

**PENGEMBANGAN TOPIK PEMBELAJARAN *CONTENT PROVIDER* DENGAN BAHASA JAVA PADA *ANDROID* PROGRAMMING LEARNING ASSISTANCE SYSTEM**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV  
Politeknik Negeri Malang

**Oleh:**

DEVANY CHRISTABELLA WIJAYA      NIM. 1741720092



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
JULI 2021**



**HALAMAN PENGESAHAN**  
**PENGEMBANGAN TOPIK PEMBELAJARAN CONTENT**  
**PROVIDER DENGAN BAHASA JAVA PADA ANDROID**  
**PROGRAMMING LEARNING ASSISTANCE SYSTEM**

**Disusun oleh:**

**DEVANY CHRISTABELLA WIJAYA      NIM. 1741720092**

**Skripsi ini telah diuji pada tanggal 04 Agustus 2021**

**Disetujui oleh:**

- |                             |  |  |
|-----------------------------|--|--|
| 1. Pembimbing<br>Utama      | : <u>Yan Watequlis Syaifuddin, ST., M.MT.</u><br>NIP. 198101052005011005     | <br>.....   |
| 2. Pembimbing<br>Pendamping | : <u>Rudy Ariyanto, ST., M.Cs.</u><br>NIP. 197111101999031002                | <br>.....  |
| 3. Penguji Utama            | : <u>Hendra Pradibta, SE., M.Sc.</u><br>NIP. 198305212006041003              | <br>..... |
| 4. Penguji<br>Pendamping    | : <u>Muhammad Afif Hendrawan., S.Kom.,<br/>MT</u><br>NIP. 199111282019031013 | <br>..... |

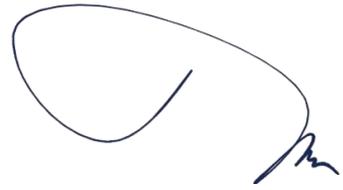
Mengetahui,

Ketua Jurusan  
Teknologi Informasi

Ketua Program Studi  
Teknik Informatika



Rudy Ariyanto, S.T., M.CS.  
NIP. 197111101999031002

  
Imam Fahrur Rozi, ST., MT.  
NIP. 198406102008121004

## PERNYATAAN

Dengan ini saya menyatakan bahwa pada Skripsi ini tidak terdapat karya, baik seluruh maupun sebagian, yang sudah pernah diajukan untuk memperoleh gelar akademik di Perguruan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan Oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini serta disebutkan dalam daftar sitasi/pustaka.

Malang, 21 Juni 2021



Devany Christabella Wijaya

## ABSTRAK

**Christabella W., Devany.** “Pengembangan Topik Pembelajaran *Content Provider* dengan Bahasa Java pada *Android Programming Learning Assistance System*”.  
**Pembimbing: (1) Yan Watequlis Syaifudin S.T., M.MT (2) Rudy Ariyanto, S.T., M.Cs.,.**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2021.**

Pemrograman aplikasi *Android* menjadi salah satu mata pelajaran yang populer di sekolah dan di universitas pada Jurusan Teknik Informatika. Pada aplikasi *Android*, sering kali penggunaan data digunakan untuk membuat aplikasi yang dinamis dan interaktif. Dari topik yang sebelumnya, *Android Programming Learning Assistance System* (APLAS) sudah menjadi platform yang dapat digunakan belajar *Android* secara mandiri. *Stage* yang sekarang diimplementasikan adalah *Stage Content Provider*. Dari *stage* tersebut terdapat tahap belajar pertama yaitu topik *Basic Data Storage*. *Basic Data Storage* terdiri dari materi mengenai arsitektur *Model-View-Viewmodel* (MVVM) yang menggunakan *Data Binding*, pemanfaatan *Shared Preferences* dan penyimpanan secara *internal*. Untuk pengujian pengerjaan dari topik *Basic Data Storage* yang dilakukan oleh 50 mahasiswa D4 IT di Politeknik Negeri Malang tingkat tiga. Untuk topik berikutnya adalah topik Database SQLite. Topik Database SQLite ini berkaitan tentang pembuatan database SQLite dan pengelolaan database SQLite. Dalam pengelolaan database SQLite dilakukan dengan cara menambahkan, mengedit dan menghapus tabel yang berada pada database SQLite tersebut.

**Kata Kunci:** *Android*, APLAS, *Content Provider*, *Basic Data Storage*, Database SQLite

## **ABSTRACT**

**Christabella W., Devany.** *“Development of Content Provider Learning Topics in Java on the Android Programming Learning Assistance System”*. **Supervisors: (1) Yan Watequlis Syaifudin S.T., M.MT (2) Rudy Ariyanto, S.T., M.Cs.,.**

**Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2021.**

*From the increasing level of popularity of smartphones that use the Android operating system, which has increased very rapidly. Android application programming has become one of the most popular subjects in schools and universities in the department of informatics engineering. In Android applications, data usage is often used to create dynamic and interactive applications. From the previous topic, the Android Programming Learning Assistance System (APLAS) has become a platform for Android self-learning. The stage that is currently being implemented is the Content Provider stage. From this stage, there is the first learning topic, namely Basic Data Storage. Basic Data Storage consists of material on the Model-View-Viewmodel (MVVM) architecture that uses Data Binding, utilization of Shared Preferences and internal storage. For testing of work Basic Data Storage Topic, a trial was carried out by 50 IT students at the third class of State Polytechnic of Malang. For the next stage is SQLite Database. This SQLite database contained SQLite databases created, edited and deleted. Editing the database could be added, edited and deleted tables in the database.*

**Keywords:** *Android, APLAS, Stage Content Provider, Topic Basic Data Storage, Topic SQLite Database*

## KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah SWT/Tuhan YME atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “PENGEMBANGAN TOPIK PEMBELAJARAN *CONTENT PROVIDER* DENGAN BAHASA JAVA PADA *ANDROID* PROGRAMMING LEARNING ASSISTANCE SYSTEM”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari bahwasannya dengan tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Yan Watequlis Syaifudin S.T., M.MT., selaku dosen pembimbing utama dalam penyusunan laporan akhir.
2. Bapak Rudy Ariyanto, ST., M.Cs., selaku dosen pembimbing kedua dan ketua jurusan Teknologi Informasi.
3. Dosen-dosen pengajar Jurusan Teknologi Informasi Politeknik Negeri Malang yang telah memberikan bimbingan dan ilmunya.
4. Bapak, ibu, dan keluarga tercinta yang dengan sepenuh hati memberikan doa dan motivasi dalam menyelesaikan laporan akhir.
5. Teman-teman Jurusan Teknologi Informasi Politeknik Negeri Malang atas dukungan dan motivasinya.
6. Seluruh relawan yang ikut berpartisipasi dalam proses uji coba. Sehingga mendukung terselesaikannya laporan akhir ini.
7. Bapak Imam Fahrur Rozi, ST., MT., selaku ketua program studi Teknik Informatika.
8. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan

kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, 21 Juni 2021

Penulis

# DAFTAR ISI

Halaman

SAMPUL DEPAN .....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN .....	iii
ABSTRAK .....	iv
ABSTRACT.....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR .....	ix
DAFTAR TABEL.....	x
BAB I. PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	2
BAB II. LANDASAN TEORI .....	5
2.1 Peneletian Terdahulu .....	5
2.2 <i>Test-Driven Development</i> (TDD).....	6
2.3 <i>Unit Test</i> .....	6
2.4 Junit .....	6
2.5 Robolectric API .....	6
2.6 Review APLAS .....	7
BAB III. METODOLOGI PENGEMBANGAN .....	8
3.1 Tahapan Pengembangan .....	8
3.2 Studi Literatur.....	8
3.3 Metode Pengembangan.....	8
3.3.1 Perencanaan dengan cepat ( <i>Quick Plan</i> ).....	9
3.3.2 Permodelan desain secara cepat ( <i>Modeling Quick Design</i> ) .....	9
3.3.3 Membangun prototype ( <i>Consturction of Prototype</i> ).....	9
3.3.4 Penerapan uji coba dan umpan balik ( <i>Deployment Delivery and Feedback</i> ).....	10
3.3.5 Komunikasi ( <i>Communication</i> ).....	10
3.4 Perancangan Sistem.....	10
3.5 Implementasi .....	10
3.6 Pengujian dan Analisis Sistem .....	11
3.7 Pengambilan Keputusan .....	11

BAB IV. ANALISIS DAN PERANCANGAN SISTEM.....	12
4.1 Analisa Kebutuhan Sistem.....	12
4.1.1 Kebutuhan Fungsional .....	12
4.1.2 Kebutuhan Non-Fungsional .....	13
4.2 Proses Pembelajaran oleh Siswa.....	13
4.3 Desain Topik <i>Basic Data Storage</i> .....	14
4.3.1 Activity Diagram Proses Validasi Mahasiswa di Android Studio 19	
4.3.2 Activity Diagram Melihat Hasil Validasi Mahasiswa di Platform APLAS .....	19
4.3.3 Activity Diagram Melihat Hasil Validasi Tugas Mahasiswa oleh Guru .....	21
4.4 Application for Assignment.....	21
BAB V. IMPLEMENTASI DAN PENGUJIAN.....	24
5.1 Implementasi .....	24
5.1.1 <i>Guide File</i> Topik <i>Basic Data Storage</i> .....	24
5.1.2 <i>Supplement File</i> Topik <i>Basic Data Storage</i> .....	27
5.1.3 <i>Test File</i> Topik <i>Basic Data Storage</i> .....	28
5.1.4 <i>Guide File</i> Topik Database SQLite .....	29
5.1.5 <i>Supplement File</i> Topik Database SQLite .....	34
5.1.6 <i>Test File</i> Topik <i>Basic Data Storage</i> .....	34
5.2 Pengujian .....	35
BAB VI. HASIL Pengujian DAN PEMBAHASAN .....	36
6.1 Hasil Pengujian.....	36
6.2 Pembahasan Hasil Pengujian.....	36
6.2.1 Contoh Tampilan Aplikasi <i>MyLibrary</i> .....	38
6.2.2 Umpan Balik Mahasiswa .....	38
BAB VII. KESIMPULAN DAN SARAN .....	39
7.1 Kesimpulan.....	39
7.2 Saran .....	39
DAFTAR PUSTAKA .....	40

## DAFTAR GAMBAR

	Halaman
Gambar 3. 1 Tahapan Pengembangan.....	8
Gambar 3. 2 SDLC Provider.....	9
Gambar 4. 1 Proses Pembelajaran Siswa.....	13
Gambar 4. 2 Gambaran Umum Pengerjaan Topik <i>CI Basic Data Storage</i> .....	14
Gambar 4. 3 <i>Activity Diagram</i> Mahasiswa Memvalidasi di Android Studio.....	19
Gambar 4. 4 <i>Activity Diagram</i> Melihat Hasil Validasi Tugas Mahasiswa.....	20
Gambar 4. 5 <i>Activity Diagram</i> Guru Melihat Hasil Tugas Mahasiswa.....	21
Gambar 4. 6 Implementasi <i>Guide File</i> pada <i>XML file</i> .....	27
Gambar 4. 7 Pengujian <i>Test Code</i> pada <i>XML file</i> .....	28
Gambar 4. 8 <i>Failed Test</i> .....	36
Gambar 5. 1 Penerapan Arsitektur MVVM dengan <i>Data Binding</i> .....	26
Gambar 5. 2 <i>key.png</i> .....	28
Gambar 5. 3 <i>mylibrary.png</i> .....	28
Gambar 5. 4 <i>myshop.png</i> .....	34
Gambar 5. 5 <i>Test Code</i> Menguji Pembuatan Tabel.....	34

## DAFTAR TABEL

	Halaman
Tabel 2. 1 Tabel Penelitian Terdahulu .....	5
Tabel 4. 1 Tujuan Pembelajaran Topik <i>Basic Data Storage</i> .....	15
Tabel 4. 2 Deskripsi Tugas Topik <i>Basic Data Storage</i> .....	16
Tabel 4. 3 Tujuan Pembelajaran Topik <i>Database SQLite</i> .....	17
Tabel 4. 4 Deskripsi Tugas Topik <i>Database SQLite</i> .....	17
Tabel 4. 5 Fitur Aplikasi <i>MyLibrary</i> .....	22
Tabel 4. 6 Fitur Aplikasi <i>MyShop</i> .....	22
Tabel 5. 1 Deskripsi <i>Guide File</i> Aplikasi <i>MyLibrary</i> .....	24
Tabel 5. 2 Deskripsi <i>Guide File</i> Aplikasi <i>MyShop</i> .....	29

## BAB I. PENDAHULUAN

### 1.1 Latar Belakang

Sejak 2014 penggunaan smartphone atau yang sering disebut ponsel pintar terus meningkat dan berkembang pesat. Smartphone ini memiliki sistem operasional. Sistem operasional paling populer dan paling banyak digunakan adalah *Android*. Penggunaan *Android* hampir mencapai angka 75% (Statcounter, 2020). Oleh karena itu, para ahli Teknologi Informasi mulai berlomba-lomba membuat aplikasi *Android* sehingga kebutuhan pembuatan aplikasi *Android* meningkat pesat.

Peningkatan pembuatan aplikasi *Android* membuat banyak sekali orang berlomba-lomba untuk mempelajarinya. Untuk awal pembelajaran dalam pembuatan aplikasi *Android* pasti diperlukan ahli untuk menuntun dan mengajari. Dan banyak juga mahasiswa Teknik Informatika yang belajar *Android* serta dosen sebagai yang mengajarkan. Namun banyak dosen yang merasa lelah karena memeriksa pekerjaan mahasiswa satu per satu dalam jumlah banyak.

Sehingga pembuatan APLAS (*Android Programming Learning Assistant System*) dapat membantu dosen dalam pengecekan tugas mahasiswa (Syarifudin et al., 2019). APLAS ini merupakan implementasi dari pembelajaran dasar *Android* berbasis Java. APLAS ini membantu pembelajaran *Android* sederhana dan dasar sehingga mudah dipahami oleh pelajar.

Topik pembelajaran terdahulu, APLAS sudah digunakan menjadi platform untuk belajar *Android* self-learning. Untuk topik yang akan diimplementasikan adalah topik *Content Provider*. Topik ini memiliki empat *stage* yaitu *Basic Data Storage*, *Database SQLite*, *Network Connection* dan *Data Service*.

*Stage Basic Data Storage* berkaitan dengan penggunaan arsitektur *Model-View-Viewmodel* (MVVM) yang menggunakan *Data Binding*, pemanfaatan *Shared Preferences* dan penyimpanan di *internal*. *Stage Basic Data Storage* ini telah diimplementasikan dalam sebuah aplikasi yaitu *MyLibrary*. Sedangkan *stage Database SQLite* berkaitan dengan pengelolaan database *internal* yang ada smartphone bernama *SQLite*. Implementasi dari *stage Database SQLite* dalam bentuk aplikasi *MyShop*.

Dalam pengimplementasian kedua *stage* terdapat *guide*, *test code* dan *supplement file*. *Guide* merupakan *file* yang berisikan tentang Langkah-langkah pembuatan aplikasi. *Test code* merupakan *file* yang digunakan untuk melakukan pengecekan pekerjaan di *Android Studio* yang bisa dilakukan secara mandiri. Untuk *supplement file* merupakan *file* tambahan yang digunakan dalam pembuatan aplikasi. Sedangkan untuk *stage Network Connection* dan *Data Service* belum dilakukan pengimplementasian.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang dapat diambil adalah sebagai berikut:

1. Bagaimana merancang dan membuat topik *Content Provider* pada APLAS?
2. Bagaimana cara melakukan pemeriksaan topik ini sudah berhasil atau belum di *Android Studio*?

## 1.3 Tujuan

Tujuan dari pembuatan sistem ini adalah sebagai berikut :

1. Untuk memudahkan mahasiswa untuk belajar mandiri terkait *Content Provider*
2. Mengintegrasikan platform ini pada APLAS

## 1.4 Batasan Masalah

Agar skripsi yang berjudul Pengembangan Topik Pembelajaran *Content Provider* pada APLAS dapat berjalan sesuai rencana dan tujuan awal, maka penulis menetapkan batasan-batasan masalah yaitu:

1. *Android* berbasis Jawa
2. Melakukan pengujian di Politeknik Negeri Malang
3. Menggunakan *Android Studio* sebagai media pengerjaan

## 1.5 Sistematika Penulisan

Uraian dalam laporan skripsi penulisan menyusun dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang dilakukan riset, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan terkait pengembangan topik *Content Provider*.

## BAB II LANDASAN TEORI

Bab ini berisikan penelitian terdahulu tentang APLAS dan *Test-driven Development*. Serta teori-teori yang mendasari dan berkaitan dengan masalah dalam perencanaan dan pengembang topik *Content Provider*. Perencanaan ini untuk mempermudah pemahaman dan penyelesaian masalah pengembangan topik *Content Provider*.

## BAB III METODOLOGI PENGEMBANGAN

Bab ini menjelaskan tentang metodologi yang digunakan dalam menyelesaikan tugas akhir. Metodologi pengembangan berisi tentang penjelasan empat tahap pengembangan, studi literatur yang dilakukan, dan penggunaan *prototyping models* untuk metode yang digunakan untuk pengembangan serta melakukan pengujian kepada *volunteer* untuk mengambil kesimpulan dari pengujian yang dilakukan.

## BAB IV ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisa kebutuhan fungsional dan non fungsional, arsitektur sistem tentang pengerjaan topik *Content Provider* dalam bentuk diagram dan pengenalan APLAS secara terperinci.

## BAB V IMPLEMENTASI DAN PENGUJIAN

Pada bab ini membahas tentang analisa alur kerja dari sistem dan penjelasan secara teknis pengujian yang dikerjakan 50 orang mahasiswa yang bersedia menjadi *volunteer*. Implementasi yang digunakan berdasarkan tahap analisis dan perancangan.

## BAB VI HASIL PENGUJIAN DAN PEMBAHASAN

Pada bab ini berisikan tentang hasil pengujian dari topik *Content Provider* pada *stage Basic Data Storage*. Hasil yang didapat terdiri dari umpan balik dan tingkat kesulitan dari *stage Basic Data Storage* pada setiap *test* berdasarkan dari waktu rata-rata pengerjaan untuk menyelesaikan masing-masing *guide* dari *stage Basic Data Storage*.



## BAB II. LANDASAN TEORI

### 2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama dengan judul penelitian penulis. Namun, penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang akan dilakukan.

Tabel 2. 1 Tabel Penelitian Terdahulu

<b>Nama Peneliti</b>	<b>Judul Penelitian</b>	<b>Hasil Penelitian</b>
Watequlis, Yan Nobuo, Syaifudin Minoru, Funabiki 2019	A Proposal of Advanced Widgets Learning Topic for Interactive Application in <i>Android</i> Programming Learning Assistance System	APLAS merupakan sistem yang digunakan untuk membantu belajar <i>Android</i> berbasis Bahasa Java dasar.
Syaifudin, Yan Watequlis Funabiki, Nobuo Kuribayashi, Minoru Kao, Wen Chung 2019	A Proposal of <i>Android</i> <i>Programming Learning</i> <i>Assistant System</i> with Implementation of Basic Application Learning	Tujuan dari dibuatnya APLAS untuk membantu mahasiswa belajar <i>Android</i> dasar dengan mudah dan membantu dosen memeriksa tugas mahasiswa

Dari beberapa judul penelitian yang telah dipaparkan, terpilih penelitian yang dilakukan oleh Yan Watequlis Syaifudin, Nobuo Funabiki, Minoru Kuribayashi, Wen Chung Kao yang berjudul “A Proposal of *Android Programming Learning Assistant System* with Implementation of Basic Application Learning” sebagai

referensi utama penulis dalam melakukan penelitian. Hal ini dikarenakan adanya pembahasan yang serupa dan pegangan karena merupakan seri terdahulu.

## 2.2 Test-Driven Development (TDD)

Metode TDD dapat diimplementasikan dengan cara pengujian aplikasi *Android* yang dilakukan melalui proses pengujian *blackbox* dan *whitebox* secara otomatis. Dalam proses pengujian ini menggunakan *test code* untuk menjadi patokan dari sebuah pengujian yang dilakukan secara otomatis. TDD ini yang diadopsi dari APLAS untuk melakukan pengujian mulai bagian yang kecil hingga besar. Bagian tersebut terdiri dari bagian kecil (*unit test*), bagian sedang (*integration test*) dan bagian besar (*UI test*). APLAS mengadopsi *unit testing* yang menggunakan *JUnit* dan *Robolectric* (Syarifudin et al., 2019).

## 2.3 Unit Test

*Unit test* merupakan praktek untuk melakukan pengujian pada bagian unit kecil, individu dan terisolasi dari *source code* yang berjalan pada *Java Virtual Machine* (JVM). *JUnit* menjadi standar untuk pengujian unit pada aplikasi *Android*.

## 2.4 Junit

*JUnit* merupakan framework untuk pengujian bahasa pemrograman Java yang paling populer dan yang bisa digunakan di *Android Studio*. Dalam pengujian unit diperlukan *source code* yang berisi metode pengujian yang ditunjukkan dengan nama *@Test*. Setiap metode pengujian berisi metode pernyataan untuk menjalankan fungsi tunggal dalam *source code* dan menentukan *success* atau *failed* pada kasus pengujian ini. Metode pernyataan disediakan oleh library *org.junit.Assert class extend java.lang.Object class*.

## 2.5 Robolectric API

*Robolectric* adalah framework yang menghadirkan pengujian integrasi yang cepat dan handal. *Robolectric* menyediakan *framework* pengujian yang memungkinkan untuk menguji aplikasi *Android* di *Java Virtual Machine* (JVM).

## 2.6 Review APLAS

APLAS merupakan platform pembelajaran *Android* secara online, mandiri dan otomatis. Dalam proses pembelajaran dilakukan dengan cara mahasiswa melakukan registrasi akun jika belum memiliki akun, jika sudah memiliki akun maka siswa melakukan login. Kemudian pilih menu *Download Material* dan memilih topik yang akan dikerjakan. Unduh semua file tugas yang terdiri dari *guide file*, *test file* dan *supplement file*. Selanjutnya siswa mengerjakan topik pembelajaran di *Android Studio* dengan mengikuti *guide file* sebagai petunjuk pengerjaan. Pengecekan dilakukan dengan menjalankan *test code* pada *Android Studio* dan mendapatkan hasilnya otomatis. Jika terjadi *error* atau *failed*, maka mahasiswa perlu memperbaiki lagi pekerjaannya agar dapat mendapatkan hasil *success*. Jika sudah mendapatkan hasil *success*, maka siswa dapat mengunggahnya ke *platform* APLAS untuk dilakukan validasi.

Pada proses pembelajaran *Android* dilakukan dengan *stage* yang paling dasar yaitu *User Interface*, *Interactive Application*, *Content Provider* dan *Service Interactive*. *Stage* dibuat oleh Bapak Yan Watequlis Syaifudin. *Stage* yang dibuat berdasarkan dari tingkat kesulitannya. Setiap *stage* memiliki topik pembelajaran. Topik yang dibuat setiap *stage* juga dibuat berdasarkan tingkat kesulitannya. Pada *Stage Content Provider* terdiri dari *Basic Data Storage*, *SQLite Database*, *Network Connection* dan *Data Service*. Masing-masing topik memiliki fungsinya masing-masing. *Basic Data Storage* untuk mempelajari tentang untuk arsitektur *Model-View-Viewmodel* (MVVM) dengan *data binding*, pemanfaatan *Shared Preferences* dan penyimpanan *internal*.

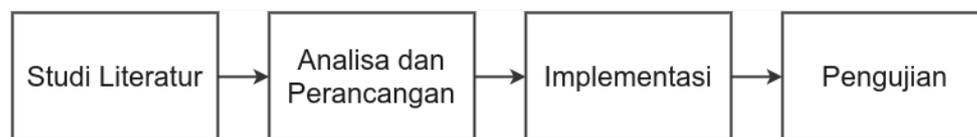
*SQLite Database* ini memiliki fungsi penyimpanan data pada database yang bersifat pribadi atau privat, yang dinamai *SQLite*. *Network connection* untuk menyimpan data pada internet termasuk pada server jaringan dan *cloud*. *Data service* memiliki tujuan menunjukkan data pribadi atau *private* menjadi data umum atau *public*.

## BAB III. METODOLOGI PENGEMBANGAN

Pada bab metodologi pengembangan, menjelaskan metodologi yang digunakan dalam pengembangan aplikasi interaktif untuk skripsi ini. Tahapan tersebut antara lain: studi literatur, analisa kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis sistem, dan pengambilan kesimpulan.

### 3.1 Tahapan Pengembangan

Tahapan pengembangan dalam proses pembuatan topik pembelajaran *Content Provider* pada *stage Basic Data Storage* di APLAS, seperti pada gambar 3.1 di bawah.



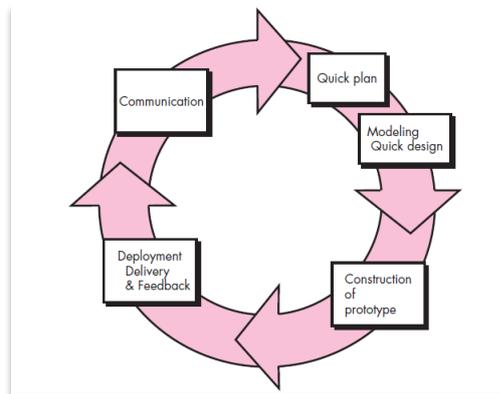
Gambar 3. 1 Tahapan Pengembangan

### 3.2 Studi Literatur

Studi Literatur dilakukan dengan mengumpulkan dan mempelajari literatur seperti buku, jurnal dan beberapa sumber terkait dengan topik *Content Provider* serta pengimplemetasi *Test-driven development*, *JUnit* dan *Robolectric*. Studi literatur yang dilakukan memperoleh data yang menunjang untuk proses selanjutnya.

### 3.3 Metode Pengembangan

Dalam rancang bangun penerapan sistem parkir untuk mengembangkan *Content Provider* dengan *Android* bahasa Java pada APLAS yaitu dengan alur dari *Software Development Life Cycle (SDLC) Model Provider*. *SDLC Provider* memiliki alur sebagai berikut,



Gambar 3. 2 SDLC Provider

### 3.3.1 Perencanaan dengan cepat (*Quick Plan*)

Seperti yang ditampilkan pada gambar 3.1 bahwa pengembangan sistem diawali dengan *Quick Plan*. *Quick plan* ini merupakan pembuatan perencanaan secara cepat dan tepat. Dalam pengembangan *Content Provider* sangat memerlukan perencanaan. Hal ini untuk meminimalisir kesalahan yang terjadi. Serta perencanaan untuk mengorganisir pengerjaan agar dapat diselesaikan dengan maksimal.

*Quick plan* pada pengembangan *Content Provider* ini melakukan perencanaan bagaimana sistem dikembangkan dan target lama pengembangan. Dari bagaimana sistem dikembangkan, kita melakukan penataan apa saja yang diperlukan dalam pengembangan tersebut. Sedangkan untuk target lama pengembangan ini mengatur waktu pengerjaan pengembangan dengan sebaik-baiknya.

### 3.3.2 Permodelan desain secara cepat (*Modeling Quick Design*)

Setelah *Quick Plan* selesai dilakukan, langkah selanjutnya adalah dengan melakukan design cara kerja menggunakan flowchart serta proses fungsionalnya menggunakan use case diagram, sequence diagram dan activity diagram.

### 3.3.3 Membangun prototype (*Consturction of Prototype*)

*Construction of prototype* merupakan tahapan dimana melakukan pembuatan prototype atau yang sering disebut sketsa dasar. Dalam pembuatan protyep ini pasti dibuatlah *Content Provider* secara sederhana untuk dilakukan pengujian. Sehingga pembuatan prototype sangat diperlakukan untuk memperoleh hasil yang maksimal dan baik.

### **3.3.4 Penerapan uji coba dan umpan balik (*Deployment Delivery and Feedback*)**

Deployment Delivery and Feedback merupakan fase uji coba secara bebas dan menerima kritikan dari *tester*. Uji coba ini bertujuan mengetahui *Content Provider* sudah layak atau belum. Proses pengujian dilakukan dengan cara mahasiswa melakukan pengerjaan sesuai dengan *guide file* yang telah tersedia. Kemudian mahasiswa melakukan pengecekan secara mandiri dengan *test file* yang dijalankan di *Android Studio*. *Test file* sendiri berfungsi untuk melakukan pengecekan apakah yang dikerjakan sudah sesuai atau tidak dengan *guide* yang diberikan.

Selanjutnya jika telah sesuai dengan *guide file* dan telah lolos dari *test file*, dilakukan pengunggahan *file* kerja sesuai yang diperlukan di APLAS untuk dilakukan pengecekan secara menyeluruh oleh Robolectric. Setelah divalidasi maka mahasiswa akan mendapatkan hasil validasi.

### **3.3.5 Komunikasi (*Communication*)**

Communication merupakan fase melakukan komunikasi ulang untuk melakukan evaluasi dan pembenahan dari *stage Content Provider*. Communication ini bertujuan untuk mempercepat perbaikan dari *stage Content Provider* sehingga menghasilkan hasil yang maksimal dan baik.

## **3.4 Perancangan Sistem**

Perancangan sistem merupakan tahapan yang menjelaskan tentang desain dari sistem dari segi model ataupun arsitektur yang digunakan. Perancangan sistem bertujuan untuk mempermudah proses pembuatan *stage Content Provider* pada APLAS.

## **3.5 Implementasi**

Implementasi merupakan tahapan membangun dan membuat *stage Content Provider*. Pada tahapan ini semua literatur yang didapatkan akan diterapkan. Proses pembangunan *stage Content Provider* ini mengacu pada tahap perancangan. Untuk tahapan implementasi yaitu:

1. Pembuatan desain antarmuka setiap aplikasi.
2. Pembuatan *supplement file*.
3. Pembuatan *guide file*.

4. Pembuatan *test file* yang digunakan untuk pengecekan secara *blackbox*.
5. Mengunggah tugas yang telah dikerjakan dan dilakukan validasi oleh website APLAS.
6. Laporan hasil pembelajaran.

### **3.6 Pengujian dan Analisis Sistem**

Setelah melakukan implementasi pada *stage Content Provider*. Untuk melakukan pengujian apakah topik yang dibuat sesuai tujuan yang diharapkan atau tidak, maka diperlukan langkah pengujian. Pengujian dilakukan dengan meminta *volunteer* mengerjakan aplikasi pada masing-masing *stage*.

### **3.7 Pengambilan Keputusan**

Pengambilan keputusan merupakan tahap terakhir pada proses pembuatan *stage Content Provider*. Pengambilan keputusan dilakukan setelah semua tahapan sebelumnya selesai dilakukan. Kesimpulan diperoleh dari hasil dari pengujian topik. Serta penulisan waktu pengerjaan, saran dan komentar oleh *volunteer* dapat membantu membuat kesimpulan dan memperbaiki kekurangan pada *stage Content Provider*.

## **BAB IV. ANALISIS DAN PERANCANGAN SISTEM**

Bab ini membahas tentang perancangan dari pengembangan *stage Content Provider* pada topik *Basic Data Storage* dan *SQLite Database* pada APLAS. Tahap perancangan yang digunakan terbagi menjadi analisa kebutuhan sistem, perancangan antarmuka dan perancangan kebutuhan sistem.

### **4.1 Analisa Kebutuhan Sistem**

Tahapan analisa kebutuhan sistem memiliki tujuan untuk menjelaskan kebutuhan pada sistem yang akan dibuat. Analisa kebutuhan sistem yang digunakan untuk identifikasi pengguna dan daftar kebutuhan.

#### **4.1.1 Kebutuhan Fungsional**

Tahap identifikasi pengguna platform bertujuan untuk mengidentifikasi pengguna yang berinteraksi dengan platform online APLAS. Pengidentifikasi pengguna ini berdasar dari tingkatan pemilik hak akses yang dimiliki.

Admin memiliki akses yang terdiri dari pengelolaan data pembelajaran, data pengguna dan melihat data hasil pembelajaran. Pengelolaan data pembelajaran terdiri dari mengunggah, mengedit dan menghapus data pembelajaran yang terdiri dari *guide file*, *test file* dan *supplement file*. Pengelolaan data pengguna terdiri dari pengaturan hak akses terhadap pengguna, contohnya mengubah pengguna baru menjadi guru. Sedangkan untuk melihat semua data pembelajaran dari semua mahasiswa.

Dosen memiliki hak akses atas mahasiswa yang diampunya. Hak akses dosen terdiri dari mendaftarkan mahasiswa dan melihat hasil pembelajaran mahasiswa. Mendaftarkan mahasiswa terdiri dari melakukan penerimaan mahasiswa yang telah mendaftar ke dalam platform APLAS. Melihat hasil pembelajaran mahasiswa yang diampunya yang terdiri topik yang telah dikerjakan oleh mahasiswa dan jumlah mahasiswa yang *success*, *failed* dan *error* dari hasil pembelajaran.

Mahasiswa memiliki hak akses yang terdiri dari mengunduh file tugas, mengunggah hasil pembelajaran dan melihat hasil validasi dari hasil pembelajarannya. Mahasiswa mengunduh file tugas yang terdiri *guide file*, *test file* dan *supplement file* sesuai dengan topik pembelajaran yang diinginkan. Mahasiswa

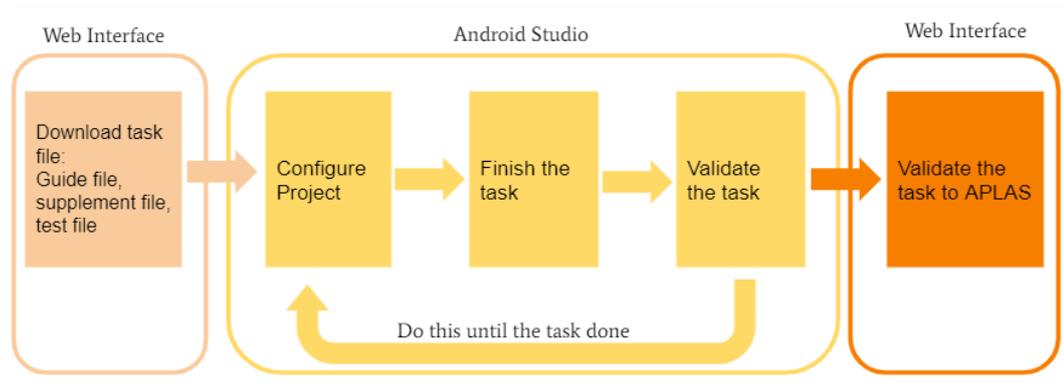
dapat mengunggah hasil pembelajarannya dengan tiga cara. Cara mengunggah hasil pembelajaran dari *multi files*, *github link* dan *zip file* dari proyek *Android*. Setelah mengunggah hasil pembelajaran, mahasiswa dapat melihat hasil validasinya.

#### 4.1.2 Kebutuhan Non-Fungsional

Berikut merupakan kebutuhan non fungsional dalam implementasi topik *Basic Data Storage* dan topik *Database SQLite* dari *Stage Content Provider*:

1. Platform online APLAS dapat dijalankan pada beberapa aplikasi *web browser* seperti *Chrome*, *Mozilla Firefox*, *Microsoft Edge*.
2. Dalam proses pengerjaan topik pembelajaran di *Android Studio* menggunakan bahasa pemrograman *Java*

#### 4.2 Proses Pembelajaran oleh Siswa



Gambar 4. 1 Proses Pembelajaran Siswa

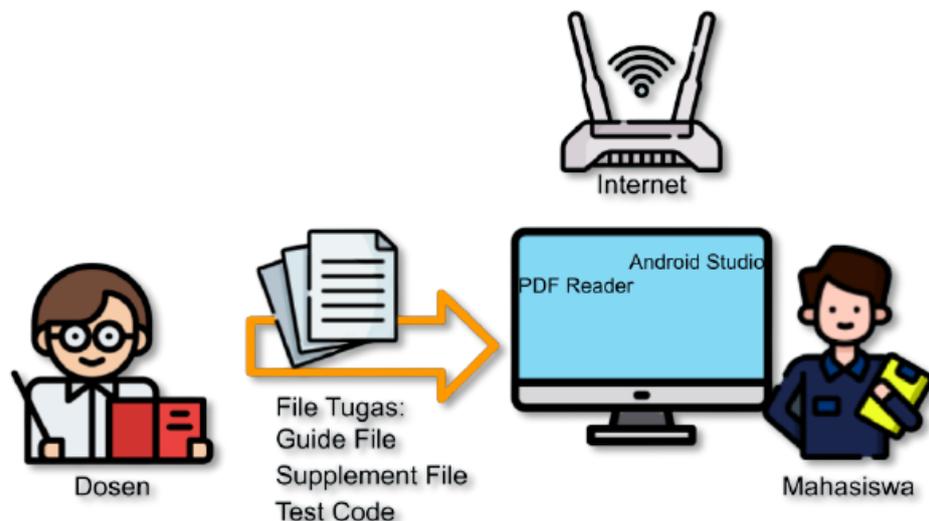
Proses pembelajaran siswa merupakan proses dimana mahasiswa mempelajari dari *stage Content Provider*. Proses pembelajaran siswa ini terbagi menjadi 5 proses. Proses pertama yaitu mengunduh *file* tugas yang terdiri dari *guide file*, *supplement file* dan *test file* di platform APLAS pada topik *CI Basic Data Storage*. Kemudian mengerjakan tugas sesuai *guide file* di *Android Studio*. Dalam proses pengerjaan akan membutuhkan *supplement file* untuk membantu dalam pengerjaan. Dan *test file* digunakan untuk melakukan pengecekan *blackbox* dan *whitebox* secara otomatis apakah tugas yang dikerjakan sesuai dengan *guide file*. Jika belum sesuai

maka hasil dari pengecekan adalah *failed*. Jika sudah sesuai maka hasil pengecekan adalah *success*.

Setelah aplikasi pembelajaran jadi dan sudah dilakukan pengecekan dengan *test code* secara keseluruhan serta mendapat hasil *success*. Maka dapat melakukan pengumpulan hasil akhir ke APLAS. Dan APLAS akan melakukan validasi hasil pengerjaan secara otomatis. Jika hasil validasi tersebut *failed* maka mahasiswa gagal dalam mengerjakan tugas. Jika hasil validasi tersebut *success* maka mahasiswa berhasil dalam mengerjakan tugasnya.

### 4.3 Desain Topik *Basic Data Storage*

Desain Topik *Basic Data Storage* adalah gambaran proses pembuatan hingga proses pengerjaan topik pembelajaran *Basic Data Storage*. Perancangan topik *Basic Data Storage* sebagai pengembangan dari topik sebelumnya.



Gambar 4. 2 Gambaran Umum Pengerjaan Topik *CI Basic Data Storage*

Untuk mengerjakan *stage Content Provider*, mahasiswa harus memiliki koneksi internet agar dapat mengakses platform APLAS menggunakan web browser. Bagi mahasiswa yang belum memiliki akun maka diharuskan untuk melakukan *register*. Berikutnya mahasiswa melakukan *login* dengan menggunakan *email* dan *password* yang telah didaftarkan sebelumnya. Selanjutnya mahasiswa memilih menu *Download Materials* dan memilih topik *CI Basic Data Storage* serta mengunduh *file* tugas yang terdiri dari *guide file*, *test file* dan *supplement file*.

Kemudian mahasiswa mengerjakan tugasnya menggunakan *Android Studio* dan bantuan *pdf reader* untuk membuka *guide file*. Lalu mahasiswa mengecek tugasnya dengan menjalankan *test file* di *Android Studio*. Jika hasil yang didapat masih *failed* maka mahasiswa harus memperbaikinya. Jika hasil yang didapat sudah *success* maka mahasiswa dapat mengunggah hasil tugasnya di website APLAS pada menu *Submit Your Project*. Pada saat akan mengunggah tugas harus dipastikan topik yang dipilih adalah topik *CI Basic Data Storage*.

Validator akan melakukan validasi dengan metode *Test-driven Development* (TDD). Dan hasil validasi akan disimpan di database pada server dan akan tampil pada website APLAS untuk menunjukkan hasil kerja mahasiswa. Hasil tersebut terdiri dari *success*, *failed* dan *error*. Tujuan dari pembelajaran untuk topik *Basic Data Storage* sebagai berikut.

Tabel 4. 1 Tujuan Pembelajaran Topik *Basic Data Storage*

No	Tujuan	Deskripsi
LO1	Application Resources	siswa dapat menerapkan aplikasi sumber daya dalam proyek <i>Android</i> , seperti <i>drawable</i> , <i>string</i> , dan sumber daya lainnya
LO2	UI Components	mahasiswa mampu mengimplementasikan komponen UI termasuk <i>Color Picker</i> , <i>ImageView</i> , <i>ListView</i> , dan <i>Spinner</i> .
LO3	Data Binding	siswa memahami cara penerapan Arsitektur <i>Model-View-Viewmodel</i> (MVVM) dengan Data Binding.
LO4	Shared Preferences	siswa memahami cara penerapan <i>SharedPreferences</i> untuk menyimpan persisten data.
LO5	<i>Internal Storage</i>	siswa memahami cara penerapan menyimpan data di penyimpanan <i>internal</i>

Dari lima tujuan dari topik pembelajaran topik *Basic Data Storage* yang memenuhi dalam sepuluh tugas dari *guide file* terdiri sebagai berikut.

Tabel 4. 2 Deskripsi Tugas Topik *Basic Data Storage*

No.	Deskripsi Tugas	Learning Object
1	Starting the project and configuring the resources	LO1
2	Designing the UI of MainActivity	LO2
3	Designing the UI of InputDataActivity	LO2
4	Designing the UI of ShowDataActivity	LO2
5	Creating SharedPreferences for user data and book data	LO4
6	Developing input data and show data with data binding	LO3
7	Creating data binding models for MVVM architecture	LO3
8	Developing MainActivity class	LO2, LO3
9	Developing InputDataActivity class	LO3, LO4
10	Developing ShowDataActivity class	LO3, LO4, LO5

Implementasi dari topik *Basic Data Storage* adalah aplikasi *MyLibrary*. *MyLibrary* memiliki tiga *activity* yang terdiri dari *MainActivity*, *InputDataActivity* dan *ShowDataActivity*. *MainActivity* merupakan *activity* untuk memasukkan data dari pengguna. Data pengguna tersebut terdiri dari *username*, *country* dan *phone number*. *InputDataActivity* merupakan *activity* untuk memasukkan, mengedit dan menyimpan data buku. Data buku tersebut terdiri dari *Title*, *type* dan *year*. *ShowDataActivity* merupakan *activity* untuk melihat data buku yang telah disimpan.

#### 4.4 Desain Topik Database SQLite

Desain Topik *Database SQLite* adalah gambaran proses pembuatan hingga proses pengerjaan topik pembelajaran *Database SQLite*. Perancangan topik

*Database SQLite* sebagai pengembangan dari topik sebelumnya. Dalam pembuatan topik Database SQLite memiliki enam tujuan pembelajaran yang terdiri dari:

Tabel 4. 3 Tujuan Pembelajaran Topik *Database SQLite*

No	Tujuan	Deskripsi
L01	Application Resources	mahasiswa dapat menerapkan aplikasisumber daya dalam proyek Android, seperti drawable, string, dan sumber daya lainnya
L02	UI Components	mahasiswa mampu mengimplementasikan komponen UI termasuk <i>Color Picker, ImageView, ListView, dan Spinner.</i>
L03	Database SQLite	mahasiswa mampu membuat dan menghapus database SQLite
L04	Create, drop table	mahasiswa mampu membuat dan menghapus tabel pada database SQLite
L05	Fill table	mahasiswa mampu mengisi tabel
L06	Update, delete table contents	mahasiswa mampu mengubah dan menghapus isi tabel

Dari enam tujuan dari topik pembelajaran topik *Database SQLite* yang memenuhi dalam sebelas tugas dari *guide file* terdiri sebagai berikut

Tabel 4. 4 Deskripsi Tugas Topik *Database SQLite*

No.	Deskripsi Guide	Learning Objectives
1	Starting the project, configuring the resources and designing the UI of LoginActivity	L01, L02
2	Designing the UI of RegisterActivity, creating database, table 'user' and developing LoginActivity class and RegisterActivity class	L01,L02,L03,L04
3	Designing the UI of SellerHomeActivity, creating model 'product' and 'productRecyclerAdapter'	L01,L02
4	Designing the UI of AddProductActivity, creating table 'product' and developing AddProductActivity class	L01,L02,L04,L05
5	Designing the UI of SellerDetailProductActivity and developing SellerDetailProductActivity class and SellerHomeActivity class	L01,L02,L06
6	Designing the UI of BuyerHomeActivity,creating table 'cart' and developing BuyerHomeActivity class	L01,L02,L03,L04
7	Designing the UI of BuyerDetailProductActivity, creating model 'cart' and developing BuyerDetailProductActivity class	L01,L02
8	Designing the UI of BuyerCartActivity and developing BuyerCartActivity class	L01,L02,L04,L05

9	Designing the UI of <i>DetailCartBuyerActivity</i> and developing <i>DetailCartBuyerActivity</i> class	L01,L02,L06
10	Designing the UI of <i>SellerCartActivity</i> and developing <i>SellerCartActivity</i> class	L01,L02,L04,L05,L06
11	Designing the UI of <i>DetailCartSellerActivity</i> and developing <i>DetailCartSellerActivity</i> class	L01,L02,L04,L05,L06

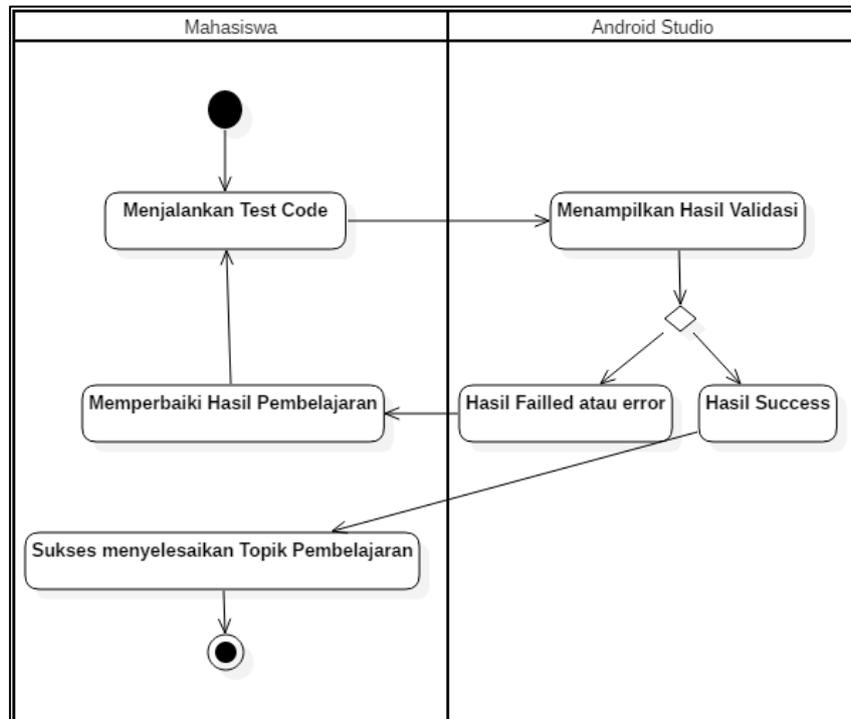
Implementasi dari topik *Databse SQLite* adalah aplikasi *MyShop*. *MyShop* memiliki dua activity utama yaitu *loginActivity* dan *registerActivity*. Untuk melakukan registrasi akun pengguna membutuhkan *username* dan *password*. Untuk meningkatkan keamanan registrasi maka ditambahkan *confirm password*. User memilih role sebagai *buyer* atau *seller*. Kemudian melakukan login pada *loginActivity*.

Kemudian jika login menggunakan role *seller*, maka *seller* memiliki lima activity yaitu *SellerHomeActivity*, *AddProductActivity*, *SellerProductDetailActivity*, *SellerCartActivity* dan *SellerCartDetail*. *SellerHomeActivity* memiliki *recycler view* untuk menampilkan barang yang telah ditambahkan oleh *seller*. *AddProductActivity* merupakan *activity* yang digunakan untuk menambahkan *product* oleh *seller*. *SellerProductDetailActivity* digunakan untuk menampilkan detail dari *product* dan *seller* dapat menghapus *product* tersebut. *SellerCartActivity* memiliki *recycler view* yang menampilkan *product* yang dipesan oleh *buyer*. *SellerCartDetailActivity* digunakan untuk menampilkan detail dari *product* yang dipesan oleh *buyer*.

Jika login menggunakan role *buyer*, maka *buyer* memiliki 4 activity utama yaitu *BuyerHomeActivity*, *BuyerProductDetailActivity*, *BuyerCartActivity* dan *BuyerProductDetailActivity*. *BuyerHomeActivity* memiliki *recycler view* yang digunakan untuk menampilkan *product* yang ditambahkan oleh *seller*. *BuyerProductDetailActivity* menampilkan detail *product* dan *buyer* dapat memesan *product* dengan *quantity* yang diinginkan. *BuyerCartActivity* digunakan untuk menampilkan barang yang dipesan oleh *buyer* tersebut. *BuyerProductCartActivity* digunakan untuk mengedit dan menghapus *product* yang telah dimasukkan ke dalam *cart*.

#### 4.4.1 Activity Diagram Proses Validasi Mahasiswa di Android Studio

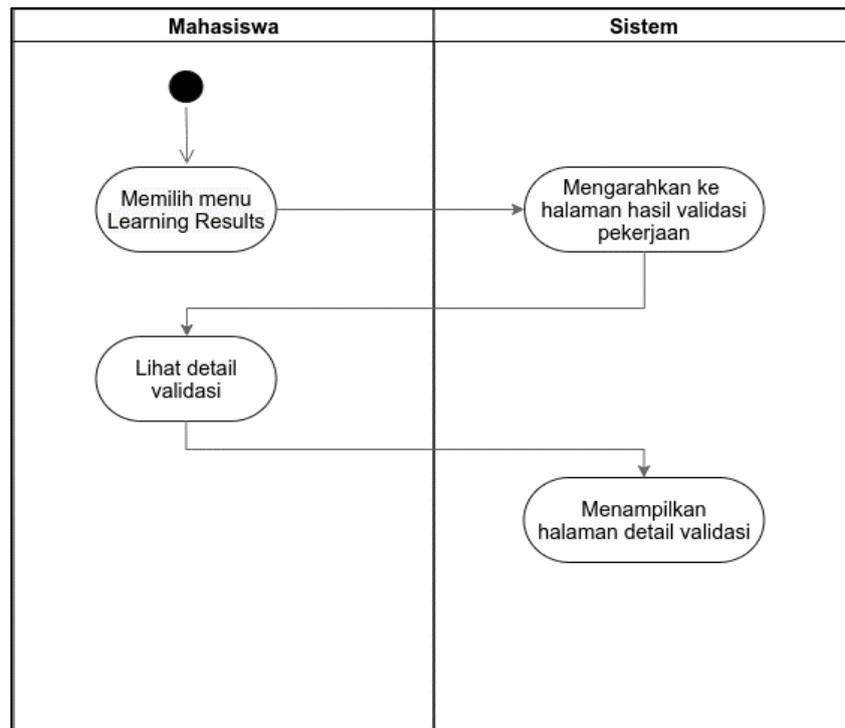
Alur proses *Android Studio* melakukan validasi terhadap hasil kerja mahasiswa. Akan dijelaskan dengan menggunakan *activity diagram* sebagai berikut:



Gambar 4. 3 *Activity Diagram* Mahasiswa Memvalidasi di Android Studio

#### 4.4.2 Activity Diagram Melihat Hasil Validasi Mahasiswa di Platform APLAS

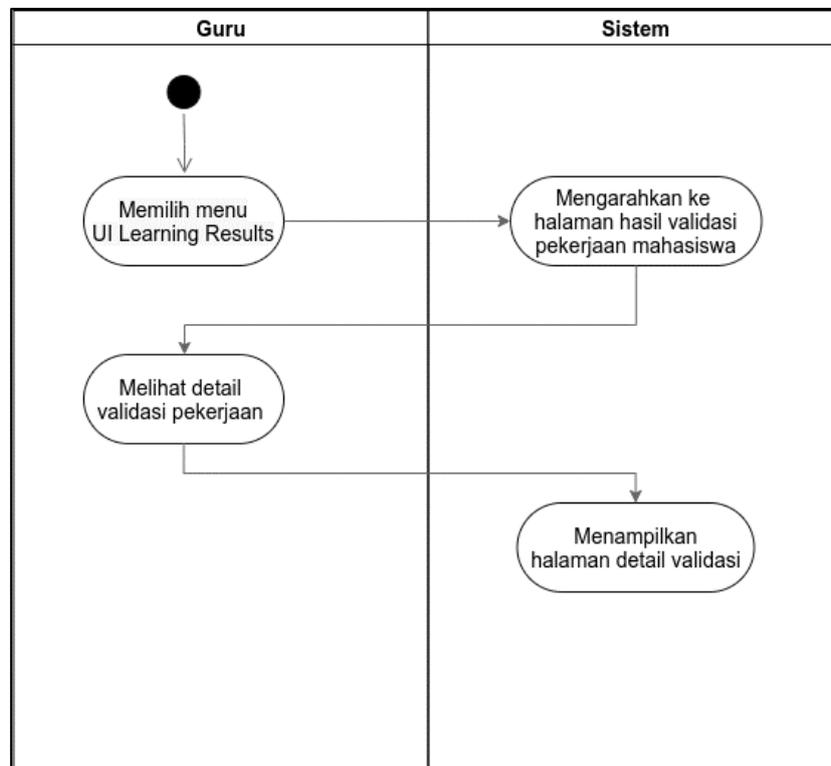
Alur proses sistem bagi mahasiswa untuk melihat hasil validasi tugas yang telah diunggah ke platform APLAS. Akan dijelaskan dengan menggunakan *activity diagram* sebagai berikut:



Gambar 4. 4 Activity Diagram Melihat Hasil Validasi Tugas Mahasiswa

#### 4.4.3 Activity Diagram Melihat Hasil Validasi Tugas Mahasiswa oleh Guru

Alur proses sistem untuk guru untuk melihat hasil validasi tugas yang telah diunggah ke website APLAS oleh mahasiswa yang akan dijelaskan *activity diagram* seperti berikut.



Gambar 4. 5 Activity Diagram Guru Melihat Hasil Tugas Mahasiswa

#### 4.5 Application for Assignment topik Basic Data Storage

*Application for assignment* topik *Basic Data Storage* merupakan aplikasi yang diserahkan untuk memenuhi tugas dari topik *Basic Data Storage*. Pengembangan *Stage Content Provider* merupakan pengembangan dari topik-topik sebelumnya yaitu *User Interface* dan *Interactive Application*. Untuk topik *Basic Data Storage* menggunakan aplikasi pembejaran yaitu *MyLibrary*. Sedangkan untuk topik *Database SQLite* menggunakan aplikasi pembelajaran yaitu *MyShop*.

Topik *Basic Data Storage* digunakan untuk mempelajari arsitektur *Model-View-Viewmodel (MVVM)* dengan *Data Binding*, pemanfaatan *shared preferences* dan penyimpan *internal*. Topik *Basic Data Storage* menggunakan aplikasi *MyLibrary* yang memiliki fitur sebagai berikut:

Tabel 4. 5 Fitur Aplikasi *MyLibrary*

<b>Fitur</b>	<b>Fungsi</b>
Input user data	Untuk memasukkan data user yang akan menggunakan aplikasi
Input data book	Untuk memasukkan data buku milik <i>user</i>
Save data book	Untuk menyimpan data buku milik <i>user</i>
Edit data book	Untuk mengubah data buku milik <i>user</i>
Delete data book	Untuk menghapus data buku milik <i>user</i>
Background settings	Untuk mengatur background aplikasi sesuai keinginan <i>user</i>

#### 4.6 Application for Assignment Topik *Database SQLite*

*Application for assignment* topik *Database SQLite* merupakan aplikasi yang diserahkan untuk memenuhi tugas dari topik *Database SQLite*. Pengembangan *Stage Content Provider* merupakan pengembangan dari topik-topik sebelumnya yaitu *User Interface* dan *Interactive Application*. Untuk topik *Database SQLite* menggunakan aplikasi pembejaran yaitu *MyShop*.

Topik *Database SQLite* digunakan untuk mempelajari pembuatan *Database SQLite*, membuat tabel pada *Database SQLite* dan mengisi, mengedit serta menghapus isi dari tabel. Topik *Database SQLite* menggunakan aplikasi *MyShop* yang memiliki fitur sebagai berikut:

Tabel 4. 6 Fitur Aplikasi *MyShop*

<b>Fitur</b>	<b>Fungsi</b>
Membuat user	Untuk menambahkan user
Login user	Untuk user dapat masuk ke dalam aplikasi
Add product	Untuk seller dapat menambahkan product
Edit product	Untuk seller dapat mengubah detail product
Delete product	Untuk seller dapat menghapus product
Delete product from cart	Untuk seller menghapus product dalam cart
Add product to cart	Untuk buyer menambahkan product ke dalam cart
Edit Cart	Untuk buyer mengedit product yang dipesan



## BAB V. IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan tentang pembuatan aplikasi berdasarkan teori yang digunakan dalam pembuatan sistem. Penjelasan berisi langkah-langkah dalam membangun aplikasi. Pada implementasi sistem ini menggunakan bahasa pemrograman Java dengan bantuan aplikasi *Android Studio*.

### 5.1 Implementasi

Dalam proses implementasi aplikasi *MyLibrary* untuk topik pembelajaran *Basic Data Storage* di APLAS. Pembelajaran topik dimulai dengan mengunduh *file* pembelajaran yang terdiri dari 10 buah *guide file*, 10 buah *test file* dan 2 buah *supplement file*. Dalam pengujian pada *guide file* dilakukan dengan cara mencobanya dari awal hingga menjadi aplikasi. Dalam pengerjaan aplikasi tersebut apakah aplikasi yang dibuat sesuai dengan yang diharapkan. Jika belum sesuai maka *guide file* perlu diperbaiki. Jika sudah sesuai maka *guide test* siap digunakan oleh para mahasiswa.

#### 5.1.1 Guide File Topik Basic Data Storage

*Guide file* merupakan *file* yang berisikan mengenai arahan pembuatan aplikasi pembelajaran. *Guide file* memberikan arahan dalam mengatur *user interface* aplikasi hingga membuat *function* untuk masing-masing *activity* aplikasi. Pada topik *Basic Data Storage* akan diimplementasikan dalam bentuk aplikasi bernama *MyLibrary*. Aplikasi *MyLibrary* menerapkan arsitektur *Model-View-Viewmodel* (MVVM) dengan menggunakan *data binding*, pemanfaatan *shared preferences* dan melakukan penyimpanan pada *internal*.

Dalam pembuatan aplikasi *MyLibrary* membutuhkan sepuluh *guide file*, sepuluh *test file* dan dua *supplement file*. *Test file* digunakan untuk menguji tugas yang dikerjakan pada setiap *guide file*. *Guide file* tersebut berisikan tentang:

Tabel 5. 1 Deskripsi *Guide File* Aplikasi *MyLibrary*

No.	Deskripsi <i>Guide</i>
1	Starting the project and configuring the resources
2	Designing the UI of MainActivity
3	Designing the UI of InputDataActivity

4	Designing the UI of ShowDataActivity
5	Creating SharedPreferences for user data and book data
6	Developing input data and show data with data binding
7	Creating data binding models for MVVM architecture
8	Developing MainActivity class
9	Developing InputDataActivity class
10	Developing ShowDataActivity class

a. Starting the project and configuring the resource

Memulai sebuah projek dengan melakukan konfigurasi sumber. Sumber tersebut terdiri dari nama *package name*, *SDK minimal*, *target device*, nama aplikasi, *color resource*, *drawable resource*, *image resource* dan *gradle*. *Package name* yang diperlukan adalah ‘*org.aplas*’, sedangkan *SDK minimal* yang diperlukan adalah 21.

*Target device* yang dibutuhkan adalah 9, nama aplikasi adalah *MyLibrary*. *Color resource*, *drawable resource* merupakan warna dan *item XML* yang dibutuhkan untuk menunjang pembuatan aplikasi. *Image resource* merupakan gambar yang disediakan sebagai *supplement file*. *Gradle* merupakan alat otomatis *build* untuk pengembangan aplikasi *Android*.

b. Designing the UI of *MainActivity*, *InputDataActivity* and *ShowDataActivity*

Mendesain tampilan dari *MainActivity*, *InputDataActivity* dan *ShowDataActivity* dilakukan pada file XML setiap *activity*. Dalam mendesain yang perlukan adalah *item* yang dibutuhkan setiap *activity* seperti *Text View*, *Edit Text*, *Image View*, *Spinner* dan lain-lain. Dan dapat mengatur *item* sesuai dengan kebutuhan aplikasi.

c. Creating SharedPreferences for user data and book data

Membuat *sharedpreferences* untuk *user data* dan *book data* membutuhkan *models*. *Models* dibuat untuk mengambil dan menyiapkan *item* dari *user* dan *book*. *Item* yang dibutuhkan *user* adalah *color*, *name*, *country* dan *phone*. Kemudian membuat fungsi di ‘*UserData.java*’ yang digunakan untuk mengambil nilai dari semua *item user* dan membuat fungsi yang digunakan untuk menyimpan, menghapus serta mengecek *item* yang dimiliki *user*.

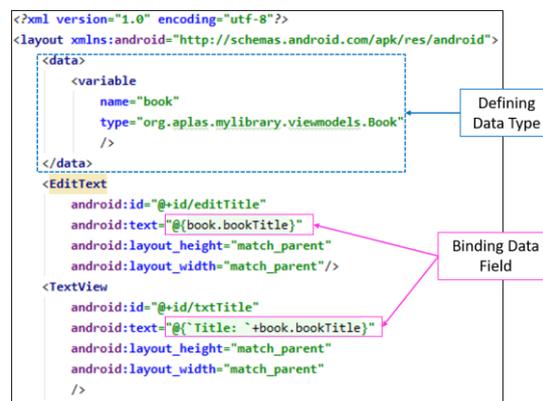
Membuat fungsi di *'BookData.java'* untuk menyimpan item *book* yang terdiri dari *title*, *year* dan *type*. Fungsi yang dibuat juga meliputi menyimpan, mengubah dan menghapus *item book* serta fungsi yang digunakan memuat hasil penyimpanan *item book*.

d. Developing input data and show data with data binding

Mengembangkan input data dan show data dengan data binding dilakukan dengan membuat *viewmodels*. *Viewmodels* mengambil dan mengatur nilai yang dimiliki *user* dan *book* pada tampilan. Nilai tersebut diambil dari *Edit Text* pada setiap tampilan *activity*.

e. Creating data binding models for MVVM architecture

Membuat *data binding* dengan arsitektur *Model View Viewmodel* (MVVM). *Model* bertugas menyediakan data yang diperlukan aplikasi. *View* bertugas merepresentasikan data yang berisi *User Interface*. *Viewmodel* bertugas untuk menyimpan dan mengambil data dari *Model* dan digunakan oleh *View* untuk ditampilkan. Contoh dari penggunaan arsitektur MVVM dengan *data binding* sebagai berikut:



Gambar 5. 1 Penerapan Arsitektur MVVM dengan *Data Binding*

f. Developing *MainActivity* class, *InputDataActivity* class and *ShowDataActivity* class

Mengembangkan *MainActivity* digunakan untuk mendeklarasikan semua *item* pada tampilan *MainActivity* dan mengatur fungsi untuk setiap *button*. *Button PickColor* memiliki fungsi untuk membuka *color pallete* dan bisa mengaturnya menjadi latar belakang aplikasi. *Button SaveUser* memiliki

fungsi menyimpan data pengguna dan warna latar belakang yang telah dipilih.

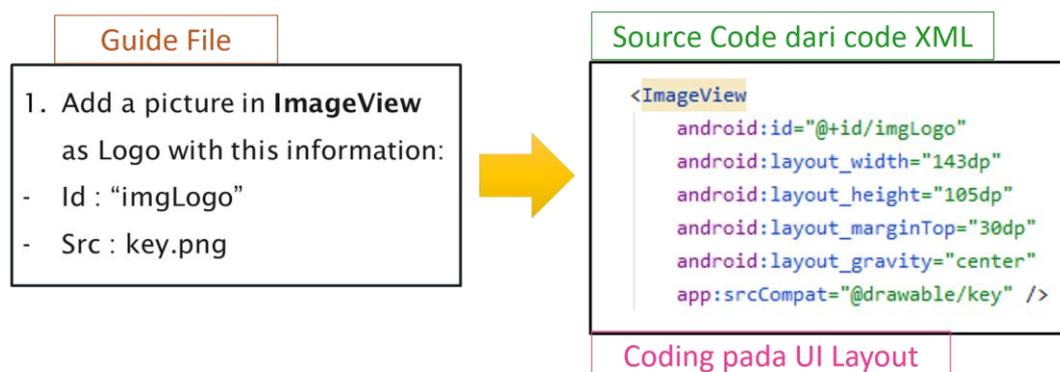
g. Developing InputDataActivity class

Mengembangkan *InputDataActivity* digunakan untuk mendeklarasikan semua *item* pada tampilan *InputDataActivity* dan mengatur fungsi untuk setiap *button*. *Button Logout* memiliki fungsi untuk menghapus pengguna yang masuk. *Button SaveBook* memiliki fungsi untuk menyimpan data buku yang ke dalam penyimpanan internal. *Button Show* memiliki fungsi untuk melihat data buku yang telah disimpan.

h. Developing ShowDataActivity class

Mengembangkan *ShowDataActivity* digunakan untuk mendeklarasikan semua *item* pada tampilan *ShowDataActivity* dan mengatur fungsi untuk setiap *button*. *Spinner spBook* memiliki fungsi untuk menampilkan pilihan file buku yang telah disimpan. *Button Logout* memiliki fungsi untuk menghapus pengguna yang masuk. *Button AddData* memiliki fungsi untuk menambahkan data buku baru. *Button EditData* memiliki fungsi untuk mengubah data buku. *Button DelData* memiliki fungsi untuk menghapus data buku.

Contoh implementasi *guide file* menjadi *source code* pada *XML file* sebagai berikut:



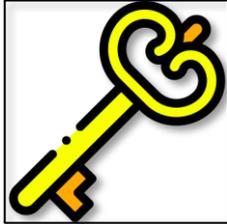
Gambar 4. 6 Implementasi *Guide File* pada *XML file*

### 5.1.2 Supplement *File Topik Basic Data Storage*

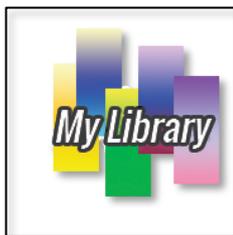
*Supplement file* merupakan *file* pendukung dalam pembuatan aplikasi. *Supplement file* antara lain gambar logo untuk aplikasi, gambar untuk *Image Button*. *Supplement file* dalam pembuatan aplikasi *MyLibrary* terdiri dari dua gambar yang

menjadi logo activity. *Supplement file* tersebut menggunakan ekstensi file *' .png '* dan gambarnya sebagai berikut:

Gambar 5. 2 key.png

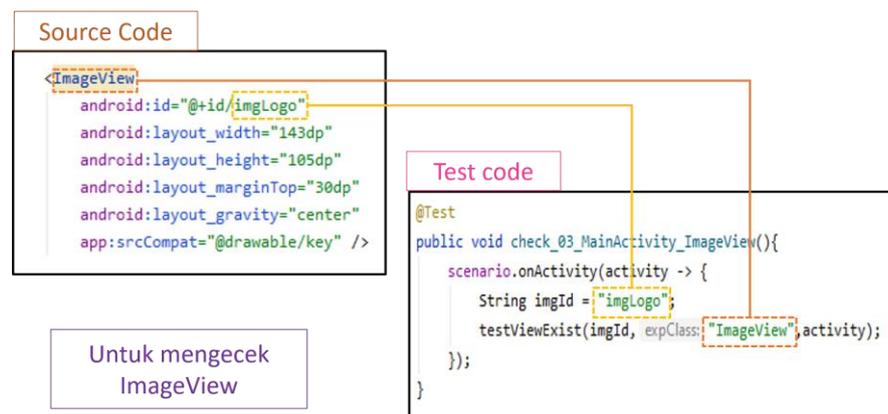


Gambar 5. 3 mylibrary.png



### 5.1.3 Test File Topik Basic Data Storage

*Test file* merupakan *file* yang digunakan untuk menguji hasil kerja dari mahasiswa secara *blackbox* dan *whitebox* dengan otomatis di *Android Studio* dengan menggunakan metode *Test-Driven Development*. Satu *test file* menguji hasil kerja mahasiswa setiap satu *guide file*. Pada pengerjaan aplikasi *MyLibrary* memerlukan pengecekan menggunakan sepuluh *test code*. Berikut contoh *test code* yang diujikan pada *XML file*:



Gambar 4. 7 Pengujian Test Code pada XML file

#### 5.1.4 Guide File Topik Database SQLite

Pada topik Database SQLite akan diimplementasikan dalam bentuk aplikasi bernama *MyShop*. Aplikasi *MyShop* menerapkan pembuatan dan mengelola *database*, melakukan pembuatan dan menghapus tabel, menambahkan, mengedit dan menghapus data pada tabel.

Dalam pembuatan aplikasi *MyShop* membutuhkan sebelas *guide file*, sebelas *test file* dan satu *supplement file*. *Test file* digunakan untuk menguji tugas yang dikerjakan pada setiap *guide file*. *Guide file* tersebut berisikan tentang:

Tabel 5. 2 Deskripsi *Guide File* Aplikasi *MyShop*

No.	Deskripsi <i>Guide File</i>
1	Starting the project, configuring the resources and designing the UI of <i>LoginActivity</i>
2	Designing the UI of <i>RegisterActivity</i> , creating database, table 'user' and developing <i>LoginActivity</i> class and <i>RegisterActivity</i> class
3	Designing the UI of <i>SellerHomeActivity</i> , creating model ' <i>product</i> ' and ' <i>productRecyclerViewAdapter</i> '
4	Designing the UI of <i>AddProductActivity</i> , creating table ' <i>product</i> ' and developing <i>AddProductActivity</i> class
5	Designing the UI of <i>SellerDetailProductActivity</i> and developing <i>SellerDetailProductActivity</i> class
6	Designing the UI of <i>BuyerHomeActivity</i> , creating table ' <i>cart</i> ' and developing <i>BuyerHomeActivity</i> class
7	Designing the UI of <i>BuyerDetailProductActivity</i> , creating model ' <i>cart</i> ' and developing <i>BuyerDetailProductActivity</i> class
8	Designing the UI of <i>BuyerCartActivity</i> and developing <i>BuyerCartActivity</i> class
9	Designing the UI of <i>DetailCartBuyerActivity</i> and developing <i>DetailCartBuyerActivity</i> class
10	Designing the UI of <i>SellerCartActivity</i> and developing <i>SellerCartActivity</i> class
11	Designing the UI of <i>DetailCartSellerActivity</i> and developing <i>DetailCartSellerActivity</i> class

- a. Starting the project, configuring the resources and designing the UI of *LoginActivity*

Dalam memulai projek pertama kali harus mengkonfigurasi seperti minimal *SDK*, nama aplikasi, nama *package* yang digunakan dan

konfigurasi *gradle*. Kemudian mengubah nama '*MainActivity*' menjadi '*LoginActivity*' dan mengubahnya menjadi activity utama.

Mendesain tampilan '*LoginActivity*' sebagai tampilan login yang terdiri dari dua *InputEditText* digunakan untuk memasukkan *username* dan *password*, sebuah *TextView* digunakan untuk mengarahkan ke '*RegisterActivity*' dan sebuah *button* digunakan untuk mengarah ke *Home*.

- b. Designing the UI of *RegisterActivity*, creating database, table 'user' and developing *LoginActivity* class and *RegisterActivity* class

Mendesain tampilan '*RegisterActivity*' sebagai tampilan untuk pengguna mendaftarkan diri ke dalam sistem. Tampilan '*RegisterActivity*' memerlukan tiga buah *InputEditText* yang digunakan untuk mengisi *username*, *password* dan *confirm password*, sebuah *Spinner* digunakan untuk memilih *role* sebagai *buyer* atau *seller*, sebuah *button* digunakan untuk mendaftarkan data pengguna ke dalam sistem dan sebuah *TextView* digunakan untuk mengarahkan ke '*LoginActivity*'.

Pembuatan *models user* digunakan penyedia data yang terdiri dari *username*, *password* dan *role*. Pembuatan database SQLite yang bernama '*MyShop.db*' memerlukan versi database yang digunakan adalah 1. Selanjutnya membuat tabel '*user*' memerlukan *id*, *name*, *password* dan *role* yang digunakan untuk menyimpan dan mengecek data *user*.

Pengembangan *LoginActivity* class dan *RegisterActivity* class dengan cara mendeklarasikan semua *item* yang ada pada tampilan, membuat koneksi *class* dengan *table* '*user*' agar dapat menyimpan, mengecek data pengguna.

- c. Designing the UI of *SellerHomeActivity*, creating model 'product' and '*productRecyclerAdapter*'

Mendesain tampilan dari *SellerHomeActivity* sebagai tampilan utama bagi pengguna *seller*. Tampilan tersebut terdiri dari sebuah *RecyclerView* yang digunakan untuk menampilkan daftar *product*, sebuah *TextView* yang digunakan sebagai judul dari halaman *SellerHomeActivity* dan empat buah *button* yang memiliki fungsi yaitu mengecek keranjang *buyer*, menyegarkan data *product*, menambahkan *product* dan melakukan *logout*.

Membuat *models product* digunakan penyedia data *product* yang terdiri dari *id*, *name* dan *price*. Pembuatan *productRecyclerAdapter* digunakan untuk mengatur tampilan *RecyclerView* yang terdiri dari *name* dan *price* sehingga lebih mudah dalam pengecekan.

- d. Designing the UI of *AddProductActivity*, creating table '*product*' and developing *AddProductActivity* class

Mendesain tampilan *AddProductActivity* sebagai tampilan untuk menambahkan *product* oleh *seller*. Tampilan tersebut terdiri dari sebuah *TextView* yang digunakan sebagai judul tampilan, dua buah *InputEditText* yang digunakan menambahkan *name product* dan *price product*, dan dua buah *button* yang digunakan untuk menambahkan *product* dan kembali ke tampilan sebelumnya.

Pembuatan tabel *product* yang terdiri dari *id*, *name* dan *price*. Pengembangan *AddProductActivity* class digunakan untuk mendeklarasikan semua *item* yang berada pada tampilan *AddProductActivity* dan membuat fungsi untuk menyimpan data *product* ke dalam tabel *product*.

- e. Designing the UI of *SellerDetailProductActivity* and developing *SellerDetailProductActivity* class

Mendesain tampilan *SellerDetailProductActivity* sebagai tampilan yang menampilkan detail *product* kepada *seller*. Tampilan tersebut dari sebuah *TextView* yang digunakan sebagai judul dari tampilan, dua buah *InputEditText* yang digunakan menampilkan *name* dan *price product* serta dua buah *button* yang digunakan untuk menghapus *product* dan kembali ke tampilan sebelumnya.

Mengembangkan *SellerDetailProductActivity* digunakan untuk mendeklarasikan semua *item* pada tampilan *SellerDetailProductActivity* dan mengatur fungsi untuk setiap *button*.

- f. Designing the UI of *BuyerHomeActivity*, creating table '*cart*' and developing *BuyerHomeActivity* class

Mendesain tampilan dari *BuyerHomeActivity* sebagai tampilan utama bagi pengguna *seller*. Tampilan tersebut terdiri dari sebuah *RecyclerView*

yang digunakan untuk menampilkan daftar *product*, sebuah *TextView* yang digunakan sebagai judul dari halaman *BuyerHomeActivity* dan tiga buah *button* yang memiliki fungsi yaitu mengecek keranjang, menyegarkan data *product* dan melakukan *logout*.

Pembuatan tabel *cart* yang terdiri dari *id*, *username*, *product id*, *product name*, *product price* dan *product quantity*. Pengembangan *BuyerHomeActivity class* digunakan untuk mendeklarasikan semua *item* yang berada pada tampilan *BuyerHomeActivity*.

- g. Designing the UI of *BuyerDetailProductActivity*, creating model '*cart*' and developing *BuyerDetailProductActivity class*

Mendesain tampilan *BuyerDetailProductActivity* sebagai tampilan yang menampilkan detail *product* kepada *buyer*. Tampilan tersebut dari sebuah *TextView* yang digunakan sebagai judul dari tampilan, tiga buah *InputEditText* yang digunakan menampilkan *name* dan *price product* dan mengisi *quantity product* serta dua buah *button* yang digunakan untuk menambahkan *product* ke dalam *cart* dan kembali ke tampilan sebelumnya.

Membuat *models cart* digunakan penyedia data *cart* yang terdiri dari *id*, *username*, *product id*, *product name*, *product price* dan *product quantity*. Mengembangkan *BuyerDetailProductActivity* digunakan untuk mendeklarasikan semua *item* pada tampilan *BuyerDetailProductActivity* dan mengatur fungsi untuk setiap *button*.

- h. Designing the UI of *BuyerCartActivity* and developing *BuyerCartActivity class*

Mendesain tampilan dari *BuyerCartActivity* sebagai tampilan *cart* bagi pengguna *buyer*. Tampilan tersebut terdiri dari sebuah *RecyclerView* yang digunakan untuk menampilkan daftar *product*, sebuah *TextView* yang digunakan sebagai judul dari halaman *BuyerCartActivity* dan dua buah *button* yang memiliki fungsi yaitu menyegarkan keranjang, dan kembali ke tampilan sebelumnya.

Mengembangkan *BuyerCartActivity* digunakan untuk mendeklarasikan semua *item* pada tampilan *BuyerCartActivity* dan mengatur fungsi untuk setiap *button*.

- i. Designing the UI of *DetailCartBuyerActivity* and developing *DetailCartBuyerActivity* class

Mendesain tampilan dari *DetailCartBuyerActivity* sebagai tampilan *cart* bagi pengguna *seller*. Tampilan tersebut terdiri dari sebuah *TextView* yang digunakan sebagai judul dari halaman *DetailCartBuyerActivity*, empat buah *InputEditText* yang digunakan untuk menampilkan *product name*, *product price* dan *product buyer* dan mengisi *quantity product* serta dua buah *button* yang memiliki fungsi yaitu menghapus *product* dari keranjang dan kembali ke tampilan sebelumnya.

Mengembangkan *DetailCartBuyerActivity* digunakan untuk mendeklarasikan semua item pada tampilan *DetailCartBuyerActivity* dan mengatur fungsi untuk setiap *button*.

- j. Designing the UI of *SellerCartActivity* and developing *SellerCartActivity* class

Mendesain tampilan dari *SellerCartActivity* sebagai tampilan *cart* bagi pengguna *seller*. Tampilan tersebut terdiri dari sebuah *RecyclerView* yang digunakan untuk menampilkan daftar *product*, sebuah *TextView* yang digunakan sebagai judul dari halaman *SellerCartActivity* dan dua buah *button* yang memiliki fungsi yaitu menyegarkan keranjang, dan kembali ke tampilan sebelumnya.

Mengembangkan *SellerCartActivity* digunakan untuk mendeklarasikan semua item pada tampilan *SellerCartActivity* dan mengatur fungsi untuk setiap *button*.

- k. Designing the UI of *DetailCartSellerActivity* and developing *DetailCartSellerActivity* class

Mendesain tampilan dari *DetailCartSellerActivity* sebagai tampilan *cart* bagi pengguna *seller*. Tampilan tersebut terdiri dari sebuah *TextView* yang digunakan sebagai judul dari halaman *DetailCartSellerActivity*, empat buah *InputEditText* yang digunakan untuk menampilkan *product name*, *product price*, *quantity* dan *product buyer* serta dua buah *button* yang memiliki fungsi yaitu menghapus keranjang dan kembali ke tampilan sebelumnya.

Mengembangkan *DetailCartSellerActivity* digunakan untuk mendeklarasikan semua item pada tampilan *DetailCartSellerActivity* dan mengatur fungsi untuk setiap button.

### 5.1.5 Supplement File Topik Database SQLite

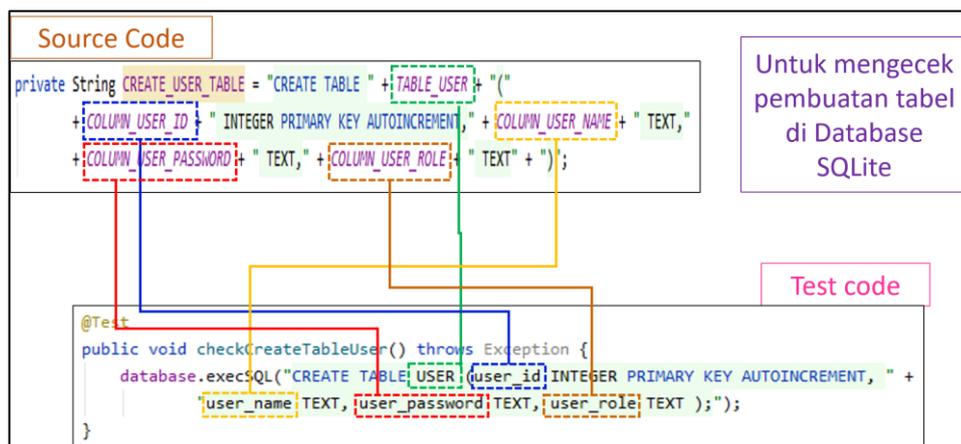
*Supplement file* dalam pembuatan aplikasi *MyShop* terdiri dari satu gambar yang menjadi logo activity. *Supplement file* tersebut menggunakan ekstensi file *.png* dan gambarnya sebagai berikut:



Gambar 5. 4 myshop.png

### 5.1.6 Test File Topik Database SQLite

*Test file* merupakan *file* yang digunakan untuk menguji hasil kerja dari mahasiswa secara *blackbox* dan *whitebox* dengan otomatis di *Android Studio* dengan menggunakan metode *Test-Driven Development*. Satu *test file* menguji hasil kerja mahasiswa setiap satu *guide file*. Pada pengerjaan aplikasi *MyShop* memerlukan pengecekan menggunakan sebelas *test code*. Berikut adalah contoh pengujian untuk pembuatan tabel di Database SQLite:



Gambar 5. 5 Test Code Menguji Pembuatan Tabel

## 5.2 Pengujian

Pengujian topik *Basic Data Storage* ini dilakukan dengan bantuan dari 50 orang mahasiswa Politeknik Negeri Malang. Mahasiswa tersebut merupakan mahasiswa tingkat 3 dari Jurusan Teknik Informatika dari kelas TI 3B dan TI 3C. Semua mahasiswa sudah mendapatkan pelajaran *Android* sebelumnya. Pengujian dilakukan dalam kurun waktu seminggu.

Sebelum pengujian dipastikan bahwa semua *volunteer* sudah terdaftar sebagai mahasiswa pada platform APLAS dan memiliki aplikasi *Android Studio* pada perangkat mereka. *Volunteer* melakukan login pada *platform APLAS* kemudian masuk ke menu *Download Materials* dan memilih topik *Basic Data Storage*. *Volunteer* mengunduh *task file* yang terdiri dari *guide file*, *test file* dan *supplement file*. Kemudian *volunteer* mengerjakan di *Android Studio* sesuai petunjuk pada *guide file*.

Selanjutnya *volunteer* melakukan pengujian menggunakan *test file*. Pengujian dilakukan dengan cara menjalankan *test code*. Jika hasil yang diperoleh *failed* atau *error*, maka *volunteer* harus memperbaiki hasil pengerjaannya. Jika hasil yang diperoleh *success*, maka *volunteer* telah mengerjakan dengan benar dan *volunteer* bisa mengunggah hasil pengerjaan di platform APLAS untuk divalidasi.

Mengunggah hasil pengerjaan dilakukan pada menu *Submit your result* dan pastikan topik yang dipilih *Basic Data Storage*. Dalam mengunggah hasil pengerjaan dapat dilakukan dengan tiga acara yaitu *zip file*, *multi files* dan *github link*. Hasil validasi akan muncul pada platform APLAS pada menu *Validation Result*. Jika hasil validasi *success*, maka *volunteer* telah menyelesaikan topik *Basic Data Storage* dengan benar. Namun jika hasil validasi *error* atau *failed*, maka *volunteer* perlu memperbaiki hasil pengerjaannya lagi.

## BAB VI. HASIL PENGUJIAN DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil pengujian dan pembahasan dari pengerjaan topik *Content Provider* pada APLAS.

### 6.1 Hasil Pengujian

Pengujian dilakukan kepada 50 orang *volunteer*. Hasil dari pengujian aplikasi pembelajaran topik *Basic Data Storage* yaitu 49 orang *success* mengerjakan topik pembelajaran dan 1 orang *failed* mengerjakan topik pembelajaran. Mahasiswa mengerjakan topik *Basic Data Storage* dengan waktu 7 hari. Dalam mengunggah hasil kerja dapat memilih cara pengunggahan.

Cara pengunggahan yang pertama adalah *multi files*, yaitu dengan mengunggah satu per satu *file* yang diperlukan. Untuk cara kedua adalah *zip file*, yaitu dengan mengunggah *zip file* sesuai kriteria. Dan yang terakhir menggunakan *github link*, yaitu dengan mengunggah *link github* yang telah berisikan pekerjaan.

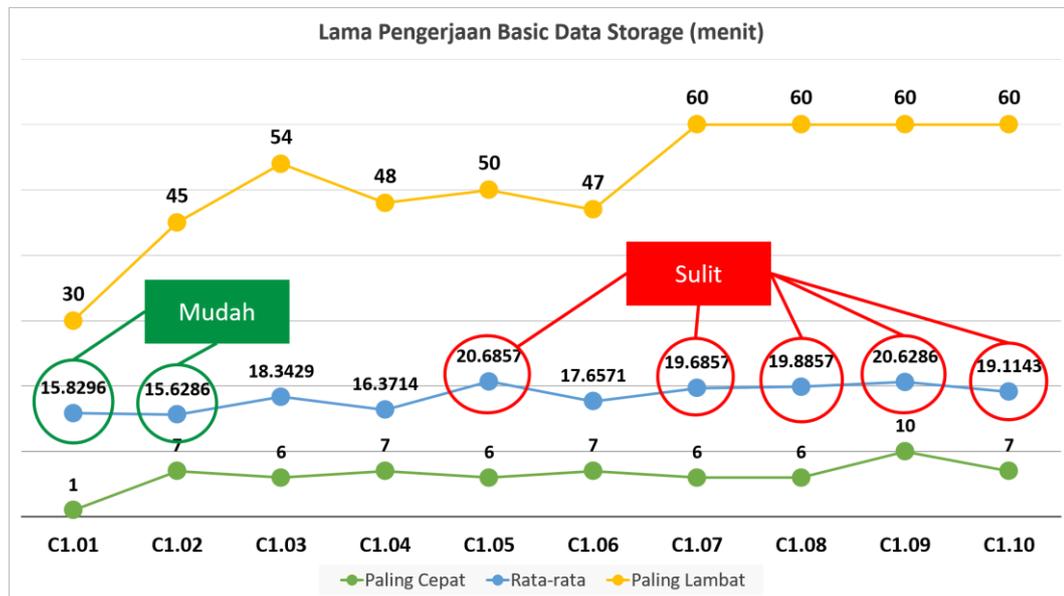
### 6.2 Pembahasan Hasil Pengujian

Berdasarkan hasil uji coba kepada 50 orang mahasiswa yang mengerjakan topik *Basic Data Storage* selama tujuh hari. Didapatkan 49 orang mahasiswa dapat menyelesaikan dengan berhasil dan 1 orang mahasiswa menyelesaikannya namun mendapatkan hasil *failed*. Seorang mahasiswa mengalami *failed* dikarenakan adanya kesalahan dalam menulis *id* pada *text view*. *Id* yang benar adalah *txtTitleInput* namun mahasiswa tersebut menulis *id* nya menjadi *txtTitleinput*. Sehingga *test code* tidak dapat menemukan *txtTitleInput* pada *InputDataActivity* yang menyebabkan *error* seperti gambar berikut.



Gambar 4. 8 Failed Test

Setiap mahasiswa memiliki kurun waktu pengerjaan yang berbeda-beda. Waktu tersebut dirangkum seperti pada gambar grafik waktu sebagai berikut.



Gambar 6. 1 Grafik Waktu Pengerjaan Topik *Basic Data Storage*

Dari grafik tersebut terdapat tiga jenis garis. Untuk garis berwarna oren menunjukkan waktu terlama dalam menyelesaikan setiap *task* yang ada pada topik. Dan untuk garis berwarna hijau menunjukkan waktu tercepat dalam menyelesaikan setiap *task* pada topik. Sedangkan untuk garis berwarna biru menunjukkan rata-rata waktu yang dibutuhkan untuk menyelesaikan setiap *task* yang ada pada topik. Rata-rata waktu pengerjaan semakin singkat maka semakin mudah *task* tersebut diselesaikan. Sedangkan rata-rata waktu pengerjaan semakin lama, maka semakin sulit *task* tersebut diselesaikan.

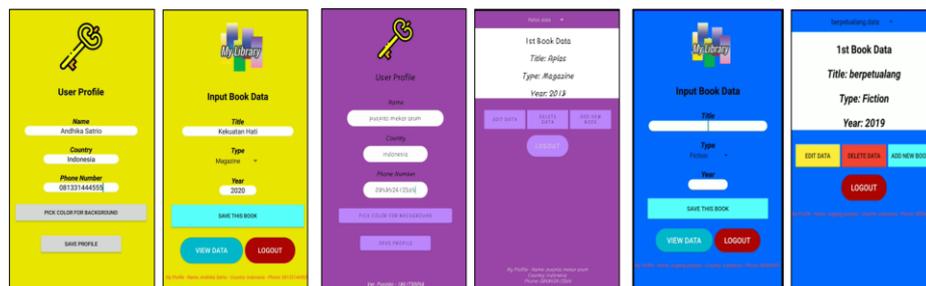
Dari rata-rata waktu terdapat waktu yang paling cepat bisa disimpulkan sebagai *task* yang paling mudah yaitu *guide* C1.01 dan C1.02. *Guide* C1.01 merupakan *guide* tentang pembuatan projek dan konfigurasi. *Guide* C1.02 merupakan *guide* tentang mendesain tampilan dari *MainActivity*. Penyebab pengerjaan *guide* C1.01 dan C1.02 dikarenakan mahasiswa sudah terbiasa membuat projek, melakukan konfigurasi dan membuat tampilan.

Sedangkan untuk waktu rata-rata yang paling lambat dapat disimpulkan sebagai *task* paling sulit yaitu *task* C1.05, C1.07, C1.08, C1.09 dan C1.10. *Guide* C1.05 merupakan *guide* tentang membuat *shared preferences* untuk *user data* dan *book data*. *Guide* C1.07 merupakan *guide* tentang penerapan arsitektur *Model-*

*View-Viewmodel (MVVM) dengan Data Binding. Guide C1.08, C1.09 dan C1.10* merupakan *guide* tentang membangun *source code* pada *MainActivity*, *InputDataActivity* dan *ShowDataActivity*.

### 6.2.1 Contoh Tampilan Aplikasi *MyLibrary*

Dalam pembuatan aplikasi, mahasiswa diperbolehkan memodifikasi tampilan dari aplikasi *MyLibrary*. Sehingga mahasiswa meningkatkan kreativitasnya dalam membuat aplikasi supaya lebih menarik dan mudah dipahami. Berikut adalah contoh tampilan aplikasi mahasiswa.



Gambar 6. 2 Contoh Tampilan Aplikasi *MyLibrary* Milik Mahasiswa

### 6.2.2 Umpan Balik Mahasiswa

Hasil dari umpan balik mahasiswa yang berhasil dikumpulkan. Komentar tersebut terdiri dari komentar positif dan komentar saran dari setiap task pada topik. Berikut adalah kumpulan umpan balik dari mahasiswa:

- 1 Komentar positif
  - materi pembelajaran disediakan dalam dua bahasa yang membuatnya mudah dipelajari
  - mudah dilakukan
  - mudah dimengerti
  - alhamdulillah, saya bisa menyelesaikan
- 2 Komentar saran
  - perlu ketelitian dalam membaca instruksi
  - mendapat masalah saat mendefinisikan data binding

## BAB VII. KESIMPULAN DAN SARAN

Bab ini menjelaskan tentang kesimpulan dan saran terhadap skripsi. Kesimpulan dan saran disajikan dengan cara terpisah dengan penjelasan sebagai berikut.

### 7.1 Kesimpulan

Setelah berhasil membuat topik pembelajaran *Content Provider* berdasarkan tahap analisis, perancangan serta implementasi. Untuk melakukan evaluasi tingkat efektivitas topik pembelajaran ini. Telah dilakukan pengujian kepada 50 orang mahasiswa dari Jurusan Teknologi Informasi. Berdasarkan hasil pengujian yang dilakukan selama tujuh hari. Mendapatkan hasil bahwa 49 orang mahasiswa mendapatkan hasil berhasil dan 1 orang mahasiswa mendapatkan hasil error.

Seluruh mahasiswa yang telah diuji memberikan umpan balik yang positif dan saran. Dari hasil tersebut mengkonfirmasi bahwa keefektifan topik *Basic Data Storage* pada platform APLAS sangat membantu mahasiswa.

### 7.2 Saran

Saran yang dapat membantu dalam pengembangan topik pembelajaran pada APLAS adalah sebagai berikut:

1. Dalam melakukan pengembangan topik pembelajaran selanjutnya. Diharapkan membuat sketsa kerja lebih mendalam mulai dari aplikasi, *guide* dan *test code*.
2. Disediakan fitur *plagiarism detection* untuk mendeteksi kecurangan dalam proses pengerjaan topik. Yang dapat mendeteksi dari kemiripan *source code*, ukuran *file*.

## DAFTAR PUSTAKA

Statcounter Globalstat, "Mobile Operating System Market Share Worldwide Sept 2019 - Sept 2020". <https://gs.statcounter.com/os-market-share/mobile/world-wide/#monthly-201909-202009>

Tran, C. H., Truscan, D., & Ahmad, T. (2020). Applying *Test-driven Development* in Evaluating Student Projects. *6th International Conference on Higher Education Advances (HEAD'20)*. Published. <https://doi.org/10.4995/head20.2020.11218>

T. Yogesh and P. Vimala. (2020). *Test-Driven Development of Automotive Software Functionality*. International Conference on Smart Systems and Inventive Technology (ICSSIT). <https://doi.org/10.1109/ICSSIT48917.2020.9214078>

Funabiki, N., Kusaka, R., Ishihara, N., & Kao, W. C. (2017). A Proposal of *Test code* Generation Tool for Java Programming Learning Assistant System. *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. Published. <https://doi.org/10.1109/aina.2017.60>

Syaifudin, Y. W., Funabiki, N., (2021). Introduction to *Android Programming Learning Assistance Systems*. <https://doi.org/10.13140/RG.2.2.18913.94560>

Syaifudin, Y. W., Funabiki, N., Wijaya, C. D., & Mu'aasyiqiin, I. (2021). Performance Investigation of Unit *Testing* in *Android* Programming Learning Assistance System. *Global Conference on Life Sciences and Technologies*, 2(3). <https://doi.org/10.1109/LifeTech52111.2021.9391971>

Syaifudin, Y. W., Funabiki, N., Kuribayashi, M., Mentari, M., Saputra, P. Y., Yunhasnawa, Y., & Ulfa, F. (2021). Web application implementation of *Android* programming learning assistance system and its evaluations. *IOP Conference Series: Materials Science and Engineering*, 1073(1), 012060. <https://doi.org/10.1088/1757-899x/1073/1/012060>

Syaifudin, Y. W., Funabiki, N., & Kuribayashi, M. (2020). An Implementation and Evaluation of Advanced Widgets Topic for Interactive Application *Stage* in *Android* Programming Learning Assistance System. *Proceedings of the 2020 8th International Conference on Information and Education Technology*. Published. <https://doi.org/10.1145/3395245.3396198>

Vasquez, Mario & Bernal-Cárdenas, Carlos & Moran, Kevin & Poshyvanyk, Denys. (2018). How do Developers *Test Android* Applications.

Bissi, W., Serra Seca Neto, A. G., & Emer, M. C. F. P. (2016). The effects of *test driven development* on *internal quality*, *external quality* and *productivity*: A systematic review. *Information and Software Technology*, 74, 45–54.  
<https://doi.org/10.1016/j.infsof.2016.02.004>

KOCAKOYUN, E. (2017). Developing of *Android Mobile Application Using Java and Eclipse: An Application*. *International Journal of Electronics, Mechanical and Mechatronics Engineering*, 7(1), 1335–1354.  
<https://doi.org/10.17932/iau.ijemme.21460604.2017.7/1.1335-1354>

Anandakrishnan, S. (2018). *Test-Driven Development*. *Propeller Programming*, 45–52. [https://doi.org/10.1007/978-1-4842-3354-2\\_4](https://doi.org/10.1007/978-1-4842-3354-2_4)

Syaifudin, Y. W., Funabiki, N., Kuribayashi, M., & Kao, W. C. (2021). A Proposal of Advanced Widgets Learning Topic for Interactive Application in *Android Programming Learning Assistance System*. *SN Computer Science*, 2(3).  
<https://doi.org/10.1007/s42979-021-00580-1>

Syaifudin, Y. W., Funabiki, N., Kuribayashi, M., & Kao, W. C. (2019). A proposal of *Android Programming Learning Assistant System* with implementation of basic application learning. *International Journal of Web Information Systems*, 16(1), 115–135.  
<https://doi.org/10.1108/ijwis-08-2019-0038>

Robolectric (2019), “Robolectric”, available at: <http://robolectric.org/>

JUnit (2019), “JUnit”, available at: <https://junit.org/junit4/>



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN  
RISET DAN TEKNOLOGI  
POLITEKNIK NEGERI MALANG  
JURUSAN TEKNOLOGI INFORMASI  
PROGRAM STUDI TEKNIK INFORMATIKA

JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



No. Skripsi : 472

**FORM VERIFIKASI ABSTRAK BAHASA INGGRIS DAN TATA TULIS BUKU**  
**LAPORAN SKRIPSI**

**Nama Mahasiswa** : Devany Christabella Wijaya **NIM** : 1741720092  
**Tanggal Ujian** : 30 Juli 2021  
**Judul** : Pengembangan Topik Pembelajaran *Content Provider* dengan Bahasa *Java* pada *Android Programming Learning Assistance System*

NO	BAGIAN YANG DIVERIFIKASI	NAMA VERIFIKATOR	TANGGAL VERIFIKASI	TTD
1	Abstrak Berbahasa Inggris	Atiqah Nurul Asri, S.Pd, M.Pd	28 Agustus 2021	
2	Tata Tulis Buku Laporan Skripsi	Rudy Ariyanto, ST., M.Cs.	1 September 2021	



**JUDUL SKRIPSI**

**DOSEN PEMBIMBING**

**1**  
**ttd**

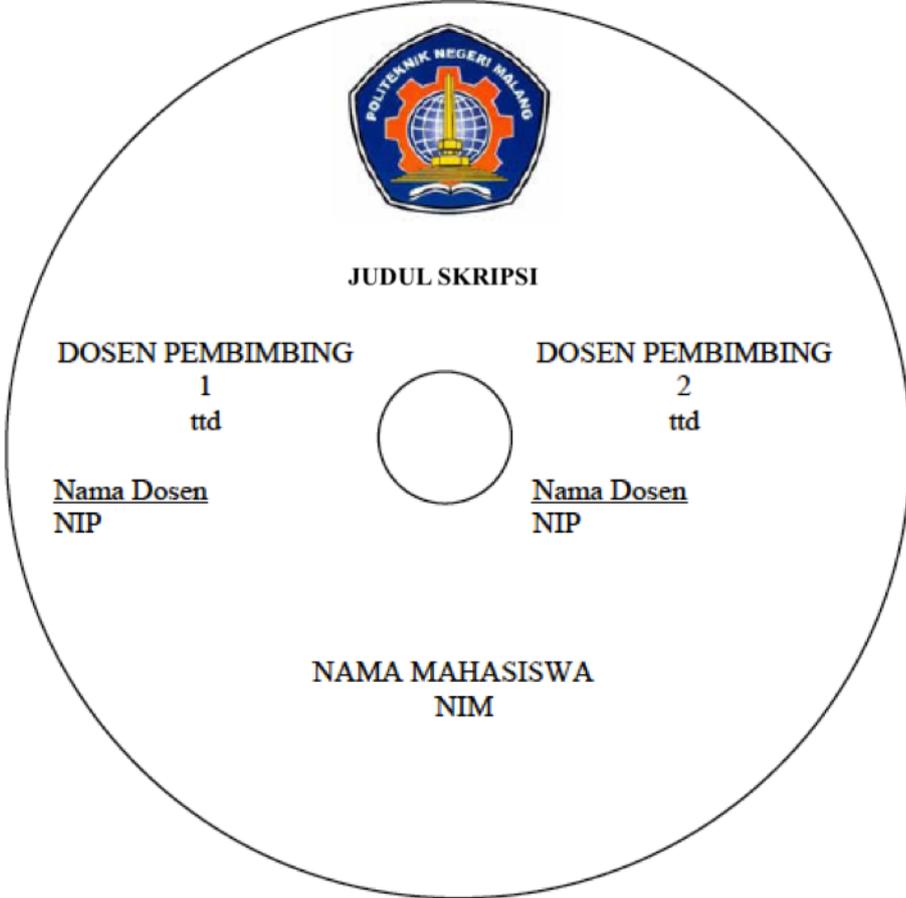
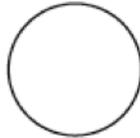
Nama Dosen  
**NIP**

**DOSEN PEMBIMBING**

**2**  
**ttd**

Nama Dosen  
**NIP**

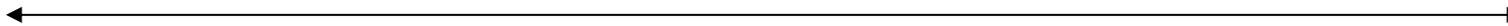
**NAMA MAHASISWA**  
**NIM**



Format Punggung Halaman Sampul (hard cover)

Sesuai Tebal Buku

<b>DEVANY CHRISTABELLA W.</b>	<b>NIM. 1741720092</b>	<b>PENGEMBANGAN TOPIK PEMBELAJARAN <i>CONTENT</i> <i>PROVIDER</i> DENGAN BAHASA <i>JAVA</i> PADA <i>ANDROID</i> <i>PROGRAMMING LEARNING ASSISTANCE SYSTEM</i></b>	<b>2021</b>
-----------------------------------	------------------------	---	-------------



Sesuai Panjang Sampul Buku



