**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.

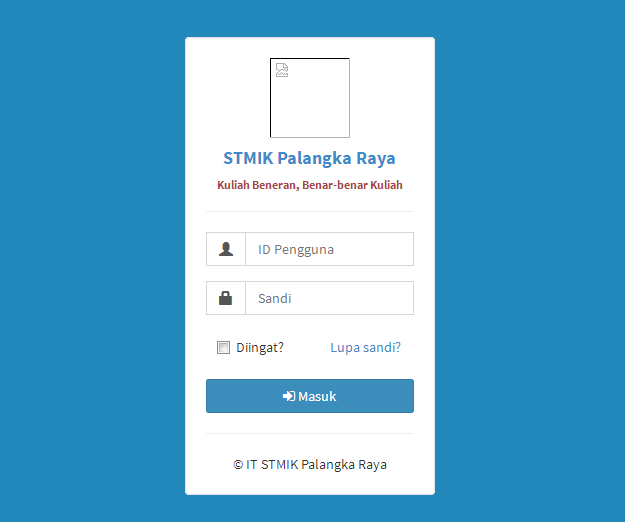
1. Pengujian *login* pada *website server*

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

Tabel 8. Hasil Pengujian *login*

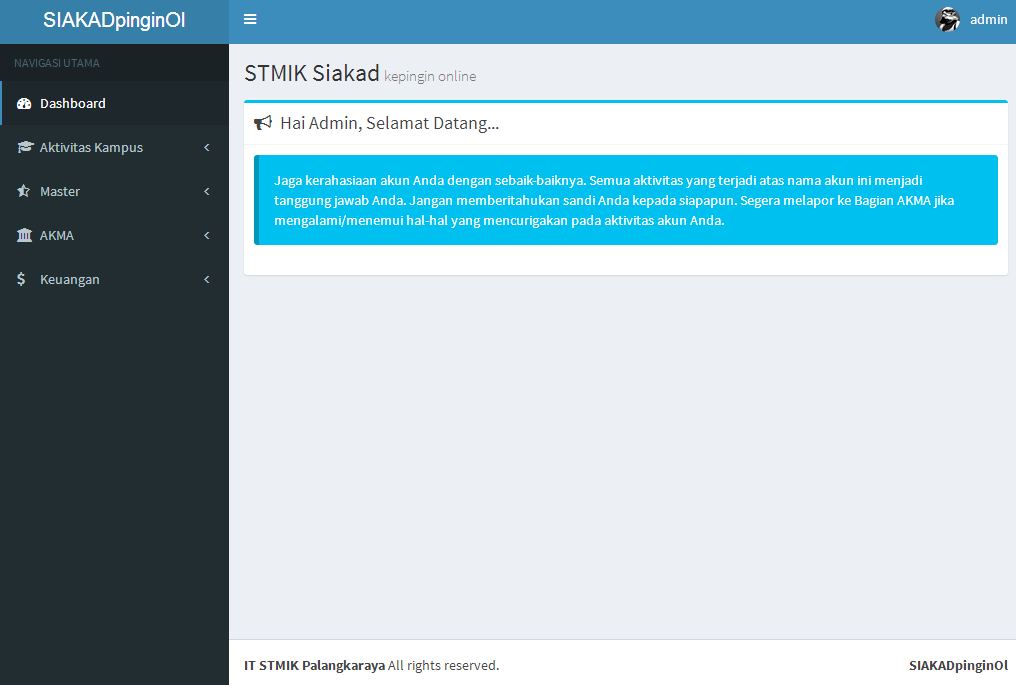
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian *login* admin | Pengguna memasukkan *username* dan *password* kemudian menekan tombol login | Website akan menampilkan halaman *dashboard* untuk admin | Halaman *dashboard* admin tampil setelah tombol login ditekan | Sesuai |

Adapun tampilan dari *implementasi* *login* pada *website server* adalah sebagai berikut :



Gambar 49. *form login* pada *website server*

Dalam pengujian ini akan tampil halaman *dashboard* setelah meng-*input* *username* dan *password* dengan benar. Berikut tampilan dashboard setelah proses login selesai:



Gambar 50. Halaman dashboard

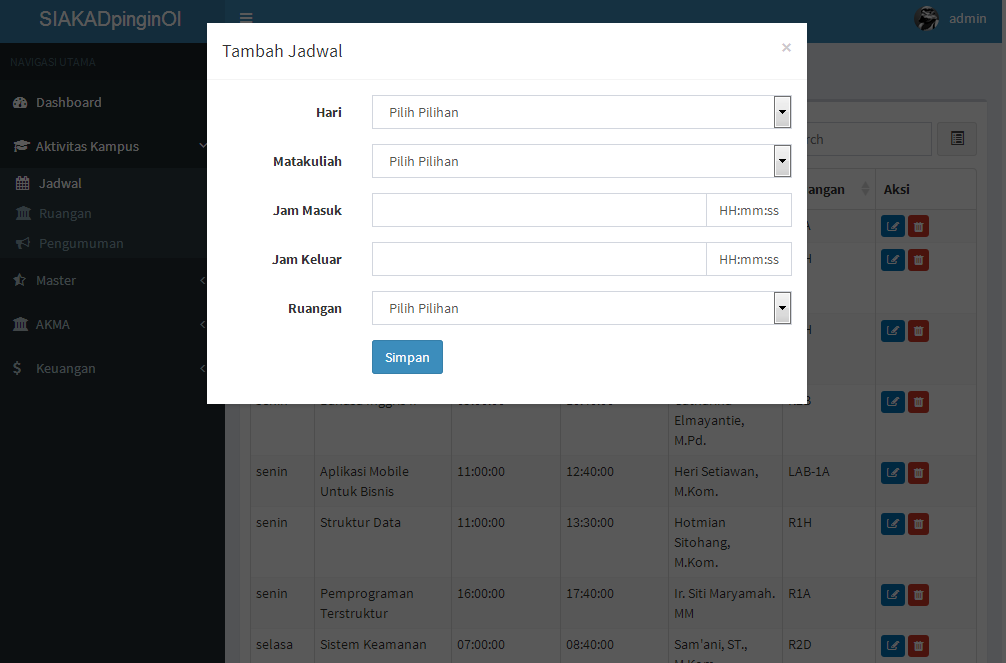
1. Pengujian tambah jadwal

Hasil pengujian ini dilakukan untuk mengetahui fungsi tambah jadwal telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 9. Hasil pengujian tambah jadwal

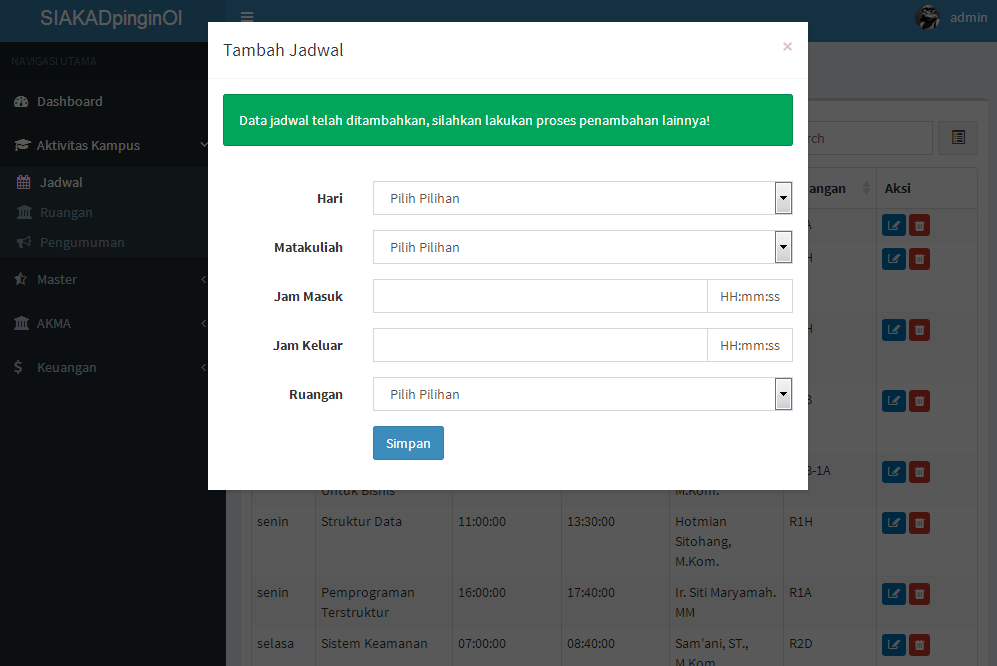
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian tambah jadwal | Admin memasukan data jadwal pada *form* tambah jadwal | Data jadwal bertambah | Muncul pesan “jadwal telah ditambahkan” dan data jadwal telah bertambah | Sesuai |

Adapun tampilan dari *implementasi* tambah jadwalpada *website server* adalah sebagai berikut :



Gambar 51. form tambah jadwal

Dalam pengujian ini akan bertambah data jadwal setelah admin meng-input data pada form tambah jadwal. Berikut tampilan setelah admin berhasil menambahkan data jadwal :



Gambar 1. Tampilan setelah berhasil tambah data

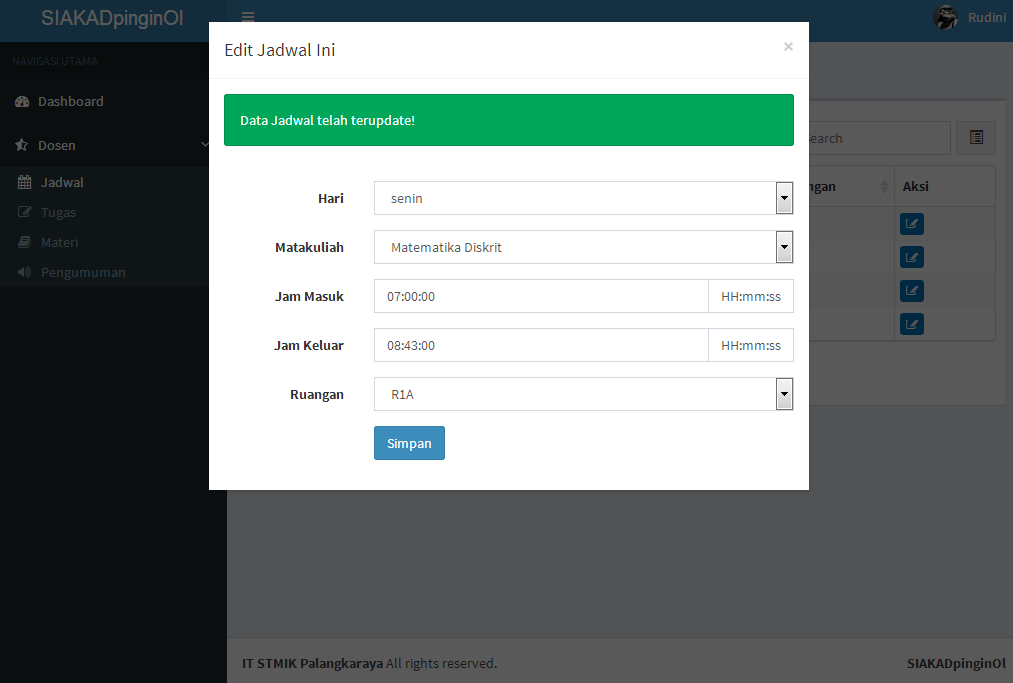
1. Pengujian edit jadwal

Hasil pengujian ini dilakukan untuk mengetahui apakah fungsi edit jadwal sudah berjalan dengan baik atau tidak. Berdasarkan hasil pengujian yang dilakukan, fitur edit jadwal sudah berjalan dengan baik. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel 8.

Tabel 9. Hasil Pengujian edit jadwal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian edit jadwal | Dosen memilih data jadwal yang hendak dirubah dan mengubah data jadwal pada form edit. | Data jadwal berubah sesuai dengan perubahan yang dilakukan dosen | Muncul pesan “jadwal telah ter-*update*” pada form edit jadwal dan data jadwal telah berubah | Sesuai |

Dalam pengujian ini data jadwal akan berubah setelah dosen meng-*edit* data pada form edit jadwal. Berikut tampilan saat dosen mengubah data jadwal pada form edit jadwal :



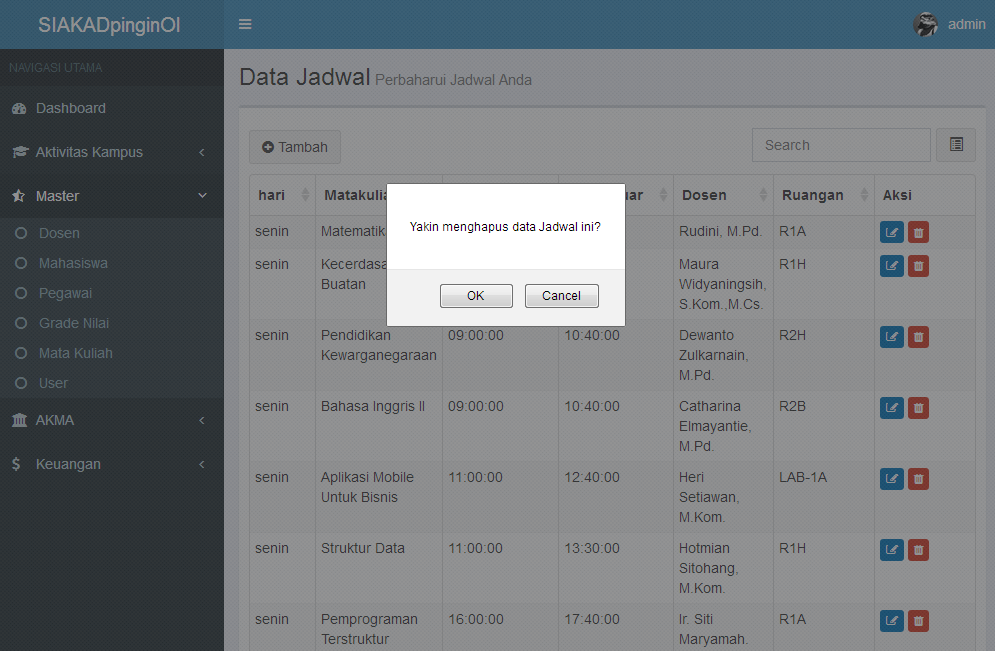
1. Pengujian hapus data jadwal

Hasil pengujian ini dilakukan untuk mengetahui fungsi hapus jadwal telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 10. Hasil pengujian hapus jadwal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian hapus data jadwal | Admin menekan tombol hapus | Data jadwal terhapus | Data jadwal terhapus setelah menekan tombol yes pada kotak dialog peringatan | Sesuai |

Dalam pengujian ini admin menekan tombol hapus, kemudian akan muncul kotak dialog untuk memvalidasi apakah admin memang ingin menghapus jadwal, apabila iya maka tekan tombol yes pada kotak dialog dan data jadwal akan terhapus. Adapun tampilan dari *implementasi* hapus jadwalpada *website server* adalah sebagai berikut :



Gambar 2. implementasi hapus jadwal

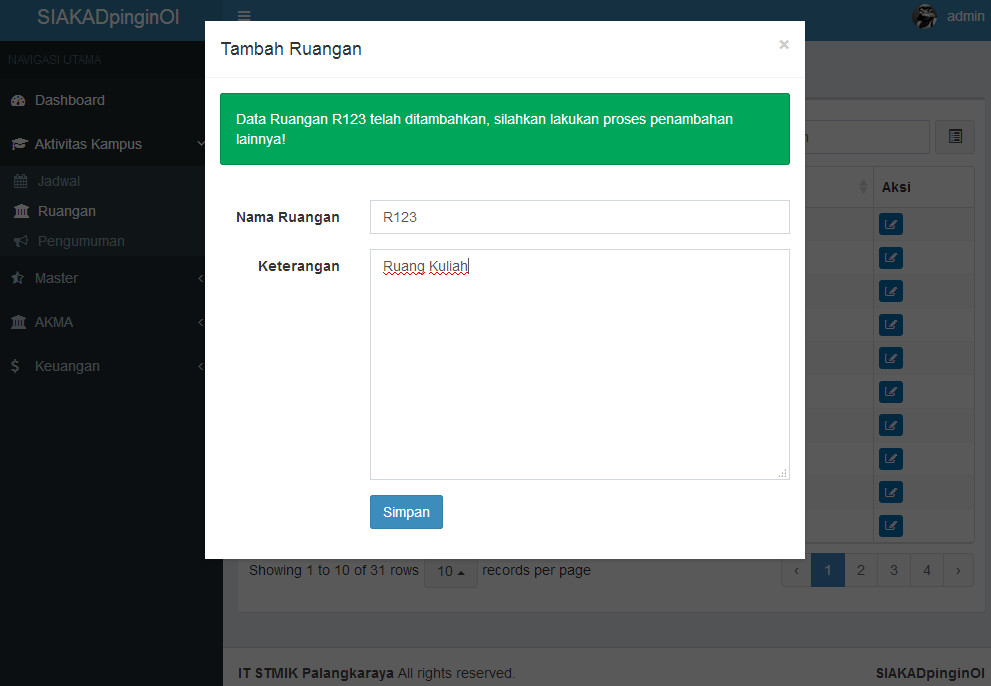
1. Pengujian tambah ruangan

Hasil pengujian ini dilakukan untuk mengetahui fungsi tambah ruangan telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 11. Hasil pengujian tambah ruangan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian tambah ruangan | Admin menekan tombol tambah ruangan | Data ruangan bertambah | Muncul pesan “ruangan telah ditambahkan” dan data ruangan telah bertambah pada tabel daftar ruangan | Sesuai |

Dalam pengujian ini data ruangan akan bertambah setelah admin meng-input data pada form tambah ruangan. Adapun tampilan dari *implementasi* tambah ruangan adalah sebagai berikut :

Gambar . implementasi tambah ruangan

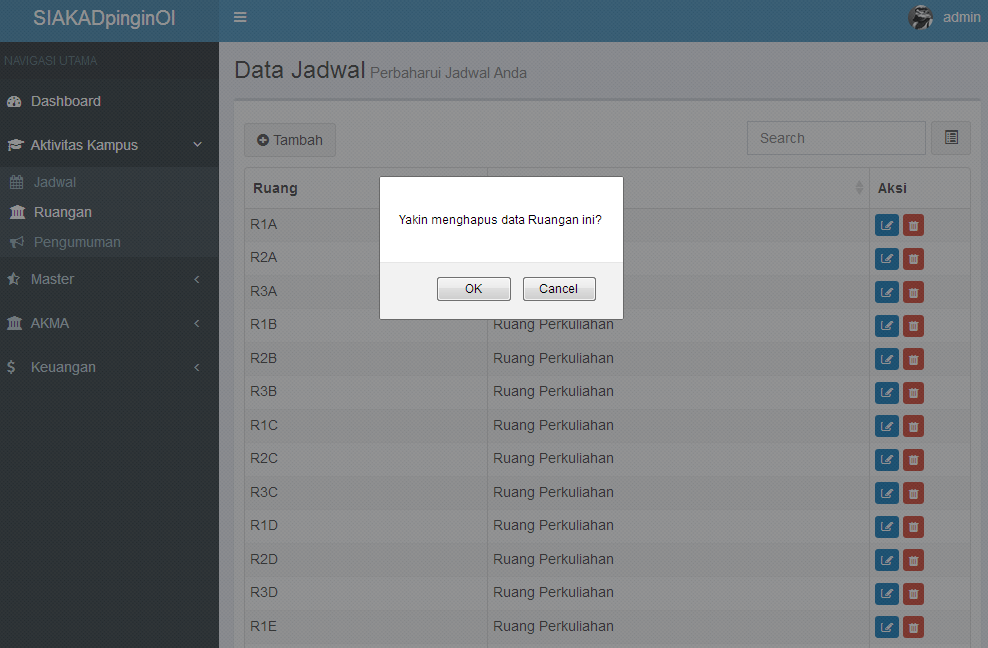
1. Pengujian hapus ruangan

Hasil pengujian ini dilakukan untuk mengetahui fungsi hapus jadwal telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 10. Hasil pengujian hapus jadwal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian hapus data jadwal | Admin menekan tombol hapus | Data jadwal terhapus | Data jadwal terhapus setelah menekan tombol yes pada kotak dialog peringatan | Sesuai |

Dalam pengujian ini admin menekan tombol hapus, kemudian akan muncul kotak dialog untuk memvalidasi apakah admin memang ingin menghapus ruangan, apabila iya maka tekan tombol yes pada kotak dialog dan data ruangan akan terhapus. Adapun tampilan dari *implementasi* hapus ruangan adalah sebagai berikut :

Gambar . implementasi hapus ruangan

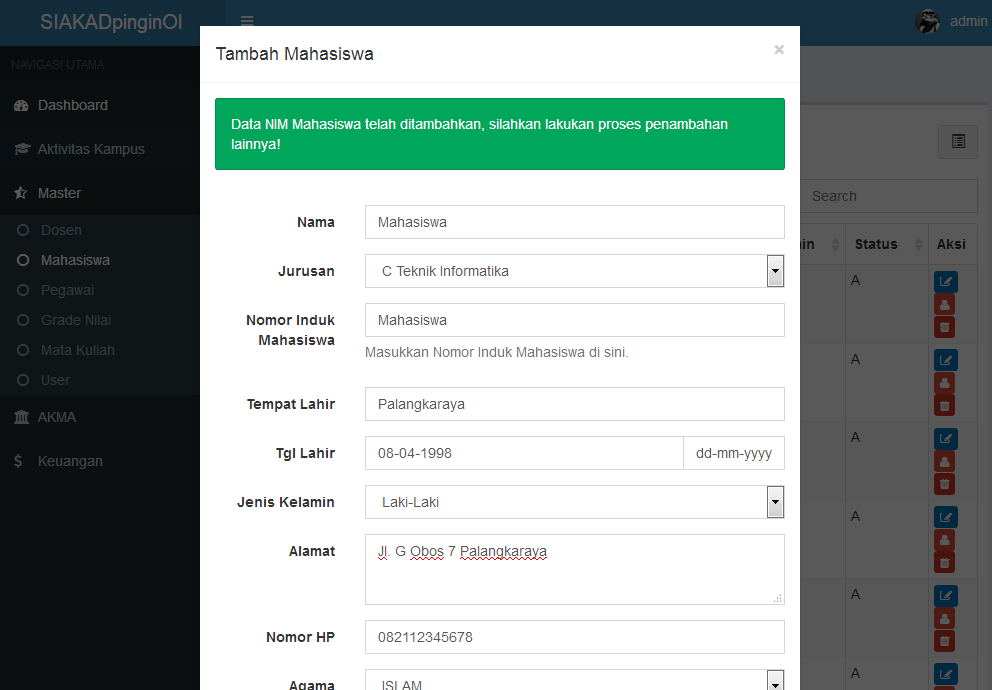
1. Pengujian tambah mahasiswa

Hasil pengujian ini dilakukan untuk mengetahui fungsi tambah mahasiswa telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 11. Hasil pengujian tambah mahasiswa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian tambah mahasiswa | Admin meng-input data tugas pada form tambah mahasiswa | Data mahasiswa bertambah | Muncul pesan “mahasiswa telah ditambahkan” dan data mahasiswa telah bertambah pada tabel daftar mahasiswa | Sesuai |

Dalam pengujian ini data mahasiswa akan bertambah setelah admin meng-input data pada form tambah mahasiswa. Adapun tampilan dari *implementasi* tambah matakuliah adalah sebagai berikut :



Gambar . implementasi tambah mahasiswa

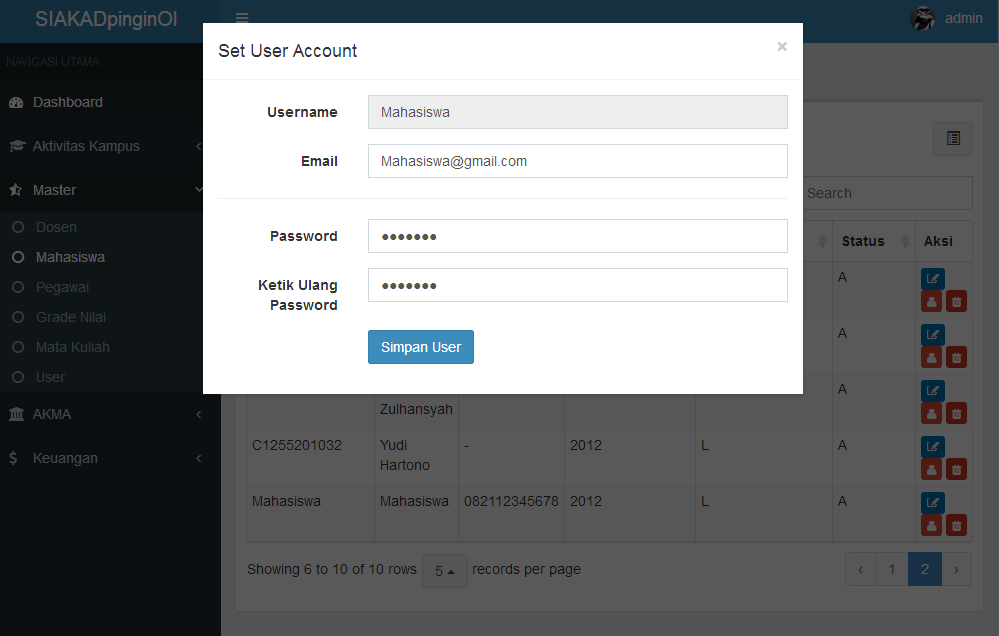
1. Pengujian *Set User Account*

Hasil pengujian ini dilakukan untuk mengetahui fungsi set user account telah berjalan dengan baik dan benar. Set user account adalah menu untuk menambahkan atau mengelola username dan password mahasiswa serta dosen. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

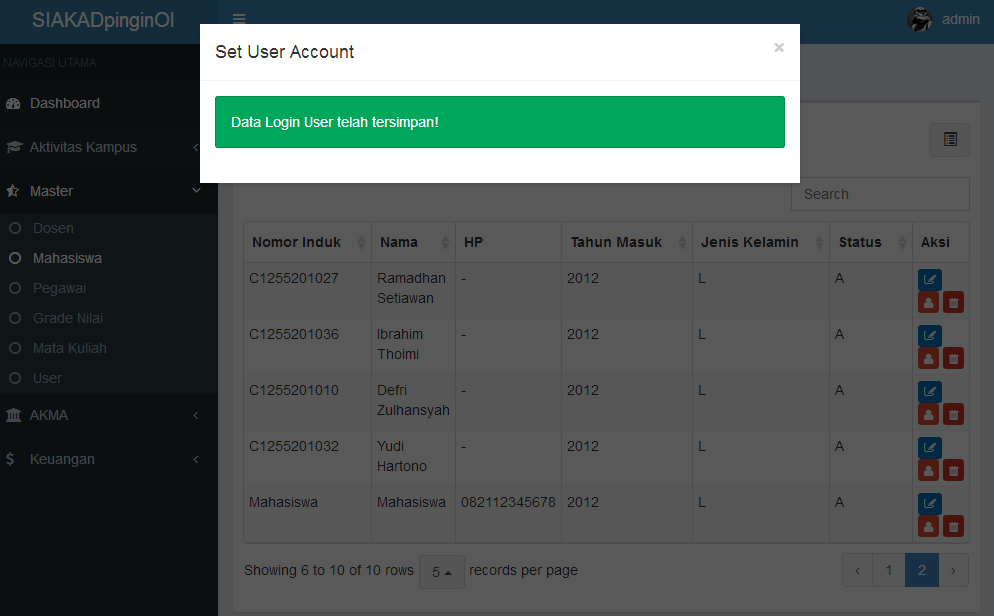
Tabel 11. Hasil pengujian *set user account*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian *set user account* | Admin meng-input data username dan password pada form *set user account* | Data account user bertambah atau berubah | Muncul pesan data login user telah tersimpan dan data *user account* telah bertambah | Sesuai |

Dalam pengujian ini data pada tabel *user* akan bertambah atau berubah setelah admin meng-input data pada form *set user account*. Adapun tampilan dari *implementasi* *set user account* adalah sebagai berikut :



Gambar . implementasi set user account



Gambar . pesan setelah set user account berhasil

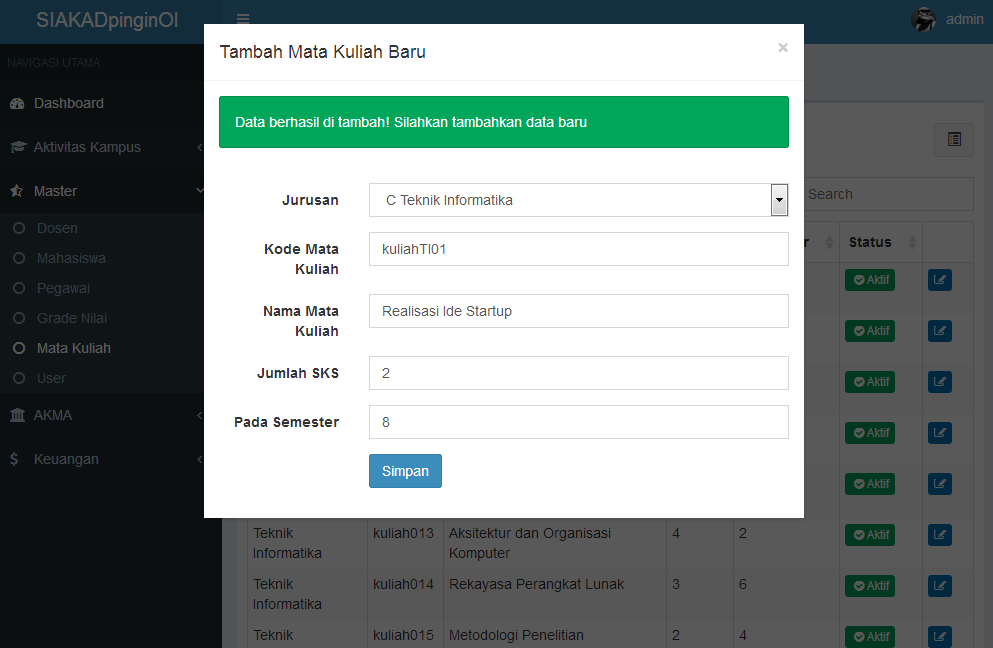
1. Pengujian tambah matakuliah

Hasil pengujian ini dilakukan untuk mengetahui fungsi tambah matakuliah telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 11. Hasil pengujian tambah tugas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian tambah matakuliah | Admin meng-input data tugas pada form tambah matakuliah | Data matakuliah bertambah | Muncul pesan “matakuliah telah ditambahkan” dan data matakuliah telah bertambah pada tabel daftar matakuliah | Sesuai |

Dalam pengujian ini data matakuliah akan bertambah setelah admin meng-input data pada form tambah matakuliah. Adapun tampilan dari *implementasi* tambah matakuliah adalah sebagai berikut :



Gambar 8. implementasi tambah matakuliah

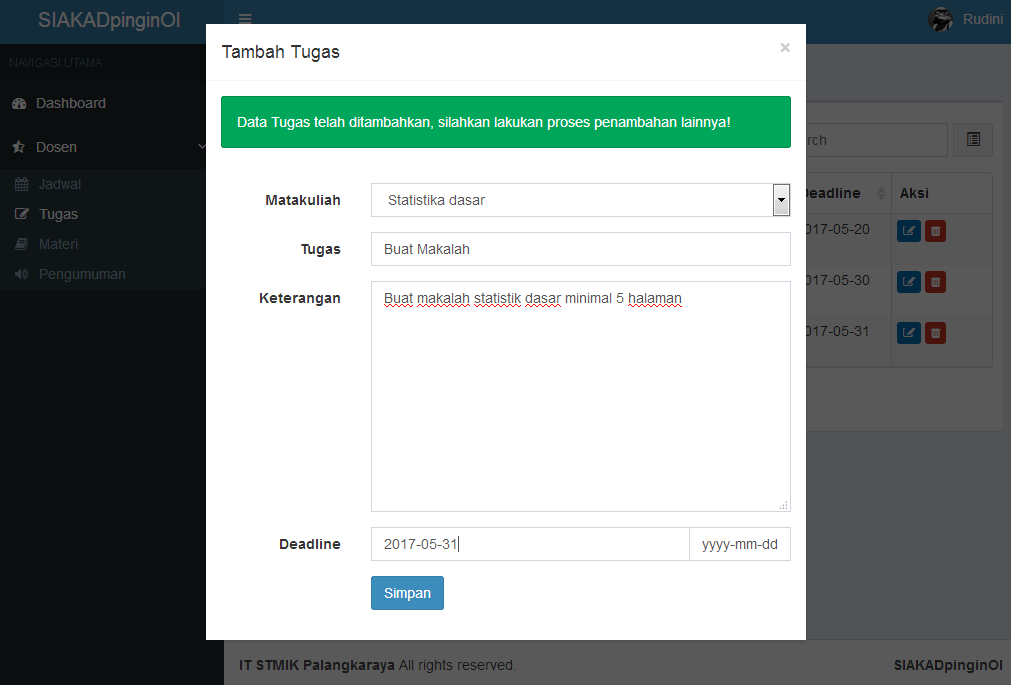
1. Pengujian tambah tugas

Hasil pengujian ini dilakukan untuk mengetahui fungsi tambah tugas telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 11. Hasil pengujian tambah tugas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian tambah tugas | Dosen meng-input data tugas pada form tambah tugas | Data tugas bertambah | Muncul pesan “tugas telah ditambahkan” dan data tugas telah bertambah pada tabel daftar tugas | Sesuai |

Dalam pengujian ini data tugas akan bertambah setelah dosen meng-input data pada form tambah tugas. Adapun tampilan dari *implementasi* tambah tugas adalah sebagai berikut :

Gambar . implementasi tambah tugas

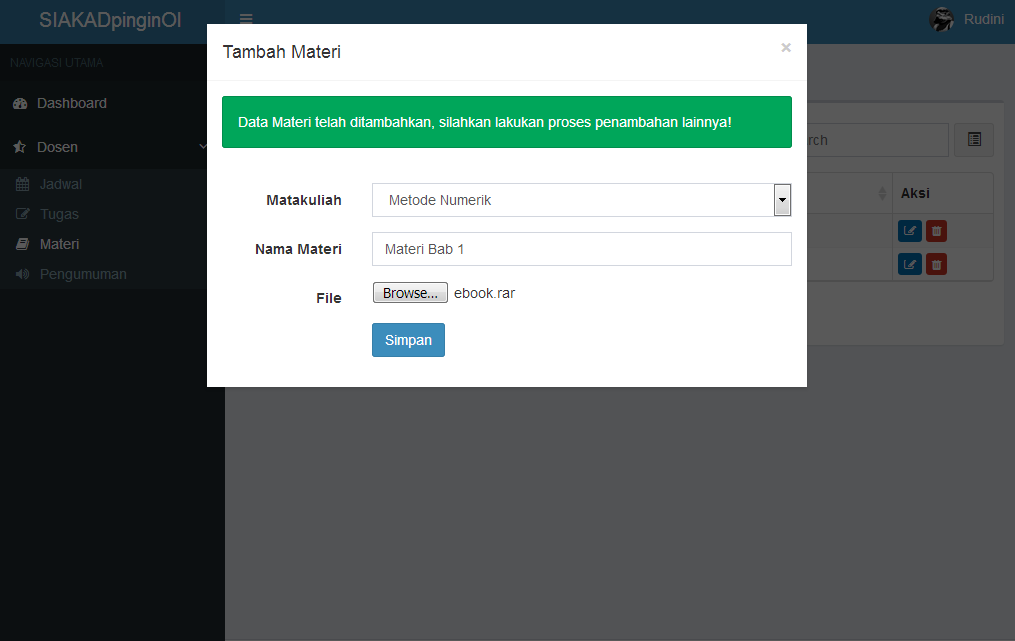
1. Pengujian upload materi

Hasil pengujian ini dilakukan untuk mengetahui fungsi upload materi telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

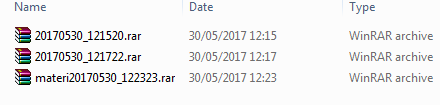
Tabel 11. Hasil pengujian tambah tugas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian upload materi | Dosen meng-input file materi pada form upload materi | Data materi bertambah dan file materi tesimpan pada folder materi | Muncul pesan “materi telah ditambahkan” dan file materi telah tersimpan pada folder materi | Sesuai |

Dalam pengujian ini data materi akan bertambah dan file materi akan tersimpan dalam folder materi di dalam server setelah dosen meng-input data pada form upload materi. Adapun tampilan dari *implementasi* tambah tugas adalah sebagai berikut :

Gambar . implementasi upload materi

Adapun hasil file yang di upload kedalam server akan tersimpan pada folder materi, isi folder materi dapat dilihat pada gambar

Gambar . file materi yang tersimpan dalam server

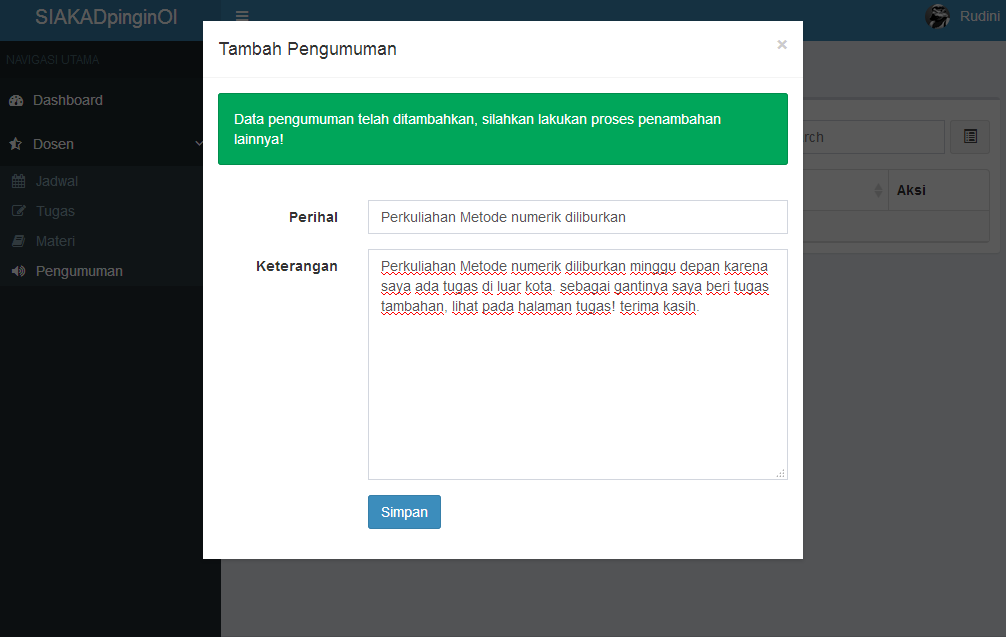
1. Pengujian tambah pengumuman

Hasil pengujian ini dilakukan untuk mengetahui fungsi tambah oengumuman telah berjalan dengan baik dan benar. Adapun hasil dari pengujian tersebut dapat dilihat pada tabel

Tabel 11. Hasil pengujian tambah pengumuman

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian tambah pengumuman | Dosen meng-input data pengumuman pada form pengumuman | Data pengumuman bertambah | Muncul pesan “ pengumuman telah ditambahkan” dan data pengumuman bertambah pada tabel daftar pengumuman | Sesuai |

Dalam pengujian ini data pengumuman akan bertambah setelah dosen meng-input data pada form tambah pengumuman. Adapun tampilan dari *implementasi* tambah pengumuman adalah sebagai berikut :



Gambar . implementasi tambah pengumuman

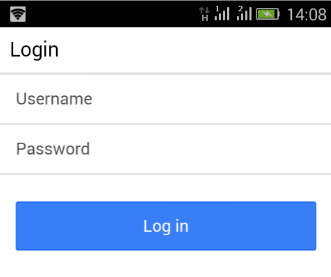
1. Pengujian login aplikasi client

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

Tabel 8. Hasil Pengujian *login* aplikasi client

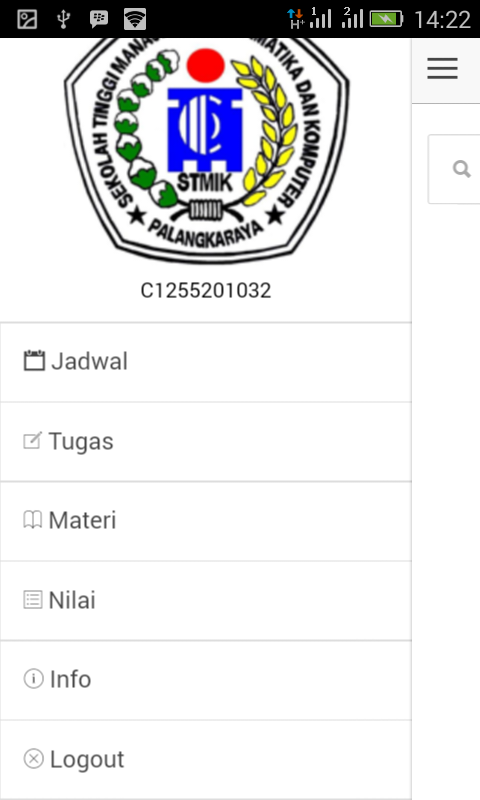
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian *login* mahasiswa pada aplikasi client | Pengguna memasukkan *username* dan *password* kemudian menekan tombol login | aplikasi akan menampilkan side menu aplikasi client | Halaman side menu aplikasi muncul | Sesuai |

Adapun tampilan dari *implementasi* *login* pada aplikasi client adalah sebagai berikut :



Gambar . implementasi login aplikasi client

Adapun tampilan dari *side menu* pada aplikasi client adalah sebagai berikut :



Gambar . implementasi side menu

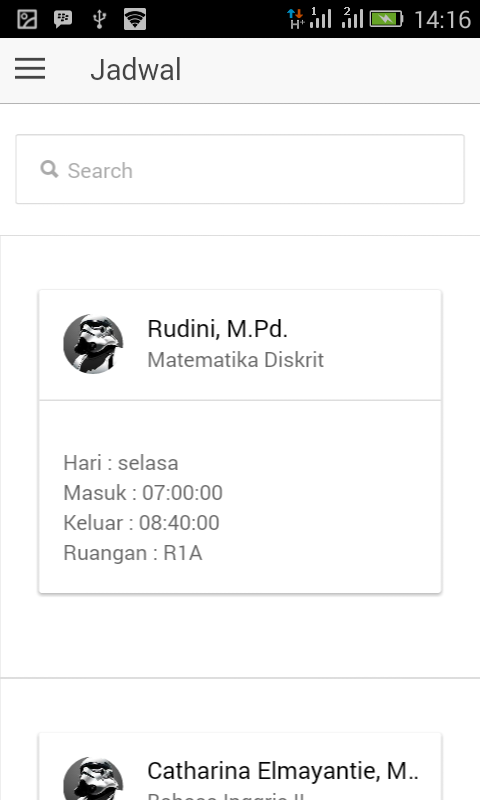
1. Pengujian menu jadwal aplikasi client

Hasil pengujian ini dilakukan untuk mengetahui apakah fungsi menu jadwal pada aplikasi client sudah berjalan dengan baik dan sesuai harapan. Adapun hasil dari pengujian tersebut dapat dilihat pada Tabel hasil pengujian login

Tabel 8. Hasil Pengujian menu jadwalaplikasi client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian menu jadwal aplikasi client | Pengguna menekan menu jadwal | aplikasi akan menampilkan halamanjadwal | Halaman jadwal tampil setelah loading | Sesuai |

Adapun tampilan dari *implementasi* menu jadwal pada aplikasi client adalah sebagai berikut :



Gambar . implementasi menu jadwal

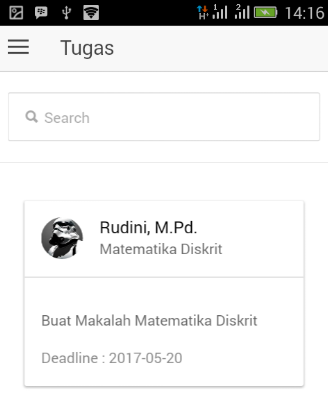
1. Pengujian menu tugas aplikasi client

Hasil pengujian ini dilakukan untuk mengetahui apakah fungsi menu jadwal pada aplikasi client sudah berjalan dengan baik dan sesuai harapan. Adapun hasil dari pengujian tersebut dapat dilihat pada Tabel hasil pengujian login

Tabel 8. Hasil Pengujian menu jadwalaplikasi client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hasil Pengujian** | | | | |
| **Nama pengujian** | **Data Masukan** | **Harapan** | **Pengamatan** | **Kesimpulan** |
| Pengujian menu jadwal aplikasi client | Pengguna menekan menu jadwal | aplikasi akan menampilkan halamanjadwal | Halaman jadwal tampil setelah loading | Sesuai |

Adapun tampilan dari *implementasi* menu tugas pada aplikasi client adalah sebagai berikut :



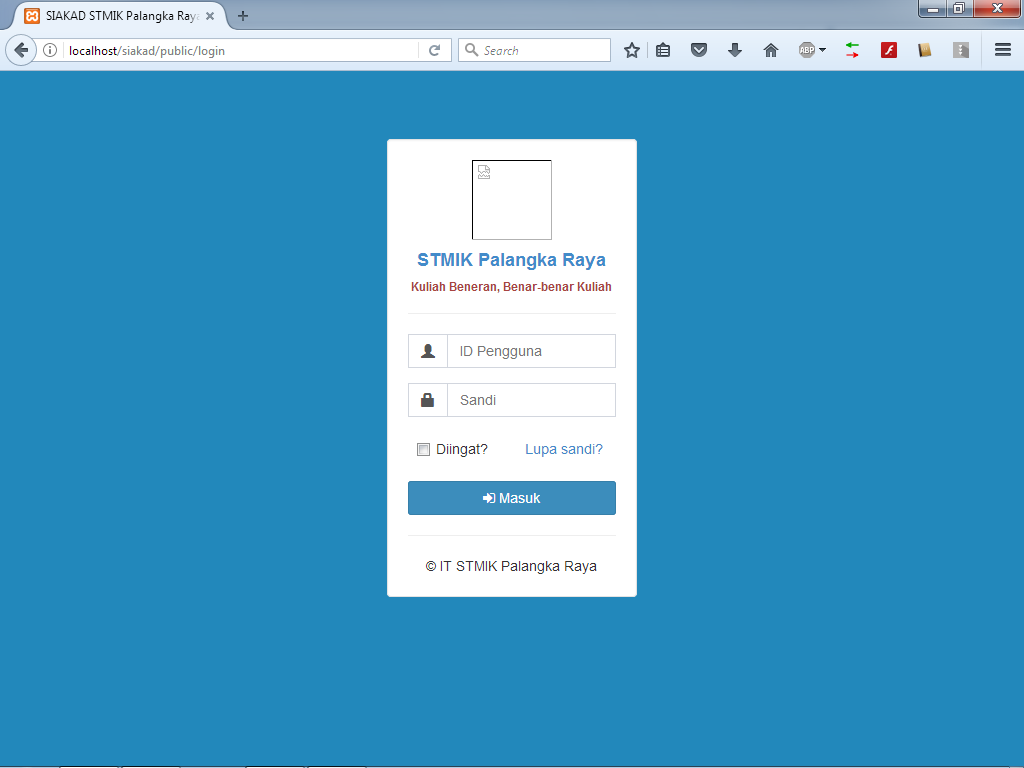
Gambar . implementasi menu tugas

1. Pengujian menu nilai aplikasi client
2. Pengujian menu informasi / pengumuman aplikasi client
3. Pengujian download materi aplikasi client
4. 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.

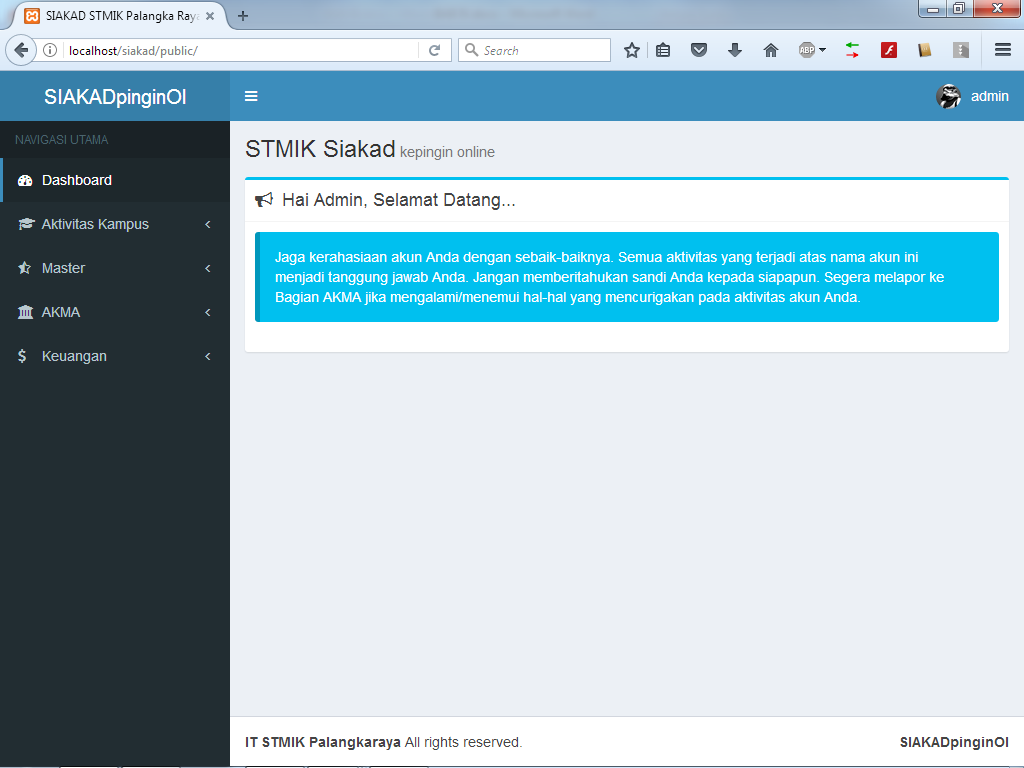
1. Admin

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 50



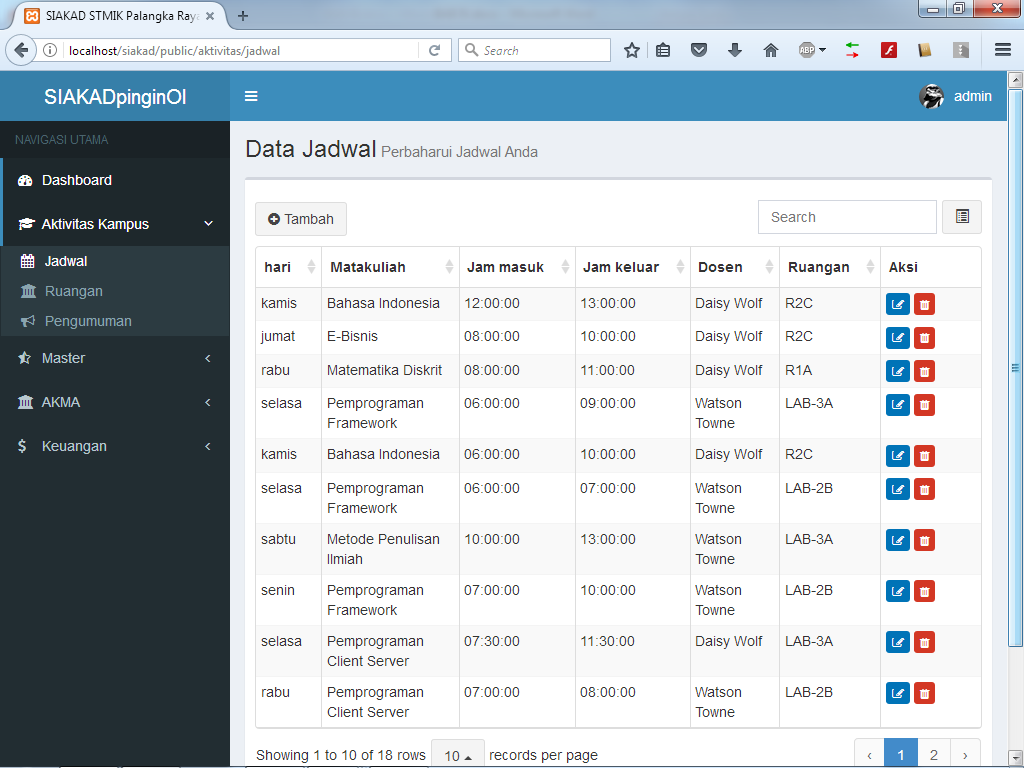
Gambar 50. login website server

Setelah memasukkan username dan password maka halaman akan beralih ke halaman dashboard admin. Tampilan dashboard admin dapat dilihat pada gambar 51

Gambar 51. Halaman *dashboard admin*

1. Menu jadwal bagian admin

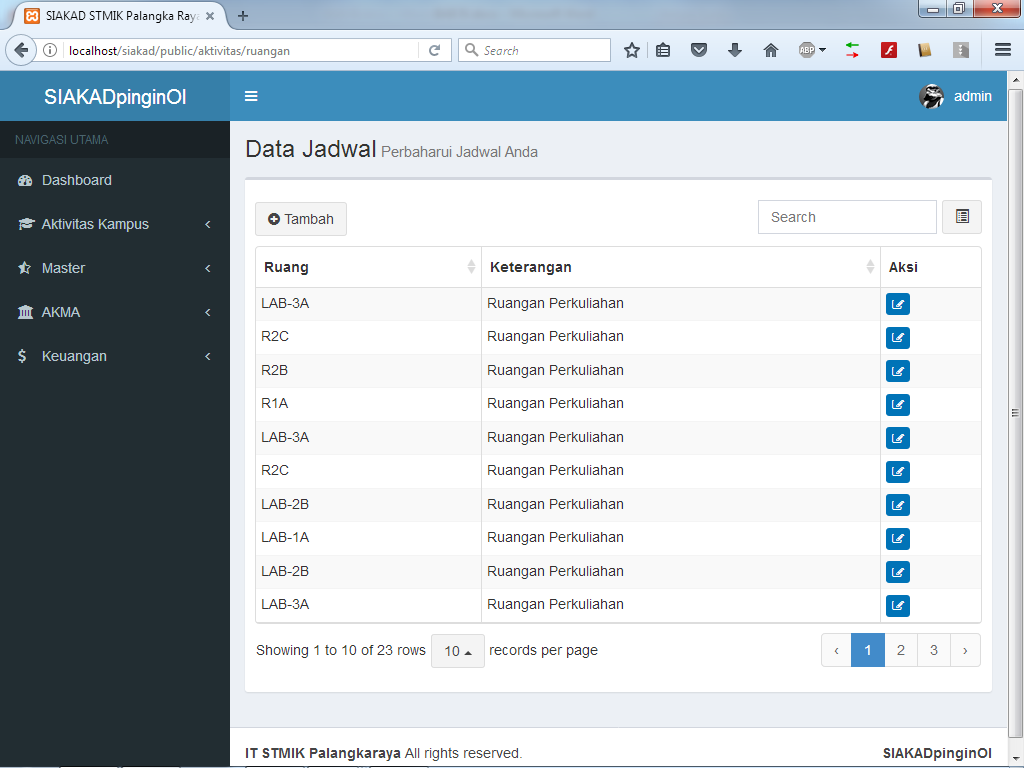
Menu jadwal digunakan untuk mengelola jadwal admin dapat menambahkan, mengedit dan menghapus jadwal matakuliah. Tampilan halaman jadwal dapat dilihat pada gambar 52



Gambar 52 halaman jadwal bagian admin

1. Menu ruangan

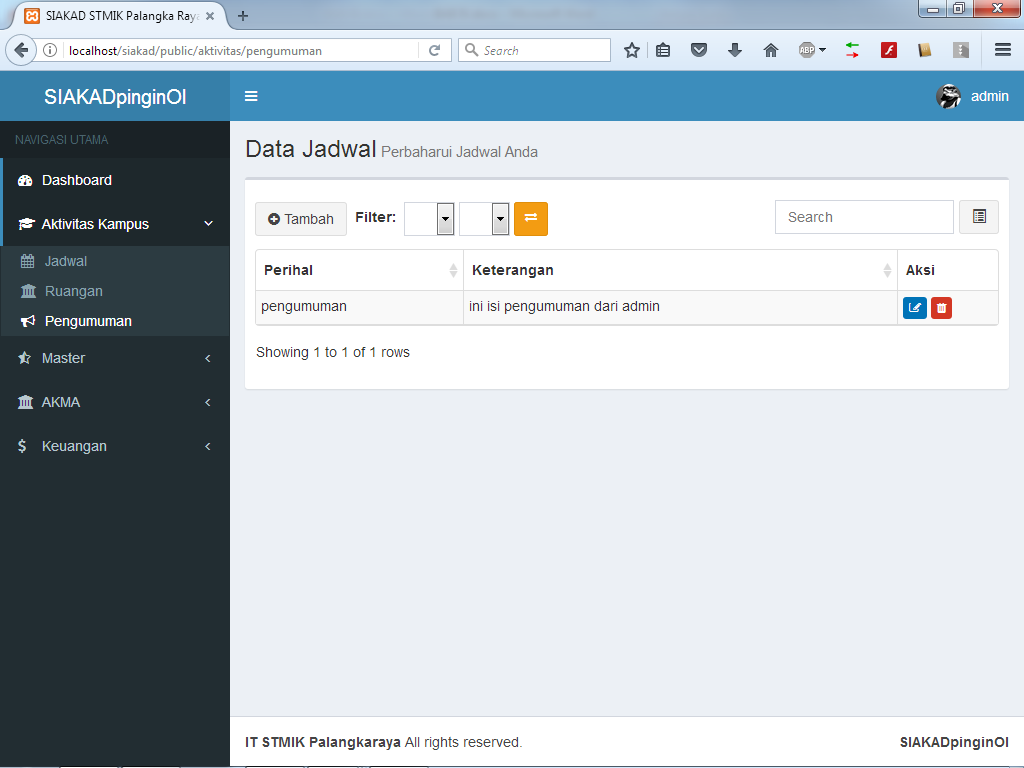
Menu ruangan digunnakan untuk mengelola ruangan, admin dapat menambah dan menghapus ruangan yang akan dipakai oleh jadwal matakuliah. Halaman ruangan dapat dilihat pada gambar 53



Gambar 53. Halaman ruangan

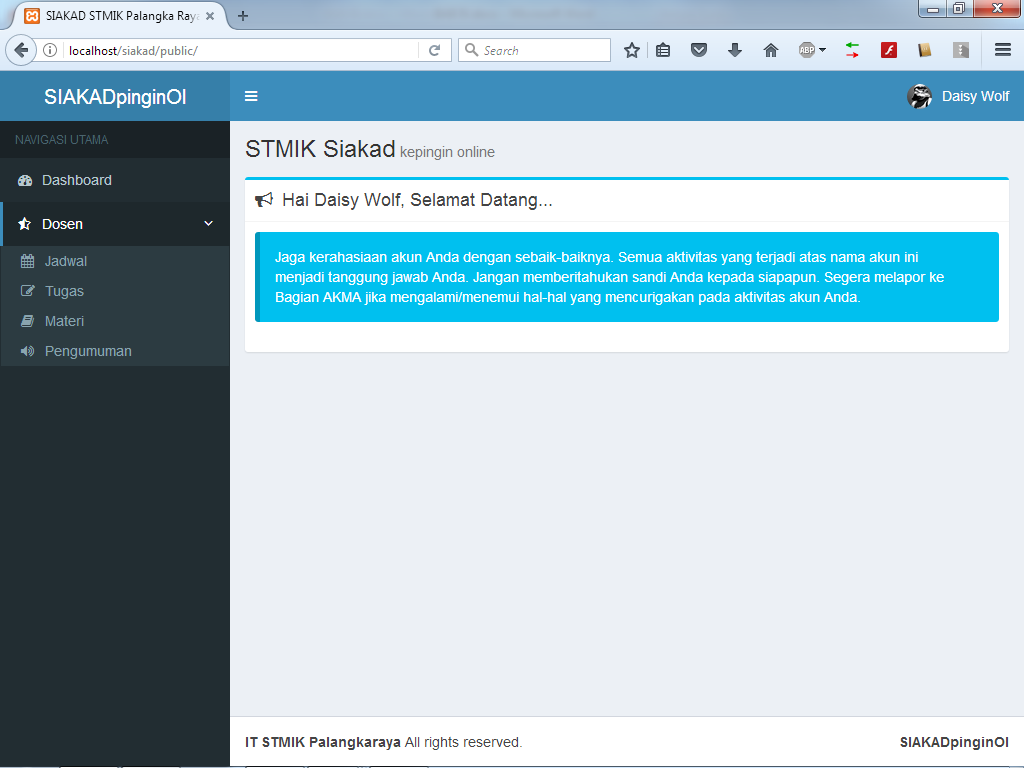
1. Menu pengumuman bagian admin

Menu pengumuman digunnakan untuk menambahkan dan menghapus pengumuman untuk mahasiswa. Halaman pengumuman dapat dilihat pada gambar 54

Gambar 54. halaman ruangan

1. Dosen

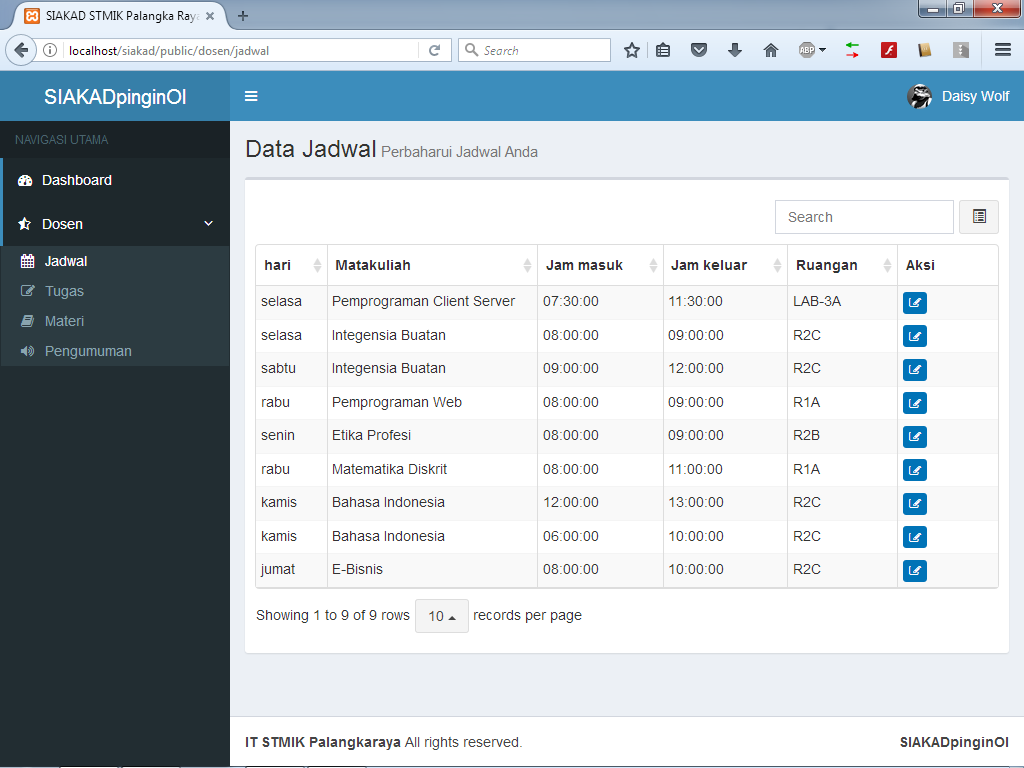
Bagian dosen adalah bagian program yang mengelola data – data jadwal, materi, tugas dan juga pengumuman. untuk masuk sebagai dosen pertama buka browser dikomputer dan buka website server pada browser. Maka halaman yang pertama kali muncul adalah halaman login lalu masukkan username dan password. Kemudian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



Gambar 55. Halaman dashboard bagian dosen

1. Menu jadwal bagian dosen

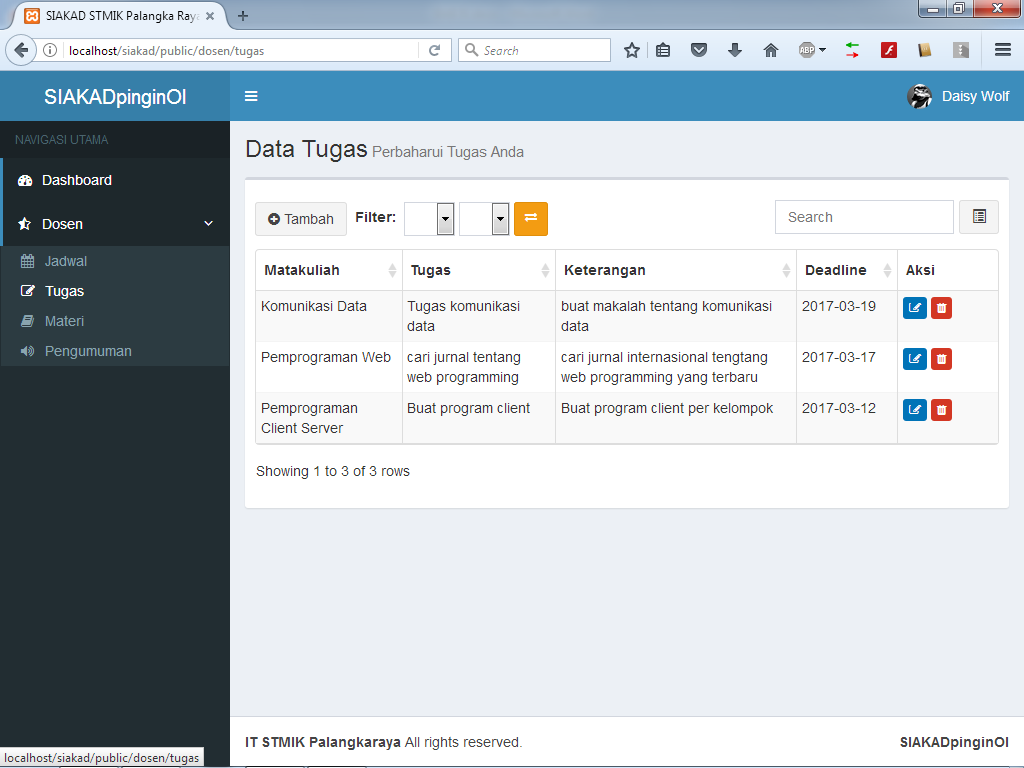
Menu jadwal bagian dosen digunnakan untuk mengelola jadwal, dosen dibatasi hanya bisa mengedit jadwal, dosen tidak bisa menambah dan menghapus jadwal. Halaman jadwal bagian dosen dapat dilihat pada gambar 56.



Gambar 56. Halaman jadwal bagian dosen

1. Menu tugas bagian dosen

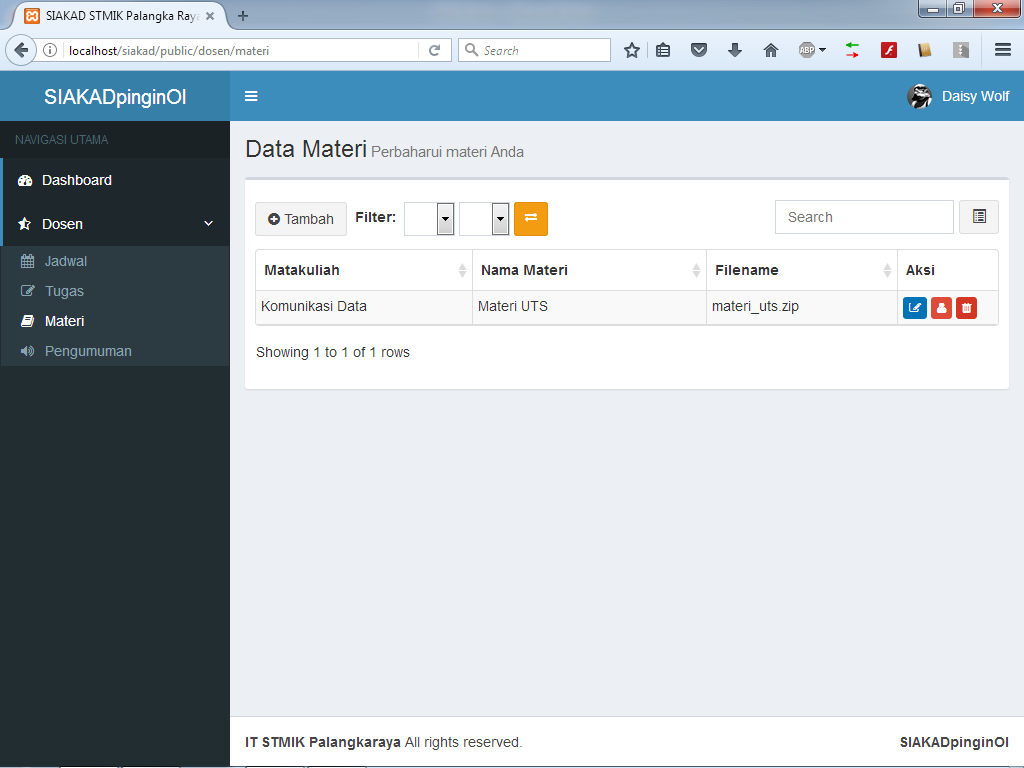
Menu tugas bagian dosen digunnakan untuk mengelola tugas, dosen dapat menambah, mengedit dan menghapus tugas. Halaman tugas dapat dilihat pada gambar 57.



Gambar 57. Halaman tugas bagian dosen

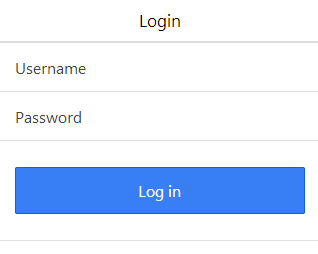
1. Menu materi bagian dosen

Menu materi bagian dosen digunnakan untuk mengelola materi, dosen dapat menambah, mengedit dan menghapus materi. Halaman materi dapat dilihat pada gambar 58

Gambar 58. 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 59.



Gambar 59. Halaman login aplikasi client

1. Side menu

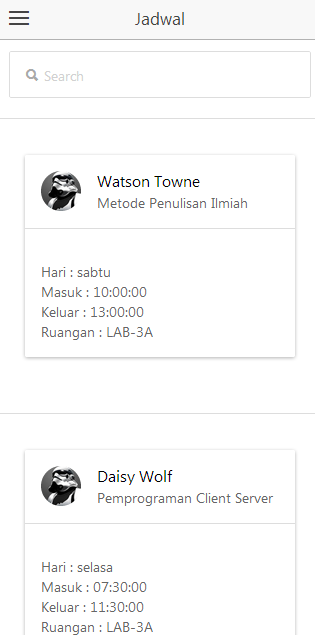
Side menu merupakan halaman berisi daftar menu - menu pada aplikasi client yang berada disamping halaman utama aplikasi. Halaman side menu dapat dilihat pada gambar 60.



Gambar 60. *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 61.



Gambar 61. halaman jadwal pada aplikasi client

1. Menu tugas

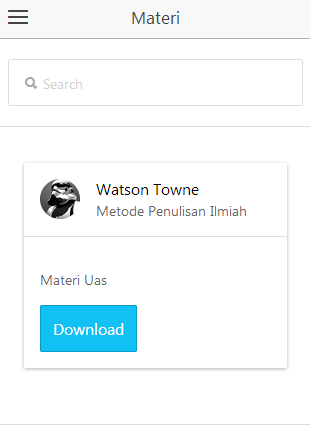
Halaman menu tugas merupakan halaman berisi data tugas yang diberikan oleh dosen kepada mahasiswa. Adapaun tampilan halaman menu tugas dapat dilihat pada Gambar 62.

****

Gambar 62. Halaman tugas pada aplikasi client

1. Halaman materi

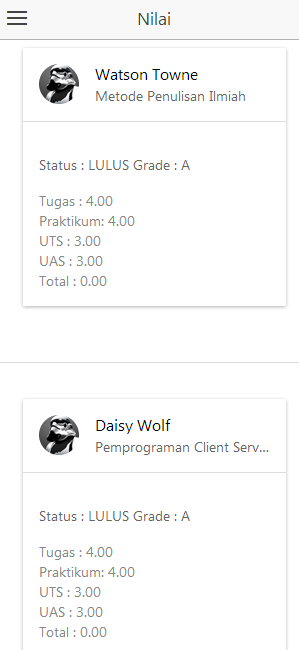
Halaman materi merupakan halaman berisi data materi, pada halaman materi terdapat tombol download untuk menngunduh file materi dari server. Halaman materi dapat dilihat pada Gambar 63.



Gambar 63. Halaman materi pada aplikasi client

1. Halaman nilai

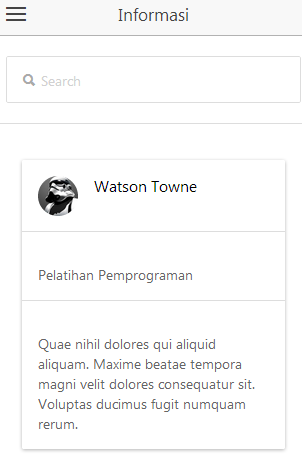
Halaman nilai merupakan halaman yang memuat daftar nilai mahasiswa. Halaman nilai dapat dilihat pada Gambar 64.



Gambar 64. 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 65.



Gambar 65. 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');  }); |

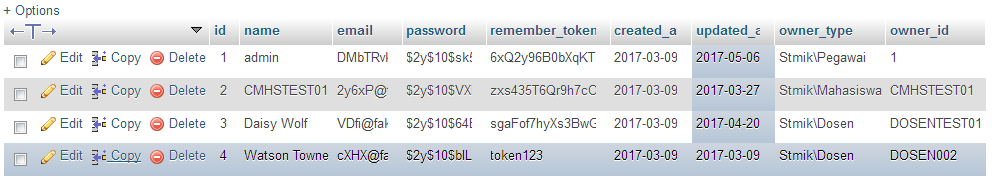
1. Kode program controller.js

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. Tabel users

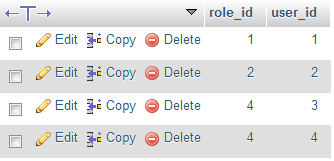
Tabel users digunakan untuk menyimpan data user account pengguna Aplikasi Pengelolaan Mata Kuliah Teknik Informatika Berbasis *Client Server* Pada STMIK Palangkaraya. Tabel users dapat dilihat pada gambar 66



Gambar 66. Tabel users

1. Tabel role\_user

Tabel role\_user adalah tabel yang menyimpan data berisi informasi hak akses user berdasarkan kategori hak akses dalam tabel roles. Tabel role\_user dapat di lihat pada gambar 67



Gambar 67. Tabel role\_user

1. Tabel roles

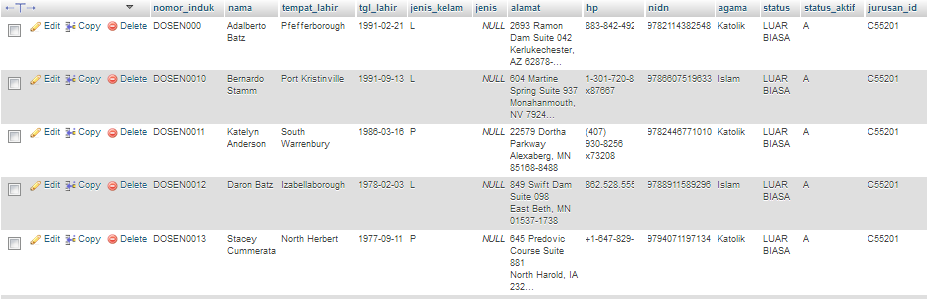
Tabel roles adalah tabel yang menyimpan data kategori hak akses user



Gambar 68. Tabel roles

1. Tabel dosen

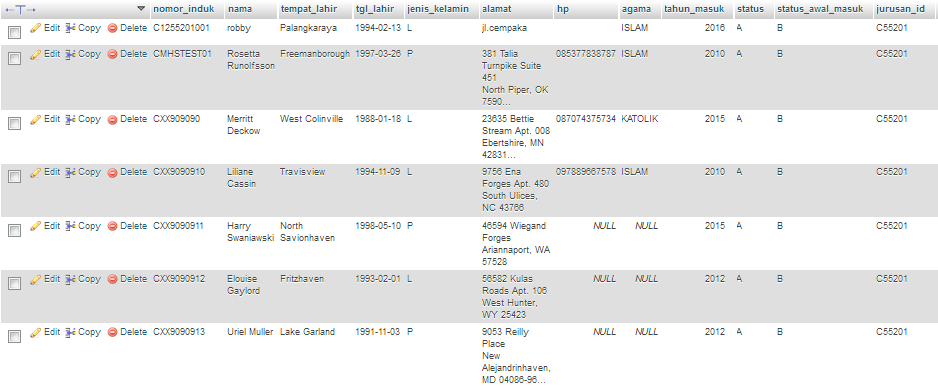
Tabel dosen digunakan untuk menyimpan data diri dosen - dosen Stmik palangkaraya. Tabel dosen dapat dilihat pada gambar 69.



Gambar 69. Tabel dosen

1. Tabel mahasiswa

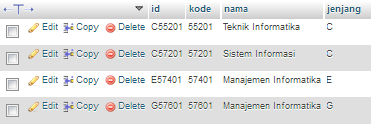
Tabel mahasiswa digunakan untuk menyimpan data diri semua mahasiswa Stmik palangkaraya. Tabel mahasiswa dapat dilihat pada gambar 70.



Gambar 70. Tabel mahasiswa

1. Tabel jurusan

Tabel jurusan digunakan untuk menyimpan data jurusan. Tabel jurusan dapat dilihat pada gambar 71



Gambar 71. Tabel jurusan

1. Tabel mata\_kuliah

Tabel mata\_kuliah digunakan untuk menyimpan data matakuliah. Tabel mata\_kuliah dapat dilihat pada gambar 72



Gambar 72. Tabel mata\_kuliah

1. Tabel pengampu\_kelas

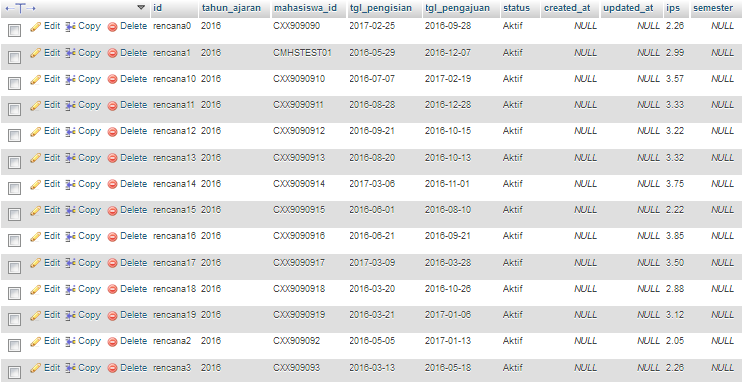
Tabel pengampu\_kelas digunakan untuk menyimpan data yang menghubungkan data dosen dalam tabel dosen dengan data matakuliah dalam tabel mata\_kuliah. Tabel pengampu\_kelas dapat dilihat pada gambar 73



Gambar 73. Tabel pengampu\_kelas

1. Tabel rencana\_studi

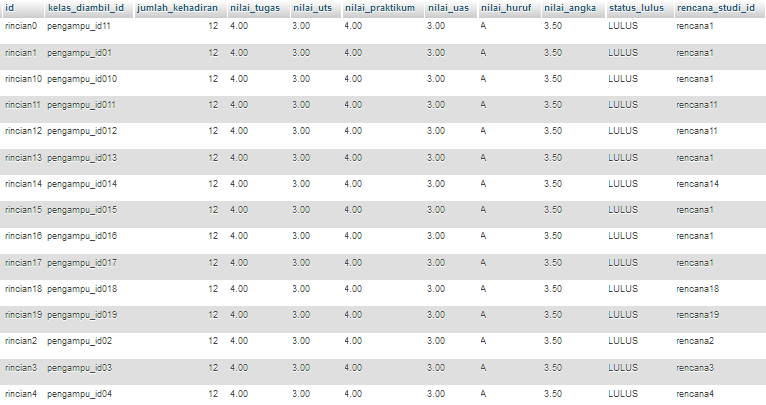
Tabel rencana\_studi digunakan untuk menyimpan data rencana stud mahasiswa stmik palangkaraya. Tabel rencana\_studi dapat dilihat pada gambar



Gambar 74. Tabel rencana\_studi

1. Tabel rincian\_studi

Tabel rincian\_studi digunakan untuk menyimpan rincian matakuliah seperti nilai dan jumlah kehadiran dari setiap matakuliah yang terdaftar dalam rencana studi mahasiswa. Tabel rincian\_studi dapat dilihat pada gambar 75.



Gambar 75. Tabel rencana\_studi

1. Tabel jadwal

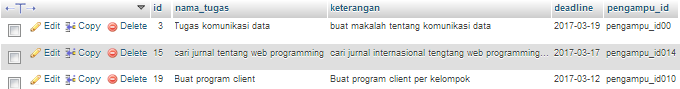
Tabel jadwal digunakan untuk menyimpan data jadwal matakuliah stmik palangkaraya. Tabel jadwal dapat dilihat pada gambar 76.



Gambar 76. Tabel jadwal

1. Tabel tugas

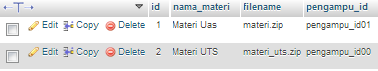
Tabel tugas digunakan untuk menyimpan data tugas yang dibuat oleh dosen. Tabel tugas dapat dilihat pada gambar 77.

****

Gambar 77. Tabel tugas

1. Tabel materi

Tabel materi digunakan untuk menyimpan nama materi dan nama file materi yang telah di upload oleh dosen. Tabel materi dapat dlihat pada gambar 78.



Gambar 78. Tabel materi

1. Tabel pengumuman

Tabel pengumuman digunakan untuk menyimpan data pengumuman. Tabel pengumuman dapat dilihat pada gambar 79.



Gambar 79. Tabel pengumuman

1. 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.