

BAB IV

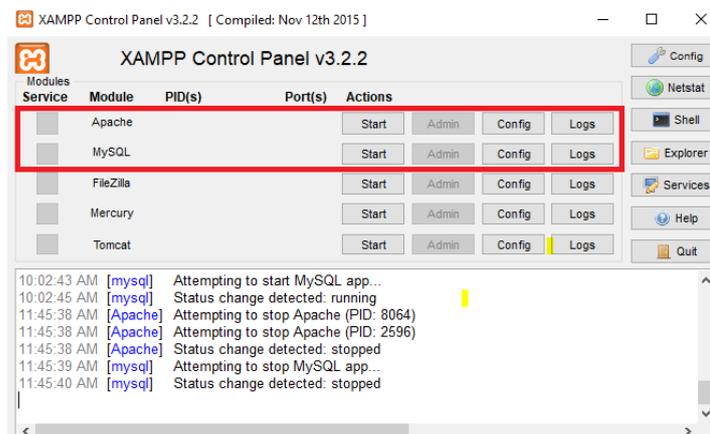
HASIL DAN PEMBAHASAN

4.1 Implementasi

Hasil dari penelitian ini merupakan sistem *assessment* yang diterapkan pada aplikasi *mobile* dan *desktop*. Sistem yang dimaksud digunakan untuk membantu instruktur atau guru serta siswa dalam pelaksanaan *assessment* yang dilaksanakan dalam bentuk ujian maupun penilaian harian.

4.1.1 Database Sistem

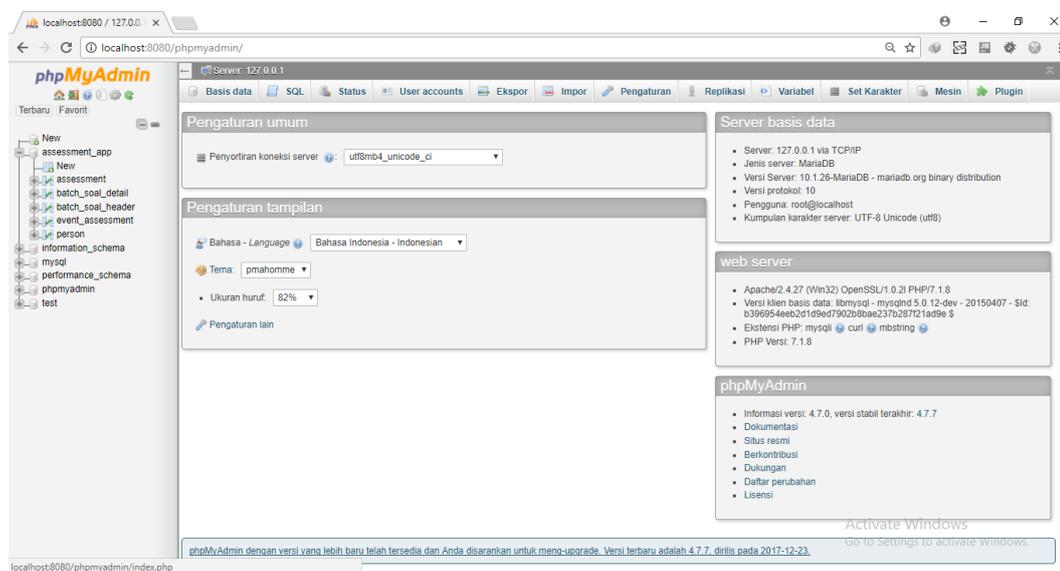
Database pada sistem ini dibuat menggunakan *MySQL Server* yang telah terinstal satu paket dengan *XMPP Server*. Agar dapat mengakses *MySQL Server* secara lokal, *hosting* secara lokal harus diaktifkan. Dengan memanfaatkan fitur *XAMPP server*, komputer dapat dijadikan *server* sekaligus *client*. Hal ini dilakukan guna mempermudah proses pengembangan aplikasi, sehingga proses pengembangan dapat dilakukan tanpa menggunakan koneksi internet. Adapun cara mengaktifkan *database* dan *hosting* secara lokal ditunjukkan oleh gambar 4.1 dibawah ini.



Gambar 4. 1 Aktivasi *database* dan *hosting* secara lokal pada komputer

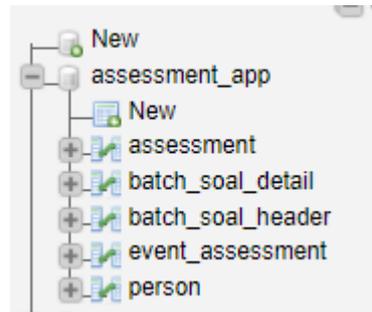
Dapat dilihat pada gambar 4.1 terdapat daerah yang ditandai dengan kolom berwarna merah terdapat Apache dan MySQL. Komponen tersebut merupakan hal yang dibutuhkan guna mengaktifkan *database* dan *hosting* secara lokal. Dengan menekan tombol *start* pada komponen tersebut maka sistem akan memulai aktivasi. Jika tulisan pada tombol berubah menjadi *stop* menandakan bahwa proses aktivasi telah berhasil.

Setelah selesai melakukan aktivasi, *database* dan *server* dapat diakses melalui *web browser* dengan cara mengakses alamat <http://localhost:8080/phpmyadmin> sehingga akan muncul tampilan *database engine* seperti yang ditunjukkan pada gambar 4.2 di bawah ini.



Gambar 4. 2 Database Engine Server

Database yang dibuat untuk sistem ini dibuat dengan nama *assessment_app*. *Database* ini terdiri dari 5 tabel yaitu; *assessment*, *batch_soal_detail*, *batch_soal_header*, *event_assessment*, dan *person* seperti yang ditunjukkan pada gambar 4.3 di bawah ini.



Gambar 4.3 Database dan Tabel Sistem

Setiap tabel memiliki fungsi sebagai penyimpanan data yang berbeda. Tabel *assessment* digunakan sebagai penyimpan data *assessment* yang telah dibuat oleh instruktur. Tabel *batch_soal_header* digunakan sebagai penyimpanan data *batch* soal yang telah dibuat instruktur. Tabel *batch_soal_detail* berisi data soal yang berhubungan dengan tabel *batch_soal_header*. Tabel *event_assessment* berisi data hasil *assessment* yang telah dikerjakan oleh siswa. Tabel *person* berisi data siswa dan instruktur yang digunakan sebagai identitas untuk *login*.

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Komentar	Ekstra	Tindakan
1	ID	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
2	NAMA_ASSESSMENT	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	TANGGAL_MULAI	datetime		Tidak	CURRENT_TIMESTAMP			Ubah Hapus Lainnya
4	AKTIF_SAMPAI_DENGAN	datetime		Tidak	Tidak ada			Ubah Hapus Lainnya
5	DURASI_PENGERJAAN	int(11)		Tidak	Tidak ada			Ubah Hapus Lainnya
6	JUMLAH_SOAL	int(11)		Tidak	Tidak ada			Ubah Hapus Lainnya
7	MATA_KULIAH	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
8	ENROLLMENT_CODE	varchar(50) latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
9	ID_INSTRUKTUR	varchar(30) latin1_swedish_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
10	ID_BATCH_SOAL_HEADER	int(11)		Tidak	Tidak ada			Ubah Hapus Lainnya

Gambar 4.4 Tabel assessment

Gambar 4.4 menunjukkan kolom yang ada pada tabel *assessment*. Tabel ini terdiri dari kolom *ID* sebagai *primary key*, kolom *ID_INSTRUKTUR* dan *ID_BATCH_SOALHEADER* sebagai *foreign key*. Kolom *foreign key* *ID_INSTRUKTUR* menandakan bahwa adanya relasi antar tabel *assessment* dengan tabel *person*. Sedangkan adanya kolom *foreign key*

ID_BATCH_SOAL_HEADER menandakan adanya relasi antar tabel ini dengan tabel batch_soal_header.

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Komentar Ekstra	Tindakan
1	ID	int(11)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Lainnya
2	NAMA_BATCH_SOAL	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Lainnya
3	JUMLAH_SOAL	int(11)		Tidak	Tidak ada		Ubah Hapus Lainnya
4	TANGGAL_DIBUAT	datetime		Tidak	CURRENT_TIMESTAMP		Ubah Hapus Lainnya
5	ID_INSTRUKTUR	varchar(50)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Lainnya

Gambar 4.5 Tabel batch_soal_header

Gambar 4.5 menunjukkan kolom yang terdapat pada tabel batch_soal_header. Terdapat kolom ID sebagai *primary key* dan ID_INSTRUKTUR sebagai *foreign key* yang menandakan bahwa tabel ini terelasi dengan tabel person.

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Komentar Ekstra	Tindakan
1	ID	int(11)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Lainnya
2	SOAL	varchar(1000)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
3	A	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
4	B	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
5	C	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
6	D	varchar(500)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
7	JAWABAN	varchar(1)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
8	ID_HEADER	int(11)		Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya

Gambar 4.6 Tabel batch_soal_detail

Gambar 4.6 di atas menunjukkan kolom yang ada pada tabel batch_soal_detail. Tabel ini merupakan rincian dari tabel batch_soal_header yang ditunjukkan pada gambar 4.5. Ditandai dengan kolom ID_HEADER sebagai *foreign key* yang menunjukkan bahwa tabel ini terelasi dengan tabel batch_soal_header.

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Komentar Ekstra	Tindakan
1	ID	int(11)		Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Lainnya
2	ID_SISWA	varchar(100)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
3	ID_ASSESSMENT	varchar(100)	latin1_swedish_ci	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya
4	SKOR	int(11)		Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Lainnya

Gambar 4.7 Tabel event_assessment

Gambar 4.7 menunjukkan kolom yang ada pada tabel event_assessment. Tabel ini merupakan tabel yang digunakan untuk menyimpan data hasil pekerjaan atau skor *assessment* siswa. Tabel ini terelasi dengan tabel person dan assessment

yang ditandai dengan adanya kolom `ID_SISWA` dan `ID_ASSESSMENT` sebagai *foreign key*.

#	Nama	Jenis	Penyortiran	Atribut Kosong	Bawaan	Komentar	Ekstra	Tindakan
1	ID	varchar(100)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Lainnya
2	NAMA	varchar(100)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Kunci Utama Unik Indeks Lainnya
3	PASS	varchar(10)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Kunci Utama Unik Indeks Lainnya
4	EMAIL	varchar(100)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Kunci Utama Unik Indeks Lainnya
5	ROLE	varchar(20)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Kunci Utama Unik Indeks Lainnya
6	NOMOR_INDUK	varchar(50)	latin1_swedish_ci	Ya	NULL			Ubah Hapus Kunci Utama Unik Indeks Lainnya

Gambar 4. 8 Tabel person

Gambar 4.8 menunjukkan kolom yang ada pada tabel person. Tabel ini digunakan untuk menyimpan data siswa maupun instruktur yang akan digunakan sebagai identitas sekaligus otorisasi dan autentikasi. Data dibedakan berdasarkan *role*-nya yang ditandai dengan adanya kolom `ROLE` pada tabel ini.

4.1.2 Web Service

Web service pada sistem ini berupa *RESTful service* yang dibangun menggunakan PHP dengan memanfaatkan *Slim framework* yang berguna untuk mempermudah *routing* terhadap setiap fungsi berdasarkan alamat URI.

```
$app->get( pattern: "/assessment/get/assmnt_id/{id}", function ($request) {
```

Gambar 4. 9 Contoh penggunaan Slim Framework pada PHP

Gambar 4.9 menunjukkan salah satu cara penggunaan *Slim framework* di PHP. Terdapat sintaks “*get*” yang menandakan bahwa fungsi tersebut berupa *method GET* atau fungsi yang digunakan untuk mengambil data. *Pattern* yang ada di dalamnya merupakan URI yang digunakan untuk *routing* terhadap fungsi yang dibuat. Setiap fungsi memiliki URI yang berbeda karena setiap fungsi memiliki *routing*-nya masing-masing.

Setiap fungsi yang dibuat untuk kebutuhan *service* bekerja dengan cara melakukan eksekusi sintaks *query* dan merepresentasikan hasilnya ke dalam format JSON yang nantinya akan digunakan oleh *client*. Adapun contoh fungsi yang dimaksud dapat dilihat pada gambar 4.10 di bawah ini.

```

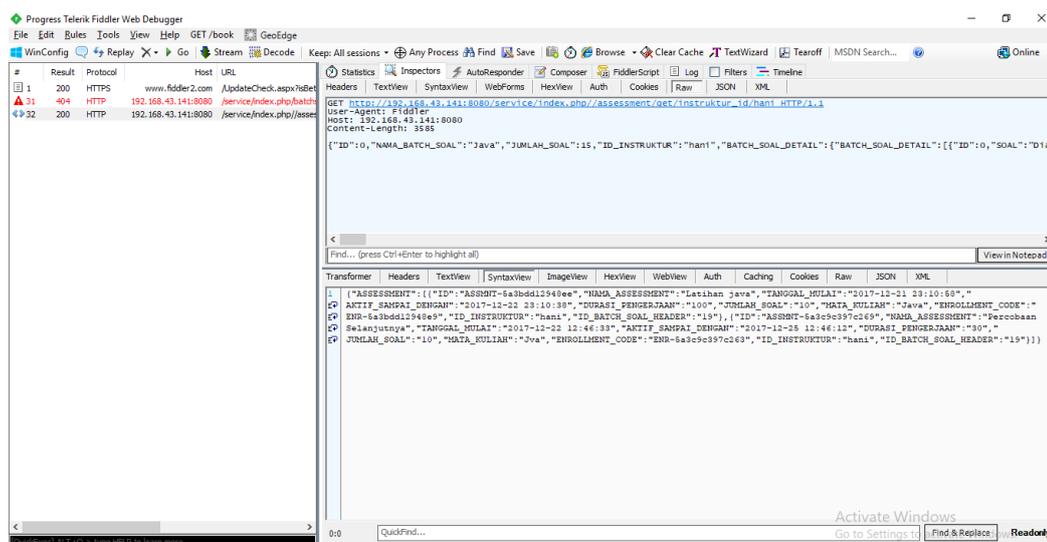
$app->get( pattern: '/assessment/get/instruktur_id/{id}', function ($request) {
    $id = $request->getAttribute('id');

    try{
        $db = getConnection();
        $query = "SELECT * FROM assessment WHERE ID_INSTRUKTUR='{$id}'";
        $result = mysqli_query($db,$query) or die($db->error);
        $data = array();
        if (mysqli_num_rows($result)>0){
            while ($row = mysqli_fetch_assoc($result)){
                $data[] = $row;
            }
            echo json_encode(array('ASSESSMENT'=>$data));
        }
    }catch (PDOException $e){
        echo '{"error":{"text":'. $e->getMessage() .'}}';
    }
});

```

Gambar 4. 10 Contoh fungsi untuk *service* pada PHP

Agar *web service* dapat berjalan *file* PHP yang dibuat harus diletakkan pada direktori *server*. *Web Service* dapat diuji dengan menggunakan aplikasi *web debugging proxy*, salah satunya yaitu Fiddler. Adapun hasil pengujian menggunakan Fiddler ditunjukkan pada gambar 4.11 di bawah ini.



Gambar 4. 11 Hasil pengujian *web service* menggunakan Fiddler

Pengujian *web service* dilakukan dengan melakukan *request* terhadap alamat URI

http://192.168.43.141:8080/service/index.php//assessment/get/instruktur_id/hani.

Alamat URI tersebut dihasilkan dari gabungan antara alamat IP *server*, *port* yang digunakan, direktori *file* PHP pada *server*, dan URI yang digunakan sebagai *routing* terhadap fungsi yang dituju. Hasil pengujian dapat dilihat dari *response* yang ditentukan dengan *result code*, apabila hasilnya 200 maka *request* mendapatkan *response* sesuai dengan yang diharapkan. Adapun isi pesan dari *response* yaitu berupa bentuk kembalian JSON dapat dilihat pada gambar 4.12 dibawah ini.



Gambar 4. 12 Isi pesan dari *response* yang merupakan data dalam bentuk JSON

Data JSON tersebutlah yang nantinya akan diolah di bagian *client*. Dengan memanfaatkan sebuah *library*, yaitu Retrofit pengelolaan data dalam bentuk JSON pada *client* akan lebih mudah. *Library* ini bekerja dengan cara melakukan proses *request* dan *response* berdasarkan fungsi yang dibuat dalam sebuah *class interface*.

Gambar 4.13 di bawah menunjukkan contoh *class interface* yang dimaksud.

```

public interface ApiService {

    @GET("person/get/id/{id}/pass/{pass}/role/{role}")
    Call <Person> getAuthResult(@Path("id") String id,
                               @Path("pass") String pass,
                               @Path("role") String role);

    @GET("person/get/id/{id}/role/{role}")
    Call <Person> getPersonByIdandRole(@Path("id") String id,
                                       @Path("role") String role);

    @POST("person/add")
    Call <Person> addPerson(@Body Person person);

    @POST("nilai/add")
    Call <Nilai> addNilai(@Body Nilai nilai);

    @GET("assessment/get/id/{id}/id_siswa/{id_siswa}")
    Call <Assessment> checkAssessmentAvailability(@Path("id") String id,
                                                  @Path("id_siswa") String id_siswa);

    @GET("assessment/get/assmnt_id/{id}")
    Call <Assessment> getAssessmentById(@Path("id") String id);

    @GET("batchsoaldetail/get/id_batchsoal/{id}/assmnt_id/{assmnt_id}")
    Call <BatchSoalDetailList> getBatchSoalDetailByBatchId(@Path("id") String id,
                                                           @Path("assmnt_id") String assmntId);
}

```

Gambar 4.13 Class interface yang berisi fungsi untuk memanggil service

Setiap fungsi yang dibuat ditambahkan dengan sebuah anotasi di atasnya. Anotasi tersebut merupakan penanda jenis *request method* sekaligus *routing* menuju *resource* yang dituju. Bisa dilihat pada gambar 4.13 di atas. Terdapat sintaks yang berupa anotasi @GET yang menandakan bahwa fungsi tersebut memiliki tipe *request method GET* atau mengambil data. Adapun anotasi lainnya yaitu @PUT untuk fungsi memperbarui data, @POST untuk fungsi menambah data, dan @DELETE untuk fungsi menghapus data. Di dalam anotasi tersebut terdapat sebuah *URI template* yang akan dikombinasikan dengan *base address* sehingga terbentuk alamat URI yang sesuai yang akan digunakan oleh sistem untuk melakukan proses *request*.

Base address merupakan potongan alamat yang konsisten digunakan pada saat proses *request* berisi alamat *server* dan direktori dari *service*. Definisi dari *base address* ini dilakukan pada kelas terpisah seperti yang ditunjukkan pada gambar 4.14 di bawah ini.

```

public class ApiUtils {
    private ApiUtils() {}

    private static final String baseUrl="http://192.168.43.141:8080/service/index.php/";
    // private static final String baseUrl="http://10.69.8.39:8080/service/index.php/";
    //private static final String baseUrl="http://10.89.10.123:8080/service/index.php/";

    public static ApiService getAPIService() {

        return RetrofitClient.getClient(baseUrl).create(ApiService.class);
    }
}

```

Gambar 4. 14 kelas ApiUtils

Gambar 4.14 menunjukkan kelas ApiUtils. Kelas ini digunakan sebagai definisi *base address* atau pada gambar dituliskan sebagai variabel *baseUrl*. Pada kelas ini juga terdapat sebuah *method* statis yang nantinya akan digunakan untuk menggabungkan *base address* dengan *URI template*. *Method* tersebut memanggil sebuah kelas yaitu *RetrofitClient*. Kelas ini berisi definisi dari *library* Retrofit dan batasan atau konfigurasi dari penggunaan *library* ini. Adapun isi dari kelas tersebut bisa dilihat pada gambar 4.15 di bawah ini.

```

public class RetrofitClient {

    private static Retrofit retrofit = null;

    public static Retrofit getClient(String baseUrl) {
        Gson gson = new GsonBuilder()
            .setDateFormat("yyyy-MM-dd HH:mm:ss")
            .setLenient()
            .create();
        if (retrofit==null) {
            retrofit = new Retrofit.Builder()
                .baseUrl(baseUrl)
                .addConverterFactory(GsonConverterFactory.create(gson))
                .client(getCli())
                .build();
        }
        return retrofit;
    }
    private static OkHttpClient getCli() {
        HttpLoggingInterceptor interceptor = new HttpLoggingInterceptor();
        interceptor.setLevel(HttpLoggingInterceptor.Level.BODY);
        OkHttpClient client =new OkHttpClient.Builder()
            .addInterceptor(interceptor)
            .connectTimeout(50, TimeUnit.SECONDS)
            .readTimeout(50, TimeUnit.SECONDS)
            .build();

        return client;
    }
}

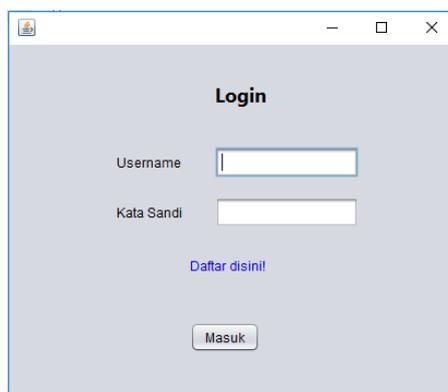
```

Gambar 4. 15 Kelas RetrofitClient

4.1.3 Aplikasi *Desktop* Instruktur

Implementasi pada aplikasi untuk instruktur dilakukan dengan menggunakan bahasa pemrograman Java dalam bentuk desktop. Aplikasi yang digunakan adalah NetBeans IDE yang merupakan produk *open source* milik Oracle. Berikut adalah tampilan antarmuka dari aplikasi instruktur.

A. *Form Login*

The image shows a window titled "Login" with a light gray background. It contains two text input fields: "Username" and "Kata Sandi". Below the "Kata Sandi" field is a blue link that says "Daftar disini!". At the bottom center is a button labeled "Masuk".

Gambar 4.16 *Form Login* untuk Instruktur

Gambar 4.16 di atas menunjukkan *Form Login* yang digunakan untuk instruktur melakukan proses autentikasi dan otorisasi. Dari *form* ini juga terdapat teks *link* menuju *form* daftar.

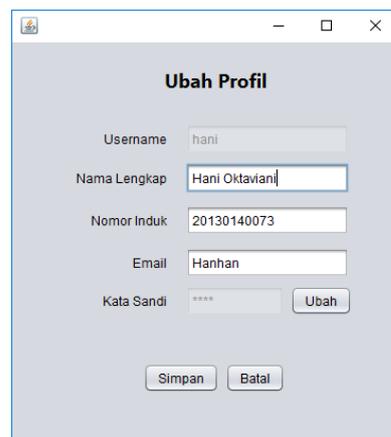
B. *Form Daftar*

The image shows a window titled "Daftar" with a light gray background. It contains six text input fields: "Username", "Nama Lengkap", "Nomor Induk", "Email", "Kata Sandi", and "Ulangi Kata Sandi". At the bottom are two buttons: "Simpan" and "Batal".

Gambar 4.17 *Form Daftar*

Gambar 4.17 di atas adalah tampilan dari *form* daftar yang digunakan oleh *user* agar dapat mendaftarkan dirinya sebagai instruktur.

C. *Form* Ubah Profil



The screenshot shows a web browser window titled "Ubah Profil". The form contains the following fields and buttons:

- Username: hani
- Nama Lengkap: Hani Oktavian
- Nomor Induk: 20130140073
- Email: Hanhan
- Kata Sandi: ****
- Buttons: Ubah, Simpan, Batal

Gambar 4.18 *Form* Ubah Profil

Gambar 4.18 di atas adalah tampilan ubah profil yang digunakan oleh instruktur untuk mengubah data diri.

D. *Form* Ubah Kata Sandi



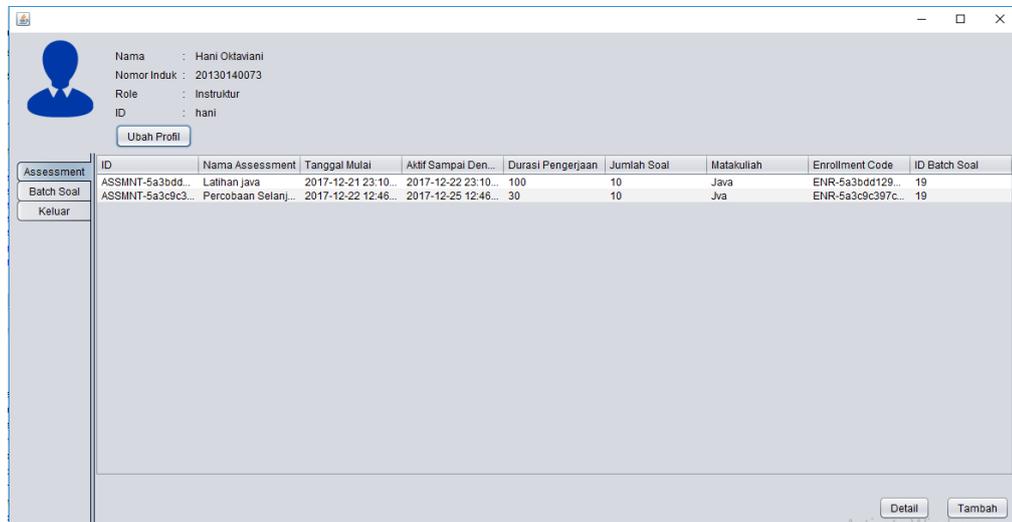
The screenshot shows a web browser window titled "Ubah Kata Sandi". The form contains the following fields and buttons:

- Kata Sandi Lama: [input field]
- Kata Sandi Baru: [input field]
- Konfirmasi Kata Sandi Baru: [input field]
- Buttons: Simpan, Batal

Gambar 4.19 *Form* Ubah Kata Sandi

Gambar 4.19 di atas merupakan tampilan ubah kata sandi yang digunakan oleh instruktur apabila ingin mengubah kata sandi.

E. Form Utama Tab Assessment

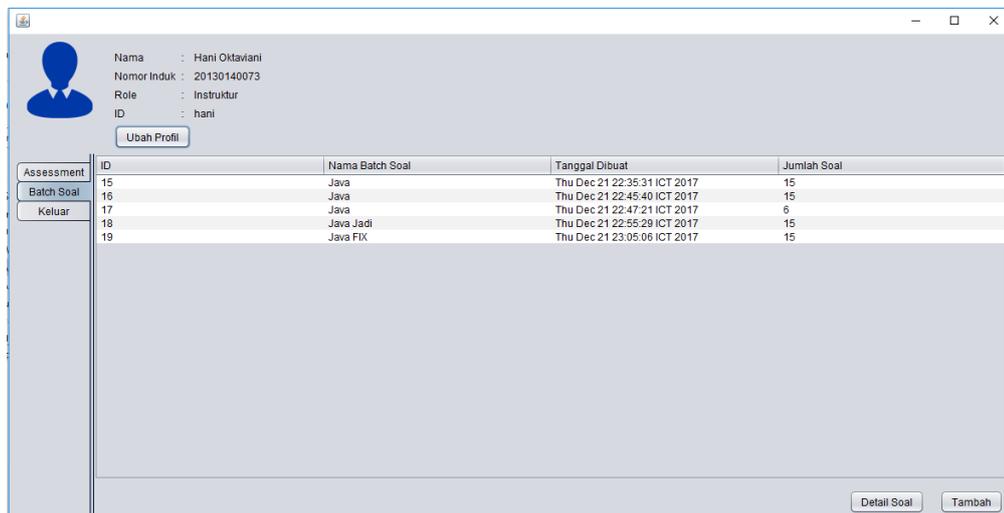


ID	Nama Assessment	Tanggal Mulai	Aktif Sampai	Durasi Pengerjaan	Jumlah Soal	Matakuliah	Enrollment Code	ID Batch Soal
ASSMNT-5a3bdd...	Latihan java	2017-12-21 23:10...	2017-12-22 23:10...	100	10	Java	ENR-5a3bdd129...	19
ASSMNT-5a3c9c3...	Percobaan Selanj...	2017-12-22 12:46...	2017-12-25 12:46...	30	10	Jva	ENR-5a3c9c397c...	19

Gambar 4. 20 Tampilan Utama Bagian *Assessment*

Gambar 4.20 di atas menunjukkan tampilan utama bagian *assessment* yang menyajikan data *assessment* yang telah dibuat dan ditampilkan dalam bentuk tabel.

F. Form Utama Tab Batch Soal

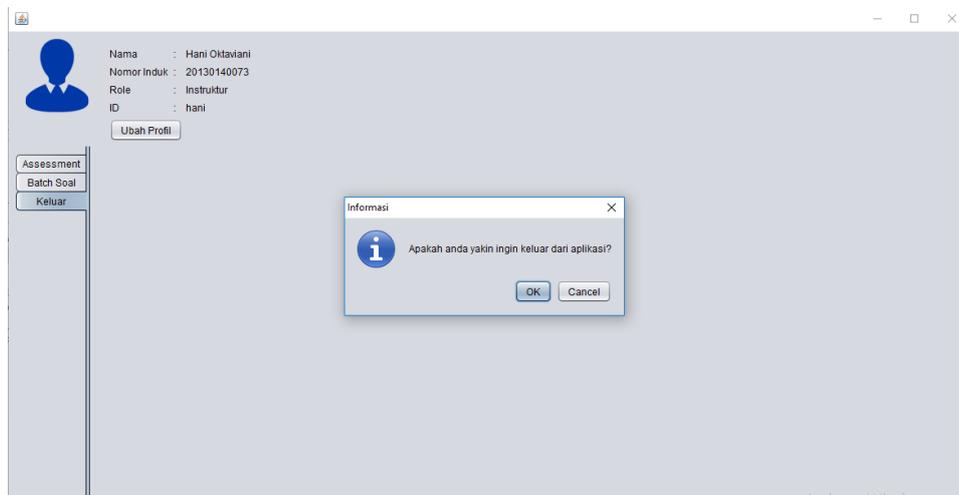


ID	Nama Batch Soal	Tanggal Dibuat	Jumlah Soal
15	Java	Thu Dec 21 22:35:31 ICT 2017	15
16	Java	Thu Dec 21 22:45:40 ICT 2017	15
17	Java	Thu Dec 21 22:47:21 ICT 2017	6
18	Java Jadi	Thu Dec 21 22:55:29 ICT 2017	15
19	Java FIX	Thu Dec 21 23:05:06 ICT 2017	15

Gambar 4. 21 Tampilan Utama Bagian *Batch Soal*

Instruktur dapat melihat daftar batch soal yang telah dibuat pada tampilan utama bagian batch soal seperti yang ditunjukkan pada gambar 4.21 di atas.

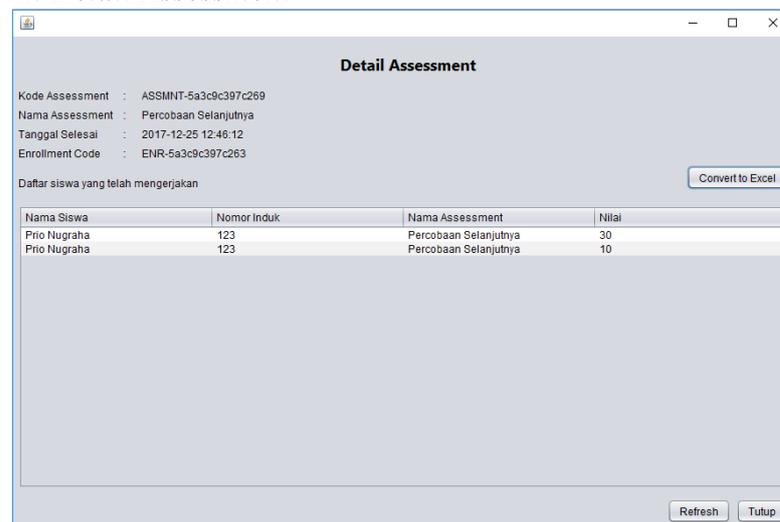
G. Form Utama Tab Keluar



Gambar 4. 22 Tampilan Utama Bagian Keluar Aplikasi

Ketika *user* akan keluar dari aplikasi, pada tampilan utama terdapat *tab* keluar. Ketika *tab* tersebut dipilih maka akan muncul dialog yang akan mengkonfirmasi *user* akan keluar atau tidak seperti yang ditunjukkan pada gambar 4.22 diatas.

H. Form Detail Assessment



Gambar 4. 23 Tampilan *Detail Assessment*

Gambar 4.23 di atas merupakan tampilan dari *detail assessment* yang berisi rincian suatu *assessment*. Dalam tampilan ini juga terdapat daftar siswa yang telah mengikuti *assessment* tersebut.

I. Form Tambah Assessment

Gambar 4. 24 Form Tambah Assessment

Gambar 4.24 di atas menunjukkan *form* tambah *assessment* yang digunakan *user* untuk membuat *assessment* baru.

J. Form Detail Batch Soal

Soal	A	B	C	D	Jawaban	ID	ID HEADER
Diantara perintah...	System.out.println...	System Out.println...	System.out.Printl...	System.out.Printl...	A	108	19
Sintaks java untuk...	java	javac	javaclass	javax	B	109	19
System yang berg...	System.in.	System.out.	System.err	System.exit	B	110	19
Hasil kompilasi d...	File BAK	File Bytecode	File executable	File class	B	111	19
Fungsi method Gy...	Mengirimkan kelu...	Menangani pemb...	Menampilkan pes...	Menangani suatu...	B	112	19
Diantara pemnyata...	Polymorphisme	Encapsulation	Multiple inheritance	Single inheritance	C	113	19
Arsitektur java yan...	J2SE	J2ME	J2EE	J2EM	B	114	19
Berikut adalah pe...	_3_One	_0_3ne	S13h	B3_Ta	A	115	19
Method yang digu...	parseInt()	converseInt()	Cont()	ChangeInt()	A	116	19
Diantara pemnyata...	int data =Buffered...	int data =Buffered...	int data=(new Inte...	int data = (new Int...	D	117	19
Berikut ini yang ter...	Boolean	character	byte	Double	C	118	19
Berikut ini yang ter...	float	Character	short	int	B	119	19
Keyword yang dig...	protected	private	public	final	D	120	19
Apa yang akan ter...	Alt:14	Alt:10	Alt:15	Alt:9	C	121	19
Method yang digu...	equal()	concat()	length()	equals()	D	122	19

Gambar 4. 25 Form Detail Soal

Gambar 4.25 di atas menunjukkan tampilan *detail* soal yang berisi rincian dari *batch* soal.

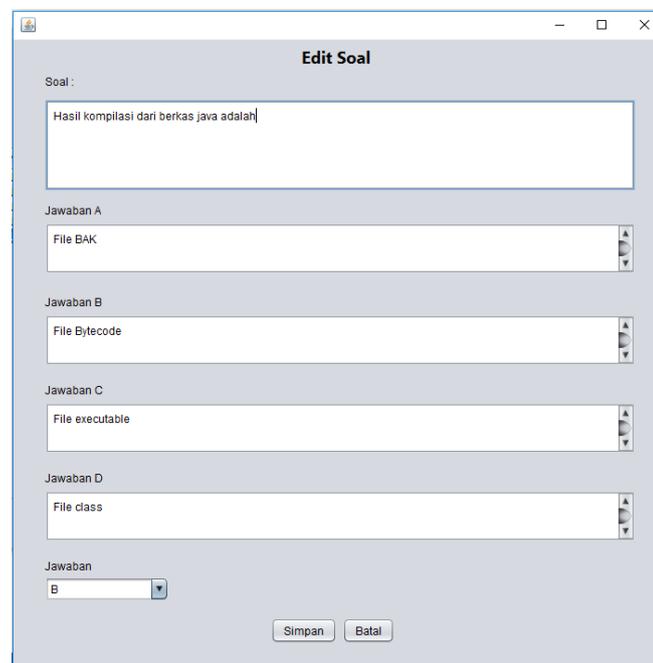
K. Form Tambah Batch Soal



Gambar 4. 26 Form Tambah Batch Soal

Gambar 4.26 menunjukkan tampilan tambah *batch* soal yang digunakan oleh instruktur untuk membuat *batch* soal baru.

L. Form Ubah Soal



Gambar 4. 27 Form Ubah Soal

Gambar 4.27 menunjukkan tampilan ubah soal yang digunakan oleh instruktur untuk mengubah isi soal.

4.1.4 Aplikasi *Mobile* Siswa

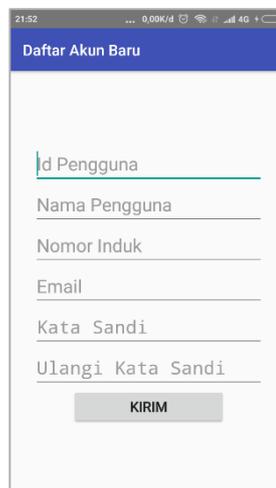
A. *Login Activity*



Gambar 4. 28 *Login Activity*

Gambar 4.28 merupakan tampilan *login activity* yang digunakan oleh siswa untuk melakukan proses autentikasi dan otorisasi.

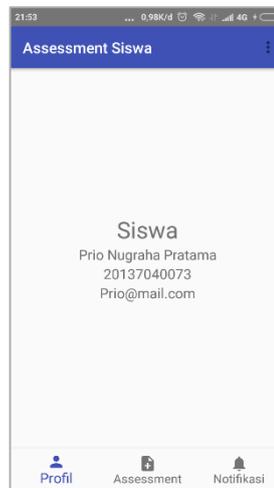
B. *Daftar Activity*



Gambar 4. 29 *Daftar Activity*

Gambar 4.29 merupakan tampilan daftar *activity* yang digunakan oleh siswa untuk melakukan proses registrasi akun baru.

C. Main Activity



Gambar 4. 30 Main Activity

Gambar 4.30 merupakan tampilan *main activity* yang pertama ditampilkan oleh sistem setelah berhasil melalui proses autentikasi dan otorisasi.

D. Ubah Profil Activity



Gambar 4. 31 Ubah Profil Activity

Gambar 4.31 merupakan tampilan ubah profil *activity* yang digunakan siswa untuk mengubah profil.

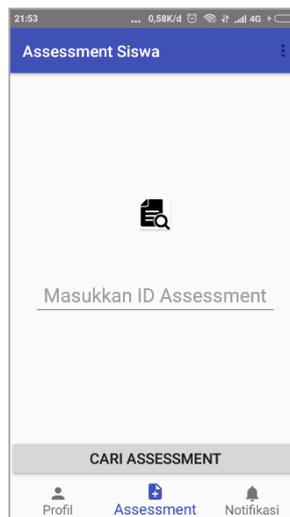
E. Ubah Kata Sandi Activity



Gambar 4. 32 Ubah Kata Sandi Activity

Gambar 4.32 adalah tampilan *activity* ubah kata sandi yang digunakan oleh siswa untuk untuk mengubah kata sandi.

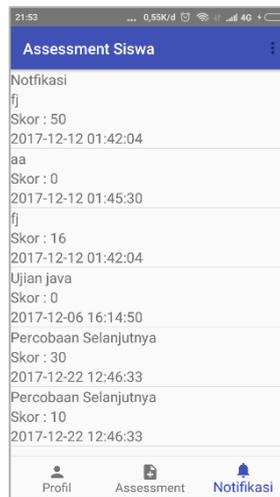
F. Main Activity (Tab Assessment)



Gambar 4. 33 Main Activity (Tab Assessment)

Gambar 4.33 merupakan tampilan *main activity* bagian *assessment* yang digunakan siswa untuk mencari *assessment* yang akan dikerjakan.

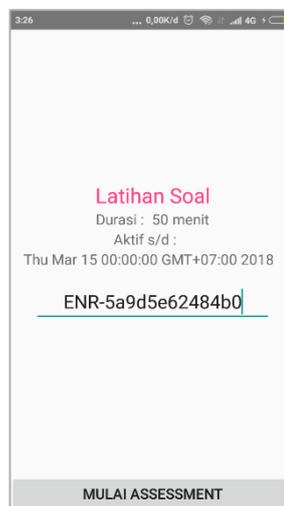
G. Main Activity (Tab Nilai)



Gambar 4.34 Main Activity (Tab Nilai)

Gambar 4.34 merupakan tampilan *main activity* bagian nilai yang menampilkan seluruh nilai dari *assessment* yang pernah dikerjakan oleh siswa tersebut.

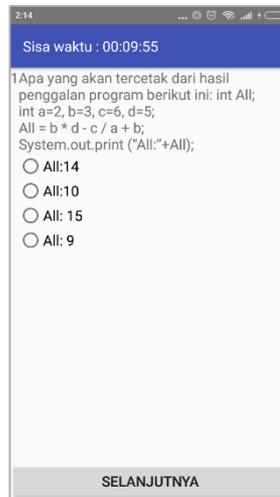
H. Cek Enrollment Activity



Gambar 4.35 Cek Enrollment Activity

Gambar 4.35 merupakan tampilan cek *enrollment* yang akan muncul setelah melalui proses pencarian *assessment*.

I. Kerjakan Soal Activity



Gambar 4. 36 Kerjakan Soal Activity

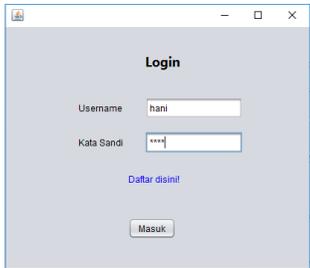
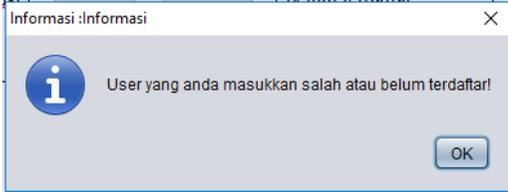
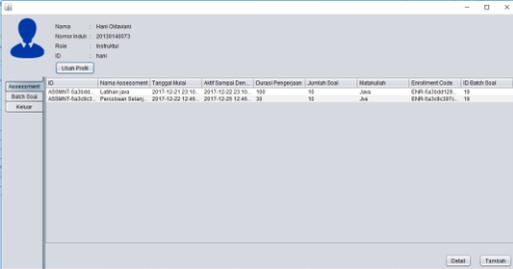
Gambar 4.36 diatas menunjukkan tampilan yang digunakan siswa untuk mengerjakan soal.

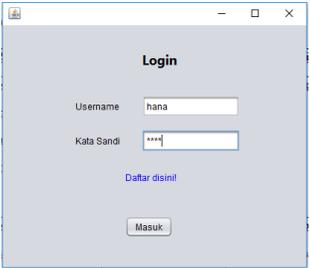
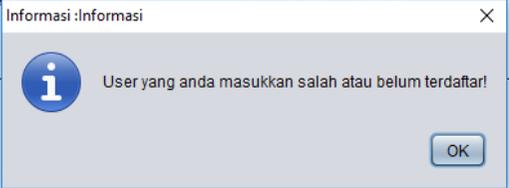
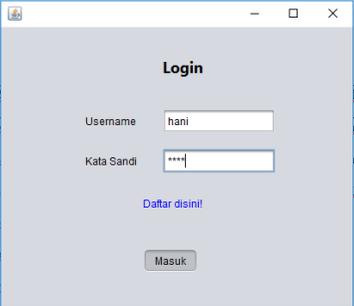
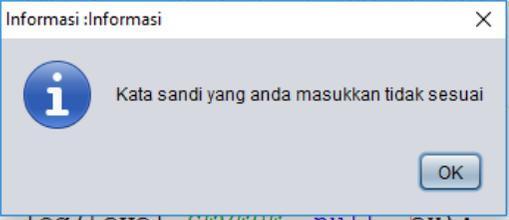
4.2 Pengujian Aplikasi

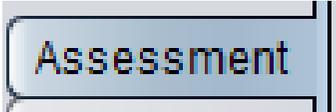
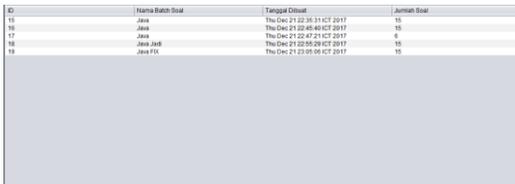
Metode pengujian yang dilakukan oleh peneliti adalah Metode Black Box Testing. Peneliti melakukan serangkaian pengujian terhadap aplikasi baik untuk siswa maupun instruktur dengan cara menguji setiap fungsi dan melihat kesesuaian fungsi dengan kebutuhan sistem. Apabila hasil sudah sesuai dengan kebutuhan, maka pengujian terhadap fungsi tersebut telah berhasil dan fungsi sudah memenuhi persyaratan sistem. Adapun tabel hasil pengujian telah disajikan pada tabel 4.1 dan 4.2 dibawah ini.

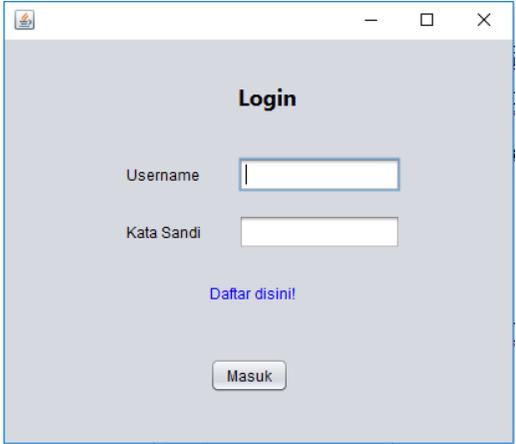
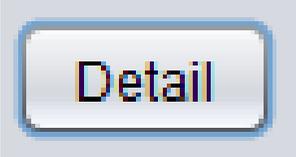
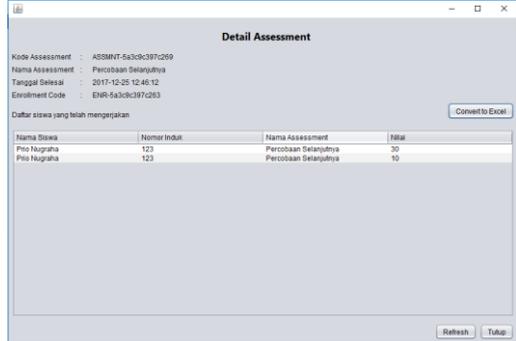
A. Aplikasi *Desktop* Instruktur

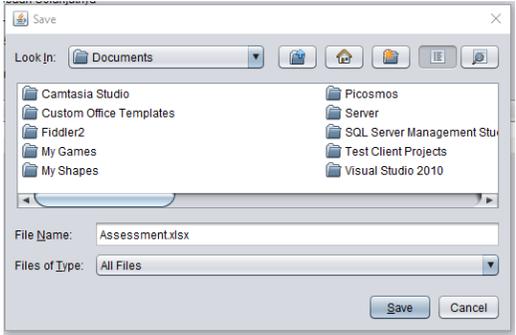
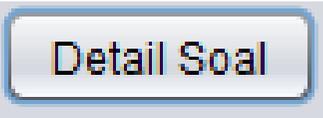
Tabel 4. 1 Pengujian Aplikasi *Desktop* Instruktur

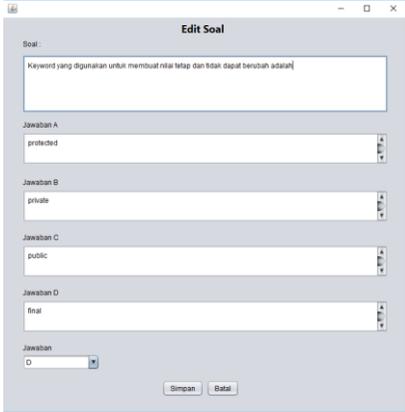
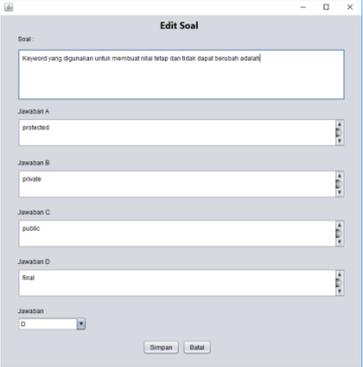
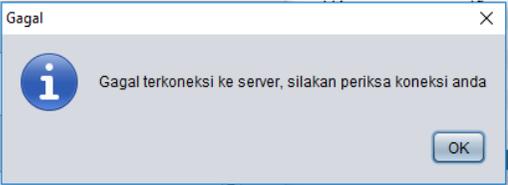
No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
	Masuk ke <i>form</i> utama melalui proses autentikasi dan otorisasi dengan <i>ID</i> dan Kata Sandi yang valid.		Sistem menampilkan <i>form</i> utama.		Gagal
1	Masuk ke <i>form</i> utama melalui proses autentikasi dan otorisasi dengan <i>ID</i> dan Kata Sandi yang valid.		Sistem menampilkan <i>form</i> utama.		Berhasil

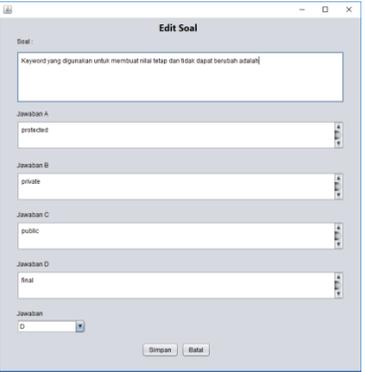
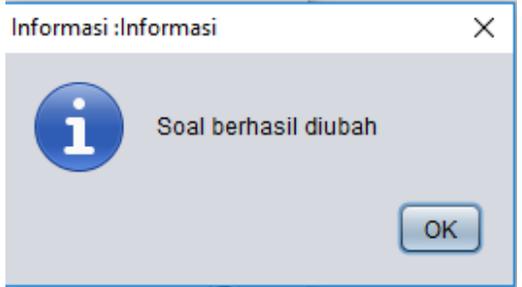
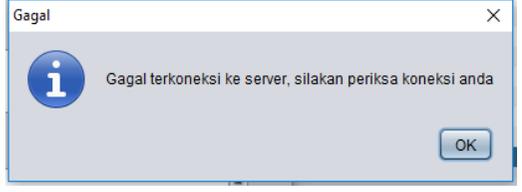
No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
2	Menampilkan pesan gagal jika <i>ID</i> yang dimasukkan ketika proses autentikasi dan otorisasi tidak sesuai.		Sistem menampilkan pesan “User yang anda masukkan salah atau belum terdaftar”.		Berhasil
3	Menampilkan pesan gagal jika kata sandi yang dimasukkan ketika proses autentikasi dan otorisasi tidak sesuai.		Sistem menampilkan pesan “Kata sandi yang anda masukkan tidak sesuai”		Berhasil

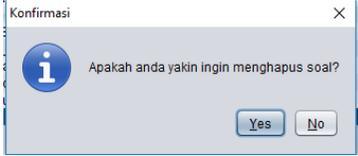
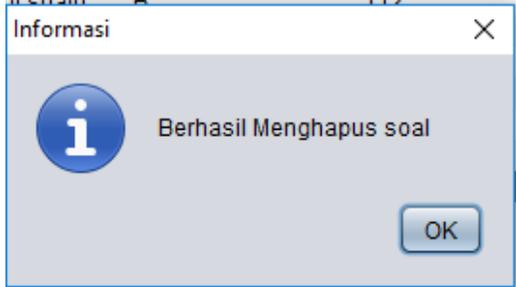
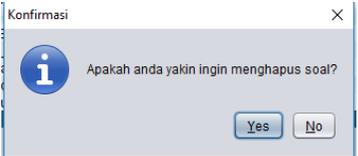
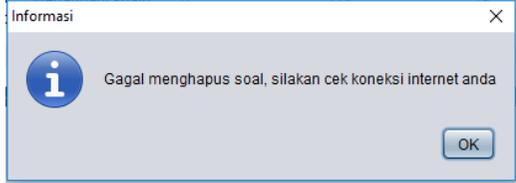
No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
4	Menampilkan tabel <i>assessment</i> ketika memilih <i>tab assessment</i> pada <i>form</i> utama.		Sistem menampilkan tabel <i>assessment</i> .		Berhasil
5	Menampilkan tabel <i>batch</i> soal ketika memilih <i>tab batch</i> soal pada <i>form</i> utama.		Sistem menampilkan tabel <i>batch</i> soal.		Berhasil

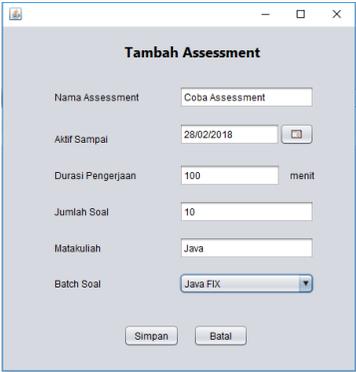
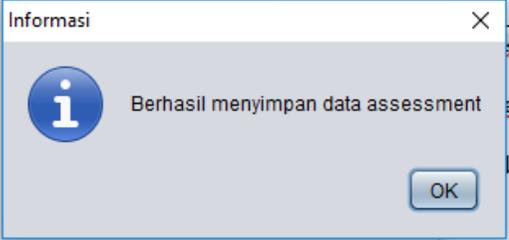
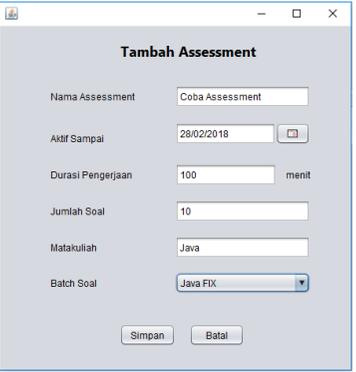
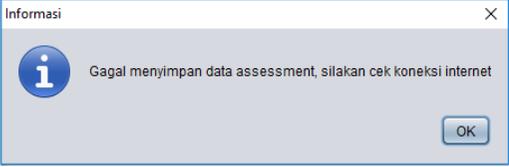
No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan												
6	Kembali ke <i>form login</i> ketika menekan <i>tab keluar</i> pada <i>form</i> utama.		Sistem menutup <i>form</i> utama dan menampilkan <i>form login</i> .		Berhasil												
7	Menampilkan <i>detail assessment</i> .		Sistem menampilkan <i>detail assessment</i> .	 <table border="1" data-bbox="1335 1075 1827 1114"> <thead> <tr> <th>Nama Siswa</th> <th>Nomor Induk</th> <th>Nama Assessment</th> <th>Nilai</th> </tr> </thead> <tbody> <tr> <td>Pilo Nugraha</td> <td>123</td> <td>Perobaan Selanjutnya</td> <td>30</td> </tr> <tr> <td>Pilo Nugraha</td> <td>123</td> <td>Perobaan Selanjutnya</td> <td>10</td> </tr> </tbody> </table>	Nama Siswa	Nomor Induk	Nama Assessment	Nilai	Pilo Nugraha	123	Perobaan Selanjutnya	30	Pilo Nugraha	123	Perobaan Selanjutnya	10	Berhasil
Nama Siswa	Nomor Induk	Nama Assessment	Nilai														
Pilo Nugraha	123	Perobaan Selanjutnya	30														
Pilo Nugraha	123	Perobaan Selanjutnya	10														

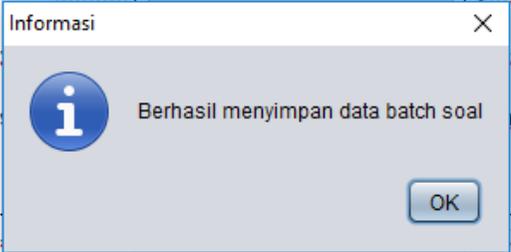
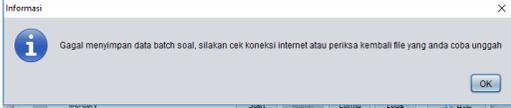
No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
8	Konversi tabel nilai ke dalam format .xlsx		Sistem menyimpan tabel nilai ke dalam format <i>file</i> .xlsx		Berhasil
9	Menampilkan <i>detail batch</i> soal.		Sistem menampilkan <i>detail batch</i> soal		Berhasil

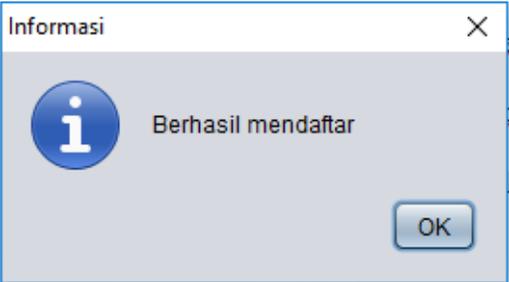
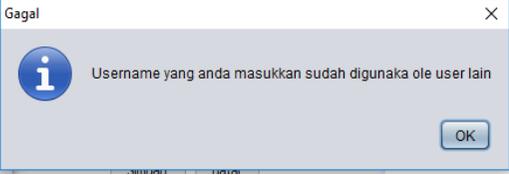
No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
10	Menampilkan <i>detail</i> soal ketika menekan tombol ubah soal.		Sistem menampilkan <i>detail</i> soal.		Berhasil
	Menampilkan pesan sukses jika berhasil melakukan perubahan pada soal.		Sistem menampilkan pesan “Soal berhasil diubah”.		Gagal

No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
11	Menampilkan pesan sukses jika berhasil melakukan perubahan pada soal.		Sistem menampilkan pesan “Soal berhasil diubah”.		Berhasil
12	Menampilkan pesan gagal ketika terjadi gangguan koneksi ke server ketika mengubah soal.		Sistem menampilkan pesan “Gagal terkoneksi ke server, silakan periksa koneksi anda”.		Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
13	Menampilkan pesan sukses ketika berhasil menghapus soal.		Sistem menampilkan pesan “Berhasil menghapus soal”.		Berhasil
14	Menampilkan pesan gagal ketika tidak berhasil menghapus soal.		Sistem menampilkan pesan “Gagal menghapus soal, silakan cek koneksi internet anda”.		Berhasil

No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
15	Menampilkan pesan sukses ketika berhasil menyimpan <i>assessment</i> .		Sistem menampilkan pesan “Berhasil menyimpan data <i>assessment</i> ”.		Berhasil
16	Menampilkan pesan gagal menyimpan <i>assessment</i> ketika terdapat gangguan koneksi internet.		Sistem menampilkan pesan “Gagal menyimpan data <i>assessment</i> , silakan cek koneksi internet”.		Berhasil

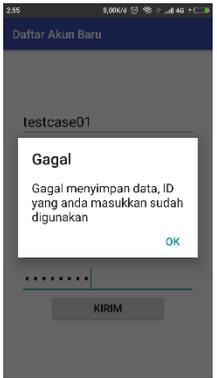
No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
17	Menampilkan pesan sukses ketika berhasil menyimpan data <i>batch</i> soal.		Sistem menampilkan pesan “Berhasil menyimpan data <i>batch</i> soal.		Berhasil
18	Menampilkan pesan gagal ketika tidak berhasil menyimpan <i>batch</i> soal.		Sistem menampilkan pesan “Gagal menyimpan data <i>batch</i> soal, silakan cek koneksi internet atau periksa kembali file yang anda coba unggah”.		Berhasil

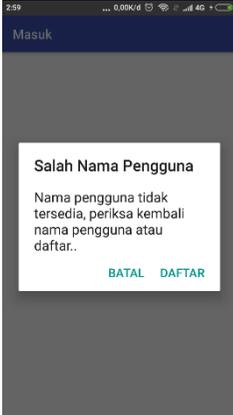
No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
19	Menampilkan pesan sukses ketika berhasil mendaftar.		Sistem menampilkan pesan “Berhasil mendaftar”.		Berhasil
20	Menampilkan pesan gagal ketika tidak berhasil mendaftar menggunakan <i>username</i> yang sudah digunakan <i>user</i> lain.		Sistem menampilkan pesan “Username yang anda masukkan sudah digunakan oleh user lain”.		Berhasil

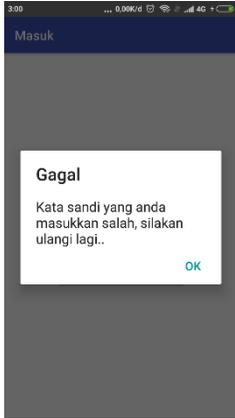
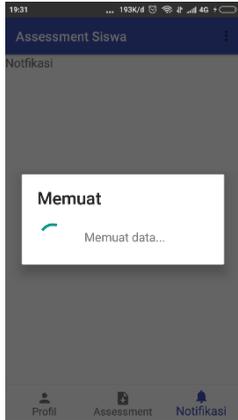
B. Aplikasi *Mobile* Siswa

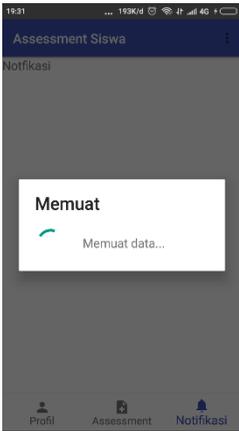
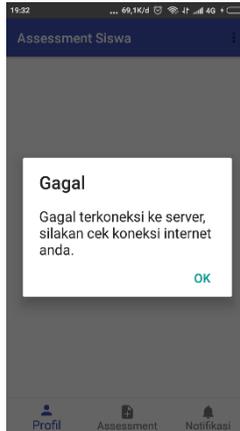
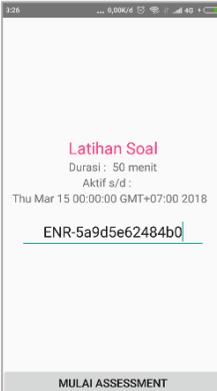
Tabel 4. 2 Pengujian Aplikasi *Mobile* Siswa

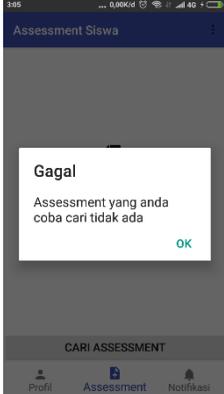
No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Menampilkan pesan sukses ketika berhasil mendaftar.		Sistem menampilkan pesan “Data telah berhasil tersimpan”.		Gagal
2	Menampilkan pesan sukses ketika berhasil mendaftar.		Sistem menampilkan pesan “Data telah berhasil tersimpan”.		Berhasil

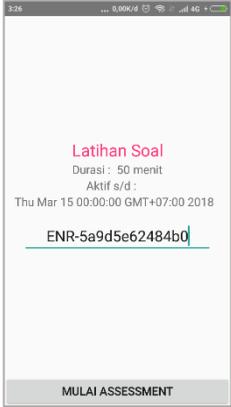
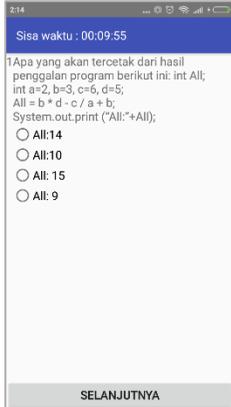
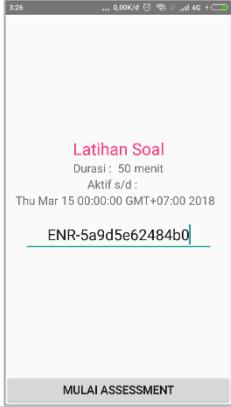
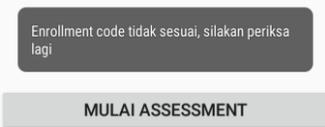
No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
3	Menampilkan pesan gagal ketika tidak berhasil mendaftar karena tidak terkoneksi ke server.		Sistem menampilkan pesan “Gagal terkoneksi ke server, silakan cek koneksi internet anda”.		Berhasil
4	Menampilkan pesan gagal ketika mendaftar dengan ID yang sudah digunakan		Sistem menampilkan pesan “Gagal menyimpan data, ID yang anda masukkan sudah digunakan”.		Berhasil

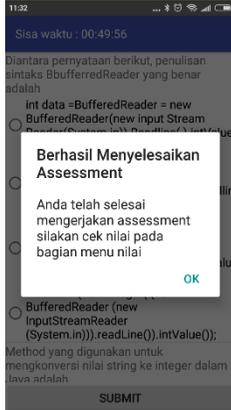
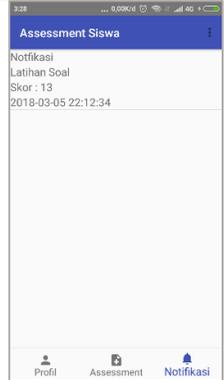
No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
5	Masuk ke <i>activity</i> utama melalui <i>login</i> .		Sistem menampilkan <i>activity</i> utama.		Berhasil
6	Menampilkan pesan gagal ketika tidak berhasil melakukan <i>login</i> karena <i>ID</i> tidak sesuai.		Sistem menampilkan pesan “Nama pengguna tidak tersedia, periksa kembali nama pengguna atau daftar”.		Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
7	Menampilkan pesan gagal ketika tidak berhasil melakukan <i>login</i> karena kata sandi tidak sesuai		Sistem menampilkan pesan “Gagal masuk, kata sandi yang anda masukkan salah, silakan ulangi lagi”.		Berhasil
8	Menampilkan data profil <i>user</i> ketika masuk ke <i>activity</i> utama.		Sistem menampilkan data profil <i>user</i> ketika masuk ke <i>activity</i> utama.		Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
9	Menampilkan pesan gagal ketika tidak berhasil memuat data karena gangguan koneksi.		Sistem menampilkan pesan “Gagal memuat data, silakan periksa koneksi internet anda dan coba ulangi lagi”.		Berhasil
10	Masuk ke <i>activity</i> cek <i>enrollment</i> melalui pencarian <i>assessment</i> .		Sistem menampilkan <i>activity</i> cek <i>enrollment</i> .		Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
11	Menampilkan pesan gagal melakukan pencarian <i>assessment</i> ketika <i>ID assessment</i> tidak sesuai.		Sistem menampilkan pesan “Assessment yang anda coba cari tidak ada”.		Berhasil
12	Menampilkan pesan gagal ketika melakukan pencarian <i>assessment</i> yang pernah dikerjakan		Sistem menampilkan pesan “Anda sudah pernah mengikuti assessment ini”.		Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
13	Masuk ke <i>activity</i> kerjakan soal melalui cek <i>enrollment</i> .		Sistem menampilkan <i>activity</i> kerjakan soal.		Berhasil
14	Menampilkan pesan gagal ketika kode <i>enrollment</i> tidak sesuai.		Sistem menampilkan pesan “Kode <i>enrollment</i> tidak sesuai, silakan periksa lagi”.		Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
15	Melakukan submisi setelah melakukan pengerjaan soal.	<input type="radio"/> private <input type="radio"/> public SUBMIT	Sistem menampilkan pesan “Anda telah mengerjakan assessment silakan cek nilai pada bagian menu nilai”.		Berhasil
16	Menampilkan <i>list</i> nilai dari <i>assessment</i> yang sudah dikerjakan.		Sistem menampilkan <i>list</i> nilai dari <i>assessment</i> yang sudah dikerjakan.		Berhasil

No	Skenario Pengujian	<i>Test Case</i>	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
17	Menampilkan pesan gagal ketika memuat data nilai karena gangguan koneksi.		Sistem menampilkan pesan “Gagal memuat data nilai, silakan cek koneksi internet”		Berhasil

4.3 Hasil Penelitian

Hasil dari penelitian direpresentasikan dalam bentuk data yang selanjutnya akan diolah menjadi sebuah informasi. Data tersebut diperoleh dari kuisioner yang dibagikan kepada responden. Responden yang terlibat dalam pengambilan data sebanyak 32 orang yang merupakan siswa SMA kelas XI yang sudah menguji secara langsung aplikasi Assessment untuk siswa.

Hasil dari penelitian ditentukan berdasarkan kelayakan aplikasi yang ditentukan menggunakan skala Likert sebagai alat ukur. Penilaian tersebut direpresentasikan dengan skor sangat setuju = 5, setuju = 4, cukup setuju = 3, tidak setuju = 2, sangat tidak setuju = 1. Berdasarkan aturan tersebut maka hasil dari uji kelayakan dari aplikasi ini adalah sebagai berikut:

Perhitungan jumlah skor dari data pengguna yang telah mengisi kuisioner aplikasi Hidden Kuliner adalah sebagai berikut:

Persamaan 1:

$$\text{Skor} = T \times P_n$$

T = Total jumlah responden yang memilih

P_n = Pilihan angka skor Likert

Persamaan 2:

$$\text{Index Kelayakan} = \text{Total Skor} / Y \times 100$$

Y = skor tertinggi likert x jumlah responden

Contoh:

Hasil perhitungan uji kelayakan pada pertanyaan kuisioner nomor satu adalah sebagai berikut:

- Responden yang menjawab Sangat Setuju (5) = $13 \times 5 = 70$
- Responden yang menjawab Setuju (4) = $15 \times 4 = 60$
- Responden yang menjawab Cukup Setuju (3) = $3 \times 3 = 15$
- Responden yang menjawab Tidak Setuju (2) = $1 \times 2 = 2$
- Responden yang menjawab Sangat Tidak Setuju (1) = $0 \times 1 = 0$

Total Skor = 136

Indeks Kelayakan = $136 / 160 \times 100$

$$= 0,85 \times 100$$

$$= 85$$

Jadi hasil yang didapat dari perhitungan di atas kelayakan aplikasi mendapat nilai sebesar 85% dan dapat dikategorikan “sangat layak” untuk pernyataan nomor satu pada kuisioner. Untuk hasil uji kelayakan secara keseluruhan dapat dilihat pada tabel 4.3.

Keterangan:

SS : Sangat Setuju

TS : Tidak Setuju

S : Setuju

STS : Sangat Tidak Setuju

CS : Cukup Setuju

Tabel 4. 3 Data Hasil Uji Kelayakan Aplikasi *Mobile*

No	Pernyataan	Jawaban					Kelayakan (%)	Kategori
		SS	S	CS	TS	STS		
		5	4	3	2	1		
1	Aplikasi <i>assessment</i> membantu siswa mempermudah mengerjakan penilaian dari guru.	13	15	3	1	0	85	Sangat Layak
2	Tampilan soal <i>assessment</i> dapat dilihat dengan jelas oleh siswa.	10	16	3	2	1	80	Layak
3	Siswa dapat melihat dengan jelas informasi nilai yang ditampilkan.	12	11	5	2	2	78	Layak
4	Siswa dapat mengerjakan soal dimana saja dan kapan saja.	9	14	6	2	1	78	Layak

No	Pernyataan	Jawaban					Kelayakan (%)	Kategori
		SS	S	CS	TS	STS		
5	Aplikasi <i>assessment</i> mudah dioperasikan oleh siswa.	10	11	8	2	1	77	Layak

4.4 Pembahasan

Dari hasil pengujian yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi *assessment* ini sudah dapat dioperasikan dengan cukup maksimal. Hal tersebut dapat dilihat dari hasil pengujian fungsi yang dilakukan sebelumnya. Setiap fungsi pada aplikasi sudah berjalan sesuai dengan kebutuhan. Untuk kelayakan aplikasi dapat disimpulkan bahwa aplikasi sudah memenuhi kategori layak. Hal tersebut disimpulkan berdasarkan data yang sudah diambil dari kuisioner dengan perhitungan rata-rata kelayakan sebesar 80%.

Dari hasil pemaparan di atas tidak menutup kemungkinan bahwa aplikasi ini akan menemui kendala dalam hal pengoperasiannya di masa yang akan datang. Oleh karena itu sangat diperlukan untuk mengoptimalkan kembali serta melakukan *maintenance* pada aplikasi ini.