

WEB PROGRAMMING III

MEMBANGUN APLIKASI PERPUSTAKAAN

Front-End Programming



HTML



JS



CSS



php



KATA PENGANTAR

Segala puji syukur kepada Tuhan Yang Maha Esa karena berkat rahmatNya penulisan modul Mata Kuliah Web Programming III dapat terselesaikan dengan baik. Modul ini disusun untuk memenuhi kebutuhan mahasiswa dalam mata kuliah Web Programming III yang disajikan dalam bentuk praktikum dan diharapkan dapat membekali mahasiswa dalam memahami Pembuatan web dengan menggunakan Framework Codeigniter.

Modul Web Programming III merupakan kelanjutan dari modul sebelumnya yaitu Web Programming II. Pada web Programming II materi bahasan dibatasi sampai dengan halaman Administrator (Back-End), dan pada modul ini dilanjutkan dengan membuat halaman pengunjung (Front-End), pengelolaan transaksi booking buku, laporan-laporan, dan setting IP untuk membuat client-server. Diakhir perkuliahan mahasiswa diharapkan mampu mengimplementasikannya dalam bentuk final proect yang harus dipresentasikan sebagai syarat kelulusan mata kuliah Web Programming III. Teknik penyajiannya dilakukan secara terpadu dan sistematis.

Seperti layaknya sebuah modul, maka pembahasan dimulai dengan menjelaskan target pembelajaran yang hendak dicapai. Dengan demikian pengguna modul ini secara mandiri dapat mengukur tingkat ketuntasan yang dicapainya.

Penulis menyadari sepenuhnya bahwa modul ini tentu memiliki banyak kekurangan. Untuk itu penulis dengan lapang dada menerima masukan dan kritik yang konstruktif dari berbagai pihak demi kesempurnaannya di masa yang akan datang. Semoga modul ini dapat bermanfaat bagi para penggunanya.

Jakarta, Februari 2020

Tim Penulis

DAFTAR ISI

Halaman

KATA PENGANTAR.....	2
DAFTAR ISI	3
PERTEMUAN 1 Penjelasan Kontrak Kuliah, Penegasan tugas project, Persiapan aplikasi pustaka-booking.....	4
PERTEMUAN 2 Membuat Tampilan Katalog Buku dan Detail Buku.....	8
PERTEMUAN 3 Membuat Halaman Login dan Pendaftaran Anggota	18
PERTEMUAN 4 Membuat Halaman Transaksi booking peminjaman	32
PERTEMUAN 5 Membuat Halaman Lihat Keranjang Peminjaman dan Selesai Booking.....	45
PERTEMUAN 6 Membuat Halaman Pengambilan dan Pengembalian Buku.....	53
PERTEMUAN 7 Membuat dan Mencetak Laporan Data Buku ke Printer.....	70
PERTEMUAN 8 Ujian Tengah Semester (UTS)	78
PERTEMUAN 9 Membuat dan Mencetak Laporan Data Buku ke Tampilan Pdf dan Excel ..	79
PERTEMUAN 10 Membuat Laporan Peminjaman Dan Setting Akses Client-Server	85
PERTEMUAN 11-15 Presentasi Project.....	99
PERTEMUAN 16 Ujian Akhir Semester (UAS)	100

PERTEMUAN 1

Penjelasan Kontrak Kuliah, Penegasan tugas project, Persiapan aplikasi pustaka-booking

Deskripsi:

Membahas persiapan dasar untuk memulai sebuah projek aplikasi berbasis web, dalam hal ini kasus yang di angkat adalah perpustakaan, dan dapat memahami desain dan pembuatan database.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Memahami projek yang harus mereka buat dan presentasikan
2. Merancang dan membuat database
3. Melakukan Koneksi database

A. Penjelasan Kontrak Kuliah dan Penegasan Tugas Project

1. Tugas Projek (Kelompok)

Tugas project diadakan untuk memperoleh nilai UTS dan UAS, dengan kata lain tugas projek ini sebagai pengganti UTS dan UAS. Tugas ini dikerjakan secara kelompok dengan maksimal 1 kelompok sebanyak 5 mahasiswa.

2. Pembagian Kelompok

Untuk pembagian kelompok ditentukan oleh dosen pengampu matakuliah, disesuaikan dengan kelompok tugas pada matakuliah yang lain. Seperti matakuliah APSI, IMK, dan MPSI

3. Bentuk Tugas (Kelompok)

Bentuk Tugas Project :

- a. Project merupakan program aplikasi jadi mulai dari *Back-End, Front-End* , dengan menambahkan laporan pada project aplikasi, tema project harus sama dengan tema yang diambil pada mata kuliah APSI, IMK, MPSI.

- b. Setelah pertemuan ke 11, masing-masing kelompok wajib demo program yang telah dikerjakan (Presentasi)

B. Persiapan Aplikasi Perpustakaan

1. Persiapan Database **pustaka**

Pastikan database dan Tabel sudah terbentuk seperti di bawah ini

a. Tabel user

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id 	int(11)			Tidak	<i>Tidak ada</i>		AUTO_INCREMENT
nama	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
alamat	text	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
email	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
image	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
password	varchar(256)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
role_id	int(11)			Tidak	<i>Tidak ada</i>		
is_active	int(1)			Tidak	<i>Tidak ada</i>		
tanggal_input	int(11)			Tidak	<i>Tidak ada</i>		

b. Tabel role

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id 	int(11)			Tidak	<i>Tidak ada</i>		AUTO_INCREMENT
role	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		

c. Tabel kategori

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id 	int(11)			Tidak	<i>Tidak ada</i>		AUTO_INCREMENT
kategori	varchar(45)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		

d. Tabel buku

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id 	int(11)			Tidak	<i>Tidak ada</i>		AUTO_INCREMENT
judul_buku	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
id_kategori	int(11)			Tidak	<i>Tidak ada</i>		
pengarang	varchar(64)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
penerbit	varchar(64)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
tahun_terbit	year(4)			Tidak	<i>Tidak ada</i>		
isbn	varchar(64)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
stok	int(11)			Tidak	<i>Tidak ada</i>		
dipinjam	int(11)			Tidak	<i>Tidak ada</i>		
dibooking	int(11)			Tidak	<i>Tidak ada</i>		
image	varchar(256)	latin1_swedish_ci		Ya	book-default-cover.jpg		

e. Tabel temp

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id 	int(11)			Tidak	<i>Tidak ada</i>		AUTO_INCREMENT
tgl_booking	datetime			Ya	<i>NULL</i>		
id_user	varchar(12)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
email_user	varchar(128)	latin1_swedish_ci		Ya	<i>NULL</i>		
id_buku	int(11)			Ya	<i>NULL</i>		
judul_buku	varchar(255)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
image	varchar(255)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
penulis	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
penerbit	varchar(128)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
tahun_terbit	year(4)			Tidak	<i>Tidak ada</i>		

f. Tabel booking

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id_booking 	varchar(12)	latin1_swedish_ci		Tidak	<i>Tidak ada</i>		
tgl_booking	date			Tidak	<i>Tidak ada</i>		
batas_ambil	date			Tidak	<i>Tidak ada</i>		
id_user	int(11)			Tidak	<i>Tidak ada</i>		

g. Tabel booking_detail

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
id 	int(11)			Tidak	Tidak ada		AUTO_INCREMENT
id_booking	varchar(12)	latin1_swedish_ci		Tidak	Tidak ada		
id_buku	int(11)			Tidak	Tidak ada		

h. Tabel pinjam

Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Komentar	Ekstra
no_pinjam 	varchar(12)	utf8mb4_general_ci		Tidak	Tidak ada		
tgl_pinjam	date			Tidak	Tidak ada		
id_booking	varchar(12)	utf8mb4_general_ci		Tidak	Tidak ada		
id_user	int(11)			Tidak	Tidak ada		
tgl_kembali	date			Tidak	Tidak ada		
tgl_pengembalian	date			Tidak	Tidak ada		
status	enum('Pinjam', 'Kembali')	utf8mb4_general_ci		Tidak	Pinjam		
total_denda	int(11)			Tidak	Tidak ada		

i. Tabel detail_pinjam

Nama	Jenis	Penyortiran	Atribut
no_pinjam	varchar(12)	utf8mb4_general_ci	
id_buku	int(11)		
denda	int(11)		

C. Persiapan aplikasi perpustakaan halaman admin (*Back-End*)

Pastikan mahasiswa sudah mengerjakan materi pustaka-booking halaman admin yang sudah dipelajari pada mata kuliah Web Programming II . materi dapat didownload pada link berikut: [download materi pustaka-booking \(Web II\)](#). Untuk tools pendukung pembelajaran bisa di download pada link: [download tools web programming III](#).

Tugas 1

1. Dosen memeriksa pekerjaan mahasiswa di point B (database) dan C (aplikasi perpustakaan halaman admin)
2. Mahasiswa mengerjakan materi yang ada di pertemuan 2 yaitu membuat halaman katalog buku dan detail buku
3. Point 2 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

PERTEMUAN 2

Membuat Tampilan Katalog Buku dan Detail Buku

Deskripsi:

Membahas bagaimana cara membuat tampilan katalog buku dan halaman detail buku. Halaman katalog buku digunakan untuk menampilkan keseluruhan data buku yang ada di database.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Membuat halaman katalog buku
2. Membuat halaman detail buku
3. Membuat halaman controller member

Hasil akhir dari pertemuan 2 ini adalah tampilan halaman katalog buku dan detail buku seperti berikut:

Pustaka Beranda Login Selamat Datang Pengunjung

Gunawan Budi Sulistyono; Pudji Widodo Graha Ilmu 2020 Booking Detail	Maruloh, Nining Suryani, Evy Priyanti Graha Ilmu 2020 Booking Detail	M. Nandi S, Rahayu N, Widiarina, Suharyanto Graha Ilmu 2020 Booking Detail	Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi Graha Ilmu 2020 Booking Detail

Halaman Detail buku

Pustaka Beranda Daftar Log in Selamat Datang



Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi

Judul Buku:	Analisa Perancangan Sistem Informasi	Kategori:	Komputer
Penerbit:	Graha Ilmu	Dipinjam:	0
Tahun Terbit:	2020	Dibooking:	0
ISBN:	978-623-228-179-0	Tersedia:	12

[Booking](#) [Kembali](#)

A. Membuat Controller Home

Buatlah sebuah controller baru dengan nama Home.php. Dengan script sebagai berikut:

```
<?php

class Home extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model(['ModelBuku', 'ModelUser', 'ModelBooking']);
    }

    public function index()
    {
        $data = [
            'judul' => "Katalog Buku",
            'buku' => $this->ModelBuku->getBuku()->result(),
        ];
    }
}
```

ModelBooking belum dibuat, untuk sementara dihapus dr script. Setelah dibuat di pertemuan 4, baru diketikkan lagi

```

//jika sudah login dan jika belum login
if ($this->session->userdata('email')) {
    $user = $this->ModelUser->cekData(['email' => $this->session-
>userdata('email')])->row_array();

    $data['user'] = $user['nama'];

    $this->load->view('templates/templates-user/header', $data);
    $this->load->view('buku/daftarbuku', $data);
    $this->load->view('templates/templates-user/modal');
    $this->load->view('templates/templates-user/footer', $data);

} else {
    $data['user'] = 'Pengunjung';
    $this->load->view('templates/templates-user/header', $data);
    $this->load->view('buku/daftarbuku', $data);
    $this->load->view('templates/templates-user/modal');
    $this->load->view('templates/templates-user/footer', $data);
}
}

```

View modal
dinonaktifkan dulu
karena belum
dibuat. Setelah
dibuat dipertemuan
2 silakan aktifkan
kembali

Pada controller Home dilakukan pemanggilan view header, daftar buku, modal, dan footer. Oleh karena itu pastikan view-view tersebut telah dibuat sebelum memanggil controller Home. Pada controller ini juga dilakukan pengecekan terhadap user. Jika sudah login, maka akan muncul nama user pada menu header, dan jika belum login maka akan muncul kata pengunjung pada header.

B. Membuat View untuk Template User

1. Membuat view header

Buat sebuah file dengan nama **header.php** pada folder **Application/views/templates/templates-user**. Tulis script berikut:

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Pustaka-Booking | <?= $judul; ?></title>
    <link rel="icon" type="image/png" href="<?= base_url('assets/img/logo/'); ?>logo-pb.png">
    <link rel="stylesheet" href="<?= base_url('assets/'); ?>user/css/bootstrap.css">

```

```

<link href="= base_url('assets/'); ?vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
<link href="= base_url('assets/'); ?datatable/datatables.css" rel="stylesheet" type="text/css">
</head>

<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container">
<a class="navbar-brand" href="= base_url(); ?">Pustaka</a>
<button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarNavAltMarkup">
<div class="navbar-nav">
<a class="nav-item nav-
link active" href="= base_url(); ?">Beranda <span class="sr-
only">(current)</span></a>
<?php
if (!empty($this->session->userdata('email'))){ ?>
<a class="nav-item nav-
link" href="= base_url('booking'); ?">Booking <b><?= $this->ModelBooking-
>getDataWhere('temp', ['email_user' => $this->session->userdata('email')])->num_rows(); ?></b> Buku</a>

<a class="nav-item nav-
link" href="= base_url('member/myprofil'); ?">Profil Saya</a>
<a class="nav-item nav-
link" href="= base_url('member/logout'); ?"><i class="fas fw fa-
login"></i> Log out</a>

<?php } else { ?>

<a class="nav-item nav-link" data-toggle="modal" data-
target="#daftarModal" href="#"></i> Daftar</a>

<a class="nav-item nav-link" data-toggle="modal" data-
target="#loginModal" href="#"></i> Log in</a>

<?php } ?>

```

```
        <span class="nav-item nav-link nav-right" style="display:block; margin-left:20px;">Selamat Datang <b><?= $user; ?></b></span>
    </div>
</div>
</nav>
<div class="container mt-5">
```

2. Membuat view footer

Buat sebuah file dengan nama footer.php pada folder Application/views/templates/templates-user. Tulis script berikut:

```
</div>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01c1HTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

<script src="= base_url(); ?&gt;assets/user/js/bootstrap.js"&gt;&lt;/script&gt;

&lt;script src="<?= base_url('assets/'); ?&gt;vendor/jquery/jquery.min.js"&gt;&lt;/script&gt;
&lt;script src="<?= base_url('assets/'); ?&gt;vendor/bootstrap/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;

&lt;!-- Core plugin JavaScript--&gt;
&lt;script src="<?= base_url('assets/'); ?&gt;vendor/jquery-easing/jquery.easing.min.js"&gt;&lt;/script&gt;

&lt;!-- Custom scripts for all pages--&gt;
&lt;script src="<?= base_url('assets/'); ?&gt;js/sb-admin-2.min.js"&gt;&lt;/script&gt;
&lt;script&gt;
    $('.alert').alert().delay(3000).slideUp('slow');
&lt;/script&gt;

&lt;/body&gt;

&lt;/html&gt;</pre
```

C. Membuat View daftarbuku

Buatlah sebuah file view baru dengan nama daftarbuku.php pada folder Application/buku. Ketikkan script berikut:

```
<?= $this->session->flashdata('pesan'); ?>

<div style="padding: 25px;">
    <div class="x_panel">

        <div class="x_content">

            <!-- Tampilkan semua produk -->
            <div class="row">
                <!-- looping products -->
                <?php foreach ($buku as $buku) { ?>
                    <div class="col-md-2 col-md-3">
                        <div class="thumbnail" style="height: 370px;">
                            

                            <div class="caption">
                                <h5 style="min-height:30px;"><?= $buku-
pengarang ?></h5>
                                <h5><?= $buku->penerbit ?></h5>
                                <h5><?= substr($buku-
>tahun_terbit, 0, 4) ?></h5>
                            <p>
                                <?php
                                    if ($buku->stok < 1) {
                                        echo "<i class='btn btn-outline-
primary fas fw fa-shopping-cart'> Booking&ampnbsp&ampnbsp0</i>";
                                    } else {
                                        echo "<a class='btn btn-outline-
primary fas fw fa-shopping-
cart' href='<?php echo base_url('booking/tambahBooking/' . $buku-
>id) . ''> Booking</a>";
                                    }
                                ?>

                                <a class="btn btn-outline-
warning fas fw fa-search" href="<?php echo base_url('home/detailBuku/' . $buku-
>id); ?>"> Detail</a></p>
                            </div>
                        </div>
                    </div>
                <?php endforeach; ?>
            </div>
        </div>
    </div>
</div>
```

```

        </div> <?php } ?>
        <!-- end looping -->
    </div>

    </div>
</div>
</div>

```

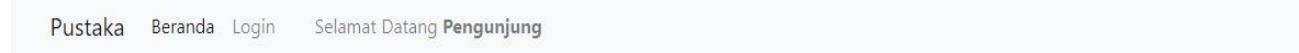
D. Mengatur controller default

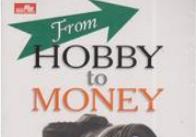
Pada file **routes.php**, ubah controller default menjadi home.

```

$route['default_controller'] = 'home';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;

```



 <p>Gunawan Budi Sulistyono; Pudji Widodo Graha Ilmu 2020</p> <p>Booking Detail</p>	 <p>Maruloh, Nining Suryani, Evy Priyanti Graha Ilmu 2020</p> <p>Booking Detail</p>	 <p>M. Nandi S. Rahayu N, Widiarina, Suharyanto Graha Ilmu 2020</p> <p>Booking Detail</p>	 <p>Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi Graha Ilmu 2020</p> <p>Booking Detail</p>
			

E. Membuat Halaman Detail Buku

1. Membuat fungsi detailBuku

Pada halaman dafatar buku, terdapat link untuk melihat detail buku. Dimana link tersebut mengarahkan ke controller **halaman/detailbuku**. Oleh karena itu kita harus membuat fungsi detailBuku.php pada controller Home

```
<?php
if ($buku->stok < 1) {
    echo "<i class='btn btn-outline-primary fas fw fa-shopping-cart'> Booking&nbsp;&nbsp;0</i>";
} else {
    echo "<a class='btn btn-outline-primary fas fw fa-shopping-cart' href='".$base_url('booking/tambahBooking/' . $buku->id) . "'> Booking</a>";
}
?>

<a class="btn btn-outline-warning fas fw fa-search" href="= base_url('home/detailBuku/' . $buku-&gt;id); ?&gt;"&gt; Detail&lt;/a&gt;&lt;/p&gt;</pre
```

Tambahkan sebuah fungsi baru dengan nama detailBuku pada controller Home

```
public function detailBuku()
{
    $id = $this->uri->segment(3);
    $buku = $this->ModelBuku->joinKategoriBuku(['buku.id' => $id])->result();

    $data['user'] = "Pengunjung";
    $data['title'] = "Detail Buku";

    foreach ($buku as $fields) {
        $data['judul'] = $fields->judul_buku;
        $data['pengarang'] = $fields->pengarang;
        $data['penerbit'] = $fields->penerbit;
        $data['kategori'] = $fields->kategori;
        $data['tahun'] = $fields->tahun_terbit;
        $data['isbn'] = $fields->isbn;
        $data['gambar'] = $fields->image;
        $data['dipinjam'] = $fields->dipinjam;
        $data['dibooking'] = $fields->dibooking;
        $data['stok'] = $fields->stok;
        $data['id'] = $id;
    }

    $this->load->view('templates/templates-user/header', $data);
    $this->load->view('buku/detail-buku', $data);
    $this->load->view('templates/templates-user/modal');
```

```
        $this->load->view('templates/templates-user/footer');
    }
```

2. Membuat view detail-buku

Buat file detail-buku.php pada folder Application/views/buku untuk menampilkan halaman detail buku.

```
<div class="x_panel" align="center">

    <div class="x_content">
        <div class="row">
            <div class="col-sm-3 col-md-3">
                <div class="thumbnail" style="height: auto; position: relative; left: 100%; width: 200%;">
                    
    <th>ISBN: </th>
    <td><?= $isbn ?></td>
    <td>&nbsp;</td>
    <th>Tersedia: </th>
    <td><?= $stok ?></td>
</tr>
</table>
</center>
<p>
    <a class="btn btn-outline-primary fas fw fa-shopping-cart" href=<?= base_url('booking/tambahBooking/' . $id); ?>"> Booking</a>
    <span class="btn btn-outline-secondary fas fw fa-reply" onclick="window.history.go(-1)"> Kembali</span>
</p>
</div>
</div>
</div>
</div>
</div>
</div>

</div>

```

Hasil akhir

Judul Buku:	Analisa Perancangan Sistem Informasi	Kategori:	Komputer
Penerbit:	Graha Ilmu	Dipinjam:	0
Tahun Terbit:	2020	Dibooking:	0
ISBN:	978-623-228-179-0	Tersedia:	12

[Booking](#)
[Kembali](#)

Tugas 2

1. Mahasiswa mengerjakan materi yang ada di pertemuan 3 yaitu membuat halaman Login dan Pendaftaran Anggota
2. Point 1 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

PERTEMUAN 3

Membuat Halaman Login dan Pendaftaran Anggota

Deskripsi:

Membahas bagaimana cara membuat halaman daftar dan login anggota serta memodifikasi profile anggota. Untuk halaman daftar dan login menggunakan modal.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Membuat halaman daftar anggota dengan modal
2. Membuat halaman login anggota dengan modal
3. Memodifikasi data anggota

Secara default jika sistem belum melakukan login maka menu pada header berisi **Daftar** dan **Login**. Sedangkan jika sudah melakukan login menu akan berubah menjadi **Booking**, **Profile saya** dan **Logout**. Pada pertemuan ini akan dibahas mengenai fungsi-fungsi tersebut. Untuk menu Booking akan dibahas dipertemuan berikutnya.

Kondisi menu sebelum login



kondisi menu setelah login



Untuk membuat halaman Login dan daftar, kita menggunakan modal. Pemanggilan modal ada pada halaman **header.php** yang telah kita buat pada pertemuan ke 2

```
<?php } else { ?>
    <a class="nav-item nav-link" data-toggle="modal" data-target="#daftarModal" href="#"> Daftar</a>
    <a class="nav-item nav-link" data-toggle="modal" data-target="#loginModal" href="#"> Log in</a>
<?php } ?>
<span class="nav-item nav-link nav-right" style="display:block; margin-left:20px;">Selamat Datang <b><?= $user; ?></b></span>
</div>
```

A. Membuat Halaman Login dan Daftar Anggota

Buat sebuah file dengan nama **modal.php** pada folder **Application/views/templates/templates-user**. view modal berfungsi untuk membuat tampilan halaman login dan daftar member. Ketikkan script berikut:

```
<!-- login modal -->
<div class="modal fade" tabindex="-1" id="loginModal" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Login Member</h5>
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="= base_url('member'); ?" method="post">
                <div class="modal-body">
                    <div class="form-group row">
                        <label for="email" class="col-sm-2 col-form-
label">Email</label>
                        <div class="col-sm-10">
                            <input type="text" class="form-
control" id="email" name="email" placeholder="Alamat Email">
                        </div>
                    </div>
                    <div class="form-group row">
                        <label for="inputPassword" class="col-sm-2 col-form-
label">Password</label>
                        <div class="col-sm-10">
                            <input type="password" class="form-
control" id="password" name="password" placeholder="Password">
                        </div>
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-outline-secondary" data-
dismiss="modal">Close</button>
                        <button type="submit" class="btn btn-outline-
primary">Log in</button>
                    </div>
                </form>
            </div>
```

```

        </div>
    </div>
<!--/login modal -->

<!-- daftar modal -->
<div class="modal fade" tabindex="-1" id="daftarModal" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Daftar Anggota</h5>
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <form action="= base_url('member/daftar'); ?" method="post">
                <div class="modal-body">
                    <div class="form-group">
                        <input type="text" class="form-control form-control-
user" id="nama" name="nama" placeholder="Nama Lengkap">
                    </div>
                    <div class="form-group">
                        <input type="text" class="form-control form-control-
user" id="alamat" name="alamat" placeholder="Alamat Lengkap">
                    </div>
                    <div class="form-group">
                        <input type="text" class="form-control form-control-
user" id="email" name="email" placeholder="Alamat Email">
                    </div>
                    <div class="form-group">
                        <input type="password" class="form-control form-control-
user" id="password1" name="password1" placeholder="Password">
                    </div>
                    <div class="form-group">
                        <input type="password" class="form-control form-control-
user" id="password2" name="password2" placeholder="Ulangi Password">
                    </div>
                </div>

                <div class="modal-footer">
                    <button type="button" class="btn btn-outline-secondary" data-
dismiss="modal">Close</button>
                    <button type="submit" class="btn btn-outline-
primary">Simpan</button>
                </div>
            </form>
        </div>
    </div>
</div>

```

```

        </form>
    </div>
</div>
<!--/end of Modal Daftar -->

<!-- modal info-->
<div class="modal fade" tabindex="-1" id="modalinfo" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Informasi</h5>
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <span class="alert alert-message alert-
success">Waktu Pengambilan Buku 1x24 jam dari Booking!!!</span>
            </div>
            <div class="modal-footer">
                <a class="btn btn-outline-info" href="= base_url(); ?&gt;">Ok</a>
            </div>
        </div>
    </div>
<!--/modal info -->

```

The image shows a screenshot of a web application interface. At the top, there is a large blue header with the word "UNIVERSITAS" in white. Below the header, there are two overlapping modal windows.

The first modal, titled "Daftar Anggota", contains the following fields:

- Nama Lengkap
- Alamat tempat Tinggal
- Alamat Email
- Password
- Ulangi Password

The second modal, titled "Login Member", contains the following fields:

- Email
- Alamat Email
- Password

Both modals have "Close" and "Simpan" buttons at the bottom right.

Untuk memanggil halaman login, sistem memanggil id **loginModal** dari yang ada pada file modal.php. sedangkan untuk memanggil halaman daftar sistem memanggil id **daftarModal**

```
<!-- login modal -->
<div class="modal fade" tabindex="-1" id="loginModal" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Login Member</h5>
```

```
<!-- daftar modal -->
<div class="modal fade" tabindex="-1" id="daftarModal" role="dialog">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Daftar Anggota</h5>
```

B. Membuat Controller Member

Controller member berisi fungsi-fungsi seperti pengecekan login, pendaftaran anggota, management profile, dan fungsi logout. Buatlah sebuah controller dengan nama **Member.php**

1. Membuat fungsi cek login

Buatlah fungsi _login pada controller Member yang berisi coding untuk melakukan pengecekan login. Sistem akan mengarahkan ke halaman home setelah proses login dilakukan.

```
<?php

class Member extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model(['ModelBuku', 'ModelUser', 'ModelBooking']);
    }

    public function index()
```

ModelBooking belum dibuat, untuk sementara dihapus dr script. Setelah dibuat di pertemuan 4, baru diketikkan lagi

```

        $this->_login();
    }

private function _login()
{
    $email = htmlspecialchars($this->input->post('email', true));
    $password = $this->input->post('password', true);

    $user = $this->ModelUser->cekData(['email' => $email])->row_array();

    //jika usernya ada
    if ($user) {
        //jika user sudah aktif
        if ($user['is_active'] == 1) {
            //cek password
            if (password_verify($password, $user['password'])) {
                $data = [
                    'email' => $user['email'],
                    'role_id' => $user['role_id'],
                    'id_user' => $user['id'],
                    'nama' => $user['nama']
                ];

                $this->session->set_userdata($data);
                redirect('home');
            } else {
                $this->session-
>set_flashdata('pesan', '<div class="alert alert-danger alert-
message" role="alert">Password salah!!</div>');
                redirect('home');
            }
        } else {
            $this->session->set_flashdata('pesan', '<div class="alert alert-
danger alert-message" role="alert">User belum diaktifasi!!</div>');
            redirect('home');
        }
    } else {
        $this->session->set_flashdata('pesan', '<div class="alert alert-
danger alert-message" role="alert">Email tidak terdaftar!!</div>');
        redirect('home');
    }
}

```

2. Membuat fungsi daftar

Fungsi daftar member berisi script tentang proses penyimpanan data pendaftar. Silakan tambahkan fungsi dengan nama **daftar** pada controller **Member**, dan ketikkan coding berikut:

```
public function daftar()
{
    $this->form_validation->set_rules('nama', 'Nama Lengkap', 'required', [
        'required' => 'Nama Belum diisi!!'
    ]);
    $this->form_validation-
>set_rules('alamat', 'Alamat Lengkap', 'required', [
        'required' => 'Alamat Belum diisi!!'
    ]);
    $this->form_validation-
>set_rules('email', 'Alamat Email', 'required|trim|valid_email|is_unique[user.email]', [
        'valid_email' => 'Email Tidak Benar!!!',
        'required' => 'Email Belum diisi!!!',
        'is_unique' => 'Email Sudah Terdaftar!'
    ]);
    $this->form_validation-
>set_rules('password1', 'Password', 'required|trim|min_length[3]|matches[password2]', [
        'matches' => 'Password Tidak Sama!!!',
        'min_length' => 'Password Terlalu Pendek'
    ]);
    $this->form_validation-
>set_rules('password2', 'Repeat Password', 'required|trim|matches[password1]');

    $email = $this->input->post('email', true);
    $data = [
        'nama' => htmlspecialchars($this->input->post('nama', true)),
        'alamat' => $this->input->post('alamat', true),
        'email' => htmlspecialchars($email),
        'image' => 'default.jpg',
        'password' => password_hash($this->input-
>post('password1'), PASSWORD_DEFAULT),
        'role_id' => 2,
        'is_active' => 1,
        'tanggal_input' => time()
    ];
    $this->ModelUser->simpanData($data);
```

```

        $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-message" role="alert">Selamat!! akun anggota anda sudah dibuat.</div>');
        redirect(base_url());
    }

```

3. Membuat fungsi myProfil

Fungsi myProfil digunakan untuk menampilkan data profil anggota yang sedang aktif/login. Dengan mengirimkan data ke dalam view index. Berikut scrip fungsi myProfil

```

public function myProfil()
{
    $data['judul'] = 'Profil Saya';
    $user = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();

    foreach ($user as $a) {
        $data = [
            'image' => $user['image'],
            'user' => $user['nama'],
            'email' => $user['email'],
            'tanggal_input' => $user['tanggal_input'],
        ];
    }

    $this->load->view('templates/templates-user/header', $data);
    $this->load->view('member/index', $data);
    $this->load->view('templates/templates-user/modal');
    $this->load->view('templates/templates-user/footer', $data);
}

```

Pada script diatas, sistem memanggil view **index** yang ada di folder member. Oleh karena itu kita akan membuat sebuah file **index.php** di folder **Application/views/member**. Isi dari script index sebagai berikut:

```

<!-- Begin Page Content -->
<div class="container-fluid">

    <div class="row">
        <div class="col-lg-6 justify-content-x">
            <?= $this->session->flashdata('pesan'); ?>
        </div>
    </div>

```

```

<div class="card mb-3" style="max-width: 540px;">
    <div class="row no-gutters">
        <div class="col-md-4">
            ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();

    foreach ($user as $a) {
        $data = [
            'image' => $user['image'],
            'user' => $user['nama'],
            'email' => $user['email'],
            'tanggal_input' => $user['tanggal_input'],
        ];
    }

    $this->form_validation->set_rules('nama', 'Nama Lengkap', 'required|trim', [
        'required' => 'Nama tidak Boleh Kosong'
    ]);

    if ($this->form_validation->run() == false) {
        $this->load->view('templates/templates-user/header', $data);
        $this->load->view('member/ubah-anggota', $data);
        $this->load->view('templates/templates-user/modal');
        $this->load->view('templates/templates-user/footer', $data);
    } else {
        $nama = $this->input->post('nama', true);
        $email = $this->input->post('email', true);

        //jika ada gambar yang akan diupload
        $upload_image = $_FILES['image']['name'];

        if ($upload_image) {
            $config['upload_path'] = './assets/img/profile/';
            $config['allowed_types'] = 'gif|jpg|png';
            $config['max_size']     = '3000';
            $config['max_width']   = '1024';
            $config['max_height']  = '1000';
            $config['file_name']   = 'pro' . time();
        }
    }
}
```

```

        $this->load->library('upload', $config);

        if ($this->upload->do_upload('image')) {
            $gambar_lama = $data['user']['image'];
            if ($gambar_lama != 'default.jpg') {
                unlink(FCPATH . 'assets/img/profile/' . $gambar_lama);
            }

            $gambar_baru = $this->upload->data('file_name');
            $this->db->set('image', $gambar_baru);
        } else {
        }

    }

    $this->db->set('nama', $nama);
    $this->db->where('email', $email);
    $this->db->update('user');

    $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-message" role="alert">Profil Berhasil diubah </div>');
    redirect('member/myprofil');
}
}

```

Pada saat tombol ubah profil diklik, sistem akan mengarahkan ke view ubah-anggota.php, dimana file view tersebut menampilkan data profil anggota untuk diedit. Adapun data yang bisa diedit hanya nama anggota dan gambar. Berikut script **ubah-anggota.php** yang ditulis di folder Application/views/member.

```

<!-- Begin Page Content -->
<div class="container-fluid">

    <div class="row">
        <div class="col-lg-9">
            <?= form_open_multipart('member/ubahprofil'); ?>
            <div class="form-group row">
                <label for="email" class="col-sm-2 col-form-label">Email</label>
                <div class="col-sm-10">
                    <input type="text" class="form-control" id="email" name="email" value="<?= $email; ?>" readonly>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<div class="form-group row">
    <label for="nama" class="col-sm-2 col-form-label">Nama Lengkap</label>
        <div class="col-sm-10">
            <input type="text" class="form-control" id="nama" name="nama" value=<?= $user; ?>>
                <?= form_error('nama', '<small class="text-danger pl-3">', '</small>'); ?>
            </div>
        </div>
        <div class="form-group row">
            <div class="col-sm-2">Gambar</div>
            <div class="col-sm-10">
                <div class="row">
                    <div class="col-sm-3">
                        <img src=<?= base_url('assets/img/profile/') . $image; ?>" class="img-thumbnail" alt="">
                    </div>
                    <div class="col-sm-9">
                        <div class="custom-file">
                            <input type="file" class="custom-file-input" id="image" name="image">
                            <label class="custom-file-label" for="image">Pilih file</label>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="form-group row justify-content-end">
        <div class="col-sm-10">
            <button type="submit" class="btn btn-primary">Ubah</button>
            <button class="btn btn-dark" onclick="window.history.go(-1)"> Kembali</button>
        </div>
    </div>

    </form>
</div>
</div>

</div>
<!-- /.container-fluid -->

```

```
</div>
<!-- End of Main Content -->
```

Hasil akhir perubahan data profil.

The screenshot shows a user profile card. On the left is a QR code. To its right, the name "Maruloh" is displayed above the email address "maru@gmail.com". Below that, it says "Jadi member sejak" followed by the date "31 January 2020". At the bottom of the card is a teal-colored button with the white text "Ubah Profil" (Edit Profile) and a small user icon.

5. Membuat fungsi Logout

Berikut script untuk fungsi logout

```
public function logout()
{
    $this->session->unset_userdata('email');
    $this->session->unset_userdata('role_id');

    $this->session->set_flashdata('pesan', '<div class="alert alert-success alert-message" role="alert">Anda telah logout!!</div>');
    redirect('home');
}
```

6. Merubah Pustaka Helper

Buka kembali pustaka helper yang pernah dibuat (pustaka_helper.php). edit file tersebut seperti script berikut:

```

<?php

function cek_login()
{
    $ci = get_instance();

    if (!$ci->session->userdata('email')) {
        $ci->session->set_flashdata('pesan', '<div class="alert alert-danger" role="alert">Akses ditolak. Anda belum login! </div>');
        if ($ci->session->userdata('role_id') == 1) {
            redirect('autentifikasi');
        } else {
            redirect('home');
        }
    } else {
        $role_id = $ci->session->userdata('role_id');
        $id_user = $ci->session->userdata('id_user');
    }
}

function cek_user()
{
    $ci = get_instance();

    $role_id = $ci->session->userdata('role_id');
    if ($role_id != 1) {
        $ci->session->set_flashdata('pesan', '<div class="alert alert-danger" role="alert">Akses tidak diizinkan </div>');
        redirect('home');
    }
}

```

Tugas 3

1. Mahasiswa mengerjakan materi yang ada di pertemuan 4 yaitu membuat halaman Transaksi Booking Peminjaman
2. Point 1 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

PERTEMUAN 4

Membuat Halaman Transaksi booking peminjaman

Deskripsi:

Membahas bagaimana cara membuat dan memodifikasi halaman transaksi booking peminjaman. Halaman transaksi booking peminjaman digunakan untuk menyimpan data buku yang dipinjam, anggota dapat menghapus atau menambah buku yang telah dibooking.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Membuat sebuah controller booking dan pinjam
2. Memodifikasi controller booking dan pinjam
3. Memodifikasi halaman booking buku

Pada tahap ini user melakukan klik tombol booking pada buku yang diinginkan, kemudian data buku yang di klik tersebut akan dimasukkan ke dalam keranjang dalam hal ini tabel temp sebagai tabel sementara yang digunakan untuk menyimpan data buku pada saat klik tombol booking.

Setelah booking, untuk menyelesaikan transaksi booking user harus melihat tampilan keranjang. Pada tampilan keranjang ini ada dua(2) tombol yaitu tombol lanjutkan booking dan tombol selesai booking.

Tombol lanjutkan booking digunakan untuk bisa booking buku yang lainnya apabila user ingin melakukan booking buku yang lain. Sedangkan tombol selesai booking digunakan untuk menyelesaikan booking agar data booking buku ini masuk ke sistem dan di proses oleh sistem dan bisa dilakukan peminjaman buku.

A. Membuat Model Booking

Model booking berisi script mengenai fungsi-fungsi pengaksesan database. Seperti perintah simpan, edit, hapus, update, dll. Pada model booking, perintah hanya untuk proses booking buku.

Buatlah sebuah sebuah model baru dengan nama **ModelBooking.php** lalu ketikkan perintah berikut:

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class ModelBooking extends CI_Model
{
    public function getData($table)
    {
        return $this->db->get($table)->row();
    }

    public function getDataWhere($table, $where)
    {
        $this->db->where($where);
        return $this->db->get($table);
    }

    public function getOrderByLimit($table, $order, $limit)
    {
        $this->db->order_by($order, 'desc');
        $this->db->limit($limit);
        return $this->db->get($table);
    }

    public function joinOrder($where)
    {
        $this->db->select('*');
        $this->db->from('booking bo');
        $this->db->join('booking_detail d', 'd.id_booking=bo.id_booking');
        $this->db->join('buku bu ', 'bu.id=d.id_buku');
        $this->db->where($where);

        return $this->db->get();
    }

    public function simpanDetail($where = null)
```

```

{
    $sql = "INSERT INTO booking_detail (id_booking,id_buku) SELECT booking.id
_booking,temp.id_buku FROM booking, temp WHERE temp.id_user=booking.id_user AND b
ooking.id_user='$where'";
    $this->db->query($sql);
}

public function insertData($table, $data)
{
    $this->db->insert($table, $data);
}

public function updateData($table, $data, $where)
{
    $this->db->update($table, $data, $where);
}

public function deleteData($where, $table)
{
    $this->db->where($where);
    $this->db->delete($table);
}

public function find($where)
{
    //Query mencari record berdasarkan ID-nya
    $this->db->limit(1);
    return $this->db->get('buku', $where);
}

public function kosongkanData($table)
{
    return $this->db->truncate($table);
}

public function createTemp()
{
    $this->db-
>query('CREATE TABLE IF NOT EXISTS temp(id_booking varchar(12), tgl_booking DATET
IME, email_user varchar(128), id_buku int)');
}

public function selectJoin()
{
    $this->db->select('*');
}

```

```

$this->db->from('booking');
$this->db-
>join('booking_detail', 'booking_detail.id_booking=booking.id_booking');
    $this->db->join('buku', 'booking_detail.id_buku=buku.id');
    return $this->db->get();
}

public function showtemp($where)
{
    return $this->db->get('temp', $where);
}

public function kodeOtomatis($tabel, $key)
{
    $this->db->select('right(' . $key . ',3) as kode', false);
    $this->db->order_by($key, 'desc');
    $this->db->limit(1);
    $query = $this->db->get($tabel);
    if ($query->num_rows() <> 0) {
        $data = $query->row();
        $kode = intval($data->kode) + 1;
    } else {
        $kode = 1;
    }

    $kodemax = str_pad($kode, 3, "0", STR_PAD_LEFT);
    $kodejadi = date('dmY') . $kodemax;

    return $kodejadi;
}
}

```

B. Membuat Model Pinjam

Model pinjam pada dasarnya memiliki fungsi yang sama seperti model booking. Yang berbeda hanya tabel yang terlibat didalamnya. Model pinjam khusus untuk menangani proses yang berhubungan dengan tabel pinjam.

Buatlah sebuah sebuah model baru dengan nama **ModelPinjam.php** lalu ketikkan perintah berikut:

```

<?php if (!defined('BASEPATH')) exit('No Direct Script Access Allowed');

class ModelPinjam extends CI_Model
{
    //manip table pinjam
    public function simpanPinjam($data)
    {
        $this->db->insert('pinjam', $data);
    }

    public function selectData($table, $where)
    {
        return $this->db->get($table, $where);
    }

    public function updateData($data, $where)
    {
        $this->db->update('pinjam', $data, $where);
    }

    public function deleteData($tabel, $where)
    {
        $this->db->delete($tabel, $where);
    }

    public function joinData()
    {
        $this->db->select('*');
        $this->db->from('pinjam');
        $this->db-
>join('detail_pinjam', 'detail_pinjam.no_pinjam=pinjam.no_pinjam', 'Right');

        return $this->db->get()->result_array();
    }

    //manip tabel detail pinjam
    public function simpanDetail($idbooking, $nopinjam)
    {
        $sql = "INSERT INTO detail_pinjam (no_pinjam,id_buku) SELECT pinjam.no_pi
njam,booking_detail.id_buku FROM pinjam, booking_detail WHERE booking_detail.id_b
ooking=$idbooking AND pinjam.no_pinjam='$nopinjam'";
        $this->db->query($sql);
    }
}

```

C. Membuat Controller Booking

Controller Booking.php berisi fungsi-fungsi yang berhubungan dengan proses booking buku. Didalamnya terdapat fungsi tambahBuku, hapusbooking, bookingSelesai, info, dan exportToPdf. Pada pertemuan ini akan kita buat fungsi index, tambahBuku, viewDataBooking, dan hapusbooking.

Buatlah sebuah controller baru dengan nama Booking.php. ketikkan pada controller script berikut:

```
<?php
defined('BASEPATH') or exit('No Direct Script Access Allowed');
date_default_timezone_set('Asia/Jakarta');

class Booking extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        cek_login();
        $this->load->model(['ModelBooking', 'ModelUser']);
    }

    public function index()
    {
        $id = ['bo.id_user' => $this->uri->segment(3)];
        $id_user = $this->session->userdata('id_user');
        $data['booking'] = $this->ModelBooking->joinOrder($id)->result();

        $user = $this->ModelUser->cekData(['email' => $this->session-
>userdata('email')])->row_array();
        foreach ($user as $a) {
            $data = [
                'image' => $user['image'],
                'user' => $user['nama'],
                'email' => $user['email'],
                'tanggal_input' => $user['tanggal_input']
            ];
        }
        $dtb = $this->ModelBooking->showtemp(['id_user' => $id_user])->
num_rows();

        if ($dtb < 1) {
```

```

        $this->session->set_flashdata('pesan', '<div class="alert alert-
massege alert-danger" role="alert">Tidak Ada Buku dikeranjang</div>');
        redirect(base_url());
    } else {
        $data['temp'] = $this->db-
>query("select image, judul_buku, penulis, penerbit, tahun_terbit,id_buku from te
mp where id_user='$id_user'")->result_array();
    }
    $data['judul'] = "Data Booking";

    $this->load->view('templates/templates-user/header', $data);
    $this->load->view('booking/data-booking', $data);
    $this->load->view('templates/templates-user/modal');
    $this->load->view('templates/templates-user/footer');
}

```

Fungsi index pada controller Booking adalah menampilkan data booking yang telah dilakukan oleh member.

```
$this->load->view('booking/data-booking', $data);
```

Jika sistem memeriksa tidak ditemukan adanya buku yang dipinjam, maka sistem melakukan feedback berupa pesan.

```

if ($dtb < 1) {
    $this->session->set_flashdata('pesan', '<div class="alert alert-
massege alert-danger" role="alert">Tidak Ada Buku dikeranjang</div>');
    redirect(base_url());
}

```

1. Membuat fungsi tambahBooking

Fungsi tambahBooking berisi script proses penyimpanan data booking buku. Dimana penyimpanan pada proses ini dilakukan di table **temp**, yang melibatkan table buku sebagai objek yang dipinjam.

Tambahkan sebuah fungsi baru dengan nama **tambahBooking**, dan ketikkan script berikut:

```

public function tambahBooking()
{
    $id_buku = $this->uri->segment(3);
}

```

```

//memilih data buku yang untuk dimasukkan ke tabel temp/keranjang melalui
variabel $isi
$d = $this->db->query("Select*from buku where id='$id_buku'")->row();

//berupa data2 yang akan disimpan ke dalam tabel temp/keranjang
$isi = [
    'id_buku' => $id_buku,
    'judul_buku' => $d->judul_buku,
    'id_user' => $this->session->userdata('id_user'),
    'email_user' => $this->session->userdata('email'),
    'tgl_booking' => date('Y-m-d H:i:s'),
    'image' => $d->image,
    'penulis' => $d->pengarang,
    'penerbit' => $d->penerbit,
    'tahun_terbit' => $d->tahun_terbit
];

//cek apakah buku yang di klik booking sudah ada di keranjang
$temp = $this->ModelBooking-
>getDataWhere('temp', ['id_buku' => $id_buku])->num_rows();

$userid = $this->session->userdata('id_user');

//cek jika sudah memasukan 3 buku untuk dibooking dalam keranjang
$tempuser = $this->db-
>query("select*from temp where id_user ='$userid'")->num_rows();

//cek jika masih ada booking buku yang belum diambil
$databooking = $this->db-
>query("select*from booking where id_user='$userid'")->num_rows();
if ($databooking > 0) {
    $this->session->set_flashdata('pesan', '<div class="alert alert-
danger alert-
message" role="alert">Masih Ada booking buku sebelumnya yang belum diambil.<br> A
bmil Buku yang dibooking atau tunggu 1x24 Jam untuk bisa booking kembali </div>');
    redirect(base_url());
}

//jika buku yang diklik booking sudah ada di keranjang
if ($temp > 0) {
    $this->session->set_flashdata('pesan', '<div class="alert alert-
danger alert-message" role="alert">Buku ini Sudah anda booking </div>');
    redirect(base_url() . 'home');
}

```

```

}

//jika buku yang akan dibooking sudah mencapai 3 item
if ($tempuser == 3) {
    $this->session->set_flashdata('pesan', '<div class="alert alert-danger alert-message" role="alert">Booking Buku Tidak Boleh Lebih dari 3</div>');
    redirect(base_url() . 'home');
}

//membuat tabel temp jika belum ada
$this->ModelBooking->createTemp();
$this->ModelBooking->insertData('temp', $isi);

//pesan ketika berhasil memasukkan buku ke keranjang
$this->session->set_flashdata('pesan', '<div class="alert alert-success alert-message" role="alert">Buku berhasil ditambahkan ke keranjang </div>');
redirect(base_url() . 'home');
}

```

Pada fungsi tambahBooking, dilakukan pengecekan jumlah buku yang telah dibooking. Jika buku yang dibooking sudah 3, maka member tidak bisa melakukan proses booking kembali

```

if ($tempuser == 3) {
    $this->session->set_flashdata('pesan', '<div class="alert alert-danger alert-message" role="alert">Booking Buku Tidak Boleh Lebih dari 3</div>');
    redirect(base_url() . 'home');
}

```

System akan membuat table temp jika pada database belum ditemukan table temp, lalu menyimpan data kedalam table tersebut

```

$this->ModelBooking->createTemp();
$this->ModelBooking->insertData('temp', $isi);

```

Hasil dari fungsi tambahBooking



Pustaka Beranda Booking 1 Buku Profil Saya Log out Selamat Datang Maruloh, M.Kom

Saat menu booking diklik, akan menampilkan data booking seperti berikut:

Pustaka Beranda Booking 1 Buku Profil Saya Log out Selamat Datang Maruloh, M.Kom

No.	Buku	Penulis	penerbit	Tahun	Pilihan
1		Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	<input type="checkbox"/>

 Lanjutkan Booking Buku  Selesaikan Booking

Untuk membuat halaman tersebut, buatlah sebuah view baru dengan nama **data-booking.php**, letakkan pada folder **Application/views/booking**.

Berikut script **data-booking.php**

```
<div class="container">
    <center>
        <table>
            <tr>
                <td>
                    <div class="table-responsive full-width">
                        <table class="table table-bordered table-striped table-hover" id="table-datatable">
                            <tr>
                                <th>No.</th>
                                <th>Buku</th>
                                <th>Penulis</th>
                                <th>penerbit</th>
                                <th>Tahun</th>
                                <th>Pilihan</th>
                            </tr>
                            <?php
                                $no = 1;
                                foreach ($temp as $t) {
                            ?>
                            <tr>
                                <td><?= $no; ?></td>
                                <td>
                                    
                                </td>
```

```

        <td nowrap><?= $t['penulis']; ?></td>
        <td nowrap><?= $t['penerbit']; ?></td>
        <td nowrap><?= substr($t['tahun_terbit'], 0,
4); ?></td>
                <td nowrap>
                        <a href=<?= base_url('booking/hapusbooki
ng/' . $t['id_buku']); ?>" onclick="return_konfirm('Yakin tidak Jadi Booking '.$t
['judul_buku'])"><i class="btn btn-sm btn-outline-danger fas fw fa-
trash"></i></a>
                </td>
            </tr>
            <?php $no++;
        } ?>
        </table>
    </div>
</td>
</tr>
<tr>
    <td colspan="3">
        <hr>
    </td>
</tr>
<tr>
    <td colspan="3">
        <a class="btn btn-sm btn-outline-
primary" href=<?php echo base_url(); ?>"><span class="fas fw fa-
play"></span> Lanjutkan Booking Buku</a>
        <a class="btn btn-sm btn-outline-
success" href=<?php echo base_url() . 'booking/bookingSelesai/' . $this-
>session->userdata('id_user'); ?>"><span class="fas fw fa-
stop"></span> Selesaikan Booking</a>
    </td>
</tr>
</table>
</center>
</div>

```

Pada view data-booking, terdapat 2 tombol. **Lanjutkan Booking Buku** dan **Selesaikan Booking**. Untuk tombol Lanjutkan Booking Buku, system akan mengarahkan ke controller awal (Home.php) untuk melanjutkan booking buku

```

<a class="btn btn-sm btn-outline-
primary" href=<?php echo base_url(); ?>"><span class="fas fw fa-
play"></span> Lanjutkan Booking Buku</a>

```

Sedangkan tombol Selesaikan Booking, system mengarahkan ke fungsi bookingSelesai yang ada di controller Booking.

```
<a class="btn btn-sm btn-outline-success" href="php echo base_url() . 'booking/bookingSelesai/' . $this-&gt;session-&gt;userdata('id_user'); ?&gt;"&gt;&lt;span class="fas fw fa-stop"&gt;&lt;/span&gt; Selesaikan Booking&lt;/a&gt;</pre
```

2. Membuat fungsi hapusbooking

function ini akan di jalankan ketika member melakukan interaksi pada data booking dengan mengklik tombol hapus.

Pustaka Beranda Booking 1 Buku Profil Saya Log out Selamat Datang Maruloh, M.Kom

No.	Buku	Penulis	penerbit	Tahun	Pilihan
1		Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	

Lanjutkan Booking Buku Selesaikan Booking

Dalam fungsi ini terdapat script untuk menghapus data booking buku yang ada pada tabel temp. buatlah sebuah fungsi baru pada controller Booking.php. ketikkan script berikut:

```
public function hapusbooking()
{
    $id_buku = $this->uri->segment(3);
    $id_user = $this->session->userdata('id_user');

    $this->ModelBooking->deleteData(['id_buku' => $id_buku], 'temp');
    $kosong = $this->db->query("select*from temp where id_user='$id_user'")->num_rows();

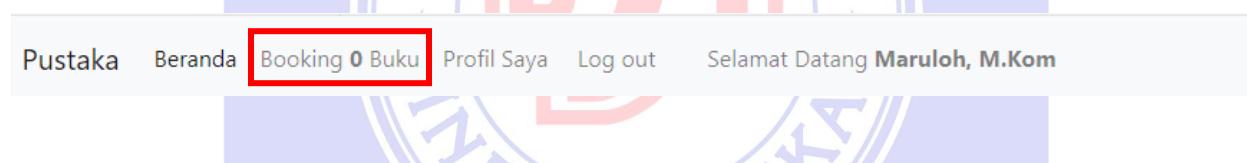
    if ($kosong < 1) {
        $this->session->set_flashdata('pesan', '<div class="alert alert-massege alert-danger" role="alert">Tidak Ada Buku dikeranjang</div>');
        redirect(base_url());
    } else {
        redirect(base_url() . 'booking');
    }
}
```

Pada script tersebut terdapat perintah untuk menghapus data booking yang terdapat pada tabel temp berdasarkan id buku yang dipilih. Dan menampilkan kembali data terbaru (setelah dihapus). Jika data terbaru ternyata buku sudah tidak ada lagi (kosong) maka sistem akan menampilkan pesan. Setelah proses hapusbooking, sistem secara otomatis mengarahkan ke controller Booking.php

```
$this->ModelBooking->deleteData(['id_buku' => $id_buku], 'temp');
    $kosong = $this->db->query("select*from temp where id_user='".$id_user"->num_rows();

    if ($kosong < 1) {
        $this->session->set_flashdata('pesan', '<div class="alert alert-massege alert-danger" role="alert">Tidak Ada Buku dikeranjang</div>');
        redirect(base_url());
    } else {
        redirect(base_url() . 'booking');
    }
```

Hasil setelah proses hapus booking, booking buku kembali kosong



Tugas 4

1. Mahasiswa mengerjakan materi yang ada di pertemuan 5 yaitu membuat Halaman Lihat Keranjang Peminjaman dan Selesai Booking
2. Point 1 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

PERTEMUAN 5

Membuat Halaman Lihat Keranjang Peminjaman dan Selesai Booking

Deskripsi:

Membahas bagaimana cara menyelesaikan booking buku pada halaman keranjang peminjaman. Fungsi selesai booking buku digunakan untuk menyelesaikan transaksi peminjaman dan fungsi view info berfungsi untuk menampilkan hasil akhir dari booking buku. Anggota dapat melakukan cetak bukti booking buku dalam bentuk pdf. Pada pembahasan ini akan digunakan library DOMPDF untuk membuat buktincetak tersebut.

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. Membuat tampilan hasil booking
2. Memahami pembuatan cetak bukti booking dengan library DOMPDF

A. Membuat Fungsi Selesai Booking

Pada tahap ini. untuk menyelesaikan booking buku, maka member harus melakukan interaksi pada web dengan melakukan klik pada tombol **selesaikan booking** yang ada pada tampilan keranjang.

No.	Buku	Penulis	penerbit	Tahun	Pilihan
1		Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	<input type="checkbox"/>

► Lanjutkan Booking Buku  Selesaikan Booking

Agar tombol selesaikan booking bisa berjalan, maka kita akan menambahkan function baru pada file controller Booking.php yaitu function **bookingSelesai**. Berikut kode program fungsi bookingSelesai

```
public function bookingSelesai($where)
{
```

```

        //mengupdate stok dan dibooking di tabel buku saat proses booking diselesaikan
        $this->db-
>query("UPDATE buku, temp SET buku.dibooking=buku.dibooking+1, buku.stok=buku.sto
k-1 WHERE buku.id=temp.id_buku");

        $tglsekarang = date('Y-m-d');
        $isibooking = [
            'id_booking' => $this->ModelBooking-
>kodeOtomatis('booking', 'id_booking'),
            'tgl_booking' => date('Y-m-d H:m:s'),
            'batas_ambil' => date('Y-m-
d', strtotime('+2 days', strtotime($tglsekarang))),
            'id_user' => $where
        ];

        //menyimpan ke tabel booking dan detail booking, dan mengosongkan tabel t
emporari
        $this->ModelBooking->insertData('booking', $isibooking);
        $this->ModelBooking->simpanDetail($where);
        $this->ModelBooking->kosongkanData('temp');

        redirect(base_url() . 'booking/info');
    }

```

Pada fungsi bookingSelesai terdapat beberapa perintah yang dijalankan, diantaranya pengupdate-an stok dan info buku yang dibooking.

```

$this->db-
>query("UPDATE buku, temp SET buku.dibooking=buku.dibooking+1, buku.stok=buku.sto
k-1 WHERE buku.id=temp.id_buku");

```

Terdapat juga fungsi untuk melakukan proses penyimpanan pada tabel booking, detail booking, dan penghapusan data pada tabel temp. sistem akan secara otomatis mengarahkan ke fungsi info pada controller Booking.php

```

$this->ModelBooking->insertData('booking', $isibooking);
    $this->ModelBooking->simpanDetail($where);
    $this->ModelBooking->kosongkanData('temp');

    redirect(base_url() . 'booking/info');

```

B. Membuat Fungsi Info

Fungsi info dipanggil saat sistem mengeksekusi fungsi booking selesai. Pada fungsi ini dilakukan pemanggilan terhadap view info-booking. Dimana info booking ini akan menampilkan data akhir dari proses booking buku. Buatlah fungsi baru dengan nama **info** didalam controller Booking.php dan ketikkan script berikut:

```
public function info()
{
    $where = $this->session->userdata('id_user');
    $data[ 'user' ] = $this->session->userdata('nama');
    $data[ 'judul' ] = "Selesai Booking";
    $data[ 'useraktif' ] = $this->ModelUser->cekData(['id' => $this->session->userdata('id_user')])->result();
    $data[ 'items' ] = $this->db->query("select*from booking bo, booking_detail d, buku bu where d.id_booking=bo.id_booking and d.id_buku=bu.id and bo.id_user='$where'")->result_array();

    $this->load->view('templates/templates-user/header', $data);
    $this->load->view('booking/info-booking', $data);
    $this->load->view('templates/templates-user/modal');
    $this->load->view('templates/templates-user/footer');
}
```

C. Membuat View Info Booking

Seperti dijelaskan sebelumnya, view info booking digunakan untuk menampilkan data akhir dari proses booking buku. Buatlah file dengan nama **info-booking.php** dalam folder **Application/views/booking**. Ketikkan script berikut:

```
<div class="container">
<center>
<table>
<?php
foreach ($useraktif as $u) {
?>
<tr>
    <td nowrap>Terima Kasih <b><?= $u->nama; ?></b>
    </td>
</tr>
<tr>
    <td>Buku Yang ingin Anda Pinjam Adalah Sebagai berikut:</td>
</tr>
```

```

<?php } ?>
<tr>
    <td>
        <div class="table-responsive">
            <table class="table table-bordered table-striped table-hover" id="table-datatable">
                <tr>
                    <th>No.</th>
                    <th>Buku</th>
                    <th>Penulis</th>
                    <th>Penerbit</th>
                    <th>Tahun</th>
                </tr>
                <?php
                    $no = 1;
                    foreach ($items as $i) {
                ?>
                <tr>
                    <td><?= $no; ?></td>
                    <td>
                        
                    </td>
                    <td nowrap><?= $i['pengarang']; ?></td>
                    <td nowrap><?= $i['penerbit']; ?></td>
                    <td nowrap><?= $i['tahun_terbit']; ?></td>
                </tr>
                <?php $no++;
                } ?>
            </table>
        </div>
    </td>
</tr>
<tr>
    <td>
        <hr>
    </td>
</tr>
<tr>
    <td>
        <a class="btn btn-sm btn-outline-
danger" onclick="information('Waktu Pengambilan Buku 1x24 jam dari Booking!!!')"
        href="<?php echo base_url() . 'booking/exportToPdf/' . $this->session-
        >userdata('id_user'); ?>"><span class="far fa-lg fa-fw fa-file-
        pdf"></span> Pdf</a>

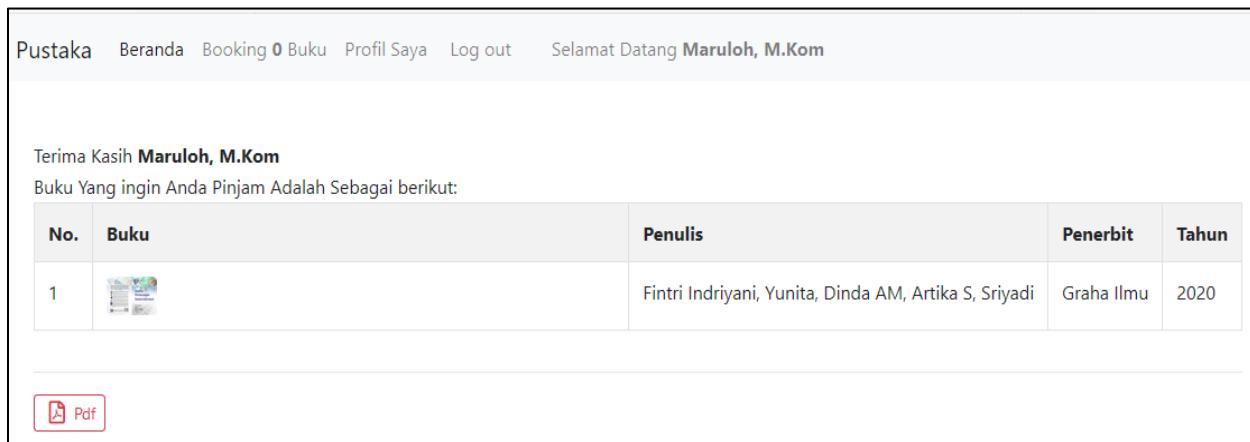
```

```

        </td>
    </tr>
</table>
</center>
</div>

```

Hasil dari view **info-booking.php** adalah sebagai berikut:



Pustaka Beranda Booking 0 Buku Profil Saya Log out Selamat Datang **Maruloh, M.Kom**

Terima Kasih **Maruloh, M.Kom**
Buku Yang ingin Anda Pinjam Adalah Sebagai berikut:

No.	Buku	Penulis	Penerbit	Tahun
1		Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020

[Pdf](#)

Dapat dilihat data booking kembali menjadi 0. Dan terdapat tombol cetak pdf sebagai bukti booking saat pengambilan buku.

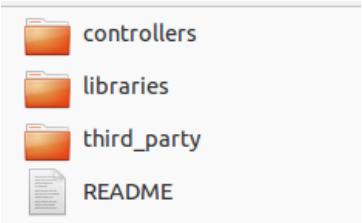
D. Mencetak Bukti Booking

Untuk membuat cetak laporan PDF pada PHP, kita bisa menggunakan library DOMPDF. DOMPDF salah satu library PHP yang bisa di gunakan untuk membuat cetak laporan PDF dengan sangat mudah.

1. Penggunaan DOMPDF

Langkah-langkah menggunakan DOMPDF adalah sebagai berikut:

- pertama kita download terlebih dahulu librarynya dompdf for codeigniter, untuk download silakan klik link [disini](#)
- selanjutnya, ekstrak library dompdf yang sudah kita download. hasil ekstrak folder dompdf to codeigniter akan terlihat seperti pada gambar



- c. pindahkan atau copy folder dompdf pada folder third_party ke folder application/third_party aplikasi kita. pindahkan juga file dompdf_gen.php dari application/libraries ke folder application/libraries aplikasi kita
2. Membuat fungsi exportToPdf

Tambahkan sebuah fungsi baru pada controller Booking.php dengan nama exportToPdf, ketikkan script berikut:

```
public function exportToPdf()
{
    $id_user = $this->session->userdata('id_user');
    $data['user'] = $this->session->userdata('nama');
    $data['judul'] = "Cetak Bukti Booking";
    $data['useraktif'] = $this->ModelUser->cekData(['id' => $this->session->userdata('id_user')])->result();
    $data['items'] = $this->db->query("select*from booking bo, booking_detail d, buku bu where d.id_booking=bo.id_booking and d.id_buku=bu.id and bo.id_user='$id_user'")->result_array();

    $this->load->library('dompdf_gen');

    $this->load->view('booking/bukti-pdf', $data);

    $paper_size  = 'A4'; // ukuran kertas
    $orientation = 'landscape'; //tipe format kertas potrait atau landscape
    $html = $this->output->get_output();

    $this->dompdf->set_paper($paper_size, $orientation);
    //Convert to PDF
    $this->dompdf->load_html($html);
    $this->dompdf->render();
    $this->dompdf->stream("bukti-booking-
$id_user.pdf", array('Attachment' => 0));
    // nama file pdf yang di hasilkan
}
```

Ukuran kertas dapat diatur sesuai dengan kebutuhan, begitupun nama file pdf yang akan dihasilkan dapat kita rubah sesuai keinginan

```
$paper_size = 'A4'; // ukuran kertas
```

```
$this->dompdf->stream("bukti-booking-$id_user.pdf", array('Attachment' => 0));
```

3. Membuat view bukti-pdf

View bukti-pdf berisi data hasil cetakan dari pdf, kita bisa mengatur seperti apa tampilannya. Berikut scrip **bukti-pdf.php** yang dibuat pada folder **Application-views-booking**

```
<table border=1>
    <?php
        foreach ($useraktif as $u) {
    ?>
        <tr>
            <th>Nama Anggota : <?= $u->nama; ?></th>
        </tr>
        <tr>
            <th>Buku Yang dibooking:</th>
        </tr>
    <?php } ?>
    <tr>
        <td>
            <div class="table-responsive">
                <table border=1>
                    <tr>
                        <th>No.</th>
                        <th>Buku</th>
                        <th>Penulis</th>
                        <th>penerbit</th>
                        <th>Tahun</th>
                    </tr>
                    <?php
                        $no = 1;
                        foreach ($items as $i) {
                    ?>
                    <tr>
                        <td><?= $no; ?></td>
                        <td>
```

```

                <?= $i['judul_buku']; ?>
            </td>
            <td><?= $i['pengarang']; ?></td>
            <td><?= $i['penerbit']; ?></td>
            <td><?= $i['tahun_terbit']; ?></td>
        </tr>
        <?php $no++;
    } ?>
    </table>
</div>
</td>
</tr>
<tr>
    <td>
        <hr>
    </td>
</tr>
<tr>
    <td align="center">
        <?= md5(date('d M Y H:i:s')); ?>
    </td>
</tr>
</table>

```

hasil dari cetak bukti pdf:

Nama Anggota : Maruloh, M.Kom				
Buku Yang dibooking:				
No.	Buku	Penulis	penerbit	Tahun
1	Analisa Perancangan Sistem Informasi	Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020
57ab56750233425dbdf5a30aa2fe2306				

Tugas 5

1. Mahasiswa mengerjakan materi yang ada di pertemuan 6 yaitu membuat Halaman Pengambilan dan Pengembalian Buku
2. Point 1 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

PERTEMUAN 6

Membuat Halaman Pengambilan dan Pengembalian Buku

Deskripsi:

Membahas mengenai pengolahan data booking buku yang dilakukan anggota. Admin melakukan proses perubahan dari booking menjadi pinjam saat anggota mengambil buku, dan merubah status buku yang dipinjam menjadi kembali saat buku telah dikembalikan

Tujuan Pembelajaran:

Setelah melakukan bagian ini mahasiswa mampu:

1. mengolah data booking menjadi pinjam
2. menampilkan detail booking buku
3. mengolah data pinjam menjadi kembali

A. Merubah Index Controller Admin

Sebelum kita memulai proses pengambilan dan pengembalian buku, terlebih dahulu kita edit controller Admin. Jika sebelumnya index controller admin berisi tampilan data yang ada di database, seperti data buku, data anggota, buku dibooking dll. Sekarang kita akan modifikasi controller dengan menambahkan script update data buku setelah dibooking, serta hapus otomatis data booking saat buku tidak diambil/dipinjam oleh anggota.

Oleh karena silakan buka kembali controller admin, dan sesuaikan dengan script berikut:

```
<?php
defined('BASEPATH') or exit('No direct script access allowed');

class Admin extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load-
>model(['ModelBuku', 'ModelUser', 'ModelBooking', 'ModelPinjam']);
        cek_login();
    }
}
```

```

        cek_user();
    }

    public function index()
    {
        $data['judul'] = 'Dashboard';
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
        $data['anggota'] = $this->ModelUser->getUserLimit()->result_array();
        $data['buku'] = $this->ModelBuku->getLimitBuku()->result_array();

        //mengupdate stok dan dibooking pada tabel buku
        $detail = $this->db-
>query("SELECT*FROM booking,booking_detail WHERE DAY(curdate()) < DAY(batas_ambil)
) AND booking.id_booking=booking_detail.id_booking")->result_array();
        foreach ($detail as $key) {
            $id_buku = $key['id_buku'];
            $batas = $key['tgl_booking'];
            $tgławal = date_create($batas);
            $tglskrg = date_create();
            $beda = date_diff($tgławal, $tglskrg);

            if ($beda->days > 2) {
                $this->db-
>query("UPDATE buku SET stok=stok+1, dibooking=dibooking-1 WHERE id='$id_buku'");
            }
        }

        //menghapus otomatis data booking yang sudah lewat dari 2 hari
        $booking = $this->ModelBooking->getData('booking');
        if (!empty($booking)) {
            foreach ($booking as $bo) {
                $id_booking = $booking->id_booking;
                $tglbooking = $booking->tgl_booking;
                $tgławal = date_create($tglbooking);
                $tglskrg = date_create();
                $beda = date_diff($tgławal, $tglskrg);

                if ($beda->days > 2) {
                    $this->db-
>query("DELETE FROM booking WHERE id_booking='$id_booking'");
                    $this->db-
>query("DELETE FROM booking_detail WHERE id_booking='$id_booking'");
                }
            }
        }
    }
}

```

```
}

    $this->load->view('templates/header', $data);
    $this->load->view('templates/sidebar', $data);
    $this->load->view('templates/topbar', $data);
    $this->load->view('admin/index', $data);
    $this->load->view('templates/footer');

}

}
```

Pada ModelBuku silakan tambahkan fungsi getLimitBuku() dengan script berikut:

```
public function getLimitBuku()
{
    $this->db->limit(5);
    return $this->db->get('buku');
}
```

B. Membuat Menu Transaksi

Untuk membuat menu laporan buku, kita harus membuka kembali project pustaka-booking yang pernah kita kerjakan pada Pemrograman Web II. Silakan buka file sidebar.php yang berada di folder **application/views/templates/sidebar.php**, silakan tambahkan coding berikut:

UNIVERSITAS

```

        <a class="nav-link pb-0" href="#"><?= base_url('user/anggota'); ?>>
            <i class="fa fa-fw fa-users"></i>
            <span>Data Anggota</span></a>
        </li>
        </li>

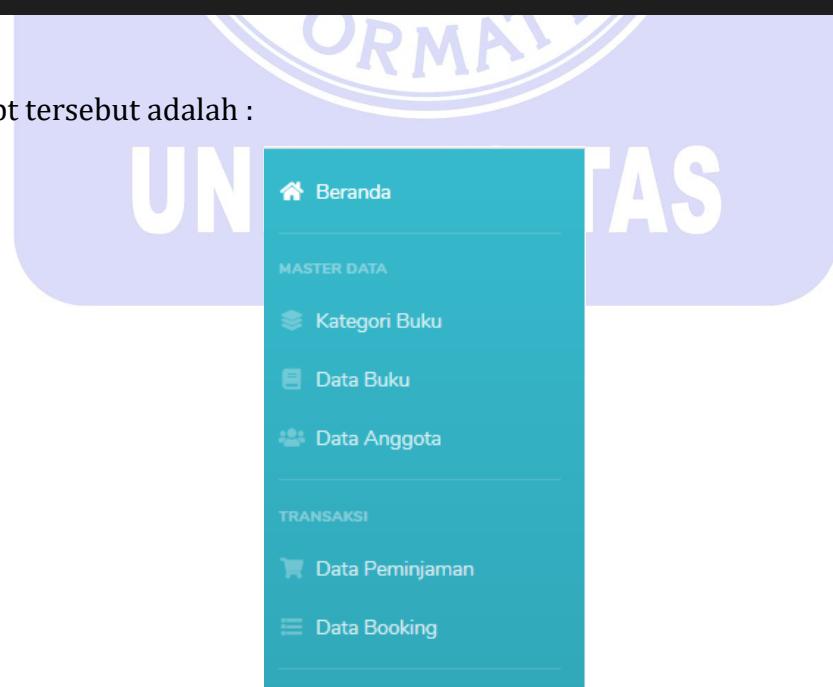
        <!-- Divider -->
        <hr class="sidebar-divider mt-3">

        <!-- Heading -->
        <div class="sidebar-heading">
            Transaksi
        </div>
        <!-- Nav Item - Dashboard -->
        <li class="nav-item active">
            <!-- Nav Item - Dashboard -->
        <li class="nav-item">
            <a class="nav-link pb-0" href="#"><?= base_url('pinjam'); ?>>
                <i class="fa fa-fw fa-shopping-cart"></i>
                <span>Data Peminjaman</span></a>
            </li>
            <li class="nav-item">
                <a class="nav-link pb-0" href="#"><?= base_url('pinjam/daftarBooking'); ?>>
                    <i class="fa fa-fw fa-list"></i>
                    <span>Data Booking</span></a>
                </li>
            </li>
        </li>

        <!-- Divider -->
        <hr class="sidebar-divider mt-3">

```

Hasil dari script tersebut adalah :



C. Membuat Halaman Data Booking

Pada sidebar.php yang telah kita edit diatas. sebelum menampilkan halaman data booking buku, terlebih dahulu kita buat fungsi daftar booking yang ada di controller pinjam

```
<li class="nav-item">
    <a class="nav-link pb-0" href="= base_url('pinjam/daftarBooking'); ?&gt;"&gt;
        &lt;i class="fa fa-fw fa-list"&gt;&lt;/i&gt;
        &lt;span&gt;Data Booking&lt;/span&gt;&lt;/a&gt;</pre
```

Oleh karena itu diawal kita buat terlebih dahulu controller **Pinjam.php** dan menambahkan fungsi daftarBooking di controller tersebut

```
<?php if (!defined('BASEPATH')) exit('No Direct Script Access Allowed');

class Pinjam extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model(['ModelUser', 'ModelBuku', 'ModelPinjam']);
        cek_login();
        cek_user();
    }

    public function index()
    {
    }

    public function daftarBooking()
    {
        $data['judul'] = "Daftar Booking";
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
        $data['pinjam'] = $this->db->query("select*from booking")->result_array();

        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
        $this->load->view('templates/topbar', $data);
        $this->load->view('booking/daftar-booking', $data);
        $this->load->view('templates/footer');
    }
}
```

Pada controller Pinjam.php, model yang digunakan adalah ModelBuku, ModelUser, dan ModelPinjam. Fungsi daftarBooking digunakan untuk mengirim data judul, user, dan data booking buku, serta memanggil view daftar-booking yang ada pada folder booking.

Langkah berikutnya adalah kita membuat view dengan nama **daftar-booking.php** didalam folder **Application/views/booking**. Ketikkan script berikut:

```
<div class="container">

<center>
    <table>
        <tr>
            <td>
                <div class="table-responsive full-width">
                    <table class="table table-bordered table-striped table-hover" id="table-datable">
                        <tr>
                            <th>No.</th>
                            <th>ID Booking</th>
                            <th>Tanggal Booking</th>
                            <th>ID User</th>
                            <th>Aksi</th>
                            <th>Denda / Buku /Hari</th>
                            <th>Lama Pinjam</th>
                        </tr>
                        <?php
                            $no = 1;
                            foreach ($pinjam as $p) {
                        ?>
                        <tr>
                            <td><?= $no; ?></td>
                            <td><a class="btn btn-
link" href="<?= base_url('pinjam/bookingDetail/' . $p['id_booking']); ?>"><?= $p[
'id_booking']; ?></a></td>
                            <td><?= $p['tgl_booking']; ?></td>
                            <td><?= $p['id_user']; ?></td>
                            <form action="<?= base_url('pinjam/pinjamAct/
' . $p['id_booking']); ?>" method="post">
                                <td nowrap>
                                    <button type="submit" class="btn btn-
sm btn-outline-info"><i class="fas fa-fw fa-cart-plus"></i> Pinjam</button>
                                </td>
                            <td>
```

```

        <input class="form-check-
user rounded-
sm" style="width:100px" type="text" name="denda" id="denda" value="5000">
            <?= form_error(); ?>
        </td>
        <td>
            <input class="form-check-
user rounded-sm" style="width:100px" type="text" name="lama" id="lama" value="3">
            <?= form_error(); ?>
        </td>
    </form>
</tr>
<?php $no++;
} ?>
</table>
</div>
</td>
</tr>
<tr>
    <td align="center"><a href=<?= base_url('pinjam/daftarBooking'); ?>" class="btn btn-link"><i class="fas fa-fw fa-refresh"></i> Refresh</a></td>
</tr>
</table>
</center>
</div>

```

Hasil dari view daftar-booking.php adalah sebagai berikut:

Daftar Booking

No.	ID Booking	Tanggal Booking	ID User	Aksi	Denda / Buku /Hari	Lama Pinjam
1	23042020001	2020-04-23	2	 Pinjam	5000	3

Refresh

Pada tampilan diatas, terlihat 3 tombol (link) yang bisa dipilih oleh admin. Yaitu **tombol Pinjam** saat member ingin mengambil buku, **link ID Booking** untuk melihat detail buku yang dibooking, dan **Link Refresh** untuk merefresh halaman daftar-booking.

Untuk link ID Booking sistem mengarahkan ke fungsi **bookingDetail** yang ada di controller Pinjam.php (harus buat fungsi bookingDetail)

```
<td><a class="btn btn-link" href="= base_url('pinjam/bookingDetail/' . $p['id_booking']); ?&gt;"&gt;&lt;?= $p['id_booking']; ?&gt;&lt;/a&gt;&lt;/td&gt;</pre

```

Untuk tombol Pinjam, sistem akan mengarahkan ke fungsi pinjamAct yang juga ada di controller Pinjam.php (harus buat fungsi pinjamAct)

```
<form action="= base_url('pinjam/pinjamAct/' . $p['id_booking']); ?&gt;" method="post"&gt;</pre

```

Sedangkan untuk link refresh, sistem hanya akan mengarahkan ke fungsi daftarBooking yang ada controller Pinjam (tidak perlu membuat fungsi daftarBooking, karena sudah dibuat)

```
<td align="center"><a href="= base_url('pinjam/daftarBooking'); ?&gt;" class="btn btn-link"&gt;&lt;i class="fas fa-fw fa-refresh"&gt;&lt;/i&gt; Refresh&lt;/a&gt;&lt;/td&gt;</pre

```

1. Membuat Fungsi bookingDetail

Seperti penjelasan sebelumnya, untuk melihat detail booking, kita akan membuat fungsi **bookingDetail** dalam controller Pinjam.php. Ketikkan scrip berikut :

```
public function bookingDetail()
{
    $id_booking = $this->uri->segment(3);
    $data['judul'] = "Booking Detail";
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
    $data['agt_booking'] = $this->db->query("select*from booking b, user u where b.id_user=u.id and b.id_booking='$id_booking'")->result_array();
    $data['detail'] = $this->db->query("select id_buku,judul_buku,pengarang,penerbit,tahun_terbit from booking_detail d, buku b where d.id_buku=b.id and d.id_booking='$id_booking'")->result_array();

    $this->load->view('templates/header', $data);
    $this->load->view('templates/sidebar', $data);
    $this->load->view('templates/topbar', $data);
    $this->load->view('booking/booking-detail', $data);
    $this->load->view('templates/footer');
}
```

Script diatas, akan memanggil view booking-detail yang ada di folder booking. Oleh karena itu kita akan membuat sebuah view baru di folder Application/views/booking dengan nama booking-detail.php. berikut scriptnya:

```
<div class="container">
<center>
<table>
    <?php foreach ($agt_booking as $ab) { ?>
        <tr>
            <td>Data Anggota</td>
            <td>:</td>
            <th><?= $ab['nama']; ?></th>
        </tr>
        <tr>
            <td>ID Booking</td>
            <td>:</td>
            <th><?= $ab['id_booking']; ?></th>
        </tr>
        <?php } ?>
        <tr>
            <td colspan="3">&ampnbsp</td>
        </tr>
        <tr>
            <td colspan="3">
                <hr>
            </td>
        </tr>
        <tr>
            <td colspan="3">
                <div class="table-responsive full-width">
                    <table class="table table-bordered table-striped table-hover" id="table-datatable">
                        <tr>
                            <th>No.</th>
                            <th>ID Buku</th>
                            <th>Judul Buku</th>
                            <th>Pengarang</th>
                            <th>Penerbit</th>
                            <th>Tahun</th>
                        </tr>
                        <?php
                            $no = 1;
                            foreach ($detail as $d) {
                        ?>
```

```

<tr>
    <td><?= $no; ?></td>
    <td><?= $d[ 'id_buku' ]; ?></td>
    <td><?= $d[ 'judul_buku' ]; ?></td>
    <td><?= $d[ 'pengarang' ]; ?></td>
    <td><?= $d[ 'penerbit' ]; ?></td>
    <td><?= $d[ 'tahun_terbit' ]; ?></td>
</tr>
<?php $no++;
} ?>
</table>
</div>
</td>
</tr>
<tr>
    <td align="center" colspan="3"><a href="#" onclick="window.history.go(-1)" class="btn btn-outline-dark"><i class="fas fa-fw fa-reply"></i> Kembali</a></td>
</tr>
</table>
</center>
</div>

```

Hasil dari detail booking:

Booking Detail

Imam Nawawi 

Data Anggota	:	Maruloh, M.Kom
ID Booking	:	23042020001

No.	ID Buku	Judul Buku	Pengarang	Penerbit	Tahun
1	4	Analisa Perancangan Sistem Informasi	Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020

 Kembali

2. Membuat Fungsi pinjamAct

Untuk menjalankan mengeksekusi tombol pinjam, kita harus membuat terlebih dahulu fungsi pinjamAct di dalam controller Pinjam.php

Adapun scriptnya adalah sebagai berikut:

```

public function pinjamAct()
{
    $id_booking = $this->uri->segment(3);
    $lama = $this->input->post('lama', TRUE);

    $bo = $this->db-
>query("SELECT*FROM booking WHERE id_booking='".$id_booking"")->row();

    $tglsekarang = date('Y-m-d');
    $no_pinjam = $this->ModelBooking->kodeOtomatis('pinjam', 'no_pinjam');
    $databooking = [
        'no_pinjam' => $no_pinjam,
        'id_booking' => $id_booking,
        'tgl_pinjam' => $tglsekarang,
        'id_user' => $bo->id_user,
        'tgl_kembali' => date('Y-m-
d', strtotime('+ ' . $lama . ' days', strtotime($tglsekarang))),
        'tgl_pengembalian' => '0000-00-00',
        'status' => 'Pinjam',
        'total_denda' => 0
    ];
    $this->ModelPinjam->simpanPinjam($databooking);
    $this->ModelPinjam->simpanDetail($id_booking, $no_pinjam);
    $denda = $this->input->post('denda', TRUE);
    $this->db->query("update detail_pinjam set denda='$denda'");

    //hapus Data booking yang bukunya diambil untuk dipinjam
    $this->ModelPinjam->deleteData('booking', ['id_booking' => $id_booking]);
    $this->ModelPinjam-
>deleteData('booking_detail', ['id_booking' => $id_booking]);
    // $this->db->query("DELETE FROM booking WHERE id_booking='".$id_booking"");

    //update dibooking dan dipinjam pada tabel buku saat buku yang dibooking
    //diambil untuk dipinjam
    $this->db-
>query("UPDATE buku, detail_pinjam SET buku.dipinjam=buku.dipinjam+1, buku.dibooking=buku.dibooking-1 WHERE buku.id=detail_pinjam.id_buku");

    $this->session->set_flashdata('pesan', '<div class="alert alert-
message alert-success" role="alert">Data Peminjaman Berhasil Disimpan</div>');
    redirect(base_url() . 'pinjam');
}

```

Pada fungsi tambahAct, tanggal kembali akan ditentukan dengan menambahkan lama pinjam dengan tanggal yang berlaku di sistem.

```
$lama = $this->input->post('lama', TRUE);
```

```
$tglsekarang = date('Y-m-d');
```

```
'tgl_kembali' => date('Y-m-d', strtotime('+ ' . $lama . ' days', strtotime($tglsekarang))),
```

Data yang dikirim akan disimpan kedalam tabel pinjam dan detail pinjam, dan data booking serta data detail booking otomatis dihapus (pindah ke tabel pinjam dan detail)

```
$this->ModelPinjam->simpanPinjam($databooking);
$this->ModelPinjam->simpanDetail($id_booking);
```

```
$this->ModelPinjam->deleteData('booking', ['id_booking' => $id_booking]);
$this->ModelPinjam->deleteData('booking_detail', ['id_booking' => $id_booking]);
```

Data buku dipinjam akan ditambah 1 dan data buku booking akan dihapus 1

```
$this->db->query("UPDATE buku, detail_pinjam SET buku.dipinjam=buku.dipinjam+1, buku.dibooking=buku.dibooking-1 WHERE buku.id=detail_pinjam.id_buku");
```

Sistem secara otomatis akan mengalihkan ke controller Pinjam.php

```
$this->session->set_flashdata('pesan', '<div class="alert alert-message alert-success" role="alert">Data Peminjaman Berhasil Disimpan</div>');
redirect(base_url() . 'pinjam');
```

D. Membuat Halaman Data Peminjaman

1. Membuat Fungsi Index pada Controller Pinjam

fungsi index ini akan dijalankan pertama kali saat sistem memanggil controller Pinjam.

Tambahkan scrip index pada controller pinjam sebagai berikut:

```
public function index()
{
    $data['judul'] = "Data Pinjam";
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
    $data['pinjam'] = $this->ModelPinjam->joinData();

    $this->load->view('templates/header', $data);
```

```

    $this->load->view('templates/sidebar', $data);
    $this->load->view('templates/topbar', $data);
    $this->load->view('pinjam/data-pinjam', $data);
    $this->load->view('templates/footer');
}

```

Dari script diatas, terlihat bahwa saat controller Pinjam dipanggil, maka dia akan meload halaman view data-pinjam yang ada di folder pinjam. Termasuk mengirimkan data-data seperti user, termasuk isian data yang akan ditampilkan. Silakan tambahkan sebuah file view baru dengan nama **data-pinjam.php** didalam folder **Application/views/pinjam** dan ketikkan script berikut:

```

<div class="container">
    <center>
        <table>
            <tr>
                <td>
                    <div class="table-responsive full-width">
                        <table class="table table-bordered table-striped table-hover" id="table-datatable">
                            <tr>
                                <th>No Pinjam</th>
                                <th>Tanggal Pinjam</th>
                                <th>ID User</th>
                                <th>ID Buku</th>
                                <th>Tanggal Kembali</th>
                                <th>Tanggal Pengembalian</th>
                                <th>Terlambat</th>
                                <th>Denda</th>
                                <th>Status</th>
                                <th>Total Denda</th>
                                <th>Pilihan</th>
                            </tr>
                            <?php

```



```

                                foreach ($pinjam as $p) {
                            ?>
                            <tr>
                                <td><?= $p['no_pinjam']; ?></td>
                                <td><?= $p['tgl_pinjam']; ?></td>
                                <td><?= $p['id_user']; ?></td>
                                <td><?= $p['id_buku']; ?></td>

```

```

<td><?= $p['tgl_kembali']; ?></td>
<td>
    <?= date('Y-m-d'); ?>
    <input type="hidden" name="tgl_pengembalian" id="tgl_pengembalian" value="<?= date('Y-m-d'); ?>">
</td>
<td>
    <?php
        $tgl1 = new DateTime($p['tgl_kembali']);
        $tgl2 = new DateTime();
        $selisih = $tgl2->diff($tgl1)-
    >format("%a");
    echo $selisih;
    ?> Hari
</td>
<td><?= $p['denda']; ?></td>

<?php if ($p['status'] == "Pinjam") {
    $status = "warning";
} else {
    $status = "secondary";
} ?>
<td><i class="btn btn-outline-
<?= $status; ?> btn-sm"><?= $p['status']; ?></i></td>

<?php
if ($selisih < 0) {
    $total_denda = $p['denda'] * 0;
} else {
    $total_denda = $p['denda'] * $selisih;
}
?>

<td><?= $total_denda; ?>
    <input type="hidden" name="totaldenda" id="totaldenda" value="<?= $total_denda; ?>">
</td>
<td nowrap>
    <?php if ($p['status'] == "Kembali") { ?>
        <i class="btn btn-sm btn-outline-
secondary"><i class="fas fa-fw fa-edit"></i>Ubah Status</i>
        <?php } else { ?>
            <a class="btn btn-sm btn-outline-
info" href="<?= base_url('pinjam/ubahStatus/' . $p['id_buku'] . '/' . $p['no_pinjam']); ?>"><i class="fas fa-fw fa-edit"></i>Ubah Status</a>

```

```

        <?php } ?>
    </td>
</tr>
<?php
} ?>
</table>
</div>
</td>
</tr>

</table>
</center>
</div>

```



Hasil dari script diatas adalah :

Data Pinjam

Imam Nawawi

No Pinjam	Tanggal Pinjam	ID User	ID Buku	Tanggal Kembali	Tanggal Pengembalian	Terlambat	Denda	Status	Total Denda	Pilihan
23042020001	2020-04-23	2	4	2020-04-26	2020-04-23	2 Hari	5000	Pinjam	10000	Ubah

Didalam halaman view data-pinjam.php, terdapat sebuah link untuk merubah status peminjaman dari pinjam menjadi kembali. Link tersebut mengarahkan ke fungsi ubahStatus dengan meminta data id_buku dan no_pinjam yang ada di controller Pinjam.php

```

<td nowrap>
    <?php if ($p['status'] == "Kembali") { ?>
        <i class="btn btn-sm btn-outline-secondary"><i class="fas fa-fw fa-edit"></i>Ubah Status</i>
    <?php } else { ?>

```

```

        <a class="btn btn-sm btn-outline-
info" href="= base_url('pinjam/ubahStatus/' . $p['id_buku'] . '/' . $p['no_pinj
am']); ?&gt;&lt;i class="fas fa-fw fa-edit"&gt;&lt;/i&gt;Ubah Status&lt;/a&gt;
    &lt;?php } ?&gt;
    &lt;/td&gt;
</pre

```

Karena itu langkah berikutnya kita akan buat sebuah fungsi dengan nama **ubahStatus** di dalam controller Pinjam.php. adapun scriptnya adalah sebagai berikut:

```

public function ubahStatus()
{
    $id_buku = $this->uri->segment(3);
    $no_pinjam = $this->uri->segment(4);
    $where = ['id_buku' => $this->uri->segment(3),];

    $tgl = date('Y-m-d');
    $status = 'Kembali';
    //update status menjadi kembali pada saat buku dikembalikan
    $this->db-
>query("UPDATE pinjam, detail_pinjam SET pinjam.status='$status', pinjam.tgl_peng
embalian='$tgl' WHERE detail_pinjam.id_buku='$id_buku' AND pinjam.no_pinjam='$no_
pinjam'");

    //update stok dan dipinjam pada tabel buku
    $this->db-
>query("UPDATE buku, detail_pinjam SET buku.dipinjam=buku.dipinjam-
1, buku.stok=buku.stok+1 WHERE buku.id=detail_pinjam.id_buku");

    $this->session->set_flashdata('pesan', '<div class="laert alert-
message alert-success" role="alert"></div>');
    redirect(base_url('pinjam'));
}

```

Pada script fungsi ubah status, akan diambil data berupa id buku dan no.pinjam, kemudian merubah status menjadi kembali. Sistem juga akan mengupdate data buku dipinjam menambah stok buku yang dikembalikan.

```

$id_buku = $this->uri->segment(3);
$no_pinjam = $this->uri->segment(4);

```

```

$this->db-
>query("UPDATE pinjam, detail_pinjam SET pinjam.status='$status', pinjam.tgl_peng
embalian='$tgl' WHERE detail_pinjam.id_buku='$id_buku' AND pinjam.no_pinjam='$no_
pinjam'");

```

```
$this->db->query("UPDATE buku, detail_pinjam SET buku.dipinjam=buku.dipinjam-1, buku.stok=buku.stok+1 WHERE buku.id=detail_pinjam.id_buku");
```

Hasil dari script :

Data Pinjam

Imam Nawawi 

No Pinjam	Tanggal Pinjam	ID User	ID Buku	Tanggal Kembali	Tanggal Pengembalian	Terlambat	Denda	Status	Total Denda	Pilihan
23042020001	2020-04-23	2	4	2020-04-26	2020-04-23	2 Hari	5000	Kembali	10000	<input checked="" type="button"/> Ubah Status

Tugas 6

1. Mahasiswa mengerjakan materi yang ada di pertemuan 7 yaitu membuat dan Mencetak Laporan Data Buku ke Printer
2. Point 1 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya



PERTEMUAN 7

Membuat dan Mencetak Laporan Data Buku ke Printer

Deskripsi:

Membahas bagaimana cara membuat dan memodifikasi pencetakan laporan data buku dengan menggunakan printer

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. Membuat halaman tampilan cetak buku
2. Membuat dan mencetak laporan data buku dengan media printer

A. Membuat Menu Laporan

Untuk membuat menu laporan buku, kita harus membuka kembali project pustaka-booking yang pernah kita kerjakan pada Pemrograman Web II. Silakan buka file sidebar.php yang berada di folder **application/views/templates/sidebar.php**, silakan tambahkan coding berikut:

UNIVERSITAS

```

<!-- Heading -->


| Laporan


<!-- Nav Item - Dashboard -->
- | <!-- Nav Item - Dashboard -->
- | <a class="nav-link pb-0" href="#">></a>
        | <span>Laporan Data Buku</span></a>

</li>
<li class="nav-item">
    | <a class="nav-link pb-0" href="#">></a>
        | <span>Laporan Data Anggota</span></a>

</li>
<li class="nav-item">
    | <a class="nav-link pb-0" href="#">></a>
        | <span>Laporan Peminjaman</span></a>

</li>
</li>

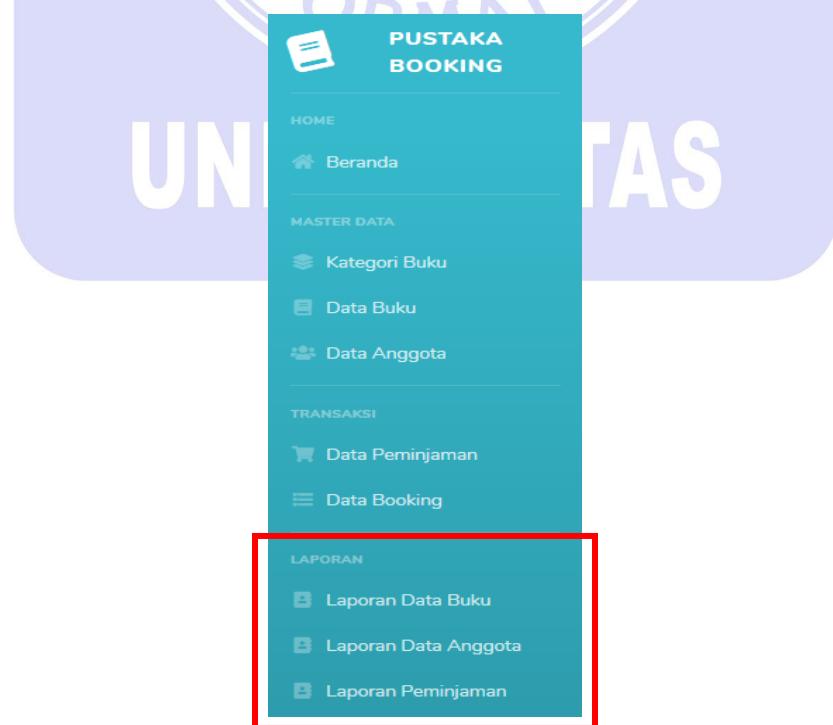
<!-- Divider -->
<hr class="sidebar-divider mt-3">

<!-- Sidebar Toggler (Sidebar) -->


| <button class="rounded-circle border-0" id="sidebarToggle"></button>


```

Hasil dari coding diatas adalah sebagai berikut:



B. Membuat halaman Laporan Buku

Langkah-langkah untuk membuat halaman laporan buku adalah sebagai berikut:

1. Buatlah Controller baru dengan nama **Laporan**.
2. Buat fungsi **laporan_buku** pada controller Laporan.

```
<?php

defined('BASEPATH') or exit('No Direct script access allowed');

class Laporan extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->model(['ModelUser', 'ModelBuku', 'ModelPinjam']);

    }
    public function laporan_buku()
    {
        $data['judul'] = 'Laporan Data Buku';
        $data['user'] = $this->ModelUser->cekData(['email' => $this->session->userdata('email')])->row_array();
        $data['buku'] = $this->ModelBuku->getBuku()->result_array();
        $data['kategori'] = $this->ModelBuku->getKategori()->result_array();

        $this->load->view('templates/header', $data);
        $this->load->view('templates/sidebar', $data);
        $this->load->view('templates/topbar', $data);
        $this->load->view('buku/laporan_buku', $data);
        $this->load->view('templates/footer');
    }
}
```

Fungsi tersebut digunakan untuk memanggil view **laporan_buku** yang ada didalam folder **buku**, dengan mengirimkan data buku kedalamnya.

3. Setelah itu kita buat file laporan_buku.php untuk menampilkan data buku di dalam folder **application/views/buku/laporan_buku.php**

```
<!-- Begin Page Content -->

<div class="container-fluid">

    <?= $this->session->flashdata('pesan'); ?>
    <div class="row">
        <div class="col-lg-12">
```

```

<?php if(validation_errors()){?>
    <div class="alert alert-danger" role="alert">
        <?= validation_errors();?>
    </div>
<?php }?>
<?= $this->session->flashdata('pesan'); ?>
<a href="= base_url('laporan/cetak_laporan_buku'); ?&gt;" class="btn btn-primary mb-3"&gt;&lt;i class="fas fa-print"&gt;&lt;/i&gt; Print&lt;/a&gt;
&lt;a href="<?= base_url('laporan/laporan_buku_pdf'); ?&gt;" class="btn btn-warning mb-3"&gt;&lt;i class="far fa-file-pdf"&gt;&lt;/i&gt; Download Pdf&lt;/a&gt;
&lt;a href="<?= base_url('laporan/export_excel'); ?&gt;" class="btn btn-success mb-3"&gt;&lt;i class="far fa-file-excel"&gt;&lt;/i&gt; Export ke Excel&lt;/a&gt;
&lt;table class="table table-hover"&gt;
    &lt;thead&gt;
        &lt;tr&gt;
            &lt;th scope="col"&gt;#&lt;/th&gt;
            &lt;th scope="col"&gt;Judul&lt;/th&gt;
            &lt;th scope="col"&gt;Pengarang&lt;/th&gt;
            &lt;th scope="col"&gt;Penerbit&lt;/th&gt;
            &lt;th scope="col"&gt;Tahun Terbit&lt;/th&gt;
            &lt;th scope="col"&gt;ISBN&lt;/th&gt;
            &lt;th scope="col"&gt;Stok&lt;/th&gt;
        &lt;/tr&gt;
    &lt;/thead&gt;
    &lt;tbody&gt;

        &lt;?php
            $a = 1;
            foreach ($buku as $b) { ?&gt;
        &lt;tr&gt;
            &lt;th scope="row"&gt;&lt;?= $a++; ?&gt;&lt;/th&gt;
            &lt;td&gt;&lt;?= $b['judul_buku']; ?&gt;&lt;/td&gt;
            &lt;td&gt;&lt;?= $b['pengarang']; ?&gt;&lt;/td&gt;
            &lt;td&gt;&lt;?= $b['penerbit']; ?&gt;&lt;/td&gt;
            &lt;td&gt;&lt;?= $b['tahun_terbit']; ?&gt;&lt;/td&gt;
            &lt;td&gt;&lt;?= $b['isbn']; ?&gt;&lt;/td&gt;
            &lt;td&gt;&lt;?= $b['stok']; ?&gt;&lt;/td&gt;
        &lt;/tr&gt;
        &lt;?php } ?&gt;
    &lt;/tbody&gt;
&lt;/table&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;!-- /.container-fluid --&gt;
</pre

```

```
</div>
<!-- End of Main Content -->
```

4. Hasil dari coding di atas adalah sebagai berikut:

#	Judul	Pengarang	Penerbit	Tahun Terbit	ISBN	Stok
1	Membuat Web dengan Framework Codeigniter; Studi Kasus Sistem Informasi Perpustakaan	Gunawan Budi Sulisty; Pudji Widodo	Graha Ilmu	2020	978-623-228-403-6	6
2	Web Programming Membangun Aplikasi Perpustakaan dengan Framework CodeIgniter	Marloh, Nining Suryani, Evy Priyanti	Graha Ilmu	2020	978-623-228-220-9	9
3	Aplikasi Basis Data; Mahir Menggunakan SQL	M. Nandi S, Rahayu N, Widiarina, Suharyanto	Graha Ilmu	2020	978-623-228-180-6	12
4	Analisa Perancangan Sistem Informasi	Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	978-623-228-179-0	12
5	Belajar Mudah Pemrograman Web Backend Programming; Studi Kasus Membangun Aplikasi Reservasi Perpustakaan	Imam Nawawi, Adriansah, Frans. E. Schaduw	Graha Ilmu	2020	978-623-228-212-4	14

C. Membuat Print Laporan Data Buku

Berdasarkan view laporan buku yang sudah kita buat sebelumnya, untuk membuat print laporan data buku, kita diharuskan menambahkan fungsi **cetak_laporan_buku** di dalam controller Laporan.

```
<?= $this->session->flashdata('pesan'); ?>
 class="btn btn-primary mb-3"> Print
 class="btn btn-warning mb-3"> Download Pdf
 class="btn btn-success mb-3"> Export ke Excel

||
||
||


```

Adapun langkah-langkah detail dalam membuat Print Laporan Data Buku adalah sebagai berikut:

1. Tambahkan fungsi **cetak_laporan_buku** pada controller Laporan

```

public function cetak_laporan_buku()
{
    $data['buku'] = $this->ModelBuku->getBuku()->result_array();
    $data['kategori'] = $this->ModelBuku->getKategori()->result_array();

    $this->load->view('buku/laporan_print_buku', $data);
}

```

2. Membuat file laporan_print_buku.php pada folder
application/views/buku/laporan_print_buku.php

```

<!DOCTYPE html>

<html>
<head>
    <title></title>
</head>
<body>
    <style type="text/css">
        .table-data{
            width: 100%;
            border-collapse: collapse;
        }

        .table-data tr th,
        .table-data tr td{
            border:1px solid black;
            font-size: 11pt;
            font-family:Verdana;
            padding: 10px 10px 10px 10px;
        }
        h3{
            font-family:Verdana;
        }
    </style>

    <h3><center>Laporan Data Buku Perputakaan Online</center></h3>
    <br/>
    <table class="table-data">
        <thead>
            <tr>
                <th>No</th>
                <th>Judul Buku</th>
                <th>Pengarang</th>
                <th>Terbit</th>

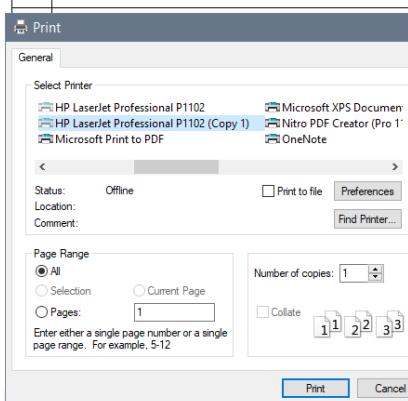
```

```
<th>Tahun Penerbit</th>
<th>ISBN</th>
<th>Stok</th>
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach($buku as $b){
?>
<tr>
<th scope="row"><?= $no++; ?></th>
<td><?= $b['judul_buku']; ?></td>
<td><?= $b['pengarang']; ?></td>
<td><?= $b['penerbit']; ?></td>
<td><?= $b['tahun_terbit']; ?></td>
<td><?= $b['isbn']; ?></td>
<td><?= $b['stok']; ?></td>
</tr>
<?php
}
?>
</tbody>
</table>

<script type="text/javascript">
window.print();
</script>

</body>
</html>
```

3. Hasil dari codingan tersebut adalah :

No	Judul Buku	Pengarang	Terbit	Tahun Penerbit	ISBN	Stok
1	Membuat Web dengan Framework Codeigniter; Studi Kasus Sistem Informasi Perpustakaan	Gunawan Budi Sulistyo; Pudji Widodo	Graha Ilmu	2020	978-623-228-403-6	6
		Maruloh, Nining Suryani, Evy Priyanti	Graha Ilmu	2020	978-623-228-220-9	9
		M. Nandi S, Rahayu N, Widiarina, Suharyanto	Graha Ilmu	2020	978-623-228-180-6	12
		Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	978-623-228-179-0	12
		Imam Nawawi, Adriansah, Frans. E. Schaduw	Graha Ilmu	2020	978-623-228-212-4	14
		Ruhul Amin, Fernando B. Siahaan, Dicky Hariyanto	Published	2020	978-623-228-194-3	19
		Anton Subagia	Elex Media komputindo	2017	43345356577	10
		Deasyalwati	Elex Media Komputindo	2015	87968686787879	8
		Rudi Himawan	Pusat Perbukuan	2016	97868687978796	13

Tugas 7

1. Mahasiswa mengerjakan materi yang ada di pertemuan 9 yaitu membuat dan Mencetak Laporan Data Buku ke Tampilan Pdf dan Excel
2. Point 1 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

UNIVERSITAS

PERTEMUAN 8
Ujian Tengah Semester (UTS)

Tidak ada perkuliahan



UNIVERSITAS

PERTEMUAN 9

Membuat dan Mencetak Laporan Data Buku ke Tampilan Pdf dan Excel

Deskripsi:

Membahas bagaimana cara membuat dan memodifikasi pencetakan laporan data buku kedalam bentuk Pdf dan Ms. Excel

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. Membuat dan mencetak laporan yang dirubah menjadi bentuk Pdf
2. Membuat dan mencetak laporan yang dirubah menjadi bentuk Ms. Excel

A. Membuat Cetak Laporan Data Buku Pdf

Langkah-langkah untuk melakukan cetak laporan data buku pdf adalah sebagai berikut:

1. Pastikan DOMPDF sudah diatur seperti pada langkah sebelumnya
2. Menambahkan fungsi **laporan_buku_pdf** pada **controller Laporan**

```
<?= $this->session->flashdata('pesan'); ?>
<a href="= base_url('laporan/cetak_laporan_buku'); ?&gt;" class="btn btn-primary mb-3"&gt;&lt;i class="fas fa-print"&gt;&lt;/i&gt; Print&lt;/a&gt;
&lt;a href="<?= base_url('laporan/laporan_buku_pdf'); ?&gt;" class="btn btn-warning mb-3"&gt;&lt;i class="far fa-file-pdf"&gt;&lt;/i&gt; Download Pdf&lt;/a&gt;
&lt;a href="<?= base_url('laporan/export_excel'); ?&gt;" class="btn btn-success mb-3"&gt;&lt;i class="far fa-file-excel"&gt;&lt;/i&gt; Export ke Excel&lt;/a&gt;
&lt;table class="table table-hover"&gt;
    &lt;thead&gt;
        &lt;tr&gt;</pre
```

Menambahkan fungsi **laporan_buku_pdf**

```

public function laporan_buku_pdf()
{
{
    $this->load->library('dompdf_gen');

    $data['buku'] = $this->ModelBuku->getBuku()->result_array();

    $this->load->view('buku/laporan_pdf_buku', $data);

    $paper_size = 'A4'; // ukuran kertas
    $orientation = 'landscape'; //tipe format kertas potrait atau landscape
    $html = $this->output->get_output();

    $this->dompdf->set_paper($paper_size, $orientation);
    //Convert to PDF
    $this->dompdf->load_html($html);
    $this->dompdf->render();
    $this->dompdf->stream("laporan_data_buku.pdf", array('Attachment' => 0));
    // nama file pdf yang di hasilkan
}
}

```

Teman-teman bisa mengatur ukuran kertas, format kertas serta nama file pdf yang akan di hasilkan nantinya

3. Buat view baru dengan nama **laporan_pdf_buku**

Application/views/buku/laporan_pdf_buku.php

```

<!DOCTYPE html>
<html>
<head>
| <title></title>
</head>
<body>
<style type="text/css">
.table-data{
    width: 100%;
    border-collapse: collapse;
}

.table-data tr th,
.table-data tr td{
    border:1px solid black;
    font-size: 11pt;
    padding: 10px 10px 10px 10px;
}
</style>

```

```

<h3><center>Laporan Data Buku Perputakaan Online</center></h3>
<br/>
<table class="table-data">
  <thead>
    <tr>
      <th>No</th>
      <th>Judul Buku</th>
      <th>Pengarang</th>
      <th>Terbit</th>
      <th>Tahun Penerbit</th>
      <th>ISBN</th>
      <th>Stok</th>
    </tr>
  </thead>

  <tbody>
    <?php
    $no = 1;
    foreach($buku as $b){
    ?>
    <tr>
      <th scope="row"><?= $no++; ?></th>
      <td><?= $b['judul_buku']; ?></td>
      <td><?= $b['pengarang']; ?></td>
      <td><?= $b['penerbit']; ?></td>
      <td><?= $b['tahun_terbit']; ?></td>
      <td><?= $b['isbn']; ?></td>
      <td><?= $b['stok']; ?></td>
    </tr>
    <?php
    }
    ?>
  </tbody>
</table>
</body>
</html>

```

4. Hasil dari codingan tersebut adalah sebagai berikut:

Laporan Data Buku Perputakaan Online

No	Judul Buku	Pengarang	Terbit	Tahun Penerbit	ISBN	Stok
1	Membuat Web dengan Framework Codeigniter; Studi Kasus Sistem Informasi Perpustakaan	Gunawan Budi Sulistyo; Pudji Widodo	Graha Ilmu	2020	978-623-228-403-6	6
2	Web Programming Membangun Aplikasi Perpustakaan dengan Framework CodeIgniter	Maruloh, Nining Suryani, Evy Priyanti	Graha Ilmu	2020	978-623-228-220-9	9
3	Aplikasi Basis Data; Mahir Menggunakan SQL	M. Nandi S, Rahayu N, Widiarina, Suharyanto	Graha Ilmu	2020	978-623-228-180-6	12
4	Analisa Perancangan Sistem Informasi	Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	978-623-228-179-0	12
5	Belajar Mudah Pemrograman Web Backend Programming; Studi Kasus Membangun Aplikasi Reservasi Perpustakaan	Imam Nawawi, Adriansah, Frans. E. Schaduw	Graha Ilmu	2020	978-623-228-212-4	14
6	Dasar Pemrograman	Ruhul Amin, Fernando B. Siahaan, Dicky Hariyanto	Published	2020	978-623-228-194-3	19

B. Mengekspor Data kedalam File Excel

Langkah-langkah untuk melakukan cetak laporan data buku pdf adalah sebagai berikut:

1. Menambahkan fungsi **export_excel** pada controller **Laporan**

```
public function export_excel()
{
    $data = array( 'title' => 'Laporan Buku',
        'buku' => $this->ModelBuku->getBuku()->result_array();
        $this->load->view('buku/export_excel_buku', $data);
}
```

2. Membuat file **export_excel_buku.php** pada folder **Application/views/buku/export_excel_buku.php**

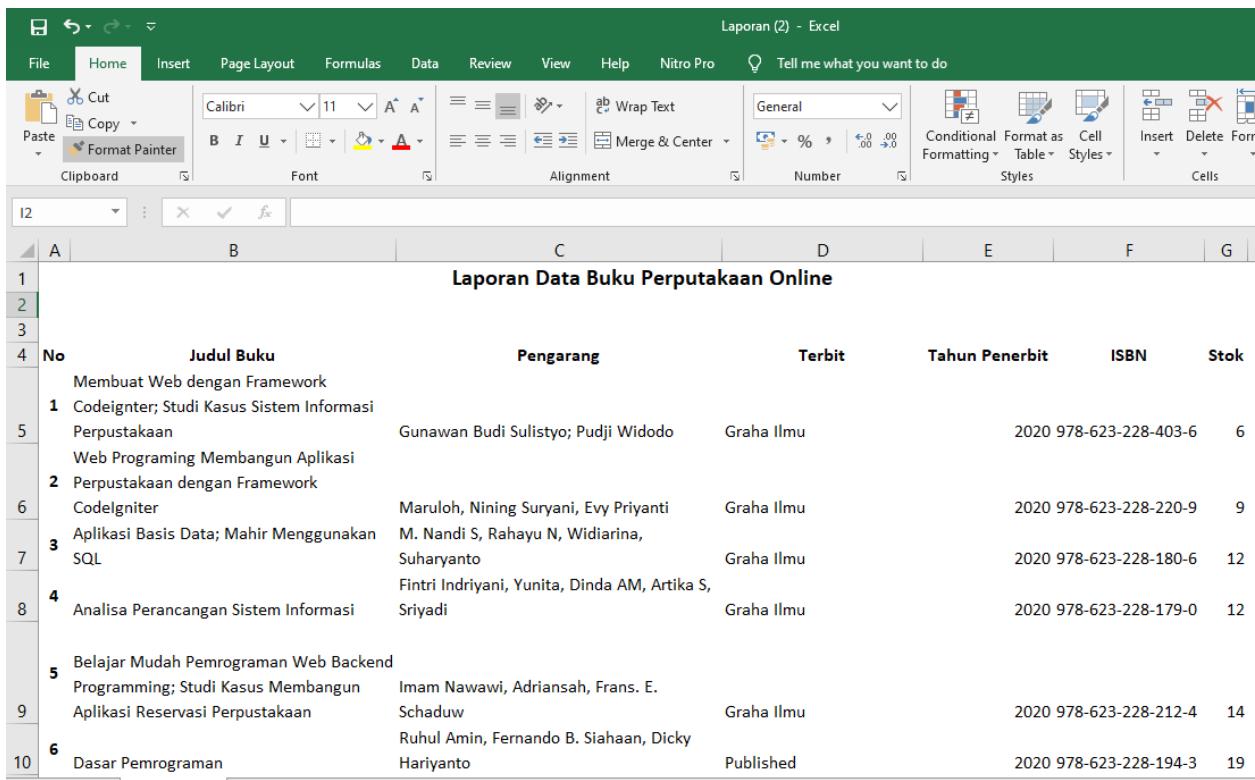
```

<?php
header("Content-type: application/vnd-ms-excel");
header("Content-Disposition: attachment; filename=$title.xls");
header("Pragma: no-cache");
header("Expires: 0");
?>

<h3><center>Laporan Data Buku Perputakaan Online</center></h3>
<br/>
<table class="table-data">
<thead>
<tr>
<th>No</th>
<th>Judul Buku</th>
<th>Pengarang</th>
<th>Terbit</th>
<th>Tahun Penerbit</th>
<th>ISBN</th>
<th>Stok</th>
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach($buku as $b){
?>
<tr>
<th scope="row"><?= $no++; ?></th>
<td><?= $b['judul_buku']; ?></td>
<td><?= $b['pengarang']; ?></td>
<td><?= $b['penerbit']; ?></td>
<td><?= $b['tahun_terbit']; ?></td>
<td><?= $b['isbn']; ?></td>
<td><?= $b['stok']; ?></td>
</tr>
<?php
}
?>
</tbody>
</table>

```

3. Hasil dari eksport data:



Laporan Data Buku Perputakaan Online							
No	Judul Buku	Pengarang	Terbit	Tahun Penerbit	ISBN	Stok	
1	Membuat Web dengan Framework Codeigniter; Studi Kasus Sistem Informasi Perpustakaan	Gunawan Budi Sulistyо; Pudji Widodo	Graha Ilmu	2020	978-623-228-403-6	6	
2	Web Programming Membangun Aplikasi Perpustakaan dengan Framework CodeIgniter	Maruloh, Nining Suryani, Evy Priyanti	Graha Ilmu	2020	978-623-228-220-9	9	
3	Aplikasi Basis Data; Mahir Menggunakan SQL	M. Nandi S, Rahayu N, Widiarina, Suharyanto	Graha Ilmu	2020	978-623-228-180-6	12	
4	Analisa Perancangan Sistem Informasi	Fintri Indriyani, Yunita, Dinda AM, Artika S, Sriyadi	Graha Ilmu	2020	978-623-228-179-0	12	
5	Belajar Mudah Pemrograman Web Backend Programming; Studi Kasus Membangun Aplikasi Reservasi Perpustakaan	Imam Nawawi, Adriansah, Frans. E. Schaduw	Graha Ilmu	2020	978-623-228-212-4	14	
6	Dasar Pemrograman	Ruhul Amin, Fernando B. Siahaan, Dicky Haryanto	Published	2020	978-623-228-194-3	19	

Tugas 8

1. Buat laporan Data anggota dalam bentuk print out (Print), bentuk Pdf, dan bentuk excel (Untuk panduan silakan sihat kembali materi pembuatan laporan data anggota pada pertemuan 7 dan 9)
2. Mahasiswa mengerjakan materi yang ada di pertemuan 10 yaitu membuat laporan peminjaman dan setting Akses Client-Server
3. Point 1 dan 2 akan didemonstrasikan dan dikumpulkan di pertemuan selanjutnya

PERTEMUAN 10

Membuat Laporan Peminjaman Dan Setting Akses Client-Server

Deskripsi:

Membahas bagaimana cara membuat transaksi peminjaman dengan menggunakan printer, format pdf, dan Ms. Excel serta melakukan setting akses client-server.

Tujuan Pembelajaran:

Setelah melakukan bagian ini pembaca mampu:

1. Membuat dan mencetak laporan transaksi peminjaman dengan media printer
2. Membuat dan mencetak laporan transaksi peminjaman dengan format pdf
3. Membuat dan mencetak laporan transaksi peminjaman dengan format Ms. Excel
4. Membuat Akses Client-Server
5. Melakukan setting Jaringan dan IP menggunakan Share wifi melalui HP

A. Membuat Halaman Laporan Peminjaman

Yang terakhir yang akan kita buat adalah fitur laporan transaksi. Pada dasarnya pembuatan laporan peminjaman sama seperti pembuatan laporan buku, hanya saja pada laporan peminjaman ada beberapa tabel yang terlibat selain tabel pinjam itu sendiri. Untuk outputnya masih sama yaitu cetak printer, cetak pdf, dan eksport excel

Coba teman-teman perhatikan pada menu laporan yang ada di file sidebar.php. Kita telah membuat link nya menuju ke fungsi laporan_pinjam() di controller Laporan.php

```

<!-- Heading -->


| Laporan


<!-- Nav Item - Dashboard -->
- | <!-- Nav Item - Dashboard -->
- <a class="nav-link pb-0" href="= base_url('laporan/laporan_buku'); ?&gt;"&gt;
            | &lt;i class="fa fa-fw fa-address-book"&gt;&lt;/i&gt;
            | &lt;span&gt;Laporan Data Buku&lt;/span&gt;&lt;/a&gt;
</li
</li>
- <a class="nav-link pb-0" href="= base_url('laporan/laporan_anggota'); ?&gt;"&gt;
            | &lt;i class="fa fa-fw fa-address-book"&gt;&lt;/i&gt;
            | &lt;span&gt;Laporan Data Anggota&lt;/span&gt;&lt;/a&gt;
</li
</li>
- <a class="nav-link pb-0" href="= base_url('laporan/laporan_pinjam'); ?&gt;"&gt;
            | &lt;i class="fa fa-fw fa-address-book"&gt;&lt;/i&gt;
            | &lt;span&gt;Laporan Peminjaman&lt;/span&gt;&lt;/a&gt;
</li
</li>

<!-- Divider -->


---


    | <button class="rounded-circle border-0" id="sidebarToggle"></button>

```

Oleh karena itu kita akan membuat fungsi baru lagi di controller Laporan.php. yaitu fungsi laporan_pinjam().

Application/controllers/Laporan.php

```

public function laporan_pinjam()
{
    $data['judul'] = 'Laporan Data Peminjaman';
    $data['user'] = $this->ModelUser->cekData(['email' => $this->session-
>userdata('email')])->row_array();
    $data['laporan'] = $this->db-
>query("select * from pinjam p,detail_pinjam d,
        buku b,user u where d.id_buku=b.id and p.id_user=u.id
        and p.no_pinjam=d.no_pinjam")->result_array();

    $this->load->view('templates/header', $data);
    $this->load->view('templates/sidebar');
    $this->load->view('templates/topbar', $data);
    $this->load->view('pinjam/laporan-pinjam', $data);
}

```

```

        $this->load->view('templates/footer');
    }

```

Pada script diatas, data laporan tidak hanya memanggil data pada tabel pinjam saja, tetapi juga melibatkan tabel detail_pinjam, user, dan buku.

```

$data['laporan'] = $this->db->query("select * from pinjam p,detail_pinjam d,
                                         buku b,user u where d.id_buku=b.id and p.id_user=u.id
                                         and p.no_pinjam=d.no_pinjam")->result_array();

```

berdasarkan script diatas, kita akan membuat sebuah view baru dalam folder pinjam. Dengan nama **laporan-pinjam.php**

Application/views/pinjam/laporan-pinjam.php

```

<!-- Begin Page Content -->
<div class="container-fluid">

    <?= $this->session->flashdata('pesan'); ?>
    <div class="row">
        <div class="col-lg-12">
            <?php if(validation_errors()){?>
                <div class="alert alert-danger" role="alert">
                    <?= validation_errors();?>
                </div>
            <?php }?>
            <?= $this->session->flashdata('pesan'); ?>
            <a href="= base_url('laporan/cetak_laporan_pinjam'); ?&gt;" class="btn
            btn-primary mb-3"&gt;&lt;i class="fas fa-print"&gt;&lt;/i&gt; &lt;/a&gt;
            &lt;a href="<?= base_url('laporan/laporan_pinjam_pdf'); ?&gt;" class="btn
            btn-warning mb-3"&gt;&lt;i class="far fa-file-pdf"&gt;&lt;/i&gt; &lt;/a&gt;
            &lt;a href="<?= base_url('laporan/export_excel_pinjam'); ?&gt;" class="btn
            btn-success mb-3"&gt;&lt;i class="far fa-file-excel"&gt;&lt;/i&gt; &lt;/a&gt;
            &lt;table class="table table-hover"&gt;
                &lt;thead&gt;
                    &lt;tr&gt;
                        &lt;th scope="col"&gt;#&lt;/th&gt;
                        &lt;th scope="col"&gt;Nama Anggota&lt;/th&gt;
                        &lt;th scope="col"&gt;Judul Buku&lt;/th&gt;
                        &lt;th scope="col"&gt;Tanggal Pinjam&lt;/th&gt;
                        &lt;th scope="col"&gt;Tanggal Kembali &lt;/th&gt;
                        &lt;th scope="col"&gt;Tanggal Dikembalikan&lt;/th&gt;
                        &lt;th scope="col"&gt;Total Denda&lt;/th&gt;
                        &lt;th scope="col"&gt;Status&lt;/th&gt;
</pre

```

```

        </tr>
    </thead>
    <tbody>

        <?php
            $a = 1;
            foreach ($laporan as $l) { ?>
        <tr>
            <th scope="row"><?= $a++; ?></th>
            <td><?= $l['nama']; ?></td>
            <td><?= $l['judul_buku']; ?></td>
            <td><?= $l['tgl_pinjam']; ?></td>
            <td><?= $l['tgl_kembali']; ?></td>
            <td><?= $l['tgl_pengembalian']; ?></td>
            <td><?= $l['total_denda']; ?></td>
            <td><?= $l['status']; ?></td>
        </tr>
        <?php } ?>
    </tbody>
</table>
</div>
</div>
</div>
<!-- /.container-fluid --&gt;
&lt;/div&gt;
<!-- End of Main Content --&gt;
</pre>

```

Tetap ada 3 aksi yang bisa kita pilih untuk laporan pinjam, yaitu print, cetak pdf, dan eksport ke excel



```

<a href=<?= base_url('laporan/cetak_laporan_pinjam'); ?>" class="btn btn-primary mb-3"><i class="fas fa-print"></i> </a>
        <a href=<?= base_url('laporan/laporan_pinjam_pdf'); ?>" class="btn btn-warning mb-3"><i class="far fa-file-pdf"></i> </a>
        <a href=<?= base_url('laporan/export_excel_pinjam'); ?>" class="btn btn-success mb-3"><i class="far fa-file-excel"></i> </a>

```

Data yang dikirim adalah kombinasi antara tabel pinjam, detail pinjam, user, dan buku.

```

<td><?= $l['nama']; ?></td>
<td><?= $l['judul_buku']; ?></td>
<td><?= $l['tgl_pinjam']; ?></td>
<td><?= $l['tgl_kembali']; ?></td>
<td><?= $l['tgl_pengembalian']; ?></td>
<td><?= $l['total_denda']; ?></td>
<td><?= $l['status']; ?></td>

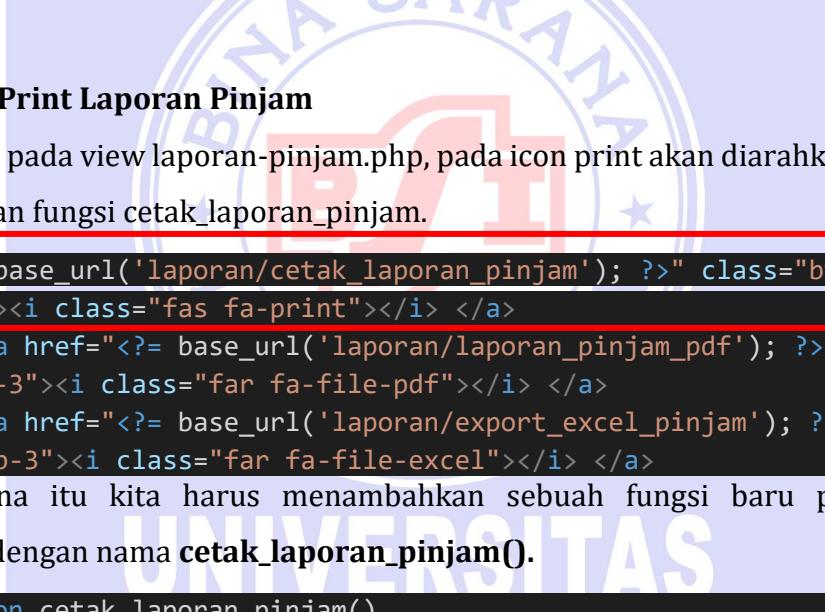
```

Sekarang kita coba klik pada menu “laporan Transaksi”

Maka akan diarahkan ke fungsi laporan_pinjam() yang menampilkan data peminjaman buku.

Laporan Data Peminjaman

Imam Nawawi 



  	Nama Anggota #	Judul Buku	Tanggal Pinjam	Tanggal Kembali	Tanggal Dikembalikan	Total Denda	Status
1	Maruloh, M.Kom	Analisa Perancangan Sistem Informasi	2020-04-23	2020-04-26	2020-04-23	0	Kembali

C. Membuat Print Laporan Pinjam

Perhatikan pada view laporan-pinjam.php, pada icon print akan diarahkan ke controller Laporan.php dan fungsi cetak_laporan_pinjam.

```
<a href="= base_url('laporan/cetak_laporan_pinjam'); ?&gt;" class="btn btn-primary mb-3"&gt;&lt;i class="fas fa-print"&gt;&lt;/i&gt; &lt;/a&gt;
    &lt;a href="<?= base_url('laporan/laporan_pinjam_pdf'); ?&gt;" class="btn btn-warning mb-3"&gt;&lt;i class="far fa-file-pdf"&gt;&lt;/i&gt; &lt;/a&gt;
        &lt;a href="<?= base_url('laporan/export_excel_pinjam'); ?&gt;" class="btn btn-success mb-3"&gt;&lt;i class="far fa-file-excel"&gt;&lt;/i&gt; &lt;/a&gt;</pre
```

Oleh karena itu kita harus menambahkan sebuah fungsi baru pada controller **Laporan.php** dengan nama **cetak_laporan_pinjam()**.

```
public function cetak_laporan_pinjam()
{
    $data['laporan'] = $this->db->query("select * from pinjam p,detail_pinjam d,
        buku b,user u where d.id_buku=b.id and p.id_user=u.id
        and p.no_pinjam=d.no_pinjam")->result_array();

    $this->load->view('pinjam/laporan-print-pinjam', $data);
}
```

Fungsi diatas digunakan untuk memanggil data dan mengirimkannya ke view laporan-print-pinjam yang ada di folder pinjam.

Langkah berikutnya kita akan membuat sebuah view dengan nama **laporan-print-pinjam.php** di dalam folder pinjam

Application/views/pinjam/laporan-print-pinjam.php

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>
<body>
    <style type="text/css">
        .table-data{
            width: 100%;
            border-collapse: collapse;
        }

        .table-data tr th,
        .table-data tr td{
            border:1px solid black;
            font-size: 11pt;
            font-family:Verdana;
            padding: 10px 10px 10px 10px;
        }
        .table-data th{
            background-color:grey;
        }
        h3{
            font-family:Verdana;
        }
    </style>

    <h3><center>LAPORAN DATA PEMINJAMAN BUKU</center></h3>
    <br/>
    <table class="table-data">
        <thead>
            <tr>
                <th>No</th>
                <th>Nama Anggota</th>
                <th>Judul Buku</th>
                <th>Tanggal Pinjam</th>
                <th>Tanggal Kembali</th>
                <th>Tanggal Pengembalian</th>
                <th>Total Denda</th>
                <th>Status</th>
            
```

```

</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach($laporan as $l){
?>
<tr>
<td scope="row"><?= $no++; ?></td>
<td><?= $l['nama']; ?></td>
<td><?= $l['judul_buku']; ?></td>
<td><?= $l['tgl_pinjam']; ?></td>
<td><?= $l['tgl_kembali']; ?></td>
<td><?= $l['tgl_pengembalian']; ?></td>
<td><?= $l['total_denda']; ?></td>
<td><?= $l['status']; ?></td>
</tr>
<?php
}
?>
</tbody>
</table>
<script type="text/javascript">
window.print();
</script>

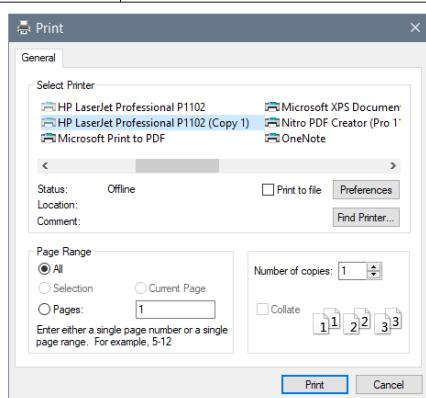
</body>
</html>

```

Hasil akhir dari script diatas adalah :

LAPORAN DATA PEMINJAMAN BUKU

No	Nama Anggota	Judul Buku	Tanggal Pinjam	Tanggal Kembali	Tanggal Pengembalian	Total Denda	Status
1	Maruloh, M.Kom	Analisa Perancangan Sistem Informasi	2020-04-23	2020-04-26	2020-04-23	0	Kembali



D. Membuat Cetak Pdf Laporan Peminjaman

Untuk membuat laporan peminjaman cetak pdf kita buka kembali view laporan-pinjam.php untuk melihat controller dan fungsi apa yang dipanggil.

```
<a href="= base_url('laporan/cetak_laporan_pinjam'); ?" class="btn btn-primary mb-3"><i class="fas fa-print"></i> </a>
<a href="= base_url('laporan/laporan_pinjam_pdf'); ?" class="btn btn-warning mb-3"><i class="far fa-file-pdf"></i> </a>
<a href="= base_url('laporan/export_excel_pinjam'); ?" class="btn btn-success mb-3"><i class="far fa-file-excel"></i> </a>
```

Dari script view laporan-pinjam.php dapat kita lihat bahwa sistem mengarahkan ke controller Laporan.php dan memanggil fungsi laporan_pinjam_pdf. Oleh karena itu langkah berikutnya kita akan menambahkan fungsi baru pada controller **Laporan.php** yang diberi nama **laporan_pinjam_pdf()**

Berikut script fungsi **laporan_pinjam_pdf()**

```
public function laporan_pinjam_pdf()
{
{
    $this->load->library('dompdf_gen');

    $data['laporan'] = $this->db->query("select * from pinjam p,detail_pinjam d,
        buku b,user u where d.id_buku=b.id and p.id_user=u.id
        and p.no_pinjam=d.no_pinjam")->result_array();

    $this->load->view('pinjam/laporan-pdf-pinjam', $data);

    $paper_size  = 'A4'; // ukuran kertas
    $orientation = 'landscape'; //tipe format kertas potrait atau landscape
    $html = $this->output->get_output();

    $this->dompdf->set_paper($paper_size, $orientation);
    //Convert to PDF
    $this->dompdf->load_html($html);
    $this->dompdf->render();
    $this->dompdf-
>stream("laporan data peminjaman.pdf", array('Attachment' => 0));
    // nama file pdf yang di hasilkan
}
}
```

Fungsi laporan_pinjam_pdf() berisi script untuk mengatur tampilan file pdf (orientation, size, name, dll) sedangkan isi dari file pdf sistem akan memanggil view laporan-pdf-pinjam.php yang ada di folder pinjam.

Jadi langkah berikutnya adalah membuat view **laporan-pdf-pinjam.php** untuk menampilkan data dari laporan.

Application/views/pinjam/laporan-pdf-pinjam.php

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body>
<style type="text/css">
.table-data{
    width: 100%;
    border-collapse: collapse;
}

.table-data tr th,
.table-data tr td{
    border:1px solid black;
    font-size: 11pt;
    font-family:Verdana;
    padding: 10px 10px 10px 10px;
}
.table-data th{
    background-color:grey;
}
h3{
    font-family:Verdana;
}
</style>

<h3><center>LAPORAN DATA PEMINJAMAN BUKU</center></h3>
<br/>
<table class="table-data">
<thead>
<tr>
<th>No</th>
<th>Nama Anggota</th>
<th>Judul Buku</th>
```

```

<th>Tanggal Pinjam</th>
<th>Tanggal Kembali</th>
<th>Tanggal Pengembalian</th>
<th>Total Denda</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach($laporan as $l){
?>
<tr>
<td scope="row"><?= $no++; ?></td>
<td><?= $l['nama']; ?></td>
<td><?= $l['judul_buku']; ?></td>
<td><?= $l['tgl_pinjam']; ?></td>
<td><?= $l['tgl_kembali']; ?></td>
<td><?= $l['tgl_pengembalian']; ?></td>
<td><?= $l['total_denda']; ?></td>
<td><?= $l['status']; ?></td>
</tr>
<?php
}
?>
</tbody>
</table>

</body>
</html>

```

Hasil dari file pdf yang telah didownload:

LAPORAN DATA PEMINJAMAN BUKU

No	Nama Anggota	Judul Buku	Tanggal Pinjam	Tanggal Kembali	Tanggal Pengembalian	Total Denda	Status
1	Maruloh, M.Kom	Analisa Perancangan Sistem Informasi	2020-04-23	2020-04-26	2020-04-23	0	Kembali

E. Mengexport Laporan Peminjaman ke Bentuk Excel

Sama seperti cetak laporan sebelumnya, kita lihat kembali view laporan-pinjam.php untuk melihat controller yang dipanggil

```

<a href="<?= base_url('laporan/cetak_laporan_pinjam'); ?>" class="btn btn-primary mb-3"><i class="fas fa-print"></i> </a>

```

```
<a href="= base_url('laporan/laporan_pinjam_pdf'); ?&gt;" class="btn btn-warning mb-3"&gt;&lt;i class="far fa-file-pdf"&gt;&lt;/i&gt; &lt;/a&gt;
&lt;a href="<?= base_url('laporan/export_excel_pinjam'); ?&gt;" class="btn btn-success mb-3"&gt;&lt;i class="far fa-file-excel"&gt;&lt;/i&gt; &lt;/a&gt;</pre
```

Untuk proses export data menjadi excel fungsi yang digunakan adalah export_excel_pinjam yang masih berada di controller Laporan.php

Berikut script fungsi export_excel_pinjam()

```
public function export_excel_pinjam()
{
    $data = array( 'title' => 'Laporan Data Peminjaman Buku',
    'laporan' => $this->db->query("select * from pinjam p,detail_pinjam d,
    buku b,user u where d.id_buku=b.id and p.id_user=u.id
    and p.no_pinjam=d.no_pinjam")->result_array());
    $this->load->view('pinjam/export-excel-pinjam', $data);
}
```

Pada script fungsi export_excel_pinjam() diatur judul untuk laporan yang dibuat, serta mengirim nilai dari tabel yang terlibat didalamnya. Data-data tersebut akan dikirim kedalam file view export-excel-pinjam.php

Langkah terakhir dari pembuatan export data ke excel adalah membuat view dengan nama export-excel-pinjam.php yang ada di folder pinjam

Application/views/pinjam/export-excel-pinjam.php

```
<?php
header("Content-type: application/vnd-ms-excel");
header("Content-Disposition: attachment; filename=$title.xls");
header("Pragma: no-cache");
header("Expires: 0");
?>
<style type="text/css">
.table-data{
    width: 100%;
    border-collapse: collapse;
}

.table-data tr th,
.table-data tr td{
    border:1px solid black;
```

```

    font-size: 11pt;
    font-family: Verdana;
    padding: 10px 10px 10px 10px;
}
.table-data th{
    background-color: grey;
}
h3{
    font-family: Verdana;
}

```

</style>

```

<h3><center>LAPORAN DATA PEMINJAMAN BUKU</center></h3>
<br/>
<table class="table-data" border=1>

<thead>
<tr>
<th>No</th>
<th>Nama Anggota</th>
<th>Judul Buku</th>
<th>Tanggal Pinjam</th>
<th>Tanggal Kembali</th>
<th>Tanggal Pengembalian</th>
<th>Total Denda</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<?php
$no = 1;
foreach($laporan as $l){
?>
<tr>
<td scope="row"><?= $no++; ?></td>
<td><?= $l['nama']; ?></td>
<td><?= $l['judul_buku']; ?></td>
<td><?= $l['tgl_pinjam']; ?></td>
<td><?= $l['tgl_kembali']; ?></td>
<td><?= $l['tgl_pengembalian']; ?></td>
<td><?= $l['total_denda']; ?></td>
<td><?= $l['status']; ?></td>
</tr>
<?php
}

```

```
?>
</tbody>
</table>
```

Hasil akhir dari file yang telah diubah kedalam bentuk excel adalah :

A	B	C	D	E	F	G	H
LAPORAN DATA PEMINJAMAN BUKU							
No	Nama Anggota	Judul Buku	Tanggal Pinjam	Tanggal Kembali	Tanggal Pengembalian	Total Denda	Status
1	Maruloh, M.Kom	Analisa Perancangan Sistem Informasi	23/04/2020	26/04/2020	23/04/2020	0	Kembali

F. Setting Akses Client-Server

Untuk Untuk dapat melakukan pemakaian database bersama, makakita memerlukan sebuah jaringan lokal. Bisa menggunakan kabel jaringan, atau bisa juga menggunakan intranet dalam 1 ruangan, bisa juga menggunakan share wifi menggunakan Handphone (Teathering). dalam kasus ini agar mudah kita gunakan share wifi dengan menggunakan HP. Ada 2 buah perangkat yang akan kita pakai (sebagai contoh). Satu komputer sebagai server (punya database dan program aplikasi penjualan), satu Handphone sebagai client.

Langkah-langkah dalam setting akses client-server

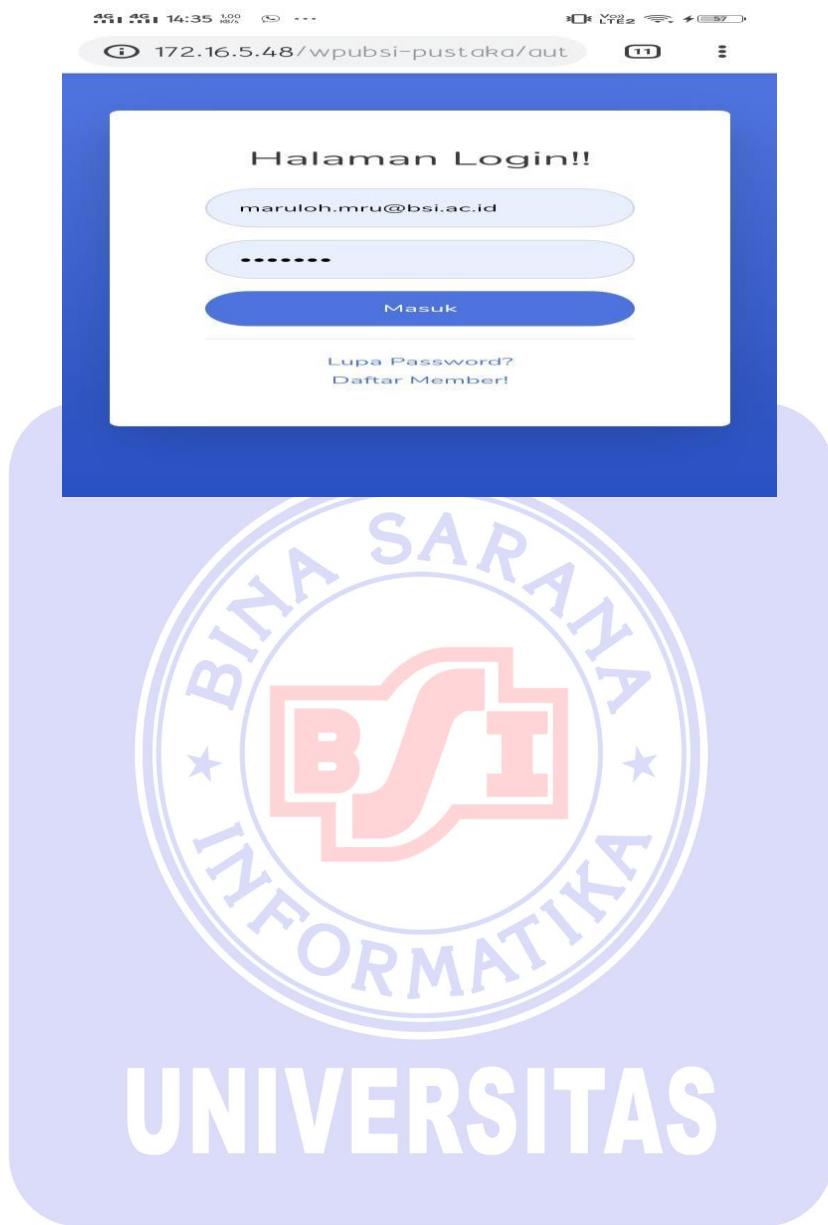
- Pastikan project sudah selesai dikerjakan. Sebagai contoh nama project yang kita buat adalah pustaka-booking
- Lakukan konfigurasi **base_url** pada file **config.php**, seperti berikut:

```
$root=(isset($_SERVER['HTTPS']) ? "https://" : "http://").$_SERVER['HTTP_HOST'];
$root.= str_replace(basename($_SERVER['SCRIPT_NAME']), '',
$_SERVER['SCRIPT_NAME']);
$config['base_url'] = $root;
```

- Pastikan laptop/computer dan handpone berada dalam satu jaringan yang sama (contoh menggunakan hotspot BSI)
- Cek IP** yang ada pada Komputer (server). Contoh IP yang ada di kompuetr server adalah 172.16.5.48
- Akses web menggunakan browser pada handphone (client), dengan mengetikkan **172.16.5.48/nama_project**

Contoh : **172.16.5.48/pustaka-booking/autentifikasi**

6. Hasil yang didapat adalah sebagai berikut:



PERTEMUAN 11-15

Presentasi Project

Pada pertemuan ini, mahasiswa melakukan presentasi kelompok, sesuai dengan jadwal yang telah ditentukan oleh dosen pengajar



UNIVERSITAS

PERTEMUAN 16
Ujian Akhir Semester (UAS)

Tidak ada perkuliahan

