**BAB IV**

**IMPLEMENTASI DAN PEMBAHASAN**

1. **Implementasi**

Dalam rencana penerapan sebuah sistem yang akan dibangun, agar sistem siap dioperasikan maka perlu diadakan kegiatan-kegiatan dari penerapannya. Adapun langkah-langkah yang akan dilakukan untuk penerapan sistem tersebut adalah pembuatan program, *testing* / uji coba sistem, dan sampai pada tahap pemeliharaan sistem. Berikut adalah tahap yang dilakukan penulis dalam penerapan sistem yang dibuat:

1. Uji Coba Sistem dan Program

Proses ujicoba sistem dan program merupakan hal yang sangat penting dilakukan mengingat sangat perlunya dilakukan pengujian untuk melihat kelayakan dan kemampuan maupun stabilitas program ketika dioperasikan.

Pengujian dilakukan dengan menggunakan metode *black box testing.* Pada tahap pengujian dengan metode *black box* dilakukan testing terhadap semua fungsi. Dengan metode *black box* *testing* ini maka dapat diketahui kesalahan yang terjadi seperti ketidaksesuaian *interface*, kesalahan kinerja, dan kesalahan fungsi baik dari fungsi program maupun fungsi lainnya. Dengan memasukkan semua kemungkinan input dan menjalankan semua fitur yang tersedia pada aplikasi kemudian diperiksa apakah fungsi yang diujikan tersebut menghasilkan output sesuai dengan yang diharapkan sehingga tidak ada ditemukan kesalahan *interface*, kesalahan kinerja dan fungsi.

Adapun *black box testing* yang dilakukan meliputi beberapa pengujian yang dapat dikelompokkan seperti pada Tabel 6.

Tabel 6. Pengujian Sistem Metode *Black Box*

|  |  |  |
| --- | --- | --- |
| Nama Pengujian | Kriteria Evaluasi Hasil | Jenis Pengujian |
| 1 | 2 | 3 |
| Pengujian *login* admin pada website server | Website dapat menampilkan halaman *dashboard* admin jika login berhasil | *Black Box Testing* |
| Pengujian *login* dosen pada website server | Website dapat menampilkan halaman *dashboard* dosen jika login berhasil | *Black Box Testing* |
| Pengujian terhadap tambah data | Website akan dapat menampilkan hasil proses tambah data jadwal dengan munculnya pesan “data berhasil ditambahkan” dan adanya data baru yang ditambahkan pada bagian tabel data | *Black Box Testing* |
| Pengujian terhadap edit data | Website akan dapat menampilkan hasil proses edit data jadwal dengan adanya data yang berubah pada bagian tabel data | *Black Box Testing* |
| Pengujian terhadap hapus data | Website tidak dapat menampilkan lagi data yang dihapus | *Black Box Testing* |
| Pengujian terhadap upload file | Website dapat menyimpan file didalam direktori materi pada server | *Black Box Testing* |
| Pengujian *login* mahasiswa pada aplikasi client | aplikasi dapat menampilkan halaman informasi dan menu aplikasi jika login berhasil | *Black Box Testing* |
| Pengujian terhadap menu jadwal | Aplikasi dapat menampilkan data jadwal yang dikirim oleh server pada client | *Black Box Testing* |
| Pengujian terhadap menu tugas | Aplikasi dapat menampilkan data tugas yang dikirim oleh server pada client | *Black Box Testing* |
| Pengujian terhadap menu materi | Aplikasi dapat menampilkan data materi yang dikirim oleh server pada client | *Black Box Testing* |
| Pengujian terhadap menu informasi | Aplikasi dapat menampilkan data informasi yang dikirim oleh server pada client | *Black Box Testing* |
| Pengujian terhadap *download file materi* | *User* dapat menyimpan *file* dalam *smartphone* atau *client* | *Black Box Testing* |

Pengujian pada sistem ini merupakan tahap-tahap apakah program yang diuji sesuai dengan hasil yang diharapkan, adapun hasil pengujian pada aplikasi ini dapat dilihat dengan beberapa pengujian.

1. Pengujian *website server*

Hasil pengujian ini dilakukan untuk mengetahui apakah fitur fitur pada website server sudah berjalan dengan baik dan sesuai harapan. Adapun hasil dari pengujian tersebut dapat dilihat pada Tabel

Tabel 7. Hasil Pengujian *Splash Screen*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian *login* admin | Pengguna memasukkan *username* dan *password* kemudian menekan tombol login | Website akan menampilkan halaman *dashboard* untuk admin | Halaman *dashboard* admin tampil setelah tombol login ditekan | Sesuai |
| Pengujian tambah data jadwal | Pengguna menekan tombol tambah kemudian mengisi form yang sudah disediakan oleh website | Website menampilkanpesan bahwa data jadwal sudah ditambahkan data dalam tabel jadwal bertambah | Muncul pesan “Jadwal telah ditambahkan” dan data jadwal dalam tabel telah bertambah | Sesuai |
| Pengujian edit data jadwal | Pengguna menekan tombol edit kemudian mengganti data yang ada dalam form yang muncul lalu tekan tombol simpan | Data yang dipilh dalam tabel jadwal berubah sesuai dengan perubahan yang dlakukan oleh pengguna | Muncul pesan “Jadwal telah diupdate” dan data jadwal yang dipilih dalam tabel telah berubah | Sesuai |
| Pengujian hapus data jadwal | Pengguna memilih data yang dihapus kemudian menekan tombol hapus. | Data yang dipilih tidak ditampilkan lagi oleh website | Muncul pesan “Jadwal telah dihapus” dan data jadwal yang dihapus tidak ditampilkan lagi | Sesuai |
| Pengujian tambah data ruangan | Pengguna menekan tombol tambah kemudian mengisi nama ruangan dan keterangan | data dalam tabel ruangan bertambah | Muncul pesan “ruangan telah ditambahkan” dan data dalam tabel ruangan telah bertambah | Sesuai |
| Pengujian hapus data ruangan | Pengguna memilih data yang dihapus kemudian menekan tombol hapus. | Data yang dipilih tidak ditampilkan lagi oleh website | Muncul pesan “ruangan telah dihapus” dan data ruangan yang dipilih tidak ada lagi | Sesuai |
| Pengujian tambah data pengumuman | Pengguna menekan tombol tambah kemudian mengisi perihal pengumuman dan keterangan | data dalam tabel pengumuman bertambah | Muncul pesan “pengumuman telah ditambahkan” dan data dalam tabel pengumuman telah bertambah | Sesuai |
| Pengujian hapus data pengumuman | Pengguna memilih data yang dihapus kemudian menekan tombol hapus. | Data yang dipilih tidak ditampilkan lagi oleh website | Muncul pesan “pengumuman telah dihapus” dan data ruangan yang dipilih tidak ada lagi | Sesuai |
| Pengujian *login* dosen | Pengguna memasukkan *username* dan *password* kemudian menekan tombol login | Website akan menampilkan halaman *dashboard* untuk dosen | Halaman *dashboard* dosen tampil setelah tombol login ditekan | Sesuai |
| Pengujian tambah data tugas | Pengguna menekan tombol tambah kemudian mengisi nama tugas, keterangan dan tanggal kumpul tugas lalu tekan tombol simpan | data dalam tabel pengumuman bertambah | Muncul pesan “tugas telah ditambahkan” dan data dalam tabel pengumuman telah bertambah | Sesuai |
| Pengujian edit data tugas | Pengguna menekan tombol edit kemudian mengganti data yang ada dalam form yang muncul lalu tekan tombol simpan | Data yang dipilih dalam tabel tugas berubah sesuai dengan perubahan yang dlakukan oleh pengguna | Muncul pesan “tugas telah diupdate” dan data tugas yang dipilih dalam tabel telah berubah | Sesuai |
| Pengujian hapus data tugas | Pengguna memilih data yang dihapus kemudian menekan tombol hapus. | Data yang dipilih tidak ditampilkan lagi oleh website | Muncul pesan “tugas telah dihapus” dan data jadwal yang dipilih tidak ditampilkan lagi | Sesuai |
| Pengujian upload file materi | Pengguna memilih file lalu menekan tombol upload | File ditambahkan dalam tabel daftar materi | Muncul pesan “materi telah ditambahkan” dan nama file tampil dalam tabel materi | Sesuai |

1. Pengujian aplikasi *client*

Hasil pengujian ini dilakukan untuk mengetahui apakah aplikasi client sudah berjalan dengan baik atau tidak. Berdasarkan hasil pengujian yang dilakukan, aplikasi sudah berjalan dengan baik pada pada *smartphone* *android*. Adapun hasil dari pengujian tersebut dapat dilihat pada Tabel 8.

Tabel 8. Hasil Pengujian aplikasi *client*

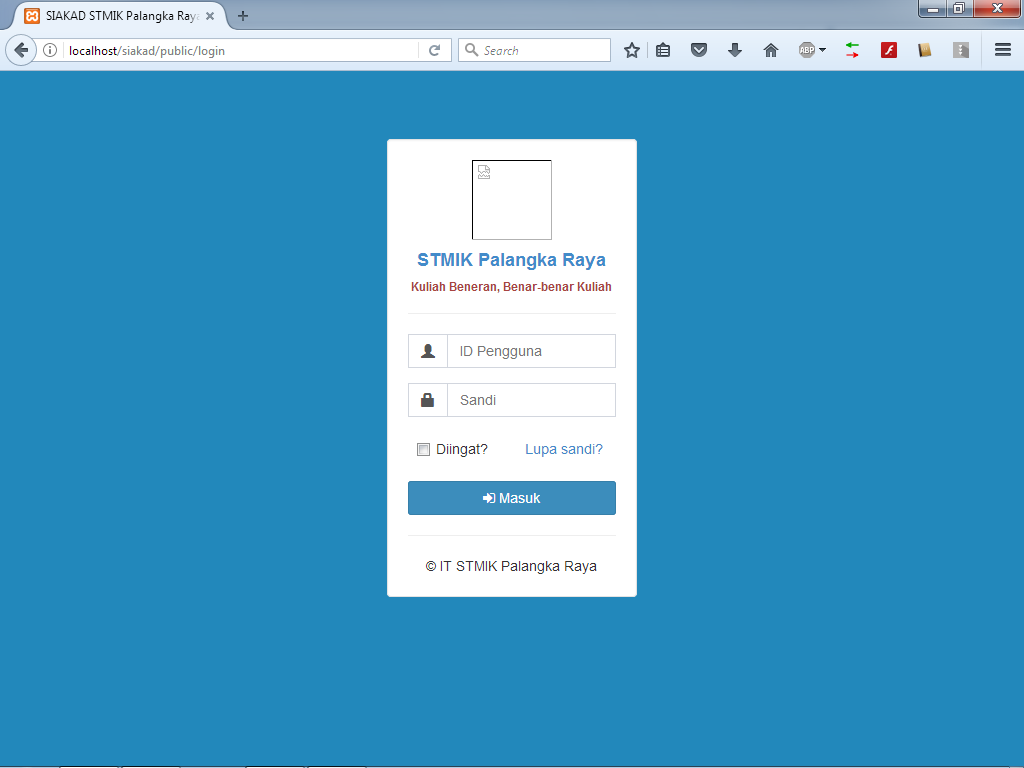
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian *login* mahasiswa | Pengguna memasukkan *username* dan *password* kemudian menekan tombol login | Aplikasi menampilkan halaman utama dan *side menu* | Halaman informasi muncul dan ada menu disisi samping halaman (*side menu)* | Sesuai |
| Pengujian menu jadwal | Pengguna menekan tombol jadwal pada side menu aplikasi | Aplikasi menampilkan halaman berisi data jadwal yang telah dikirim oleh website server | Halaman berisi data jadwal tampil setelah proses *loading* aplikasi | Sesuai |
| Pengujian menu tugas | Pengguna menekan tombol tugas pada side menu aplikasi | Aplikasi menampilkan halaman berisi data tugas yang telah dikirim oleh website server | Halaman berisi data tugas tampil setelah proses *loading* aplikasi | Sesuai |
| Pengujian menu materi | Pengguna menekan tombol materi pada side menu aplikasi | Aplikasi menampilkan halaman berisi data materi yang telah dikirim oleh website server | Halaman berisi data materi tampil setelah proses *loading* aplikasi | Sesuai |
| Pengujian menu informasi | Pengguna menekan tombol info pada side menu aplikasi | Aplikasi menampilkan halaman berisi data pengumuman yang telah dikirim oleh website server | Halaman berisi data pengumuman tampil setelah proses *loading* aplikasi | Sesuai |
| Pengujian download file materi | Pada halaman materi pengguna memilih materi yang akan di download kemudian menekan tombol download | Aplikasi dapat menyimpan *file* dari *server* ke dalam *smartphone* | File tersimpan dalam penyimpanan internal smartphone setelah proses loading | Sesuai |
| Pengujian menu logout | Penguna menekan tombol logout pada side menu aplikasi | Aplikasi kembali menampilkan halaman login | Halaman login muncul | Sesuai |

1. Manual program

Manual program merupakan konsep untuk menguraikan penggunaan program Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya secara keseluruhan, mulai dari cara menjalankan program untuk pertama kali, mengoperasikan tombol-tombol dan proses lainnya di dalam program. Sehingga dapat memudahkan pengguna untuk menggunakan aplikasi, berikut uraian penggunaan program. Dalam manual ini akan dijelaskan dari tiga bagian yaitu admin, dosen dan mahasiwa.

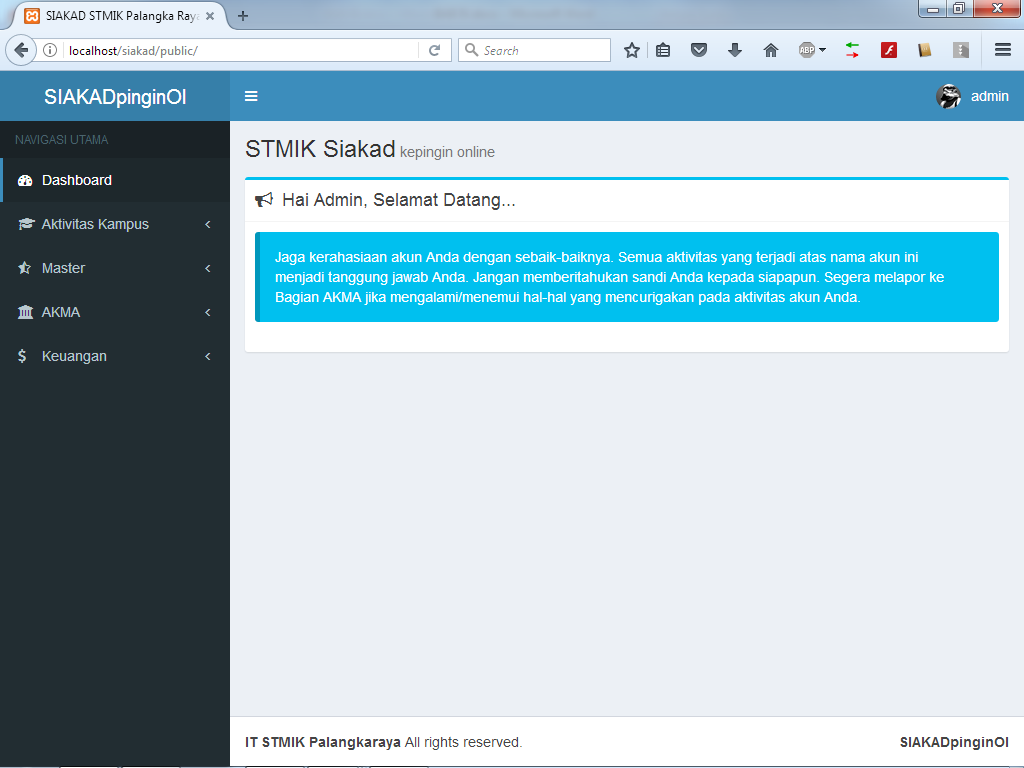
1. Admin

Bagian admin adalah bagian program yang mengelola data – data dosen dan mahasiswa. untuk masuk sebagai admin pertama buka browser dikomputer dan buka website server pada browser. Maka halaman yang pertama kali muncul adalah halaman login lalu masukkan username dan password. Berikut adalah tampilan halaman login website ditunjukkan pada gambar



Gambar 31. Halaman *login server*

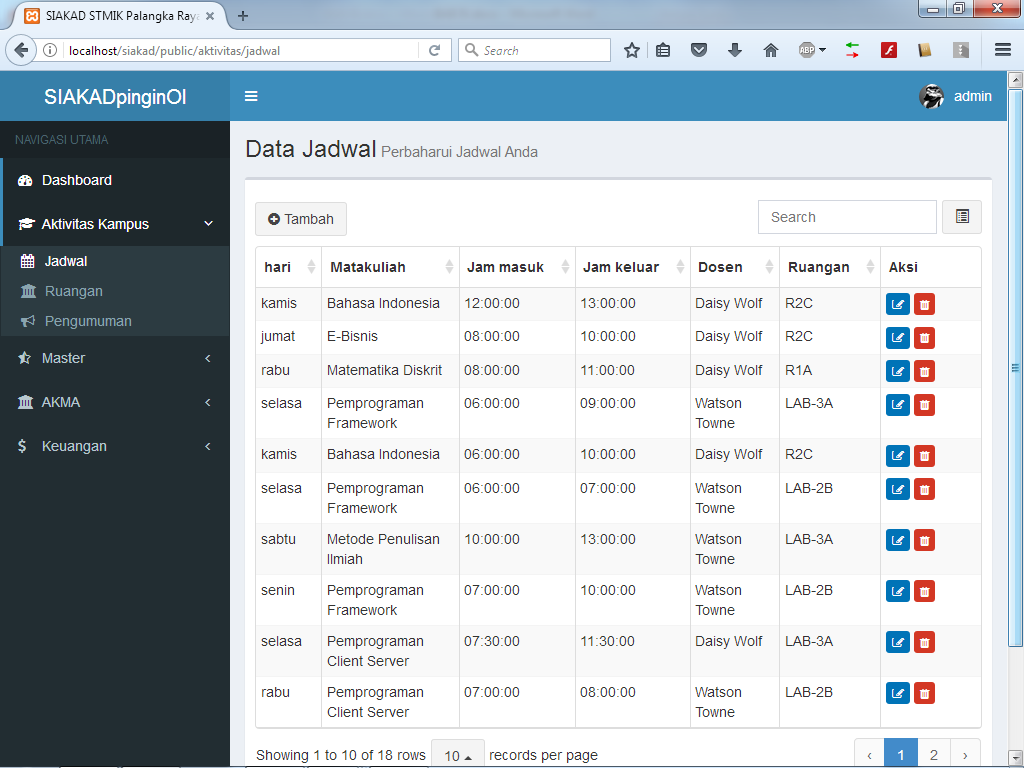
Setelah memasukkan username dan password maka halaman akan beralih ke halaman dashboard admin. Tampilan dashboard admin dapat dilihat pada gambar



Gambar 31. Halaman *dashboard admin*

1. Menu jadwal bagian admin

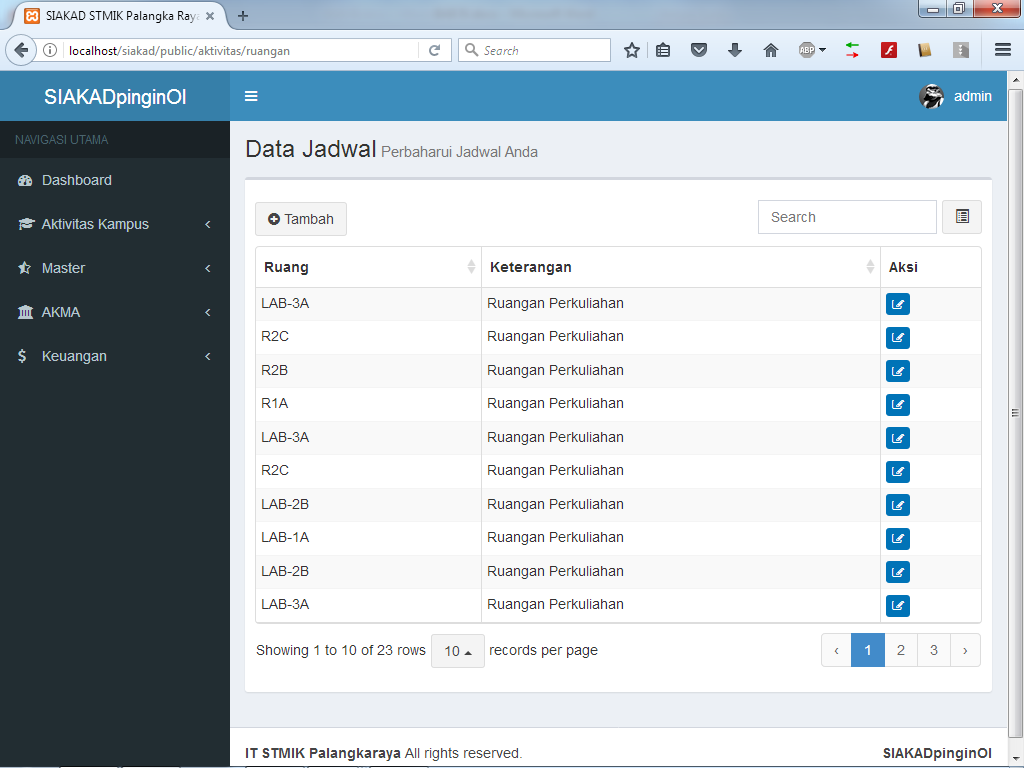
Menu jadwal digunakan untuk mengelola jadwal admin dapat menambahkan, mengedit dan menghapus jadwal matakuliah. Tampilan halaman jadwal dapat dilihat pada gambar



Gambar halaman jadwal bagian admin

1. Menu ruangan

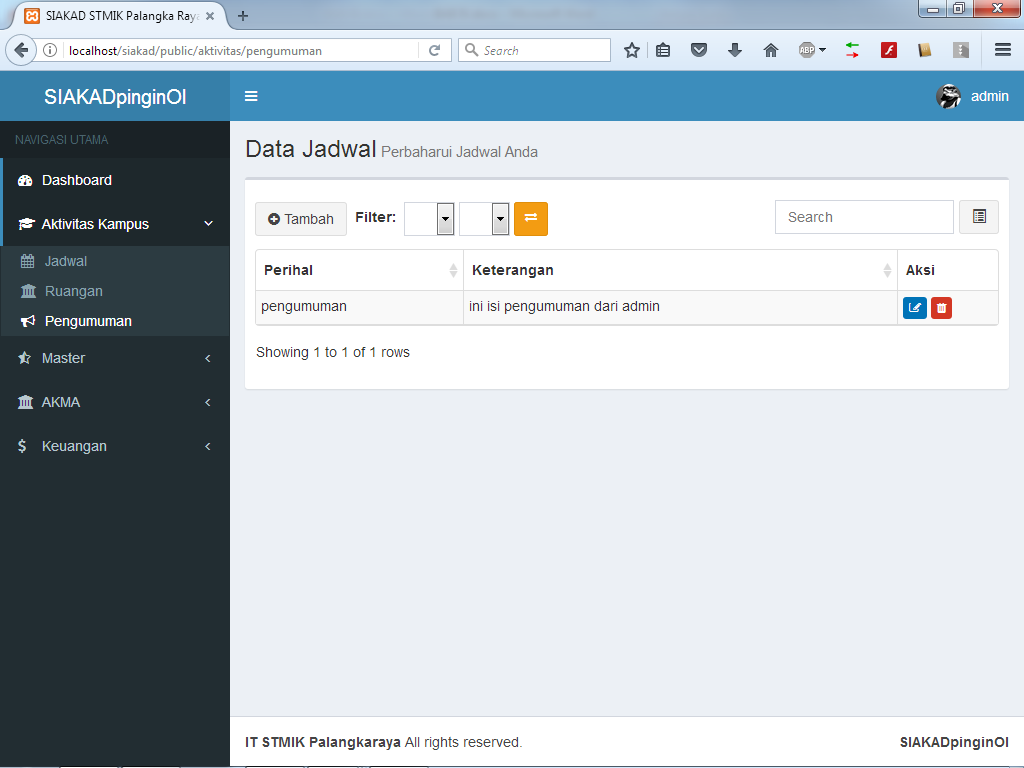
Menu ruangan digunnakan untuk mengelola ruangan, admin dapat menambah dan menghapus ruangan yang akan dipakai oleh jadwal matakuliah. Halaman ruangan dapat dilihat pada gambar



Gambar halaman ruangan

1. Menu pengumuman bagian admin

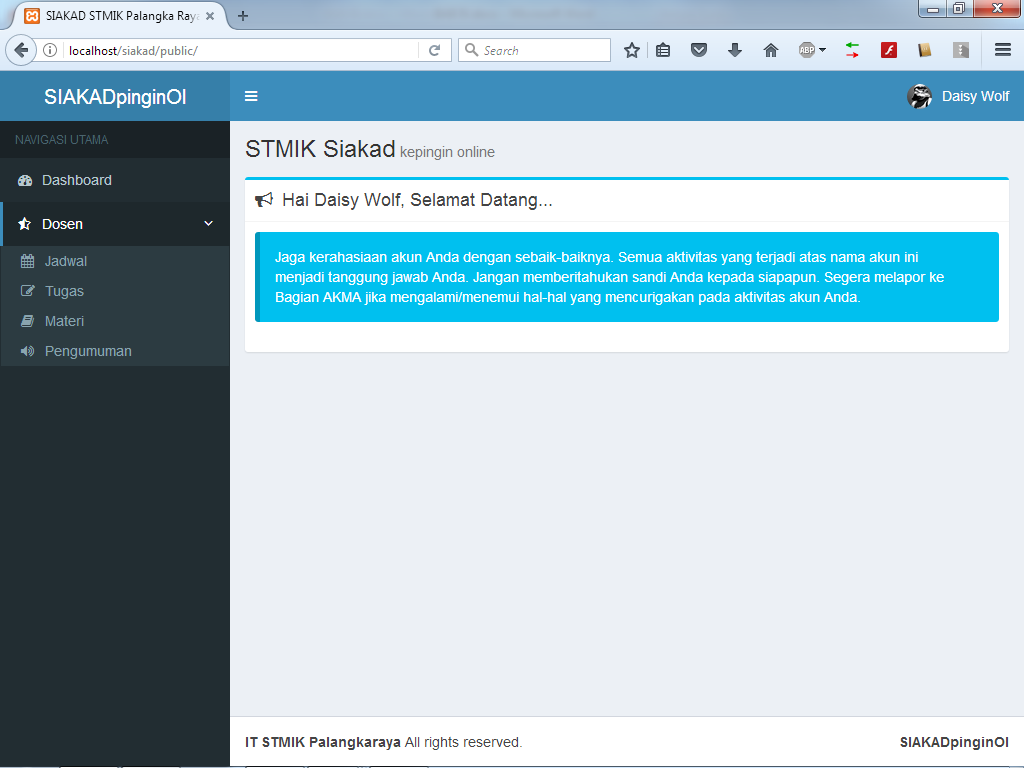
Menu pengumuman digunnakan untuk menambahkan dan menghapus pengumuman untuk mahasiswa. Halaman pengumuman dapat dilihat pada gambar



Gambar halaman ruangan

1. Dosen

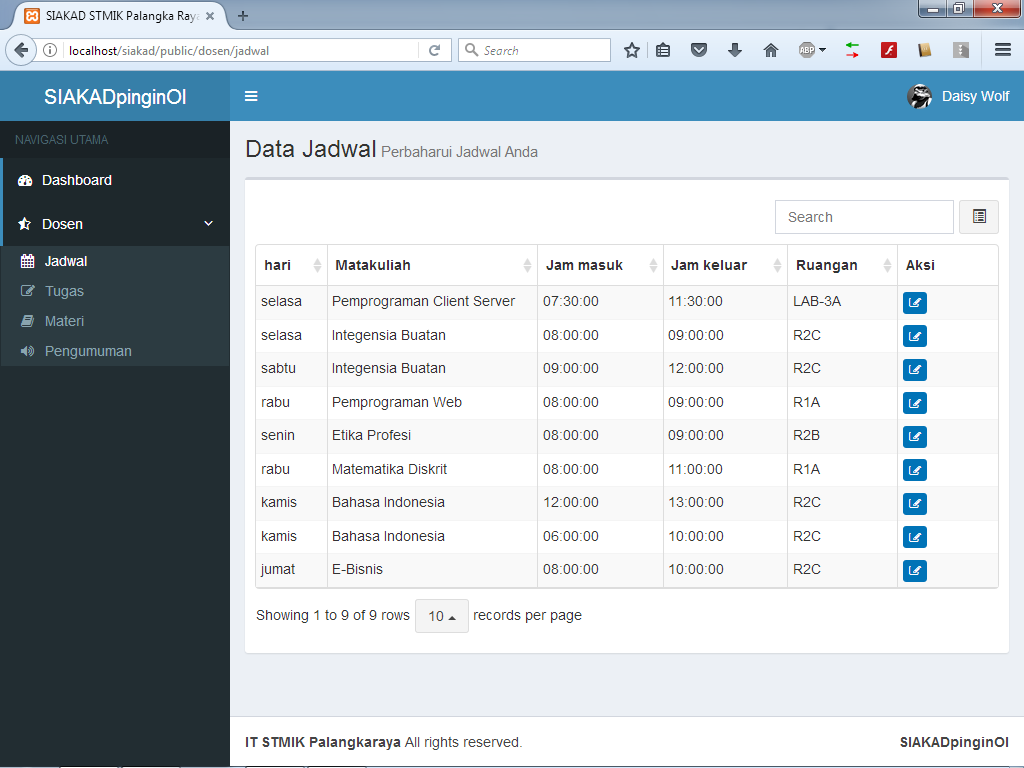
Bagian dosen adalah bagian program yang mengelola data – data jadwal, materi, tugas dan juga pengumuman. untuk masuk sebagai dosen pertama buka browser dikomputer dan buka website server pada browser. Maka halaman yang pertama kali muncul adalah halaman login lalu masukkan username dan password. Kemudian akan tampil halaman dashboard dosen. Berikut adalah tampilan halaman dashboard dosen ditunjukkan pada gambar



Gambar halaman dashboard bagian dosen

1. Menu jadwal bagian dosen

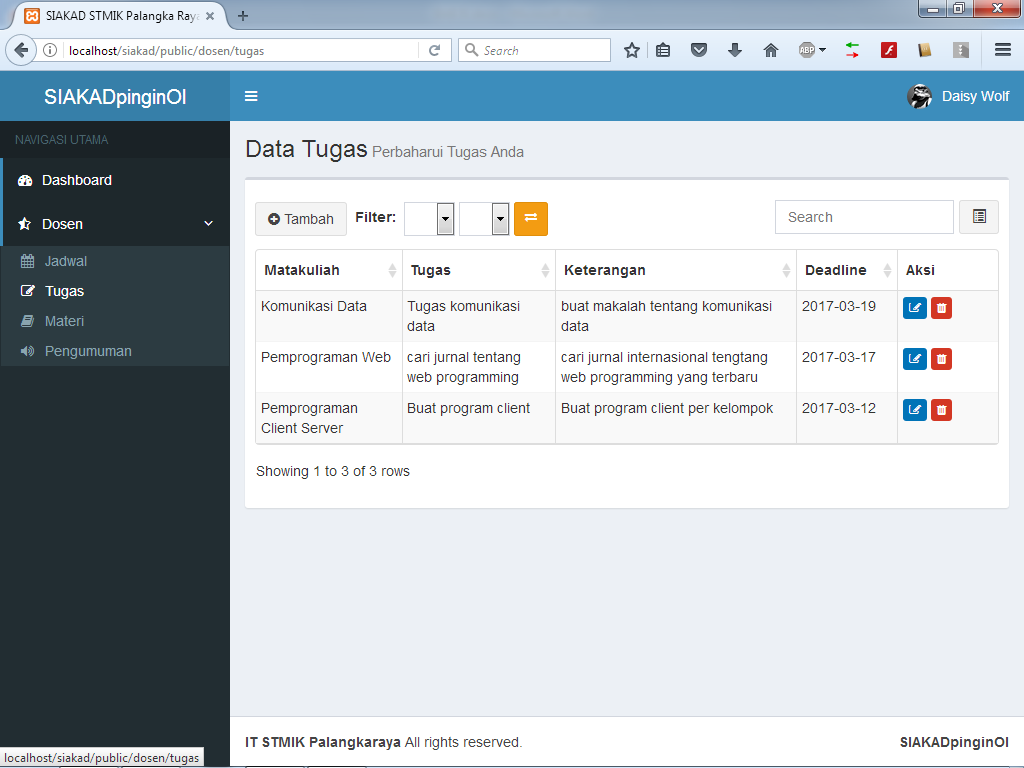
Menu jadwal bagian dosen digunnakan untuk mengelola jadwal, dosen dibatasi hanya bisa mengedit jadwal, dosen tidak bisa menambah dan menghapus jadwal. Halaman jadwal bagian dosen dapat dilihat pada gambar



Gambar halaman jadwal bagian dosen

1. Menu tugas bagian dosen

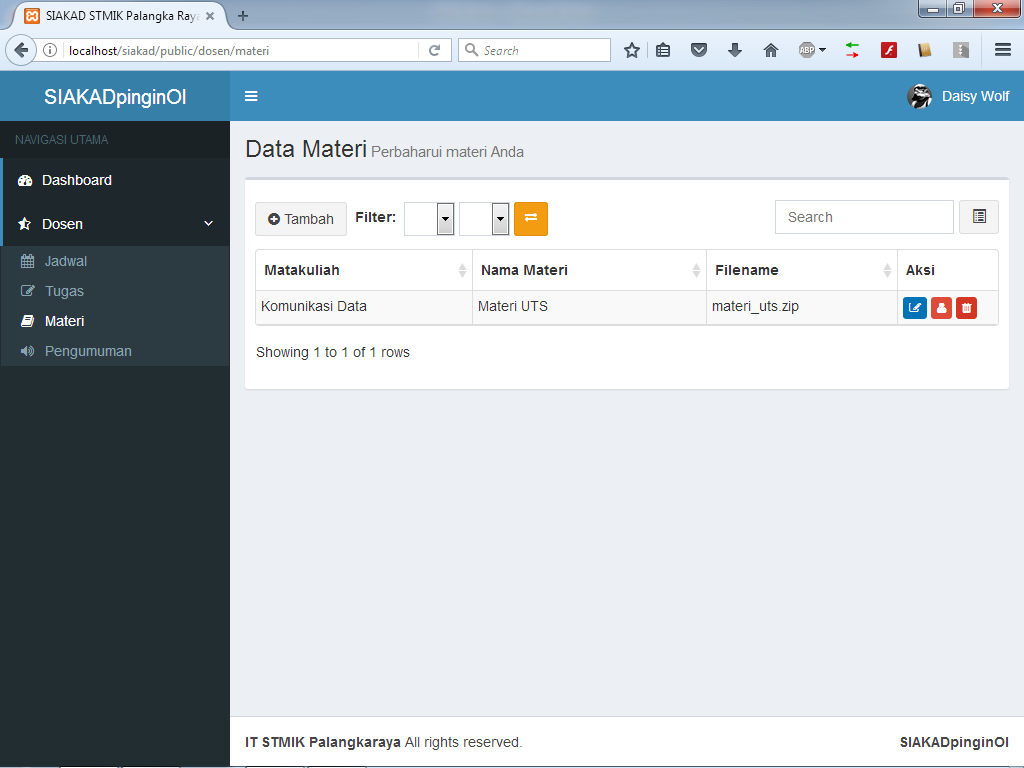
Menu tugas bagian dosen digunnakan untuk mengelola tugas, dosen dapat menambah, mengedit dan menghapus tugas. Halaman tugas dapat dilihat pada gambar.



Gambar halaman tugas bagian dosen

1. Menu materi bagian dosen

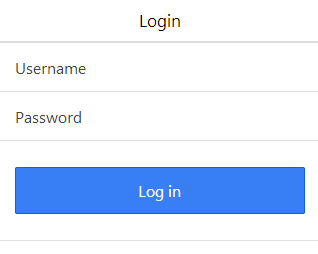
Menu materi bagian dosen digunnakan untuk mengelola materi, dosen dapat menambah, mengedit dan menghapus materi. Halaman materi dapat dilihat pada gambar



Gambar halaman materi bagian dosen

1. Mahasiswa

Bagian mahasiswa adalah bagian program yang menerima data–data dari server untuk ditampilkan kepada client/mahasiswa. untuk masuk sebagai mahasiswa pertama buka aplikasi pada smartphone. Maka akan muncul splashscreen aplikasi, kemudian akan tampil halaman login lalu masukkan username dan password mahasiswa. Berikut adalah tampilan halaman login aplikasi ditunjukkan pada gambar.



Gambar halaman login aplikasi client

1. Side menu

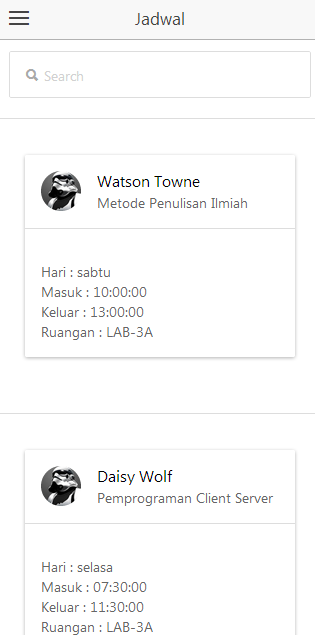
Side menu merupakan halaman berisi daftar menu - menu pada aplikasi client yang berada disamping halaman utama aplikasi. Halaman side menu dapat dilihat pada gambar.



Gambar 33. *side menu* pada aplikasi client

1. Halaman jadwal

Halaman menu jadwal merupakan halaman yang menampilkan jadwal milik mahasiswa. Halaman menu jadwal dapat dilihat pada Gambar 38.



Gambar 38. halaman jadwal pada aplikasi client

1. Menu tugas

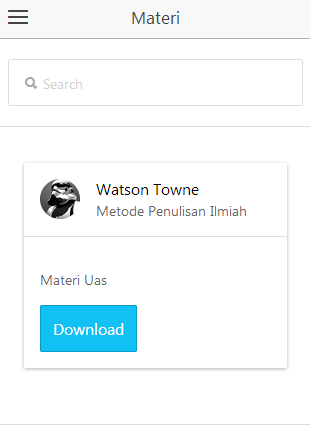
Halaman menu tugas merupakan halaman berisi data tugas yang diberikan oleh dosen kepada mahasiswa. Adapaun tampilan halaman menu tugas dapat dilihat pada Gambar 39.

****

Gambar 39. Halaman tugas pada aplikasi client

1. Halaman materi

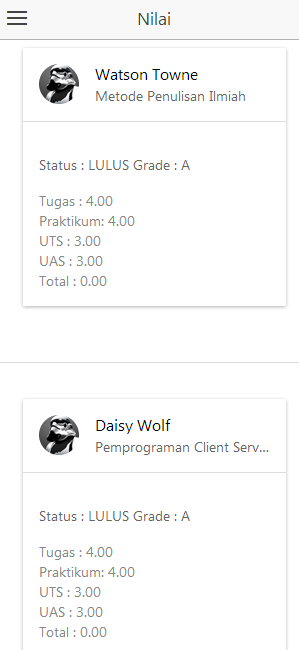
Halaman materi merupakan halaman berisi data materi, pada halaman materi terdapat tombol download untuk menngunduh file materi dari server. Halaman materi dapat dilihat pada Gambar 40.



Gambar 40. Halaman materi pada aplikasi client

1. Halaman nilai

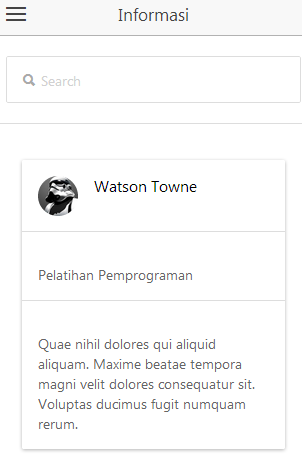
Halaman nilai merupakan halaman yang memuat daftar nilai mahasiswa. Halaman nilai dapat dilihat pada Gambar 41.



Gambar 41. Halaman nilai pada aplikasi *client*

1. Halaman informasi

Halaman informasi merupakan halaman yang menampilkan data informasi atau pengumuman. Halaman informasi/pengumuman dapat dilihat pada Gambar 42.



Gambar 42. Halaman informasi

1. Manual instalasi

Pada manual instalasi ini akan dijelaskan tentang bagaimana cara instalasi website server pada server lokal di komputer bersistem operasi *windows 7* dan instalasi aplikasi client pada sistem operasi *android*.

1. Manual instalasi website server
2. Copy file “siakad.rar” ke komputer dan ekstrak kedalam *folder* C:\xampp\htdocs
3. Kemudian jalankan aplikasi xampp dan buka command prompt, masuk kedalam path C:\xampp\htdocs\siakad lalu keitikkan perintah php artisan serve.
4. Manual instalasi aplikasi *client*
5. *Copy file* “*stmik.apk*” ke *SD card* atau *internal smartphone android* melalui kabel data, *bluetooth* atau melalui media lainnya.
6. Tap *file* “*stmik.apk*” yang sudah dipindahkan, maka instalasi aplikasi akan berjalan. Tunggu hingga proses instalasi selesai.
7. Lihat apakah *icon* doa Stmik palangkaraya sudah muncul pada layar *smartphone android* pengguna. Apabila sudah ada, aplikasi sudah siap dijalankan.
8. Pemeliharaan sistem

Pemeliharaan aplikasi perlu dilakukan agar sistem dapat berjalan lebih baik dan bertahan lebih lama, dilakukan penjagaan dari *asset* aplikasi itu sendiri maupun komponennya. Berikut kiat dan pemeliharaan aplikasi:

1. Nama *file* aplikasi asli sebaiknya tidak diganti karena akan mengalami masalah dalam kompresi *file* aslinya.
2. Penyimpanan *asset* aplikasi sangat penting, karena akan memudahkan *programmer* dalam menyusun kembali aplikasi apabila terjadi perubahan, *error,* maupun pengembangan ke tahap selanjutnya.
3. Melakukan *backup* aplikasi untuk menghindari kehilangan aplikasi.
4. Melakukan *backup database* untuk menhidari kehilangan data.
5. **Pembahasan**
6. Pembahasan *listing* kode program
7. Kode program database.php

Kode program pada file database.php yang berada di folder config berguna untuk membuat koneksi pada database mysql.

|  |
| --- |
| 'connections' => [  'mysql' => [  'driver' => 'mysql',  'host' => env('DB\_HOST', 'localhost'),  'database' => env('DB\_DATABASE', 'forge'),  'username' => env('DB\_USERNAME', 'root'),  'password' => env('DB\_PASSWORD', ''),  'charset' => 'utf8',  'collation' => 'utf8\_unicode\_ci',  'prefix' => '',  'strict' => false,  ], |

1. Kode program AuthSelfController.php

Kode program pada file AuthSelfController berfungsi untuk mengatur login user pada website

|  |
| --- |
| namespace Stmik\Http\Controllers\Auth;  use Illuminate\Http\Request;  class AuthSelfController extends AuthController  {  protected $redirectTo = '/';  protected function getCredentials(Request $request)  {  $field = filter\_var($request->input('email'), FILTER\_VALIDATE\_EMAIL) ? 'email' : 'name';  $request->merge([$field=>$request->input('email')]);  return $request->only($field, 'password');  }  protected function authenticated(Request $request, $guard)  {  return \Response::make()  ->header('X-IC-Redirect', url('/'));  }  protected function sendFailedLoginResponse(Request $request)  {  if($request->ajax()) {  return response()->json([  $this->loginUsername()=>$this->getFailedLoginMessage()  ], 422); // 422 adalah unprocessable entity  }  return parent::sendFailedLoginResponse($request);  }  } |

1. Kode program route.php

Kode program pada file route.php berfungsi untuk mengatur seluruh *url* website, dan juga code pada route.php yang menentukan halaman pada *url* tertentu perlu login admin, dosen atau mahasiswa.

|  |
| --- |
| <?php  //Route::auth();  Route::group(['middleware' => ['web']], function () {  // Authentication Routes...  $this->get('login', 'Auth\AuthSelfController@showLoginForm');  $this->post('login', 'Auth\AuthSelfController@login');  $this->get('logout', 'Auth\AuthSelfController@logout');  Route::group(['middleware' =>['auth']], function() {  Route::get('/', ['as' => 'home', 'uses' => 'SiteController@index']);  //Jadwal Untuk Master  Route::get('/aktivitas/jadwal/', ['as' => 'aktivitas.jadwal', 'uses' => 'Master\MasterJadwalController@index']);  Route::get('/aktivitas/jadwal/getDT', ['as' => 'aktivitas.jadwal.getDT', 'uses' => 'Master\MasterJadwalController@getDataBtTable']);  Route::get('/aktivitas/jadwal/create', ['as' => 'aktivitas.jadwal.create', 'uses' => 'Master\MasterJadwalController@create']);  Route::post('/aktivitas/jadwal/store', ['as' => 'aktivitas.jadwal.store', 'uses' => 'Master\MasterJadwalController@store']);  Route::get('/aktivitas/jadwal/edit/{id}', ['as' => 'aktivitas.jadwal.edit', 'uses' => 'Master\MasterJadwalController@edit']);  Route::post('/akativitas/jadwal/update/{id}', ['as' => 'aktivitas.jadwal.update', 'uses' => 'Master\MasterJadwalController@update']);  Route::delete('/aktivitas/jadwal/delete/{id}', ['as' => 'aktivitas.jadwal.delete', 'uses' => 'Master\MasterJadwalController@delete']);  //Pengumuman untuk Master  Route::get('/aktivitas/pengumuman/', ['as' => 'aktivitas.pengumuman', 'uses' => 'Master\PengumumanController@index']);  Route::get('/aktivitas/pengumuman/getDT', ['as' => 'aktivitas.pengumuman.getDT', 'uses' => 'Master\PengumumanController@getDataBtTable']);  Route::get('/aktivitas/pengumuman/create', ['as' => 'aktivitas.pengumuman.create', 'uses' => 'Master\PengumumanController@create']);  Route::post('/aktivitas/pengumuman/store', ['as' => 'aktivitas.pengumuman.store', 'uses' => 'Master\PengumumanController@store']);  Route::get('/aktivitas/pengumuman/edit/{id}', ['as' => 'aktivitas.pengumuman.edit', 'uses' => 'Master\PengumumanController@edit']);  Route::post('/aktivitas/pengumuman/update/{id}', ['as' => 'aktivitas.pengumuman.update', 'uses' => 'Master\PengumumanController@update']);  Route::delete('/aktivitas/pengumuman/delete/{id}', ['as' => 'aktivitas.tugas.delete', 'uses' => 'Master\PengumumanController@delete']);  //ruangan  Route::get('/aktivitas/ruangan/', ['as' => 'aktivitas.ruangan', 'uses' => 'Master\RuanganController@index']);  Route::get('/aktivitas/ruangan/getDT', ['as' => 'aktivitas.ruangan.getDT', 'uses' => 'Master\RuanganController@getDataBtTable']);  Route::get('/aktivitas/ruangan/create', ['as' => 'aktivitas.ruangan.create', 'uses' => 'Master\RuanganController@create']);  Route::post('/aktivitas/ruangan/store', ['as' => 'aktivitas.ruangan.store', 'uses' => 'Master\RuanganController@store']);  Route::get('/aktivitas/ruangan/edit/{id}', ['as' => 'aktivitas.ruangan.edit', 'uses' => 'Master\RuanganController@edit'])  Route::post('/akativitas/ruangan/update/{id}', ['as' => 'aktivitas.ruangan.update', 'uses' => 'Master\RuanganController@update']);  Route::delete('/aktivitas/ruangan/delete/{id}', ['as' => 'aktivitas.ruangan.delete', 'uses' => 'Master\RuanganController@delete']);  //Jadwal Untuk dosen  Route::get('/dosen/jadwal/', ['as' => 'dosen.jadwal', 'uses' => 'Dosen\JadwalController@index']);  Route::get('/dosen/jadwal/getDT', ['as' => 'dosen.jadwal.getDT', 'uses' => 'Dosen\JadwalController@getDataBtTable']);  Route::post('/dosen/jadwal/store', ['as' => 'dosen.jadwal.store', 'uses' => 'Dosen\JadwalController@store']);  Route::get('/dosen/jadwal/edit/{id}', ['as' => 'dosen.jadwal.edit', 'uses' => 'Dosen\JadwalController@edit']);  Route::post('/dosen/jadwal/update/{id}', ['as' => 'dosen.jadwal.update', 'uses' => 'Dosen\JadwalController@update']);  //Tugas  Route::get('/dosen/tugas/', ['as' => 'dosen.tugas', 'uses' => 'Dosen\TugasController@index']);  Route::get('/dosen/tugas/getDT', ['as' => 'dosen.tugas.getDT', 'uses' => 'Dosen\TugasController@getDataBtTable']);  Route::get('/dosen/tugas/create', ['as' => 'dosen.tugas.create', 'uses' => 'Dosen\TugasController@create']);  Route::post('/dosen/tugas/store', ['as' => 'dosen.tugas.store', 'uses' => 'Dosen\TugasController@store']);  Route::get('/dosen/tugas/edit/{id}', ['as' => 'dosen.tugas.edit', 'uses' => 'Dosen\TugasController@edit']);  Route::post('/dosen/tugas/update/{id}', ['as' => 'dosen.tugas.update', 'uses' => 'Dosen\TugasController@update']);  Route::delete('/dosen/tugas/delete/{id}', ['as' => 'dosen.tugas.delete', 'uses' => 'Dosen\TugasController@delete']);  //Materi  Route::get('/dosen/materi/', ['as' => 'dosen.materi', 'uses' => 'Dosen\MateriController@index']);  Route::get('/dosen/materi/getDT', ['as' => 'dosen.materi.getDT', 'uses' => 'Dosen\MateriController@getDataBtTable']);  Route::get('/dosen/materi/create', ['as' => 'dosen.materi.create', 'uses' => 'Dosen\MateriController@create']);  Route::post('/dosen/materi/store', ['as' => 'dosen.materi.store', 'uses' => 'Dosen\MateriController@store']);  Route::get('/dosen/materi/edit/{id}', ['as' => 'dosen.materi.edit', 'uses' => 'Dosen\MateriController@edit']);  Route::post('/dosen/materi/update/{id}', ['as' => 'dosen.materi.update', 'uses' => 'Dosen\MateriController@update']);  Route::delete('/dosen/materi/delete/{id}', ['as' => 'dosen.tugas.delete', 'uses' => 'Dosen\MateriController@delete']);  //Pengumuman untuk dosen  Route::get('/dosen/pengumuman/', ['as' => 'dosen.pengumuman', 'uses' => 'Dosen\PengumumanController@index']);  Route::get('/dosen/pengumuman/getDT', ['as' => 'dosen.pengumuman.getDT', 'uses' => 'Dosen\PengumumanController@getDataBtTable']);  Route::get('/dosen/pengumuman/create', ['as' => 'dosen.pengumuman.create', 'uses' => 'Dosen\PengumumanController@create']);  Route::post('/dosen/pengumuman/store', ['as' => 'dosen.pengumuman.store', 'uses' => 'Dosen\PengumumanController@store']);  Route::get('/dosen/pengumuman/edit/{id}', ['as' => 'dosen.pengumuman.edit', 'uses' => 'Dosen\PengumumanController@edit']);  Route::post('/dosen/pengumuman/update/{id}', ['as' => 'dosen.pengumuman.update', 'uses' => 'Dosen\PengumumanController@update']);  Route::delete('/dosen/pengumuman/delete/{id}', ['as' => 'dosen.tugas.delete', 'uses' => 'Dosen\PengumumanController@delete']);  });  });  Route::group(array('prefix'=>'api'),function(){  Route::group(['middleware' => 'cors'], function() {  Route::get('/login/', ['as' => 'api.login', 'uses' => 'ApiAuthController@user\_login']);  Route::group(['middleware' => 'token.check'], function() {  Route::get('/jadwal/{nim}/{token}', ['as' => 'api.jadwal', 'uses' => 'MahasiswaApiController@jadwal']);  Route::get('/tugas/{nim}/{token}', ['as' => 'api.tugas', 'uses' => 'MahasiswaApiController@tugas']);  Route::get('/nilai/{nim}/{token}', ['as' => 'api.nilai', 'uses' => 'MahasiswaApiController@nilai']);  Route::get('/materi/{nim}/{token}', ['as' => 'api.materi', 'uses' => 'MahasiswaApiController@materi']);  Route::get('/pengumuman/{nim}/{token}', ['as' => 'api.pengumuman', 'uses' => 'MahasiswaApiController@pengumuman']);  });  });  Route::get('/download/{filename}', ['as' => 'api.download', 'uses' => 'MahasiswaApiController@download']);  }); |

1. Kode program MasterJadwalController.php

Kode program file MasterJadwalController.php berfungsi untuk menambah, menghapus dan mengedit jadwal. File ini memerlukan hak akses sebagai admin untuk menjalankannya.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers\Master;  use Stmik\Factories\MasterJadwalFactory;  use Stmik\Http\Controllers\Controller;  use Stmik\Http\Controllers\input;  use Stmik\Http\Controllers\GetDataBTTableTrait;  use Stmik\Http\Requests\JadwalRequest;  class MasterJadwalController extends Controller  {  use GetDataBTTableTrait;  /\*\* @var MasterMahasiswaFactory \*/  protected $factory;  public function \_\_construct(MasterJadwalFactory $factory)  {  $this->factory = $factory;  $this->middleware('auth.role:master');  }  public function index()  {  return view('aktivitas.jadwal.index')  ->with('layout', $this->getLayout());  }  public function create()  {  return view('aktivitas.jadwal.form')  ->with('data', null)  ->with('action', route('aktivitas.jadwal.store'));  }  public function edit($id)  {  return view('aktivitas.jadwal.form')  ->with('data', $this->factory->getDataJadwal($id))  ->with('action', route('aktivitas.jadwal.update', ['id'=>$id]));  }  public function update($id, JadwalRequest $request)  {  $input = $request->all();  if($this->factory->checkValidJadwal($id,$request)){  return $this->edit($id)->with('error', "ada jadwal yang sama!");  }else{  if($this->factory->update($id, $input)) {  return $this->edit($id)->with('success', "Data Jadwal telah terupdate!");  }  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function store(JadwalRequest $request)  {  $input = $request->all();  if($this->factory->store($input)) {  return $this->create()->with('success', "Data NIM {$this->factory->getLastInsertId()} telah ditambahkan, silahkan lakukan proses penambahan lainnya!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function delete($id)  {  if($this->factory->delete($id)) {  return response("", 200,['X-IC-Remove'=>true]);  }  return response(json\_encode($this->factory->getErrors()), 500,['X-IC-Remove'=>false]);  }  } |

1. Kode program RuanganController.php

Kode program pada file RuanganController.php berfungsi untuk mengelola ruangan seperti menambah, mengedit dan menghapus ruangan. File ini memerlukan hak akses sebagai admin untuk menjalankannya.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers\Master;  use Stmik\Factories\RuanganFactory;  use Stmik\Http\Controllers\Controller;  use Stmik\Http\Controllers\GetDataBTTableTrait;  use Stmik\Http\Requests\RuanganRequest;  class RuanganController extends Controller  {  use GetDataBTTableTrait;  /\*\* @var MasterMahasiswaFactory \*/  protected $factory;  public function \_\_construct(RuanganFactory $factory)  {  $this->factory = $factory;  $this->middleware('auth.role:master');  }  public function index()  {  return view('aktivitas.ruangan.index')  ->with('layout', $this->getLayout());  }  public function create()  {  return view('aktivitas.ruangan.form')  ->with('data', null)  ->with('action', route('aktivitas.ruangan.store'));  }  public function edit($id)  {  return view('aktivitas.ruangan.form')  ->with('data', $this->factory->getDataRuangan($id))  ->with('action', route('aktivitas.ruangan.update', ['id'=>$id]));  }  public function update($id, RuanganRequest $request)  {  $input = $request->all();  if($this->factory->update($id, $input)) {  return $this->edit($id)->with('success', "Data Ruangan telah terupdate!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function store(RuanganRequest $request)  {  $input = $request->all();  if($this->factory->store($input)) {  return $this->create()->with('success', "Data Ruangan {$this->factory->ruang} telah ditambahkan, silahkan lakukan proses penambahan lainnya!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function delete($id)  {  if($this->factory->delete($id)) {  return response("", 200,['X-IC-Remove'=>true]);  }  return response(json\_encode($this->factory->getErrors()), 500,['X-IC-Remove'=>false]);  }  } |

1. Kode program JadwalController.php

Kode program JadwalController digunakan oleh dosen untuk mengelola jadwal matakuliah. dalam kode program file ini dosen hanya dapat mengedit jadwal saja. File ini memerlukan hak akses sebagai dosen untuk menjalankannya.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers\Dosen;  use Stmik\Factories\JadwalFactory;  use Stmik\Http\Controllers\Controller;  use Stmik\Http\Controllers\GetDataBTTableTrait;  use Stmik\Http\Requests\JadwalRequest;  class JadwalController extends Controller  {  use GetDataBTTableTrait;  /\*\* @var MasterMahasiswaFactory \*/  protected $factory;  public function \_\_construct(JadwalFactory $factory)  {  $this->factory = $factory;  $this->middleware('auth.role:dosen');  }  public function index()  {  return view('dosen.jadwal.index')  ->with('layout', $this->getLayout());  }  public function edit($id)  {  return view('dosen.jadwal.form')  ->with('data', $this->factory->getDataJadwal($id))  ->with('action', route('dosen.jadwal.update', ['id'=>$id]));  }  public function update($id, JadwalRequest $request)  {  $input = $request->all();  if($this->factory->update($id, $input)) {  return $this->edit($id)->with('success', "Data Jadwal telah terupdate!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function store(JadwalRequest $request)  {  $input = $request->all();  if($this->factory->store($input)) {  return $this->create()->with('success', "Data jadwal {$this->factory->getLastInsertId()} telah ditambahkan, silahkan lakukan proses penambahan lainnya!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  } |

1. Kode program MateriController.php

Kode program pada file MateriController.php digunakan untuk bisa melakukan upload materi ke dalam server. File ini memerlukan hak akses dosen untuk menjalankannya.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers\Dosen;  use Stmik\Factories\MateriFactory;  use Stmik\Http\Controllers\Controller;  use Stmik\Http\Controllers\GetDataBTTableTrait;  use Stmik\Http\Requests\MateriRequest;  use Illuminate\Http\UploadedFile;  class MateriController extends Controller  {  use GetDataBTTableTrait;  /\*\* @var MasterMahasiswaFactory \*/  protected $factory;  public function \_\_construct(MateriFactory $factory)  {  $this->factory = $factory;  $this->middleware('auth.role:dosen');  }  public function index()  {  return view('dosen.materi.index')  ->with('layout', $this->getLayout());  }  public function create()  {  return view('dosen.materi.form')  ->with('data', null)  ->with('action', route('dosen.materi.store'));  }  public function store(MateriRequest $request)  {  $input = $request->all();  if($request->hasFile('filename')) {  $file = $request->file('filename');  //getting timestamp  $timestamp = date('Ymd\_His', strtotime('now', time()));  $input['filename'] = $timestamp. '.' .$request->filename->extension();  $file->move(public\_path().'/materi/', $input['filename']);  }  if($this->factory->store($input)) {  return $this->create()->with('success', "Data Materi {$this->factory->getLastInsertId()} telah ditambahkan, silahkan lakukan proses penambahan lainnya!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function edit($id)  {  return view('dosen.materi.form')  ->with('data', $this->factory->getDataMateri($id))  ->with('action', route('dosen.materi.update', ['id'=>$id]));  }  public function update($id, MateriRequest $request)  {  $input = $request->all();  if($this->factory->update($id, $input)) {  return $this->edit($id)->with('success', "Data materi telah terupdate!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function delete($id)  {  if($this->factory->delete($id)) {  return response("", 200,['X-IC-Remove'=>true]);  }  return response(json\_encode($this->factory->getErrors()), 500,['X-IC-Remove'=>false]);  }  } |

1. Kode program TugasController.php

Kode program pada file TugasController.php digunakan untuk mengelola tugas, dosen dapat menambah, mengedit dan menghapus tugas. File ini memerlukan hak akses sebagai dosen untuk menjalankannya.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers\Dosen;  use Stmik\Factories\TugasFactory;  use Stmik\Http\Controllers\Controller;  use Stmik\Http\Controllers\GetDataBTTableTrait;  use Stmik\Http\Requests\TugasRequest;  class TugasController extends Controller  {  use GetDataBTTableTrait;  /\*\* @var MasterMahasiswaFactory \*/  protected $factory;  public function \_\_construct(TugasFactory $factory)  {  $this->factory = $factory;  $this->middleware('auth.role:dosen');  }  public function index()  {  return view('dosen.tugas.index')  ->with('layout', $this->getLayout());  }  public function create()  {  return view('dosen.tugas.form')  ->with('data', null)  ->with('action', route('dosen.tugas.store'));  }  public function edit($id)  {  return view('dosen.tugas.form')  ->with('data', $this->factory->getDataTugas($id))  ->with('action', route('dosen.tugas.update', ['id'=>$id]));  }  public function update($id, TugasRequest $request)  {  $input = $request->all();  if($this->factory->update($id, $input)) {  return $this->edit($id)->with('success', "Data Tugas telah terupdate!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function store(TugasRequest $request)  {  $input = $request->all();  if($this->factory->store($input)) {  return $this->create()->with('success', "Data Tugas {$this->factory->getLastInsertId()} telah ditambahkan, silahkan lakukan proses penambahan lainnya!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function delete($id)  {  if($this->factory->delete($id)) {  return response("", 200,['X-IC-Remove'=>true]);  }  return response(json\_encode($this->factory->getErrors()), 500,['X-IC-Remove'=>false]);  }  } |

1. Kode program PengumumanController.php

Kode program pada file PengumumanController.php digunakan untuk mengelola data pengumuman, dosen dapat menambah dan menghapus pengumuman. File ini memerlukan hak akses dosen untuk menjalankannya.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers\Dosen;  use Stmik\Factories\PengumumanFactory;  use Stmik\Http\Controllers\Controller;  use Stmik\Http\Controllers\GetDataBTTableTrait;  use Stmik\Http\Requests\PengumumanRequest;  class PengumumanController extends Controller  {  use GetDataBTTableTrait;  /\*\* @var MasterMahasiswaFactory \*/  protected $factory;  public function \_\_construct(PengumumanFactory $factory)  {  $this->factory = $factory;  $this->middleware('auth.role:dosen');  }  public function index()  {  return view('dosen.pengumuman.index')  ->with('layout', $this->getLayout());  }  public function create()  {  return view('dosen.pengumuman.form')  ->with('data', null)  ->with('action', route('dosen.pengumuman.store'));  }  public function edit($id)  {  return view('dosen.pengumuman.form')  ->with('data', $this->factory->getDataPengumuman($id))  ->with('action', route('dosen.pengumuman.update', ['id'=>$id]));  }  public function update($id, PengumumanRequest $request)  {  $input = $request->all();  if($this->factory->update($id, $input)) {  return $this->edit($id)->with('success', "Data Jadwal telah terupdate!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function store(PengumumanRequest $request)  {  $input = $request->all();  if($this->factory->store($input)) {  return $this->create()->with('success', "Data NIM {$this->factory->getLastInsertId()} telah ditambahkan, silahkan lakukan proses penambahan lainnya!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function delete($id)  {  if($this->factory->delete($id)) {  return response("", 200,['X-IC-Remove'=>true]);  }  return response(json\_encode($this->factory->getErrors()), 500,['X-IC-Remove'=>false]);  }  } |

1. Kode program MahasiswaApiController.php

Kode program pada file MahasiswaApiController.php berfungsi untuk *Application Programming Interface* website. File ini menangani permintaan data dari aplikasi client, setelah itu data akan dikirim kepada aplikasi client sesuai dengan permintaan client.

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers;  use Illuminate\Http\Request;  use Stmik\Http\Requests\PengumumanRequest;  use Stmik\Mahasiswa;  use Stmik\Jadwal;  use Stmik\RincianStudi;  use Stmik\RencanaStudi;  use Stmik\Tugas;  use Stmik\Pengumuman;  use Stmik\Materi;  class MahasiswaApiController extends Controller  {  public function jadwal($nim)  {  $jadwal = RincianStudi::join('pengampu\_kelas', 'rincian\_studi.kelas\_diambil\_id','=','pengampu\_kelas.id')  ->join('mata\_kuliah','pengampu\_kelas.mata\_kuliah\_id',  '=','mata\_kuliah.id')  ->join('dosen','pengampu\_kelas.dosen\_id','=','dosen.nomor\_induk')  >join('jadwal','rincian\_studi.kelas\_diambil\_id','=','jadwal.pengampu\_id')  ->join('ruangans','jadwal.ruangan\_id','=','ruangans.id')  ->join('rencana\_studi','rincian\_studi.rencana\_studi\_id',  '=','rencana\_studi.id')  ->select(['mata\_kuliah.nama as matakuliah','jadwal.hari','jadwal.jam\_masuk','jadwal.jam\_keluar','ruangans.ruang','dosen.nama as dosen'])  ->where('mahasiswa\_id','=',$nim)  ->paginate(20);  return Response()->json($jadwal, 200);  }  public function tugas($nim)  {  $tugas = Tugas::join('pengampu\_kelas','tugas.pengampu\_id', '=','pengampu\_kelas.id')  ->join('mata\_kuliah','pengampu\_kelas.mata\_kuliah\_id', '=','mata\_kuliah.id')  ->join('dosen','pengampu\_kelas.dosen\_id','=','dosen.nomor\_induk')  ->join('rincian\_studi','pengampu\_kelas.id', '=','rincian\_studi.kelas\_diambil\_id')  ->join('rencana\_studi','rincian\_studi.rencana\_studi\_id', '=','rencana\_studi.id')  ->select(['tugas.nama\_tugas','tugas.deadline','tugas.keterangan', 'mata\_kuliah.nama as matakuliah','dosen.nama as dosen'])  ->where('mahasiswa\_id','=',$nim)  ->paginate(20);  if($tugas){  return Response()->json($tugas, 200);  } else {  return response()->json(array("message" => "nim tidak ditemukan"), 403);  }  }  public function materi($nim)  {  $materi = Materi::join('pengampu\_kelas','materis.pengampu\_id',  '=','pengampu\_kelas.id')  ->join('mata\_kuliah','pengampu\_kelas.mata\_kuliah\_id',  '=','mata\_kuliah.id')  ->join('dosen','pengampu\_kelas.dosen\_id','=','dosen.nomor\_induk')  ->join('rincian\_studi','pengampu\_kelas.id',  '=','rincian\_studi.kelas\_diambil\_id')  ->join('rencana\_studi','rincian\_studi.rencana\_studi\_id',  '=','rencana\_studi.id')  ->select(['materis.id','materis.nama\_materi',  'materis.filename','mata\_kuliah.nama as matakuliah','dosen.nama as dosen'])  ->where('mahasiswa\_id','=',$nim)  ->paginate(20);  return Response()->json($materi, 200);  }  public function nilai($nim)  {  $nilai = RencanaStudi::join('rincian\_studi as r','rencana\_studi.id','=','r.rencana\_studi\_id')  ->join('pengampu\_kelas as p','r.kelas\_diambil\_id','=','p.id')  ->join('mata\_kuliah','p.mata\_kuliah\_id','=','mata\_kuliah.id')  ->join('dosen','p.dosen\_id','=','dosen.nomor\_induk')  ->join('mata\_kuliah as m','p.mata\_kuliah\_id','=','m.id')  ->select(['m.nama as matakuliah','r.nilai\_tugas','r.nilai\_praktikum','r.nilai\_uts','r.nilai\_uas','r.nilai\_akhir','r.status\_lulus','r.nilai\_huruf','dosen.nama as dosen'])  ->where('mahasiswa\_id','=',$nim)  ->paginate(20);  return Response()->json($nilai, 200);  }  public function pengumuman($nim)  {  $info = RencanaStudi::join('rincian\_studi as r','rencana\_studi.id','=','r.rencana\_studi\_id')  ->join('pengampu\_kelas as p','r.kelas\_diambil\_id','=','p.id')  ->join('pengumuman','p.dosen\_id','=','pengumuman.user\_id')  ->join('dosen as d','pengumuman.user\_id','=','d.nomor\_induk')  ->select(['d.nama as dosen','pengumuman.perihal','pengumuman.keterangan'])  ->where('mahasiswa\_id','=',$nim)  ->paginate(20);  return Response()->json($info, 200);  }  public function download($filename)  {  $file= public\_path(). "/materi/".$filename;  $headers = array(  'Content-Type: application/zip',  );  return Response()->download($file);  }  } |

1. Kode program app.js

Kode program pada file app.js berfungsi untuk mengatur tampilan halaman aplikasi client. isi kode program file app.js dapat dilihat sebagai berikut :

|  |
| --- |
| angular.module('starter', ['ionic', 'starter.controllers','starter.services','ngCordova'])  .run(function($ionicPlatform) {  $ionicPlatform.ready(function() {  if (window.cordova && window.cordova.plugins.Keyboard) {  cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);  cordova.plugins.Keyboard.disableScroll(true);  }  if (window.StatusBar) {  // org.apache.cordova.statusbar required  StatusBar.styleDefault();  }  });  })  .config(function($stateProvider, $urlRouterProvider) {  $stateProvider  .state('login', {  url: '/login',  templateUrl: 'templates/login.html',  controller: 'LoginCtrl'  }  )  .state('app', {  url: '/app',  abstract: true,  templateUrl: 'templates/menu.html',  controller: 'AppCtrl'  })  .state('app.materi', {  url: '/materi/:nim/:token',  views: {  'menuContent': {  templateUrl: 'templates/materi.html',  controller: 'MateriCtrl'  }  }  })  .state('app.info', {  url: '/info/:nim/:token',  views: {  'menuContent': {  templateUrl: 'templates/informasi.html',  controller: 'InfoCtrl'  }  }  })  .state('app.jadwal', {  url: '/jadwal/:nim/:token',  views: {  'menuContent': {  templateUrl: 'templates/jadwal.html',  controller: 'JadwalCtrl'  }  }  })  .state('app.tugas', {  url: '/tugas/:nim/:token',  views: {  'menuContent': {  templateUrl: 'templates/tugas.html',  controller: 'TugasCtrl'  }  }  })  .state('app.nilai', {  url: '/nilai/:nim/:token',  views: {  'menuContent': {  templateUrl: 'templates/nilai.html',  controller: 'NilaiCtrl'  }  }  });  $urlRouterProvider.otherwise('/login');  }); |

1. Kode program controller.js

Kode program pada file controller.js berfungsi untuk mengatur data – data dalam aplikasi client. File controller.js dapat mengirim permintaan kepada server dan menampilkan data yang diterima pada halaman aplikasi. Berikut isi kode program controller.js :

|  |
| --- |
| angular.module('starter.controllers', [])  .controller('AppCtrl', function($scope,$rootScope, $state, $ionicModal, $timeout) {  $scope.logout = function(){  $rootScope.token == null;  $state.go("login");  }  })  .controller('LoginCtrl', function($scope, $rootScope, $stateParams, $state, $http, $ionicPopup, Service) {  if($rootScope.token != null){  $state.go("app.info",{nim:$rootScope.nim, token: $rootScope.token});  }  $scope.user = {};  $scope.login = function(){  Service.loadingShow();  $http.get(  Service.baseUrl()+'login',  {  params : {  name: $scope.user.username,  password: $scope.user.password  }  }  ).success(function(data){  Service.loadingHide();  $rootScope.token = data.token;  $rootScope.nim = data.user.name;  Service.setToken(data.token);  Service.setNim(data.user.name);  $state.go("app.info",{nim:$rootScope.nim, token: $rootScope.token});  }).error(function(data){  Service.loadingHide();  if(data != null){  var alertPopup = $ionicPopup.alert({  title: 'Login failed!',  template: data.message  });  }else{  var alertPopup = $ionicPopup.alert({  title: 'Login failed!',  template: 'gagal tersambung ke server check koneksi internet anda'  });  }  });  };  })  .controller('JadwalCtrl', function($scope,$rootScope,$state,$http,$stateParams, $ionicLoading,$ionicPopup, Service) {  if($rootScope.token == null){  $state.go("login");  }  $scope.getJadwal = function(){  Service.loadingShow();  $http.get(  Service.baseUrl()+'jadwal/'+$rootScope.nim+'/'+$rootScope.token  ).success(function(data){  $scope.model = [];  angular.forEach(data.data, function(value, key) {  $scope.model.push(value);  })  }).error(function(data){  var alertPopup = $ionicPopup.alert({  title: 'Login failed!',  template: 'gagal tersambung ke server check koneksi internet anda'  });  }).finally(function() {  Service.loadingHide();  $scope.$broadcast('scroll.refreshComplete');  });  };  $scope.getJadwal();  })  .controller('NilaiCtrl', function($scope,$rootScope,$state,$http,$stateParams, $ionicLoading,$ionicPopup, Service) {  if($rootScope.token == null){  $state.go("login");  }  $scope.getNilai = function(){  Service.loadingShow();  $http.get(  Service.baseUrl()+'nilai/'+$rootScope.nim+'/'+$rootScope.token  ).success(function(data){  $scope.nilai= [];  angular.forEach(data.data, function(value, key) {  $scope.nilai.push(value);  })  }).error(function(){  var alertPopup = $ionicPopup.alert({  title: 'Login failed!',  template: 'gagal tersambung ke server check koneksi internet anda'  });  }).finally(function() {  Service.loadingHide();  $scope.$broadcast('scroll.refreshComplete');  });  };  $scope.getNilai();  })  .controller('InfoCtrl', function($scope,$rootScope,$state, $http,$stateParams, $ionicLoading,$ionicPopup, Service) {  if($rootScope.token == null){  $state.go("login");  }  $scope.getInfo = function(){  Service.loadingShow();  $http.get(  Service.baseUrl()+'pengumuman/'+$rootScope.nim+'/' +$rootScope.token  ).success(function(data){  $scope.informasi= [];  angular.forEach(data.data, function(value, key) {  $scope.informasi.push(value);  })  }).error(function(){  var alertPopup = $ionicPopup.alert({  title: 'Login failed!',  template: 'gagal tersambung ke server check koneksi internet anda'});  }).finally(function() {  //stop loading...  Service.loadingHide();  // Stop the ion-refresher from spinning  $scope.$broadcast('scroll.refreshComplete');  });  };  $scope.getInfo();  })  .controller('MateriCtrl', function($cordovaFileTransfer,$scope,$rootScope,$state,$http, $stateParams,$ionicLoading,$ionicPopup,Service) {  if($rootScope.token == null){  $state.go("login");  }  $scope.getMateri = function(){  Service.loadingShow();  $http.get( Service.baseUrl()+'materi/'+$rootScope.nim+'/'+$rootScope.token  ).success(function(data){  $scope.materis= [];  angular.forEach(data.data, function(value, key) {  $scope.materis.push(value);  })  }).error(function(){  var alertPopup = $ionicPopup.alert({  title: 'Login failed!',  template: 'gagal tersambung ke server check koneksi internet anda' });  }).finally(function() {  Service.loadingHide();  $scope.$broadcast('scroll.refreshComplete');  });  };  $scope.getMateri();  $scope.Download = function (filename) {  Service.loadingShow();  ionic.Platform.ready(function(){  var url = "http://matkul.esy.es/public/api/download/";  var targetPath = cordova.file.externalRootDirectory + filename;    $cordovaFileTransfer.download(url+filename, targetPath, {}, true).then(function (result) {  Service.loadingHide(); $scope.$broadcast('scroll.refreshComplete');  $scope.hasil = 'Save file on '+targetPath+' success!';  var alertPopup = $ionicPopup.alert({  title: 'Success',  template: 'berhasil download, Save file on '+targetPath})  }, function (error) {  Service.loadingHide(); $scope.$broadcast('scroll.refreshComplete');  var alertPopup = $ionicPopup.alert({  title: 'Alert!',  template: 'gagal download'  });  alert(error);  }, function (progress) {  $scope.downloadProgress = (progress.loaded / progress.total) \* 100;  });  });  }  })  .controller('TugasCtrl', function($scope, $http, $rootScope, $state, $stateParams, $ionicLoading,$ionicPopup, Service) {  if($rootScope.token == null){  $state.go("login");  }  $scope.getTugas = function(){  Service.loadingShow();  $http.get( Service.baseUrl()+'tugas/'+$rootScope.nim+'/'+$rootScope.token  ).success(function(data){  $scope.tugas = [];  angular.forEach(data.data, function(value, key) {  $scope.tugas.push(value);  });  Service.loadingHide();  }).error(function(data){  var alertPopup = $ionicPopup.alert({  title: 'Error!',  template: 'gagal tersambung ke server check koneksi internet anda'  });  }).finally(function() {  Service.loadingHide();  $scope.$broadcast('scroll.refreshComplete');  });  };  $scope.getTugas();  }) |

1. Pembahasan database
2. Pembahasan Hasil Responden Pengguna

Penilain terhadap Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya, penulis melakukan penyebaran kuisioner yang diberikan kepada 10 orang. Selanjutnya mereka menggunakan Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya kemudian dilakukan pengumpulan data dengan mengisi kuesioner yang telah dibagikan sebelumnya oleh penulis.

1. Pernyataan kuesioner

Pernyataan kuesioner merupakan tahap untuk mengetahui pernyataan apa saja yang diajukan kepada responden terhadap Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya. Berikut pernyataan kuesioner seperti pada Tabel 19.

Tabel 19. Pernyataan Kuesioner

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Pernyataan** | **Alternatif Jawaban** | | | | |
| **5** | **4** | **3** | **2** | **1** |
| **SB** | **B** | **CB** | **KB** | **TB** |
| 1. | Tampilan aplikasi sudah bersifat baik / *user friendly.* |  |  |  |  |  |
| 2. | Fungsional aplikasi sudah berjalan sesuai dengan fungsinya. |  |  |  |  |  |
| 3. | Aplikasi mampu melakukan *download file* materi. |  |  |  |  |  |
| 4. | informasi yang ditampilkan oleh aplikasi sesuai dengan data yang dibutuhkan mahasiswa. |  |  |  |  |  |
| 5. | Aplikasi terlihat menarik, baik dari segi tampilan dan teks. |  |  |  |  |  |
| 6. | Aplikasi ini dapat membantu ketika mencari informasi tentang matakuliah. |  |  |  |  |  |
| 7. | Tentang keseluruhan aplikasi. |  |  |  |  |  |

Keterangan bobot yang digunakan dalam penilaian alternatif jawaban dapat dilihat pada Tabel 20.

Tabel 20. Keterangan Bobot

|  |  |
| --- | --- |
| **Keterangan** | **Nilai** |
| Sangat Baik | 5 |
| Baik | 4 |
| Cukup Baik | 3 |
| Kurang Baik | 2 |
| Tidak Baik | 1 |

1. Hasil Kuesioner

Hasil kuesioner merupakan hasil perhitungan dari penilaian responden terhadap Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya. Hasil kuesioner dapat dilihat pada Tabel 21.

Tabel 21. Hasil kuesioner

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Responden** | **Jawaban Pertanyaan** | | | | | | | **Jumlah** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |
| **Jumlah Skor Hasil Pengumpulan Data** | | | | | | | |  |

Sedangkan untuk bobot daftar pertanyaan dapat dilihat pada Tabel 22.

Tabel 22. Bobot daftar pertanyaan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Keterangan** | **Daftar Pertanyaan** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Sangat Baik |  |  |  |  |  |  |  |
| Baik |  |  |  |  |  |  |  |
| Cukup Baik |  |  |  |  |  |  |  |
| Kurang Baik |  |  |  |  |  |  |  |
| Tidak Baik |  |  |  |  |  |  |  |

Berdasarkan hasil kuesioner pada tabel dapat dilihat menggunakan skala pengukuran, disini penulis menggunakan likert sebagai skala pengukuran yaitu sebagai berikut:

Jumlah skor kriteria = (skor tertinggi tiap item) x (jumlah pertanyaan) x (jumlah responden)

Jumlah skor kriteria = 5 x 7 x 10

= 350

Berdasarkan tabel untuk total jawaban responden adalah .... dan jumlah skor kriterium adalah 350. Dengan demikian Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya menurut pendapat 10 orang responden yaitu:

Total hasil responden = .... / 350 x 100

= .... x 100

= .... %

Jadi, sikap responden terhadap aplikasi yang dibangun sebesar 91 % dan termasuk dalam interval sangat baik.

0 20% 40% 60% 80% 100%

TB KB CB B SB

Gambar 44. Persentase *range* kelompok responden

Keterangan :

TB = Tidak Baik

KB = Kurang Baik

CB = Cukup Baik

B = Baik

SB = Sangat Baik

Berdasarkan hasil perhitungan kuesioner dengan skala *likert* yang didapat dari 10 respon dan 7 pertanyaan maka didapat hasil interpretasi sebesar ....% atau dengan nilai .... dari nilai maksimal 350, maka dapat disimpulkan bahwa Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya sangat baik dan layak untuk digunakan.