**BAB IV**

**IMPLEMENTASI DAN PEMBAHASAN**

1. **Implementasi**

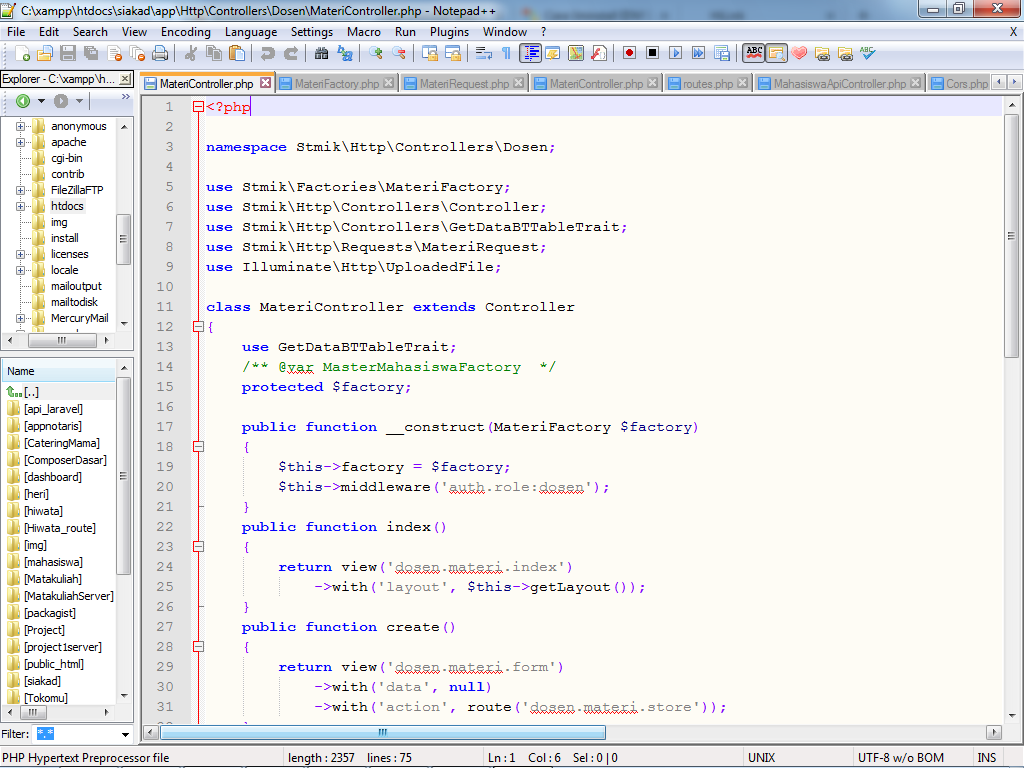
Dalam rencana penerapan sebuah sistem yang akan dibangun, agar sistem siap dioperasikan maka perlu diadakan kegiatan-kegiatan dari penerapannya. Adapun langkah-langkah yang akan dilakukan untuk penerapan sistem tersebut adalah pembuatan program, *testing* / uji coba sistem, dan sampai pada tahap pemeliharaan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

1. Proses pembuatan program

Setelah menentukan desain *interface* maka penulis mulai melakukan pengumpulan data sebagai bahan untuk pembuatan program. Penulis membuat program menggunakan bahasa pemrograman *php* untuk website server dan bahasa pemprograman javascript untuk aplikasi *client*. Adapun software yang digunakan dalam pembuatan program adalah sebagai berikut :

1. *Notepad ++*

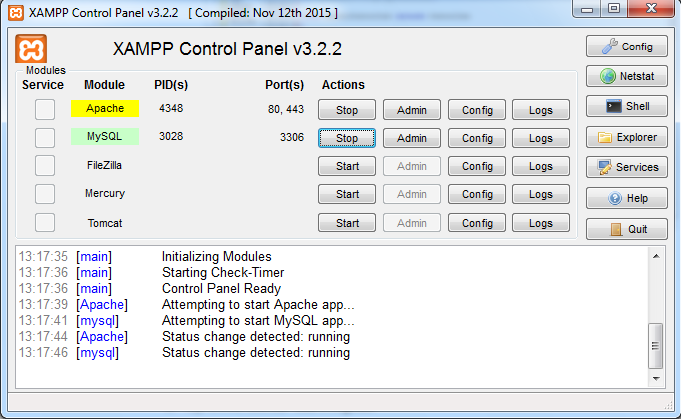
Penulis menggunakan editor notepad++ untuk menulis kode program dalam pembuatan aplikasi atau website. Seperti pada gambar berikut



Gambar 30. Tampilan Antarmuka notepad++

1. Xampp

Penulis menggunakan xampp versi 3.2.2 untuk menjalankan kode php dan database mysql selama pembuatan website. Seperti pada gambar berikut



Gambar 31. Xampp untuk lokal server selama pembuatan website

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 dipilih tida 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 tida 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 info | 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. Penjelasn manual program akan dibagi menjad dua yaitu manual *website server* dan manual aplikasi *client*.

1. Manual *website server*

Untuk dapat menggunakan website server ini maka perlu dilakukan instalasi program kedalam server pribadi atau menggunakan jasa hosting oleh penyedia hosting website. Setelah instalasi program selesai, maka website sudah dapat di operasikan. Berikut langkah - langkah dalam mengoperasikannya :

1. Ketikkan nama domain pada yang diberikan penyedia hosting *web browser* anda atau masuk ke dalam localhost jika program di-*install* dalam *server* pribadi.
2. Halaman login akan muncul kemudian masukkan *username* dan *password*, tekan tombol login.
3. Kemudian akan tampil halaman *dashboard* dengan daftar menu disamping kiri.
4. Jika menu jadwal ditekan maka akan muncul tabel jadwal, diatas tabel jadwal ada tombol tambah untuk tambah data. Pada tabel terdapat kolom aksi yang berisi tombol edit untuk mengedit data dan tombol hapus untuk menghapus data jadwal.
5. Jika menu ruangan ditekan maka akan muncul tabel ruangan, diatas tabel ruangan ada tombol tambah untuk tambah data. Pada tabel terdapat kolom aksi yang berisi tombol edit untuk mengedit data dan tombol hapus untuk menghapus data ruangan.
6. Jika menu pengumuman ditekan maka akan muncul tabel pengumuman, diatas tabel pengumuman ada tombol tambah untuk tambah data. Pada tabel terdapat kolom aksi yang berisi tombol edit untuk mengedit data dan tombol hapus untuk menghapus data pengumuman.
7. Jika menu tugas ditekan maka akan muncul tabel tugas, diatas tabel tugas ada tombol tambah untuk tambah data. Pada tabel terdapat kolom aksi yang berisi tombol edit untuk mengedit data dan tombol hapus untuk menghapus data tugas.
8. Jika menu materi ditekan maka akan muncul tabel materi, diatas tabel materi ada tombol tambah untuk tambah data. Pada tabel terdapat kolom aksi yang berisi tombol edit untuk mengedit data dan tombol hapus untuk menghapus data materi.
9. Manual aplikasi *client*

Aplikasi client ini terdiri *file* *setup/packge* berektensi *.apk* berupa file master yang dapat diinstal disetiap *handphone* berbasis *android*. Untuk dapat menggunakan program aplikasi ini maka perlu dilakukan instalasi program. Setelah terinstal aplikasi pada *handphone* yang digunakan, maka akan terdapat ikon stmik palangkaraya sebagai jalan pintas yang dapat digunakan untuk menjalankan aplikasi tersebut. Berikut langkah-langkah dalam menggunakan aplikasi *client*:

1. Tap / sentuh *icon* aplikasiyang ada pada *smartphone android* pengguna.
2. Layar *splash screen* akan muncul dengan gambar stmik palangkaraya.
3. Kemudian akan muncul tampilan *login*. Masukkan username dan password mahasiswa.
4. Setelah login selesai akan muncul halaman utama berisi data – data yang dikirim oleh server. Di pojok kiri aplikasi ada tombol yang jika ditekan akan memunculkan menu jadwal, menu tugas, menu materi, menu pengumuman dan menu logout.
5. Jika memilih menu jadwal maka mahasiswa akan mendapat halaman berisi data jadwal sesuai mata kuliah yang ditempuh.
6. Jika memilih menu tugas maka mahasiswa akan mendapat halaman berisi data tugas yang diberikan oleh dosen.
7. Jika memilih menu materi maka mahasiswa akan mendapat halaman berisi data materi yang diberikan oleh dosen. Pada masing – masing data materi terdapat tombol download, apabila mahasiswa menekan tombol download maka materi yang bersangkutan akan disimpan oleh aplikasi ke dalam *smartphone*.
8. Jika memilih menu tugas maka mahasiswa akan mendapat halaman berisis data pengumuman.
9. jika memilih menu logout maka aplikasi akan kembali menampilkan halaman login.
10. Manual instalasi

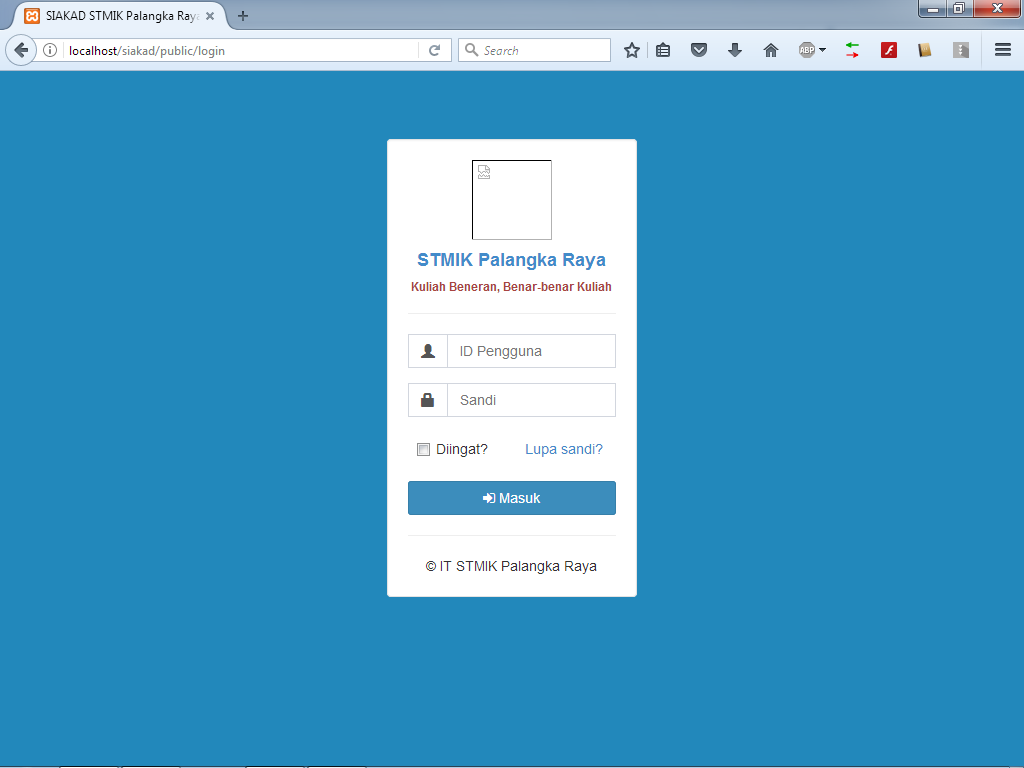
Pada manual instalasi ini akan dijelaskan tentang bagaimana cara instalasi website server pada server lokal di komputer bersistem operasi *windows 7* dan instalasi aplikasi client pada sistem operasi *android*.

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 *interface* / antarmuka program *server*
7. Halaman *Login*

Halaman *login* merupakanhalaman yang digunakan oleh pengguna untuk memasukkan identitas pengguna seperti username dan passoword agar bisa dikenali oleh sistem. Adapun implementasi dari halaman *login* dapat dilihat pada Gambar 31.



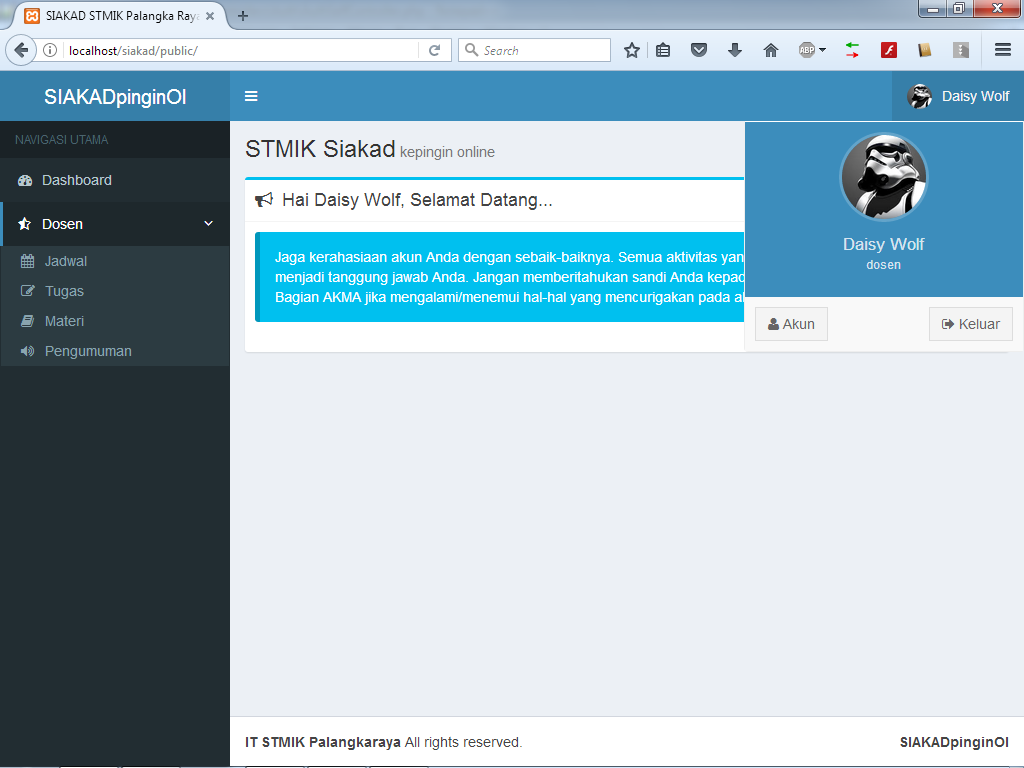
Gambar 31. Halaman *login server*

Kode program untuk halaman *login*:

|  |
| --- |
| <?php  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('/')); // di sini header X-IC-Redirect akan melakukan redirect  }  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. Halaman dashboard

Halaman dashboard merupakan halaman yang ditampilkan pertama kali ketika pengguna sudah berhasil login. Halaman dashboard dapat dilihat pada Gambar 32.

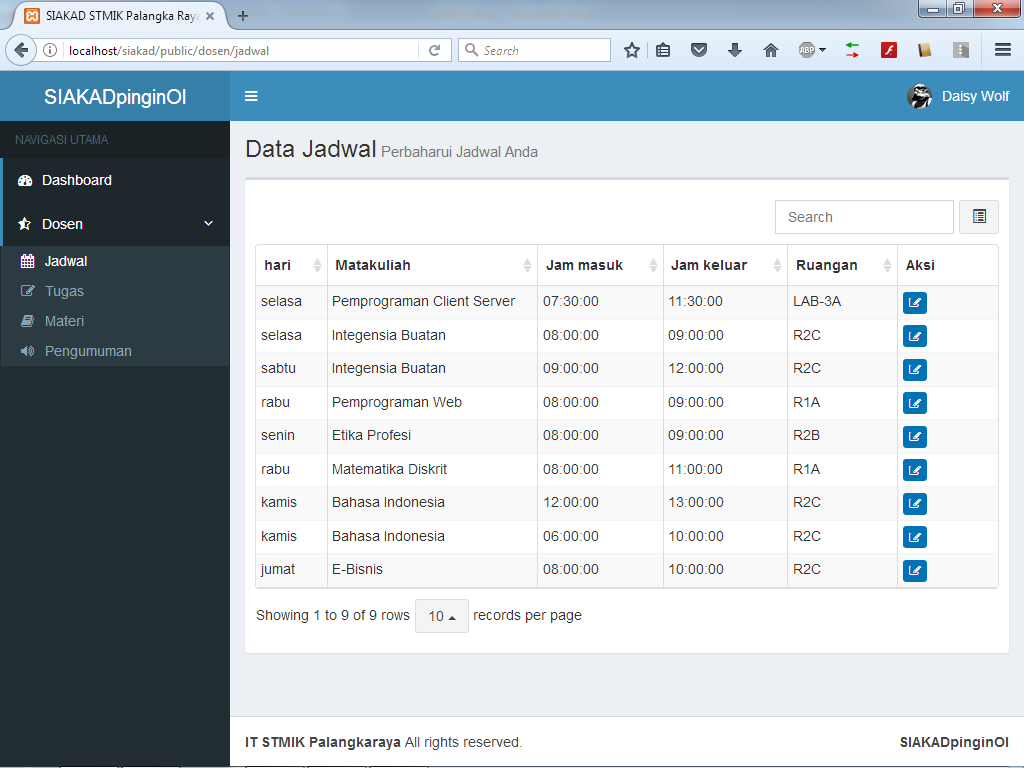
  
Gambar 32. Halaman dashboard

Kode program untuk menampilkan halaman dashboard:

|  |
| --- |
| <?php  namespace Stmik\Http\Controllers;  class SiteController extends Controller  {  public function \_\_construct()  {  $this->middleware('auth');  }  public function index()  {  return view('site.index')  ->with('layout', $this->getLayout());  }  } |

1. Halaman menu jadwal

Halaman menu jadwal merupakan halaman yang digunnakan untuk menambah, mengedit dan menghapus data jadwal. Halaman menu jadwal dapat dilihat pada Gambar 33.



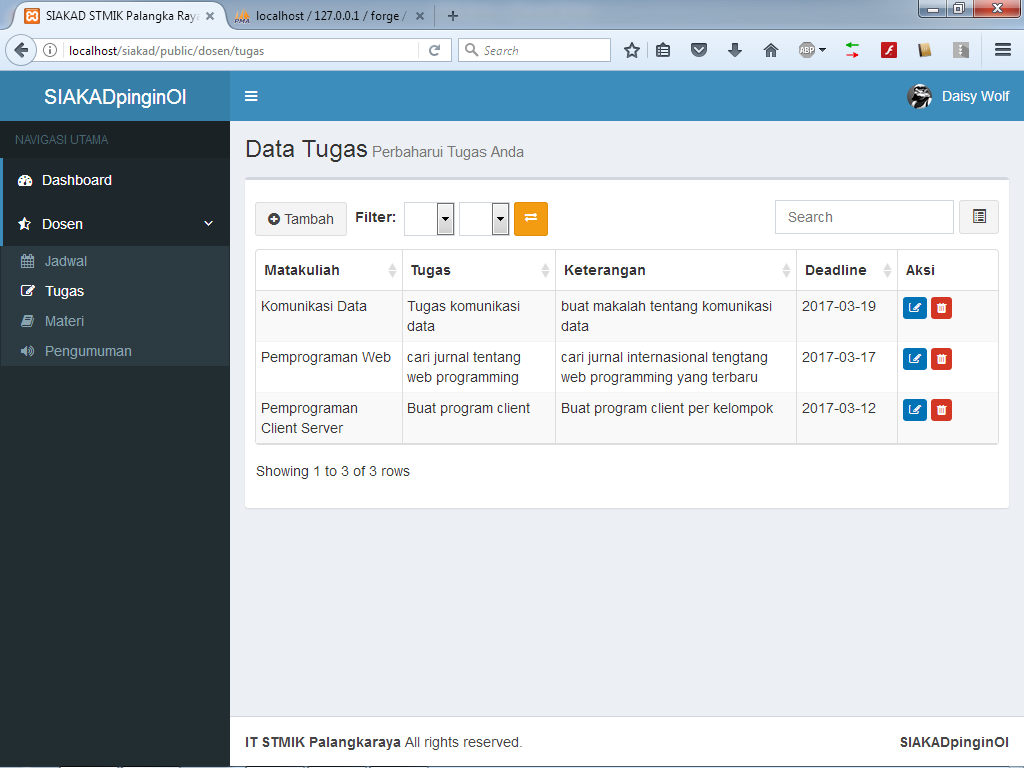
Gambar 33. Halaman Menu jadwal

Kode program untuk menu jadwal:

|  |
| --- |
| <?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. Halaman menu tugas

Halaman menu tugas merupakan halaman yang digunakan untuk menambah, mengedit dan menghapus data tugas. Halaman menu tugas dapat dilihat pada Gambar 33.



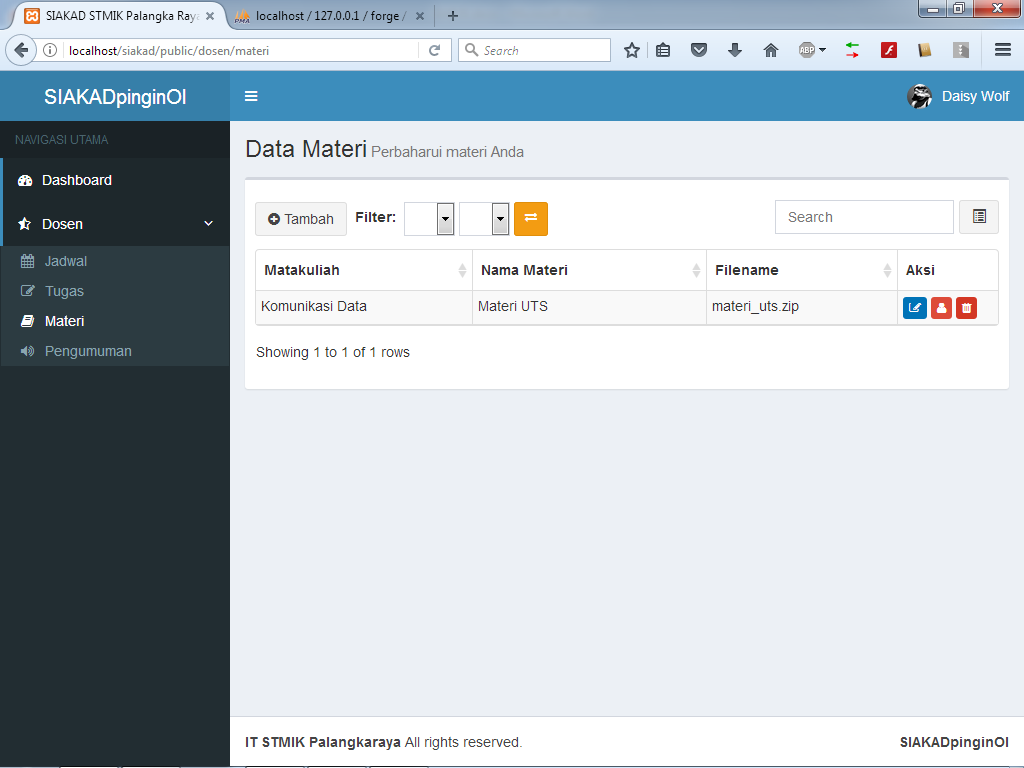
Gambar 34. Menu tugas

Kode program untuk menampilkan menu tugas:

|  |
| --- |
| <?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 Jadwal telah terupdate!");  }  return response(json\_encode($this->factory->getErrors()), 500);  }  public function store(TugasRequest $request)  {  $input = $request->all();  if($this->factory->store($input)) {  return $this->create()->with('success', "Data NIM {$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. Halaman menu materi

Halaman menu materi digunakan untuk melakukan upload file materi. Halaman menu materi dapat dilihat pada Gambar 35.



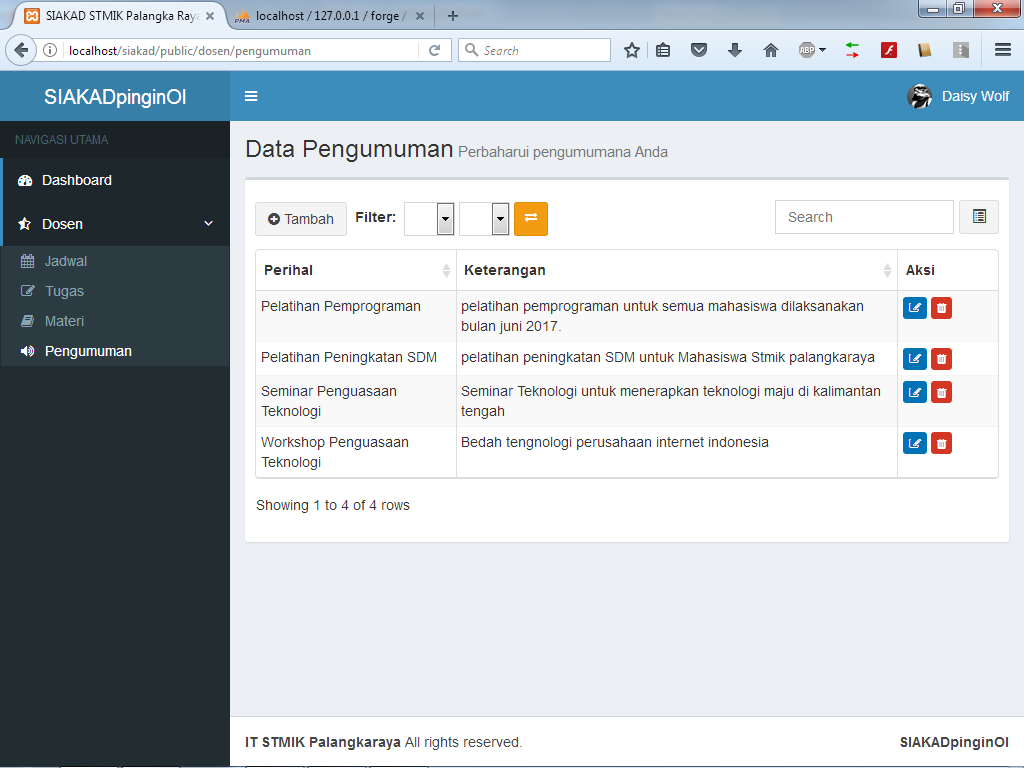
Gambar 35. Halaman menu materi

Kode program untuk halaman menu materi:

|  |
| --- |
| <?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 $this->create()->with('success', "Data NIM {$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 Jadwal 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. Halaman menu pengumuman

Halaman menu tugas merupakan halaman yang digunakan untuk menambah, mengedit dan menghapus data tugas. Halaman menu pengumuman dapat dilihat pada Gambar 36.



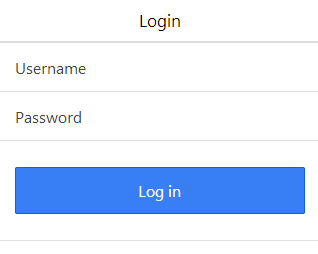
Gambar 36. Halaman menu pengumuman

Kode program untuk halaman menu pengumuman:

|  |
| --- |
| <?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. Pembahasan *interface* / antarmuka aplikasi *client*
2. Halaman *login* pada aplikasi

Halaman *login* merupakanhalaman yang digunakan oleh pengguna untuk memasukkan identitas pengguna seperti username dan passoword agar bisa dikenali oleh sistem. Adapun implementasi dari halaman *login* dapat dilihat pada Gambar 31.



Gambar 37. Halaman login pada aplikasi client

Kode program untuk halaman login pada aplikasi client

|  |
| --- |
| **<ion-view view-title="Login">**  **<ion-modal-view>**  **<ion-header-bar>**  **<h1 class="title">Login</h1>**  **</ion-header-bar>**  **<ion-content>**  **<form ng-submit="login()">**  **<div class="list">**  **<label class="item item-input">**  **<span class="input-label">Username</span>**  **<input type="text" ng-model="user.username">**  **</label>**  **<label class="item item-input">**  **<span class="input-label">Password</span>**  **<input type="password" ng-model="user.password">**  **</label>**  **<label class="item">**  **<button class="button button-block button-positive" type="submit">Log in</button>**  **</label>**  **</div>**  **</form>**  **</ion-content>**  **</ion-modal-view>**  **</ion-view>**  **.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: $scope.user.password**  **}**  **}**  **).success(function(data){**  **Service.loadingHide();**  **//set token dan nim**  **$rootScope.token = data.token;**  **$rootScope.nim = data.user.name;**  **Service.setToken(data.token);**  **Service.setNim(data.user.name);**  **//pergi ke halaman informasi**  **$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'**  **});**  **}**  **});**  **};**  **})** |

1. *Side menu* aplikasi *client*

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



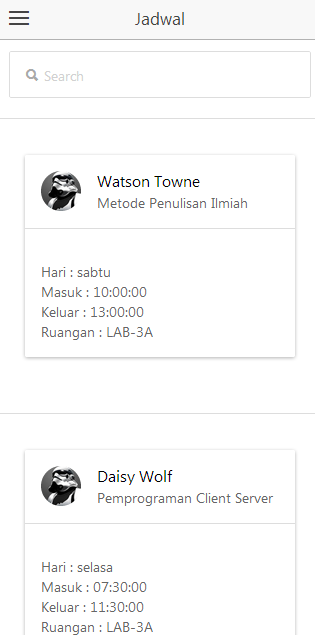
Gambar 33. *side menu* pada aplikasi client

Kode program untuk *side menu*:

|  |
| --- |
| <ion-side-menus enable-menu-with-back-views="false">  <ion-side-menu-content>  <ion-nav-bar class="bar-stable">  <ion-nav-back-button>  </ion-nav-back-button>  <ion-nav-buttons side="left">  <button class="button button-icon button-clear ion-navicon" menu-toggle="left">  </button>  </ion-nav-buttons>  </ion-nav-bar>  <ion-nav-view name="menuContent"></ion-nav-view>  </ion-side-menu-content>  **<ion-side-menu side="left">**  **<ion-content>**  **<div align="center">**  **<img src="img/STMIK.png" width="192" height="192"/>**  **<p>{{nim}}</p>**  **</div>**  **<ion-list>**  **<ion-item menu-close href="#/app/jadwal/{{nim}}/{{token}}">**  **<i class="icon ion-calendar"></i> Jadwal**  **</ion-item>**  **<ion-item menu-close href="#/app/tugas/{{nim}}/{{token}}">**  **<i class="icon ion-ios-compose-outline"></i> Tugas**  **</ion-item>**  **<ion-item menu-close href="#/app/materi/{{nim}}/{{token}}">**  **<i class="icon ion-ios-book-outline"></i> Materi**  **</ion-item>**  **<ion-item menu-close href="#/app/nilai/{{nim}}/{{token}}">**  **<i class="icon ion-ios-list-outline"></i> Nilai**  **</ion-item>**  **<ion-item menu-close href="#/app/info/{{nim}}/{{token}}">**  **<i class="icon ion-ios-information-outline"></i> Info**  **</ion-item>**  **<ion-item menu-close ng-click="logout()">**  **<i class="icon ion-ios-close-outline"></i> Logout**  **</ion-item>**  **</ion-list>**  **</ion-content>**  **</ion-side-menu>**  **</ion-side-menus>** |

1. Halaman jadwal pada aplikasi client

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

Kode program untuk halaman jadwal:

|  |
| --- |
| **<ion-view view-title="Jadwal">**  **<ion-content>**  **<ion-refresher**  **pulling-text="Pull to refresh..."**  **on-refresh="getJadwal()">**  **</ion-refresher>**  **<div class="list list-inset">**  **<label class="item item-input">**  **<i class="icon ion-search placeholder-icon"></i>**  **<input type="text" placeholder="Search" ng-model="search">**  **</label>**  **</div>**  **<ion-list>**  **<ion-item ng-repeat="jadwal in model|filter : search">**  **<div class="list card">**  **<div class="item item-avatar">**  **<img src="img/no-user-profile.jpg">**  **<h2>{{jadwal.dosen}}</h2>**  **<p>{{jadwal.matakuliah}}</p>**  **</div>**  **<div class="item item-body">**  **<p>**  **Hari : {{jadwal.hari}}</br>**  **Masuk : {{jadwal.jam\_masuk}}</br>**  **Keluar : {{jadwal.jam\_keluar}}</br>**  **Ruangan : {{jadwal.ruang}}</br>**  **</p>**  **</div>**  **</div>**  **</ion-item>**  **</ion-list>**  **</ion-content>**  **</ion-view>**  **.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 = $ionicPopup.alert({**  **title: 'Login failed!',**  **template: 'gagal tersambung ke server check koneksi internet anda'**  **});**  **}).finally(function() {**  **//stop loading...**  **Service.loadingHide();**  **// Stop the ion-refresher from spinning**  **$scope.$broadcast('scroll.refreshComplete');**  **});**  **};**  **$scope.getJadwal();**  **})** |

1. Halaman menu tugas pada aplikasi *client*

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

****

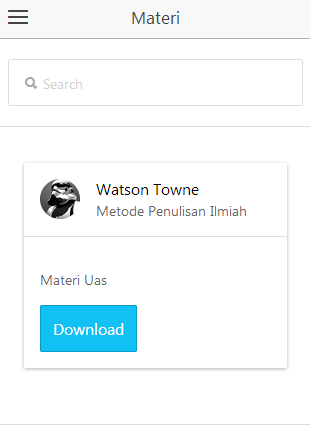
Gambar 39. Halaman tugas pada aplikasi client

Kode program untuk menampilkan halaman tugas:

|  |
| --- |
| **<ion-view view-title="Tugas">**  **<ion-content>**  **<ion-refresher**  **pulling-text="Pull to refresh..."**  **on-refresh="getTugas()">**  **</ion-refresher>**  **<div class="list list-inset">**  **<label class="item item-input">**  **<i class="icon ion-search placeholder-icon"></i>**  **<input type="text" placeholder="Search" ng-model="search">**  **</label>**  **</div>**  **<ion-list>**  **<ion-item ng-repeat="tugas in tugas|filter : search">**  **<div class="list card">**  **<div class="item item-avatar">**  **<img src="img/no-user-profile.jpg">**  **<h2>{{tugas.dosen}}</h2>**  **<p>{{tugas.matakuliah}}</p>**  **</div>**  **<div class="item item-body">**  **<p>**  **{{tugas.keterangan}}**  **</p>**  **<p>**  **<a href="#" class="subdued">Deadline : {{tugas.deadline}}</a>**  **</p>**  **</div>**  **</div>**  **</ion-item>**  **</ion-list>**  **</ion-content>**  **</ion-view>**  **.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 = $ionicPopup.alert({**  **title: 'Error!',**  **template: 'gagal tersambung ke server check koneksi internet anda'**  **});**  **}).finally(function() {**  **//stop loading...**  **Service.loadingHide();**  **// Stop the ion-refresher from spinning**  **$scope.$broadcast('scroll.refreshComplete');**  **});**  **};**  **$scope.getTugas();**  **})** |

1. Halaman materi pada aplikasi *client*

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.



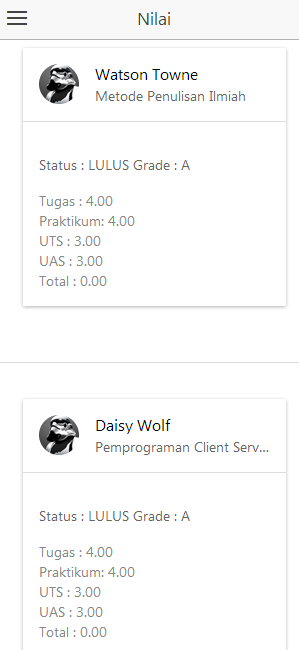
Gambar 40. Halaman materi pada aplikasi client

Kode program untuk halaman materi:

|  |
| --- |
| **<ion-view view-title="Materi">**  **<ion-content>**  **<ion-refresher**  **pulling-text="Pull to refresh..."**  **on-refresh="getMateri()">**  **</ion-refresher>**  **<div class="list list-inset">**  **<label class="item item-input">**  **<i class="icon ion-search placeholder-icon"></i>**  **<input type="text" placeholder="Search" ng-model="search">**  **</label>**  **</div>**  **<ion-list>**  **<ion-item ng-repeat="materi in materis|filter : search" >**  **<div class="list card">**  **<div class="item item-avatar">**  **<img src="img/no-user-profile.jpg">**  **<h2>{{materi.dosen}}</h2>**  **<p>{{materi.matakuliah}}</p>**  **</div>**  **<div class="item item-body">**  **<p>**  **{{materi.nama\_materi}}**  **</p>**  **<p>**  **<button class="button button-calm" ng-click="Download(materi.filename)">Download</button>**  **</p>**  **</div>**  **</div>**  **</ion-item>**  **</ion-list>**  **</ion-content>**  **</ion-view>**  **.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 = $ionicPopup.alert({**  **title: 'Login failed!',**  **template: 'gagal tersambung ke server check koneksi internet anda'**  **});**  **}).finally(function() {**  **//stop loading...**  **Service.loadingHide();**  **// Stop the ion-refresher from spinning**  **$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) {**  **//stop loading...**  **Service.loadingHide();**  **// Stop the ion-refresher from spinning**  **$scope.$broadcast('scroll.refreshComplete');**  **$scope.hasil = 'Save file on '+targetPath+' success!';**  **var alertPopup = $ionicPopup.alert({**  **title: 'Success',**  **template: 'berhasil download, Save file on '+targetPath**  **})**  **}, function (error) {**  **//stop loading...**  **Service.loadingHide();**  **// Stop the ion-refresher from spinning**  **$scope.$broadcast('scroll.refreshComplete');**  **var alertPopup = $ionicPopup.alert({**  **title: 'Alert!',**  **template: 'gagal download'**  **});**  **alert(error);**  **}, function (progress) {**  **$scope.downloadProgress = (progress.loaded / progress.total) \* 100;**  **});**  **});**  **}**  **})** |

1. Halaman nilai pada aplikasi client

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



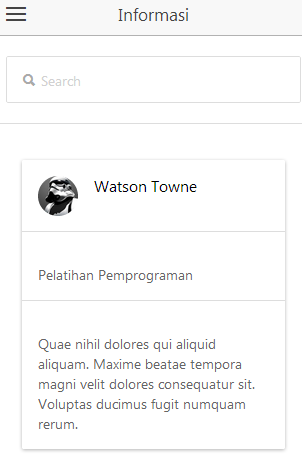
Gambar 41. Halaman nilai pada aplikasi *client*

Kode program untuk halaman nilai:

|  |
| --- |
| **<ion-view view-title="Nilai">**  **<ion-content>**  **<ion-refresher**  **pulling-text="Pull to refresh..."**  **on-refresh="getNilai()">**  **</ion-refresher>**  **<div class="list list-inset">**  **<label class="item item-input">**  **<i class="icon ion-search placeholder-icon"></i>**  **<input type="text" placeholder="Search" ng-model="search">**  **</label>**  **</div>**  **<ion-list>**  **<ion-item ng-repeat="data in nilai|filter : search">**  **<div class="list card">**  **<div class="item item-avatar">**  **<img src="img/no-user-profile.jpg">**  **<h2>{{data.dosen}}</h2>**  **<p>{{data.matakuliah}}</p>**  **</div>**  **<div class="item item-body">**  **<p>**  **Status : {{data.status\_lulus}} Grade : {{data.nilai\_huruf}}**  **</p>**  **<p>**  **<a href="#" class="subdued">Tugas : {{data.nilai\_tugas}}</a></br>**  **<a href="#" class="subdued">Praktikum: {{data.nilai\_praktikum}}</a></br>**  **<a href="#" class="subdued">UTS : {{data.nilai\_uts}}</a></br>**  **<a href="#" class="subdued">UAS : {{data.nilai\_uas}}</a></br>**  **<a href="#" class="subdued">Total : {{data.nilai\_akhir}}</a>**  **</p>**  **</div>**  **</div>**  **</ion-item>**  **</ion-list>**  **</ion-content>**  **</ion-view>**  **.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 = $ionicPopup.alert({**  **title: 'Login failed!',**  **template: 'gagal tersambung ke server check koneksi internet anda'**  **});**  **}).finally(function() {**  **//stop loading...**  **Service.loadingHide();**  **// Stop the ion-refresher from spinning**  **$scope.$broadcast('scroll.refreshComplete');**  **});**  **};**  **$scope.getNilai();**  **})** |

1. Halaman informasi atau pengumuman

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



Gambar 42. Halaman informasi

Kode program untuk halaman informasi:

|  |
| --- |
| **<ion-view view-title="Informasi">**  **<ion-content>**  **<ion-refresher**  **pulling-text="Pull to refresh..."**  **on-refresh="getInfo()">**  **</ion-refresher>**  **<div class="list list-inset">**  **<label class="item item-input">**  **<i class="icon ion-search placeholder-icon"></i>**  **<input type="text" placeholder="Search" ng-model="search">**  **</label>**  **</div>**  **<ion-list>**  **<ion-item ng-repeat="info in informasi|filter : search">**  **<div class="list card">**  **<div class="item item-avatar">**  **<img src="img/no-user-profile.jpg">**  **<h2>{{info.dosen}}</h2>**  **</div>**  **<div class="item item-body">**  **<p>{{info.perihal}}</p>**  **</div>**  **<div class="item item-body">**  **<p>**  **{{info.keterangan}}**  **</p>**  **</div>**  **</div>**  **</ion-item>**  **</ion-list>**  **</ion-content>**  **</ion-view>**  **.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();**  **})** |

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 ini sangat baik dan layak untuk digunakan.