

## **BAB III**

### **PELAKSANAAN KERJA PROFESI**

#### **3.1. Bidang Kerja**

Informatika merupakan salah satu program studi yang ada di Universitas Pembangunan Jaya. Lulusan informatika di Universitas Pembangunan Jaya akan mendapatkan gelar sarjana komputer (S. Kom). Dalam dunia kerja Sarjana Komputer memiliki banyak profesi di bidangnya. Salah satu profesi yang memiliki hubungan dengan bidang informatika adalah web *developer*. Pada kerja profesi ini, praktikan mendapatkan pekerjaan sebagai web *developer* di Klinik Khitan Center Permata Cileungsi. Namun pekerjaan sebagai web *developer* dalam setiap perusahaan bisa saja berbeda-beda. Tugas utama praktikan sebagai web *developer* yaitu membuat *website* untuk perusahaan, mulai dari desain *User Interface* (UI) dan *User Experience* (UX), membuat kode program untuk sisi *frontend* dan *backend* melakukan *testing* dan *maintenance*.

Web *developer* merupakan pekerjaan yang erat kaitannya dengan pembuatan aplikasi berbasis *website* yang akan diluncurkan kedalam internet yang nanti dapat diakses melalui peramban (Adani, 2020). Sebelum membuat *website* praktikan melakukan tahap perencanaan serta menganalisa bersama pembimbing kerja guna menentukan tujuan dari pembuatan *website* yang akan dibangun. Setelah tahap tersebut praktikan membuat sebuah *mockup* atau desain awal dari aplikasi untuk diberikan kepada pembimbing kerja guna menyesuaikan apakah desain yang dibuat sudah sesuai dengan yang diinginkan. Setelah desain yang dibuat diterima praktikan melakukan pengembangan serta melakukan *testing* terhadap aplikasi yang dibangun apakah sudah berjalan dengan baik atau tidak. Setelah semua tahap sebelumnya selesai dilakukan implementasi aplikasi yang telah dibangun ke dalam *hosting* agar aplikasi dapat diakses melakukan peramban.

### 3.2. Pelaksanaan Kerja

Pada pelaksanaan kerja profesi di CV. Permata Cileungsi Jaya sebagai web *developer*. Praktikan memiliki tugas yaitu membangun sebuah aplikasi berbasis *website* dari tahap perencanaan, perancangan, implementasi, hingga tahap aplikasi dapat digunakan.

#### 3.2.1. Perencanaan

##### 3.2.1.1. *Timeline* Pelaksanaan Kerja Profesi

Pada awal pelaksanaan kerja profesi, tahap pertama yang dilakukan praktikan yaitu mencari segala kebutuhan informasi yang akan menjadi luaran kerja profesi saat ini. Untuk menunjang pelaksanaan kerja profesi dibuatkan sebuah tabel yang berisikan *timeline* alur pelaksanaan yang dilakukan oleh praktikan. *Timeline* tersebut dapat dilihat pada tabel berikut.

Tabel 3.1. *Timeline* pelaksanaan kerja profesi

Kegiatan	Juni		Juli				Agustus				
	3	4	1	2	3	4	1	2	3	4	5
Pembuatan laporan KP	■	■									■
Desain <i>mockup website</i>			■								
Perancangan pembuatan <i>website</i>				■	■	■	■	■	■	■	■
Perancangan struktur <i>database</i>						■	■	■	■	■	■
Finalisasi dan <i>testing</i>											■

Tujuan dibuatnya *timeline* di atas bertujuan guna membantu praktikan dalam mengelola waktu agar teratur. Dengan begitu praktikan tahu kapan harus maju ke tahap – tahap dalam pelaksanaan kerja profesi.

##### 3.2.1.2. Analisis Kebutuhan Sistem

Setelah pembuatan *timeline* selesai dibuat, tahap selanjutnya yaitu melakukan analisis terhadap kebutuhan – kebutuhan sistem yang akan dibangun. Sistem yang dianalisis adalah sistem yang berisikan informasi tentang segala sesuatu yang berkaitan tentang pembangunan aplikasi

berbasis *website*. Dalam Sistem aplikasi nantinya akan memiliki tiga pengguna yaitu.

1. Admin

Admin sebagai administrator dapat mengelola informasi yang ada pada *website* serta dapat menggunakan *website* untuk melakukan proses seperti.

- a) Admin dapat menambah, mengedit, dan menghapus data *user*.
- b) Admin dapat menambah, mengedit, dan menghapus data berita atau artikel.
- c) Admin dapat melihat pesan masuk yang diberikan oleh pengguna *website*.

2. Dokter

Dokter dapat menggunakan *website* untuk melakukan kegiatan sebagai berikut.

- a) Dokter dapat melihat data pasien khitan serta mencetak data pasien.
- b) Dokter dapat mengubah status pendaftaran pasien khitan.
- c) Dokter dapat melihat data pasien pengecekan Covid-19 serta mencetak data pasien.
- d) Dokter dapat mengubah status pendaftaran pasien pengecekan Covid-19.

3. Pengguna

Pengguna dapat menggunakan *website* untuk melakukan kegiatan sebagai berikut.

- a) Pengguna dapat membuat akun *user*.
- b) Pengguna dapat melakukan pendaftaran khitan.
- c) Pengguna dapat melakukan pendaftaran pengecekan Covid-19.
- d) Pengguna dapat mengirimkan pesan kepada admin *website*.
- e) Pengguna dapat melihat informasi seputar Khinik Khitan Center Permata Cileungsi.

- f) Pengguna dapat membaca informasi kesehatan dalam bentuk berita atau artikel.

#### 3.2.1.3. Analisis Kebutuhan Perangkat Keras

Agar dalam perancangan *website* dapat berjalan dengan baik dan mempunyai kemampuan yang memadai, dibutuhkan perangkat keras yang digunakan dalam perancangan *website* sebagai berikut.

Prosesor	: Intel® Core™ i5 10210U
RAM	: 12 GB DDR4
Layar	: 14.0" (16:9) LED-backlit FHD (1920x1080) Anti-Glare 60Hz Panel with 45% NTSC
Kartu Grafis	: NVIDIA® GeForce® MX250 , with 2GB GDDR5 VRAM
Penyimpanan	: Hard drive 1TB 5400 rpm SATA HDD dan Solid-state drive 512GB NVMe PCIe® SSD

#### 3.2.1.4. Analisis Kebutuhan Perangkat Lunak

Selain perangkat keras yang digunakan, dibutuhkan perangkat lunak dalam membantu perancangan *website*. Perangkat lunak yang digunakan sebagai berikut.

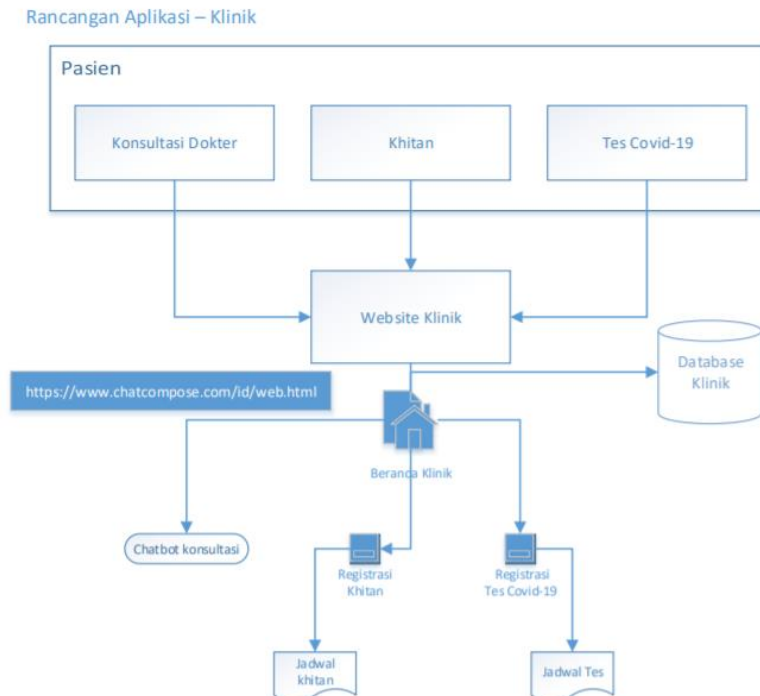
Sistem Operasi	: Windows 10 Home Single Language
Text Editor	: Visual Studio Code
Bahasa Pemrograman	: PHP 7.4
Database	: MySQL 5.7.33

### 3.2.2. Perancangan

#### 3.2.2.1. Navigasi Aplikasi

Setelah pembuatan *timeline* selesai dibuat, tahap selanjutnya yaitu membuat perencanaan dari aplikasi yang akan dibangun. Perancangan dilakukan dengan berdiskusi dengan pihak Klinik Khitan Center Permata Cileungsi mengenai tujuan, batasan, serta fitur – fitur apa saja yang akan terdapat dalam aplikasi. Setelah diskusi selesai dilakukan maka dihasilkan

rancangan navigasi aplikasi yang akan dibangun. Untuk lebih jelasnya dapat dilihat pada gambar 3.1 di bawah ini.



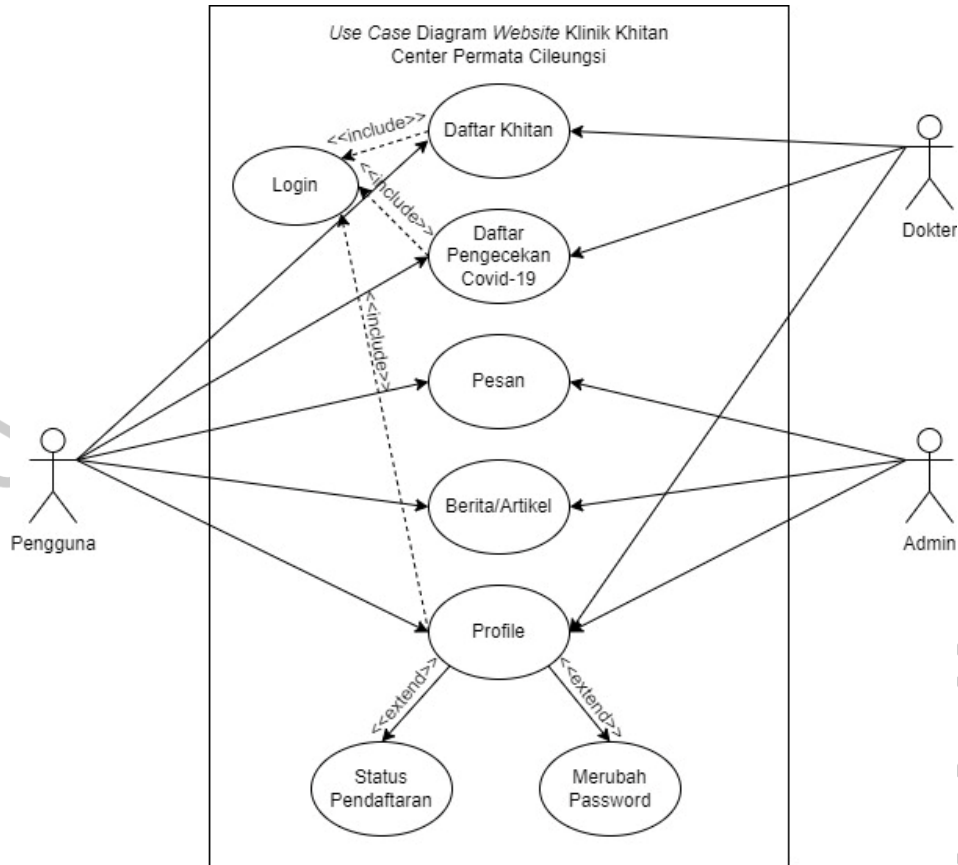
**Gambar 3.1. Rancangan Navigasi Aplikasi**

Berdasarkan gambar di atas dapat dilihat bahwa nantinya aplikasi akan memiliki fitur untuk pendaftaran khitan serta pengecekan covid-19 yang akan menjadi target kerja profesi praktikan. Dalam hal ini nantinya pasien dapat mendaftar secara langsung melalui *website* baik untuk khitan maupun pengecekan covid-19, setelah mendaftar pasien akan langsung mendapatkan notifikasi mengenai jadwal kedatangan pasien ke Klinik Khitan Center Permata Cileungsi untuk dilakukan Tindakan sesuai kriteri pendaftaran pasien sebelumnya melalui *website*.

### 3.2.2.2. Use Case Diagram

*Use case* merupakan sebuah proses pemetaan yang dilakukan untuk menunjukkan hubungan antara pengguna dan sistem yang sedang dirancang. Menurut (Hasugian & Shidiq, 2012) *use case* merupakan diagram yang menjelaskan persyaratan sistem dari sudut pandang pengguna dan menunjukkan pola operasi sistem, ini digunakan untuk

menggambarkan hubungan antara sistem internal dan sistem eksternal, atau antara sistem dan aktor. Dalam pembangunan aplikasi ini dibuat sebuah *use case* yang dapat dilihat pada gambar di bawah ini.

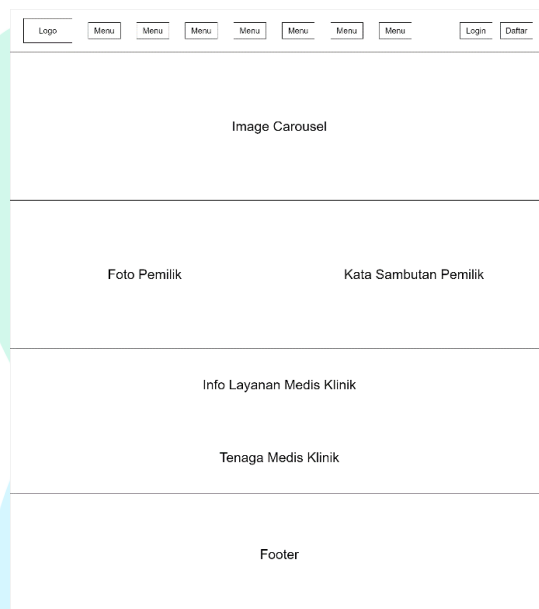


Gambar 3.2. Use Case Diagram Aplikasi

Menurut pada *use case* di atas pengguna dapat melakukan proses daftar khitan, daftar pengecekan Covid-19, dan melihat profil yang diharuskan untuk *login* terlebih dahulu sebelum menggunakan fitur tersebut. Namun pengguna tetap dapat menggunakan *website* tanpa *login* jika hanya ingin mengirimkan pesan kepada admin maupun membaca informasi terkait Klinik Khitan Center Permata Cileungsi maupun artikel mengenai kesehatan. Kemudian dokter dapat melihat data profil pasien dan Riwayat pendaftaran pada akun mereka. Terakhir admin dapat mengelola data pesan, berita atau artikel, dan profil akun setiap pengguna *website*.

### 3.2.2.3. *Mock Up*

Kemudian setelah praktikan mengetahui navigasi apa saja yang akan dilakukan aplikasi, praktikan membuat desain *mockup* atau gambaran rancangan antarmuka untuk aplikasi yang akan dibangun. *Mockup* berguna untuk memberikan gambaran kasar kepada klien mengenai bagaimana bentuk tampilan dari *website*. Selain itu, *mockup* juga berguna apabila terdapat kekurangan dalam desain tampilan praktikan dapat mengubah desain kembali sehingga tidak sulit apabila belum diterapkan ke dalam bahasa pemrograman. Untuk lebih jelasnya dapat dilihat pada desain *mockup* yang telah dibuat di bawah ini.



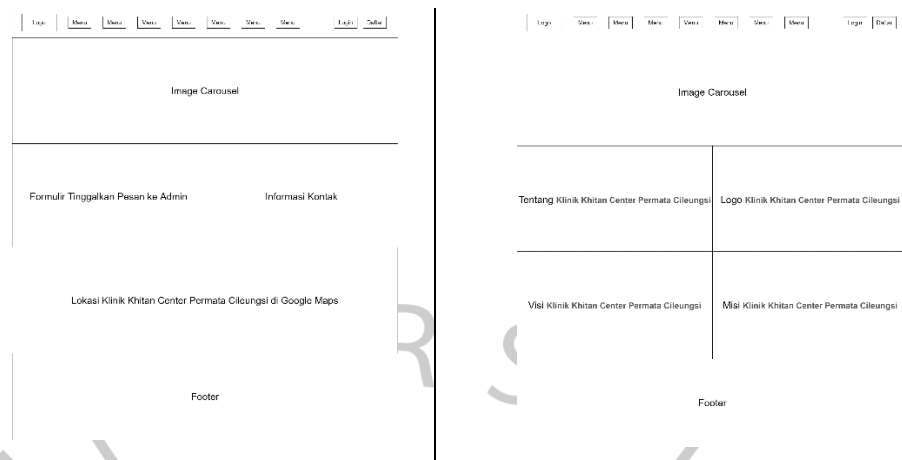
**Gambar 3.3. *Mockup* Halaman Depan Aplikasi**

**Gambar 3.4. Mockup Login dan Register Akun Aplikasi**

**Gambar 3.5. Mockup Pendaftaran Khitan dan Pengecekan Covid-19 Aplikasi**

**Gambar 3.6. Mockup List dan Baca Berita/Artikel Aplikasi**



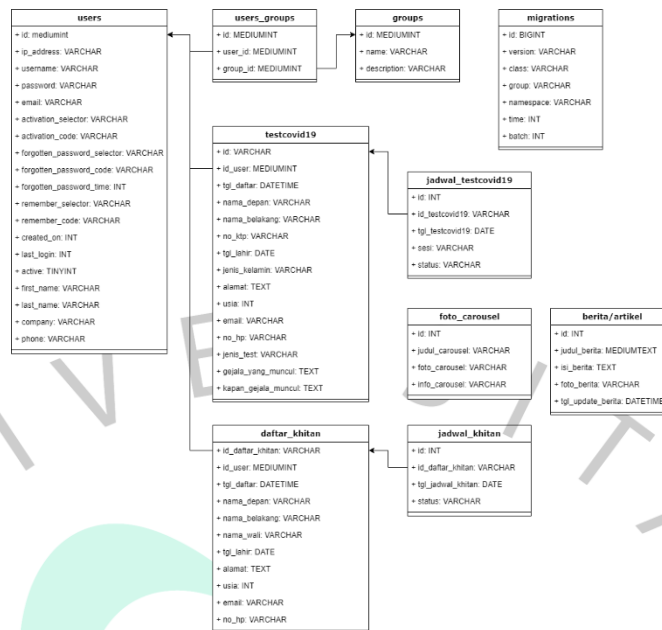


**Gambar 3.7. Mockup Tentang dan Kontak Kami Aplikasi**

Setelah desain *mockup* selesai dibentuk maka praktikan memberikan desain tersebut kepada pihak Klinik Khitan Center Permata Cileungsi untuk mengetahui apakah desain sudah sesuai dengan yang diinginkan. Setelah praktikan mendapatkan umpan balik mengenai desain *mockup* dan telah disetujui maka praktikan dapat melanjutkan proses ke tahap implementasi. Dimana implementasi ini bertujuan untuk menerapkan desain *mockup* yang ada menjadi kumpulan kode program.

#### 3.2.2.4. Database

*Database* merupakan sebuah tempat penyimpanan sekumpulan data yang dapat ditambah, diubah, maupun dihapus. Menurut (Andaru, 2018) Database merupakan kumpulan dari data yang saling berhubungan atau berelasi antara yang satu dengan yang lainnya yang diorganisasikan berdasarkan skema atau struktur tertentu. Tujuan dari dibuatnya struktur database yaitu untuk memberikan gambaran kepada praktikan dalam membangun *website* bagaimana data – data yang ada nantinya saling berhubungan. Untuk lebih jelasnya dapat melihat pada gambar di bawah ini mengenai struktur *database* yang dibuat.



Gambar 3.8. Struktur database website

### 3.2.3. Implementasi

#### 3.2.3.1. Pembuatan Frontend

Setelah desain *mockup* disetujui maka tahap selanjutnya yaitu proses pembuatan kode program dari setiap *mockup* yang ada. Proses ini biasa disebut dengan pembuatan *frontend*, dalam pembuatan *frontend* praktikan menggunakan Bahasa pemrograman PHP, CSS, dan Javascript agar dapat dibuka dengan peramban. Selain itu praktikan juga menggunakan framework dalam pembuatan *website* ini, framework yang digunakan yaitu CodeIgniter 4 dan Bootstrap 5.

##### 1. CodeIgniter 4

CodeIgniter 4 merupakan sebuah kerangka dalam pembuatan aplikasi berbasis *website* yang menggunakan PHP sebagai bahasa pemrograman yang bertujuan agar memungkinkan aplikasi dibangun lebih cepat (Igniter, 2021). CodeIgniter memiliki banyak fitur yang dapat membantu praktikan dalam membangun *website*. Selain itu dibandingkan dengan *framework* lain CodeIgniter memiliki ukuran *file* yang lebih kecil dan dokumentasi maupun tutorial yang mudah dipahami oleh praktikan dalam mencari solusi dari suatu masalah.

## 2. IonAuth

IonAuth merupakan *library* yang dapat digunakan sebagai otentikasi pada *framework* CodeIgniter. Otentikasi sendiri merupakan proses dimana pengguna diharuskan melakukan dua cara identifikasi untuk dapat mengakses akunnya. Dua cara yang dimaksud biasanya memasukkan *email* dan *password* yang sesuai untuk dapat diverifikasi bahwa memang pengguna tersebut yang memiliki akun. Fitur yang diberikan oleh IonAuth sendiri terbilang cukup lengkap yaitu adanya fitur *login*, *register*, *forget password* dan *change password*.

## 3. PHP

PHP merupakan salah satu Bahasa pemrograman berbasis *web* yang banyak digunakan saat ini, selain itu PHP dapat digunakan dengan bebas dan gratis sehingga mudah untuk dapat dipelajari oleh siapapun (Solichin, 2016). Bahasa ini digunakan oleh praktikan sebagai bahasa utama dalam pembangunan *website* karena banyak kelebihanannya salah satunya dapat membuat Halaman *website* menjadi dinamis. *Website* dinamis memiliki arti yaitu pengguna (admin) dapat menambah, memodifikasi, ataupun menghapus konten yang ditampilkan pada *website*.

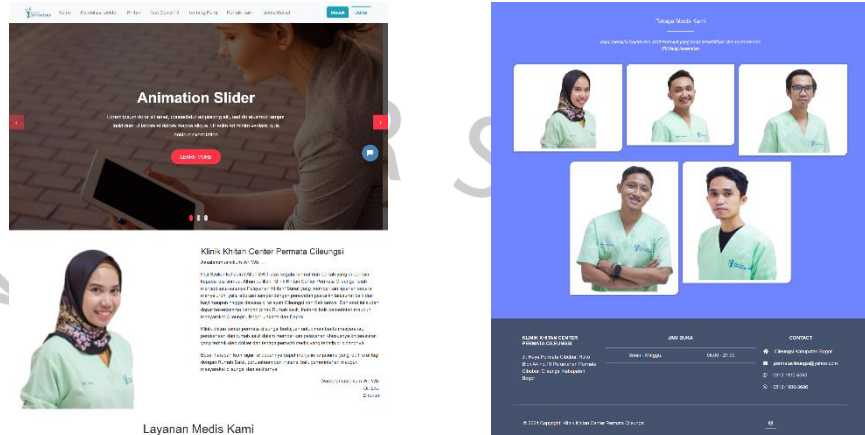
## 4. AdminLTE

AdminLTE merupakan salah satu *template* yang biasa digunakan dalam pembuatan *website* sebagai halaman antarmuka untuk tampilan dengan hak akses admin dan dokter. Sehingga AdminLTE digunakan sebagai halaman *dashboard* yang hanya dapat diakses oleh admin dan dokter.

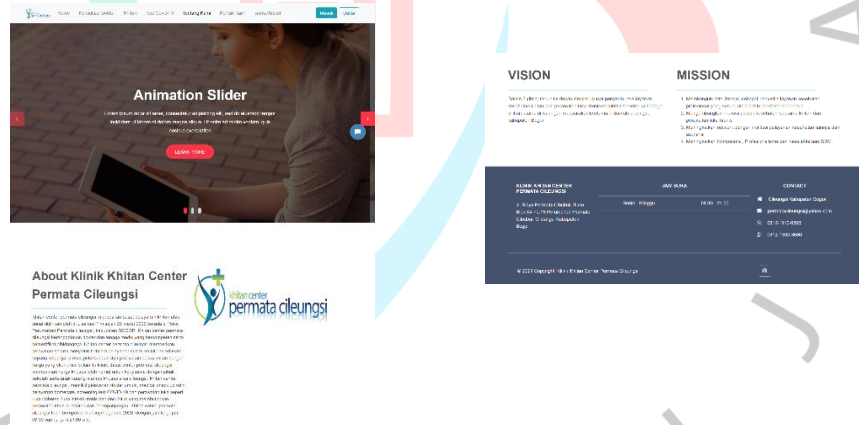
## 5. Bootstrap 5

Bootstrap merupakan *framework* CSS yang dapat digunakan secara bebas dan gratis dalam merancang tampilan *website*. Selain itu Bootstrap juga *responsive* terhadap banyak *platform*, artinya tampilan halaman *website* yang menggunakan Bootstrap ini akan tampak tetap rapi, baik versi *mobile* maupun *desktop* (Zakir, 2016). Praktikan menggunakan Bootstrap sebagai kerangka dalam pembuatan *style* saat membangun halaman *website*.

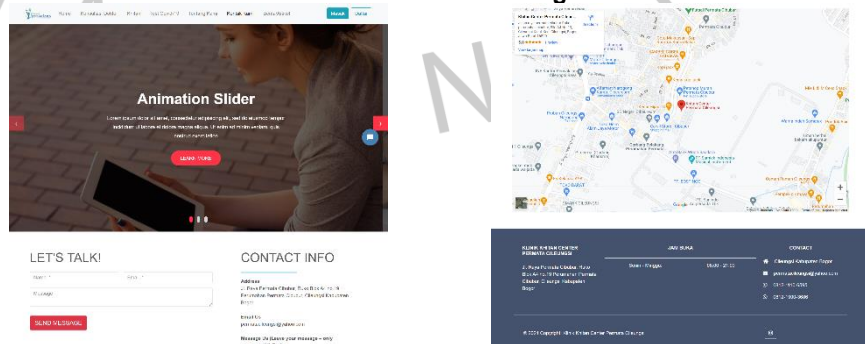
Setelah semua alat (*tools*) yang akan digunakan sudah diketahui. Selanjutnya praktikan mulai menggunakan alat – alat tersebut untuk membangun *website*. Berikut ditampilkan hasil pembuatan sisi *frontend* yang telah selesai dibangun.



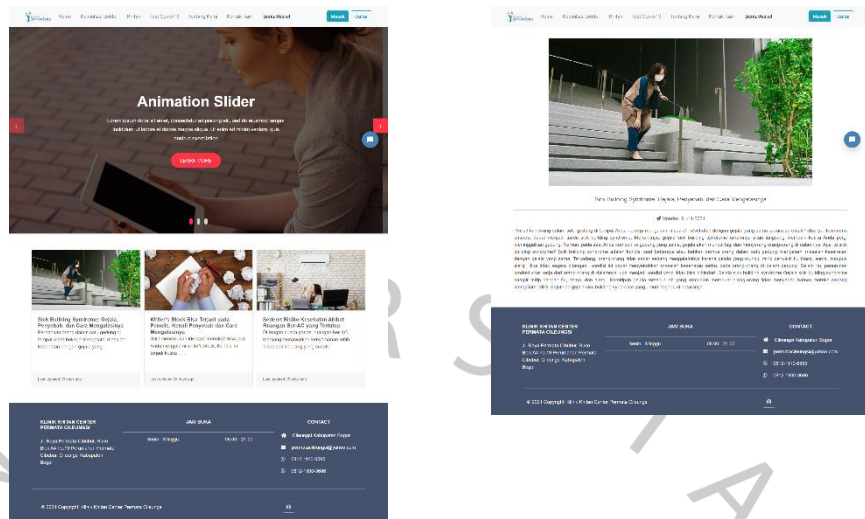
Gambar 3.9. Antarmuka halaman depan *website*



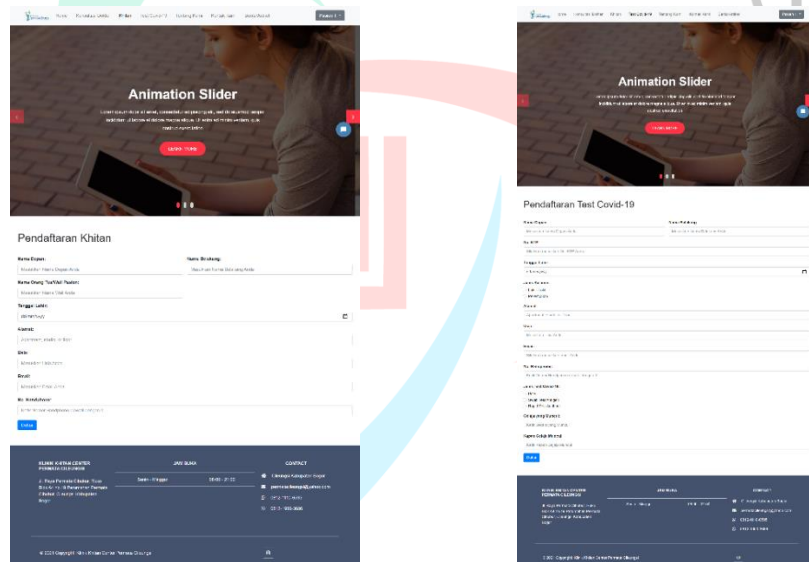
Gambar 3.10. Antarmuka halaman tentang kami *website*



Gambar 3.11. Antarmuka halaman kontak kami *website*



Gambar 3.12. Antarmuka halaman *list* berita dan baca berita *website*



Gambar 3.13. Antarmuka halaman daftar khitan dan pengecekan covid-19

### 3.2.3.2. Pembuatan *Backend*

Setelah pembuatan sisi *frontend* selesai dibangun, selanjutnya yaitu membuat sisi *backend* agar aplikasi dapat bekerja sesuai dengan tujuannya. *Backend* sendiri merupakan bahasa pemrograman yang bekerja pada sisi server. Menurut (Arhandi, 2016) *Backend* adalah suatu tempat dimana proses dari sebuah aplikasi atau sistem yang berjalan biasanya menambahkan, mengubah, atau menghapus data. Dalam

pembuatan *website* ini praktikan juga menggunakan bahasa PHP sebagai bahasa pemrograman di sisi *backend*.

```
// check to see if the user is logging in
// check for "remember me"
$remember = (bool)$this->request->getVar('remember');

if ($this->ionAuth->login($this->request->getVar('identity'), $this->request->getVar('password'), $remember)) {
    //if the login is successful
    //redirect them back to the home page
    $this->session->setFlashdata('message', $this->ionAuth->messages());
    return redirect()->to('index')->withCookies();
} else {
    // if the login was un-successful
    // redirect them back to the login page
    $this->session->setFlashdata('message', $this->ionAuth->errors($this->validationListTemplate));
    // use redirects instead of loading views for compatibility with MY_Controller libraries
    return redirect()->back()->withInput();
}
```

Gambar 3.14. *Backend login*

Pada gambar 3.14 di atas merupakan proses dalam aplikasi saat pengguna ingin melakukan *login* kedalam *website*. Program akan mengecek terlebih dahulu apakah data *remember me* terdapat dalam cache browser, apabila terdapat maka *system* akan langsung mengarahkan pengguna untuk ke halaman setelah *login* sukses. Namun jika tidak pengguna akan diminta untuk mengisi data berupa *email* dan *password* yang kemudian akan dicocokkan datanya dengan yang terdapat dalam *database*.

```
public function logout(): bool
{
    $this->ionAuthModel->triggerEvents('logout');

    $identity = $this->session->get('user_id');

    $this->session->remove([$identity, 'id', 'user_id']);

    // delete the remember me cookies if they exist
    delete_cookie($this->config->rememberCookieName);

    // Clear all codes
    if (isset($identity)) {
        $this->ionAuthModel->clearForgottenPasswordCode($identity);
        $this->ionAuthModel->clearRememberCode($identity);
    }

    // Destroy the session
    $this->session->destroy();

    // Recreate the session
    session_start();

    session_regenerate_id(true);

    $this->setMessage('IonAuth.logout_successful');
    return true;
}
```

Gambar 3.15. *Backend logout*



Pada gambar 3.15 di atas merupakan proses pembuatan sebuah *function* dimana *function* tersebut bertujuan untuk membuat fitur *logout*. Fitur tersebut berguna apabila pengguna ingin mengakhiri sesi *login* yang sebelumnya telah ia lakukan guna mengurangi pengguna akses akun oleh orang lain. Dalam *function* tersebut seluruh sesi *email* dan *password* beserta *cookies* yang tersimpan akan dihapus dari peramban.

```

$identity = $this->session->get('identity');

$change = $this->ionAuth->changePassword($identity, $this->request->getPost('old'), $this->request->getPost('new'));

if ($change) {
    //if the password was successfully changed
    $this->session->setFlashdata('message', $this->ionAuth->messages());
    return $this->logout();
} else {
    $this->session->setFlashdata('message', $this->ionAuth->errors($this->validationListTemplate));
    return redirect()->to('/auth/change_password');
}

```

Gambar 3.16. Backend change password

Pada gambar 3.16 di atas merupakan proses saat pengguna ingin merubah *password* akun sebelumnya. Apabila *password* lama sesuai dan *password* baru sesuai dengan kriteria maka *password* akan terganti.

```

$identityColumn = $this->configIonAuth->identity;
$identity = $this->ionAuth->where($identityColumn, $this->request->getPost('identity'))->users()->row();

if (empty($identity)) {
    if ($this->configIonAuth->identity !== 'email') {
        $this->ionAuth->setError('Auth.forgot_password_identity_not_found');
    } else {
        $this->ionAuth->setError('Auth.forgot_password_email_not_found');
    }

    $this->session->setFlashdata('message', $this->ionAuth->errors($this->validationListTemplate));
    return redirect()->to('/auth/forgot_password');
}

// run the forgotten password method to email an activation code to the user
$forgotten = $this->ionAuth->forgotPassword($identity->{$this->configIonAuth->identity});

if ($forgotten) {
    // if there were no errors
    $this->session->setFlashdata('message', $this->ionAuth->messages());
    return redirect()->to('/auth/login'); //we should display a confirmation page here instead of the login page
} else {
    $this->session->setFlashdata('message', $this->ionAuth->errors($this->validationListTemplate));
    return redirect()->to('/auth/forgot_password');
}

```

Gambar 3.17. Backend forgot password

Pada gambar 3.17 di atas merupakan proses saat pengguna ingin melakukan fitur lupa *password*. Pengguna akan diminta memasukkan *email* kemudian apabila email tersebut cocok, sistem akan mengirimkan *email* kepada pengguna mengenai *link* untuk merubah *password*.

```

$identity = $user->{$this->configIonAuth->identity};

// do we have a valid request?
if ($user->id != $this->request->getPost('user_id')) {
    // something fishy might be up
    $this->ionAuth->clearForgottenPasswordCode($identity);

    throw new \Exception(lang('Auth.error_security'));
} else {
    // finally change the password
    $change = $this->ionAuth->resetPassword($identity, $this->request->getPost('new'));

    if ($change) {
        // if the password was successfully changed
        $this->session->setFlashdata('message', $this->ionAuth->messages());
        return redirect()->to('/auth/login');
    } else {
        $this->session->setFlashdata('message', $this->ionAuth->errors($this->validationListTemplate));
        return redirect()->to('/auth/reset_password/' . $code);
    }
}
} else {
    // if the code is invalid then send them back to the forgot password page
    $this->session->setFlashdata('message', $this->ionAuth->errors($this->validationListTemplate));
    return redirect()->to('/auth/forgot_password');
}

```

Gambar 3.18. Backend reset password

Pada gambar 3.18 di atas merupakan proses saat pengguna telah mendapatkan *link* untuk mereset *password*. Sistem akan mengecek *link* yang juga berisi kode aktivasi apakah *link* tersebut masih aktif atau tidak, jika masih pengguna dapat mereset *password*, jika tidak pengguna akan diminta untuk mengulangi proses lupa *password*.

```

function random_strings($length_of_string)
{
    // String of all alphanumeric character
    $str_result = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';

    // Shuffle the $str_result and returns substring
    // of specified length
    return substr(
        str_shuffle($str_result),
        0,
        $length_of_string
    );
}

```

Gambar 3.19. Backend generate string

Pada gambar 3.19 di atas merupakan proses pembuatan sebuah *function* dimana *function* tersebut bertujuan untuk menghasilkan sebuah *string* yang panjang karakternya dapat ditentukan saat *function* tersebut dipanggil. *Function* tersebut berguna untuk menghasilkan id unik dari setiap data pendaftaran khitan maupun pengecekan Covid-19.



```

public function jadwal_khitan($id_khitan)
{
    $hari = 1;
    $tgl_daftar = date('Y-m-d');
    $hitung = $this->db->table('jadwal_khitan')
    ->where('tgl_jadwal_khitan', date('Y-m-d', strtotime('+'. $hari.' days', strtotime($tgl_daftar))))
    ->countAllResults();
    if ($hitung == 0) {
        $data = [
            'id_daftar_khitan' => $id_khitan,
            'tgl_jadwal_khitan' => date('Y-m-d', strtotime('+'. $hari.' days', strtotime($tgl_daftar))),
            'status' => 'Menunggu',
        ];
        $this->db->table('jadwal_khitan')
        ->insert($data);
    }
    else {
        do {
            $hari++;
            $hitung = $this->db->table('jadwal_khitan')
            ->where('tgl_jadwal_khitan', date('Y-m-d', strtotime('+'. $hari.' days', strtotime($tgl_daftar))))
            ->countAllResults();
        }
        while ($hitung > 0);
        $data = [
            'id_daftar_khitan' => $id_khitan,
            'tgl_jadwal_khitan' => date('Y-m-d', strtotime('+'. $hari.' days', strtotime($tgl_daftar))),
            'status' => 'Menunggu',
        ];
        $this->db->table('jadwal_khitan')
        ->insert($data);
    }
}

```

Gambar 3.20. Backend pendaftaran khitan

Pada gambar 3.20 di atas merupakan proses pembuatan sebuah *function* dimana *function* tersebut bertujuan untuk menghasilkan jadwal yang tersedia untuk proses pendaftaran khitan. Prosesnya yaitu dengan membaca tanggal saat pasien daftar kemudian tanggal tersebut ditambah satu apabila di keesokan hari jadwal masih kosong maka pasien akan mendapatkan jadwal di hari tersebut. Namun, apabila hari tersebut telah terisi sistem akan menambah satu hari lagi terus menerus hingga jadwal kosong didapatkan untuk pasien. Setelah itu pasien akan mendapatkan informasi mengenai jadwal melalui halaman *website* dan melalui *email* yang dimasukkan pasien.

```

public function jadwal_testcovid19($id_test_covid19)
{
    $shari = 1;
    $tgl_daftar = date('Y-m-d');

    $hitung = $this->db->table('jadwal_testcovid19')
    ->where('tgl_testcovid19', date('Y-m-d', strtotime('+'. $shari.' days', strtotime($tgl_daftar))))
    ->countAllResults();

    if ($hitung == 0) {
        $data = [
            'id_testcovid19' => $id_test_covid19,
            'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $shari.' days', strtotime($tgl_daftar))),
            'sesi' => 'Sesi 1 (10:00 - 12:00)',
            'status' => 'Menunggu',
        ];
        $this->db->table('jadwal_testcovid19')
        ->insert($data);
    }
    else if ($hitung == 1) {
        $data = [
            'id_testcovid19' => $id_test_covid19,
            'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $shari.' days', strtotime($tgl_daftar))),
            'sesi' => 'Sesi 2 (14:00 - 16:00)',
            'status' => 'Menunggu',
        ];
        $this->db->table('jadwal_testcovid19')
        ->insert($data);
    }
    else {
        do {
            $shari++;
        }
        while ($hitung > 1);

        $hitung = $this->db->table('jadwal_testcovid19')
        ->where('tgl_testcovid19', date('Y-m-d', strtotime('+'. $shari.' days', strtotime($tgl_daftar))))
        ->countAllResults();

        if ($hitung == 0) {
            $data = [
                'id_testcovid19' => $id_test_covid19,
                'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $shari.' days', strtotime($tgl_daftar))),
                'sesi' => 'Sesi 1 (10:00 - 12:00)',
                'status' => 'Menunggu',
            ];
            $this->db->table('jadwal_testcovid19')
            ->insert($data);
        }
        else if ($hitung == 1) {
            $data = [
                'id_testcovid19' => $id_test_covid19,
                'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $shari.' days', strtotime($tgl_daftar))),
                'sesi' => 'Sesi 2 (14:00 - 16:00)',
                'status' => 'Menunggu',
            ];
            $this->db->table('jadwal_testcovid19')
            ->insert($data);
        }
    }
}

```

Gambar 3.21. Backend pendaftaran pengecekan Covid-19

Pada gambar 3.21 di atas merupakan proses pembuatan sebuah *function* dimana *function* tersebut bertujuan untuk menghasilkan jadwal yang tersedia untuk proses pendaftaran pengecekan covid-19. Prosesnya yaitu dengan membaca tanggal saat pasien daftar kemudian tanggal tersebut ditambah satu apabila di keesokan hari pada sesi pagi terdapat jadwal kosong maka pasien akan mendapatkan sesi pagi. Namun, apabila sesi pagi telah terisi maka sistem akan mengecek sesi siang jika kosong akan mendapatkan sesi tersebut. Jika, pada hari tersebut sudah penuh semua sesi sistem akan menambah satu hari lagi dan melakukan

pengecekan ulang pada semua sesi dan proses ini akan terus berlangsung hingga pasien mendapatkan jadwal. Setelah itu pasien akan mendapatkan informasi mengenai jadwal melalui halaman *website* dan melalui *email* yang dimasukkan pasien.

```
public function store_add_berita()
{
    $model = new BeritaModel();
    $periksa = $this->validate([
        'image' => [
            'uploaded[image]',
            'mime_in[image,image/jpeg,image/png,image/jpg]',
            'max_size[image,4096]',
        ],
    ]);

    if ($periksa) {
        $image = $this->request->getFile('image'); //ambil data dari file upload
        $image->move($_SERVER['DOCUMENT_ROOT'] . '/img/berita'); //pindahkan file upload ke folder img
        $nama_gambar = $image->getName(); //ambil nama file
        //dd($image);
        $data = [
            'judul_berita' => $this->request->getVar('judul_berita'),
            'isi_berita' => $this->request->getVar('isi_berita'),
            'foto_berita' => $nama_gambar,
            'tgl_update_berita' => date('Y-m-d H:i:s')
        ];
        $save = $model->insert($data);

        $this->session->setFlashdata('message', 'Berhasil Menambah Berita');
        return redirect()->to('/auth/list_berita');
    } else {
        $this->session->setFlashdata('message', 'The file you entered is not an image or has a size above 4MB');
        return redirect()->to(base_url('auth/list_berita'));
    }
}
```

Gambar 3.22. Backend tambah berita

Pada gambar 3.22 di atas merupakan proses pembuatan sebuah *function* dimana *function* tersebut bertujuan untuk menyimpan data berita yang akan ditambahkan oleh admin. Dalam berita terdapat foto yang memiliki kriteria harus dalam format .jpeg, .jpg, .png dengan ukuran *file* maksimal 4MB. Konten berita ini nantinya akan dapat dilihat oleh pengguna *website* baik yang telah terdaftar maupun yang belum terdaftar.

```

public function delete_berita($id = null)
{
    if (!$this->ionAuth->loggedIn()) {
        // redirect them to the login page
        return redirect()->to('/auth/login');
    } elseif ($this->ionAuth->inGroup('admin')) {
        $session = session();
        $model = new BeritaModel();
        $gambar = $model->where('id', $id)->first();
        if ($gambar['foto_berita'] != null) {
            unlink($_SERVER['DOCUMENT_ROOT'] . '/img/berita/' . $gambar['foto_berita']);
            $data['list_berita'] = $model->where('id', $id)->delete();

            $session->setFlashdata('message', 'List Berita deleted successfully');
            return redirect()->to(base_url('auth/list_berita'));
        } else {
            $data['list_berita'] = $model->where('id', $id)->delete();

            $session->setFlashdata('message', 'List Berita deleted successfully');
            return redirect()->to(base_url('auth/list_berita'));
        }
    } else {
        echo "Anda Tidak Memiliki Hak Akses ke Modul ini <a href='" . base_url('auth/logout') . "'>Logout</a>";
    }
}

```

Gambar 3.23. *Backend* hapus berita

Pada gambar 3.23 di atas merupakan proses pembuatan sebuah *function* dimana *function* tersebut bertujuan untuk menghapus data berita yang telah disimpan sebelumnya. Data tersebut berita konten artikel berita serta foto yang disimpan akan terhapus agar tidak penuh pada sisi penyimpanan.

```

public function status_pasien_khitan($id = null, $status = null)
{
    $model = new JadwalKhitanModel();
    $save = $model->update($id, ['status' => $status]);
    $this->session->setFlashdata('message', 'Anda berhasil membuat perubahan');
    return redirect()->to('/dokter/data_pasien_khitan');
}

```

Gambar 3.24. *Backend* ubah status pasien khitan

Pada gambar 3.24 di atas merupakan proses perubahan data status pendaftaran khitan yang dilakukan oleh dokter. Perubahan tersebut bisa saja status pasien menjadi dibatalkan atau selesai.

```

public function data_pasien_covid($startdate = null, $enddate = null)
{
    if (!isset($startdate)) {
        $startdate = '1970-01-01';
        $enddate = date('Y-m-d');
    }
    $user = $this->ionAuth->user()->row();
    $data = [
        'firstname' => $user->first_name,
        'lastname' => $user->last_name,
        'totalpasienCovid' => $this->JadwalCovidModel->totalpasienCovidhariini(),
        'datap pasien' => $this->JadwalCovidModel->rangedatap pasien($startdate, $enddate),
    ];
    return $this->renderPage($this->viewsFolder . DIRECTORY_SEPARATOR . 'dokter/data_pasien_covid', $data);
}

```

Gambar 3.25. *Backend* ubah status pasien pengecekan covid-19

Pada gambar 3.25 di atas merupakan *function* perubahan data status pendaftaran pengecekan covid-19 yang dilakukan oleh dokter. Perubahan tersebut layaknya seperti perubahan status pada pendaftaran khitan bisa saja status pasien menjadi dibatalkan atau selesai.

```
public function data_khitan($id_user)
{
    return $this->db->table('jadwal_khitan')
->join('daftar_khitan', 'daftar_khitan.id_daftar_khitan=jadwal_khitan.id_daftar_khitan')
->where('id_user', $id_user)
->orderBy('tgl_daftar', 'ASC')
->get()->getResultArray();
}
```

Gambar 3.26. *Backend* melihat daftar jadwal khitan pasien

Pada gambar 3.26 di atas merupakan *function* untuk menampilkan data pendaftaran khitan yang telah dilakukan oleh pasien. Data yang terlihat hanya terbatas pada *user* akun pasien yang login saja.

```
public function data_testcovid19($id_user)
{
    return $this->db->table('jadwal_testcovid19')
->join('testcovid19', 'testcovid19.id=jadwal_testcovid19.id_testcovid19')
->where('id_user', $id_user)
->orderBy('tgl_daftar', 'ASC')
->get()->getResultArray();
}
```

Gambar 3.27. *Backend* melihat daftar jadwal pengecekan covid-19 pasien

Pada gambar 3.27 di atas merupakan *function* untuk menampilkan data pendaftaran pengecekan covid-19 yang telah dilakukan oleh pasien. Data yang terlihat hanya terbatas pada *user* akun pasien yang login saja.

```
public function rangedatapasien($startdate, $enddate)
{
    return $this->db->table('jadwal_testcovid19')
->join('testcovid19', 'jadwal_testcovid19.id_testcovid19=testcovid19.id')
->where('tgl_testcovid19 >=', $startdate)
->where('tgl_testcovid19 <=', $enddate)
->get()->getResultArray();
}
```

Gambar 3.28. *Backend* pengecekan data pasien covid-19 berdasarkan tanggal

Pada gambar 3.28 di atas merupakan *function* untuk melihat data pasien pengecekan covid-19 sesuai dengan rentang awal tanggal dan akhir tanggal yang dimasukkan dokter.

```

public function rangedatap pasien($startdate, $enddate)
{
    return $this->db->table('jadwal_khitan')
        ->join('daftar_khitan', 'jadwal_khitan.id_daftar_khitan=daftar_khitan.id_daftar_khitan')
        ->where('tgl_jadwal_khitan >=', $startdate)
        ->where('tgl_jadwal_khitan <=', $enddate)
        ->get()->getResultArray();
}

```

Gambar 3.29. Backend melihat data pasien khitan berdasarkan tanggal

Pada gambar 3.29 di atas merupakan *function* untuk melihat data pasien khitan sesuai dengan rentang awal tanggal dan akhir tanggal yang dimasukkan dokter.

### 3.2.3.3. Perancangan Pengujian

Adapun rancangan pengujian yang akan dilakukan menggunakan metode pengujian *white box* dan *black box* yang akan praktikan kelompokkan pada table di bawah ini.

#### 1. White Box

Pengujian *white box* merupakan pengujian yang dilakukan dengan menguji aplikasi dengan cara menganalisis baris program yang telah dibuat apakah sudah sesuai dengan fungsi dari baris program tersebut dan dapat berjalan atau tidak. Pada tahap ini praktikan membuat suatu tabel perancangan pengujian *white box* untuk menentukan komponen serta kode program yang akan diuji pada tahap pengujian nantinya. Untuk lebih jelasnya dapat dilihat pada tabel di bawah ini.

Tabel 3.2. Perancangan pengujian *white box*.

No.	Komponen yang diuji	Kode Program
1	Login	<pre> if(\$this-&gt;request-&gt;getPost() &amp;&amp; \$this-&gt;validation -&gt;withRequest(\$this-&gt;request)-&gt;run()) {     \$remember = (bool)\$this-&gt;request -&gt;getVar('remember'); if(\$this-&gt;ionAuth-&gt;login(\$this -&gt;request-&gt;getVar('identity'), \$this-&gt;request -&gt;getVar('password'), \$remember)) { \$this-&gt;session -&gt;setFlashdata('message', \$this-&gt;ionAuth -&gt;messages()); return redirect()-&gt;to('index') -&gt;withCookies(); } else { \$this-&gt;session </pre>

		<pre>-&gt;setFlashdata('message', \$this-&gt;ionAuth -&gt;errors(\$this-&gt;validationListTemplate)); return redirect()-&gt;back()-&gt;withInput();}</pre>
		<p>Hasil yang diharapkan</p> <p>Apabila pengguna memasukkan <i>email</i> dan <i>password</i> yang benar maka pengguna akan diarahkan ke halaman setelah <i>login</i>. Jika salah satu dari identitas yang dimasukkan salah maka pengguna akan diminta untuk <i>login</i> kembali.</p>
		<p>Hasil</p>
2	Logout	<pre>\$this-&gt;ionAuth-&gt;logout(); \$this-&gt;session-&gt;setFlashdata('message', \$this -&gt;ionAuth-&gt;messages()); return redirect()-&gt;to('/auth/login')-&gt;withCookies();</pre>
		<p>Hasil yang diharapkan</p> <p>Pengguna dapat <i>logout</i> dari akun yang sedang <i>login</i> saat ini.</p>
		<p>Hasil</p>
3	Ubah Kata Sandi	<pre>\$identity = \$this-&gt;session-&gt;get('identity'); \$change = \$this-&gt;ionAuth-&gt;changePassword (\$identity, \$this-&gt;request-&gt;getPost('old'), \$this -&gt;request-&gt;getPost('new')); if (\$change) { \$this-&gt;session-&gt;setFlashdata('message', \$this -&gt;ionAuth-&gt;messages()); return \$this-&gt;logout(); } else { \$this-&gt;session-&gt;setFlashdata('message', \$this -&gt;ionAuth-&gt;errors(\$this -&gt;validationListTemplate)); return redirect()-&gt;to('/auth/change_password');}</pre>
		<p>Hasil yang diharapkan</p> <p>Pengguna dapat mengganti kata sandi apabila kata sandi lama yang dimasukkan sesuai dengan yang tersimpan pada <i>database</i> dan kata sandi baru sesuai dengan kriteria yang ditentukan.</p>
		<p>Hasil</p>
4	Lupa Kata Sandi	<pre>\$forgotten = \$this-&gt;ionAuth-&gt;forgottenPassword (\$identity-&gt;{\$this-&gt;configIonAuth-&gt;identity});</pre>

---

```

if ($forgotten) { $this->session->setFlashdata
('message', $this->ionAuth->messages());
return redirect()->to('/auth/login'); } else { $this
->session->setFlashdata('message', $this->ionAuth
->errors($this->validationListTemplate));
return redirect()->to('/auth/forgot_password');}

```

---

Hasil yang diharapkan

---

Pengguna dapat mengatur ulang kata sandi apabila pengguna ingin mengganti kata sandi yang lupa.

---

Hasil

---

5	Membuat String acak	<pre> function random_strings(\$length_of_string){ \$str_result = '0123456789ABCDEFGHIJKLMNO PQRSTUVWXYZabcdefghijklmnopqrstu vwxyz'; \$str_result and returns; return substr(str_shuffle (\$str_result),0, \$length_of_string);} </pre>
---	---------------------	--

---

Hasil yang diharapkan

---

Sistem membuat kode unik untuk setiap pendaftaran khitan maupun pengecekan covid-19.

---

Hasil

---

6	Menentukan Jadwal Khitan	<pre> public function jadwal_khitan(\$id_khitan) {     \$hari = 1; \$tgl_daftar = date('Y-m-d');     \$hitung = \$this-&gt;db-&gt;table('jadwal_khitan') -&gt;where('tgl_jadwal_khitan', date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar))) -&gt;countAllResults(); if (\$hitung == 0) { \$data = [ 'id_daftar_khitan' =&gt; \$id_khitan, 'tgl_jadwal_khitan' =&gt; date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar))), 'status' =&gt; 'Menunggu']; \$this-&gt;db-&gt;table('jadwal_khitan') -&gt;insert(\$data);} else { do {     \$hari++; \$hitung = \$this-&gt;db-&gt;table('jadwal_khitan') -&gt;where('tgl_jadwal_khitan', date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar))) -&gt;countAllResults(); } while (\$hitung &gt; 0); </pre>
---	--------------------------	---

---



---

```

$data = ['id_daftar_khitan' =>
$id_khitan,'tgl_jadwal_khitan' => date('Y-m-d',
strtotime('+.$hari.' days', strtotime($tgl_daftar))),
'status' => 'Menunggu']; $this->db
->table('jadwal_khitan')
->insert($data);}}

```

---

Hasil yang diharapkan

Sistem dapat dengan langsung memberikan jadwal kepada pasien perihal pendaftaran khitan yang dilakukan pasien.

Hasil

7	Menentuka n Jadwal Pengeceka n Covid-19	<pre> public function jadwal_testcovid19(\$id_test_covid19) {     \$hari = 1; \$tgl_daftar = date('Y-m-d'); \$hitung = \$this-&gt;db- &gt;table('jadwal_testcovid19') -&gt;where('tgl_testcovid19', date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar)))) -&gt;countAllResults(); if (\$hitung == 0) { \$data = ['id_testcovid19' =&gt; \$id_test_covid19, 'tgl_testcovid19' =&gt; date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar))), 'sesi' =&gt; 'Sesi 1 (10:00 - 12:00)', 'status' =&gt; 'Menunggu',]; \$this-&gt;db-&gt;table('jadwal_testcovid19') -&gt;insert(\$data); } else if (\$hitung == 1) { \$data = ['id_testcovid19' =&gt; \$id_test_covid19, 'tgl_testcovid19' =&gt; date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar))), 'sesi' =&gt; 'Sesi 2 (14:00 - 16:00)', 'status' =&gt; 'Menunggu',]; \$this-&gt;db- &gt;table('jadwal_testcovid19')-&gt;insert(\$data); } else { do { \$hari++; \$hitung = \$this-&gt;db-&gt;table('jadwal_testcovid19') &gt;where('tgl_testcovid19', date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar)))) -&gt;countAllResults(); } while (\$hitung &gt; 1); if (\$hitung == 0) { \$data = ['id_testcovid19' =&gt; \$id_test_covid19,'tgl_testcovid19' =&gt; date('Y-m-d', strtotime('+.\$hari.' days', strtotime(\$tgl_daftar))), </pre>
---	--	--

---

---

```
'sesi' => 'Sesi 1 (10:00 - 12:00)',
'status' => 'Menunggu',];
$this->db->table('jadwal_testcovid19')
->insert($data);} else if ($hitung == 1) {
$data = ['id_testcovid19' => $id_test_covid19,
'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $hari.' days',
strtotime($tgl_daftar))), 'sesi' => 'Sesi 2 (14:00 - 16:00)',
'status' => 'Menunggu',];
$this->db->table('jadwal_testcovid19')
->insert($data); } } }
```

---

Hasil yang diharapkan

---

Sistem dapat memberikan jadwal pengecekan covid-19 secara langsung setelah pasien melakukan pendaftaran.

---

Hasil

---

8	Tambah Berita	<pre>\$model = new BeritaModel() \$periksa = \$this-&gt;validate([ 'image' =&gt; ['uploaded[image]', 'mime_in[image,image/jpeg,image/png,image/jpg]', 'max_size[image,4096'],,]); if (\$periksa) { \$image = \$this-&gt;request-&gt;getFile('image'); \$image-&gt;move(\$_SERVER['DOCUMENT_ROOT'] . '/img/berita'); \$nama_gambar = \$image-&gt;getName(); \$data = ['judul_berita' =&gt; \$this-&gt;request- &gt;getVar('judul_berita'), 'isi_berita' =&gt; \$this-&gt;request-&gt;getVar('isi_berita'), 'foto_berita' =&gt; \$nama_gambar, 'tgl_update_berita' =&gt; date('Y-m-d H:i:s')]; \$save = \$model-&gt;insert(\$data); \$this-&gt;session-&gt;setFlashdata('message', 'Berhasil Menambah Berita'); return redirect()-&gt;to('/auth/list_berita'); } else { \$this-&gt;session-&gt;setFlashdata('message', File yang anda masukkan bukan berupa gambar atau melebihi 4MB'); return redirect()-&gt;to(base_url('auth/list_berita'));</pre>
---	------------------	---

---

Hasil yang diharapkan

---

---

Admin dapat melakukan proses penambahan konten berita agar dapat ditampilkan pada halaman pengguna untuk dapat dibaca.

---

Hasil

---

```
9 Hapus Data if (!$this->ionAuth->loggedIn()) {
Berita return redirect()->to('/auth/login'); } elseif ($this
->ionAuth->inGroup('admin')) {
$session = session();
$model = new BeritaModel();
$gambar = $model->where('id', $id)->first();
if ($gambar['foto_berita'] != null) {
unlink($_SERVER['DOCUMENT_ROOT'] . '/img/berita/' .
$gambar['foto_berita']);
$data['list_berita'] = $model->where('id', $id)
->delete();
$session->setFlashdata('message', 'List Berita deleted
successfully');
return redirect()->to(base_url('auth/list_berita'));} else {
$data['list_berita'] = $model->where('id', $id)
->delete(); $session->setFlashdata('message', 'List Berita
berhasil dihapus');
return redirect()->to(base_url('auth/list_berita'));} else {
echo "Anda Tidak Memiliki Hak Akses ke Modul ini <a
href='". base_url('auth/logout') . "'>Logout</a>";}
```

---

Hasil yang diharapkan

---

Admin dapat melakukan proses penambahan konten berita agar dapat ditampilkan pada halaman pengguna untuk dapat dibaca.

---

Hasil

---

## 2. Black Box

Pengujian *black box* merupakan pengujian yang dilakukan dengan cara mengamati hasil masukan dari pengguna dan keluaran yang diberikan oleh aplikasi, apakah keluaran yang dihasilkan sudah sesuai dengan masukan yang diberikan pengguna atau tidak. Pada tahap ini praktikan membuat suatu tabel perancangan pengujian *black box* untuk menentukan

skenario pengujian serta hasil yang diharapkan pada pengujian yang akan diuji nantinya. Untuk lebih jelasnya dapat dilihat pada tabel di bawah ini.

**Tabel 3.3. Perancangan pengujian *black box*.**

No	Skenario Pengujian	Hasil yang diharapkan
1	Pengguna menekan tombol masuk untuk <i>login</i> ke akun.	Formulir <i>login</i> terbuka.
		Hasil
2	Pengguna menekan tombol daftar akun.	Formulir daftar akun terbuka.
		Hasil
3	Pengguna memilih menu pendaftaran khitan.	Halaman formulir pendaftaran khitan terbuka.
		Hasil
4	Pengguna memilih menu pendaftaran pengecekan Covid-19.	Halaman formulir pendaftaran pengecekan Covid-19 terbuka.
		Hasil
5	Pengguna memilih menu tentang kami.	Halaman tentang kami akan terbuka.
		Hasil
6	Pengguna memilih menu kontak kami.	Halaman kontak kami akan terbuka.
		Hasil
7	Pengguna memilih menu berita atau artikel.	Halaman daftar berita atau artikel terbuka.
		Hasil
8	Pengguna memilih berita yang akan dibaca.	Halaman mengenai konten berita yang dipilih akan terbuka
		Hasil

### 3.2.4. Pengujian

Setelah aplikasi selesai dibangun, praktikan melakukan dua metode pengujian yaitu *white box testing* dan *black box testing*. Pengujian ini dilakukan agar praktikan mengetahui aplikasi yang dibangun sudah siap untuk digunakan atau tidak.

#### 3.2.4.1. *White Box Testing*

*White Box Testing* merupakan jenis pengujian pada aplikasi yang berfokus pada isi *source code* yang dijalankan aplikasi. Menurut (Syafnidawaty, 2020) *White Box Testing* Salah satu cara untuk menguji aplikasi atau perangkat lunak Anda dengan memeriksa modul dan memeriksa dan menganalisis kode program yang Anda tulis adalah apakah itu salah. Setelah pembuatan aplikasi selesai praktikan menggunakan *white box testing* guna mengetahui semua kode program yang dibuat dapat berjalan sesuai yang diharapkan. Berikut tabel hasil pengujian *white box* yang dilakukan. Untuk melihat hasil pengujian pada *white box* dapat dilihat pada tabel di bawah ini.

**Tabel 3.4. Hasil pengujian dengan metode *white box*.**

No.	Komponen yang diuji	Kode Program
1	<i>Login</i>	<pre>if(\$this-&gt;request-&gt;getPost() &amp;&amp; \$this-&gt;validation -&gt;withRequest(\$this-&gt;request)-&gt;run()) { \$remember = (bool)\$this-&gt;request -&gt;getVar('remember'); if(\$this-&gt;ionAuth-&gt;login(\$this -&gt;request-&gt;getVar('identity'), \$this-&gt;request -&gt;getVar('password'), \$remember)) { \$this-&gt;session -&gt;setFlashdata('message', \$this-&gt;ionAuth -&gt;messages()); return redirect()-&gt;to('index') -&gt;withCookies(); } else { \$this-&gt;session -&gt;setFlashdata('message', \$this-&gt;ionAuth -&gt;errors(\$this-&gt;validationListTemplate)); return redirect()-&gt;back()-&gt;withInput();}}</pre>

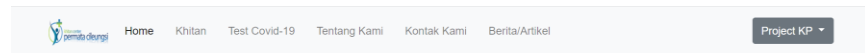
Hasil yang diharapkan

---

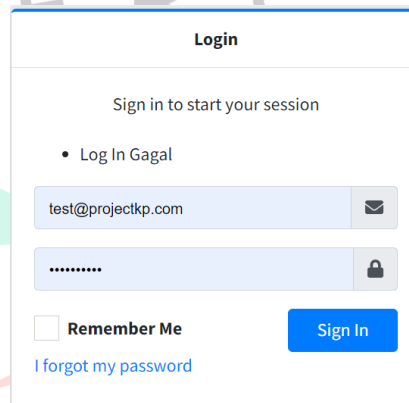
Apabila pengguna memasukkan *email* dan *password* yang benar maka pengguna akan diarahkan ke halaman setelah *login*. Jika salah satu dari identitas yang dimasukkan salah maka pengguna akan diminta untuk *login* kembali.

---

#### Hasil



Pada *header* terdapat nama akun yang sedang *login*.

The image shows a 'Login' form with the title 'Login'. Below the title, it says 'Sign in to start your session'. There is a message 'Log In Gagal' (Login Failed) in a red box. The form contains two input fields: one for email (containing 'test@projectkp.com') and one for password (masked with dots). There is a 'Remember Me' checkbox and a 'Sign In' button. A link 'I forgot my password' is also present.

Sistem akan meminta pengguna memasukkan ulang *email* dan *password* apabila tidak ditemukan data akun pada *database*.

---

```
2 Logout $this->ionAuth->logout();  
          $this->session->setFlashdata('message', $this  
          ->ionAuth->messages());  
          return redirect()->to('/auth/login')->withCookies();
```

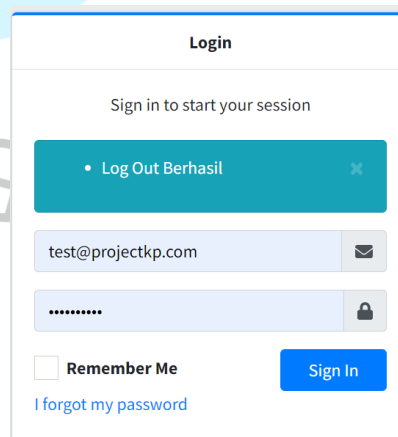
---

Hasil yang diharapkan

Pengguna dapat *logout* dari akun yang sedang *login* saat ini.

---

#### Hasil

The image shows the same 'Login' form as above, but with a message 'Log Out Berhasil' (Logout Successful) in a green box. The input fields and buttons remain the same.

Pengguna berhasil melakukan proses *logout*.

---

---

3	Ubah Kata Sandi	<pre> \$identity = \$this-&gt;session-&gt;get('identity'); \$change = \$this-&gt;ionAuth-&gt;changePassword (\$identity, \$this-&gt;request-&gt;getPost('old'), \$this -&gt;request-&gt;getPost('new')); if (\$change) { \$this-&gt;session-&gt;setFlashdata('message', \$this -&gt;ionAuth-&gt;messages()); return \$this-&gt;logout(); } else { \$this-&gt;session-&gt;setFlashdata('message', \$this -&gt;ionAuth-&gt;errors(\$this -&gt;validationListTemplate)); return redirect()-&gt;to('/auth/change_password');} </pre>
---	-----------------	--

---

Hasil yang diharapkan

Pengguna dapat mengganti kata sandi apabila kata sandi lama yang dimasukkan sesuai dengan yang tersimpan pada *database* dan kata sandi baru sesuai dengan kriteria yang ditentukan.

Hasil

The screenshot shows a login form with the following elements:
 

- Title: Login
- Text: Sign in to start your session
- Success Message Overlay (teal box):
  - Kata Sandi Berhasil Diubah
  - Log Out Berhasil
- Input fields: Email (with envelope icon) and Password (with lock icon).
- Remember Me:  Remember Me
- Sign In: A blue button labeled 'Sign In'.
- Link: [I forgot my password](#)

*Password* berhasil diganti apabila sesuai dengan kriteria yang ditentukan, dan pengguna akan diminta untuk *login* kembali.

• Bidang Kata Sandi Baru tidak cocok dengan bidang Konfirmasi Kata Sandi Baru.

Sistem akan memberikan informasi apabila terjadi kesalahan pada proses perubahan *password*.

---

4	Lupa Kata Sandi	<pre> \$forgotten = \$this-&gt;ionAuth-&gt;forgottenPassword (\$identity-&gt;{\$this-&gt;configIonAuth-&gt;identity}); if (\$forgotten) { \$this-&gt;session-&gt;setFlashdata ('message', \$this-&gt;ionAuth-&gt;messages()); return redirect()-&gt;to('/auth/login'); } else { \$this -&gt;session-&gt;setFlashdata('message', \$this-&gt;ionAuth </pre>
---	-----------------	---

---

```
->errors($this->validationListTemplate));
return redirect()->to('/auth/forgot_password');
```

Hasil yang diharapkan

Pengguna dapat mengatur ulang kata sandi apabila pengguna ingin mengganti kata sandi yang lupa.

Hasil

Sistem akan memberikan *link* perubahan *password* melalui *email* pengguna.

```
5   Membuat String acak function random_strings($length_of_string){
    $str_result = '0123456789ABCDEFGHIJKLMNPOQ
    RSTUVWXYZabcdefghijklmnopqrstuvwxyz';
    $str_result and returns; return substr(str_shuffle
    ($str_result),0, $length_of_string);}
```

Hasil yang diharapkan

Sistem membuat kode unik untuk setiap pendaftaran khitan maupun pengecekan covid-19.

Hasil

EM4l0DnLNC

Hasil dari fungsi `random_strings` yang dibuat

```
6   Menentukan Jadwal Khitan public function jadwal_khitan($id_khitan) {
    $hari = 1; $tgl_daftar = date('Y-m-d');
    $hitung = $this->db->table('jadwal_khitan')
    ->where('tgl_jadwal_khitan', date('Y-m-d',
    strtotime('+'. $hari.' days', strtotime($tgl_daftar))))
    ->countAllResults(); if ($hitung == 0) { $data = [
```

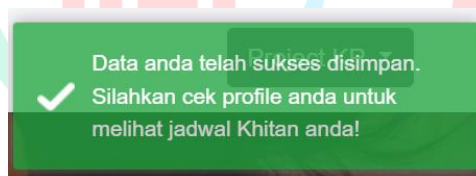


```
'id_daftar_khitan' => $id_khitan, 'tgl_jadwal_khitan' =>
date('Y-m-d', strtotime('+'. $hari.' days',
strtotime($tgl_daftar))), 'status' => 'Menunggu',];
$this->db->table('jadwal_khitan') ->insert($data);} else {
do {
$hari++; $hitung = $this->db->table('jadwal_khitan')
->where('tgl_jadwal_khitan', date('Y-m-d',
strtotime('+'. $hari.' days', strtotime($tgl_daftar)))
->countAllResults(); } while ($hitung > 0);
$data = ['id_daftar_khitan' =>
$id_khitan,'tgl_jadwal_khitan' => date('Y-m-d',
strtotime('+'. $hari.' days', strtotime($tgl_daftar))),
'status' => 'Menunggu']; $this->db
->table('jadwal_khitan')
->insert($data);}}
```

Hasil yang diharapkan

Sistem dapat dengan langsung memberikan jadwal kepada pasien perihal pendaftaran khitan yang dilakukan pasien.

Hasil



Pengguna akan mendapatkan notifikasi untuk melihat jadwal pada menu profil.

7	Menentukan Jadwal Pengecekan Covid-19	<pre>public function jadwal_testcovid19(\$id_test_covid19) { \$hari = 1; \$tgl_daftar = date('Y-m-d'); \$hitung = \$this-&gt;db- &gt;table('jadwal_testcovid19') -&gt;where('tgl_testcovid19', date('Y-m-d', strtotime('+'. \$hari.' days', strtotime(\$tgl_daftar))) -&gt;countAllResults(); if (\$hitung == 0) { \$data = ['id_testcovid19' =&gt; \$id_test_covid19, 'tgl_testcovid19' =&gt; date('Y-m-d', strtotime('+'. \$hari.' days', strtotime(\$tgl_daftar))), 'sesi' =&gt; 'Sesi 1 (10:00 - 12:00)', 'status' =&gt; 'Menunggu',]; \$this-&gt;db-&gt;table('jadwal_testcovid19') -&gt;insert(\$data); } else if (\$hitung == 1) {</pre>
---	---------------------------------------	---

---

```

$data = ['id_testcovid19' => $id_test_covid19,
'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $hari.' days',
strtotime($tgl_daftar))), 'sesi' => 'Sesi 2 (14:00 - 16:00)',
'status' => 'Menunggu']; $this->db-
>table('jadwal_testcovid19')->insert($data); } else {
do { $hari++;
$hitung = $this->db->table('jadwal_testcovid19')
>where('tgl_testcovid19', date('Y-m-d', strtotime('+'. $hari.'
days', strtotime($tgl_daftar))))
->countAllResults(); } while ($hitung > 1);
if ($hitung == 0) { $data = ['id_testcovid19' =>
$id_test_covid19,'tgl_testcovid19' => date('Y-m-d',
strtotime('+'. $hari.' days', strtotime($tgl_daftar))),
'sesi' => 'Sesi 1 (10:00 - 12:00)',
'status' => 'Menunggu'];
$this->db->table('jadwal_testcovid19')
->insert($data);} else if ($hitung == 1) {
$data = ['id_testcovid19' => $id_test_covid19,
'tgl_testcovid19' => date('Y-m-d', strtotime('+'. $hari.' days',
strtotime($tgl_daftar))), 'sesi' => 'Sesi 2 (14:00 - 16:00)',
'status' => 'Menunggu'];
$this->db->table('jadwal_testcovid19')
->insert($data); } } }

```

---

Hasil yang diharapkan

Sistem dapat memberikan jadwal pengecekan covid-19 secara langsung setelah pasien melakukan pendaftaran.

Hasil



Pengguna akan mendapatkan notifikasi untuk melihat jadwal pada menu profil.

---

8	Tambah Berita	<pre> \$model = new BeritaModel() \$periksa = \$this-&gt;validate([ 'image' =&gt; ['uploaded[image]', 'mime_in[image,image/jpeg,image/png,image/jpg]', </pre>
---	------------------	---

---

---

```

'max_size[image,4096'],,]); if ($periksa) {
$image = $this->request->getfile('image');
$image->move($_SERVER['DOCUMENT_ROOT'] .
'/img/berita'); $nama_gambar = $image->getName();
$data = ['judul_berita' => $this->request-
>getVar('judul_berita'),
'isi_berita' => $this->request->getVar('isi_berita'),
'foto_berita' => $nama_gambar,
'tgl_update_berita' => date('Y-m-d H:i:s')];
$save = $model->insert($data);
$this->session->setFlashdata('message', 'Berhasil
Menambah Berita');
return redirect()->to('/auth/list_berita');
} else { $this->session->setFlashdata('message', File
yang anda masukkan bukan berupa gambar atau
melebihi 4MB');
return redirect()->to(base_url('auth/list_berita'));}

```

---

Hasil yang diharapkan

Admin dapat melakukan proses penambahan konten berita agar dapat ditampilkan pada halaman pengguna untuk dapat dibaca.

Hasil

Berhasil Menambah Berita

Sistem akan memberikan informasi apabila berita telah tersimpan.

File yang anda masukkan bukan berupa gambar atau melebihi 4MB

Sistem akan memberikan informasi apabila terjadi kesalahan saat menyimpan konten berita.

---

9	Hapus Data Berita	<pre> if (!\$this-&gt;ionAuth-&gt;loggedIn()) { return redirect()-&gt;to('/auth/login'); } elseif (\$this -&gt;ionAuth-&gt;inGroup('admin')) { \$session = session(); \$model = new BeritaModel(); \$gambar = \$model-&gt;where('id', \$id)-&gt;first(); if (\$gambar['foto_berita'] != null) { unlink(\$_SERVER['DOCUMENT_ROOT'] . '/img/berita' . \$gambar['foto_berita']); \$data['list_berita'] = \$model-&gt;where('id', \$id) </pre>
---	----------------------	--

---

```

->delete();
$session->setFlashdata('message', 'List Berita deleted
successfully');
return redirect()->to(base_url('auth/list_berita'));} else {
$data['list_berita'] = $model->where('id', $id)
->delete(); $session->setFlashdata('message', 'List Berita
berhasil dihapus');
return redirect()->to(base_url('auth/list_berita'));} else {
echo "Anda Tidak Memiliki Hak Akses ke Modul ini <a
href='". base_url('auth/logout') . "'>Logout</a>";}

```

Hasil yang diharapkan

Admin dapat melakukan proses penambahan konten berita agar dapat ditampilkan pada halaman pengguna untuk dapat dibaca.

Hasil

List Berita berhasil dihapus

Sistem akan memberikan informasi apabila berita telah terhapus.

#### 3.2.4.2. Black Box Testing

*Black Box Testing* merupakan jenis pengujian yang dilakukan untuk mengetahui aplikasi yang dibangun apakah sudah sesuai antarmuka, fungsi, tanpa mengetahui proses *input/output* yang berjalan di dalamnya. Menurut (Astuti, 2018) *black box testing* ini mungkin terlihat mirip dengan apa yang terlihat seperti kotak hitam, anda hanya bisa melihat keluar tanpa mengetahui apa yang tersembunyi di balik cangkang hitam. Untuk melihat hasil pengujian *black box* dapat dilihat pada tabel di bawah ini.

**Tabel 3.5. Hasil pengujian dengan metode *black box*.**

No	Skenario Pengujian	Hasil yang diharapkan
1	Pengguna menekan tombol masuk untuk <i>login</i> ke akun.	Formulir <i>login</i> terbuka.

Hasil

Login

Email address

We'll never share your email with anyone else.

Password

Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.

Remember me?

Login

Setelah pengguna menekan tombol masuk maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan formulir untuk *login*.

2 Pengguna menekan tombol daftar akun. Formulir daftar akun terbuka.

#### Hasil

Login

Nama Depan: Nama Belakang:

Email:

Alamat:  
Apartment, studio, or floor

No. Handphone:  
Ketik Nomor Handphone diawali dengan 0

Password:  
Ketikan password anda 8-20 character

Confirm Password:  
Ketikan ulang password anda

Login

Setelah pengguna menekan tombol daftar akun maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan formulir untuk mendaftarkan akun.

3 Pengguna memilih menu pendaftaran khitan. Halaman formulir pendaftaran khitan terbuka.

#### Hasil

Home Khitan Test Covid-19 Tentang Kami Kontak Kami Berita/Artikel Project KP

### Pendaftaran Khitan

Nama Depan: Nama Belakang:

Masukkan Nama Depan Anda Masukkan Nama Belakang Anda

Nama Orang Tua/Wali Pasien:  
Masukkan Nama Wali Anda

Tanggal Lahir:  
dd/mm/yyyy

Alamat:  
Apartment, studio, or floor

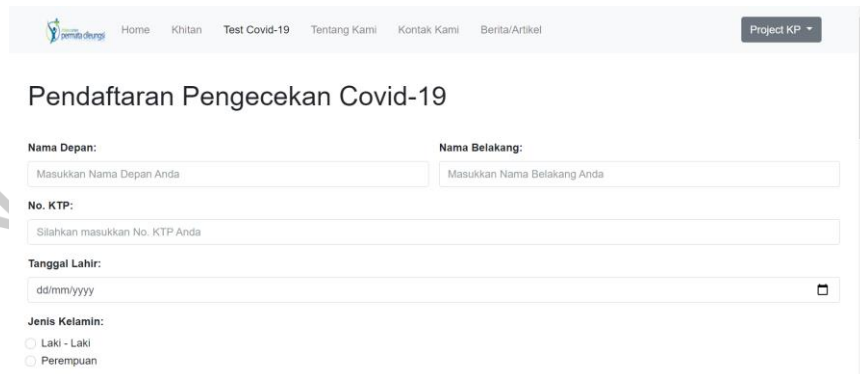
---

Setelah pengguna memilih menu pendaftaran khitan maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan formulir untuk mendaftar khitan.

---

- 4 Pengguna memilih menu pendaftaran pengecekan Covid-19. Halaman formulir pendaftaran pengecekan Covid-19 terbuka.
- 

Hasil



The screenshot shows a web browser window with the URL 'permatacileungsi.com'. The navigation menu includes 'Home', 'Khitan', 'Test Covid-19', 'Tentang Kami', 'Kontak Kami', and 'Berita/Artikel'. A 'Project KP' dropdown menu is visible in the top right. The main heading is 'Pendaftaran Pengecekan Covid-19'. The form contains the following fields: 'Nama Depan:' with a text input 'Masukkan Nama Depan Anda'; 'Nama Belakang:' with a text input 'Masukkan Nama Belakang Anda'; 'No. KTP:' with a text input 'Silahkan masukkan No. KTP Anda'; 'Tanggal Lahir:' with a date picker 'dd/mm/yyyy'; and 'Jenis Kelamin:' with radio buttons for 'Laki - Laki' and 'Perempuan'.

Setelah pengguna memilih menu pendaftaran pengecekan Covid-19 maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan formulir untuk mendaftar pengecekan Covid-19.

---

- 5 Pengguna memilih menu tentang kami. Halaman tentang kami akan terbuka.
- 

Hasil



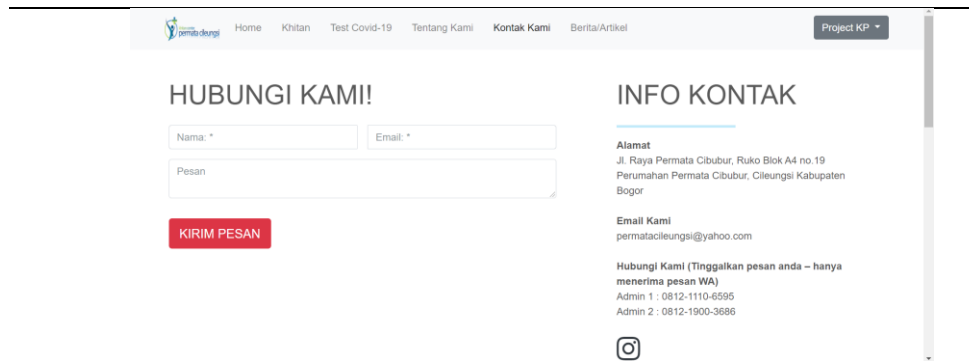
Setelah pengguna memilih menu tentang kami maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan halaman tentang kami.

---

- 6 Pengguna memilih menu kontak kami. Halaman kontak kami akan terbuka.
- 

Hasil

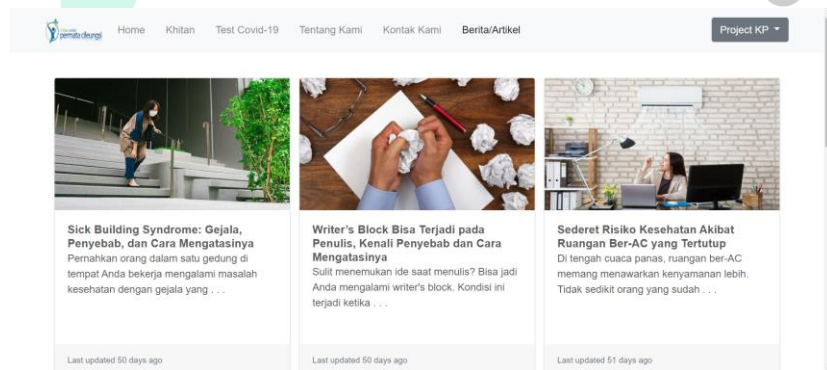
---



Setelah pengguna memilih menu kontak kami maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan halaman kontak kami.

7	Pengguna memilih menu berita atau artikel.	Halaman daftar berita atau artikel terbuka.
---	--	---

Hasil



Setelah pengguna memilih menu berita atau artikel maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan halaman berita atau artikel.

8	Pengguna memilih berita yang akan dibaca.	Halaman mengenai konten berita yang dipilih akan terbuka.
---	---	---

Hasil



Sick Building Syndrome: Gejala, Penyebab, dan Cara Mengatasinya

Monday, 01-11-2021

Pernahkan orang dalam satu gedung di tempat Anda bekerja mengalami masalah kesehatan dengan gejala yang sama secara serentak? Jika iya, fenomena tersebut dapat menjadi tanda sick building syndrome. Menariknya, gejala sick building syndrome umumnya akan langsung membaik ketika Anda pergi meninggalkan gedung. Namun, pada saat Anda kembali ke gedung yang sama, gejala akan muncul lagi dan menyerang orang-orang di dalamnya. Apa itu sick building syndrome? Sick building syndrome adalah kondisi saat beberapa atau bahkan semua orang dalam satu gedung mengalami masalah kesehatan dengan gejala yang sama. Terkadang, orang-orang tidak sadar sedang mengalaminya karena gejala yang muncul mirip penyakit flu biasa, asma, maupun alergi. Jika tidak segera diangani, kondisi ini dapat menyebabkan masalah kesehatan serius pada orang-orang di dalam gedung. Selain itu, penurunan produktivitas kerja dari orang-orang di dalamnya juga menjadi kondisi yang tidak bisa dihindari. Gejala sick building syndrome Gejala sick building syndrome hampir mirip dengan flu, asma, dan alergi. Kemiripan gejala tersebut lah yang kemudian membuat orang-orang tidak menyadari bahwa mereka sedang mengalami SBS. Sejumlah gejala sick building syndrome yang umum terjadi, di antaranya:

Setelah pengguna memilih berita atau artikel yang akan dibaca maka sistem akan membaca masukan tersebut dan memberikan umpan balik dengan menampilkan halaman baca berita atau artikel yang dipilih.

### 3.3. Kendala Yang Dihadapi

Pekerjaan adalah segala sesuatu yang dilakukan untuk mencapai tujuan tertentu. Dalam hal ini praktikan melakukan kerja profesi di CV Permata Cileungsi sebagai *web developer* untuk membangun sebuah *website* untuk Klinik Khitan Center Permata. *Website* tersebut nantinya akan digunakan secara komersial oleh Klinik Khitan Center Permata Cileungsi untuk berbagai macam hal. Namun, pada proses yang terjadi saat praktikan melakukan kerja profesi mengalami beberapa kendala yang dialami baik dari dalam diri praktikan maupun dari faktor luar. Adapun kendala – kendala yang dialami oleh praktikan selama kerja profesi sebagai berikut.

- (1) Kurangnya komunikasi dengan pembimbing kerja karena masih tingginya kasus Covid-19 di Indonesia sehingga membuat praktikan melakukan pekerjaannya melalui rumah (*Work From Home*) sehingga berkomunikasi secara *virtual* sedikit kurang efektif antara praktikan dengan pembimbing kerja.
- (2) Merupakan pertama kalinya praktikan membuat *website* untuk instansi Kesehatan, sehingga praktikan masih kurang pemahaman mengenai hal – hal yang dibutuhkan saat pembangunan aplikasi.



### 3.4. Cara Mengatasi Kendala

Manusia sebagai makhluk yang dapat berpikir dan mempunyai akal diciptakan untuk dapat menyelesaikan berbagai macam masalah yang dihadapi dalam kehidupannya, Oleh karena itu, selama praktikan melakukan kerja profesi di Klinik Khitan Center Permata Cileungsi sebagai *web developer* dapat diatasi dari dalam diri praktikan. Namun, tentu saja dalam menyelesaikan masalah praktikan mendapatkan bantuan dari pihak luar. Adapun cara yang dilakukan praktikan dalam menyelesaikan kendala yang dihadapi selama kerja profesi sebagai berikut.

- (1) Memanfaatkan jejaring sosial yang ada untuk dapat berkomunikasi dengan pembimbing kerja. Fitur *live chat* digunakan semaksimal mungkin untuk menjaga komunikasi yang terjadi antara praktikan dengan pembimbing kerja.
- (2) Mencari berbagai sumber referensi untuk praktikan dalam mencari informasi mengenai hal – hal yang dibutuhkan dalam pembuatan *website*. Mesin telusur Google dimanfaatkan untuk mencari informasi yang dibutuhkan oleh praktikan.

### 3.5. Pembelajaran Yang Diperoleh Dari Kerja Profesi

Kehidupan yang dialami setiap insan manusia pasti berbeda – beda. Namun, dalam menjalani kehidupan pasti terdapat hal yang dijadikan pembelajaran baik dari sisi positif dan negatif. Kerja profesi yang dilakukan praktikan di Klinik Khitan Center Permata Cileungsi sebagai *web developer* memberikan ilmu serta pengalaman tambahan baik dalam pembuatan *website* maupun pengalaman dalam dunia kerja. Adapun pembelajaran yang didapatkan praktikan selama bekerja di Klinik Khitan Center Permata Cileungsi sebagai berikut.

- (1) Melatih kemampuan berkomunikasi dengan orang lain secara daring dalam dunia kerja
- (2) Memahami mengenai hal – hal yang dibutuhkan oleh instansi kesehatan dalam pembangunan aplikasi.