

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1 Implementasi**

##### **4.1.1 Kebutuhan Software dan Hardware**

Spesifikasi *software* yang digunakan untuk membuat aplikasi UPracte adalah sebagai berikut :

1. Sistem Operasi : Windows 10 64 Bit
2. Bahasa Pemrograman : Java dan XML
3. Prototype : Adobe XD
4. *Software* Pengembangan : Android Studio 3.3.2
5. Java SDK : Android Development Tools V.  
Android SDK Platform-tools Rev.
6. Database : Firebase

Adapun spesifikasi *hardware* yang digunakan adalah sebagai berikut:

1. Laptop : Lenovo Z40-70
2. Processor : Core I5 Gen 3
3. Ram : 8 GB

##### **4.1.2 Implementasi Rancangan Sistem Sisi Perangkat Bergerak**

Implementasi rancangan sistem terbagi dari dua sisi, yaitu dari sisi perangkat bergerak (android) dan perangkat *database*. Rancangan sistem android memiliki 1 mode user yaitu mahasiswa. Aplikasi memiliki beberapa *class* dengan beberapa fungsinya, yaitu halaman depan (SplashScreen), halaman login, halaman registrasi, halaman introduce, halaman utama, halaman nilai praktikum, halaman informasi praktikum hingga halaman pemilihan jadwal praktikum.

Sebelum masuk ke halaman utama, akan ada beberapa *class* yang dijalankan oleh *device* android. Tampilan pertama kali saat aplikasi dibuka adalah tampilan splashscreen yang akan dijalankan adalah *class* SplashScreen.java dengan splashscreen.xml sebagai tampilannya. Tampilan ini juga ditunjukkan pada gambar 4.1. Tampilan SplashScreen.



Gambar 4.1 Tampilan SplashScreen

Adapun tampilan cuplikan kode dari fitur splashscreen ini ada pada gambar dibawah ini

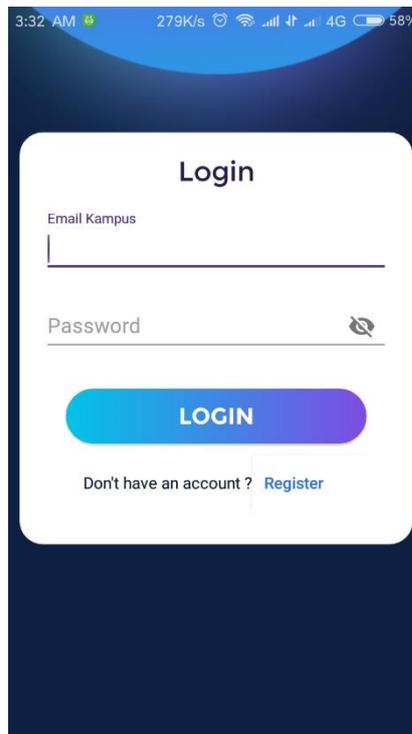
```
12 public class SplashScreen extends AppCompatActivity {
13     TextView tvSplash;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18
19         //menghilangkan ActionBar
20         requestWindowFeature(Window.FEATURE_NO_TITLE);
21         getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
22             WindowManager.LayoutParams.FLAG_FULLSCREEN);
23         setContentView(R.layout.splashscreen);
24
25         tvSplash = (TextView) findViewById(R.id.tvSplash);
26
27
28         final Handler handler = new Handler();
29         handler.postDelayed(new Runnable() {
30             @Override
31             public void run() {
32                 startActivity(new Intent(getApplicationContext(), Login.class));
33                 finish();
34             }
35         }, 3000L); //3000 L = 3 detik
36     }
37 }
```

Gambar 4.2 Cuplikan Code SplashScreen

Pada SplashScreen.java *user* akan menunggu selama 3 detik dengan progressbar sebagai penunjuk proses. Ketika proses telah selesai maka akan langsung memasuki halaman login.

#### 4.1.2.1 Halaman Login

Halaman login merupakan halaman yang memiliki peran dimana jika *user* telah memiliki *account* maka *user* akanizinkan masuk ke halaman selanjutnya. Halaman login memiliki tampilan seperti gambar 4.3



Gambar 4.3 Tampilan Login

Pada halaman login, *user* diharuskan mengisi data sesuai dengan yang telah dibuat pada saat registrasi, pada halaman ini ada 2 tipe *input* yang wajib dimasukkan oleh *user* yaitu *input* dari email meliputi data email kampus *user* dan juga input dari password meliputi data password *user* yang telah dibuat, Jika data email dan password yang dimasukkan *user* salah maka aplikasi akan menampilkan kegagalan login. Sehingga *user* akan mengulang memasukkan data yang sesuai dengan yang ada pada database. Pada penggunaannya halaman login akan memproses sesuai pada gambar 4.4 Cuplikan code login yang ada dibawah ini.

```

89
90 //authenticate user
91 auth.signInWithEmailAndPassword(email, password)
92 .addOnCompleteListener(Login.this, new OnCompleteListener<AuthResult>() {
93     @Override
94     public void onComplete(@NonNull Task<AuthResult> task) {
95 // If sign in fails, display a message to the user. If sign in succeeds
96 // the auth state listener will be notified and logic to handle the
97 // signed in user can be handled in the listener.
98     progressBar.setVisibility(View.GONE);
99     if (!task.isSuccessful()) {
100 // there was an error
101         if (password.length() < 6) {
102             inputPassword.setError(getString(R.string.minimum_password));
103         } else {
104             Toast.makeText(Login.this, getString(R.string.auth_failed),
105                 Toast.LENGTH_LONG).show();
106         }
107     } else {
108         Intent intent = new Intent(Login.this, Introduction.class);
109         startActivity(intent);
110         finish();
111     }
112     }
113     });
114 }
115 });

```

Gambar 4.4 Cuplikan Kode Login

Data login akan didapatkan melalui database firebase yang memiliki UID pengguna. Firebase merupakan database yang digunakan untuk mengambil akun user yang telah disimpan pada saat registrasi

#### 4.1.2.1.1 Database User

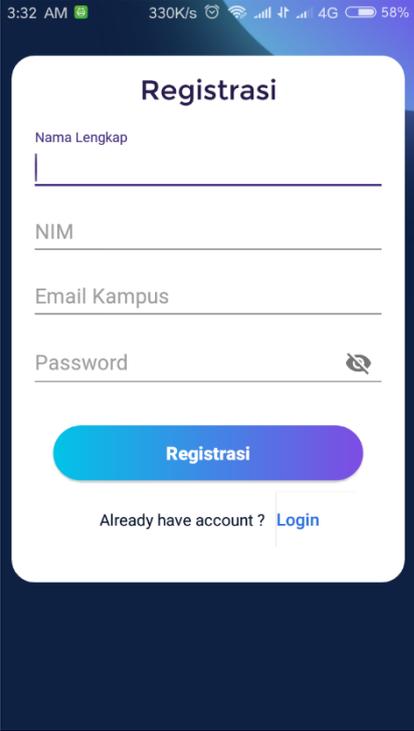
Database user masuk kedalam firebase authentication dengan pengenal user adalah email dan UID pengguna yang berbeda-beda pada tiap user

Pengenal	Penyedia	Dibuat tanggal ↑	Masuk	UID pengguna
muhammad.hafiz.2015@ft.u...	✉	11 Jun 2019	21 Jun 2019	FMLhS9p0xVRTs0FO9umlvTYJhki2
suko.ferbriyanto.2015@ft.u...	✉	16 Jun 2019	16 Jun 2019	k3WztiUlshUocbsNtzNS3MCS1cG3
dwiki.rachmanto.2015@ft.u...	✉	16 Jun 2019	16 Jun 2019	t5DoQb4kFGS7b8UdEsF1elpAVa93
muhammad.abduh.2015@ft...	✉	16 Jun 2019	16 Jun 2019	x6S93uHL0wXxiAezfFX20509793
riza.septian.2015@ft.umy.ac...	✉	17 Jun 2019	17 Jun 2019	crCnamDXeBhICzvAaEFMGIJg4AN2
nurohman.fadilah.2015@ft.u...	✉	17 Jun 2019	17 Jun 2019	SRmxp2MJuCaUtmGra64Tquw75...
muhammad.arif.2015@ft.um...	✉	17 Jun 2019	17 Jun 2019	q8Rdadh1qad1Amo57hjjugaTTAC2
fatkhurrohman.2015@ft.umy...	✉	17 Jun 2019	17 Jun 2019	D33kwHUc2Wb8SOql3nTzRZHBOf...

Gambar 4.5 Database User

#### 4.1.2.2 Halaman Registrasi

Halaman registrasi merupakan halaman yang memiliki peran apabila *user* belum memiliki *account* maka *user* mengisi data, jika data telah benar maka *user* telah membuat *account* baru sehingga bisa memasuki halaman utama. Halaman registrasi memiliki tampilan seperti gambar 4.6



Gambar 4.6 Tampilan Registrasi

Pada halaman registrasi, *user* akan mengisi 4 tipe *input* yang wajib dimasukkan oleh *user* jika ingin membuat sebuah *account* yaitu *input* dari nama lengkap, *input* dari NIM, *input* dari email kampus dan *input* dari password. Pada saat melakukan *input* data untuk registrasi ada beberapa persyaratan yang harus dilakukan *user* karena pada tiap *input* memiliki persyaratan masing – masing pada *input* NIM *user* diharuskan mengisi NIM dengan panjang lebih dari 11 karakter. Jika kurang dari itu maka sistem akan menampilkan *warning* “Masukkan NIM Lengkap, 11 Karakter.” Sementara pada *input* email kampus *user* diharuskan mengisi email kampus dengan panjang lebih dari 24 karakter, jika kurang dari itu maka sistem akan menampilkan *warning* “Masukkan Email Kampus, Contoh

([nama.tahunmasuk@ft.umy.ac.id](mailto:nama.tahunmasuk@ft.umy.ac.id))." gambar dibawah ini akan menunjukkan cuplikan code dari tampilan halaman registrasi.

```
106
107     if (TextUtils.isEmpty(user_nim)) {
108         Toast.makeText(getApplicationContext(), "Masukkan NIM",
109             Toast.LENGTH_SHORT).show();
110         return;
111     }
112
113     if (user_nim.length() < 11) {
114         Toast.makeText(getApplicationContext(), "Masukkan NIM , 11 Karakter" ,
115             Toast.LENGTH_SHORT).show();
116         return;
117     }
118
119     if (TextUtils.isEmpty(user_email)) {
120         Toast.makeText(getApplicationContext(), "Masukkan Email Kampus",
121             Toast.LENGTH_SHORT).show();
122         return;
123     }
124
125     if (user_email.length() < 24) {
126         Toast.makeText(getApplicationContext(), "Masukkan Email Kampus, Contoh (nama.tahunmasuk@ft.umy.ac.id)",
127             Toast.LENGTH_SHORT).show();
128         return;
129     }
130
131
132     if (TextUtils.isEmpty(user_password)) {
133         Toast.makeText(getApplicationContext(), "Masukkan password",
134             Toast.LENGTH_SHORT).show();
135         return;
136     }
137
138     if (user_password.length() < 4) {
139         Toast.makeText(getApplicationContext(), "Password terlalu pendek, minimal 4 karakter",
140             Toast.LENGTH_SHORT).show();
141         return;
142     }
143
```

Gambar 4.7 Cuplikan Kode Registrasi

Jika semua data telah dimasukkan dan *user* menekan tombol registrasi maka aplikasi akan menampilkan login berhasil. Sehingga *user* akan langsung memasuki halaman utama. Data registrasi yang telah dibuat akan masuk ke database firebase authentication dimana tiap *user* yang telah membuat *account* akan memiliki UID pengguna sendiri, dan firebase realtime database akan menyimpan data dari *input* nama lengkap, NIM dan email kampus yang telah dimasukkan *user*.

#### 4.1.2.2.1 Database Informasi Praktikum

Data yang telah diisi oleh user pada saat registrasi maka akan langsung masuk ke firebase realtime database untuk menyimpan data.



Gambar 4.8 Database Registrasi

Adapun cuplikan code input *user* yang telah berhasil melakukan registrasi terdapat pada gambar dibawah ini.

```

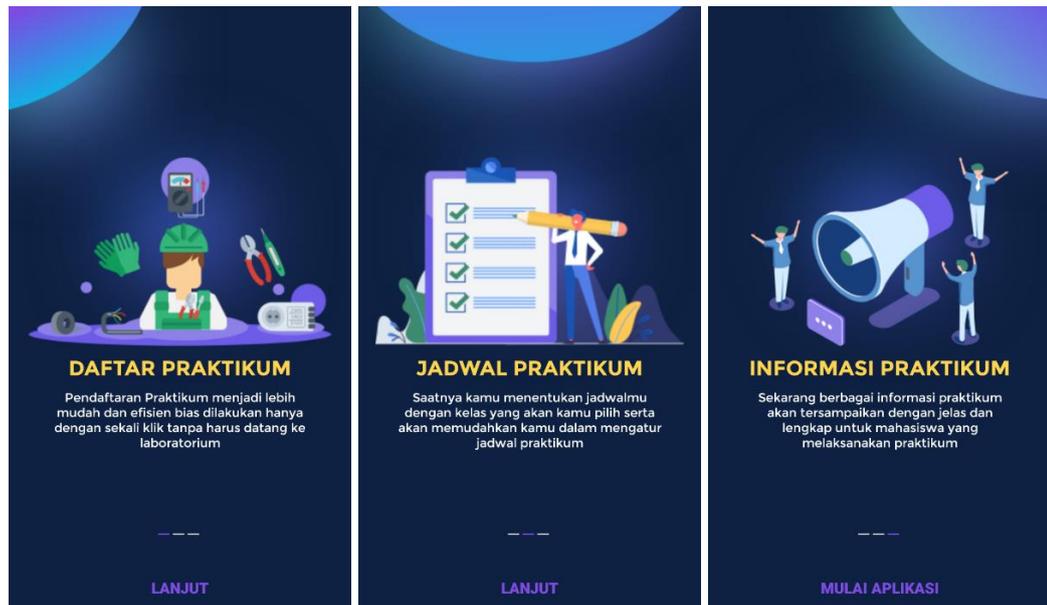
147     auth.createUserWithEmailAndPassword(user_email, user_password)
148     .addOnCompleteListener(Register.this, new OnCompleteListener<AuthResult>() {
149         @Override
150         public void onComplete(@NonNull Task<AuthResult> task) {
151             Toast.makeText(Register.this, "Proses" + task.isSuccessful(), Toast.LENGTH_SHORT).show();
152             progressBar.setVisibility(View.GONE);
153         }
154     });
155
156     if (!task.isSuccessful()) {
157         Toast.makeText(Register.this, "Pendaftaran Gagal." + task.getException(),
158             Toast.LENGTH_SHORT).show();
159     } else {
160
161         FirebaseUser user = auth.getCurrentUser();
162         String userID = user.getUid();
163         DatabaseReference uid = FirebaseDatabase.getInstance().getReference().child("Users");
164         DatabaseReference uid_ref = uid.child(userID);
165         uid_ref.child("Nama").setValue(user_name);
166         uid_ref.child("Nim").setValue(user_nim);
167         uid_ref.child("Email").setValue(user_email);
168         Toast.makeText(Register.this, "Pendaftaran Berhasil." + task.getException(),
169             Toast.LENGTH_SHORT).show();
170
171         startActivity(new Intent(Register.this, Introduction.class));
172         finish();
173     }
174 }
175 }
176 }

```

Gambar 4.9 Cuplikan Kode Input Registrasi

### 4.1.2.3 Halaman Intro

Halaman intro merupakan halaman yang akan pertama kali ditampilkan ketika *user* belum pernah melakukan login kedalam sistem. Pada halaman intro *user* akan dikenalkan dengan berbagai kegunaan dari aplikasi. Halaman intro memiliki tampilan seperti gambar 4.10



(a)

(b)

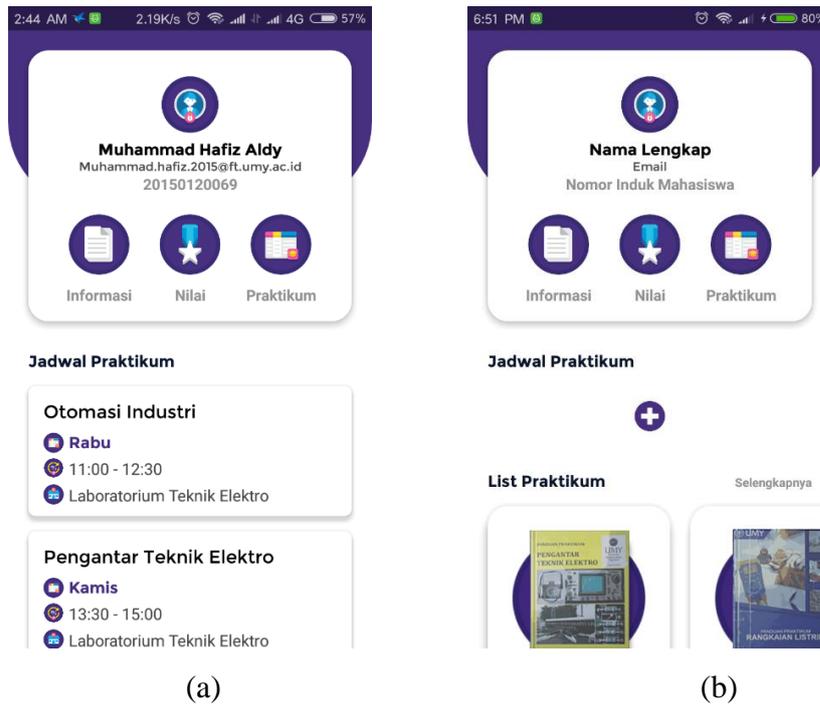
(c)

Gambar 4.10 (a) halaman introduce 1, (b) halaman introduce 2. (c) halaman introduce 3

#### 4.1.2.4 Halaman Utama

Halaman utama merupakan halaman yang akan pertama kali ditampilkan ketika *user* melakukan login kedalam sistem. Pada halaman utama terdapat 5 menu utama, yaitu informasi praktikum, nilai praktikum, list praktikum, sign out dan jadwal praktikum yang telah diambil. Selain itu, terdapat juga nama, NIM, dan email *user* yang sama ketika *user* melakukan input registrasi. Jadwal praktikum didapatkan setelah *user* melakukan pendaftaran praktikum yang didapatkan pada detail praktikum.

Pada awalnya jadwal praktikum akan menampilkan *recyclerview* yang kosong jika *user* belum mendaftar praktikum. Tampilan pada halaman utama akan menampilkan nama lengkap, NIM, email dan juga jadwal praktikum jika *user* memiliki koneksi internet. Apabila jika *user* tidak memiliki koneksi internet maka halaman utama akan menampilkan tampilan kosong tanpa *data* dari *user*. Tampilan ditujukan pada gambar 4.11



Gambar 4.11 (a) Halaman utama terkoneksi internet, (b) Halaman utama tanpa koneksi internet

Adapun tampilan dari halaman utama didapatkan dari cuplikan kode yang ada pada gambar dibawah ini.

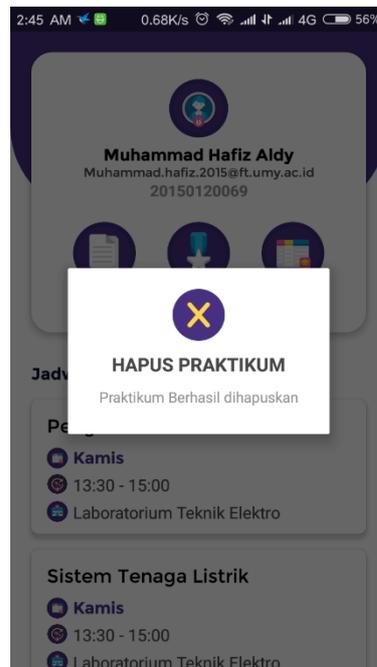
```

79
80
81     final TextView usernamal = (TextView) findViewById(R.id.user_id);
82     final TextView userniml = (TextView) findViewById(R.id.user_nomor);
83     final TextView useremail = (TextView) findViewById(R.id.user_email);
84
85     final FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
86     String userID = user.getId();
87     reference.child(userID).addValueEventListener(new ValueEventListener() {
88         @Override
89         public void onDataChange(DataSnapshot dataSnapshot) {
90             String username = dataSnapshot.child("Nama").getValue(String.class);
91             String usernim = dataSnapshot.child("Nim").getValue(String.class);
92             String useremail = dataSnapshot.child("Email").getValue(String.class);
93             String datanama = username;
94             String datanim = usernim;
95             String dataemail = useremail;
96
97             usernamal.setText(datanama);
98             userniml.setText(datanim);
99             useremail.setText(dataemail);
100         }
101
102
103         @Override
104         public void onCancelled(DatabaseError databaseError) {
105         }
106     });

```

Gambar 4.12 Cuplikan Kode Halaman Utama

Pada halaman ini juga terdapat fitur hapus praktikum, *user* bisa melakukan hapus praktikum dengan cara melakukan *slide* kearah kiri ataupun kanan pada jadwal praktikum yang telah diambil setelah itu akan ada pemberitahuan mengenai praktikum telah dihapuskan. Tampilan ditunjukkan pada gambar 4.13.



Gambar 4.13 Hapus Praktikum

Adapun tampilan cuplikan kode dari fitur hapus praktikum ini ada pada gambar dibawah ini

```

124
125 private void setUpRecyclerViewJadwal() {
126     String userID = user.getId();
127     jadwalReference = FirebaseDatabase.getInstance().getReference().child("Users")
128     .child(userID).child("Praktikum");
129     FirebaseRecyclerOptions <Jadwal> options = new FirebaseRecyclerOptions.Builder <Jadwal> ()
130     .setQuery(jadwalReference, Jadwal.class)
131     .build();
132
133     adapterjadwal = new JadwalAdapter(options);
134     RecyclerView recyclerViewJadwal = findViewById(R.id.recyclerview_jadwal);
135     recyclerViewJadwal.setLayoutManager(new LinearLayoutManager(this));
136     recyclerViewJadwal.setAdapter(adapterjadwal);
137
138     new ItemTouchHelper(new ItemTouchHelper.SimpleCallback(0,
139     ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT)) {
140     @Override
141     public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder,
142     @NonNull RecyclerView.ViewHolder viewHolder1) {
143         return false;
144     }
145
146     @Override
147     public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int i) {
148
149         dialog = new Dialog(MainActivity.this);
150         dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
151         dialog setContentView(R.layout.custom_box_berhasil);
152         dialog.show();
153         Window window = dialog.getWindow();
154         window.setLayout(LinearLayout.LayoutParams.WRAP_CONTENT, LinearLayout.LayoutParams.WRAP_CONTENT);
155         adapterjadwal.deleteitem(viewHolder.getAdapterPosition());
156
157     }
158     }.attachToRecyclerView(recyclerViewJadwal);
159
160 }

```

Gambar 4.14 Cuplikan Kode Hapus Praktikum

#### 4.1.2.5 Halaman Informasi Praktikum

Halaman informasi praktikum merupakan halaman yang memiliki fungsi untuk menyampaikan informasi dari pihak laboran langsung kepada mahasiswa. Dalam halaman informasi mahasiswa dapat melihat judul, deskripsi, tanggal pembuatan dan juga siapa pembuat dari informasi tersebut. Sehingga dengan begitu mahasiswa bisa langsung mendapatkan informasi yang akurat. Adapun tampilan dari informasi praktikum ada pada gambar dibawah ini



Gambar 4.15 Halaman Informasi Praktikum

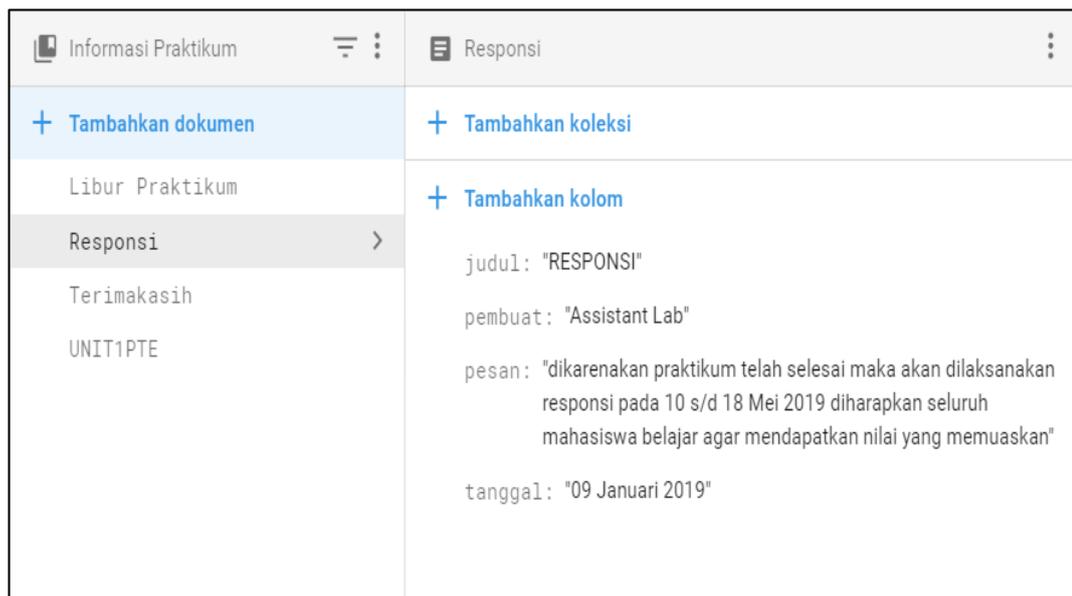
##### 4.1.2.5.1 Database Informasi Praktikum

Database informasi praktikum masuk kedalam firebase cloud firestore sehingga mahasiswa tetap bisa melihat informasi praktikum walaupun tidak memiliki koneksi internet. Adapun variable yang terdapat pada database informasi praktikum adalah sebagai berikut.

Tabel 4.1 Database Informasi Praktikum

No	Nama	Variable	Tipe Data
1	Judul	judul	String
2	Deskripsi	pesan	String
3	Pembuat	pembuat	String
4	Tanggal	tanggal	String

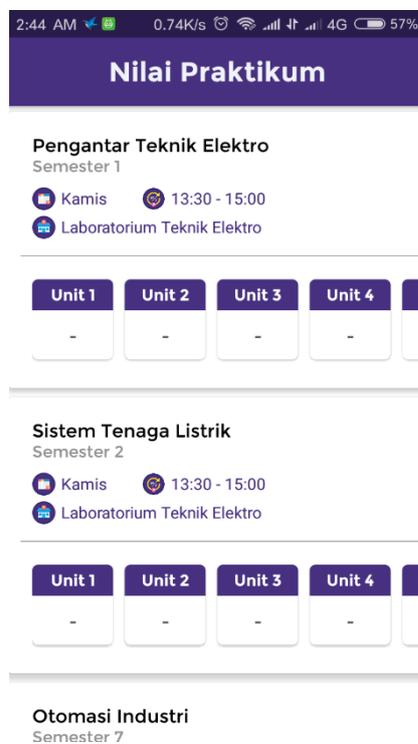
Adapun pada database firebase cloud firestore *user* hanya akan melihat data informasi praktikum yang berada pada koleksi Informasi Praktikum yang berisi dokumen – dokumen yang telah dibuat untuk memberikan informasi kepada *user*. Pada database informasi praktikum ini *user* dapat melihat informasi secara *offline* dan untuk memperbaharui data maka *user* harus mengaktifkan koneksi internet. Adapun tampilan dokumen database ditujukan pada gambar dibawah ini.



Gambar 4.16 Database Informasi Praktikum

#### 4.1.2.6 Nilai Praktikum

Halaman nilai praktikum merupakan halaman yang memiliki fungsi untuk menyampaikan nilai – nilai dari *user* yang diisi oleh dari pihak laboran langsung. Dalam halaman nilai praktikum mahasiswa dapat melihat judul, semester, hari, jam , tempat dan nilai praktikum. Sehingga dengan begitu mahasiswa bisa langsung mendapatkan informasi mengenai nilai praktikum. Adapun tampilan dari informasi praktikum ada pada gambar dibawah ini



Gambar 4.17 Halaman Nilai Praktikum

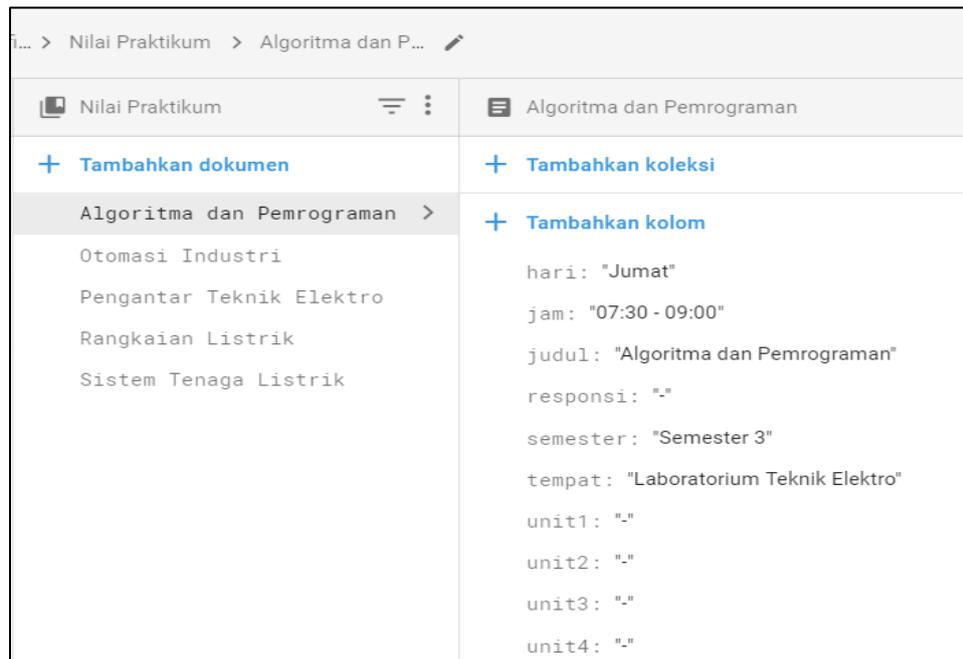
##### 4.1.2.6.1 Database Nilai Praktikum

Database nilai praktikum masuk kedalam firebase cloud firestore sehingga mahasiswa tetap bisa melihat informasi praktikum walaupun tidak memiliki koneksi internet. Adapun variable yang terdapat pada database informasi praktikum adalah sebagai berikut.

Tabel 4.2 Database Nilai Praktikum

No	Nama	Variable	Tipe Data
1	Judul Praktikum	judul	String
2	Deskripsi	hari	String
3	Pembuat	jam	String
4	Tanggal	tempat	String
5	Nilai Unit 1	unit1	String
6	Nilai Unit 2	unit2	String
7	Nilai Unit 3	unit3	String
8	Nilai Unit 4	unit4	String
9	Nilai Unit 5	unit5	String
10	Nilai Unit 6	unit6	String
11	Nilai Unit 7	unit7	String
12	Nilai Responsi	responsi	String

Adapun pada database firebase cloud firestore *user* hanya akan melihat data nilai praktikum yang berada pada koleksi Data Praktikum dengan rincian tempat /Data [Praktikum/EmailUser/Nilai](#) Praktikum>Nama Praktikum yang berisi dokumen berupa email *user* dengan didalamnya berisi koleksi Nilai Praktikum yang memiliki dokumen berupa praktikum – praktikum yang telah diambil oleh *user*. Adapun tampilan dokumen nilai praktikum database ditunjukan pada gambar dibawah ini .



Gambar 4.18 Database Nilai Praktikum

Adapun cuplikan kode yang digunakan untuk memanggil database yang berada pada firebase cloud firestore *user* ditujukan pada gambar dibawah ini .

```

43
44     class NilaiHolder extends RecyclerView.ViewHolder{
45         TextView textViewHari;
46         TextView textViewJam;
47         TextView textViewJudul;
48         TextView textViewTempat;
49         TextView textViewSemester;
50         TextView textViewResponsi;
51         TextView textViewUnit1;
52         TextView textViewUnit2;
53         TextView textViewUnit3;
54         TextView textViewUnit4;
55         TextView textViewUnit5;
56         TextView textViewUnit6;
57         TextView textViewUnit7;
58
59     public NilaiHolder(@NonNull View itemView) {
60         super(itemView);
61         textViewSemester = itemView.findViewById(R.id.text_view_semester_nilai);
62         textViewHari = itemView.findViewById(R.id.text_view_day);
63         textViewJam = itemView.findViewById(R.id.text_view_time);
64         textViewJudul = itemView.findViewById(R.id.text_view_title);
65         textViewTempat = itemView.findViewById(R.id.text_view_room);
66         textViewResponsi = itemView.findViewById(R.id.text_responsi);
67         textViewUnit1 = itemView.findViewById(R.id.text_unit1);
68         textViewUnit2 = itemView.findViewById(R.id.text_unit2);
69         textViewUnit3 = itemView.findViewById(R.id.text_unit3);
70         textViewUnit4 = itemView.findViewById(R.id.text_unit4);
71         textViewUnit5 = itemView.findViewById(R.id.text_unit5);
72         textViewUnit6 = itemView.findViewById(R.id.text_unit6);
73         textViewUnit7 = itemView.findViewById(R.id.text_unit7);
74
75     }
76

```

Gambar 4.19 Cuplikan Kode Nilai Praktikum

#### 4.1.2.7 List Praktikum

Halaman list praktikum merupakan halaman yang memiliki fungsi untuk melihat list dari banyaknya praktikum yang tersedia. Dalam halaman list praktikum mahasiswa dapat melihat gambar, judul dan semester praktikum. Sehingga dengan begitu mahasiswa bisa langsung memilih praktikum sesuai yang diinginkan. Adapun tampilan dari nilai praktikum ada pada gambar dibawah ini

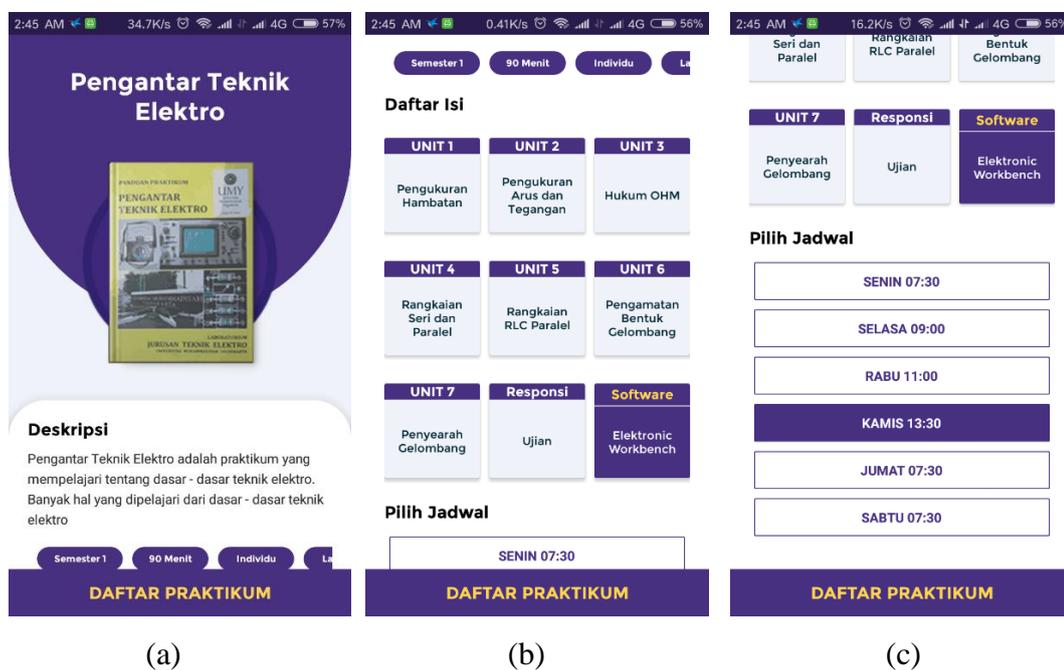


Gambar 4.20 Halaman List Praktikum

Pada halaman list praktikum ini, terdapat 16 praktikum yang bisa dipilih oleh *user* dengan rincian 1 praktikum semester 1, 2 praktikum semester 2, 3 praktikum semester 3, 3 praktikum semester 4, 3 praktikum semester 5, 3 praktikum semester 6 dan 1 praktikum semester 7. Sehingga list praktikum akan memudahkan *user* dalam memilih praktikum yang diinginkan sesuai dengan semester *user*.

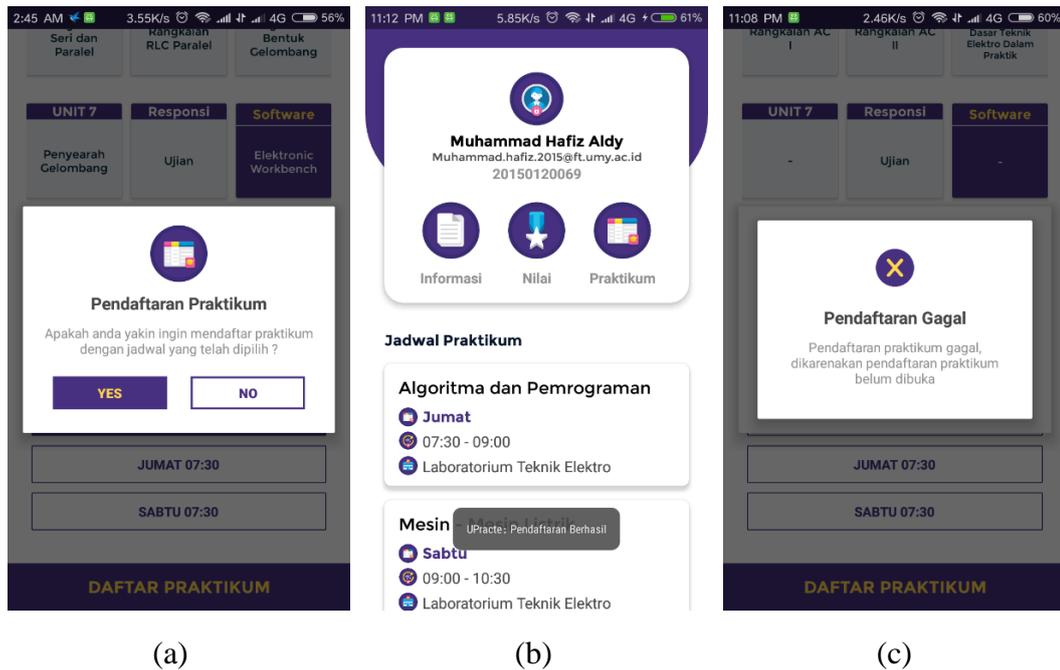
#### 4.1.2.8 Detail Praktikum

Halaman detail praktikum merupakan halaman yang memiliki fungsi untuk menyampaikan informasi mengenai praktikum yang akan diambil oleh *user*. Dimana informasi tersebut meliputi judul praktikum, gambar praktikum, deskripsi praktikum, daftar isi praktikum dari unit 1 hingga 7 dan juga informasi mengenai pendaftaran praktikum dengan pemilihan jadwal praktikum. Adapun tampilan dari informasi praktikum ada pada gambar dibawah ini.



Gambar 4.21 (a) Detail praktikum, (b) Daftar isi praktikum, (c) Jadwal praktikum

Setelah melihat penjelasan praktikum *user* bisa memilih jadwal praktikum sesuai yang diinginkan pada hari dan jam yang ditentukan. *User* hanya bisa memilih satu jadwal praktikum, dengan cara memilih jadwal yang diinginkan dan menekan tombol daftar praktikum. Tombol daftar praktikum akan memasukkan data ke firebase realtime database untuk jadwal praktikum pada halaman utama dan juga firebase cloud firestore pada halaman nilai praktikum secara bersamaan. Pendaftaran akan gagal jika *rules* pada firebase realtime database untuk *write* dalam nilai *false*.



Gambar 4.22 (a) Notifikasi daftar praktikum, (b) Daftar praktikum berhasil, (c) Daftar praktikum gagal

#### 4.1.2.8.1 Database Pendaftaran Praktikum

Database pendaftaran praktikum akan masuk kedalam 2 database, (1) firebase realtime database untuk jadwal praktikum dan (2) firebase cloud firestore untuk nilai praktikum. Pada firebase realtime database jadwal praktikum akan langsung masuk jika terkoneksi dengan internet, sementara untuk firebase cloud firestore nilai praktikum akan tetap ada walaupun tidak memiliki koneksi internet. Pada pendaftaran praktikum nilai dan jadwal akan langsung masuk dalam 1 bagian karena tombol *RadioButton* pada jadwal praktikum memiliki fungsi memasukkan satu nilai yang sama kedalam database yang berbeda. Adapun tampilan ketika *user* berhasil mendaftar ditunjukkan pada gambar dibawah ini.



Gambar 4.23 Database Jadwal Praktikum

Adapun cuplikan kode yang digunakan untuk memanggil database yang berada pada firebase cloud firestore *user* ditujukan pada gambar dibawah ini .

```

184 private void daftarselasa() {
185     daftarpraktikum.setOnClickListener(new View.OnClickListener() {
186         @Override
187         public void onClick(View view) {
188             dialog = new Dialog(SIPE.this);
189             dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);
190             dialog.setContentView(R.layout.custom_box_daftar);
191             dialog.show();
192             Window window = dialog.getWindow();
193             window.setLayout(LinearLayout.LayoutParams.MATCH_PARENT, LinearLayout.LayoutParams.WRAP_CONTENT);
194
195             Button bt_yes = (Button)dialog.findViewById(R.id.btn_yes);
196             Button bt_no = (Button)dialog.findViewById(R.id.btn_no);
197
198             bt_yes.setOnClickListener(new View.OnClickListener() {
199                 @Override
200                 public void onClick(View v) {
201                     final Map<String, Object> selasa0900= new HashMap<>();
202                     selasa0900.put("hari", "Selasa");
203                     selasa0900.put("judul", "Pengantar Teknik Elektro");
204                     selasa0900.put("tempat", "Laboratorium Teknik Elektro");
205                     selasa0900.put("jam", "09:00 - 10:30");
206                     selasa0900.put("semester", "Semester 1");
207                     selasa0900.put("unit1", "-");
208                     selasa0900.put("unit2", "-");
209                     selasa0900.put("unit3", "-");
210                     selasa0900.put("unit4", "-");
211                     selasa0900.put("unit5", "-");
212                     selasa0900.put("unit6", "-");
213                     selasa0900.put("unit7", "-");
214                     selasa0900.put("responsi", "-");
215                     uid_ref.child("Praktikum").child("Pengantar Teknik Elektro").setValue(selasa0900)
216
217                 .addOnSuccessListener(new OnSuccessListener<Void>() {
218                     @Override
219                     public void onSuccess(Void aVoid) {
220                         db.collection("Pengantar Teknik Elektro").document(userEmail).collection("Nilai Praktikum")
221                         .document("Pengantar Teknik Elektro").set(selasa0900);
222                         Toast.makeText(SIPE.this,"Pendaftaran Berhasil",Toast.LENGTH_SHORT).show();

```

Gambar 4.24 Cuplikan Kode Jadwal Praktikum

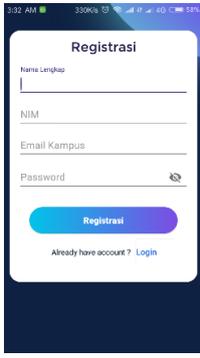
## 4.2 Pengujian

Pengujian aplikasi bertujuan untuk mengetahui apakah aplikasi yang telah dibuat sudah sesuai dengan yang diinginkan. Pengujian juga berfungsi untuk menghindari berbagai kesalahan yang bisa saja terjadi ketika digunakan oleh *user*.

#### 4.2.1 Pengujian User Interface

Pengujian user interface dilakukan untuk mengetahui fungsionalitas tiap elemen yang terdapat pada aplikasi. Elemen yang diujikan adalah fungsi dari *button* pada setiap halaman. Hasil pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Pengujian User Interface

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
1.	Tombol Login		Sistem dapat menampilkan halaman utama		Berhasil
2.	Tombol Registrasi		Sistem dapat menampilkan halaman registrasi		Berhasil
3.	Tombol Informasi Praktikum		Sistem dapat menampilkan halaman informasi praktikum		Berhasil

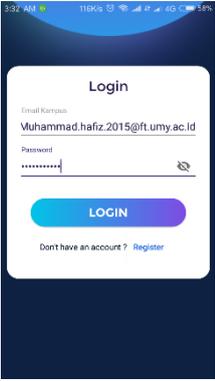
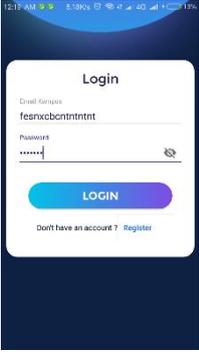
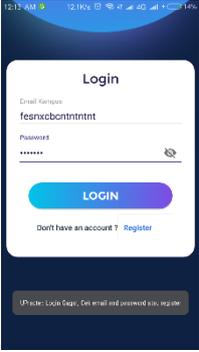
4.	Tombol Nilai Praktikum	 <p>Nilai</p>	Sistem dapat menampilkan halaman nilai praktikum		Berhasil
5.	Tombol List praktikum	 <p>Praktikum</p>	Sistem dapat menampilkan halaman list praktikum		Berhasil
6.	Tombol Pilih Jadwal		Sistem dapat menampilkan tanda jadwal telah dipilih		Berhasil

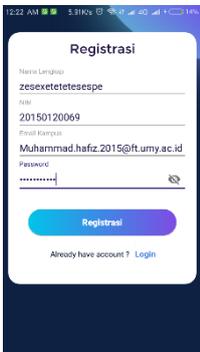
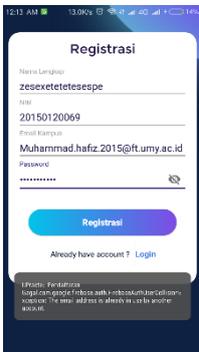
7.	Tombol Daftar Praktikum		Sistem dapat menampilkan notifikasi pendaftaran berhasil		Berhasil
----	-------------------------	---	--	--	----------

#### 4.2.2 Pengujian Fungsi Sistem

Pengujian fungsi sistem dilakukan untuk mengetahui apakah fungsi yang ada pada sistem sesuai dengan yang diharapkan. Hasil pengujian dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Pengujian Fungsi Sistem

No	Kasus yang diuji	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Status
1.	Mahasiswa melakukan login		Mahasiswa dapat masuk ke halaman utama		Berhasil
2.	Validasi Login		Sistem dapat menampilkan peringatan bahwa email tidak lengkap		Berhasil

3.	Validasi Register jika email sudah digunakan		Sistem dapat menampilkan bahwa email telah digunakan		Berhasil
4.	Fungsi daftar praktikum		Sistem dapat menampilkan Data hasil pendaftaran pada jadwal praktikum		Berhasil
5.	Fungsi hapus praktikum		Sistem dapat menghapus data dari jadwal praktikum		Berhasil

### 4.2.3 Hasil Kuisisioner Pengujian Aplikasi

Dalam penelitian ini kuisisioner digunakan sebagai alat untuk melakukan pengumpulan data. Dalam ujicoba yang dilakukan pengujian sistem aplikasi pada pengisian kuisisioner oleh mahasiswa tingkat akhir di teknik elektro. Sampel yang diambil menggunakan sampel 15 responden. Hal ini dilakukan untuk melihat apakah sistem ini sudah layak dan melihat seberapa besar kebutuhannya oleh pengguna. Data responden diperoleh melalui pengisian kuisisioner yang telah disebarakan kepada responden baik melalui online maupun offline. Berikut data hasil pengujian sistem yang telah dilakukan.

#### 4.2.3.1 Data Pengisi Kuisisioner Pengujian Aplikasi

Pengujian sistem dilakukan pada mahasiswa tingkat akhir dan alumni teknik elektro. Data pengisi kuisisioner dapat dilihat pada tabel 4.5

Tabel 4.5 Data Pengisi Kuisisioner

No	Nama	NIM	Angkatan
1	Riza Septian	20150120051	2015
2	Suko Ferbriyanto	20150120087	2015
3	Ratna Murti	20150120114	2015
4	Girindra Rosyadi	20150120130	2015
5	Muhammad Abduh Dahlan	20150120140	2015
6	Doane Puri Mustika	20150120163	2015
7	Salman Al Farisi	20150120092	2015
8	Hafiz Zuhri	20150120061	2015
9	Dwiki Rachmanto	20150120029	2015
10	Nurohman Fadilah	20150120081	2015
11	Muhammad Arif Zakaria	20150120094	2015
12	Gaga Hadika Revangga S.	20150120086	2015
13	Deny Febrian	20150120079	2015
14	Dyan Kurnia Pratama	20150120023	2015
15	Fatkhurrohman	20150120096	2015

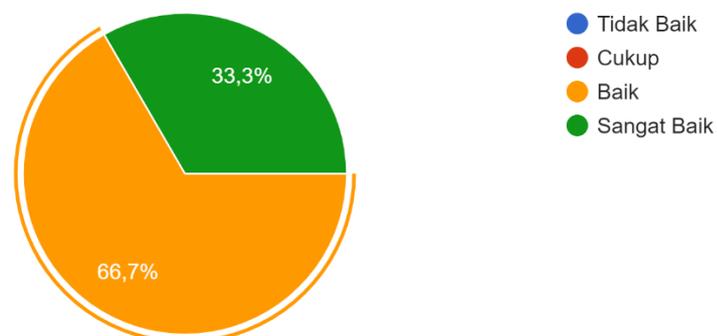
16	Reindo Aris Saputra	20150120028	2015
17	Nur Ikhsan Ramadhan	20150120090	2015
18	Anggra Eka Kurniawan	20150120058	2015
19	Akbar Kurbana	20150120080	2015
20	Elneo Bahari	20150120068	2015
21	Restu Muhammad Afdhil	20150120088	2015
22	Ilham Egi Anshori	20160120105	2016
23	Rosian Geohera	20160120092	2016
24	Fikri Haiqal Sandi	20160120109	2016
25	Rexy Libiansa	20160120097	2016
26	Dwi Nofrianto	20160120090	2016
27	Julnila Husna Lubis	20160120134	2016
28	Agil Peggy Irawan	20160120087	2016
29	Lentera Hinggis	20160120046	2016
30	Achmad Zulfikar Aditya	20160120133	2016

#### 4.2.3.2 Data Hasil Kuisisioner

Berikut adalah hasil dari kuisisioner yang telah diisi oleh data pengisi kuisisioner. Berdasarkan kuisisioner tersebut didapatkan hasil sebagai berikut :

##### 1. Bagaimana Fungsi dari Registrasi ?

30 tanggapan

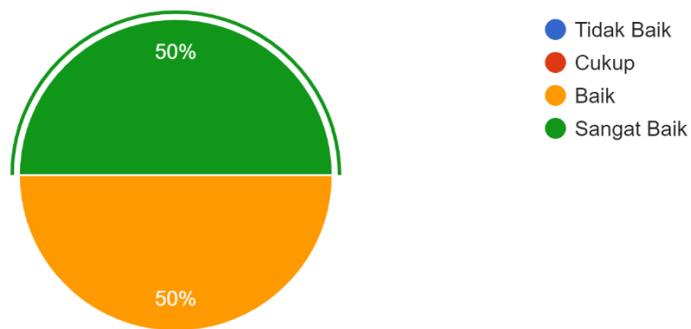


Gambar 4.25 Grafik Fungsi Registrasi

Dari hasil uji coba aplikasi, tentang fungsi dari registrasi berupa pendaftaran akun untuk *user* didapatkan dari 30 responden bahwa 66,7% mengatakan bahwa fungsinya sudah baik dan 33,3% mengatakan fungsinya sudah baik. Sehingga mayoritas responden menilai fungsi dari registrasi sudah baik.

## 2. Bagaimana Fungsi dari Login ?

30 tanggapan

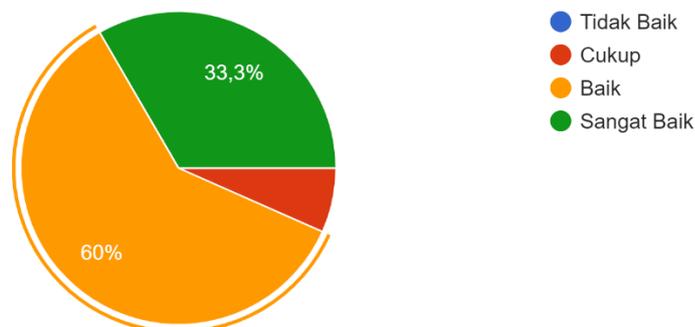


Gambar 4.26 Grafik Fungsi Login

Dari hasil uji coba aplikasi, tentang fungsi dari login yang berupa langkah untuk memasuki halaman utama untuk *user* didapatkan dari 30 responden bahwa 50% mengatakan bahwa fungsinya sudah sangat baik dan 50% lainnya mengatakan fungsinya sudah baik.

## 3. Bagaimana Fungsi dari Informasi Praktikum ?

30 tanggapan

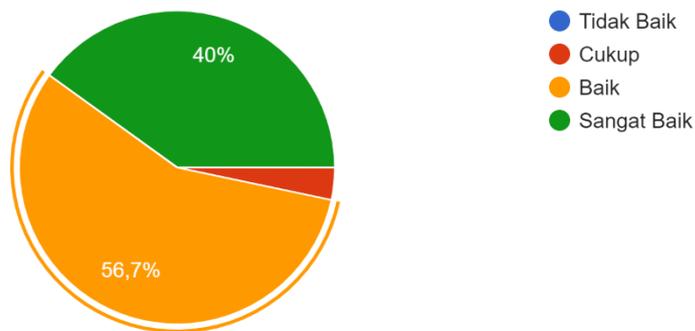


Gambar 4.27 Grafik Fungsi Informasi Praktikum

Dari hasil uji coba aplikasi, tentang fungsi dari informasi praktikum yang berupa kumpulan informasi untuk *user* dengan memiliki judul, pembuat, tanggal, hingga isi. Didapatkan dari 30 responden bahwa 60% mengatakan bahwa fungsinya sudah baik dan 33,3% mengatakan fungsinya sudah sangat baik, dan 6,7% lainnya mengatakan fungsinya sudah cukup Sehingga mayoritas responden menilai fungsi dari informasi praktikum sudah baik.

#### 4. Bagaimana Fungsi dari Nilai Praktikum ?

30 tanggapan

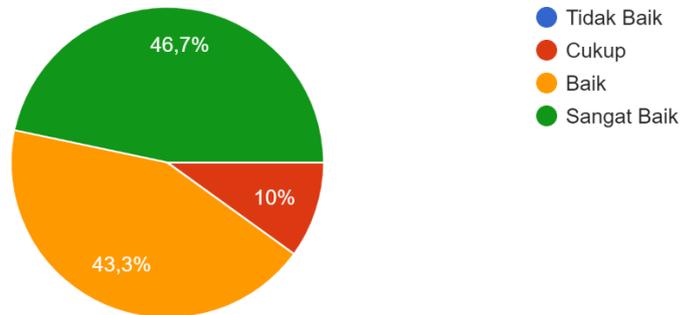


Gambar 4.28 Grafik Fungsi Nilai Praktikum

Dari hasil uji coba aplikasi, tentang fungsi dari nilai praktikum yang berupa kumpulan informasi mengenai nilai *user*. Pengguna akan mendapatkan datanya setelah mendaftar praktikum, nilai akan muncul dengan beberapa keterangan antara lain judul, semester, hari, tanggal, tempat, nilai unit 1-8 dan responsi. Dari data pengujian aplikasi didapatkan dari 30 responden bahwa 56,7% mengatakan bahwa fungsinya sudah baik, 40% mengatakan fungsinya sudah sangat baik dan 3,7% lainnya mengatakan sudah cukup Sehingga mayoritas responden menilai fungsi dari nilai praktikum baik.

#### 5. Bagaimana Penjelasan Informasi dari tiap Praktikum ?

30 tanggapan

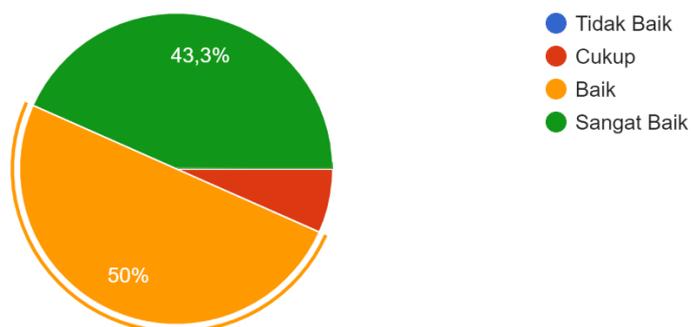


Gambar 4.29 Grafik Fungsi Penjelasan Praktikum

Dari hasil uji coba aplikasi, tentang fungsi dari penjelasan informasi pada tiap praktikum yang berupa penjelasan praktikum dengan judul, daftar isi dan juga penjadwalan praktikum, didapatkan dari data 30 responden bahwa 46,7% mengatakan fungsinya penjelasannya sangat baik, 43,3% mengatakan fungsi penjelasannya baik dan 10% lainnya mengatakan fungsinya cukup. Sehingga mayoritas responden menilai penjelasan informasi dari tiap praktikum sudah sangat baik.

#### 6. Bagaimana Pendaftaran Praktikum pada Aplikasi ?

30 tanggapan



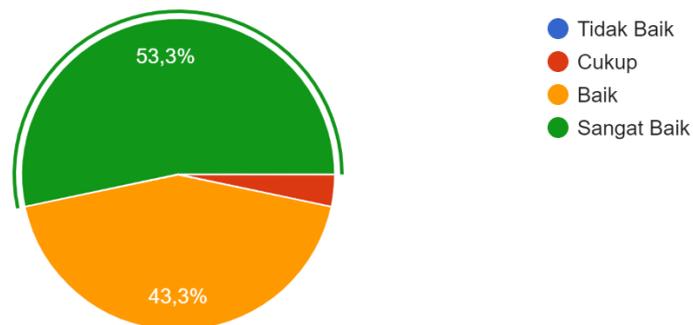
Gambar 4.30 Grafik Pendaftaran Aplikasi

Dari hasil uji coba aplikasi, tentang pendaftaran praktikum yang didapatkan dari data 30 responden bahwa 50% mengatakan fungsinya baik, dan 43,3%

mengatakan fungsinya baik dan 6,7% lainnya mengatakan sudah cukup. Sehingga mayoritas responden menilai cara pendaftaran praktikum pada aplikasi sudah sangat baik.

### 7. Bagaimana Tampilan Aplikasi ?

30 tanggapan

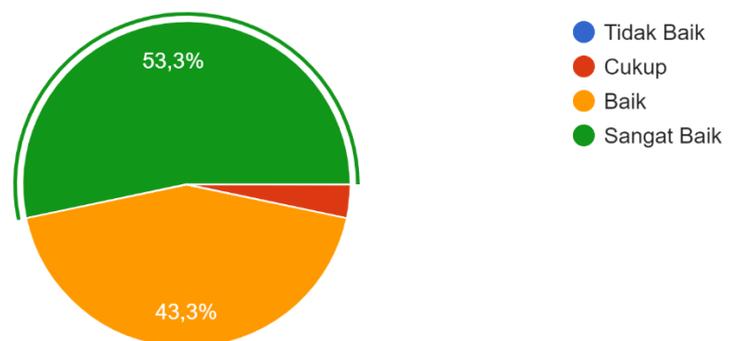


Gambar 4.31 Grafik Tampilan Aplikasi

Dari hasil uji coba aplikasi, tentang berbagai tampilan aplikasi meliputi berbagai halaman, didapatkan dari data 30 responden bahwa 53,3% mengatakan fungsinya sangat baik, 43,3% mengatakan fungsinya baik, dan 3,3% lainnya mengatakan fungsinya sudah cukup. Sehingga mayoritas responden menilai tampilan aplikasi sudah sangat baik.

### 8. Apakah Aplikasi Mudah diGunakan ?

30 tanggapan

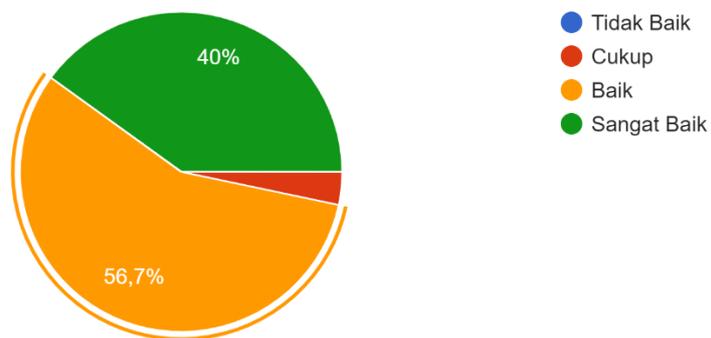


Gambar 4.32 Grafik Penggunaan Aplikasi

Dari hasil uji coba aplikasi, tentang berbagai fungsi dan kegunaan aplikasi meliputi halaman login, halaman registrasi, halaman introduce, halaman utama, halaman nilai praktikum, halaman informasi praktikum, halaman list praktikum hingga halaman details praktikuk. Pengujian aplikasi yang dilakukan dari data 30 responden bahwa 53,3% mengatakan fungsinya sudah sangat baik, 43,3% mengatakan fungsinya baik, dan 3,3% lainnya mengatakan fungsinya sudah cukup. Sehingga mayoritas responden menilai aplikasi mudah digunakan dengan sangat baik.

#### 9. Apakah Aplikasi Nyaman diGunakan ?

30 tanggapan

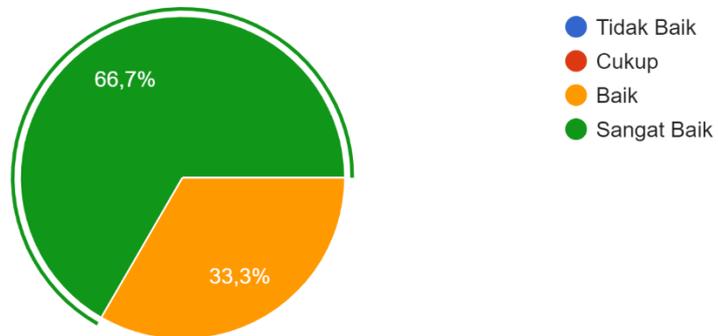


Gambar 4.33 Grafik Kenyamanan Aplikasi

Dari hasil uji coba aplikasi, tentang berbagai fungsi dan kegunaan aplikasi meliputi berbagai halaman, didapatkan dari data 30 responden bahwa 56,7% mengatakan fungsinya baik, dan 40% mengatakan fungsinya baik dan 3,3% lainnya mengatakan sudah cukup. Sehingga mayoritas responden menilai aplikasi nyaman digunakan dengan baik.

#### 10. Apakah Aplikasi Bermanfaat Bagi Mahasiswa Teknik Elektro UMY

30 tanggapan

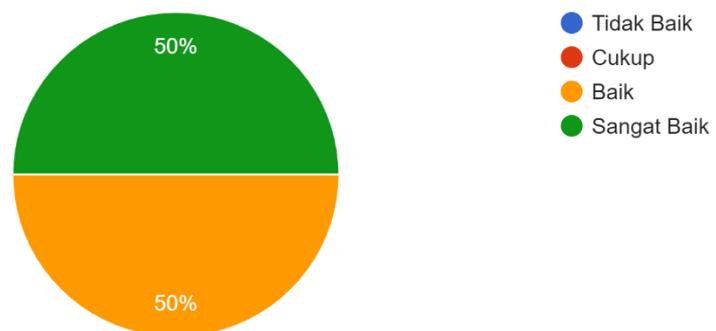


Gambar 4.34 Grafik Manfaat Aplikasi

Dari hasil uji coba aplikasi, tentang berbagai fungsi dan kegunaan aplikasi sehingga para responden mendapatkan kesimpulan, dimana dari data 30 responden bahwa hanya 33,3% yang mengatakan fungsinya baik, dan 66,7% mengatakan fungsinya sangat baik. Sehingga mayoritas responden menilai bahwa aplikasi aplikasi ini akan sangat bermanfaat bagi mahasiswa teknik elektro UMY dengan sangat baik.

#### 11. Apakah Aplikasi Mempunyai Fungsi yang diHarapkan ?

30 tanggapan



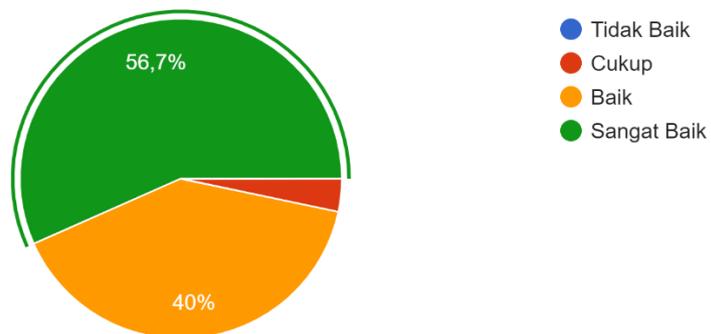
Gambar 4.35 Grafik Harapan Fungsi Aplikasi

Dari hasil uji coba aplikasi, para responden memberikan penilaian, dimana dari data 30 responden didapatkan bahwa hanya 50% yang mengatakan fungsinya sangat baik, dan 50% lainnya mengatakan fungsinya baik. Sehingga responden

menilai bahwa aplikasi aplikasi ini sudah memiliki fungsi yang sangat baik sesuai dengan harapan *user*.

## 12. Apakah Aplikasi Sudah Memenuhi Kebutuhan Praktikum diLaboratorium Teknik Elektro UMY ?

30 tanggapan



Gambar 4.36 Grafik Kebutuhan Aplikasi Pada Praktikum

Dari hasil uji coba aplikasi, tentang berbagai fungsi dan kegunaan aplikasi, didapatkan dari data 30 responden bahwa 3,3% mengatakan cukup, 40% mengatakan baik, dan 56,7% mengatakan kebutuhan aplikasi pada praktikum sangat baik. Sehingga mayoritas responden menilai aplikasi mudah memenuhi kebutuhan praktikum dilaboratorium teknik elektro UMY dengan sangat baik.

### 4.2.3.3 Pengembangan

Dari hasil pengujian terhadap 30 responden melalui pembagian kuisisioner, didapatkan beberapa saran dalam pengembangan aplikasi praktikum teknik elektro UMY. Berikut beberapa saran yang telah diberikan oleh responden :

1. Dibuat agar semua platform bisa menggunakan
2. Untukantisipasi bentrok jadwal maka sistem aplikasi dibuat agar bisa terintegrasi dengan sistem krs online sehingga pendaftaran praktikum dan juga pengisian praktikum tidak ada kesalahan dan juga tidak adanya bentrokan dengan lebih spesifik.

3. Perlu adanya sosialisasi antara mahasiswa dan dosen agar sinergis dalam pemanfaatan aplikasi ini.
4. Sebaiknya ditambahkan kuota peserta untuk setiap jadwal agar mahasiswa lebih tau tentang kuota di setiap jamnya dan mohon dicari solusinya apabila di tengah jalan ada jadwal kelas yang berubah dan bentrok dengan jadwal praktikum
5. Dibuat lebih sederhana
6. Ditambahkan menu untuk interaksi antara laboran / asisten lab kepada praktikan, dengan diberikan pengumuman yang lebih spesifik pada kelas masing-masing. Dan juga praktikan diberi kemampuan untuk menilai asisten lab dari aplikasi ini.