

## BAB IV

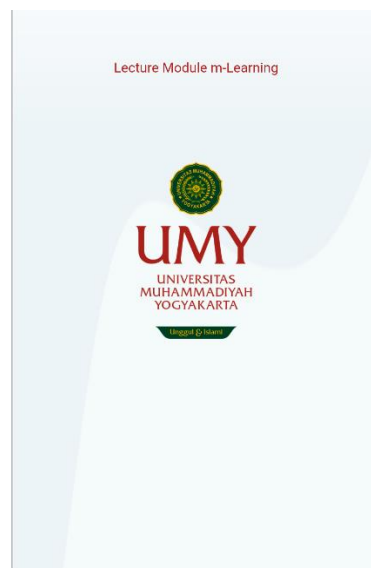
### HASIL DAN PEMBAHASAN

#### 4.1 Hasil

Hasil dari penelitian ini adalah modul aplikasi untuk pembelajaran pemrograman bahasa java yang berisi paket SCORM dan berbasis android. Modul aplikasi itu berupa modul dosen dan modul mahasiswa. Modul dosen berisi nilai seluruh mahasiswa, sedangkan modul mahasiswa berisi modul materi, quis dan niali.

##### 4.1.1 Modul Dosen

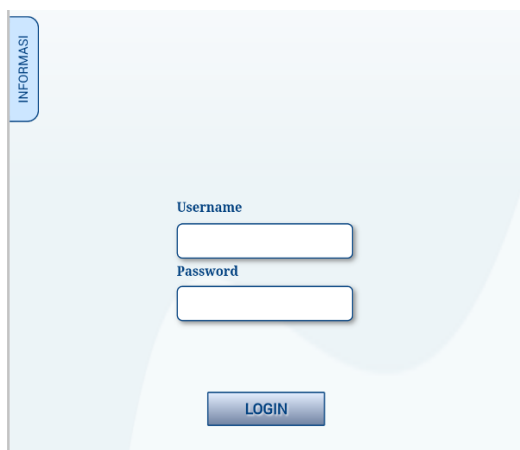
###### A. Halaman *Splash Screen*



**Gambar 4. 1** Tampilan Halaman *Splash Screen* Modul Dosen

Pada gambar 4.1 merupakan halaman yang akan muncul pertama kali saat aplikasi modul dosen dibuka. Pada proses ini user harus menekan layar agar dapat menuju halaman berikutnya.

## B. Halaman *Login* Modul Dosen



INFORMASI

Username

Password

LOGIN

**Gambar 4. 2** Tampilan Halaman *Login* Modul Dosen

Pada gambar 4.2 merupakan halaman login modul dosen untuk melakukan validasi *username* dan *password*. Pada halaman ini juga terdapat label informasi tentang modul, tatacara pemakaian aplikasi dan alamat/kontak pengaduan apabila terjadi kesalahan pada isi aplikasi, seperti yang di tampilkan pada gambar 4.3.

## C. Label Informasi Modul Dosen



INFORMASI

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Modul monitoring ini khusus untuk dosen. Ada tiga langkah untuk menggunakan modul ini:

- **Masukkan** kode kelas dan nama tugas
- **Amati** data, temukan masalah
- Bila perlu, **komunikasi** dengan mahasiswa melalui WhatsApp

Apabila ada kesalahan isi mohon berkenan memberitahu kami.

Teknik Informatika UMY  
Gedung F4 Lanti dasar Kampus Terpadu  
Jl. Brawijaya Tamantirto Bantul DI Yogyakarta 55183  
Telp: 0274-387656 Ext: 381

**Gambar 4. 3** Tampilan Label Informasi Modul Dosen

Pada gambar 4.3 merupakan label informasi untuk mempelajari penggunaan modul dosen dan informasi pengaduan aplikasi.

## D. Halaman Lihat Nilai Modul Dosen

No	Nim	WhatsApp	Aktivitas	Score
1	20180140105	<a href="https://wa.me/081380128515">081380128515</a>	Class	90
2	20180140098	<a href="https://wa.me/087886883665">087886883665</a>	Class	90
3	20180140087	<a href="https://wa.me/089665851565">089665851565</a>	Class	90
4	20180140069	<a href="https://wa.me/081258348963">081258348963</a>	Class	90
5	20180140082	<a href="https://wa.me/082139527186">082139527186</a>	Class	90
6	20180140065	<a href="https://wa.me/081347509154">081347509154</a>	Class	90
7	20180140123	<a href="https://wa.me/082198874753">082198874753</a>	Class	80
8	20180140091	<a href="https://wa.me/082158565173">082158565173</a>	Class	80
9	20180140055	<a href="https://wa.me/082143578882">082143578882</a>	Class	80
10	20180140052	<a href="https://wa.me/082242112183">082242112183</a>	Class	80
11	20180140008	<a href="https://wa.me/082133922316">082133922316</a>	Class	80
12	20180140029	<a href="https://wa.me/082277215266">082277215266</a>	Class	80
13	20180140096	<a href="https://wa.me/085357929545">085357929545</a>	Class	70
14	20180140104	<a href="https://wa.me/081326711418">081326711418</a>	Class	70
15	20180140084	<a href="https://wa.me/085267664015">085267664015</a>	Class	70
16	20180140054	<a href="https://wa.me/081326855427">081326855427</a>	Class	70
17	20180140030	<a href="https://wa.me/081914578684">081914578684</a>	Class	70
18	20180140042	<a href="https://wa.me/085228664552">085228664552</a>	Class	70
19	20180140116	<a href="https://wa.me/081335415422">081335415422</a>	Class	60
20	20180140101	<a href="https://wa.me/081558857762">081558857762</a>	Class	60
21	20180140090	<a href="https://wa.me/085959455704">085959455704</a>	Class	60
22	20180140115	<a href="https://wa.me/082227013191">082227013191</a>	Class	50
23	20180140114	<a href="https://wa.me/081242171125">081242171125</a>	Class	100
24	20180140079	<a href="https://wa.me/085210332132">085210332132</a>	Class	100
25	20180140093	<a href="https://wa.me/082289803751">082289803751</a>	Class	100
26	20180140037	<a href="https://wa.me/087888451559">087888451559</a>	Class	100
27	20180140067	<a href="https://wa.me/082353133841">082353133841</a>	Class	100
28	20180140113	<a href="https://wa.me/085365918702">085365918702</a>	Class	100
29	20180140099	<a href="https://wa.me/081350505151">081350505151</a>	Class	0
30	20180140094	<a href="https://wa.me/082269311563">082269311563</a>	Class	0

**Gambar 4. 4** Tampilan Halaman Lihat Nilai Modul Dosen

Pada gambar 4.4 diatas menunjukkan halaman lihat nilai ini dibuat supaya dosen dapat mengetahui nilai masing-masing mahasiswanya. Pada halaman ini terdapat kolom kode matakuliah, tugas dan tombol cari untuk mencari nilai mahasiswa sesuai kode matakuliah dan tugas yang diberikan.

### 4.1.2 Modul Mahasiswa

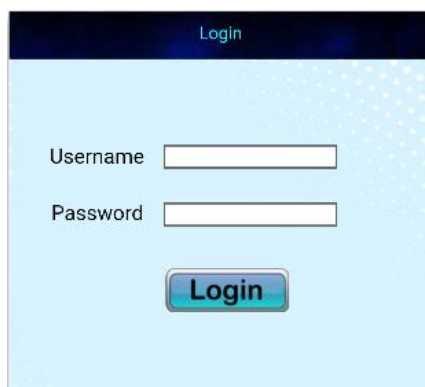
#### A. Halaman *Splash Screen*



**Gambar 4. 5** Tampilan Halaman Splash Screen Modul Mahasiswa

Pada gambar 4.5 diatas merupakan halaman yang akan muncul pertama kali saat aplikasi modul mahasiswa dibuka. Pada proses ini user harus menekan layar agar dapat menuju halaman berikutnya.

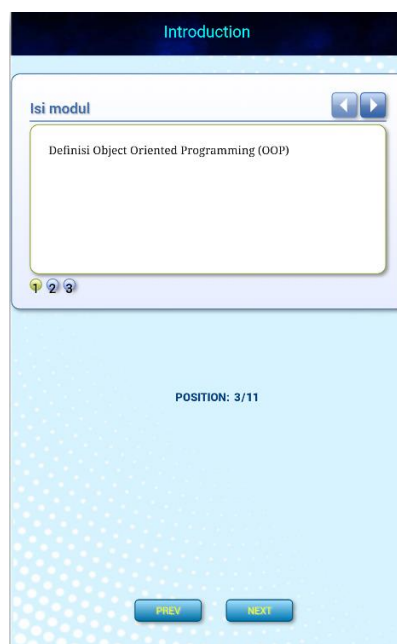
### B. Halaman *Login* Mahasiswa



**Gambar 4. 6** Tampilan Halaman *Login* Modul Mahasiswa

Pada gambar 4.6 merupakan halaman login modul mahasiswa untuk melakukan validasi *username* dan *password*.

### C. Halaman *Introduction*



**Gambar 4. 7** Tampilan Halaman *Introduction*

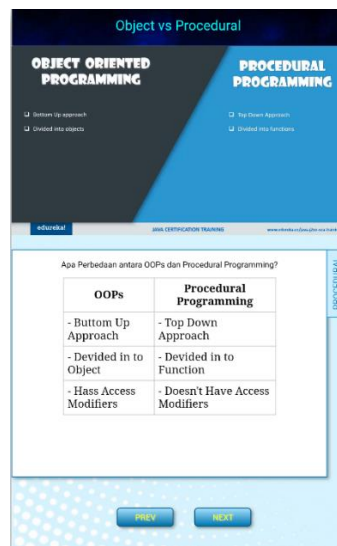
Pada gambar 4.7 merupakan halaman judul materi yang akan dipelajari pada halaman selanjutnya.

#### D. Halaman *Object Oriented Programming*



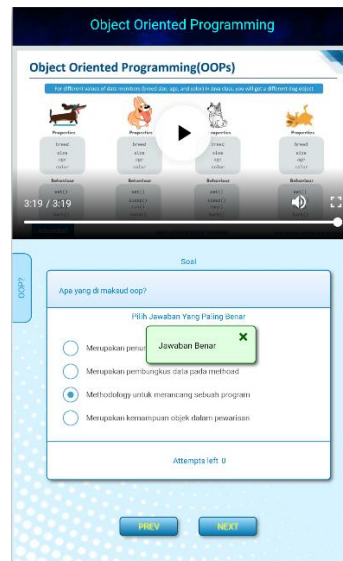
**Gambar 4. 8** Halaman OOP Tampilan Video Materi

Pada gambar 4.8 merupakan tampilan halaman oop untuk mempelajari materi melalui video. Setelah video selesai diputar maka *quiz* akan di tampilkan secara otomatis.



**Gambar 4. 9** Tampilan Label Ringkasan Materi Halaman OOP

Pada gambar 4.9 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu oop ditekan.



**Gambar 4. 10** Tampilan Label Quiz Halaman OOP

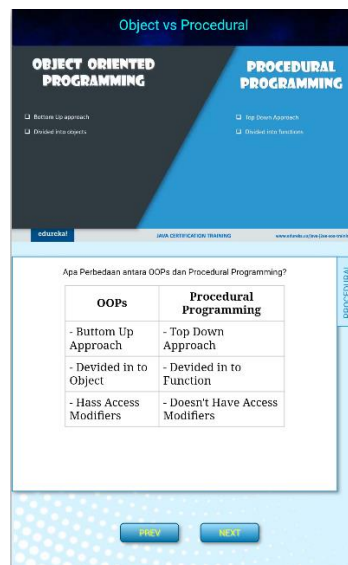
Pada gambar 4.10 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman oop. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

#### **E. Halaman OOP vs *Procedural***



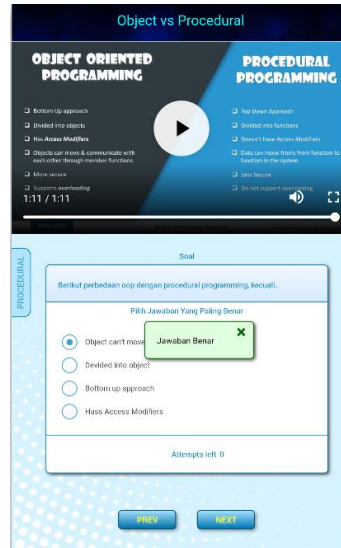
**Gambar 4. 11** Halaman Object vs Procedural Tampilan Video Materi

Pada gambar 4.11 merupakan tampilan halaman *object vs procedural* untuk mempelajari materi melalui video. Setelah video selesai diputar maka quiz akan di tampilkan secara otomatis.



**Gambar 4. 12** Tampilan Label Ringkasan Materi *Object vs Procedural*

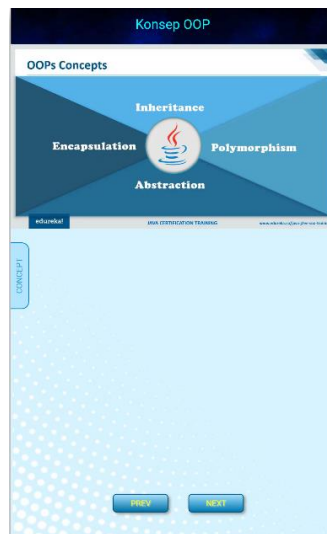
Pada gambar 4.12 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu *object vs procedural* ditekan.



**Gambar 4. 13** Tampilan Label Quiz Halaman *Object vs Procedural*

Pada gambar 4.13 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman *object vs procedural*. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

## F. Halaman Konsep OOP



**Gambar 4. 14** Halaman Konsep OOP Tampilan Video Materi

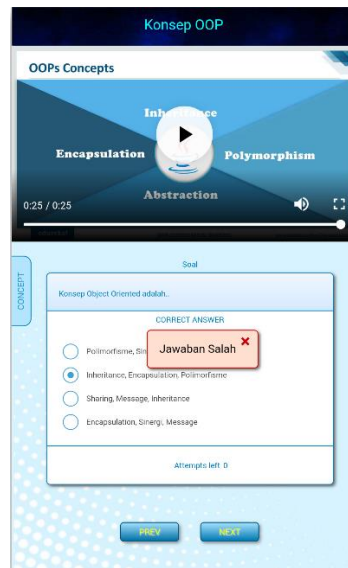
Pada gambar 4.14 merupakan tampilan halaman konsep oop untuk mempelajari materi melalui video. Setelah video selesai diputar maka quiz akan di tampilkan secara otomatis.



**Gambar 4. 15** Tampilan Label Ringkasan Materi Halaman Konsep OOP

Pada gambar 4.15 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu konsep oop ditekan.

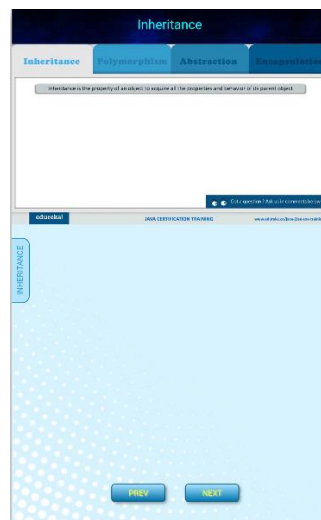




**Gambar 4. 16** Tampilan Label Quiz Halaman Konsep OOP

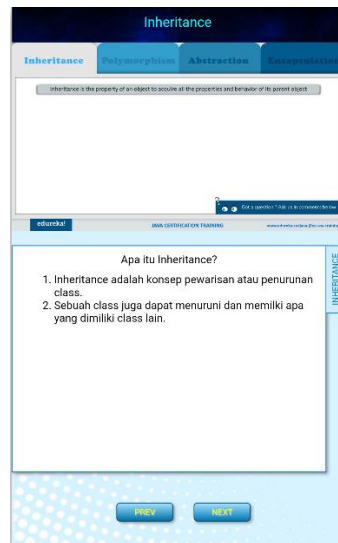
Pada gambar 4.16 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman konsep oop. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

### G. Halaman *Inheritance*



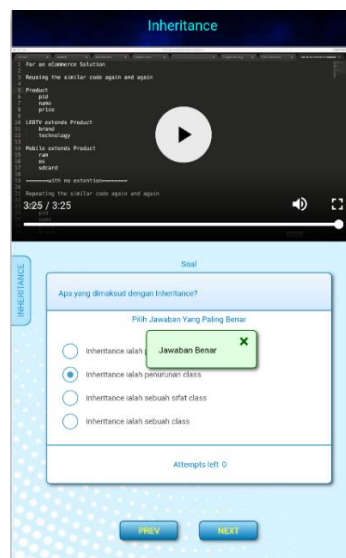
**Gambar 4. 17** Halaman *Inheritance* Tampilan Video Materi

Pada gambar 4.17 merupakan tampilan halaman *inheritance* untuk mempelajari materi melalui video. Setelah video selesai diputar maka quiz akan ditampilkan secara otomatis.



**Gambar 4. 18** Tampilan Label Ringkasan Materi Halaman *Inheritance*

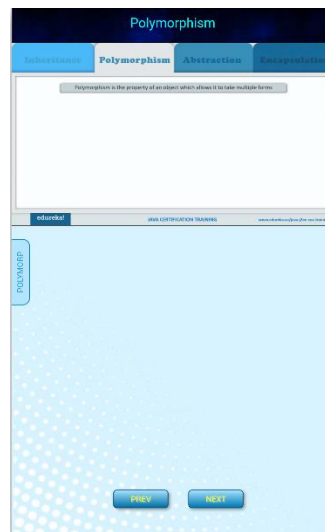
Pada gambar 4.18 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu *inheritance* ditekan.



**Gambar 4. 19** Tampilan Label Quiz Halaman *Inheritance*

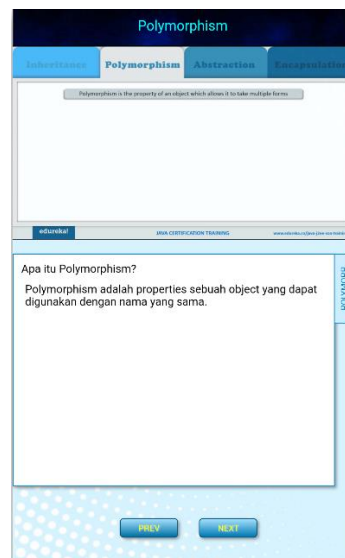
Pada gambar 4.19 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman *inheritance*. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

## H. Halaman *Polymorphism*



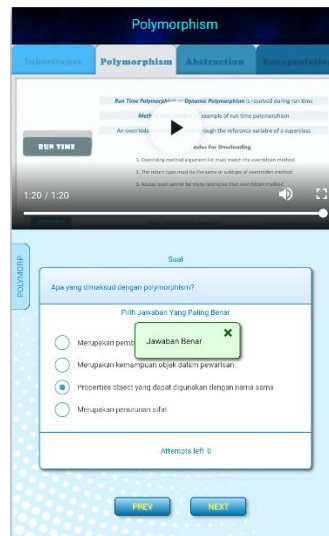
**Gambar 4. 20** Halaman *Polymorphism* Tampilan Video Materi

Pada gambar 4.20 merupakan tampilan halaman *polymorphism* untuk mempelajari materi melalui video. Setelah video selesai diputar maka quiz akan ditampilkan secara otomatis.



**Gambar 4. 21** Tampilan Label Ringkasan Materi Halaman *Polymorphism*

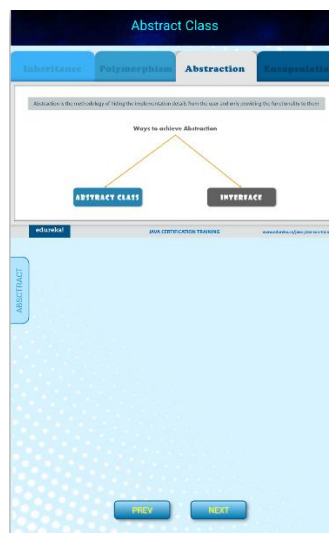
Pada gambar 4.21 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu *polymorphism* ditekan.



**Gambar 4. 22** Tampilan Label Quiz Halaman *Polymorphism*

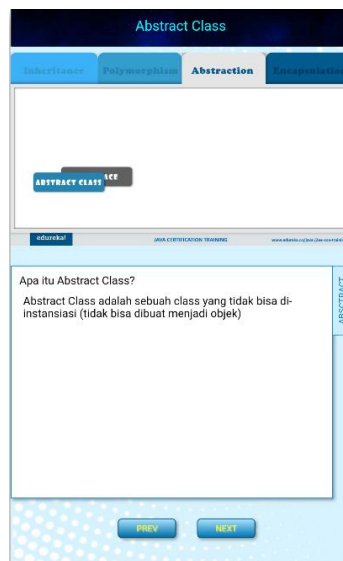
Pada gambar 4.22 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman *polymorphism*. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

### I. Halaman *Abstract Class*



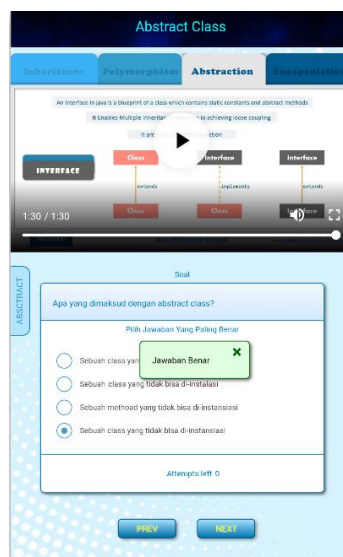
**Gambar 4. 23** Halaman *Abstract Class* Tampilan Video Materi

Pada gambar 4.23 merupakan tampilan halaman *abstract class* untuk mempelajari materi melalui video. Setelah video selesai diputar maka quiz akan ditampilkan secara otomatis.



**Gambar 4. 24** Tampilan Label Ringkasan Materi Halaman *Abstract Class*

Pada gambar 4.24 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu *abstract class* ditekan.



**Gambar 4. 25** Tampilan Label Quiz Halaman *Abstract Class*

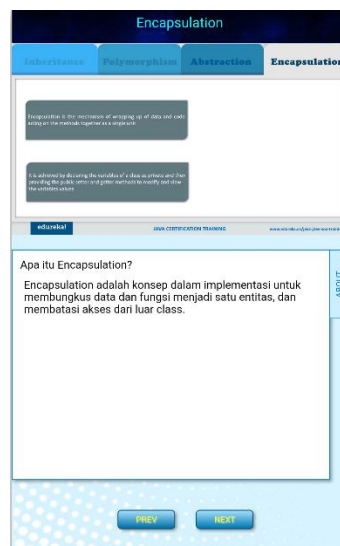
Pada gambar 4.25 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman *abstract class*. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

## J. Halaman *Encapsulation*



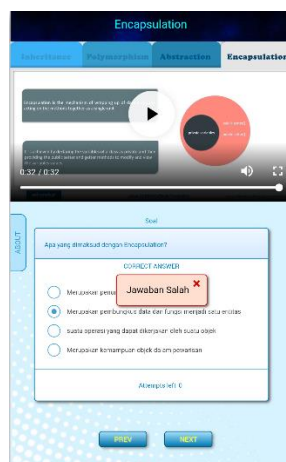
Gambar 4. 26 Halaman *Encapsulation* Tampilan Video Materi

Pada gambar 4.24 merupakan tampilan halaman *abstract class* untuk mempelajari materi melalui video. Setelah video selesai diputar maka quiz akan ditampilkan secara otomatis.



Gambar 4. 27 Tampilan Label Ringkasan Materi Halaman *Encapsulation*

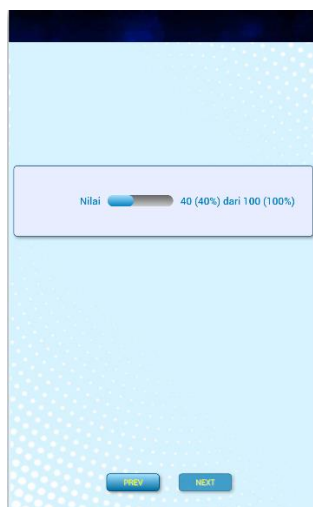
Pada gambar 4.27 merupakan tampilan yang berisi materi ringkasan pada video. Tampilan ini akan muncul apabila tombol menu *abstract class* ditekan.



**Gambar 4. 28** Tampilan Label Quiz Halaman *Encapsulation*

Pada gambar 4.28 merupakan tampilan yang digunakan mahasiswa untuk mengerjakan quiz pada halaman *abstract class*. Pada halaman ini juga terdapat *feedback* apakah jawaban benar atau salah.

#### K. Halaman Lihat Nilai



**Gambar 4. 29** Tampilan Halaman Lihat Nilai Modul Mahasiswa

Pada gambar 4.29 merupakan tampilan halaman yang digunakan untuk melihat nilai mahasiswa yang telah mengerjakan quiz.

#### 4.2 Pengujian Aplikasi

Pengujian aplikasi *mobile learning* pada penelitian ini menggunakan metode *black box testing* dan metode *real testing*. *Black box testing* merupakan metode yang dilakukan untuk mengetahui setiap fungsi berjalan sesuai sistem

yang telah ditentukan, sehingga penguji dapat mengetahui terjadinya *error* pada aplikasi. Sedangkan *real testing* merupakan metode pengujian yang dilakukan dengan cara mengambil data dari responden yang telah bersedia melakukan uji coba aplikasi dan mengisi kuisisioner yang telah diberikan.

#### 4.2.1 Hasil *Black Box Testing*

Pengujian *Black box testing* digunakan untuk mengetahui fungsi-fungsi dari aktifitas system yang terdapat pada tampilan antarmuka. Aktifitas yang diujikan adalah fungsi tombol di setiap tampilan pada aplikasi. Hasil pengujian dapat dilihat pada tabel 4.1.

**Tabel 4. 1** Hasil *Black Box Testing*

No	Kasus yang diuji	Hasil yang diharapkan	Hasil yang didapatkan	Status
1	Tombol “Login”	Memvalidasi data user keserver dan halaman <i>login</i> akan beralih ke halaman selanjutnya	Data user dapat divalidasi dan halaman dapat berpindah kehalaman selanjutnya	Berhasil
2	Tombol “Cari”	Mencari nilai mahasiswa sesuai kode matakuliah dan kode tugas.	Nilai mahasiswa dapat di cari sesuai dengan kode matakuliah dan kode tugas.	Berhasil
3	Tombol “Text”	Menampilkan materi pembelajaran	Materi pembelajaran dapat ditampilkan	Berhasil
No	Kasus yang diuji	Hasil yang diharapkan	Hasil yang didapatkan	Status
4	Tommbol “Submit Answer”	Jawaban akan di kunci, BAR nilai akan bertambah dan tombol <i>next</i> akan ditampilkan	Jawaban terkunci, BAR nilai telah bertambah dan tombol <i>next</i> ditampilkan	Berhasil



5	Tombol "Next"	Halaman akan beralih ke soal berikutnya	Halaman beralih ke soal berikutnya	Berhasil
	Tombol "Preview"	Halaman akan beralih kehalaman sebelumnya	Halaman beralih kehalaman sebelumnya	Berhasil
6	Tombol "Urutkan"	Nilai mahasiswa akan diurutkan sesuai dengan nilai yang paling besar ke nilai yang paling kecil	Nilai dapat diurutkan dari yang paling besar ke nilai yang paling kecil	Berhasil

#### 4.2.2 Hasil Real Testing

Semua data yang diperoleh dari kuesioner untuk melakukan penilaian aplikasi, selanjutnya akan diolah menjadi sebuah informasi. Responden yang dilibatkan dalam pengambilan data adalah sebanyak 31 orang yaitu mahasiswa 2018 Teknik Informatika Universitas Muhammadiyah Yogyakarta.

Metode skala likert yang digunakan untuk pengujian sistem aplikasi yang dilakukan kepada pengguna dengan penilaian sangat setuju = 5 poin, setuju = 4 poin, cukup setuju = 3 poin, tidak setuju = 2 poin, sangat tidak setuju = 1 poin. Hasil penilaian tersebut dapat dilihat pada tabel 4.2 berikut ini :

**Tabel 4. 2 Hasil Real Testing**

No	Pertanyaan	Jawaban				
		1	2	3	4	5
		STS	TS	CS	S	SS
1	Aplikasi ini mudah digunakan?		2	3	13	13
2	Isi video mudah dipahami?	1	5	11	9	5
3	Isi text mudah dipahami?		1	4	11	15

4	Aplikasi bisa digunakan setiap saat dimana saja dan kapan saja?		1	5	14	11
5	Setelah belajar, soal mudah dijawab?		1	5	13	12
6	Hasil belajar mudah diketahui?		1	5	13	12
7	Aplikasi bermanfaat untuk belajar pemrograman java?		1	4	13	13
8	Aplikasi nyaman digunakan?		1	5	15	10

Berikut adalah keterangan singkatan dari jawaban pada Tabel 4.2

SS = Sangat Setuju

TS = Tidak Setuju

S = Setuju

STS = Sangat Tidak Setuju

CS = Cukup Setuju

Jumlah total jawaban yang telah didapat pada seluruh pertanyaan sebagai berikut :

- Jumlah total jawaban Sangat Setuju = 91
- Jumlah total jawaban Setuju = 101
- Jumlah total jawaban Cukup Setuju = 42
- Jumlah total jawaban Tidak Setuju = 13
- Jumlah total jawaban Sangat Tidak Setuju = 1

Perhitungan jumlah total dari data pengguna aplikasi *Mobile Learning* Berbasis Android sebagai berikut:

Total = Jumlah Responden x Point Tertinggi x Jumlah Pertanyaan

Total = 31 x 5 x 8

Total = 1240

Sedangkan persentase kelayakan dari pengguna aplikasi *Mobile Learning* Berbasis Android sebagai berikut :

Perhitungan persentase dari kategori “**SETUJU**”

$$\text{Index} = (\text{Jumlah SS} \times 5) + (\text{Jumlah S} \times 4) : \text{Total} \times 100$$

$$\text{Index} = (91 \times 5) + (101 \times 4) : 1240 \times 100$$

$$\text{Index} = 455 + 404 : 1240 \times 100$$

$$\text{Index} = 859 : 1240 \times 100$$

$$\text{Index} = 0,6928 \times 100$$

$$\text{Index} = 69,28 \%$$

Perhitungan persentase dari kategori **“NETRAL”**

$$\text{Index} = (\text{Jumlah CS} \times 3) : \text{Total} \times 100$$

$$\text{Index} = (42 \times 3) : 1240 \times 100$$

$$\text{Index} = 126 : 1240 \times 100$$

$$\text{Index} = 0,1016 \times 100$$

$$\text{Index} = 10,16 \%$$

Perhitungan persentase dari kategori **“TIDAK SETUJU”**

$$\text{Index} = (\text{Jumlah TS} \times 2) + (\text{Jumlah STS} \times 1) : \text{Total} \times 100$$

$$\text{Index} = (13 \times 2) + (1 \times 1) : 1240 \times 100$$

$$\text{Index} = 26 + 1 : 1240 \times 100$$

$$\text{Index} = 27 : 1240 \times 100$$

$$\text{Index} = 0,0217 \times 100$$

$$\text{Index} = 2,17 \%$$

Jumlah total persentase pada semua kategori sebagai berikut :

Total Persentase = setuju + netral + tidak setuju

$$\text{Total Persentase} = 69,28 \% + 10,16 \% + 2,17 \%$$

Total Persentase = 81,61 %

Untuk menentukan kelayakan aplikasi dari data rekayasa pengguna perangkat lunak, diambil dari perhitungan persentase tertinggi. Sedangkan total point tertinggi persentase kelayakan yang didapat yaitu 69,28%. Berdasarkan kriteria pada Bab 3, Tabel 3.6, total poin tersebut termasuk dalam kategori “**Layak**”.

### 4.3 Pembahasan

#### 4.3.1 Pembahasan Fungsi *JavaScript*

##### A. Proses Validasi

Fungsi *JavaScript* Untuk membuat file data mahasiswa :

```
var moduleID = "051007DP19";
```

```
var thisChapter = "Class";
```

```
var text_assign = '';
```

```
function read_file()
{
    try
    {
        return localStorage.getItem(moduleID);
    }
    catch (err)
    {
        return 'Course ID is not found';
    }
}
```

Untuk memeriksa data mahasiswa sudah tersimpan pada *server* atau belum maka diperlukan kode *JavaScript* diatas untuk melakukan pengecekan data mahasiswa. Jika tidak ditemukan maka dilakukan pembuatan data baru. Fungsi *JavaScript* untuk membuat data baru yaitu `create_newFile()`.

```
function create_newFile(moduleID,studentId,studentName,status)
{
    var score = g_encode(score);
    var string = generate_array(100);
    array = string.split(',');
    array[0]=moduleID;array[1]=studentId;array[2]=studentName;array[3]=status;
    var Text = array_toText(array,100);
    try
    {
        localStorage.setItem(moduleID,Text);
        return "OK";
    }
    catch (err)
    {
        alert('File saved error');
    }
}
```

Fungsi javascript diatas digunakan untuk membuat data baru pada *server*. File akan diuraikan menjadi *array* menggunakan fungsi *split*, diambil nama modul (*moduleID*), nomor mahasiswanya (*studentId*), nama mahasiswa (*studentName*) dan status validasi (*status*). Apabila data berhasil disimpan maka muncul jendela konfirmasi “OK”, jika tidak berhasil maka muncul jendela “File saved error”. Kemudian data tersebut dicocokkan dengan data yang telah di simpan pada *Moodle Server*. Jika terdaftar di server berarti status diubah menjadi “VALID”. Data yang tersimpan di *server* “051007DP19” diperbarui sehingga mahasiswa tersebut dinyatakan memiliki status “VALID”. Fungsi validasi yang di gunakan :

```

function passValidation(url,json,nim){
    CLO['Loading'].Show({ bMaster: true });
    $(document).ready(function(){
        $.ajax({
            crossDomain: true,
            url: url,
            method: 'POST',
            dataType: 'text',
            data: json,
            success: function (response) {
                if (response == "VALID"){
                    updateFile();
                    synchronize(nim);
                    CL.Navigation.GoTo({ sTargetType: "slide", sTargetDir: "next" });
                    CLO['Loading'].Hide({ bMaster: true });
                }
            }
            if (response == "INVALID"){
                CLO['Loading'].Hide({ bMaster: true });
                alert("Username atau Password tidak sah");
            }
        },
        error: function (error) {
            alert("Jaringan error, code: " + error.status);
        }
    });
}

```

Fungsi *JavaScript* diatas digunakan untuk cek *Username* dan *Password* pada saat mahasiswa melakukan login aplikasi. Apabila proses validasi sukses maka data mahasiswa akan diupdate dan halaman akan dialihkan ke halaman berikutnya. Jika proses validasi gagal maka muncul notifikasi “Username atau password tidak sah”. Apabila proses validasi gagal dikarenakan tidak terhubung ke internet maka muncul notifikasi “Jaringan error”.

## **B. Fungsi Simpan Nilai**

Apabila mahasiswa berhasil melakukan validasi saat login maka fungsi yang digunakan selanjutnya yaitu `updateScore()`.

```

function updateScore(){
    var currentScore=findScore(thisChapter);
    var Q1 = CL.SCO.GetObjectiveScore({ sId: "Q1" });
    var Q2 = CL.SCO.GetObjectiveScore({ sId: "Q2" });
    var Q3 = CL.SCO.GetObjectiveScore({ sId: "Q3" });
    var Q4 = CL.SCO.GetObjectiveScore({ sId: "Q4" });
    var Q5 = CL.SCO.GetObjectiveScore({ sId: "Q5" });
    var Q6 = CL.SCO.GetObjectiveScore({ sId: "Q6" });
    var Q7 = CL.SCO.GetObjectiveScore({ sId: "Q7" });
    var score = Q1+Q2+Q3+Q4+Q5+Q6+Q7;
    if (score<currentScore){score=currentScore;}
        CL.SCO.SetObjectiveScore({ sId: "total", sSrcId: "", bReplace: true,
nScore: score });
    store_data(thisChapter, score);
}

```

Fungsi *JavaScript* di atas digunakan untuk memperbaharui dan menyimpan nilai mahasiswa pada *server*. Sebelum nilai disimpan fungsi akan mencari terlebih dahulu nilai yang sudah ada, setelah nilai ditemukan maka fungsi akan mengambil jawaban dari setiap pertanyaan. Selanjutnya nilai akan dijumlah sesuai jawaban benar pertanyaan 1 sampai 7. Setelah selesai maka data akan divaliasi terlebih dahulu menggunakan yaitu `store_data()`.

```

function store_data(chapter, text)
{
    var textfile = read_file();
    array = textfile.split(',');
    var counter = 0;
    for(i=4;i<array.length;i++)
    {
        if ((array[i]=='*')(array[i]==chapter))
        {
            counter = i;
            break;
        }
    }
    var Textfilebaru = "";
    array[counter] = chapter;
    array[counter+1] = text;
    for(i=0;i<=array.length;i++){
        if (i==array.length){
            Textfilebaru += array[i];
            break;
        }
        Textfilebaru += array[i] + ',';
    }
    try {
        localStorage.setItem(moduleID,Textfilebaru);
    }
    catch (err) {
        alert('File saved error');
    }
}

```

Fungsi diatas digunakan untuk menyimpan nilai mahasiswa yang telah mengerjakan quiz pada modul.

### C. Fungsi Sinkronisasi

Sinkronisasi digunakan supaya data mahasiswa tidak bertumpuk saat melakukan login menggunakan perangkat yang berbeda. Fungsi sinkronisasi yang digunakan sebagai berikut :

```

function updateBothSide(){

    var sArray = localStorage.getItem("server").split(',');

    //alert("s: "+localStorage.getItem("server"));

    var lArray = localStorage.getItem("localm").split(',');

```



```
//alert("l: "+localStorage.getItem("localm"));

var nArray = [];

var mArray = [];

var ntext="";

var mtext="";

var rtext="";

var rArray = [];

for (i=0;i<100;i++){

    rArray[i]="*";

}

nArray[0]=lArray[0];nArray[1]=lArray[1];

nArray[2]=lArray[2];nArray[3]=lArray[3];

mArray[0]=lArray[0];mArray[1]=lArray[1];

mArray[2]=lArray[2];mArray[3]=lArray[3];

rArray[0]=lArray[0];rArray[1]=lArray[1];

rArray[2]=lArray[2];rArray[3]=lArray[3];

for (s=4;s<sArray.length-1;s+=2){

    var find=false;

    for (l=4;l<lArray.length-1;l++){

        if (sArray[s]==lArray[l]){

            find=true;

            if (sArray[s+1]>=lArray[s+1]){
```

```
                nArray[s]=sArray[s];
                nArray[s+1]=sArray[s+1];
            }
            else {
                nArray[s]=lArray[l];
                nArray[s+1]=lArray[l+1];
            }
        }
    }
}

if (!find){
    nArray[s]=sArray[s];
    nArray[s+1]=sArray[s+1];
}

}

for (count=0;count<sArray.length-1;count++){
    ntext += nArray[count]+' ';
}

ntext += nArray[nArray.length];

//alert("First: " + ntext);

for (l=4;l<lArray.length-1;l+=2){
    var find = false;
    for (n=4;n<nArray.length-1;n++){
```

```
        if (lArray[l]==nArray[n]){
            find=true;
            if (lArray[l+1]>=nArray[n+1]){
                mArray[l]=lArray[l];
                mArray[l+1]=lArray[l+1];
            }
            else {
                mArray[l]=nArray[n];
                mArray[l+1]=nArray[n+1];
            }
        }
    }
    if (!find){
        mArray[l]=lArray[l];
        mArray[l+1]=lArray[l+1];
    }
}
for (count=0;count<mArray.length-1;count++){
    mtext += mArray[count]+' ';
}
mtext += mArray[mArray.length];
//alert("Second: " + mtext);
```

```

var indx = 0;

for (ncount=4;ncount<nArray.length-1;ncount+=2){
    if
((nArray[ncount]=="*")(nArray[ncount]=="undefined")){ break;}
    else {
        rArray[ncount]=nArray[ncount];
        rArray[ncount+1]=nArray[ncount+1];
    }
    indx=ncount;
}

for (mcount=4;mcount<mArray.length-1;mcount+=2){
    if
((mArray[mcount]=="*")(mArray[mcount]=="undefined")){ break;}
    else {
        rArray[indx]=mArray[mcount];
        rArray[indx+1]=mArray[mcount+1];
    }
    indx+=2;
}

for (rcount=0;rcount<rArray.length-1;rcount++){
    rtext += rArray[rcount]+';';
}

rtext += rArray[rArray.length];

```

```
//alert("result: " + rtext);  
  
localStorage.setItem(moduleID,rtext);  
  
sendDataPost();  
  
}
```

Fungsi diatas digunakan untuk membuat variable data mahasiswa sehingga saat mahasiswa *login* menggunakan perangkat yang berbeda data yang telah tersipan dapat diperbarui

#### 4.3.1 Pembahasan Kuesioner

Perhitungan persentase kuesioner Jumlah total persentase tidak mencapai 100 % dikarenakan penulis menggunakan kuesioner skala likert. Sehingga terdapat perpecahan jawaban pada setiap pertanyaan, sedangkan untuk mendapatkan persentasi 100 % responden harus memilih jawaban pada point tertinggi atau sangat setuju.

Setelah pengujian aplikasi dan pengujian kelayakan dilakukan, dapat dilihat bahwa aplikasi *mobile learning* ini dapat membantu mahasiswa dalam kegiatan belajar. Hal ini terlihat dari hasil pengolahan data yang penulis lakukan menggunakan kuesioner. Hasil pengolahan data menunjukkan kelayakan aplikasi yang signifikan.

Dengan hasil tersebut, dapat dikatakan bahwa tujuan dari penelitian ini sudah tercapai, yaitu membantu dan memudahkan mahasiswa jurusan Teknik Informatika untuk mempelajari pemrograman bahasa java. Setelah tujuan penelitian tercapai, diharapkan manfaat penelitian juga tercapai yaitu meningkatkan pengetahuan dalam pembuatan aplikasi *mobile learning* dan membantu mahasiswa dalam kegiatan belajar.

Penelitian ini juga masih bisa dikembangkan dikemudian hari, mengingat aplikasi yang telah dibangun ini masih memiliki jangkauan materi yang terbatas. Aplikasi dapat dikembangkan lebih lanjut, seperti penambahan video tutorial, materi, soal dan masih banyak lagi.