

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Aplikasi

Perancangan Aplikasi Sistem Informasi Pergerakan Mahasiswa Islam Indonesia (PMII) ini menggunakan metode RAD (*Rapid Application Development*) dengan tahapan *Requirement Planning*, *RAD Design Workshop* dan *Implementation*. Pelaksanaan keseluruhan prosedur pengembang penelitian ini secara rinci dapat dilihat sebagai berikut:

4.1.1 Requirement Planning (Perencanaan Kebutuhan)

Aplikasi ini diperuntukkan untuk anggota Pergerakan Mahasiswa Islam Indonesia (PMII) Jepara yang selama ini mendapatkan informasi masih dilakukan secara manual. Dengan adanya aplikasi ini akan mempermudah anggota dan pengurus mendapatkan informasi dengan lengkap seperti kegiatan PMII, pendaftaran kegiatan serta pelaksanaan kegiatan atau informasi lainnya secara online tanpa harus datang ke sekretariat/basecamp untuk mencari informasi.

4.1.1.1 Objek Penelitian

Objek penelitian merupakan permasalahan yang diteliti oleh peneliti. Objek penelitian yang peneliti pilih adalah Organisasi Pergerakan Mahasiswa Islam Indonesia (PMII) Kabupaten Jepara.

4.1.1.2 Analisis Kebutuhan Masalah

1) Analisis Sistem yang Berjalan

Pergerakan Mahasiswa Islam Indonesia (PMII) merupakan satu-satunya organisasi eksternal yang terdapat di Universitas Islam Nahdlatul Ulama (UNISNU) Jepara. Organisasi ini masih menggunakan metode manual dari dulu hingga sekarang seperti pendaftaran anggota baru, pendaftaran kegiatan, pencatatan data anggota dan lain sebagainya. Hal ini memiliki resiko yang besar apalagi jika data yang dimiliki oleh PMII hilang atau rusak, ditambah lagi pengurus tidak memiliki *backup* data. Jadi mereka tidak bisa memproses ulang data apabila data yang mereka miliki telah rusak ataupun hilang.

2) Solusi Pemecahan Masalah

Adapun solusi dari pemecah masalah yang peneliti lakukan dalam menyelesaikan masalah yang dialami oleh organisasi Pergerakan Mahasiswa Islam Indonesia adalah dengan membuat aplikasi sistem informasi berbasis android untuk mempermudah pengelolaan manajemen administrasi seperti pengelolaan data anggota, penyimpanan data dan pembuatan laporan keanggotaan. Aplikasi ini dapat membantu anggota untuk mendapatkan informasi secara cepat.

4.1.1.3 Analisa Kebutuhan Alat dan Bahan

a. Alat

Peralatan yang digunakan dalam membuat aplikasi sistem informasi pergerakan mahasiswa islam Indonesia (pmii) Jepara menggunakan Android Studio 3.4 adalah:

1) Perangkat Keras (*Hardware*)

Adapun perangkat keras (*Hardware*) yang digunakan dalam membangun aplikasi yaitu berupa laptop dan smartphone dengan sistem operasi android dengan spesifikasi sebagai berikut:

a) Laptop:

- *Operating Sistem* : Windows 10 enterprise 64-bit
- *Processor* : Intel Core i3
- *Memory* : 4.0 GB RAM (Direkomendasikan 8.0 GB)

b) Smartphone:

- *Sistem operasi* : Android 7.1.1
- *Jaringan* : GSM/HSPA/LTE

2) Perangkat Lunak(Software)

Perangkat lunak yang di gunakan di Laptop adalah

- Windows 10 Enterprise 64-bit
- Android Studio
- *JDK*
- Node JS
- SDK

- Web Browser Mozilla Firefox
- Corel Draw X7
- yEd

b. Bahan

Bahan-bahan yang diperlukan untuk menjalankan aplikasi ini antara lain :

1. Formulir pendaftaran kegiatan MAPABA dan PKD yang diadakan di Kabupaten Jepara.
2. Rekap data anggota PMII Jepara
3. Data struktur pengurus PMII Jepara yang terdiri dari lima fakultas yang ada di Universitas Islam Nahdlatul Ulama Jepara.
4. Jadwal kegiatan Pergerakan Mahasiswa Islam Indonesia (PMII) Jepara.
5. Materi ataupun modul yang harus dimiliki anggota Pergerakan Mahasiswa Islam Indonesia.

4.1.1.4 Analisis Kebutuhan Fungsionalitas Sistem

Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dapat dilakukan oleh sistem. Kebutuhan yang disediakan oleh sistem informasi Pergerakan mahasiswa Islam Indonesia (PMII) Jepara dibedakan menjadi 2 (dua) sesuai hak akses sistem informasi yaitu admin dan user. Kebutuhan tersebut antara lain :

1) Admin

- a. Admin dapat mengelola data pendaftaran kegiatan
- b. Admin dapat melihat dan merubah data jadwal pelaksanaan kegiatan
- c. Admin dapat melihat dan merubah data struktur pengurus
- d. Admin dapat melihat data keanggotaan dan mencetak data anggota
- e. Admin dapat mengupload materi ataupun modul tentang PMII

2) User

- a. User dapat melakukan pendaftaran kegiatan dan mencetak bukti pendaftaran
- b. User dapat melihat dan mencari jadwal kegiatan
- c. User dapat melihat struktur kepengurusan PMII Jepara
- d. User dapat mengunduh materi ataupun modul yang diperlukan

- e. User dapat melakukan Login dan Register untuk membuat akun

4.1.1.5 Analisis Kebutuhan Non Fungsionalitas Sistem

Kebutuhan non fungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Analisis kebutuhan non fungsional sistem yang meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak minimum agar sistem dapat berjalan dengan optimal.

4.1.2 RAD Desain Workshop

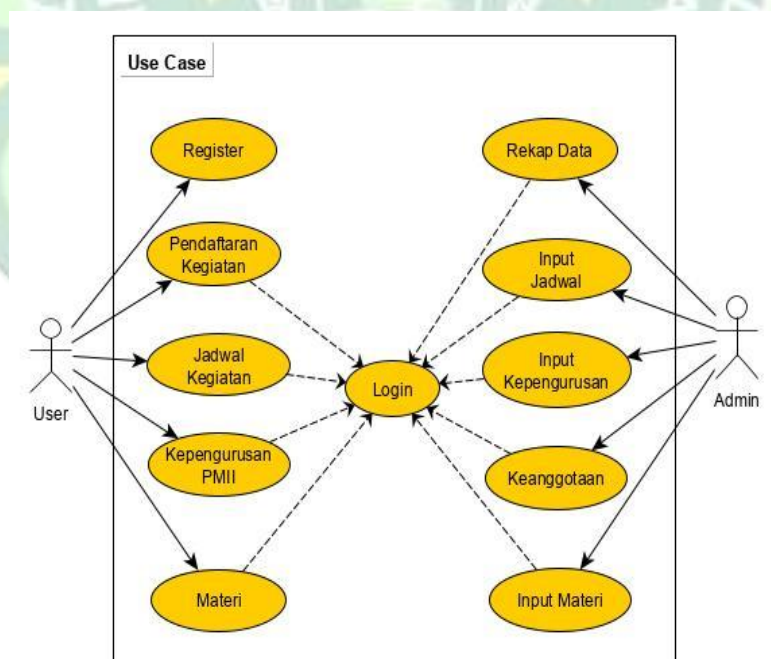
Pada tahap ini perancangan sistem yang digunakan oleh peneliti adalah *UML (Unified Modelling Language)* yang digambarkan melalui perancangan sistem seperti Use Case Diagram, Activity Diagram, Sequence Diagram dan Class Diagram.

4.1.2.1 Perancangan Berorientasi Objek

1. Use Case Diagram

Use Case Diagram merupakan diagram yang menunjukkan interaksi antara sistem aplikasi dengan actor. Pada aplikasi ini terdapat 2 aktor yang saling berinteraksi dengan sistem yaitu admin dan user.

Use Case dapat dilihat pada gambar 4.1 berikut ini:



Gambar 4.1 Use Case Diagram

Table 4.1 Deskripsi Use Case Diagram

No.	Aktor	Proses Use Case	Keterangan
1	Admin	Login	Admin melakukan login sistem dengan input <i>username</i> dan password
2	Admin	Rekap Data	Admin dapat mengelola data pendaftar kegiatan seperti melihat data dan hapus data
3	Admin	Input Jadwal	Admin mengelola jadwal kegiatan mulai dari tambah jadwal, edit jadwal dan hapus jadwal
4	Admin	Input Kepengurusan	Admin mengelola data master pengurus PMII seperti tambah jabatan dan tambah nama pengurus sesuai dengan jabatan
5	Admin	Keanggotaan	Admin mengelola data anggota yaitu lihat data anggota, cetak data dan hapus data
6	Admin	Input Materi	Admin dapat mengupload materi
7	Admin	Logout	Admin dapat melakukan Logout
No.	Aktor	Proses Use Case	Keterangan
1	User	Register	User membuat akun dengan mengisi identitas diri seperti nim, nama, fakultas, no_hp, dan password
2	User	Login	Admin melakukan login sistem dengan input <i>username</i> dan password
3	User	Pendaftaran Kegiatan	User dapat melakukan pendaftaran kegiatan seperti mapaba dan pkd, serta user dapat mencetak bukti pendaftaran
4	User	Jadwal Kegiatan	User dapat melihat jadwal kegiatan dan mencari jadwal kegiatan sesuai dengan nama

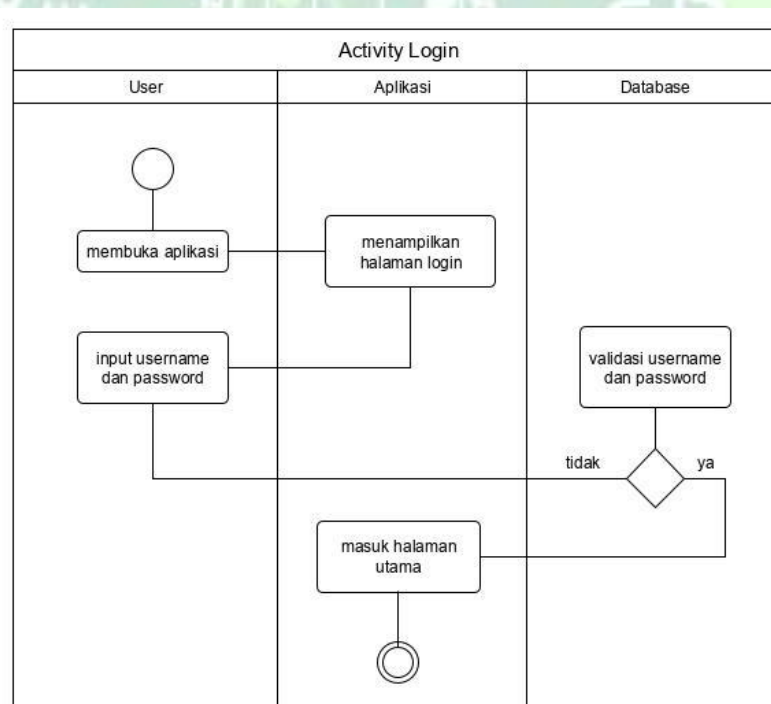
			kegiatan
5	User	Kepengurusan PMII	User dapat melihat struktur pengurus PMII Jepara
6	User	Materi	User dapat melihat materi dan mengunduh materi
7	User	Logout	User dapat melakukan Logout

2. Activity Diagram

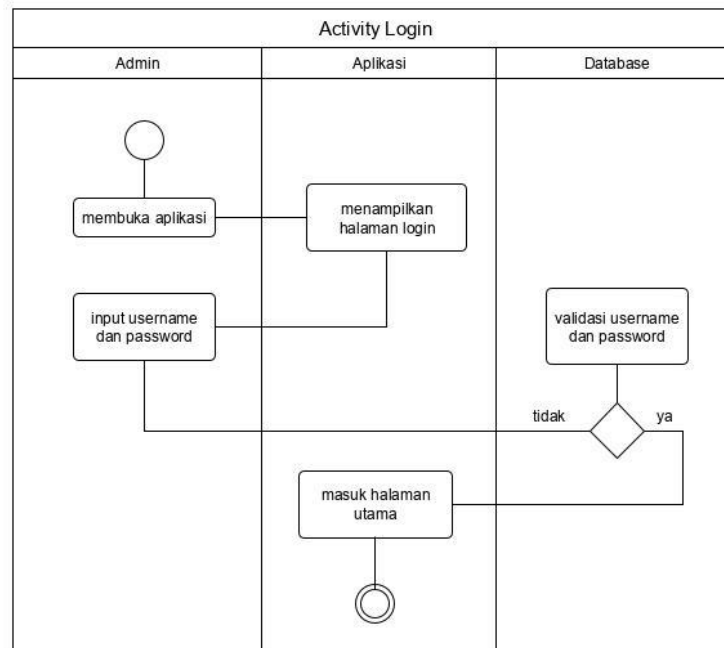
Activity Diagram merupakan diagram yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Berikut adalah activity diagram atau diagram aktivitas dari aplikasi E-PMII.

a. Activity Diagram Login

Activity Diagram Login berfungsi sebagai sistem keamanan, sehingga untuk bisa mengakses sistem harus melakukan login terlebih dahulu. Ketika login sistem akan memvalidasi data login benar atau salah. Setelah validasi sukses sistem akan masuk pada halaman utama.



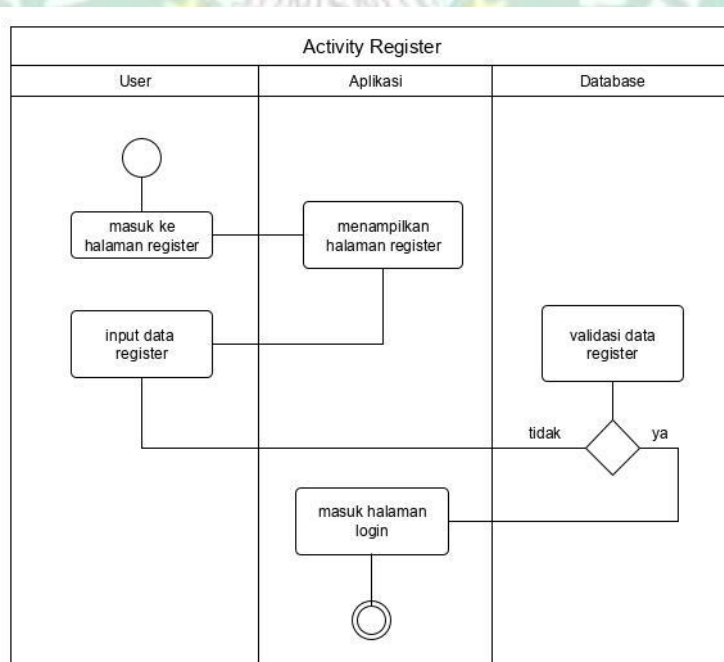
Gambar 4.2 Activity Diagram Login User



Gambar 4.3 Activity Diagram Login Admin

b. Activity Diagram Register

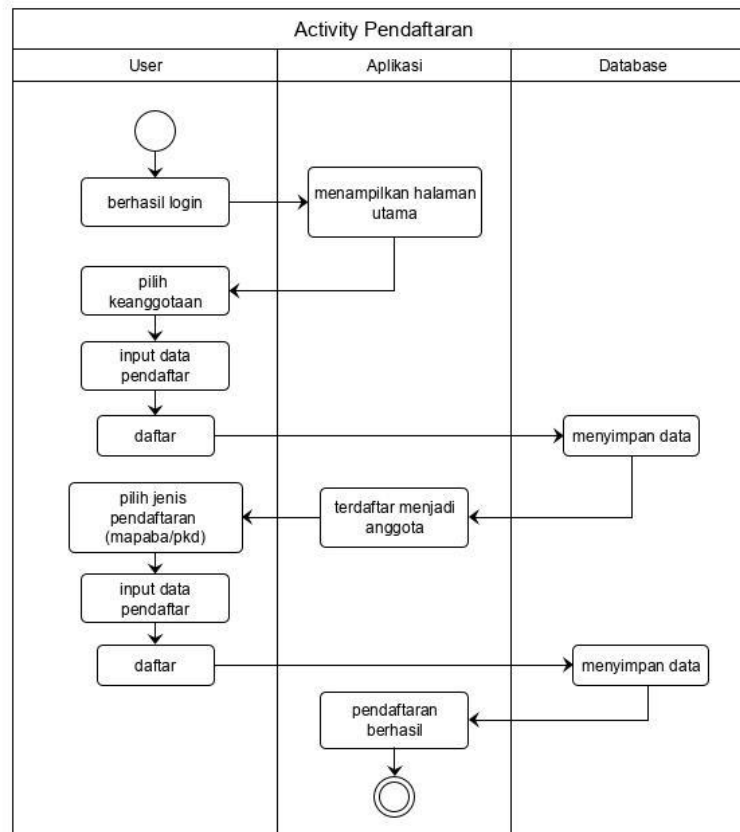
Activity Diagram register berfungsi sebagai proses pendaftaran akun untuk dapat melakukan login sebagai pengguna. Ketika pendaftaran login sistem akan masuk ke halaman login.



Gambar 4.4 Activity Diagram Register

c. *Activity Diagram Pendaftaran Kegiatan*

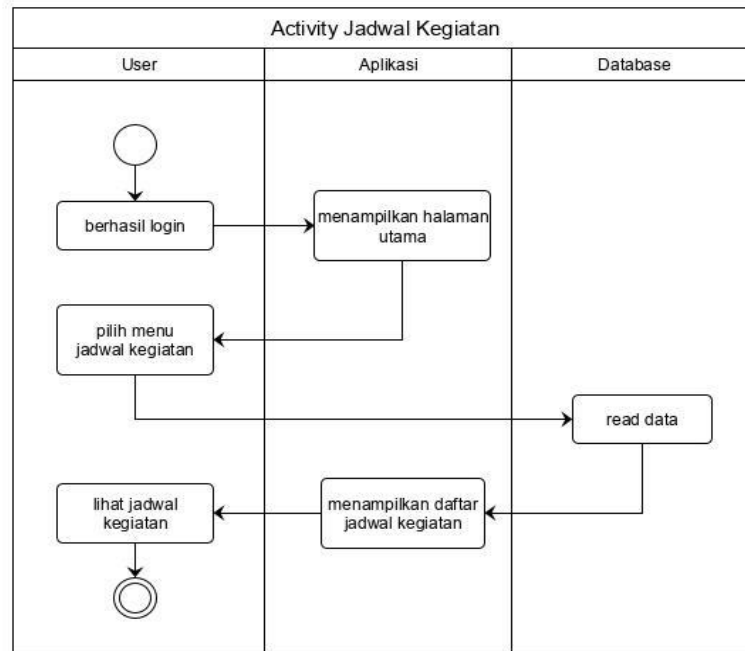
Activity Diagram Pendaftaran menjelaskan kegiatan user ketika mendaftar kegiatan.



Gambar 4.5 *Activity Diagram Pendaftaran Kegiatan*

d. *Activity Diagram Jadwal Kegiatan*

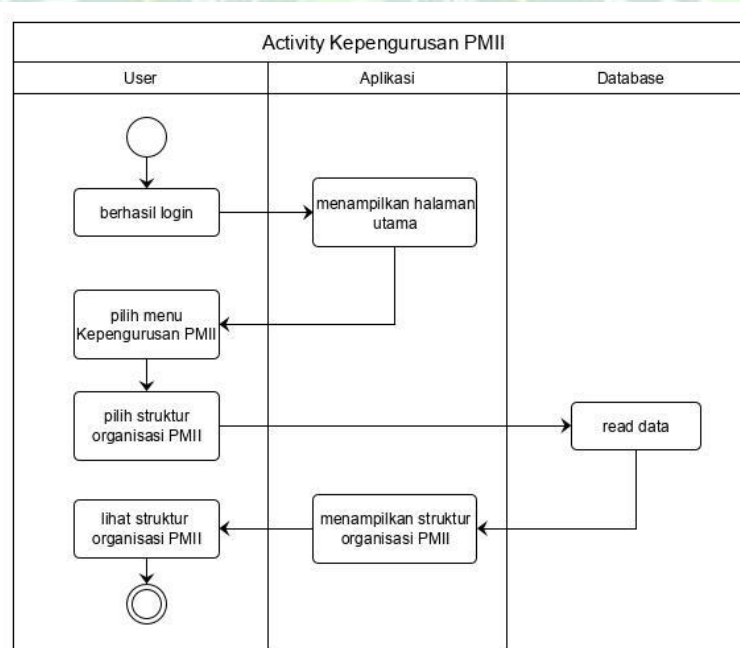
Activity Diagram ini menjelaskan tentang kegiatan ketika user melihat jadwal kegiatan.



Gambar 4.6 Activity Diagram Jadwal Kegiatan

e. Activity Diagram Kepengurusan PMII

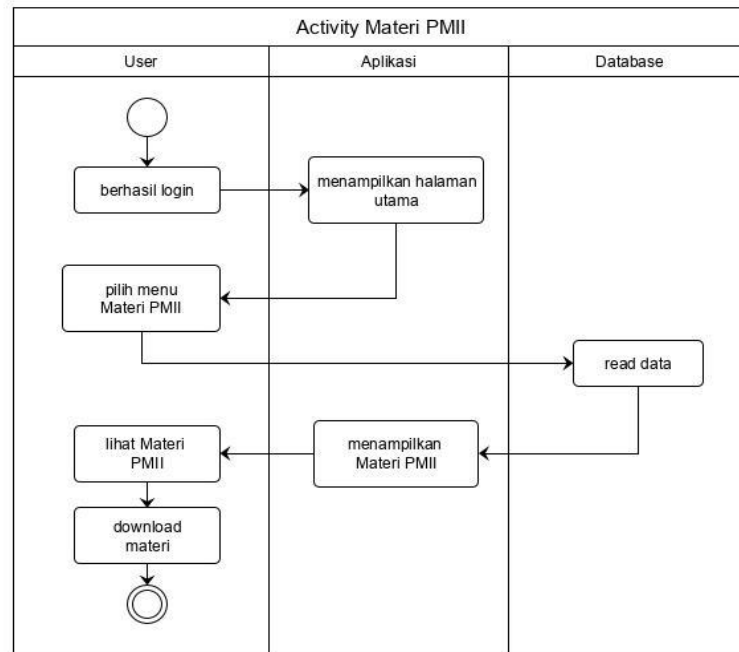
Pada Activity Diagram ini menjelaskan ketika user melihat struktur pengurus PMII Jepara.



Gambar 4.7 Activity Kepengurusan PMII

f. *Activity Diagram Materi*

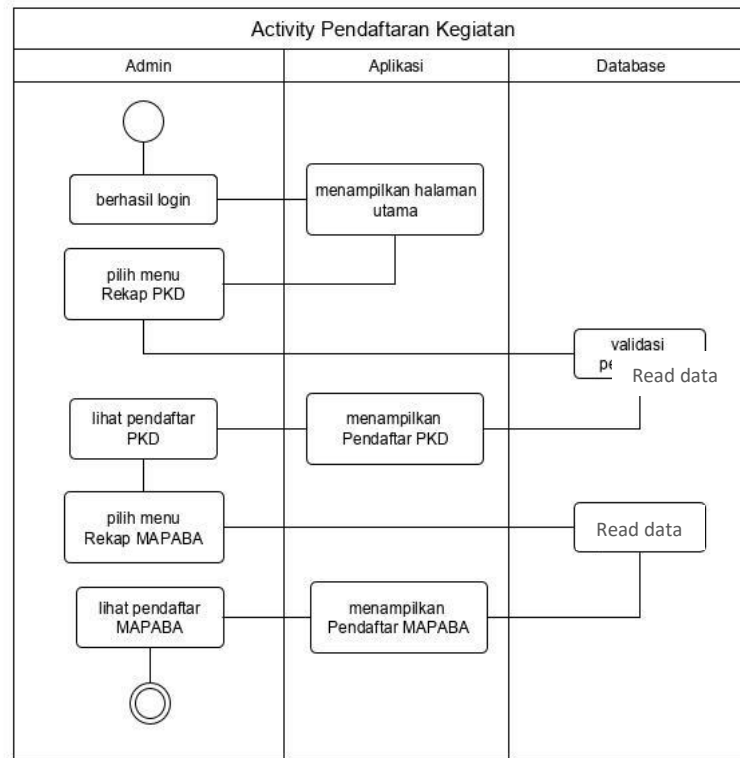
Activiy Diagram ini menjelaskan kegiatan user ketika melihat materi dan mendownload materi.



Gambar 4.8 *Activity Diagram Materi*

g. *Activity Diagram Rekap Data*

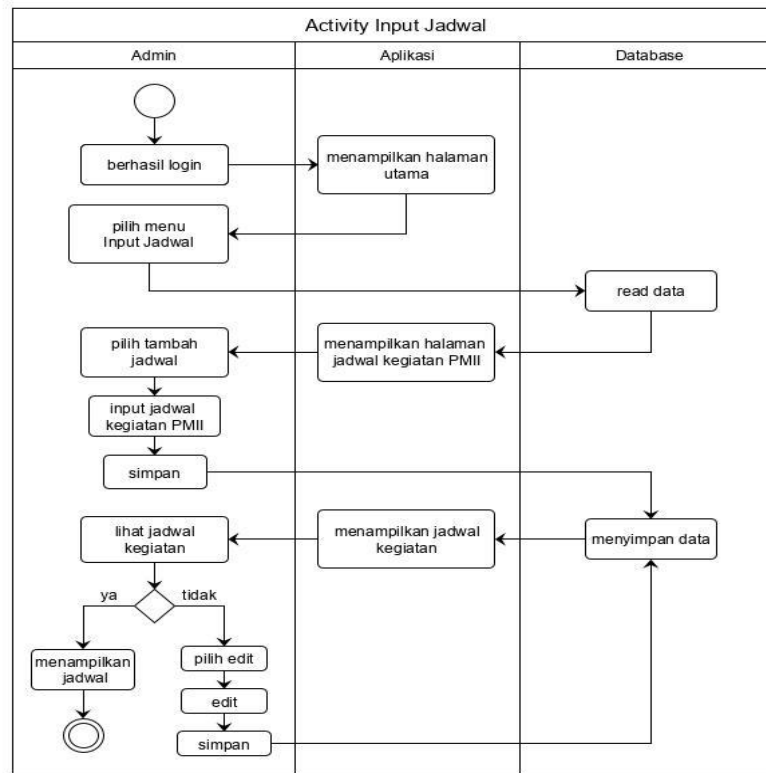
Pada *Activity Diagram Rekap Data* ini menjelaskan kegiatan admin ketika melihat rekap data pendaftar pkd dan mapaba.



Gambar 4.9 Activity Diagram Rekap Data

h. Activity Input Jadwal

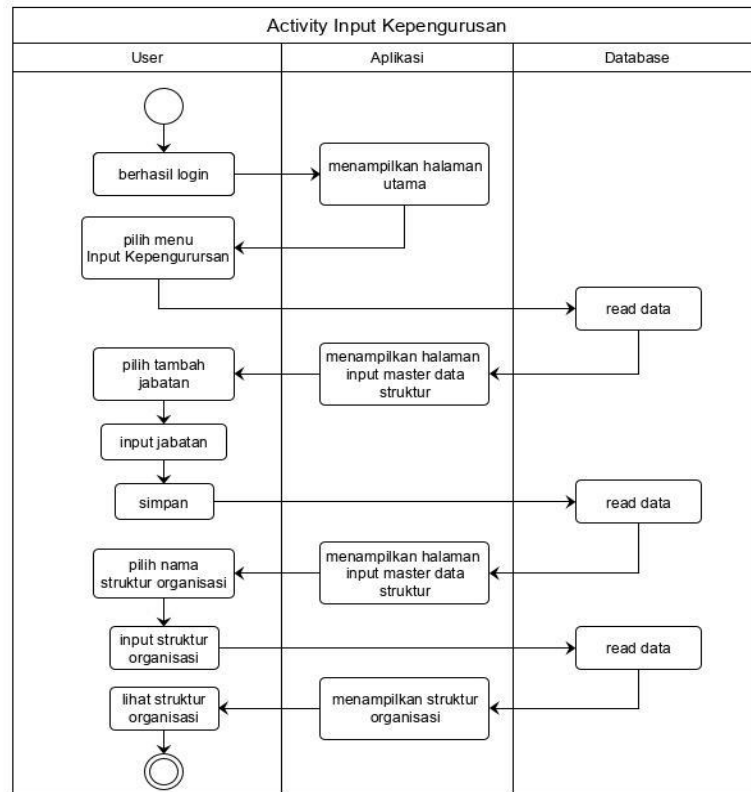
Pada Activity Diagram ini menjelaskan kegiatan admin dalam mengelola jadwal kegiatan seperti menambah jadwal, menedit jadwal dan menghapus jadwal kegiatan.



Gambar 4.10 Activity Diagram Input Jadwal

i. Activity Diagram Input Kepengurusan

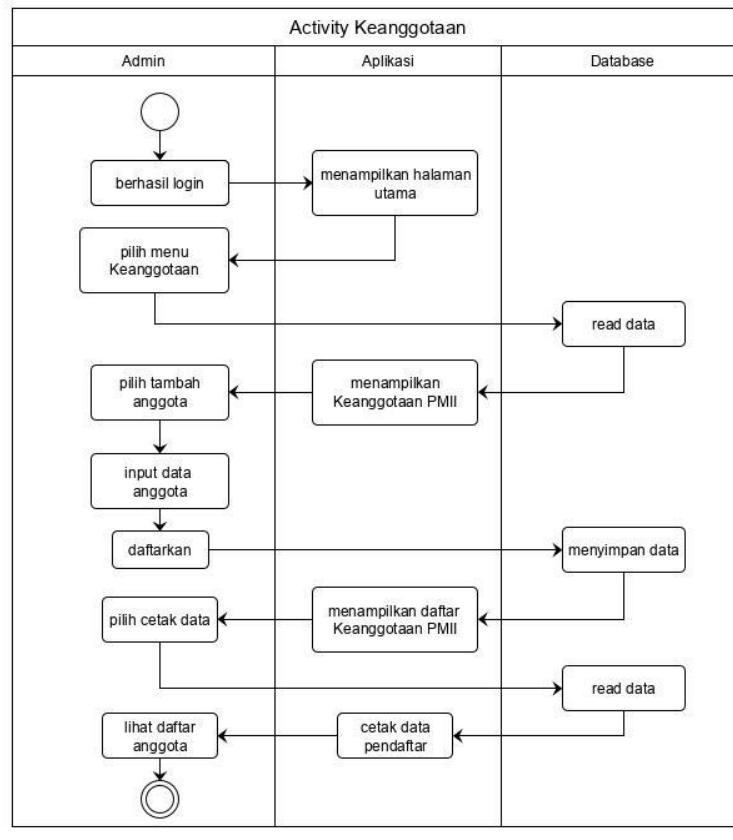
Activity Diagram Input Kepengurusan di kelola oleh admin yang menjelaskan kegiatan menambah jabatan pengurus, menginput nama pengurus dan melihat struktur pengurus.



Gambar 4.11 Activity Diagram Input Kepengurusan

j. Activity Diagram Keanggotaan

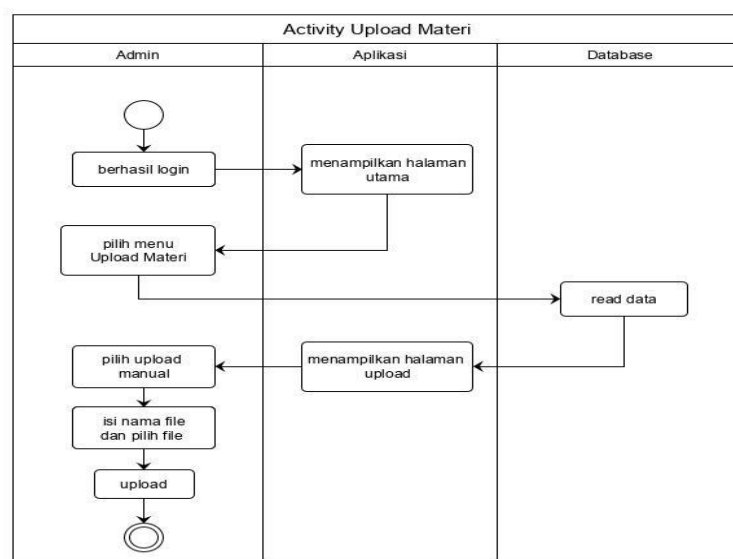
Pada Activity Diagram ini menjelaskan admin dalam mengelola data keanggotaan PMII seperti tambah anggota, mencetak data anggota dan melihat daftar anggota.



Gambar 4.12 Activity Diagram Keanggotaan

k. Activity Diagram Upload Materi

Pada Activity Diagram ini menjelaskan kegiatan admin ketika mengupload materi – materi kegiatan PMII.



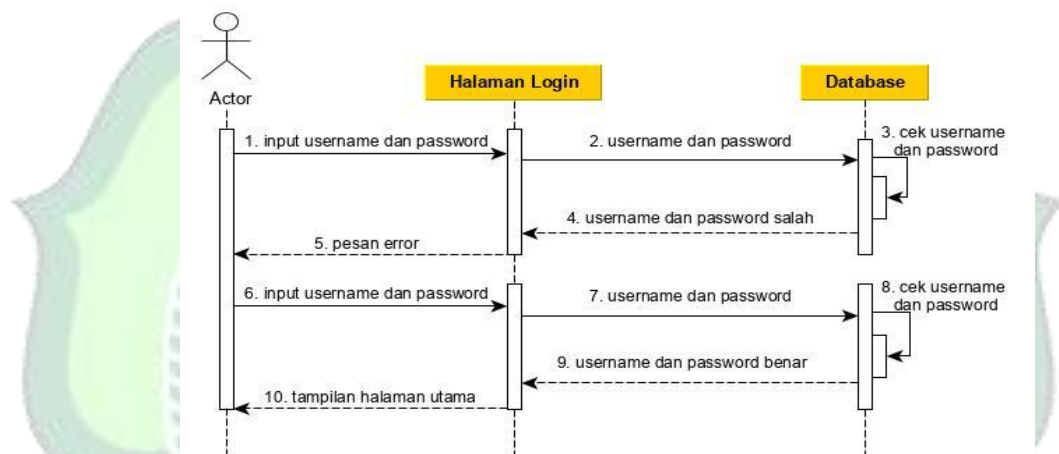
Gambar 4.13 Activity Diagram Upload Materi

3. Sequence Diagram

Sequence diagram merupakan penggambaran skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah elemen untuk menghasilkan keluaran tertentu, *sequence diagram* disusun berdasarkan urutan waktu. Berikut adalah *sequence diagram* admin dan user dari Aplikasi E-PMII.

a. *Sequence Diagram Login*

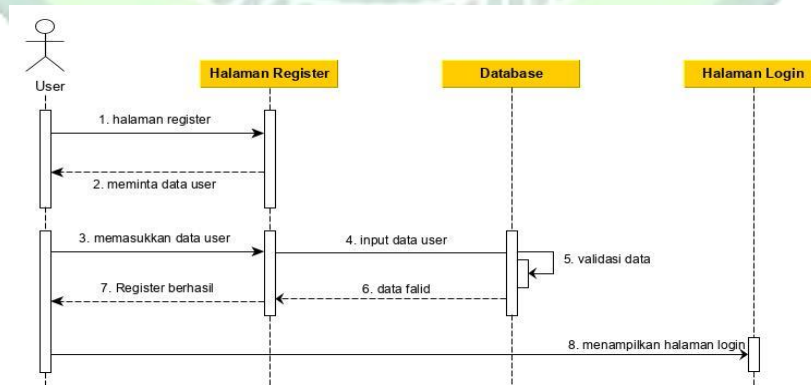
Pada proses *sequence* ini menggambarkan proses login yang dilakukan oleh admin dan user.



Gambar 4.14 *Sequence Diagram Login*

b. *Sequence Diagram Register*

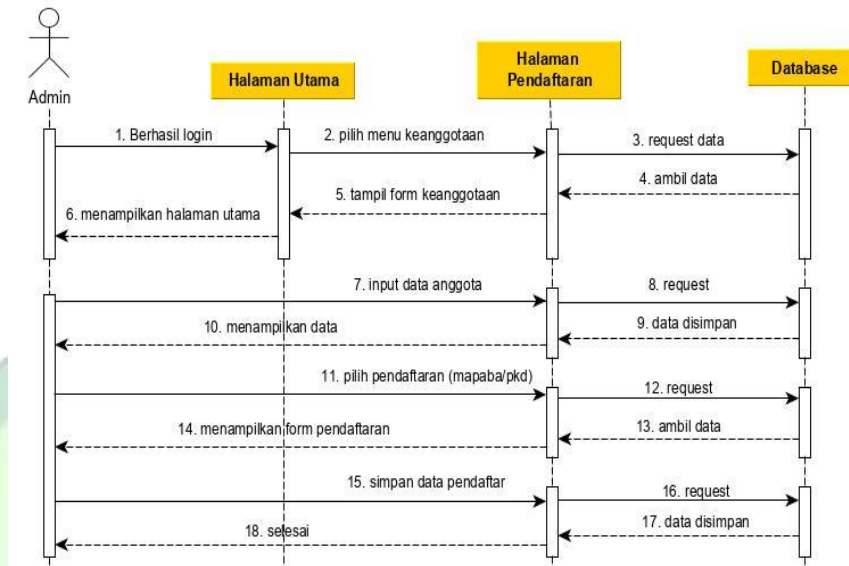
Pada proses *sequence* ini menggambarkan proses register yang dilakukan oleh user.



Gambar 4.15 *Sequence Diagram Register*

c. *Sequence Diagram* Pendaftaran Kegiatan

Pada *Sequence Diagram* ini menggambarkan proses pendaftaran kegiatan yang dilakukan oleh user.



Gambar 4.16 *Sequence Diagram* Pendaftaran Kegiatan

d. *Sequence Diagram* Jadwal Kegiatan

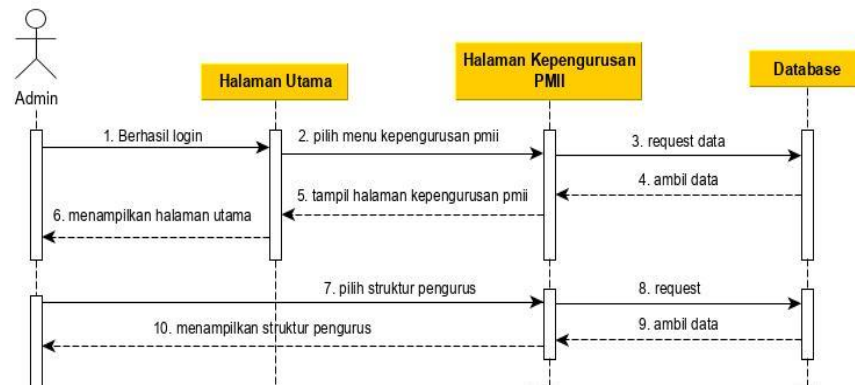
Pada *Sequence Diagram* ini menggambarkan proses user ketika melihat jadwal kegiatan.



Gambar 4.17 *Sequence Diagram* Jadwal Kegiatan

e. *Sequence Diagram* Kepengurusan

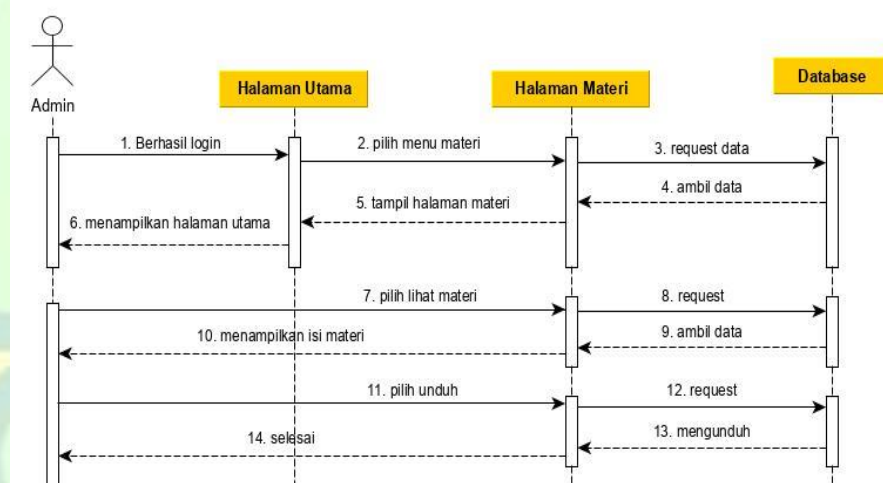
Pada *Sequence Diagram* ini menggambarkan ketika user melihat struktur organisasi oleh user.



Gambar 4.18 *Sequence Diagram* Kepengurusan

f. *Sequence Diagram* Materi

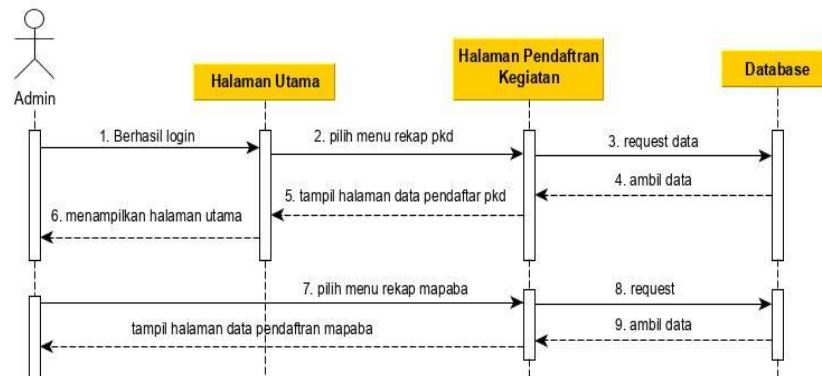
Sequence Diagram ini menggambarkan proses user ketika melihat jadwal kegiatan.



Gambar 4.19 *Sequence Diagram* Materi

g. *Sequence Diagram* Rekap Data

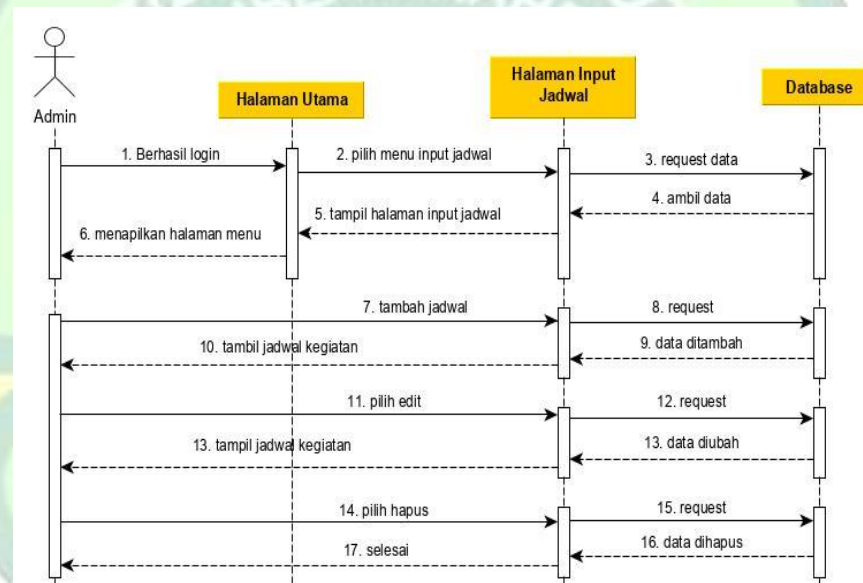
Pada *Sequence Diagram* ini menggambarkan proses admin ketika melihat rekap data pendaftar kegiatan mapaba dan pkd.



Gambar 4.20 *Sequence Diagram* Rekap Data

h. *Sequence Diagram* Input Jadwal

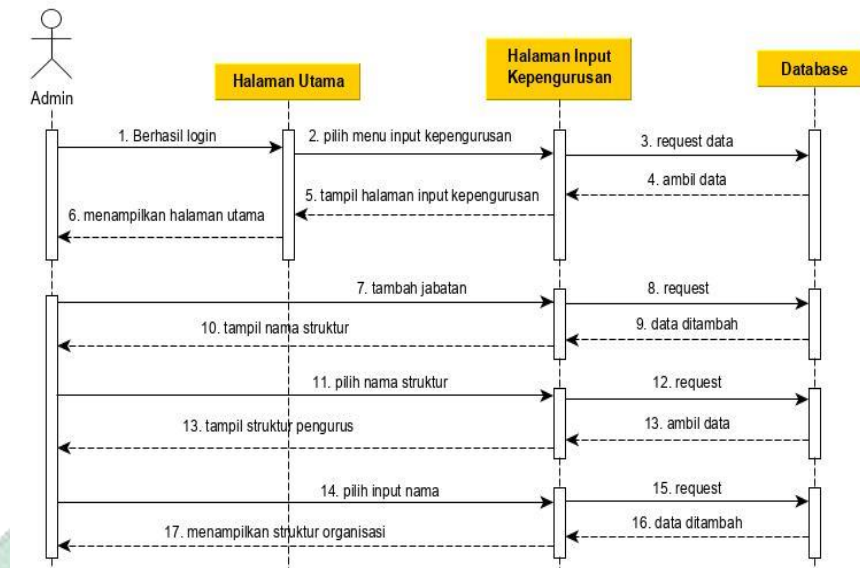
Sequence Diagram ini menggambarkan proses mengelola jadwal kegiatan oleh admin.



Gambar 4.21 *Sequence Diagram* Input Jadwal

i. *Sequence Diagram* Input Kepengurusan

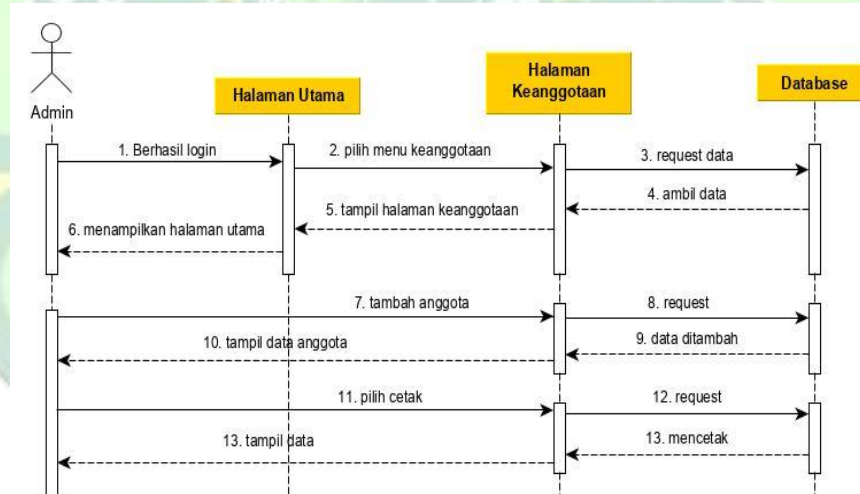
Pada gambar diagram ini menggambarkan proses pengelolaan struktur pengurus yang dilakukan oleh admin.



Gambar 4.22 *Sequence Diagram* Input Kepengurusan

j. *Sequence Diagram* Keanggotaan

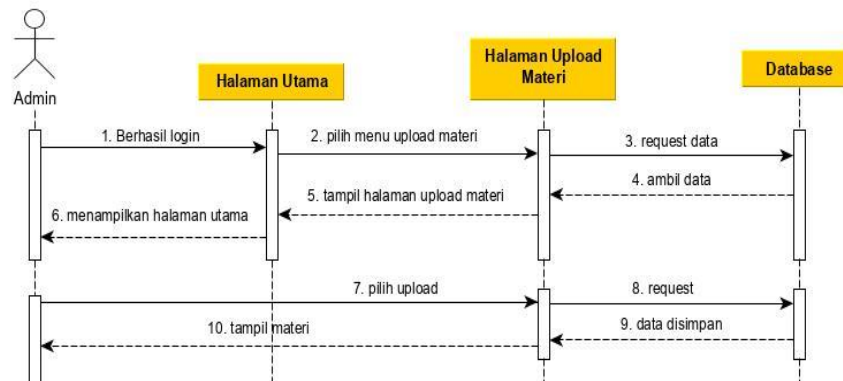
Pada diagram ini menggambarkan proses admin dalam mengelola data anggota yang telah terdaftar.



Gambar 4.23 *Sequence Diagram* Keanggotaan

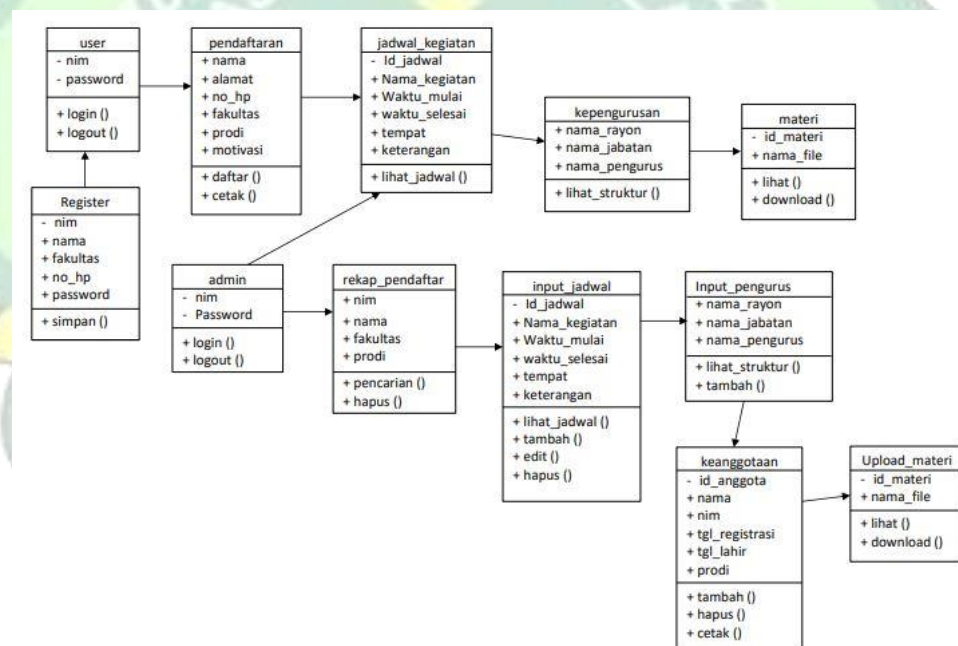
k. *Sequence Diagram* Upload Materi

Pada *Sequence Diagram* ini menggambarkan proses admin ketika melakukan upload materi.

Gambar 4.24 *Sequence Diagram* Upload Materi

4. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang ada pada sistem. Berikut adalah activity diagram atau diagram aktivitas dari aplikasi E-PMII :

Gambar 4.25 *Class Diagram*

4.1.2.2 Pembuatan Database MySQL

MySQL merupakan database yang digunakan dalam pengolahan data aplikasi E-PMII Jepara. Untuk membuat database baru, table dan juga menambahkan data ke dalam table dapat menggunakan sql query ataupun langsung menggunakan phpMyAdmin.

Ada dua jenis query pada sql yaitu query DDL (Data Definisien Language) dan query DML(Data Manipulation Language). query DDL digunakan untuk membuat table sedangkan query DML digunakan untuk mengolah data pada table.

```
--
-- Table structure for table `tb_struktur`
--

CREATE TABLE `tb_struktur` (
  `id_rayon` varchar(1) NOT NULL,
  `ketua` varchar(255) NOT NULL,
  `ketua2` varchar(255) NOT NULL,
  `sekretaris1` varchar(255) NOT NULL,
  `sekretaris2` varchar(255) NOT NULL,
  `bendahara1` varchar(255) NOT NULL,
  `bendahara2` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 ROW_FORMAT=COMPACT;
```

Gambar 4.26 DDL Query

Penjelasan: Data Definition Language (DDL) Query pembuatan table alumni yang mana *field* kolomnya berisi deskripsi dari struktur

```
INSERT INTO `tb_struktur` (`id_rayon`, `ketua`, `ketua2`, `sekretaris1`, `sekretaris2`, `bendahara1`, `bendahara2`) VALUES
('1', 'M. Syaifur Roqib', '', 'Putri Intan Nalumsari', 'Nova Siti Umayya', 'Nur Maila As Sa'adah', 'Hanik Rosidah'),
('2', 'Dimas Ananda Wijaya Al Gazali', '', 'Nur Rosi Alawiatun Nikmah', '', 'Vania Elifia Putri S', ''),
('3', 'Dimas Rizal Maulana', '', 'Linda Dewi Lestari', 'Rani Cholifah', 'Uswatun Khasanah', ''),
('4', 'M. Taksinul Khuluq', '', 'Khoirum Rodhiatul Ifa', 'Khoirul Ummah', 'Ela Shoflanah', 'Fadhilatul Amna Assafira'),
('5', 'Syahrul Juniar Setiawan', 'Muhammad Andi Setiawan', 'Wahyu Nugroho', '', 'Yuli Susanti', '');
```

Gambar 4.27 DML Query

Penjelasan: Data Manipulation Language (DML) query menambahkan data alumni kedalam table struktur yang telah dibuat sebelumnya.

Table Name	Fields and Data Types
db_pmi.tb_staff	id_staff: int(1), id_rayon: varchar(1), tingkat_jabatan: varchar(1), nama_jabatan: varchar(255), jenis_jabatan: varchar(255), nama: varchar(255)
db_pmi.tb_struktur	id_rayon: varchar(1), Ketua: varchar(255), Ketua2: varchar(255), sekretaris1: varchar(255), sekretaris2: varchar(255), bendahara1: varchar(255), bendahara2: varchar(255)
db_pmi.tb_jadwal	id_jadwal: int(11), kegiatan: varchar(255), tempat: varchar(255), waktu_mulai: date, waktu_selesai: date, status: enum('belum terlaksana','selesai')
db_pmi.tb_anggota	id_anggota: int(12), nim: varchar(50), tgl_registrasi: date, tempat_lahir: varchar(255), tgl_lahir: date, prodi: varchar(255), jk: varchar(255), email: varchar(255), alamat: varchar(255)
db_pmi.rayon	id_rayon: int(1), nm_rayon: varchar(50), singkatan_rayon: varchar(12), komisariat: varchar(25), id_fakultas: varchar(1)
db_pmi.kegiatan	id_kegiatan: int(11), nama_kegiatan: varchar(255), keterangan: varchar(255)
db_pmi.tb_user	nim: varchar(12), password: varchar(255), nama: varchar(50), no_hp: varchar(13), level: varchar(1), fakultas: varchar(1)
db_pmi.tb_materi	id_materi: int(255), nama_file: varchar(255), link: varchar(255)
db_pmi.pkd	id_pkd: int(11), nim: varchar(25), cabang: varchar(25), komisariat: varchar(255), motivasi: varchar(255), image: varchar(255), tgl_lahir: date
db_pmi.prodi	id_prodi: int(11), id_fakultas: int(2), nm_prodi: varchar(50), singkatan_prodi: varchar(50)
db_pmi.mapada	id_mapada: int(11), nim: varchar(14), prodi: varchar(50), motto: varchar(255), image: varchar(255)
db_pmi.jabatan	id_parent: int(5), id_rayon: varchar(1), jabatan: varchar(255)
db_pmi.tb_staff_new	id_staff: int(1), id_rayon: varchar(1), id_parent: varchar(255), nama_staff: varchar(255)
db_pmi.fakultas	id_fakultas: int(11), nama_fakultas: varchar(50), singkatan_fakultas: varchar(10)

Gambar 4.28 Database

4.1.2.3 Pembuatan Rest Api

Data MySQL dikonversi kedalam bentuk JSON agar dapat digunakan dalam aplikasi android. Untuk pemanggilan database melalui android maka dibuatkan Api. Api dibuat menggunakan bahasa pemrograman php dan menggunakan framework Codeigniter. Sebelum membuat Api, pastikan codeigniter sudah terkonfigurasi dengan database yang dipakai. Komponen pembuatan Api yaitu Model dan Controller, Model sebagai tempat *querying database* dimana class nya di *extend* ke *CI_Model* dan Controller sebagai tempat pengolahan data dimana class nya di *extend* ke *CI_Controller*. Fungsi-fungsi yang terdapat pada class controller nantinya yang dapat dipanggil dan outputnya nanti berupa file JSON.

```

67 $active_group = 'default';
68 $query_builder = TRUE;
69
70
71
72 $db['default'] = array(
73     'dsn' => '',
74     'hostname' => 'localhost',
75     'username' => 'root',
76     'password' => 'password',
77     'database' => 'satabase_name',
78     'dbdriver' => 'mysqli',
79     'dbprefix' => '',
80     'pconnect' => FALSE,
81     'db_debug' => (ENVIRONMENT !== 'production'),
82     'cache_on' => FALSE,
83     'cachedir' => '',
84     'char_set' => 'utf8',
85     'dbcollat' => 'utf8_general_ci',
86     'swap_pre' => '',
87     'encrypt' => FALSE,
88     'compress' => FALSE,
89     'stricton' => FALSE,
90     'fallover' => array(),
91     'save_queries' => TRUE
92 );

```

Gambar 4.29 Konfigurasi Data Codeigniter

Penjelasan: Konfigurasi database melalui *framework codeigniter* yang terletak pada folder `application > config > database.php`. Yang perlu diubah adalah data hostname atau nama hostnya, jika masih dalam server maka menggunakan nama localhost, kemudian username untuk username host nya, kemudian password, jika tidak menggunakan password maka dibiarkan kosong saja, dan yang terakhir yaitu nama database.

```

4 class M_anggota extends CI_Model {
5
6     public function getanggota()
7     {
8         $b = $this->db->query("SELECT * FROM tb_anggota a, tb_user b where a.nim = b.nim AND a.nim NOT IN (SELECT
9             nim FROM tb_anggota WHERE nim=1)");
10        return $b;
11    }
12
13    public function select_join_anggota($where){
14        // $data = array('tb_user.nim' => $where);
15
16        $this->db->select('*');
17        $this->db->from('tb_user');
18        $this->db->join('tb_anggota', 'tb_user.nim=tb_anggota.nim');
19        $this->db->where('tb_user.nim', $where);
20        return $this->db->get()->row();
21    }
22
23    public function get_user($id){
24        $data = array(
25            'nim' => $id
26        );
27        return $this->db->get_where('tb_user', $data)->row();
28    }
29
30    public function get_fakultas_by_id($id){
31        $where = array(
32            'nim' => $id
33        );
34
35        $this->db->select('*');
36        $this->db->from('tb_user');

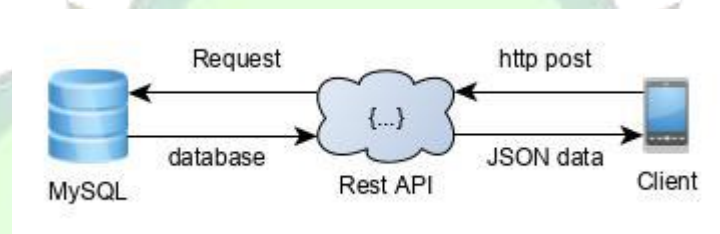
```

Gambar 4.30 Model Controller

Penjelasan: class `M_anggota` berisi *function* yang digunakan untuk querying database pada table anggota. Karena menggunakan *framework codeigniter* maka query nya tidak perlu diketik lagi.

4.1.2.4 Pemanggilan Api

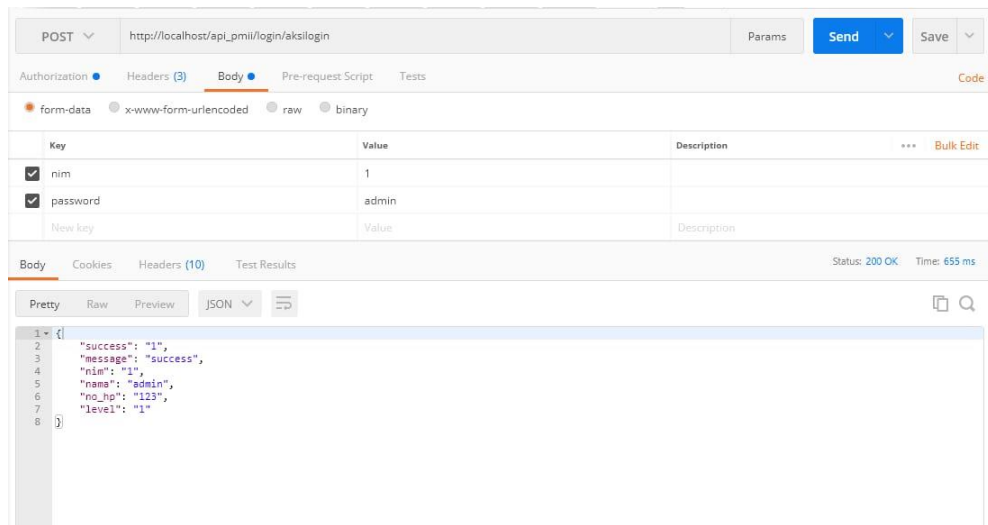
MySQL yang dikirim ke device android harus berupa JSON data, sehingga antara database MySQL dan device android terdapat rest api sebagai tempat pemanggilan dan pengolahan database MySQL ke JSON. Hubungan antara MySQL, Rest API dan Android adalah sebagai berikut :



Gambar 4.31 Rest API

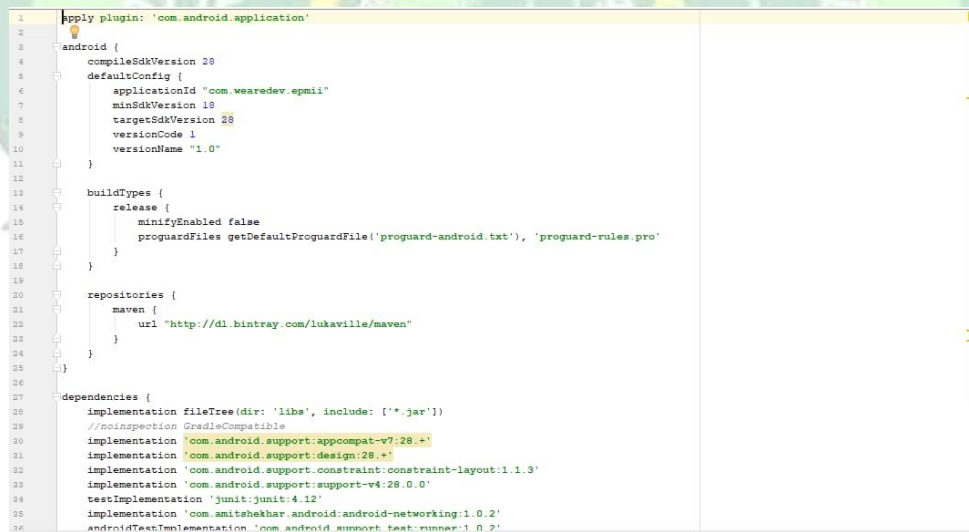
Android mengirimkan http dengan method post kepada rest api, rest api mengakses database MySQL lalu mengolahnya menjadi JSON data, dan JSON data tersebut dikirim ke *client* nya yaitu Android.

Sebelum dipanggil di android studio perlu di cek terlebih dahulu menggunakan aplikasi postman, dengan memasukkan url dan juga body parameter. Misalkan untuk memanggil data dengan memasukkan url `http://localhost/api_pmii/login/aksilogin` dengan method post dan body parameter `nim : 1, password : admin` lalu klik tombol send akan muncul hasilnya yaitu array result yang dikonversi ke JSON.



Gambar 4.32 Test Postman

Untuk pemanggilan di android studio menggunakan plugin android networking yang ditambahkan pada file build.gradle.



Gambar 4.33 Build Gradle

Pemanggilan pada activity dengan menambahkan code
 “AndroidNetworking.post()”

```

private void cekUser(){
    String nim      = etNIM.getText().toString().trim();
    String password = etPass.getText().toString().trim();

    pDialog = new ProgressDialog( context this);
    pDialog.setCancelable(false);
    pDialog.setTitle("Login Proses");
    pDialog.setMessage("Menghubungkan ke server . . . ");
    showDialog();

    AndroidNetworking.post(stringConfig.LOGIN)
        .addBodyParameter("nim", nim)
        .addBodyParameter("password", password)
        .setTag("test")
        .setPriority(Priority.HIGH)
        .build()
        .getAsJSONObject(new JSONObjectRequestListener() {
            @Override
            public void onResponse(JSONObject response) {
                hideDialog();
                // do anything with response
                int success = response.optInt( name: "success");
                Toast.makeText(getApplicationContext(), text: "ok 2",Toast.LENGTH_SHORT).show();
                if (success==1){
                    String nim      = response.optString( name: "nim");
                    String nama     = response.optString( name: "nama");
                    String no_hp    = response.optString( name: "no_hp");
                    String level    = response.optString( name: "level");
                }
            }
        });
}

```

Gambar 4.34 Android Networking

```

1  package com.wearedev.epmil.Config;
2
3  import ...
11
12  public class SessionManager {
13      SharedPreferences sharedPreferences;
14      public SharedPreferences.Editor editor;
15      public Context context;
16      int PRIVATE_MODE = 0;
17
18      private static final String PREF_NAME = "LOGIN";
19      private static final String LOGIN = "IS_LOGIN";
20
21      public static final String NIM_SES = "nim";
22      public static final String NAMA_SES = "nama";
23      public static final String HP_SES = "no_hp";
24      public static final String LVL_SES = "level";
25
26  @
27      public SessionManager(Context context){
28          this.context = context;
29          sharedPreferences = context.getSharedPreferences(PREF_NAME, PRIVATE_MODE);
30          editor = sharedPreferences.edit();
31      }
32
33      public void createSession(String nim, String nama,String no_hp,String level){
34          editor.putBoolean(LOGIN, true);
35          editor.putString(NIM_SES, nim);
36          editor.putString(NAMA_SES, nama);
37          editor.putString(HP_SES, no_hp);
38          editor.putString(LVL_SES, level);
39          editor.apply();
40      }
41
42      public boolean isLoggin() { return sharedPreferences.getBoolean(LOGIN, defValue: false); }

```

Gambar 4.35 Scraping Api

4.1.3 *Implementation (implementasi)*

Berdasarkan perancangan aplikasi pada tahap sebelumnya, maka peneliti mengimplementasikan dalam bentuk aplikasi E-PMII. Berikut implementasi sistem aplikasi E-PMII Jepara :

1. Tampilan *Shortcut* Aplikasi

Shortcut merupakan tombol cepat untuk membuka aplikasi ini. *Shortcut* aplikasi ini berupa *icon* logo pmii pada aplikasi. Berikut adalah tampilan *shortcut* aplikasi ePMII.



Gambar 4.36 *Shortcut* Aplikasi

2. Tampilan *Splash Screen*

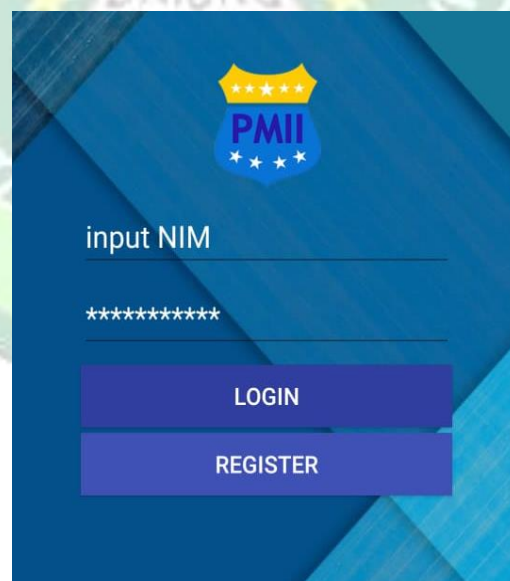
Splash Screen adalah halaman yang pertama kali muncul setelah pengguna menekan *icon* aplikasi. *Splash Screen* ini hanya berisi logo dari aplikasi ePMII.



Gambar 4.37 Tampilan *Splash Screen*

3. Tampilan Halaman Login

Halaman login merupakan halaman utama yang muncul sebelum admin atau user membuka sistem aplikasi. Pada halaman login admin atau user harus menginputkan username dan password yang telah terdaftar pada sistem aplikasi.



Gambar 4.38 Tampilan Halaman Login

- *Text Input NIM* : Pengguna harus mengisi *Email* dari pengguna.
Contoh : 151240000425
- *Text Field Password* : Pengguna harus mengisi Password yang berupa angka atau huruf yang minimal terdiri dari 6 karakter.

Tombol *Login* : Ketika pengguna menekan tombol *Login* maka sistem akan langsung mencocokkan kedalam *database* apakah *Email* dan *password* cocok atau tidak. Jika *Email* dan *password* sesuai dengan data di *firebase* maka sistem akan langsung menuju menu Home. Berikut potongan *source code* tampilan halaman login :

```

1 package com.wearedev.epmii;
2
3 import ...
4
24
25 public class Login extends AppCompatActivity {
26
27     ProgressDialog progressDialog;
28     Button btLogin, btDaftar;
29     EditText etNIM, etPass;
30     SessionManager sm;
31     boolean doubleBackToExitPressedOnce = false;
32
33
34 @Override
35 protected void onCreate(Bundle savedInstanceState) {
36     super.onCreate(savedInstanceState);
37     setContentView(R.layout.activity_login);
38     etNIM = (EditText) findViewById(R.id.etNIM);
39     etPass = (EditText) findViewById(R.id.etpassword);
40     btLogin = (Button) findViewById(R.id.btLogin);
41     btDaftar = (Button) findViewById(R.id.btDaftar);
42
43     /*deklarasasi session*/
44     sm = new SessionManager(getApplicationContext());
45
46     btDaftar.setOnClickListener((v) -> {
47         //startActivity(new Intent(getApplicationContext(), register.class));
48         startActivity(new Intent(getApplicationContext(), register_web.class));
49         finish();
50     });
51
52
53     btLogin.setOnClickListener((view) -> cekUser());
54
55 }
56
57
58
59
60
61

```

Gambar 4.39 *Source Code* Halaman Login

4. Tampilan Halaman Register

Perancangan Halaman Register ini merupakan halaman yang akan muncul jika Pengguna belum memiliki akun. Di halaman ini Pengguna dipersilahkan untuk mengisi NIM, nama, fakultas dan nomor *handphone*. Rancangan tampilannya sebagai berikut:

FORM REGISTRASI

Nomor Induk Mahasiswa

Nama Lengkap

fakultas

no Hp

password

ulangi password

Gambar 4.40 Tampilan Halaman Register

- *Text Field* NIM : Pengguna harus mengisikan *NIM* dari pengguna. Contoh : 151240000425
- *Text Field* Nama Lengkap : Pengguna harus mengisikan Nama Lengkap pengguna.
- *Text Field* Fakultas : Pengguna harus mengisikan Fakultas asal pengguna.
- *Text Field* No. *Handphone* : Pengguna harus mengisikan No. *Handphone* dari pengguna yang berisi angka-angka.
- *Text Field* Password : Pengguna harus mengisikan Password yang berupa angka atau huruf yang minimal terdiri dari 6 karakter.
- *Text Field* Ulangi Password : Pengguna harus mengisikan ulang Password yang berupa angka atau huruf yang minimal terdiri dari 6 karakter.
- Tombol Simpan : jika Pengguna menekan tombol Simpan maka sistem akan menyimpan data kedalam database. Berikut potongan *source code* halaman register :

```

1 package com.wearedev.epmii;
2
3 import ...
4
25 public class Register extends AppCompatActivity {
26     EditText nim, nama_lengkap, HP, etPasswordREGISTER, etREPasswordREGISTER;
27     Button btRegister, btLogin;
28     Spinner spFakultas;
29
30     @Override
31     protected void onCreate(@Nullable Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_register);
34
35         btRegister = (Button) findViewById(R.id.btREGISTER);
36         btLogin = (Button) findViewById(R.id.btLOGIN);
37         nim = (EditText) findViewById(R.id.etNIM);
38         nama_lengkap = (EditText) findViewById(R.id.etNAMELENGKAP);
39         HP = (EditText) findViewById(R.id.etHP);
40         etPasswordREGISTER = (EditText) findViewById(R.id.etPasswordREGISTER);
41         etREPasswordREGISTER = (EditText) findViewById(R.id.etREPasswordREGISTER);
42         //SPINNER
43         spFakultas = (Spinner) findViewById(R.id.spinnerFakultas);
44
45         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
46
47
48         // Create an ArrayAdapter using the string array and a default spinner layout
49         ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(context, this,
50             R.array.fakultas, android.R.layout.simple_spinner_item);
51         // Specify the layout to use when the list of choices appears
52         adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
53         // Apply the adapter to the spinner
54         spFakultas.setAdapter(adapter);
55

```

Gambar 4.41 Source Code Register

5. Tampilan Halaman Anggota

Halaman Anggota merupakan halaman yang menampilkan form registrasi anggota.

The screenshot displays the registration form for the ePMII application. The form is titled "FORM REGISTRASI ANGGOTA" and contains the following fields:

- NIM:** 8945
- Nama Lengkap:** Susanti susanti
- Alamat:** Tempat Lahir
- Email:** Email@email.xxx
- Tempat Lahir:** Tempat Lahir
- Tanggal Lahir:** tgl, bul, thn
- Jenis Kelamin:** Choose...
- Fakultas:** (Dropdown menu)

Gambar 4.42 Registrasi Anggota

- *Text Field* NIM : Pengguna harus mengisikan NIM dari pengguna.
Contoh : 151240000425
- *Text Field* Nama Lengkap : Pengguna harus mengisikan Nama Lengkap pengguna.
- *Text Field* Alamat : Pengguna harus mengisikan alamat tinggal pengguna
- *Text Field* Email : Penggua harus mengisikan *Email* pengguna.
Contoh : namapengguna@gmail.com
- *Text Field* Tempat Lahir : Pengguna harus mengisikan tempat lahir pengguna
- *Text Field* Tanggal Lahir : Pengguna harus mengisikan tanggal lahir pengguna
- *Text Field* Fakultas : Pengguna harus mengisikan Fakultas asal pengguna

Berikut adalah potongan *source code* halaman register anggota :

```

1 package com.wearedev.epmi;
2
3 import ...
4
5
6
7
8
9
10
11
12
13
14
15 public class AnggotaFragment extends Fragment {
16     SessionManager sm;
17
18     @Nullable
19     @Override
20     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
21         View v = inflater.inflate(R.layout.fragment_anggota, container, attachToRoot false);
22         //deklarasi session*/
23         sm = new SessionManager(getContext());
24         HashMap<String,String> data = sm.getUserDetail();
25         String nim = data.get(sm.NIM_SES);
26
27         WebView webView = (WebView)v.findViewById(R.id.webview);
28         webView.getSettings().setLoadsImagesAutomatically(true);
29         webView.getSettings().setJavaScriptEnabled(true);
30         webView.getSettings().setDomStorageEnabled(true);
31         webView.setOnKeyListener((v, keyCode, event) -> {
32             if(event.getAction() == KeyEvent.ACTION_DOWN)
33             {
34                 WebView webView = (WebView) v;
35
36                 switch (keyCode)
37                 {
38                     case KeyEvent.KEYCODE_BACK:
39                         if (webView.canGoBack())
40                         {
41                             webView.goBack();
42                             return true;
43                         }
44                         break;
45
46                 }
47             }
48         });
49     }
50 }
51
52
53

```

Gambar 4.43 *Source Code* Register Anggota

6. Tampilan Halaman Kepengurusan

Halaman Kepengurusan merupakan halaman yang menampilkan struktur organisasi.



Gambar 4.44 Tampilan Struktur Organisasi

Pada tampilan ini pengguna dapat melihat struktur pengurus sesuai dengan nama struktur.

- Rayon Dakwah dan Komunikasi : menampilkan struktur pengurus yang berasal dari mahasiswa Fakultas Dakwah Dan Komunikasi.
- Rayon R.A Kartini : menampilkan struktur pengurus yang berasal dari mahasiswa Fakultas Ekonomi Dan Bisnis.
- Rayon Ratu Shima : menampilkan struktur pengurus yang berasal dari Fakultas Sains Dan Teknologi.
- Rayon Syariah Dan Hukum : menampilkan struktur pengurus yang berasal dari mahasiswa Fakultas Syariah Dan Hukum.
- Rayon Tarbiyah dan Ilmu Keguruan : menampilkan struktur pengurus yang terdiri dari mahasiswa Fakultas Tarbiyah Dan Ilmu Keguruan.

Berikut adalah potongan *source code* tampilan struktur organisasi :

```

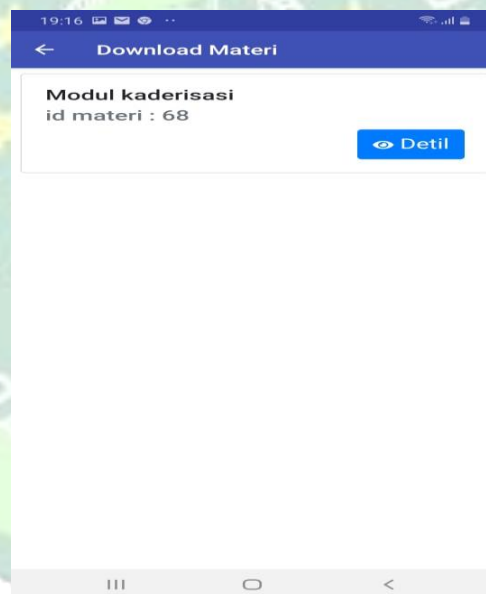
1 package com.waaredev.epmi;
2
3 import ...
4
5 public class kepengurusanFragment extends Fragment {
6     @Nullable
7     @Override
8     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
9
10        View v = inflater.inflate(R.layout.fragment_kepengurusan, container, attachToRoot false);
11        WebView webView = (WebView)v.findViewById(R.id.webview);
12        webView.getSettings().setLoadsImagesAutomatically(true);
13        webView.getSettings().setJavaScriptEnabled(true);
14        webView.getSettings().setDomStorageEnabled(true);
15        webView.setOnKeyListener((v, keyCode, event) -> {
16            if(event.getAction() == KeyEvent.ACTION_DOWN)
17            {
18                WebView webView = (WebView) v;
19
20                switch(keyCode)
21                {
22                    case KeyEvent.KEYCODE_BACK:
23                        if(webView.canGoBack())
24                        {
25                            webView.goBack();
26                            return true;
27                        }
28                        break;
29                }
30            }
31            return false;
32        });
33    }
34}

```

Gambar 4.45 Source Code Kepengurusan

7. Tampilan Materi

Halaman ini menampilkan materi-materi yang di upload oleh admin dan user dapat melihat serta mendownload materi.



Gambar 4.46 Tampilan Halaman Materi

Pada tampilan ini pengguna dapat melihat dengan detail materi yang ingin di lihat serta pengguna dapat mengunduh materi sesuai dengan kebutuhan.

- Modul kaderisasi : merupakan contoh judul materi yang ditampilkan.
- Detil : pengguna dapat melihat secara detil materi serta mendownload materi.

Berikut adalah potongan *source code* halaman materi :

```

13 import ..
14
15 public class Materi extends AppCompatActivity {
16     WebView webView;
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_materi);
21
22         getSupportActionBar().setTitle("Download Materi");
23         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
24         content();
25     }
26
27     public void content()
28     {
29         webView = (WebView)findViewById(R.id.webview);
30         webView.getSettings().setLoadsImagesAutomatically(true);
31         webView.getSettings().setJavaScriptEnabled(true);
32         webView.getSettings().setDomStorageEnabled(true);
33         webView.setDownloadListener((url, userAgent, contentDisposition, mimetype, contentLength) -> {
34             Intent i = new Intent(Intent.ACTION_VIEW);
35             i.setData(Uri.parse(url));
36             startActivity(i);
37         });
38         // Baris di bawah untuk menambahkan scrollbar di dalam WebView-nya
39         webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
40         webView.setWebViewClient(new WebViewClient());
41         webView.loadUrl(stringConfig.MATERI);
42     }
43
44     @Override
45     public void onBackPressed() {
46         startActivity(new Intent(getApplicationContext(), MainActivity.class));
47     }
48 }

```

Gambar 4.47 *Source Code* Materi

8. Tampilan Halaman PKD

Tampilan ini menampilkan form pendaftaran kegiatan PKD

The screenshot shows a mobile application interface with a blue header bar containing the text 'ePMII'. Below the header is a white box titled 'FORM REGISTRASI PKD'. Inside this box, there are four text input fields: 'Nama Lengkap' with the value 'Khois', 'Asal Cabang' with the value 'jepara', 'komisariat' with the value 'Komisariat Sulthan Hadlirin', and 'motivasi' which is currently empty. At the bottom of the form is a blue button with the text 'DAFTAR'. The background of the application is a light blue and white pattern.

Gambar 4.48 Tampilan Pendaftaran

- *Text Field* Nama Lengkap : Pengguna harus mengisi nama lengkap pengguna.
- *Text Field* Asal Cabang : Pengguna harus mengisi asal cabang daerah anggota.

Contoh : Cabang Surakarta

- *Text Field* Komisariat : Pengguna harus mengisi komisariat asal pengguna.

Contoh : Komisariat Sunan Muria

- *Text Field* Motivasi : Pengguna harus mengisi motivasi pengguna mengikuti kegiatan tersebut.

Berikut adalah source code halaman pendaftaran pkd :

```

1 package com.wearedev.epmii;
2
3 import ...
44
45 public class PkdFragment extends Fragment {
46     SessionManager sm;
47
48     @Nullable
49     @Override
50     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
51         //deklarasi session/
52         View v = inflater.inflate(R.layout.fragment_pkd, container, attachToRoot false);
53         sm = new SessionManager(getContext());
54         HashMap<String,String> data = sm.getUserDetail();
55         String nim = data.get(sm.NIM_SES);
56         String LVL = data.get(sm.LVL_SES);
57         Integer intLVL = Integer.parseInt(LVL);
58         //Integer nimINT = Integer.parseInt(nim);
59
60         WebView webView = (WebView)v.findViewById(R.id.webview);
61         webView.getSettings().setLoadsImagesAutomatically(true);
62         webView.getSettings().setJavaScriptEnabled(true);
63         webView.getSettings().setDomStorageEnabled(true);
64         webView.setOnKeyListener((v, keyCode, event) -> {
65             if(event.getAction() == KeyEvent.ACTION_DOWN)
66             {
67                 WebView webView = (WebView) v;
68
69                 switch(keyCode)
70                 {
71                     case KeyEvent.KEYCODE_BACK:
72                         if(webView.canGoBack())
73                         {
74                             webView.goBack();
75                             return true;
76                         }
77                 }
78             }
79         });
80     }

```

Gambar 4.49 Source Code Pendaftaran PKD

9. Tampilan Jadwal Kegiatan

Tampilan ini merupakan halaman yang menampilkan jadwal kegiatan yang meliputi tambah jadwal, edit jadwal dan hapus jadwal.

No	Nama Kegiatan	Mulai	Se
1	Gender Training	2020-03-06	2020-03-08
2	Pelatihan Jurnalistik	2020-01-25	2020-01-26
3	Camping	2020-03-28	2020-03-29
4	Khataman Qur'an	2020-01-23	2020-01-23
5	Makrab	2020-01-25	2020-01-26

Gambar 4.50 Tampilan Halaman Jadwal

Pada tampilan ini pengguna dapat melihat secara detail jadwal kegiatan mulai dari nama kegiatan, waktu pelaksanaan dan tempat kegiatan.

- **Tambah Jadwal** : admin dapat menambah jadwal yang akan dilaksanakan
- **Show Entries** : menampilkan jumlah daftar jadwal yang ingin ditampilkan
Contoh : 10 entries
- **Search** : admin dapat mencari jadwal kegiatan dengan mengisi nama kegiatan
- **Nomor** : merupakan id jadwal
- **Nama kegiatan** : menampilkan daftar nama kegiatan
- **Mulai** : menunjukkan waktu mulai kegiatan
- **Selesai** : menunjukkan waktu selesai kegiatan
- **Tempat** : menampilkan tempat pelaksanaan kegiatan
- **Keterangan** : menampilkan keterangan (selesai/belum terlaksana)

- Edit : admin dapat merubah jadwal kegiatan apabila ada perubahan waktu pelaksanaan

Berikut adalah potongan *source code* halaman jadwal kegiatan :

```

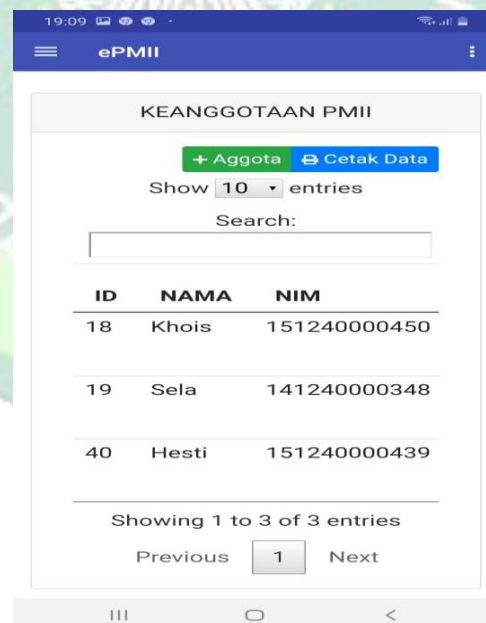
1 package com.wearedev.epmii.Admin;
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 public class Input_jadwalFragment extends Fragment {
23     @Nullable
24     @Override
25     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
26         final View view = inflater.inflate(R.layout.fragment_input_jadwal, container, attachToRoot: false);
27         WebView webView = ((WebView)view.findViewById(R.id.webview));
28         webView.getSettings().setLoadsImagesAutomatically(true);
29         webView.getSettings().setJavaScriptEnabled(true);
30         webView.getSettings().setDomStorageEnabled(true);
31         webView.setOnKeyListener((v, keyCode, event) -> {
32             if (event.getAction() == KeyEvent.ACTION_DOWN)
33             {
34                 WebView webView = (WebView) v;
35
36                 switch (keyCode)
37                 {
38                     case KeyEvent.KEYCODE_BACK:
39                         if (webView.canGoBack())
40                         {
41                             webView.goBack();
42                             return true;
43                         }
44                         break;
45                 }
46             }
47             return false;
48         });
49         // Tiga baris di bawah ini agar laman yang dimuat dapat
50         webView.setScrollBarStyle(WebView.SCROLLBARS_INSIDE_OVERLAY);
51         webView.setWebViewClient(new WebViewClient());
52     }
53 }

```

Gambar 4.51 *Source Code* Jadwal Kegiatan

10. Tampilan Keanggotaan

Pada tampilan ini merupakan halaman yang menampilkan daftar anggota yang telah terdaftar sebagai anggota.



Gambar 4.52 Tampilan Keanggotaan

Pada tampilan ini admin dapat melihat dengan detail daftar data anggota yang telah mendaftar juga dapat menambah anggota dan mencetak data anggota.

- *Show Entries* : menampilkan jumlah daftar anggota yang ingin ditampilkan

Contoh : 10 entries

- *Search* : pengguna dapat mencari jadwal yang akan dilaksanakan
- *Id* : merupakan id anggota
- *Nama* : menampilkan daftar nama anggota
- *NIM* : menampilkan nim anggota
- *Tgl Terdaftar* : menampilkan tanggal, bulan dan tahun anggota mendaftar.
- *Tempat Tanggal Lahir* : menampilkan tempat tanggal lahir anggota
- *Prodi* : menampilkan prodi asal anggota
- *Tambah Anggota* : admin dapat menambahkan data anggota yang telah terdaftar.
- *Cetak Data* : admin dapat mencetak data keanggotaan dalam bentuk pdf.

Berikut adalah potongan *source code* halaman keanggotaan :

```

23 import ...
24 public class Input_keanggotaanFragment extends Fragment {
25     @Nullable
26     @Override
27     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
28         final View view = inflater.inflate(R.layout.fragment_input_keanggotaan, container, #attachToRoot false);
29         WebView webView = (WebView) view.findViewById(R.id.webview);
30         webView.getSettings().setLoadsImagesAutomatically(true);
31         webView.getSettings().setJavaScriptEnabled(true);
32         webView.getSettings().setDomStorageEnabled(true);
33         webView.setOnKeyListener((v, keyCode, event) -> {
34             if(event.getAction() == KeyEvent.ACTION_DOWN)
35             {
36                 WebView webView = (WebView) v;
37
38                 switch(keyCode)
39                 {
40                     case KeyEvent.KEYCODE_BACK:
41                         if(webView.canGoBack())
42                         {
43                             webView.goBack();
44                             return true;
45                         }
46                         break;
47                     }
48             }
49         });
50         return false;
51     }
52 }
53
54 // Tiga baris di bawah ini agar laman yang dimuat dapat
55 webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
56 webView.setWebViewClient(new WebViewClient());
57 webView.setWebChromeClient(new MyWebChromeClient());
58 webView.loadUrl(stringConfig.KEANGGOTAAN);
59 webView.setDownloadListener((url, userAgent, contentDisposition, mimetype, contentLength) -> {

```

Gambar 4.53 *Source Code* Keanggotaan

11. Tampilan Upload Materi

Tampilan ini merupakan halaman yang menampilkan halaman upload materi.

Gambar 4.54 Tampilan Upload Materi

- *Text Field* Nama File : Admin harus mengisi nama file yang akan di uplod.
- *Choose File* : Admin harus memilih file materi yang akan di upload.

Berikut adalah potongan *source code* halaman upload materi :

```

2  import ...
26
27  public class Upload extends AppCompatActivity{
28      Uri pdf;
29      Button bt_upload, bt_pilih_file;
30      TextView status;
31      EditText et_nama_file;
32
33      @Override
34      protected void onCreate(Bundle savedInstanceState) {
35          super.onCreate(savedInstanceState);
36          setContentView(R.layout.activity_upload);
37
38
39
40          bt_pilih_file = (Button) findViewById(R.id.bt_pilih_file);
41          bt_upload = (Button) findViewById(R.id.bt_upload);
42          status = (TextView) findViewById(R.id.statusfile);
43          et_nama_file = (EditText) findViewById(R.id.et_nama_file);
44
45          /*show button back*/
46          getSupportActionBar().setTitle("Upload File Admin");
47          getSupportActionBar().setDisplayHomeAsUpEnabled(true);
48
49
50          //pilih file
51          bt_pilih_file.setOnClickListener(v) - {
52              Toast.makeText(getApplicationContext(), text: "Proses, pilih file", Toast.LENGTH_LONG).show();
53              Intent intent = new Intent();
54              intent.setType("application/pdf");
55              intent.setAction(Intent.ACTION_GET_CONTENT);
56              startActivityForResult(Intent.createChooser(intent, title: "Select PDF"), requestCode: 1);
57          }; //end click listener pilih file
58
59
60  }

```

Gambar 4.55 Source Code Upload Materi

4.2 Pengujian Metode

Pada tahap ini dilakukan uji coba terhadap sistem aplikasi yang telah dibuat. Tahap ini dilakukan untuk mengetahui apakah program aplikasi dapat menerima input dengan baik atau tidak, serta untuk mengetahui apakah *output* yang dihasilkan sudah sesuai atau belum. Untuk menguji kesesuaian sistem aplikasi, peneliti menggunakan metode *blackbox testing*.

Pengujian *blackbox* bertujuan untuk mengetahui apakah program telah menerima *input*, proses, dan *output* sesuai yang diinginkan.

Tabel 4.2 Pengujian *Black Box Testing*

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian
1	<i>Splash Screen</i>	✓	Menampilkan <i>Splash Screen</i>	Valid
2	Halaman Login	✓	Dapat masuk sistem dengan menginput <i>username</i> dan <i>password</i>	Valid
3	Halaman Register	✓	Dapat Membuat Akun dengan mengisikan NIM, Nama, Fakultas, No. <i>Handphone</i> dan password	Valid
4	Halaman Jadwal Kegiatan	✓	Dapat menampilkan data daftar jadwal kegiatan	Valid
5	Halaman Kepengurusan	✓	Dapat menampilkan struktur pengurus berdasarkan nama struktur organisasi	Valid
6	Halaman Materi	✓	Dapat menampilkan detail materi dan dapat mengunduh materi	Valid

7	Halaman Pendaftaran Keanggotaan	✓	Dapat menampilkan form pendaftaran dan melakukan pendaftaran sebagai anggota	Valid
8	Halaman Pendaftaran PKD	✓	Dapat menampilkan form pendaftaran serta dapat melakukan pendaftaran sebagai peserta pkd dan juga dapat cetak bukti pendaftaran	Valid
9	Halaman Pendaftaran MAPABA	✓	Dapat menampilkan form pendaftaran serta dapat melakukan pendaftaran sebagai peserta mapaba dan juga dapat cetak bukti pendaftaran	Valid
10	Halaman Input Jadwal	✓	Dapat menampilkan data jadwal kegiatan, menambah jadwal, edit jadwal dan hapus jadwal	valid
11	Halaman Input Kepengurusan	✓	Dapat menampilkan data kepengurusan dan menambah jabatan	Valid
12	Halaman Keanggotaan	✓	Dapat menampilkan data anggota, menambah anggota dan cetak anggota	Valid
13	Halaman Upload Materi	✓	Dapat mengupload sesuai materi yang dibutuhkan	Valid
14	Halaman Rekap PKD	✓	Dapat menampilkan data rekap peserta	Valid

			PKD	
15	Halaman Rekap MAPABA	✓	Dapat Menampilkan data rekap peserta MAPABA	Valid

4.3 Evaluasi dan Hasil Validasi

4.3.1 Evaluasi Sistem Aplikasi

Berdasarkan hasil pengujian yang telah dilakukan dengan metode *black box testing* yang menguji fitur-fitur halaman aplikasi ePMII didapat hasil pengujian bahwa aplikasi dapat berjalan sesuai keinginan. Jadi disimpulkan bahwa aplikasi E PMII Jepara sudah valid dan berjalan sesuai harapan.

4.3.2 Hasil Validasi Kelayakan Aplikasi

4.3.2.1 Validasi Ahli

1) Ahli Materi

Validasi yang dilakukan oleh ahli materi adalah mengumpulkan saran dan pendapat untuk selanjutnya dilakukan revisi terhadap aplikasi ePMII

Tabel 4.3 Hasil Penilaian Ahli Materi

No.	Indikator	Penilaian		
		V	VR	TV
1.	Materi mendukung pencapaian tujuan	1		
2.	Materi mudah dimengerti	1		
3.	Penyajian materi sesuai dengan tujuan yang dirumuskan	1		
4.	Kejelasan penyampaian materi	1		
5.	Kelengkapan materi		1	
6.	Relevansi tujuan sistem informasi	1		
7.	Kesusaian penggunaan Bahasa yang digunakan	1		
Jumlah		6	1	

Dari tabel diatas ada 7 poin yang mendapatkan respon V (Valid), dan 1 poin yang mendapatkan respon VR (Valid Revisi)

Perhitungan data tabel :

$$P = \frac{(6 \times 3) + (1 \times 2)}{21} \times 100\% = \frac{20}{21} \times 100\% = 95,2\%$$

2) Ahli Media

Validasi yang dilakukan oleh ahli materi adalah mengumpulkan saran dan pendapat untuk selanjutnya dilakukan revisi terhadap aplikasi ePMII.

Tabel 4.4 Hasil Penilaian Ahli Media

No.	Indikator	Penilaian		
		V	VR	TV
1.	Kemudahan alur materi melalui penggunaan bahasa	1		
2.	Kesesuaian menu dengan materi	1		
3.	Kejelasan uraian materi	1		
4.	Kesesuaian pemilihan jenis huruf dan ukuran huruf	1		
5.	Teks dapat terbaca dengan baik	1		
6.	Kesesuaian tata letak teks	1		
7.	Kesesuaian warna	1		
8.	Kecepatan pemrosesan dalam melakukan perintah	1		
9.	Ketepatan fungsi tombol navigasi	1		
Jumlah		9		

Dari tabel diatas ada 9 poin yang mendapatkan respon V (Valid)

Perhitungan data tabel :

$$P = \frac{(9 \times 3)}{27} \times 100\% = \frac{27}{27} \times 100\% = 100\%$$

Tabel 4.5 Penilaian Kelayakan

No.	Presentase	Kriteria
1	75 % - 100 %	Sangat layak
2	50 % - 75 %	Layak
3	25 % - 50 %	Cukup layak
4	1 % - 25 %	Kurang layak

Tabel 4.6 Validasi Ahli

No	Ahli	Instrumen	Skor Ideal	Skor Ahli	Persentase (%)	Kriteria	Keterangan Ahli
1	Materi	7	21	20	95,2 %	Sangat Layak	Kelengkapan materi masih kurang, dapat dikembangkan lagi dengan materi yang lebih lengkap
2	Media	9	27	27	100 %	Sangat Layak	Aplikasi perlu ditambah sub <i>menuchatting</i> , kelengkapan materi dan penambahan gambar

4.3.2.2 Hasil Angket Responden

Penilaian aplikasi ePMII juga dilakukan dengan menyebar angket responden yang berisi 10 pertanyaan dengan jumlah responden 40 anggota PMII Jepara, berikut rekapitulasi hasil nilai sesuai butir pertanyaan.

Tabel 4.7 Hasil Penilaian Responden

No.	Pertanyaan	Nilai
1.	Apakah anda setuju apabila pengelolaan data Integrasi Keanggotaan Pergerakan Mahasiswa Islam Indonesia (PMII) Jepara dijadikan suatu sistem informasi ?	184

2.	Aplikasi e-PMII Jepara ini akan mempermudah dalam pendataan anggota	179
3.	Bahasa dan kalimat yang digunakan dalam Aplikasi e-PMII Jepara mudah dipahami	158
4.	Penggunaan <i>font</i> , ukuran huruf, warna dan gambar dalam Aplikasi e-PMII Jepara terlihat jelas	156
5.	Tampilan menu pada Aplikasi e-PMII Jepara dan tata letak mudah dipahami	164
6.	Apakah aplikasi e-PMII jepara ini mempermudah untuk mengetahui data anggota?	167
7.	Aplikasi e-PMII Jepara ini dapat digunakan dimana saja dan kapan saja	170
8.	Menu yang ditampilkan sesuai dengan output yang diinginkan	152
9.	Apakah aplikasi ini membantu dalam pengelolaan data anggota?	165
10.	Aplikasi ini menyajikan informasi sesuai dengan kebutuhan anggota	155
Skor Total		1650

Tabel 4.8 Jawaban Angket Responden

Nomor Responden	Jawaban Responden untuk Item Nomor										Jumlah
	1	2	3	4	5	6	7	8	9	10	
1	4	4	4	3	4	5	4	3	3	4	38
2	5	5	5	5	5	5	5	5	5	5	50
3	4	3	3	3	3	3	4	3	3	3	32
4	4	4	4	4	4	4	4	4	4	4	40
5	4	5	5	5	4	5	5	4	5	4	46
6	3	4	3	3	3	3	5	3	3	3	33
7	5	4	5	5	5	4	5	5	4	3	45
8	5	5	3	3	3	5	5	3	5	3	40

9	5	5	3	3	3	5	5	4	4	3	40
10	5	5	5	5	5	5	5	4	5	4	48
11	5	5	5	5	5	5	5	4	5	4	48
12	4	4	3	3	3	4	3	3	3	3	33
13	3	3	3	3	3	3	3	3	3	3	30
14	4	4	3	4	4	5	4	4	5	4	41
15	5	5	4	4	4	4	4	3	4	4	41
16	4	4	4	3	3	4	3	4	5	3	37
17	5	5	3	4	4	5	5	4	5	5	45
18	4	5	4	4	3	4	3	3	4	3	37
19	5	5	5	5	5	5	5	5	5	5	50
20	5	4	4	4	3	3	4	4	3	3	37
21	5	4	4	3	4	3	4	4	4	4	39
22	4	4	4	4	4	4	4	4	4	4	40
23	4	4	3	3	4	3	4	4	5	4	38
24	5	4	4	4	5	4	4	3	4	3	40
25	5	5	4	4	5	3	4	4	4	4	42
26	5	4	5	4	3	3	4	4	4	3	39
27	5	5	4	4	4	4	4	4	4	4	42
28	4	4	4	4	3	3	4	4	3	4	37
29	4	4	3	3	4	4	4	4	3	3	36
30	5	5	4	4	5	4	3	3	4	4	41
31	4	4	3	3	4	4	3	3	3	4	35
32	5	4	4	4	4	4	4	4	4	4	41
33	5	4	4	4	4	4	4	4	4	4	41
33	4	4	3	3	4	3	4	3	4	4	36
34	4	4	4	4	4	4	4	4	4	4	40
35	5	5	3	3	5	4	4	3	4	4	40
36	5	5	4	4	5	5	5	4	4	4	45
37	4	4	4	4	4	4	4	3	4	4	39
38	5	5	4	4	4	5	5	4	4	5	45
39	4	4	4	4	4	4	4	3	4	4	39
40	5	5	4	4	5	5	4	4	4	4	44
Jumlah	184	179	158	156	164	167	170	152	165	155	1650

Tabel 4.9 Kelayakan Penilaian

No.	Presentase	Kriteria
1	75 % - 100 %	Sangat layak
2	50 % - 75 %	Layak
3	25 % - 50 %	Cukup layak
4	1 % - 25 %	Kurang layak

Tabel 4.10 Validasi Responden

No	Jumlah Responden	Instrumen	Skor Ideal	Skor Total	Persentase (%)	Kriteria	Keterangan
1	40	10	2000	1650	82,5 %	Sangat Layak	-

Berdasarkan penilaian oleh responden secara keseluruhan, Aplikasi E-PMII Jepara mendapat skor 1650 pada 10 instrumen atau pertanyaan disetiap angket yang kemudian dilakukan dengan menyebar 40 angket dan menghasilkan presentase sebanyak 82,5% dengan kriteria sangat layak.