ini terdiri *form* berisi *email* dan *password* beserta tombol *login* untuk masuk hak akses sistem yang dilakukan oleh *receptionist*.

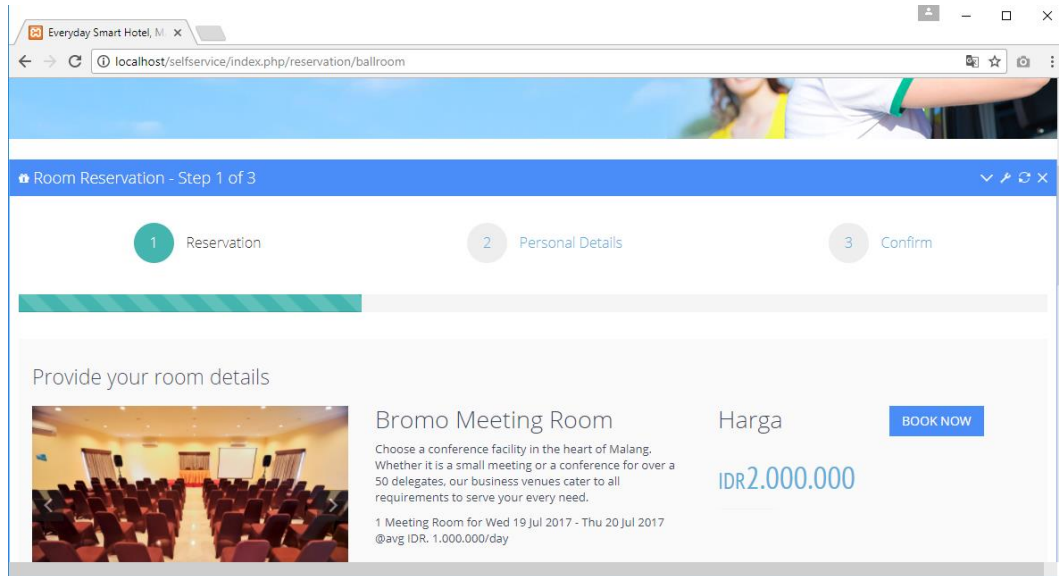
6.6.2 Implementasi Antarmuka Halaman Pemesanan Kamar



Gambar 6.3 Antarmuka Halaman Pemesanan Kamar

Gambar 6.3 menunjukkan hasil implementasi rancangan antarmuka halaman pemesanan kamar. Halaman ini menampilkan seluruh daftar kamar yang tersedia beserta gambar, deskripsi dan harganya. Pelanggan dapat memilih kamar dan melakukan pemesanan dengan menekan tombol "*book now*".

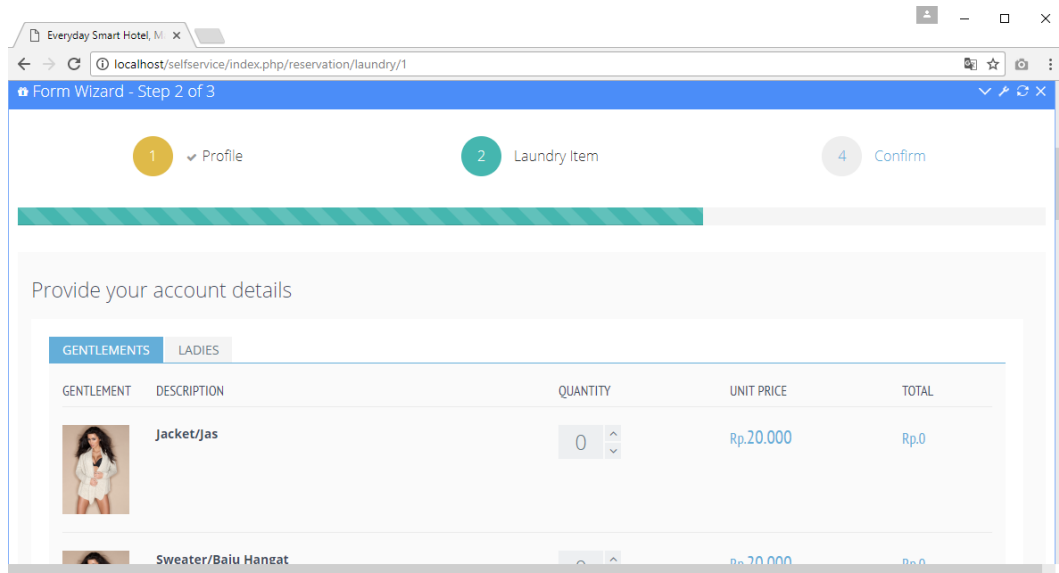
6.6.3 Implementasi Antarmuka Halaman Pemesanan *Ballroom*



Gambar 6.4 Antarmuka Halaman Pemesanan *Ballroom*

Gambar 6.4 menunjukkan hasil implementasi rancangan antarmuka halaman pemesanan *ballroom*. Halaman ini menampilkan seluruh daftar *ballroom* yang tersedia beserta gambar, deskripsi dan harganya. Pelanggan dapat memilih *ballroom* dan melakukan pemesanan dengan menekan tombol “book now”.

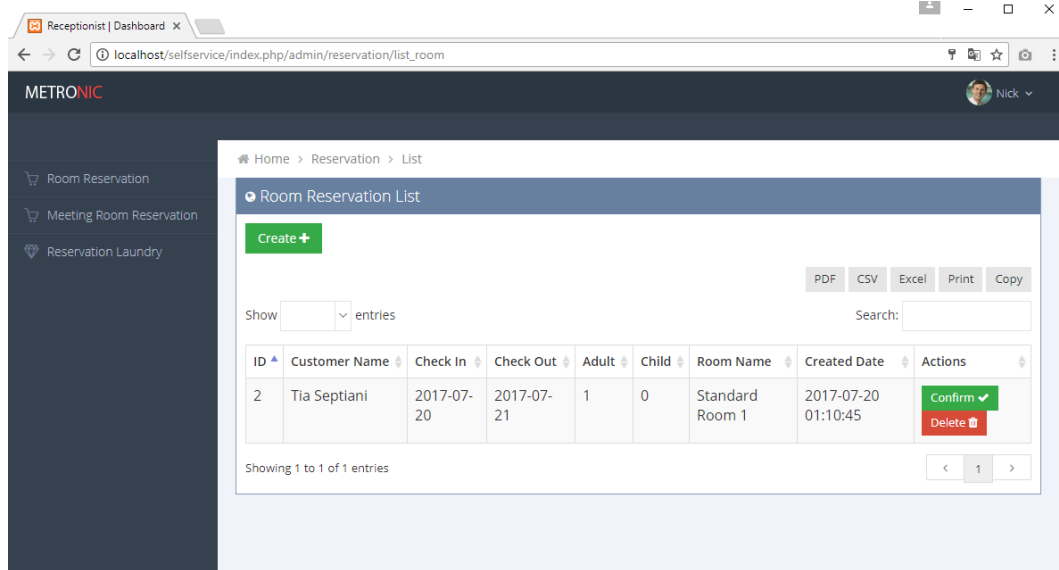
6.6.4 Implementasi Antarmuka Halaman Pemesanan *Laundry*



Gambar 6.5 Antarmuka Halaman Pemesanan *Laundry*

Gambar 6.5 menunjukkan hasil implementasi rancangan antarmuka halaman pemesanan *laundry*. Halaman ini menampilkan seluruh daftar *laundry* yang tersedia beserta gambar, deskripsi, harganya yang telah terakumulasi dan pelanggan dapat memilih dua tipe yaitu *gentlements* dan *ladies*. Pelanggan dapat memilih *laundry* dan melakukan pemesanan dengan menekan tombol “submit”

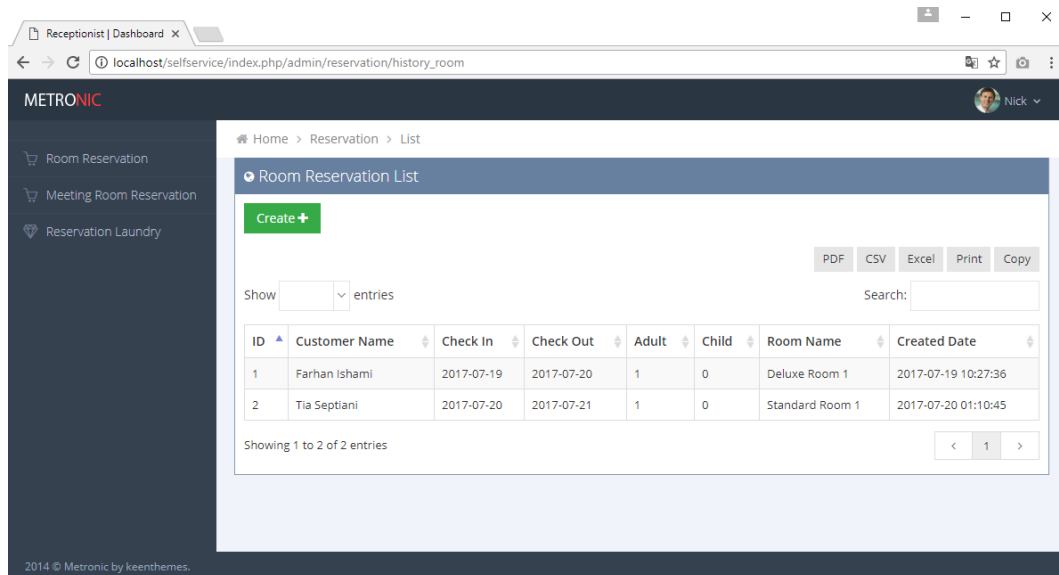
6.6.5 Implementasi Antarmuka Halaman Konfirmasi Pemesanan



Gambar 6.6 Antarmuka Halaman Konfirmasi Pemesanan

Gambar 6.6 menunjukkan hasil implementasi rancangan antarmuka halaman konfirmasi pemesanan. Halaman ini menampilkan seluruh daftar transaksi yang telah dilakukan oleh pelanggan namun belum dikonfirmasi oleh *receptionist*. Tombol *confirm* yang bertujuan untuk mengkonfirmasi pemesanan sedangkan tombol *delete* untuk membatalkan pemesanan.

6.6.6 Implementasi Antarmuka Halaman Melihat *History* Transaksi



Gambar 6.7 Antarmuka Halaman Melihat *History* Transaksi

Gambar 6.7 menunjukkan hasil implementasi rancangan antarmuka halaman melihat seluruh *history* transaksi. Halaman ini menampilkan seluruh daftar transaksi yang telah dipesan oleh pelanggan dan dikonfirmasi oleh *receptionist*.