**PENGEMBANGAN TOPIK BASIC APPLICATION UNTUK PEMBELAJARAN APLIKASI MOBILE BERBASIS FLUTTER PADA PLATFORM INTELLIGENT COMPUTER-ASSISTED PROGRAMMING LEARNING PLATFORM**

**SKRIPSI**

**Oleh:**

**AGUS SALIM HADJRIANTO NIM. 1941723007**

****

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2022**

**PENGEMBANGAN TOPIK BASIC APPLICATION UNTUK PEMBELAJARAN APLIKASI MOBILE BERBASIS FLUTTER PADA PLATFORM INTELLIGENT COMPUTER-ASSISTED PROGRAMMING LEARNING PLATFORM**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**AGUS SALIM HADJRIANTO NIM. 1941723007**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2022**

**HALAMAN PENGESAHAN**

**PENGEMBANGAN TOPIK BASIC APPLICATION UNTUK PEMBELAJARAN APLIKASI MOBILE BERBASIS FLUTTER PADA PLATFORM INTELLIGENT COMPUTER-ASSISTED PROGRAMMING LEARNING PLATFORM**

**Disusun oleh:**

**AGUS SALIM HADJRIANTO NIM. 1941723007**

**Skripsi ini telah diuji pada 18 Agustus 2022**

**Disetujui oleh:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Penguji I | : | Dhebys Suryani, S.Kom., MT  NIP. 198311092014042001 | ........................... |
| 2. | Penguji II | : | Mustika Mentari, S.Kom., M.Kom  NIP. 198806072019032016 | ........................... |
| 3. | Pembimbing I | : | Yan Watequlis Syaifudin,ST., MMT  NIP. 198101052005011005 | ........................... |
| 4. | Pembimbing II |  | Usman Nurhasan, S.Kom., MT.  NIP. 198609232015041001 | ........................... |

Mengetahui,

|  |  |
| --- | --- |
| Ketua Jurusan  Teknologi Informasi | Ketua Program Studi  Teknik Informatika |
| Rudy Ariyanto, S.T., M.CS  NIP. 19711110 199903 1 002 | 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 Perguntan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan Oleh orang Iain, kecuali yang secara tertulis disitasi dalam naskah ini serta disebutkan dalam daftar sitasi/pustaka.

Malang, 08 Agustus 2022

Agus Salim Hadjrianto

ABSTRAK

**Salim Hadjrianto., Agus**. **“Pengembangan Topik *Basic Application* untuk pembelajaran aplikasi *mobile* berbasis *flutter* pada *platform Intelligent Computer Assisted Programming Learning Platform* ”.** **Pembimbing: (1) Yan Watequlis Syaifudin S.T., M.MT. (2) Usman Nurhasan, S.Kom., MT.**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2022.**

Popularitas perangkat smartphone meningkat pesat dalam beberapa tahun terakhir dan banyak orang memanfaatkan smartphone untuk berbagai kebutuhan. Pengembangan aplikasi mobile telah ditujukan ke berbagai bidang yang membuat permintaan programmer aplikasi mobile meningkat. Baru-baru ini, *Flutter* telah menjadi perangkat pengembangan perangkat lunak untuk pengembangan aplikasi lintas platform, termasuk *Android* dan *iOS*, sehingga banyak pengembang perangkat lunak yang mengadopsinya. Untuk menyediakan sistem belajar mandiri untuk mempelajari pemrograman seluler dengan *Flutter*, kami mengusulkan sistem bantuan pembelajaran dengan fitur verifikasi kode *Dart* otomatis. Berdasarkan studi kami sebelumnya di *Android Programming Learning Assistance System* (*APLAS*), proses verifikasi kode otomatis dapat mengadopsi proses pengujian perangkat lunak untuk aplikasi *Android*. Model pembelajaran memberikan materi pembelajaran untuk dipelajari dan dipraktikkan dengan menyelesaikan suatu tugas. Sebuah topik pembelajaran mengembangkan aplikasi sederhana disiapkan untuk evaluasi sistem yang diusulkan 40 mahasiswa di departemen TI telah ditunjuk untuk mempelajari *Flutter* dan menyelesaikan tugas. Akhirnya mereka dapat menyelesaikan tugas dengan benar dan memberikan opini positif tentang penggunaan sistem ini.

**Kata Kunci:** *Android*, *iCLOP*, *State Management*, *Flutter*, *Provider, List\_Followers, self-learning system, automatic code verification, Flutter, Dart, Android, iOS*

*ABSTRACT*

**Salim Hadjrianto., Agus.** **"*Basic Applications Development for Flutter-based mobile application learning on Intelligent Computer Assisted Programming Learning Platform".*** **Supervisors: (1) Yan Watequlis Syaifudin S.T., M.MT (2) Usman Nurhasan, S.Kom., MT.**

*Thesis, Informatics Engineering Study Program, Information Technology Department, Malang State Polytechnic, 2022.*

*The popularity of smartphone devices has rapidly increased in recent years and many people utilize smartphones for various needs. The development of mobile applications has been aimed at various fields that make the demand for mobile application programmers increase. Recently, Flutter has become a software development kit for cross-platform applications devel- opment, including Android and iOS, so many software developers have adopted it. To provide a self-learning system for studying mobile programming with Flutter, we propose a learning assis- tance system with an automatic Dart code verification feature. Based on our previous studies in Android Programming Learning Assistance System (APLAS), the automatic code verification pro- cess can adopt software testing process for Android applications. The learning model provides learning materials for studying and practicing by solving an assignment. A learning topic of developing a simple application is prepared for the proposed system evaluation. 40 university students in IT department have been appointed to study Flutter and solve the assignment. Finally, they can solve the assignment correctly and give positive opinions about using this system.*

*Keywords: Android, iCLOP, State Management, Flutter, Provider, List\_Followers, self-learning system, automatic code verification, Flutter, Dart, Android, iOS*

**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 BASIC APILCATIONS UNTUK PEMBELAJARAN APLIKASI MOBILE BERBASIS FLUTTER PADA PLATFORM INTELLIGENT COMPUTER ASSISTED PROGRAMMING LEARNING PLATFORM”. 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 Usman Nurhasan, S.Kom., MT. 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, 8 Agustus 2022



Penulis

DAFTAR ISI

[ABSTRAK i](#_Toc118834885)

[ABSTRACT ii](#_Toc118834886)

[DAFTAR GAMBAR viii](#_Toc118834887)

[DAFTAR TABEL xi](#_Toc118834888)

[BAB I. PENDAHULUAN 1](#_Toc118834889)

[1.1 Latar Belakang 1](#_Toc118834890)

[1.2 Rumusan Masalah 3](#_Toc118834891)

[1.3 Batasan Masalah 3](#_Toc118834892)

[1.4 Tujuan 4](#_Toc118834893)

[1.5 Manfaat 4](#_Toc118834894)

[BAB II. LANDASAN TEORI 5](#_Toc118834895)

[2.1 Studi Literatur 5](#_Toc118834896)

[2.2 iCLOP  *(Intelligent Computer-Assisted Programming Learning Platform)* 6](#_Toc118834897)

[2.3 DART 7](#_Toc118834898)

[2.4 WIDGET 8](#_Toc118834899)

[2.5 Test Driven Development (TTD) 8](#_Toc118834900)

[2.6 Unit Test dan Widget Test 9](#_Toc118834901)

[2.7 Learning Proces for Each Task 9](#_Toc118834902)

[BAB III. METODOLOGI PENGEMBANGAN 11](#_Toc118834903)

[3.1 Metode Pengambilan Data 11](#_Toc118834904)

[3.2 Metode Pengolahan Data 11](#_Toc118834905)

[3.3 Metode Pengembangan Perangkat Lunak 12](#_Toc118834906)

[**3.3.1** **Analisa Kebutuhan Sistem** 13](#_Toc118834907)

[**3.3.2** **Desain Sistem dan Perangkat Lunak** 13](#_Toc118834908)

[**3.3.3** **Desain** 15](#_Toc118834909)

[**3.3.4** **Materi dan Pengujian Sistem** 20](#_Toc118834910)

[3.4 Metode Pengembangan 22](#_Toc118834911)

[3.5 Metode Pengujian 23](#_Toc118834912)

[BAB IV. ANALISIS DAN PERANCANGAN SISTEM 24](#_Toc118834913)

[4.1 Analisis 24](#_Toc118834914)

[**4.1.1** **Analisis Masalah** 24](#_Toc118834915)

[**4.1.2** **Analisis Kebutuhan Fungsional** 25](#_Toc118834916)

[**4.1.3** **Analisis Kebutuhan System** 25](#_Toc118834917)

[4.2 Perancangan Arsitektur Sistem 27](#_Toc118834918)

[4.3 Desain Topik 28](#_Toc118834919)

[4.4 Activity Diagram 32](#_Toc118834920)

[**4.4.1** **Activity Diagram Proses Validasi Mahasiswa Pada Android Studio** 32](#_Toc118834921)

[**4.4.2** **Activity Diagram Melihat Hasil Validasi Tugas Mahasiswa oleh Guru** 33](#_Toc118834922)

[**4.4.3** **Activity Diagram Melihat Hasil Validasi Mahasiswa di Platform iCLOP** 34](#_Toc118834923)

[BAB V. IMPLEMENTASI DAN PENGUJIAN 35](#_Toc118834924)

[5.1 Ruang Lingkup Implementasi 35](#_Toc118834925)

[5.2 Batasan Implementasi 36](#_Toc118834926)

[5.3 Implementasi Database 37](#_Toc118834927)

[5.4 Implementasi Antarmuka 44](#_Toc118834928)

[5.5 Implementasi Pembelajaran 48](#_Toc118834929)

[**5.1.1 *Guide File Basic Aplications Flutter*** 48](#_Toc118834930)

[**5.1.2 *Suplement File Basic Aplications Flutter*** 57](#_Toc118834931)

[**5.1.3 *Test File Basic Aplications Flutter*** 59](#_Toc118834932)

[5.6 Pengujian 60](#_Toc118834933)

[**5.6.1** **Performance Testing** 60](#_Toc118834934)

[BAB VI. HASIL PENGUJIAN DAN PEMBAHASAN 69](#_Toc118834935)

[6.1 Hasil Pengujian 69](#_Toc118834936)

[6.2 Pembahasan Hasil Pengujian 69](#_Toc118834937)

[6.3 Contoh Tampilan Aplikasi *List Followers* 72](#_Toc118834938)

[6.4 Umpan Balik Mahasiswa 75](#_Toc118834939)

[6.5 Pengujian Coverage 75](#_Toc118834940)

[BAB VII. KESIMPULAN DAN SARAN 77](#_Toc118834941)

[7.1 Kesimpulan 77](#_Toc118834942)

[7.2 Saran 77](#_Toc118834943)

[DAFTAR PUSTAKA 78](#_Toc118834944)

[LAMPIRAN 80](#_Toc118834945)

# DAFTAR GAMBAR

[Gambar 2. 1 Contoh materi dan soal 7](#_Toc118834818)

[Gambar 2. 2 Contoh Gambar Widgets 8](#_Toc118834819)

[Gambar 2. 3 Gambar Learning Proces for Each Task 9](#_Toc118834820)

[Gambar 3. 1 Validator System 12](#_Toc118834821)

[Gambar 3. 2 Tahapan Proses Learning 12](#_Toc118834822)

[Gambar 3. 3 Use Case iCLOP 16](#_Toc118834823)

[Gambar 3. 4 Flowchart Cara Kerja iCLOP 18](#_Toc118834824)

[Gambar 3. 5 Activity Diagram iCLOP 19](#_Toc118834825)

[Gambar 3. 6 Materi pembelajaran 20](#_Toc118834826)

[Gambar 3. 7 Agile Method 22](#_Toc118834827)

[Gambar 4. 1 feedback pengujian siswa 26](#_Toc118834828)

[Gambar 4. 2 Arsitektur System 27](#_Toc118834829)

[Gambar 4. 3 Gambaran Umum Pengerjaan Topik Basic Aplikasi Flutter 28](#_Toc118834830)

[Gambar 4. 4 Activity Diagram validasi code pada Android Studio 32](#_Toc118834831)

[Gambar 4. 5 Teacher Result 33](#_Toc118834832)

[Gambar 4. 6 Validasi pekerjaan mahasiswa 34](#_Toc118834833)

[Gambar 5. 1 Implementasi Database 37](#_Toc118834834)

[Gambar 5. 2 Perancangan tabel user 38](#_Toc118834835)

[Gambar 5. 3 Perancangan tabel file\_result 38](#_Toc118834836)

[Gambar 5. 4 Perancangan tabel learning\_files 39](#_Toc118834837)

[Gambar 5. 5 Perancangan tabel student\_submits 39](#_Toc118834838)

[Gambar 5. 6 Perancangan tabel student\_teachers 40](#_Toc118834839)

[Gambar 5. 7 Perancangan tabel student\_validations 40](#_Toc118834840)

[Gambar 5. 8 Perancangan tabel task 41](#_Toc118834841)

[Gambar 5. 9 Perancangan tabel task\_results 41](#_Toc118834842)

[Gambar 5. 10 Perancangan tabel test\_files 42](#_Toc118834843)

[Gambar 5. 11 Perancangan tabel topic 42](#_Toc118834844)

[Gambar 5. 12 Perancangan tabel topic\_files 43](#_Toc118834845)

[Gambar 5. 13 Perancangan tabel uitask\_result 43](#_Toc118834846)

[Gambar 5. 14 Perancangan tabel ui\_file\_results 44](#_Toc118834847)

[Gambar 5. 15 Halaman lending page 45](#_Toc118834848)

[Gambar 5. 16 Halaman Flutter course 45](#_Toc118834849)

[Gambar 5. 17 Halaman Download Material 46](#_Toc118834850)

[Gambar 5. 18 Halaman Submit Project 46](#_Toc118834851)

[Gambar 5. 19 Halaman submit task result 47](#_Toc118834852)

[Gambar 5. 20 Halaman submit learing file 47](#_Toc118834853)

[Gambar 5. 21 Configurasi Proyek 50](#_Toc118834854)

[Gambar 5. 22 Implementasi guide TextSpan pada dart file 51](#_Toc118834855)

[Gambar 5. 23 Implementasi Guide TextRich pada dart file 51](#_Toc118834856)

[Gambar 5. 24 Implementasi Guide TextStyle pada dart file 52](#_Toc118834857)

[Gambar 5. 25 Implementasi Guide ButtonText pada dart file 52](#_Toc118834858)

[Gambar 5. 26 Implementasi Guide Iamge pada dart file 53](#_Toc118834859)

[Gambar 5. 27 Implementasi Guide Padding & Border pada dart file 53](#_Toc118834860)

[Gambar 5. 28 Implementasi Guide ListView pada view 54](#_Toc118834861)

[Gambar 5. 29 Implementasi Guide Icons pada dart file 55](#_Toc118834862)

[Gambar 5. 30 Implementasi pada halaman home file 56](#_Toc118834863)

[Gambar 5. 31 Implementasi code state 56](#_Toc118834864)

[Gambar 5. 32 Implementasi provider pada route file 56](#_Toc118834865)

[Gambar 5. 33 Implementasi Routing pada route file 57](#_Toc118834866)

[Gambar 5. 34 Gambar pubspec.yml 58](#_Toc118834867)

[Gambar 5. 35 Gambar1.jpg 58](#_Toc118834868)

[Gambar 5. 36 Pengujian Provider pada List Item 59](#_Toc118834869)

[Gambar 5. 37 List Valunteer 1-10 60](#_Toc118834870)

[Gambar 5. 38 List Valunteer 11-23 61](#_Toc118834871)

[Gambar 5. 39 List Valunteer 23-41 61](#_Toc118834872)

[Gambar 5. 40 Testing pada laptop Asus 63](#_Toc118834873)

[Gambar 5. 41 Testing pada laptop Acer 64](#_Toc118834874)

[Gambar 5. 42 Testing pada laptop Lenovo 65](#_Toc118834875)

[Gambar 5. 43 Grafik perbandingan device 68](#_Toc118834876)

[Gambar 5. 44 Garfik perbandingan total testcode 68](#_Toc118834877)

[Gambar 6. 1 Hasil pengerjaan topik Basic Aplications Flutter 70](#_Toc118834878)

[Gambar 6. 2 Klasifikasi rata-rata pengerjaan guide 71](#_Toc118834879)

[Gambar 6. 3 Home 72](#_Toc118834880)

[Gambar 6. 4 Halaman Flutter 73](#_Toc118834881)

[Gambar 6. 5 Halaman List Followers 74](#_Toc118834882)

[Gambar 6. 6 Hasil Testing Code menggunakan LCOV Code Coverage 75](#_Toc118834883)

[Gambar 6. 7 Test Code Coverage Directory Lib 76](#_Toc118834884)

# DAFTAR TABEL

[Tabel 2. 1 Studi Literatur 5](#_Toc118834642)

[Tabel 3. 1 Deskripsi Konsep Aplikasi 13](#_Toc118834643)

[Tabel 3. 2 Tabel Use Case 16](#_Toc118834644)

[Tabel 4. 1 Tujuan Pembelajaran Basic Aplication Flutter 29](#_Toc118834645)

[Tabel 4. 2 Deskripsi Tugas Topik Basic Aplications Flutter 30](#_Toc118834646)

[Tabel 5. 3 Spesifikasi laptop 63](#_Toc118834647)

[Tabel 5. 4 Perbandingan waktu pengujian 65](#_Toc118834648)

[Tabel 5. 5 Perbandingan data testing / detik 66](#_Toc118834649)

# BAB I. PENDAHULUAN

## Latar Belakang

Ilmu pengetahuan adalah suatu sistem berbagai pengetahuan yang didapatkankan dari hasil kajian yang dilakukan dengan menggunakan suatu metode. Dengan adanya ilmu pengetahuan, teknologi dapat berkembang dengan pesat khususnya di teknologi komunikasi yang saat ini sangat di butuhkan karena sifatnya tidak terbatas yang terhalang jarak dan waktu(Sapari et al., 2021). Pendidikan akan berkembang dengan berjalanya waktu yang dulu konvensional atau tatap muka akan berubah ke model mobile learning yaitu pembelajaran dengan sistem jarak jauh, tentu saat ini model pendidikan daring menjadi cara untuk pembelajaran di saat pandemi *COVID-19* (Nasution & Nasution, 2021).

Perkembangan Smartphone telah mencapai kemajuan yang sangat besar, jumlah pengguna smartphone seiring berjalan waktu terus meningkat baik dari anak-anak bahkan sampai orang tua, karena saat menggunakan smartphone kegiatan yang dilakukan manusia lebih mudah dan cepat, Smartphone mempunyai berbagai macam sistem operasi untuk mengoperasikan aplikasi ataupun perangkat lunak sehingga aplikasi bisa berjalan dengan lancar, Menurut data yang didapatkan dari (Stat Counter , 2022) bahwa pengguna smartphone pada tahun 2019 - 2022 lebih dominan menggunakan sistem operasi dari *android* dengan *persentase* 54.37 %, *Windows* 34.84 %, *iOS* 4.37 %, OS X 3.14 %, *Uknow* 2.51 %, *Linux* 0.57 % dan pengguna sistem operasi lainnya 0,2% pada tahun 2021. Data tersebut membuktikan bahwa penggunaan sistem operasi *android* dan *iOS* sangat dibutuhkan saat ini dalam pembuatan serta pengembangan aplikasi mobile.(Prayugha, 2021)

Teknologi Informasi dan Komunikasi (TIK) telah mengalami perkembangan yang pesat, sehingga dapat menyebabkan terjadinya berbagai perubahan dari segala bidang(Septianl, 2021). Teknologi Informasi dan Komunikasi sangat berperan penting dalam mendukung dan meningkatkan efisiensi dalam mengelola kegiatan sehari-hari, serta dapat memungkinkan pekerjaan yang dilakukan darimana saja(Sirajuddin & Dahar, 2021)

Android dikembangkan menggunakan bahasa Kotlin & Java yang umumnya digunakan dalam pembuatan aplikasi pada smartphone android, bedanya jika pengembangan aplikasi mobile android dikembangkan dengan bahasa (Java & Kotlin) maka jika pengembang ingin mengembangkan lagi ke sistem operasi iOS maka developer harus mengulang code menggunakan bahasa (Swift & Objective\_c), Flutter memudahkan pengembang aplikasi mobile dalam mengembangkan aplikasi multiplatform menggunakan satu bahasa (Dart). Flutter menyediakan fitur hot\_reload yang berfungsi jika kode berubah, tampilan output juga akan otomatis berubah secara realtime, *Flutter* dikembangkan oleh *Google* yang merupakan *framework* open *source multiplatform* dengan satu basis code pemrograman yaitu bahasa *Dart*. *Flutter* menyediakan *UI* dan *Widge*t yang mudah untuk digunakan dalam membangun aplikasi multiplatform secara efisien karena dapat dipakai di berbagai platform seperti *IOS,Android,Desktop* dan *Web*.(Haryanto & Setiawan, 2021)

*iCLOP* adalah platform pembelajaran yang dikembangkan untuk menyediakan sistem cerdas dengan assistensi secara otomatis dengan menggunakan metode *TDD* yang digunakan sebagai pendukung pembelajaran *Android Flutter* dengan menggunakan bahasa *Dart* secara sistematis sesuai topik yang akan di ajarkan, Pembelajaran Basic aplikasi *UI* dan *Widget* merupakan bagian dasar dari pembelajaran *Flutter* yang menjadi salah satu topik yang akan di ajarkan dalam pengembangan aplikasi *iCLOP*.(Syaifudin, Funabiki, & Kuribayashi, 2021)

Pada Skripsi ini akan di kembangkan topik pembelajaran basic aplikasi mobile berbasis Flutter yang akan di terapkan pada *iCLOP . iCLOP* menggunakan metode *TDD* (*Test Driven Development*) yang menyediakan *Automatic Assistance* pada platform *Self Learning* yang dapat digunakan oleh mahasiswa untuk mempermudah pembelajaran mobile aplikasi berbasis *flutter . iCLOP* juga akan sangat membantu dosen dalam proses pembelajaran karena akan secara mandiri, *iCLOP* akan melakukan pengecekan tugas mahasiswa dan juga memberikan nilai sesuai test case yang sudah di tentukan oleh pengajar.

## Rumusan Masalah

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

1. Bagaimana menyediakan sebuah sistem untuk pembelajaran pemrograman aplikasi mobile berbasis  *Flutter* ?
2. Bagaimana memudahkan mahasiswa politeknik negeri malang untuk belajar pemrograman *flutter* secara mandiri dengan fungsi *automated assistance* ?

## Batasan Masalah

Agar skripsi yang berjudul Pengembangan Topik Basic Application Untuk Pembelajaran Aplikasi *Mobile* Berbasis *Flutter* Pada *Platform iCLOP* dapat berjalan sesuai rencana dan tujuan awal, maka penulis menetapkan batasan-batasan masalah yaitu:

1. Topic pembelajaran fokus pada *basic application : UI & Widgets*
2. Sistem akan diujikan pada mahasiwa perguruan tinggi.
3. Menggunakan *Android Studio* sebagai media pengerjaan.
4. Konten pembelajaran diadopsi dari situs *Udemy* atau *Udacity.*

## Tujuan

Tujuan dari pembuatan sistem ini adalah sebagai berikut :

* + - 1. Menyediakan mahasiswa sebuah sistem yang dapat membantu belajar pemrograman Mobile secara mandiri khususnya pada topik *UI* dan *Widgets* pada *Flutter..*
      2. Mengintegrasikan topik pembelajaran Flutter pada platform *iCLOP* sehingga dapat diakses oleh mahasiswa politeknik negeri malang untuk mendukung proses PBM khususnya pada mata kuliah Pemrograman *Mobile.*

## Manfaat

Manfaat dari pengembangan sistem pembelejaran ini antara lain :

1. Membantu mahasiswa dalam belajar *Mobile Aplication* berbasis *flutter* secara mandiri
2. Membantu dosen dalam proses pembelajaran pemrograman *mobile*

# BAB II. LANDASAN TEORI

## Studi Literatur

Studi Litelatur menjadi sebuah acuan dalam melakukan penelitian baru, dari studi literature penulis tidak akan menemukan judul yang identik sama dengan judul lainya, serta dapat memperkaya wawasan dan refrensi dalam memperkaya kajian peda penenelitian yang akan dilakukan, Berikut merupakan Studi Literature terdahulu yang berkaitan dengan beberapa jurnal yang terkait yang akan dilakukan.

Tabel 2. 1 Studi Literatur

|  |  |  |
| --- | --- | --- |
| **Nama Peneliti** | **Judul Penelitian** | **Hasil Penelitian** |
| (Syaifudin et al., 2019) | A Proposal of Android Programming Learning Assistant System with Implementation of Basic Application Learning | Tujuan dari dibuatnya APLAS untuk membantu mahasiswa belajar Android dasar dengan mudah dan membantu dosen memeriksa tugas mahasiswa |
| (Syaifudin, Funabiki, & Kuribayashi, 2021) | Implementation and Performance Evaluation of Unit Testing for Student’s Answer Validation in Android Programming Learning Assistance System | Menunjukkan bahawa spesifikasi perangkat keras dan proses inisialisasi oleh Gradle memiliki dampak besar pada waktu eksekusi. |
| (Syaifudin, Funabiki, Kuribayashi, et al., 2021) | Web application implementation of Android programming learning assistance system and its evaluations | Implementasi web platform untuk mengakomodasi siswa yang belajar pemrograman Android secara mandiri. |

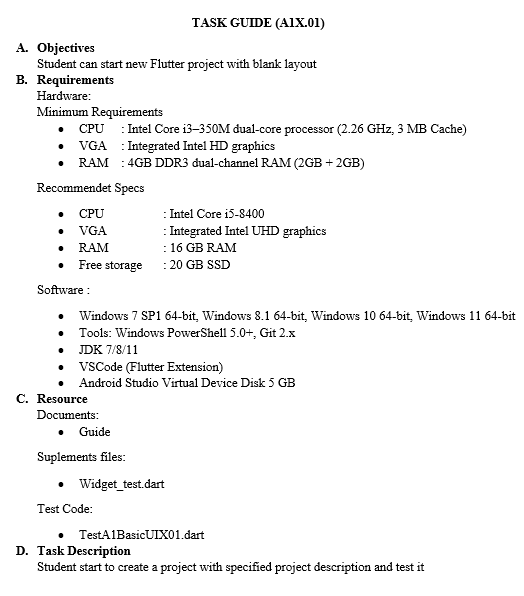
## iCLOP *(Intelligent Computer-Assisted Programming Learning Platform)*

iCLOP adalah suatu aplikasi yg dapat digunakan dalam proses pembelajaran antara dosen dan mahasiswa untuk mempermudah dosen dalam pemberian dan pengecekan tugas serta mempermudah mahasiswa untuk mengakses pembelajaran mandiri dan pengumpulan tugas yang telah diberikan oleh dosen. iCLOP juga membantu pembelajaran Android sederhana dan dasar sehingga mudah dipahami oleh pengajar dan mahasiswa, iCLOP di kembangkan dengan mengadopsi sitem dari APLAS (*Android Programming Learning Assistant System*).(Syaifudin et al., 2019)

iCLOP adalah platform pembelajaran yang dikembangkan untuk menyediakan sistem cerdas dengan assistensi secara otomatis (*Automatic Assistance*) yang mana system ini akan mempermudah dalam proses pembelajaran secara mandiri, iCLOP mempunyai beberapa topik di antaranya *User Interface, Interactive Application, Content Provider dan Service Interactive* yang masing masing topik mempunya sub topik yang meliputi :

1. *User Interface* memiliki sub topic *UI* dan *Widgets*
2. *Interactive Aplication* memiliki beberapa sub topik meliputi Basic *Activity,Advance Widgets* dan *Multiple Activities.*
3. *Content provider* yang mempunyai beberapa sub topik di antaranya Data Storage, SQLite, Network Connection, Data Service
4. *Service Interaction* memiliki beberapa sub bab di antaranya Async Task*, Web Contents, Service Application, Notification.*

*iCLOP*  mempunyai spesifikasi (*Guide file*) yang menjadi syarat minimum dalam pengerjaan materi, berikut langkah langkah pembuatanya :



Gambar 2. 1 Contoh materi dan soal

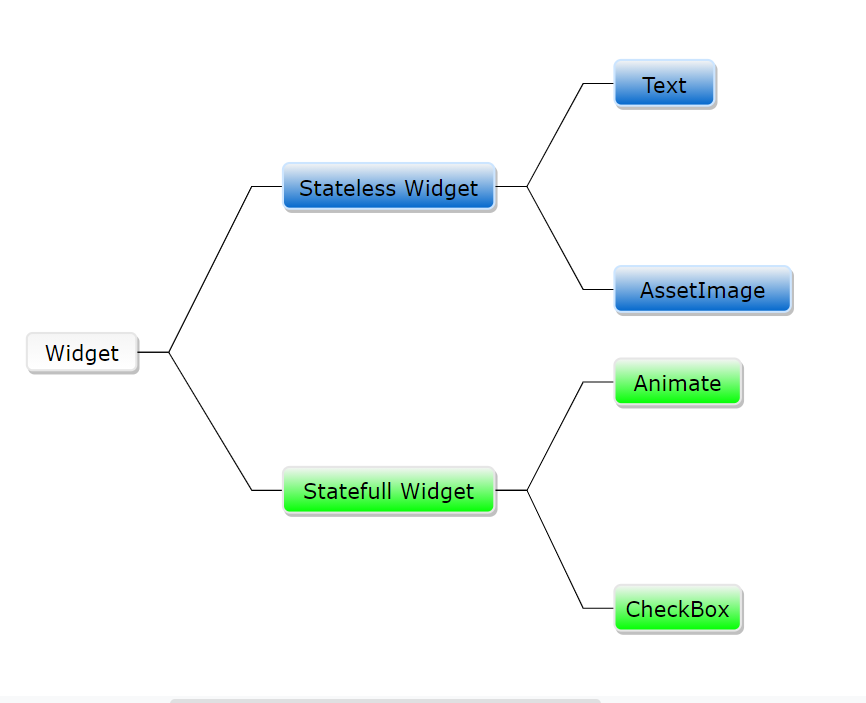
Sumber yang disediakan mempunyai 3 file berupa dokumen panduan pengerjaan (*Guide file*), *Suplement file* berupa elemen yang diperlukan aplikasi seperti gambar, *Test code* berupa file yang akan di jalankan pada Flutter Test.

## DART

Bahasa dart di kembangkan oleh google untuk membuat aplikasi multiflatform dengan satu basis code. Dengan bahasa dart dapat dikembangkan dan digunakan untuk membuat aplikasi untuk *iOS, android,* dan *web* dengan fitur yang ditujukan untuk pengembangan aplikasi *multiflatfrom* berbasis *flutter*. (Purba, 2021)

## WIDGET

Widgets merupakan bagian dari kerangka kerja *flutter* yang mana *widget* pada *flutter* dapat duganakan dalam merancang atau mengkategorikan tampilan dari kerangka aplikasi, setiap widget memiliki deklarasi dalam pembangunan tampilan yang tidak dapat diubah. *Flutter* memiliki struktur *widget* yang dapat disesuaikan dan diperluas.(Haris, 2021)



Gambar 2. 2 Contoh Gambar Widgets

## Test Driven Development (TTD)

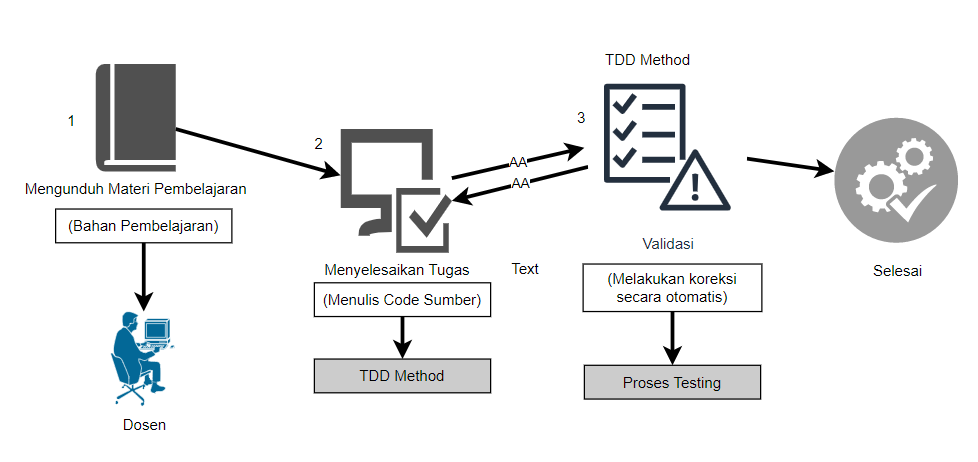
Metode TDD dapat diimplementasikan dengan cara pengujian aplikasi Android yang dilakukan melalui proses pengujian yang otomatis. Dalam proses pengujian ini menggunakan *source code* atau model desain untuk menjadi patokan dari sebuah pengujian yang dilakukan secara otomatis. TDD melakukan pengujian secara *Automated Assistance* mulai bagian yang kecil hingga besar. Bagian tersebut terdiri dari kecil (*unit test*), sedang (*widget test*) (Syaifudin et al., 2019)

## Unit Test dan Widget Test

Unit test merupakan pengujian suatu fungsi, metode dan kelas, tujuan dari unit test yaitu untuk menguji dan memverifikasi kebenaran dari sebuah logika dalam berbagai kondisi, Unit test menguji sebuah fungsi secara umum tanpa membaca atau menulis serta tindakan dari luar proses pengujian.

Widget Test merupakan pengujian komponen yang memverivikasi bahwa UI dan Widget mempunyai interaksi yang baik, serta dapat melalukan tata letak dan tindakan . Tes widget lebih komprehensif daripada tes unit. Namun, seperti tes unit, lingkungan tes widget diganti dengan implementasi yang jauh lebih sederhana daripada sistem UI penuh.(Satria et al., 2021)

## Learning Proces for Each Task



Gambar 2. 3 Gambar Learning Proces for Each Task

*Learning process for each task* merupakan proses mahasiswa belajar basic aplication menggunakan *flutter* melalui tiga proses sesuai pada gambar 2.3 (Syaifudin, Funabiki, Kuribayashi, et al., 2021). yang meliputi :

1. Proses pertama mahasiswa mengunduh materi pada platform iCLOP yang meliputi GuideDocument,Suplement dan Test File.
2. Proses kedua mahasiswa mengerjakan dan menulis code program pada android studio sesuai intruksi dari modul secara berurutan dan dapat memastikan bahwa code berjalan dengan benar,
3. Proses ketiga setelah itu code (test) akan di verifikasi pada proses testing menggunakan TDD method untuk memvalidasi *source code* menggunakan automated assistance jika *failed* maka mahasiswa harus mengerjakan ulang *guide document* secara benar dan jika *passed* maka proses selesai *tervalidasi* lulus.

# BAB III. METODOLOGI PENGEMBANGAN

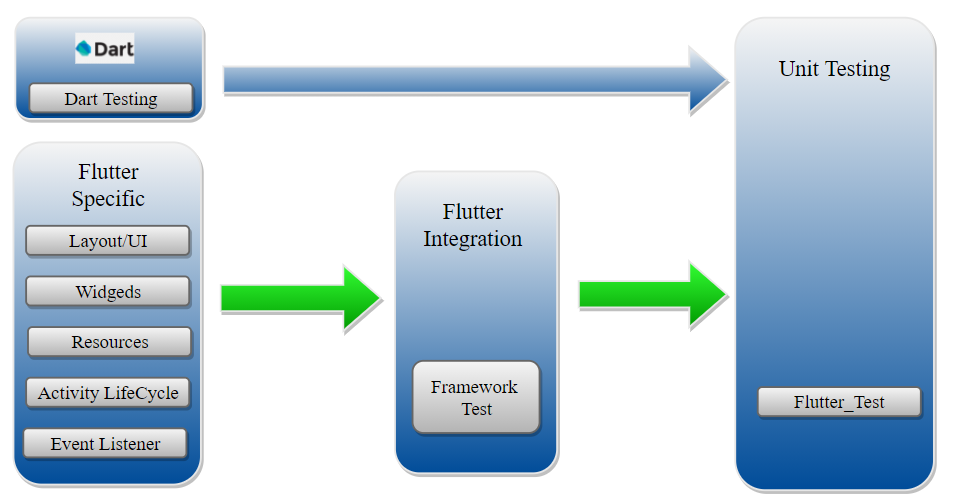
Metodelogi pengembangan menjelaskan tentang metode yang digunakan dalam pengembangan iCLOP diantaranya : metode pengambilan data, metode pengolahan data, metode pengembangan perangkat lunak dan metode pengujian

* 1. **Metode Pengambilan Data**

Metode pengambilan data pada topic Basic aplication untuk pembelajaran flutter pada platform iCLOP di ambil dari *learning* yang telah di kerjakan dan di unggah oleh mahasiswa yang meliputi : Gambar dan *Test Case* dimana data tersebut akan digunakan untuk feedback pada platform iCLOP.

* 1. **Metode Pengolahan Data**

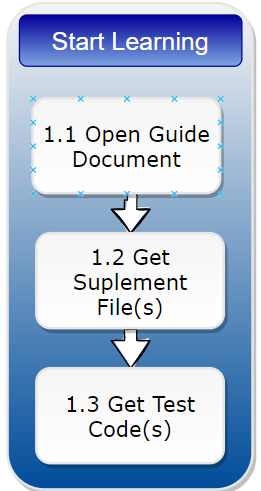
Admin akan melakukan konfigurasi terhadap data aplikasi dan materi pembelajaran. Admin melakukan konfigurasi dengan tujuan untuk mencegah terjadinya kesalahan pada pe ngunggahan data. Lalu mahasiswa mengunduh materi pembelajaran, kemudian mahasiswa mengerjakan tugas / project dari materi pembelajaran yang sudah tersedia. Setelah itu mahasiswa memasukkan hasil belajarnya. Jika sudah memasukkan hasil pembelajaran lakukan validasi untuk mendapatkan dan pengecekan hasil pembelajaran mahasiswa. Setelah itu lakukan validasi dengan flutter test.(Syaifudin et al., 2019)



Gambar 3. 1 Validator System

* 1. **Metode Pengembangan Perangkat Lunak**

Proses pengembangan basic aplication pembelajaran flutter memerlukan 3 tahap yaitu, *Guide File, Supplement File* dan *Test Code*. *Guide File* merupakan file yang berisi langkah-langkah pengerjaan, *Supplement* File merupakan file pendukung yang dibutuhkan pada saat pengerjaan. Sedangkan untuk *Test Code* merupakan file yang digunakan untuk melakukan test pada project yang dikerjakan.



Gambar 3. 2 Tahapan Proses Learning

### **Analisa Kebutuhan Sistem**

Pada tahap ini akan dilakukan analisis sesuai dengan kebutuhan sistem yang mencakup pengambilan data yang akan dibutuhkan sistem, kemudian sebagai pendukung / penunjang pengolahan data maupun penunjang pengerjaan. Berikut metode pengambilan data yang dibutuhkan sebagai pendukung dalam proses pengembangan perangkat lunak dengan studi literatur:

1. Pengumpulan Data
2. Studi Pustaka
3. Observasi dari Internet
   * 1. **Desain Sistem dan Perangkat Lunak**

Sesuai dengan latar belakang di atas maka Pengembangan Topik Basic Application Untuk Pembelajaran Aplikasi Mobile Berbasis Flutter Pada Platform iCLOP yang akan digunakan dalam membantu dalam proses belajar mandiri maka adapun pengujian seperti pada tabel 3.1 Deskripsi konsep aplikasi.

Tabel 3. 1 Deskripsi Konsep Aplikasi

|  |  |
| --- | --- |
| Judul | Pengembangan Topik Basic Application Untuk Pembelajaran Aplikasi Mobile Berbasis Flutter Pada Platform iCLOP |
| Jenis  Aplikasi | Aplikasi untuk mempermudah pembelajaran mobile untuk mahasiswa secara mandiri |
| Pengguna | Pengguna aplikasi adalah mahasiswa dan dosen |
| Konten | Konten pembelajaran berupa topik basic aplication berbasis flutter pada platform iCLOP |
| Aplikasi | Aplikasi ini berbasis mobile multiplatform yang mana bisa berupa android, IOS, web dan desktop |
| Teknologi | Aplikasi mobile menggunakan bahasa Dart |

Berdasarkan tabel 3.1 tentang deskripsi konsep aplikasi maka kebutuhan aplikasi yang akan dibuat dapat dikategorikan dalam kebutuhan perangkat lunak dan kebutuhan perangkat keras. Adalah sebagi berikut.

1. Kebutuhan Perangkat Lunak

Jenis perangkat lunak atau tool yang digunakan untuk membantu proses pengembangan topik Basic aplication untuk pembelajaran aplikasi mobile berbasis flutter pada platform iCLOP sebagai berikut.

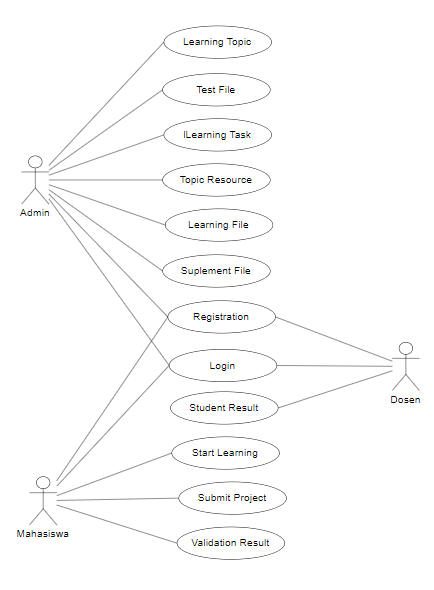
* + Sistem operasi Linux, Windows dan MacOs.
  + Microsoft Office sebagai alat bantu untuk penulisan proposal skripsi selama pengembangan sistem.
  + Bahasa pemrograman Java dan Dart
  + Aplikasi Android Studio sebagai alat dalam proses pengkodean dalam membangun aplikasi iCLOP.

Kebutuhan Perangkat Keras,Adapun perangkat keras yang terlibat pada proses pengembangan sistem adalah sebagai berikut :

* + Komputer / Laptop yang digunakan untuk membuat aplikasi mobile learning berbasis android, spesifikasi yang digunakan pada pembuatan aplikasi android ini adalah:
  + Prosesor: Intel Core i5 2.8GHz
  + Memori: 4GB DDR4
  + Prosesor: Qualcom Snapdragon 665 2.0 GHz
  + Memori: 4GB
    1. **Desain**

Pada tahap ini dibutuhkan sistem sebuah perancangan sebelum mengimplementasikan ke dalam bentuk *source code.* Tahap ini berfokus pada struktur data, arsitektur data, arsitektur perangkat lunak, dan representasi *interface.*

1. *Use Case*



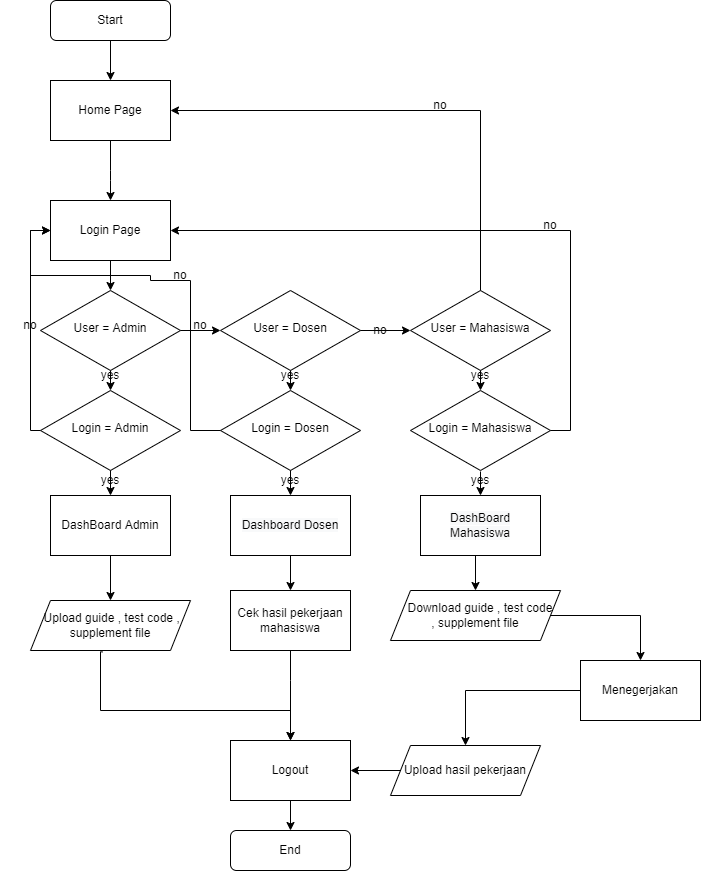
Gambar 3. 3 Use Case iCLOP

Use Case digunakan untuk mendeskripsikan model interaksi sistem yang akan di jelaskan pada tabel 3.2

Tabel 3. 2 Tabel Use Case

|  |  |
| --- | --- |
| Aktor | Deskripsi |
| Admin | Admin adalah orang yang bertanggung jawab dalam semua urusan yang berhubungan dengan server website iCLOP. Mengelola file-file pembelajaran yang terdiri dari test code, guide dan suplemen file. |
| Dosen | Dosen adalah pengguna yang bertanggung jawab dalam pengecekan hasil pekerjaan setiap mahasiswa. |
| Mahasiswa | Mahasiswa adalah pengguna yang bertanggung jawab dalam pengerjaan guide document. |

1. Flowchart Cara Kerja iCLOP



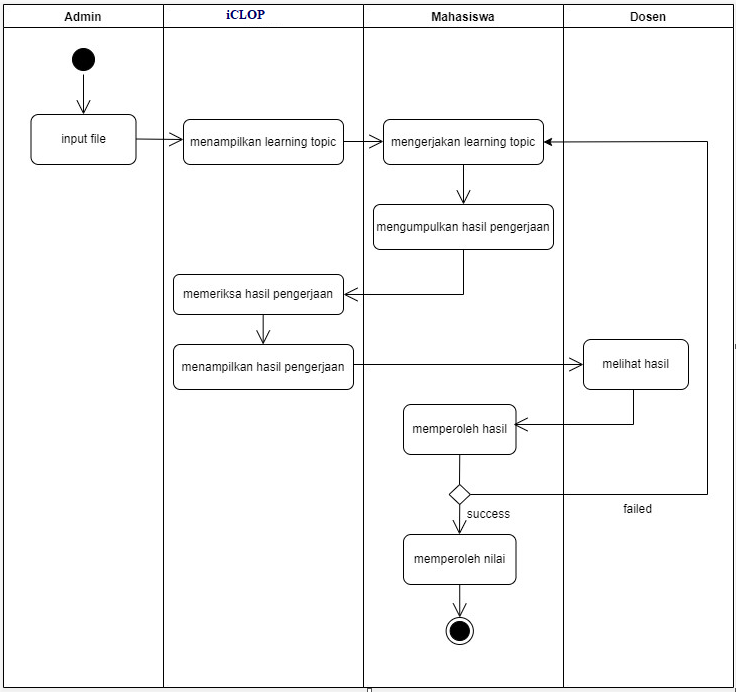
Gambar 3. 4 Flowchart Cara Kerja iCLOP

.

Pada perancangan iCLOP, pertama akan muncul home page, setelah itu pilih user sebagai admin, dosen, dan mahasiswa. Kemudian melakukan login page. Jika berhasil login sebagai admin, maka akan memasuki dashboard admin. Sebagai admin kita dapat mengunggah file yang berupa test code, guide, dan supplement file.

Jika kita berhasil login sebagai mahasiswa maka akan diarahkan memasuki dashboard mahasiswa, lalu dapat mendownload file yang telah diunggah oleh admin. Setelah itu mahasiswa mengerjakan sesuai dengan panduan yang ada di guide. Jika semua pengerjaan telah selesai maka lanjutkan dengan mengunggah hasil pekerjaan sesuai yang diperlukan untuk menyelesaikan per bagian. Jika gagal dalam pengerjaan, maka mahasiswa dianjurkan untuk mengerjakan ulang dan jika berhasil maka hasil akan diterima oleh dosen dan nilai mahasiswa akan sampai kepada mahasiswa.

1. Activity Diagram



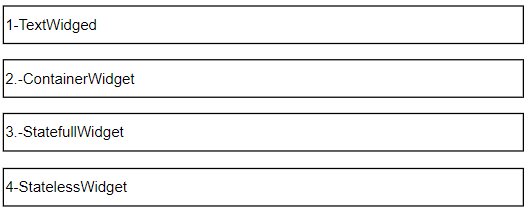
Gambar 3. 5 Activity Diagram iCLOP

Gambar 3.5 diatas merupakan *activity diagram* dari sistem iCLOP yang akan dibuat. Activity Diagram adalah bentuk visual dari alur kerja yang berisi aktivitas dan tindakan, pilihan, atau pengulangan Dalam Unified Modelling Language (UML), diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun aktivitas dari organisasi serta tujuan utama dari activity diagram adalah untuk menggambarkan urutan aktivitas sebuah proses agar lebih mudah dimengerti.

Pada gambar 3.5 menjelaskan activity diagram system yaitu admin menginputkan text setelah itu iCLOP menampilkan learning topic serta memeriksa hasil pekerjaan dari mahasiswa apakah sudah sesuai role dalam modul pembelajaran .

* + 1. **Materi dan Pengujian Sistem**

Materi yang akan di implementasikan pada pembelajaran basic aplication ui dan widget berbasis flutter pada platform iCLOP seperti pada gambar 3.6 dan akan di jelaskan pada tebel 3.3



Gambar 3. 6 Materi pembelajaran

Tabel 3. 3 Materi pembelajaran

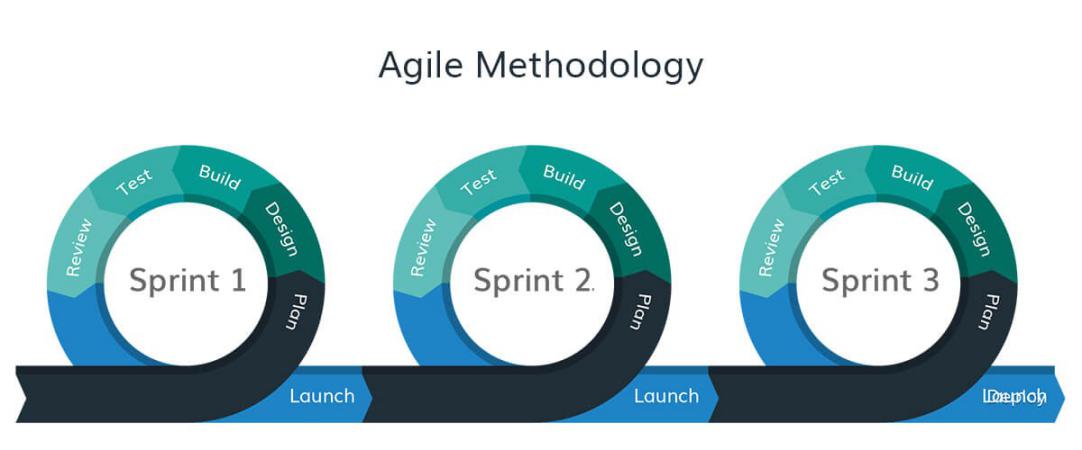
|  |  |
| --- | --- |
| TextWidgets | Siswa mampu mempelajari macam-macam jenis text pada widgets seperti TextSpan, TextRich, TextStyle, dan ButtonText. Widgets digunakan untuk menampilkan text kedalam layar Text Widgets dibagi menjadi dua properties yaitu : String yang merupakan data dan nantinya akan ditampilkan kelayar, lalu properti kedua merupakan argument berbentuk Object yang dimana berfungsi untuk memanipulasi tampilan seperti warna, ukuran, posisi. |
| ContainerWidgets | Siswa mampu memahami penggunaan ContainerWidgets meliputi : Padding, dan Border yang digunakan untuk mengatur posisi dan merancang struktur tampilan pada view |
| StatefullWidgets | Siswa mampu memahami konsep StatefullWidgets dimana Pembelajaran Widgets yang dapat berubah ubah, StateFull widgets yang dapat mengupdate tampilan,merubah warna, dan menambah jumlah baris baru. |
| StatelessWidgets | Siswa mampu memahami konsep widgets statis yang tidak dapat dirubah atau tidak akan pernah berubah karena Stateless hanya digunakan untuk mengubah tampilan seperti icon,warna dan boxDecoration. |

Pengujian sistem dilakukan untuk mevalidasi program sehingga sesuai tujuan awal pembuatan program adapun tahap yang diterapkan

Pengujian ini menggunakan framework dari flutter atau disebut flutter test yang mana digunakan untuk memvalidasi fungsi dari program yang meliputi UI\_Testing dan Widget\_Testing.

* 1. **Metode Pengembangan**

Tahapan sistem yang digunakan dalam pengembangan ini adalah Agile Software Development. Agile software development merupakan cara yang dilakukan dalam pengembangan perangkat lunak dengan sifat yang lincah dan cepat.



Gambar 3. 7 Agile Method

1. Meet and Plant

Meet and Plant merupakan tahapan paling awal dari Agile Software Development. Tahapan ini tentang bagaimana dilakukan pertemuan dan perencanaan dalam mengembangkan aplikasi. Perencanaan dilakukan agar pengembangan sesuai dengan kebutuhan.

1. Design

Design merupakan tahapan yang digunakan untuk membuat design dari sebuah software. Dengan cara membuat mockup dari tampilan software hingga membuat UI dari software.

1. Build and Test

Build and Test merupakan tahapan yang paling banyak memakan waktu. Tahapan code ini digunakan untuk programmer melakukan pemrograman secara backend, sedangkan test pada software ini merupakan test code yang digunakan pengecekan code apakah sudah sesuai dengan test yang disediakan. Test ini juga digunakan untuk melakukan uji coba pada aplikasi yang akan di release.

1. Feedback

Feedback merupakan tahapan dimana terakhir yaitu menerima masukan dari orang lain yang telah melakukan penggunaan software, sehingga kita dapat memperbaiki software yang kita buat lebih baik lagi.

1. Release

Release merupakan tahap publikasi yang dilakukan developer. Release ini digunakan untuk memperkenalkan produk yang dibuat agar dikenal oleh orang lain.

* 1. **Metode Pengujian**

Metode pengujian dari sistem iCLOP dapat dilakukan menggunakan file guide yang tersedia dengan bantuan supplement file. Setelah itu melakukan pengerjaan per guide maka akan dilakukan pengujian dengan test code yang sudah di validasi oleh TDD sehingga memperoleh hasil. Jika memperoleh hasil *failed* maka perlu dilakukan perbaikan pekerjaan yang telah dikerjakan sebelumnya pada guide tersebut. Jika *success* maka bisa melanjutkan ke guide selanjutnya. Setelah itu melakukan upload sesuai dengan yang dibutuhkan sistem serta menunggu hasilnya.

# BAB IV. ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem pada topik *Basic Aplication Flutter* pada iCLOP,Proses perancangan akan digunakan sebagai analisa kebutuhan sistem.

## Analisis

Analisis merupakan langkah untuk memahami suatu masalah sebelum mengambil keputusan atau mengambil suatu tindakan.. Sementara itu, fase desain adalah membuat suatu rincian sistem hasil atau analisis menjadi bentuk perancangan agar dimengerti oleh pengguna.

### **Analisis Masalah**

* 1. **Analisis Sistem yang Sedang Berjalan**
     1. **Identifikasi Permasalahan**

Permasalahan yang dihadapi pada sistem yang berjalan saat ini diantaranya adalah kurangnya pembelajaran android berbasis flutter terkait bahasa pemrograman dart. Aplikasi yang pernah dibuat hanya memiliki pembelajaran terkait android yang meliputi bahasa kotlin dan java, selain itu penilaian beberapa tugas masih manual, sehingga hubungan antara mahasiswa dengan dosen terkait pembelajaran Android Flutter kurang efisien.

* + 1. **Analisis sistem yang Diusulkan**

Penerapan dari fungsi Flutter test dalam sistem pembelajaran Android Flutter ini untuk mempermudah pengguna / mahasiswa dalam belajar bahasa pemrograman dart, Pembuatan sistem pembelajaran ini dilakukan menggunakan UnitTest dan WidgetTest agar dapat membantu mahasiswa melakukan validasi sesuai rquirtment sistem.

### **Analisis Kebutuhan Fungsional**

Analisis kebutuhan fungsional berisikan proses yang dilakukan oleh sistem , alam hal ini pengguna dapat menggunakan fitur pada aplikasi antara lain :

* Aplikasi dapat menmpilkan modul - modul pembelajaran
* Aplikasi dapat menampilkan download modul
* Aplikasi dapat menampilkan Uploud tugas
* Aplikasi dapat menampilkan Uploud source code

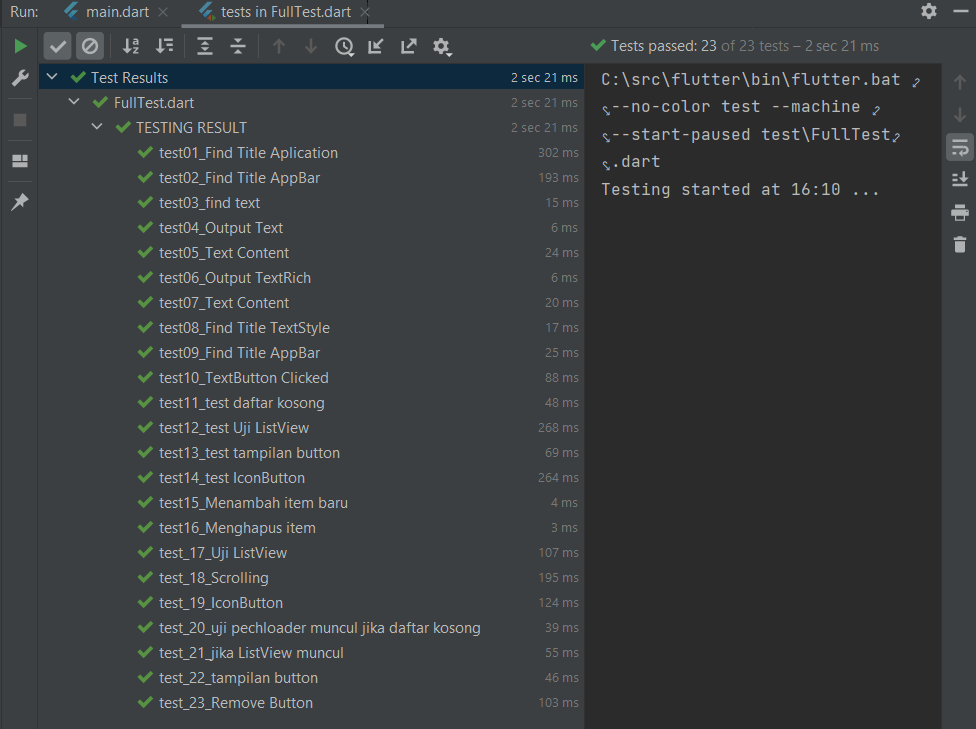
### **Analisis Kebutuhan System**

Pengelolaan aplikasi pada flutter menggunakan *StateManagement* yang memudahkan mahasiswa dalam pengembangan UI,

Basic Aplikasi Flutter yang akan dibangun pada iCLOP ini untuk memudahkan pemrogaman android mandiri khsusnya dalam stateManagement UI, agar dapat megambil topik terkait pada iCLOP. Mahasiswa diharuskan untuk mendaftarkan akun pada iCLOP dengan mengisi nama, email, password, confirm password, dan memilih dosen sesuai dengan dosen mata kuliah pembelajaran. Pendaftaran bisa dilakukan pada web iCLOP.

Setelah proses pendaftaran dilakukan sudah bisa digunakan untuk melakukan download materi pembelajaran. Ketika mahasiswa akan mengumpulkan tugas mahasiswa harus melakukan validasi testcode yang tersedia pada fitur testfile apabila susuai dengan tescode maka mahasiswa dapat melakukan submit project.

Berikut feedback ketika mahasiswa melakukan test pada suatu kasus, seperti gambar berikut:



Gambar 4. 1 feedback pengujian siswa

Pada gambar diatas menunjukkan feedback yang dari hasil pengujian, untuk setting layout pada android. Hasil pengujian yang berhasil (passed) akan ditunjukkan dengan warna hijau dengan tanda *test* (  ) dan hasil pengujian yang salah (*Failed*) akan ditunjukkan oleh warna merah dengan tanda *x-mark* (  ).

## Perancangan Arsitektur Sistem

Sistem dikembangkan berbasis website sehingga dapat di akses menggunakan web Browser. Diagram arsitektur sistem ditunjukkan pada gambar 4.2



Gambar 4. 2 Arsitektur System

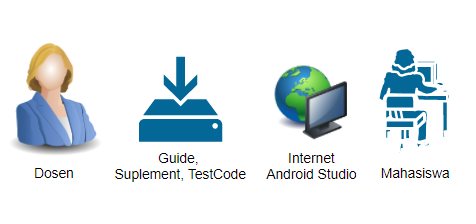
Akses sistem dapat dilakukan menggunakan laptop maupun gawai yang terhubung dengan koneksi internet. Sitem di akses sesuai *role* . Admin dapat mengelola data (user,file,materi) yang akan di akses oleh dosen dan mahasiswa, Server terhubung ke Internet dan mengirimkan respons atas permintaan yang dikirim oleh pengguna sisi klien. Respon yang dikirimkan dapat berupa data yang tersimpan di database server atau hasil pengolahan data.

Semua jenis perangkat lunak yang digunakan untuk membantu proses pembuatan aplikasi android pada flutter sebagai berikut:

* + Sistem operasi windows, linux dan chromeOS
  + Microsoft Office, sebagai alat bantu dalam pembuatan laporan dalam tahap pengembangan.
  + JDK, perangkat lunak yang digunakan untuk pengembangan bahasa pemrograman Java.
  + Flutter SDK, digunakan untuk proses running aplikasi
  + Flutter Plugins, sebagai pendukung dalam proses pengkode-an
  + Dart Plugins, sebagai pendukung dalam proses pengkode-an

## Desain Topik

Topik *Basic Aplications Flutter*  menjelaskan topik pembelajaran Basic UI dan WIDGET pada flutter, proses pembelajaran dilakukan secara bertahap meliputi : TextStyle, ContainerWidgets, Statefull dan Stateless widgets , mahasiswa akan belajar mulai basik pembuatan UI dan Widgets sampai saplikasi List Followers bisa untuk dijalankan sesuai komponen pembelajaran yang akan di sajikan.



Gambar 4. 3 Gambaran Umum Pengerjaan Topik Basic Aplikasi Flutter

Untuk mengerjakan Topic Basic Aplikasi Flutter, mahasiswa harus memiliki koneksi internet untuk akses ke website iCLOP menggunakan web browser. Untuk mahasiswa yang belum memiliki akun maka diharuskan untuk registrasi terlebih dahulu. Berikutnya mahasiswa melakukan *login* dengan menggunakan email dan password yang telah didaftarkan. Kemudian mahasiswa memli icon android dan memilih topic online database kemudian mahasiswa mendownload *file* tugas yang terdiri dari *guide file, test file, supplement file.*

Mahasiswa kemudian menggunakan Android Studio dan PDF reader untuk membuka file guide. Kemudian jalankan file tes di Android Studio dan mahasiswa akan melihat hasil tes. Jika hasilnya masih gagal, mahasiswa harus memperbaikinya. Jika hasil yang diperoleh berhasil, mahasiswa dapat mengunggah hasil tugasnya ke menu Submit Project pada website iCLOP Saat mengunggah tugas, pastikan topik yang dipilih adalah topik Online Database.

Validator akan melakukan validasi dengan menggunakan metode Test Driven Development (TDD). Selain itu, hasil validasi disimpan dalam database di server dan ditampilkan di website iCLOP untuk menampilkan hasil pekerjaan siswa. Hasilnya terdiri dari sukses, failed, dan gagal. Tujuan mempelajari topik Basik Aplikasi Flutter adalah untuk:

Tabel 4. 1 Tujuan Pembelajaran Basic Aplication Flutter

|  |  |  |
| --- | --- | --- |
| **NO** | **Tujuan** | **Deskripsi** |
| LO1 | TextWidgets | Siswa mampu mempelajari macam-macam jenis text pada widgets seperti TextSpan, TextRich, TextStyle, dan ButtonText. Widgets digunakan untuk menampilkan text kedalam layar Text Widgets dibagi menjadi dua properties yaitu : String yang merupakan data dan nantinya akan ditampilkan kelayar, lalu properti kedua merupakan argument berbentuk Object yang dimana berfungsi untuk memanipulasi tampilan seperti warna, ukuran, posisi. |

|  |  |  |
| --- | --- | --- |
| LO2 | ContainerWidgets | Siswa mampu memahami penggunaan ContainerWidgets meliputi : Padding, dan Border yang digunakan untuk mengatur posisi dan merancang struktur tampilan pada view |
| LO3 | StatefullWidgets | Siswa mampu memahami konsep StatefullWidgets dimana Pembelajaran Widgets yang dapat berubah ubah, StateFull widgets yang dapat mengupdate tampilan,merubah warna, dan menambah jumlah baris baru. |
| LO4 | StatelessWidgets | Siswa mampu memahami konsep widgets statis yang tidak dapat dirubah atau tidak akan pernah berubah karena Stateless hanya digunakan untuk mengubah tampilan seperti icon,warna dan boxDecoration. |

Dari empat tujuan topik pembelajaran *Basic Aplications Flutter* yang memenuhi dalam tigabelas dari guide file terdiri sebagai berikut :

Tabel 4. 2 Deskripsi Tugas Topik Basic Aplications Flutter

|  |  |  |
| --- | --- | --- |
| **NO** | **Deskripsi Guide** | **Learning Objective** |
| 1. | Starting Project, Configure Project and Resource | LO1,LO4 |
| 2. | Desain UI & Widgets TextSpan | LO1,LO3,LO4 |
| 3. | Desain UI & Widgets Text.Rich | LO1,LO3,LO4 |
| 4. | Desain UI & Widgets TextStyle | LO1,LO3,LO4 |
| 5. | Desain UI & Widgets ButtonText | LO1,LO2,LO3,LO4 |
| 6. | Desain UI & Widgets Image | LO3,LO4 |
| 7. | Desain UI & Widgets Padding, Border | LO2,LO3,LO4 |
| 8. | Desain UI & Widgets ListView, Development Models | LO2,LO3,LO4 |
| 9. | Desain UI & Widgets Icon | LO3,LO4 |
| 10. | Desain UI & Widgets Notifications Snackbar | LO4 |
| 11. | Desain UI & Widgets State Management | LO3,LO4 |
| 12. | Desain UI & Widgets Provider | LO3,LO4 |
| 13. | Desain UI & Widgets Route | LO3,LO4 |

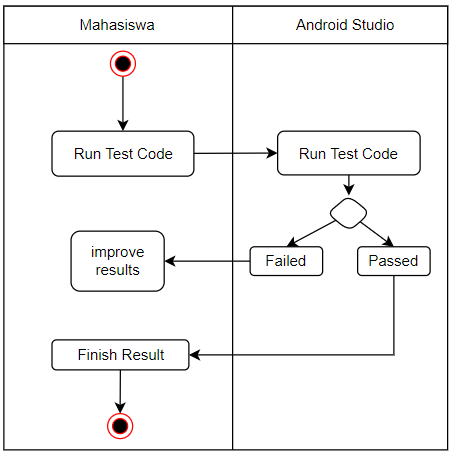
Implementasi topik Basic Aplications Flutter adalah aplikasi List Followers yang merupakan basic implementasi dari pembelajaran flutter, aplikasi list followers mempunyai dua fitur yaitu fitur add followers dan remove followers dimana pada jendela Home pengguna dapat menambahkan list followers sebanyak maksimal 20 followers sesuai looping statis yang di sediakan pada list followers, pada halaman Followers pengguna dapat menghapus list followers yang sudah ditambahkan. Aplikasi List Followers di buat untuk siswa mempelajari cara menambah dan menghapus list secara dinamis (StatefullWidget) yang di tambahkan dengan notifikasi jika list ditambahkan maka akan keluar notifikasi popUp Ditambahkan, dan jika list dihapus maka akan muncul notifikasi popUp list dihapus. Selain dua fitur di atas siswa juga mempelajari bagaimana mengelola state managemnt pada flutter contohnya provider yang berfungsi untuk memamnggil fungsi pada file dart pada direktori program.

## Activity Diagram

Aktifitas dalam pemodelan proses sistem akan di jelaskan pada activity sebagai berikut:

### **Activity Diagram Proses Validasi Mahasiswa Pada Android Studio**

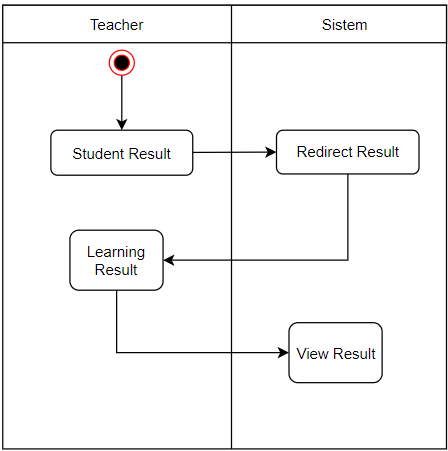
Alur proses Android Studio memvalidasi pekerjaan siswa yang akan dijelaskan pada activity diagram sebagi berikut :



Gambar 4. 4 Activity Diagram validasi code pada Android Studio

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

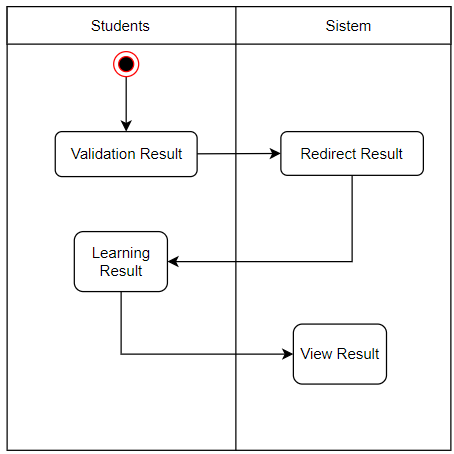
Alur proses sistem bagi guru untuk melihat hasil validasi tugas yang diunggah siswa ke website iCLOP. Diagram aktivitas digambarkan sebagai berikut:



Gambar 4. 5 Teacher Result

### **Activity Diagram Melihat Hasil Validasi Mahasiswa di Platform iCLOP**

Alur proses sistem bagi mahasiswa untuk memeriksa hasil verifikasi tugas yang diunggah ke platform iCLOP. Hal ini dijelaskan dengan menggunakan activity diagram sebagai berikut:



Gambar 4. 6 Validasi pekerjaan mahasiswa

# BAB V. IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan cara membuat aplikasi berdasarkan teori yang digunakan untuk membangun sistem. Instruksi mencakup langkah-langkah untuk membangun aplikasi. Implementasi sistem ini menggunakan bahasa pemrograman Java dengan aplikasi Android Studio.

## Ruang Lingkup Implementasi

Tahap implementasi sistem dilakukan dengan pengkodean menggunakan bahasa pemrograman sesuai dengan hasil analisis dan perancangan sistem yang telah dijelaskan pada bab sebelumnya. Implementasi ini dimaksudkan untuk menjalankan sistem pada aplikasi dan memungkinkan pengguna untuk berinteraksi dengannya. Tahap implementasi sistem biasanya berlaku untuk lingkup tertentu atau spesifikasi sistem. Spesifikasi sistem adalah alat-alat yang dibutuhkan untuk mengimplementasikan sistem yang dibuat. Spesifikasi sistem dapat dibagi menjadi dua jenis : spesifikasi perangkat lunak dan spesifikasi perangkat keras. Berikut adalah tabel spesifikasi perangkat keras yang digunakan selama tahap implementasi sistem.

Tabel 5. 1 Spesifikasi Perangkat Keras

|  |  |
| --- | --- |
| **Perangkat Keras** | **Keterangan** |
| Intel Core i3 | Processor untuk memproses data |
| RAM 8 GB | Sebagai memori internal untuk pemrosesan data |
| SSD 240 GB | Untuk menyimpan data |

Tabel 5. 2 Spesifikasi Perangkat Lunak

|  |  |
| --- | --- |
| Perangkat Lunak | Keterangan |
| Windows 11 ThinkPad | Sistem operasi untuk mengelola proses-proses yang berjalan pada komputer |
| Dart | Bahasa pemrograman untuk *coding* program |
| XAMPP | Perangkat lunak untuk menjalankan *web server* |
| Android Studio | *Software* untuk melakukan penulisan *code* program dan mengembagkan sistem |
| Chrome / Edge | *Browser* untuk mengakses topik pembelajaran Basic Aplications Flutter |
| FlutterTest | *Software* untuk menjalankan unit testing dan widget testing pada android studio |

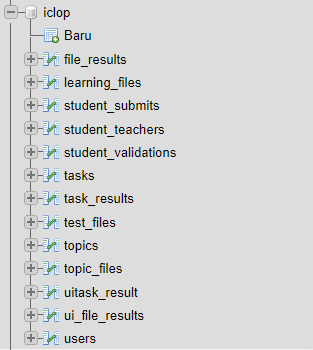
## Batasan Implementasi

Adapun batasan pada implementasi sistem yang dilakukan pada pengembangan topic basic aplication flutter yang meliputi :

* 1. Topic pembelajaran Basic Aplications Flutter di implementasikan pada Platform Intelligent Computer-Assisted Programming Learning Platform (iCLOP)
  2. Pembelajaran menggunakan Android Studio
  3. Pengujian aplikasi menggunakan tolls Flutter Test

## Implementasi Database

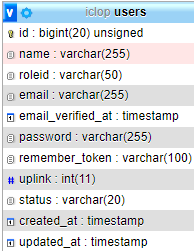
Implementasi database dari analisa dan perancangan yang telah dilakukan pada bab sebelumnya dapat dilihat pada gambar 5.1.



Gambar 5. 1 Implementasi Database

Database iclop terdiri dari 13 tabel, yaitu : file\_result, Learning\_files, students\_submit, student\_teacher, student\_validation, task, task\_results, task\_files, topic, topic\_files, uitask\_results, ui\_file\_results, user.

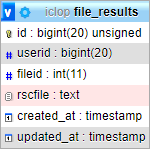
* + - 1. **Tabel User**



Gambar 5. 2 Perancangan tabel user

Pada tabel user digunakan untuk mahasiswa dan dosen untuk validasi data yang di gunakan untuk login sesuai level masing-masing, level digunakan untuk mengarahkan mahasiswa dan dosen ke dashboard masing-masing.

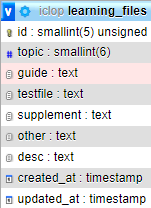
* + - 1. **Tabel file\_results**



Gambar 5. 3 Perancangan tabel file\_result

Pada tabel file\_results digunakan mahasiswa untuk melihat file pekerjaan yang telah di submit. File pekerjaan di simpan dalam bentuk “.txt”.

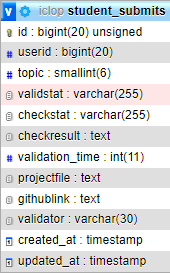
* + - 1. **Tabel Learning\_files**



Gambar 5. 4 Perancangan tabel learning\_files

Tabel learning\_files digunakan untuk admin menambahkan file berupa guide, suplement dan testfile. Dan untuk mahasiswa digunakan untuk mengunduh file guide, suplement dan testfile.

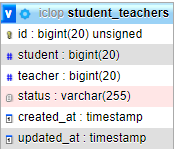
* + - 1. **Tabel students\_submits**



Gambar 5. 5 Perancangan tabel student\_submits

Tabel student\_submits digunakan untuk mahasiswa mengunggah pekerjaan dengan keterangan failed atau passed.

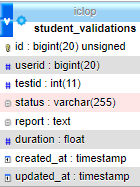
* + - 1. **Tabel student\_teachers**



Gambar 5. 6 Perancangan tabel student\_teachers

Tabel student\_teacher digunakan dosen untuk mamvalidasi mahasiswa yang melakukan registrasi dan memasukan dosen yang akan mengajar pada stage. Hasil dari registrasi akan masuk pada level dosen jika dosen memvalidasi mahasiswa tersebut maka mahasiswa baru bisa mengakses materi yang akan di ajarkan.

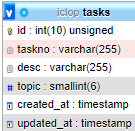
* + - 1. **Tabel student\_validations**



Gambar 5. 7 Perancangan tabel student\_validations

Tabel student\_validations digunakan untuk mevalidasi pekerjaan mahasiswa berdasarkan waktu pengerjaan guide.

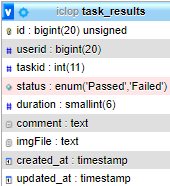
* + - 1. **Tabel task**



Gambar 5. 8 Perancangan tabel task

Tabel task digunakan mahasiswa mengunggah hasil pekerjaan berdasarkan task yang dipilih.

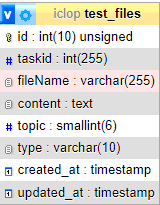
* + - 1. **Tabel task\_results**



Gambar 5. 9 Perancangan tabel task\_results

Tabel task\_result digunakan mahasiswa mengunggah hasil pekerjaan dengan memasukan keterangan atau komentar dalam proses pengerjaan guide, pekerjaan tersebut juga diperjelas dengan gambar yang akan di validasi oleh dosen.

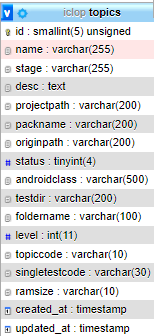
* + - 1. **Tabel test\_files**



Gambar 5. 10 Perancangan tabel test\_files

Tabel test\_files digunakan mahasiswa untuk mengunggah test code sesuai guide dan task yang dikerjakan.

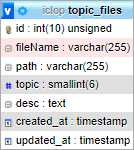
* + - 1. **Tabel topic**



Gambar 5. 11 Perancangan tabel topic

Tabel topic digunakan admin untuk mengedit topic dan menambah topic pembelajaran, bagi mahasiswa digunakan untuk memilih topic pembelajaran yang akan di kerjakan.

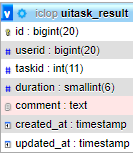
* + - 1. **Tabel topic\_files**



Gambar 5. 12 Perancangan tabel topic\_files

Tabel *topic\_files* digunakan admin untuk menambahkan *file testing* yang akan digunakan untuk memvalidasi *file* yang di unggah oleh mahasiswa, *topic\_files* juga digunakan mahasiswa mengunggah hasil penulisan code dengan memilih *topic files* yang akan di unggah.

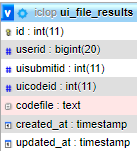
* + - 1. **Tabel uitask\_result**



Gambar 5. 13 Perancangan tabel uitask\_result

Tabel *uitask\_result* digunakan untuk menyimpan komentar dari mahasiswa dalam pengerjaan *guide.*

* + - 1. **Tabel ui\_file\_results**

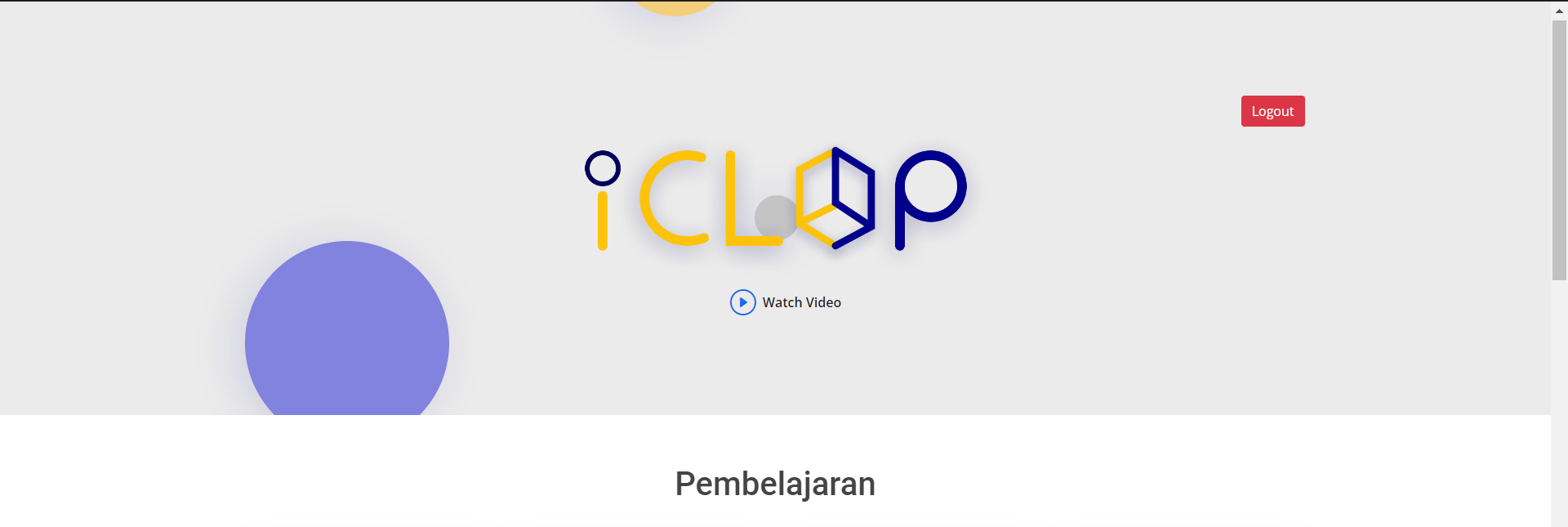


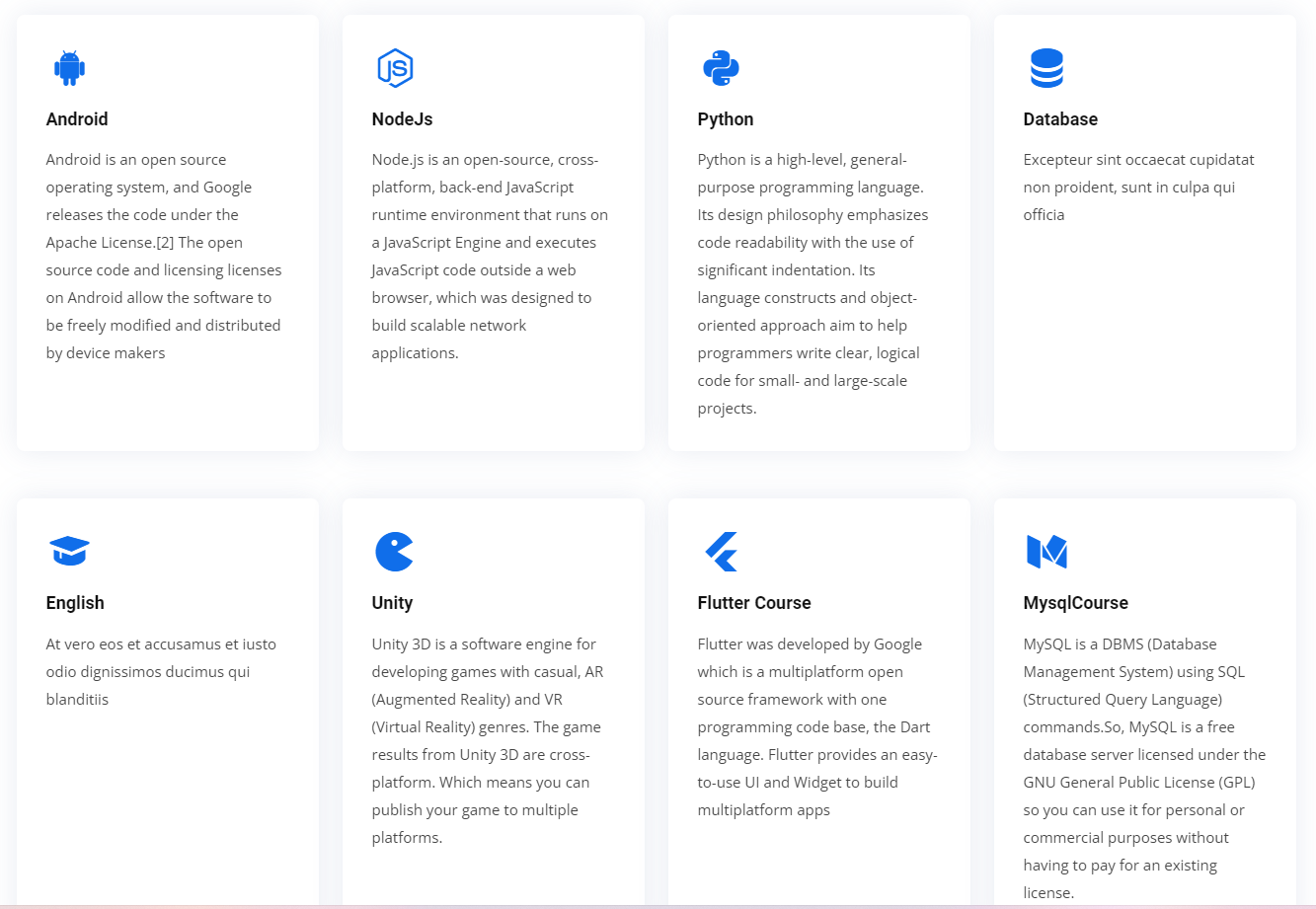
Gambar 5. 14 Perancangan tabel ui\_file\_results

Tabel *ui\_file\_results* digunakan untuk menyimpan hasil unggahan mahasiswa dan di *convert* menjadi bentuk “.txt”

## Implementasi Antarmuka

* + - * 1. Halaman awal pembelajaran

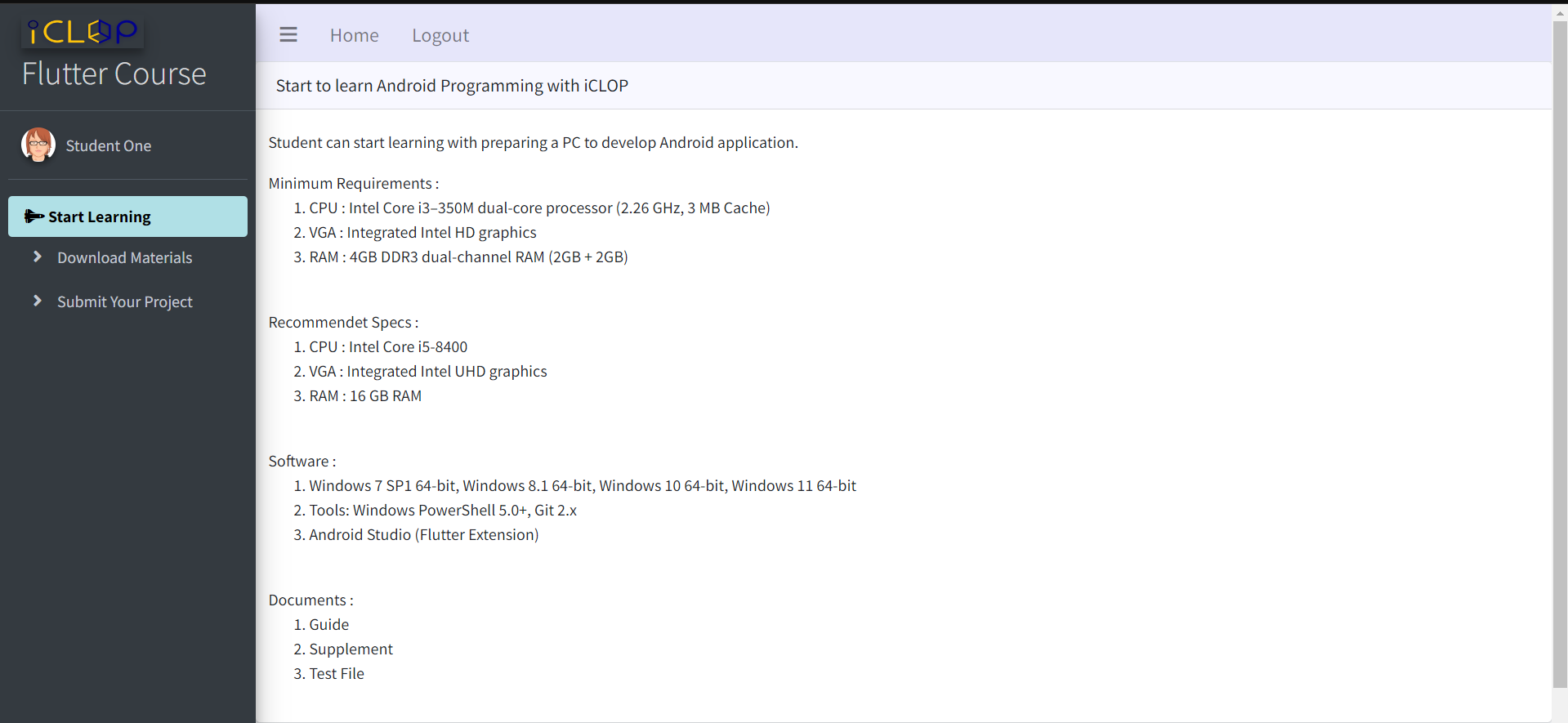




Gambar 5. 15 Halaman lending page

Halam awal home menampilkan topik pembelejaran pada *iCLOP* , pada halaman *lending page* mahasiswa memilih salah satu topik untuk memulai mengerjakan *guide* secara mandiri.

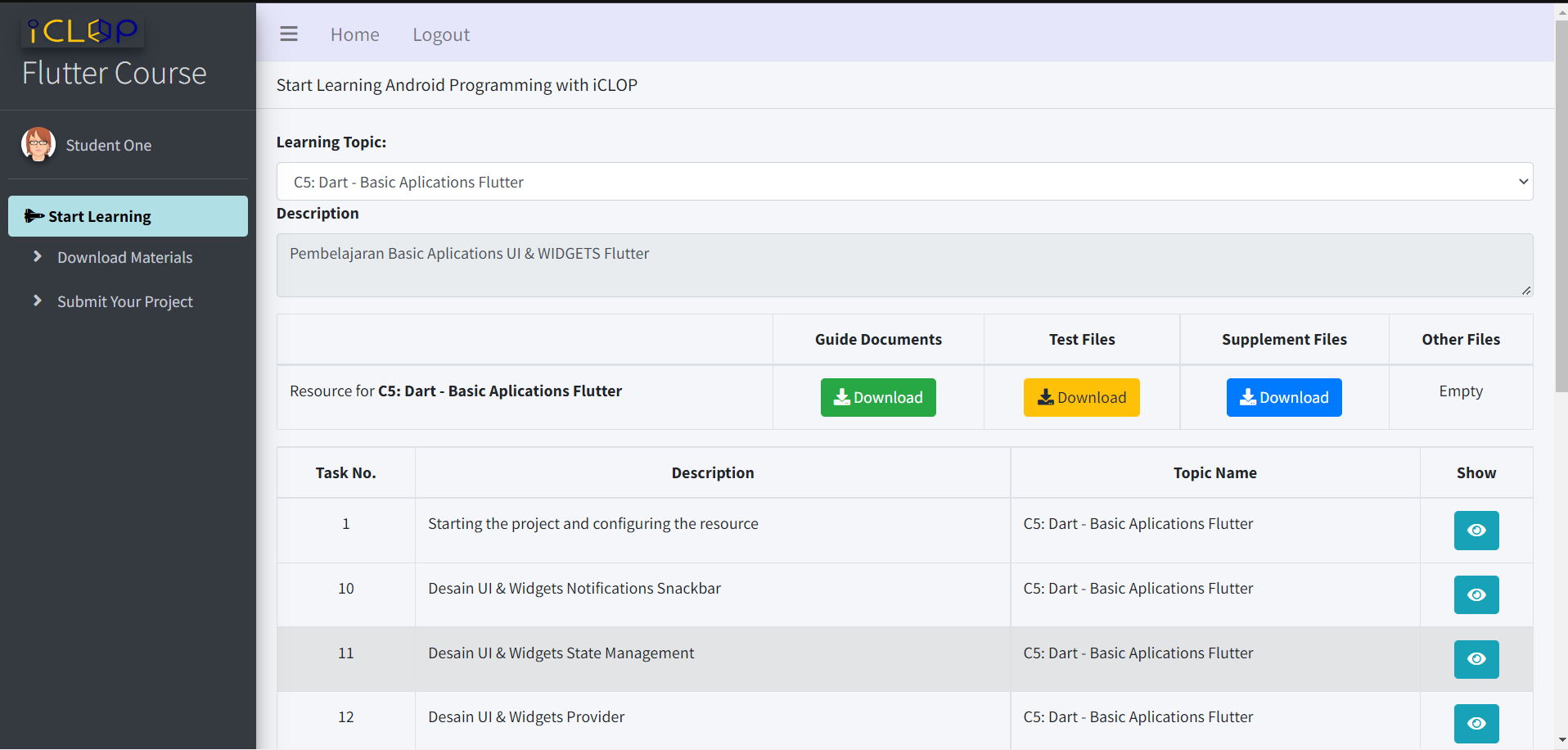
* + - * 1. Halaman Pembelajaran



Gambar 5. 16 Halaman Flutter course

Halaman *Flutter couserse* berisikan spesifikasi yang akan digunakan untuk mengerjakan topic pembelajaran basic aplikasi *flutter*.

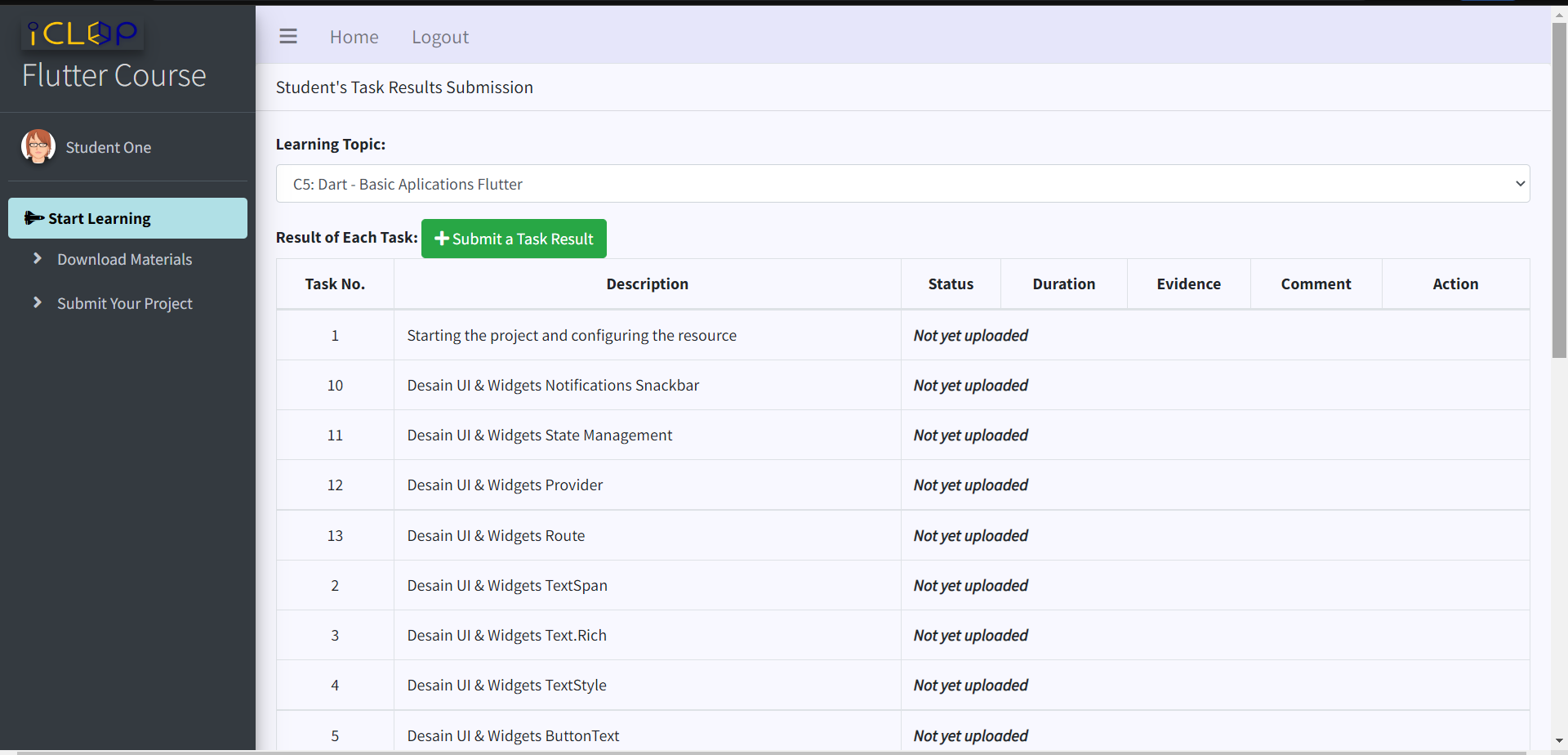
* + - * 1. Halaman Download



Gambar 5. 17 Halaman Download Material

Halaman *Download Material* digunakan untuk mahasiswa mengunduh file berupa *Guide Document, Test Files, Suplement Files* dan *Other Files,* yang akan digunakan untuk mengerjakan tugas basic aplikasi flutter.

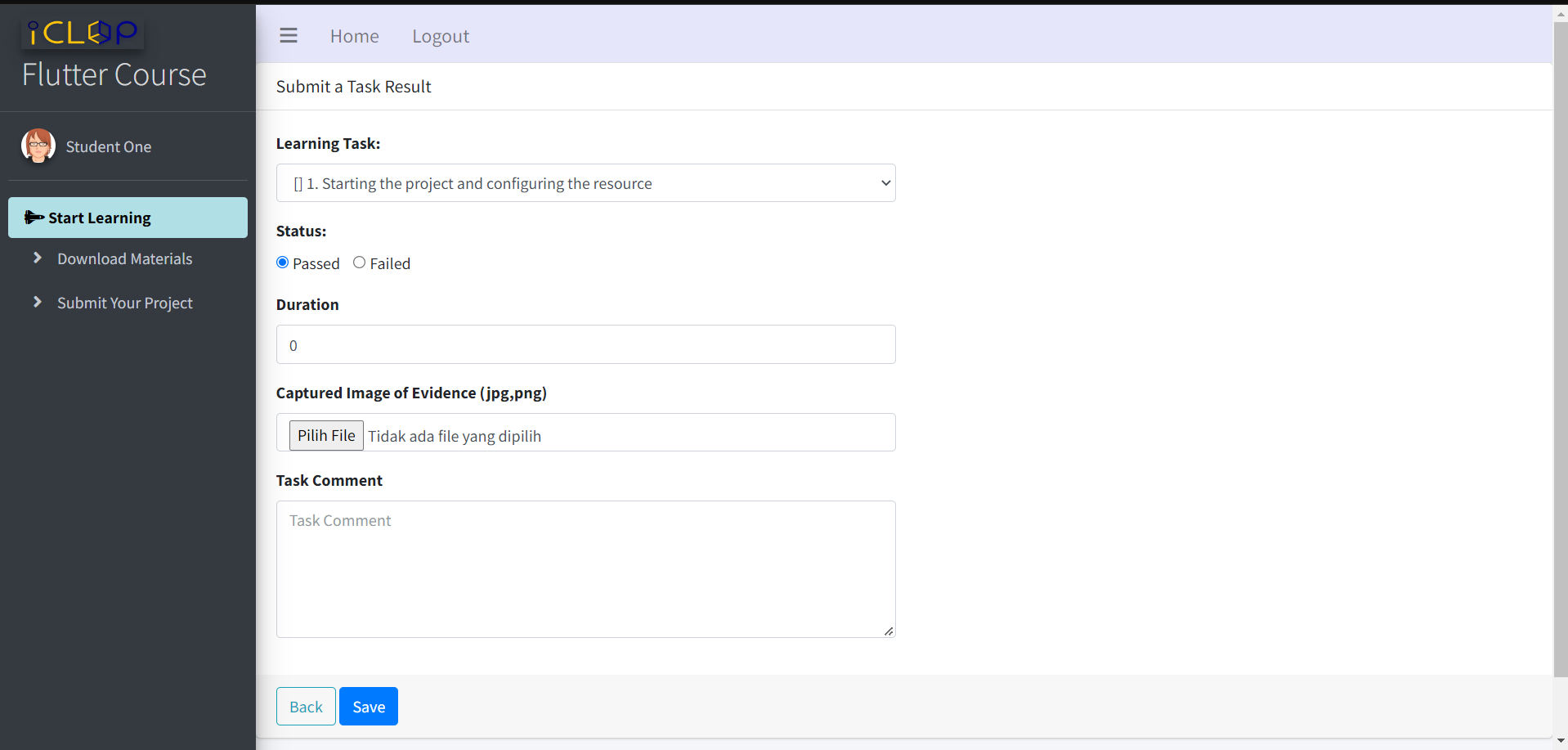
* + - * 1. Halaman submit



Gambar 5. 18 Halaman Submit Project

Halaman *Sumbit Project* digunakan mahasiswa untuk melihat hasil pekerjaan yang sudah di kerjakan dan di kumpulkan di sistem iCLOP

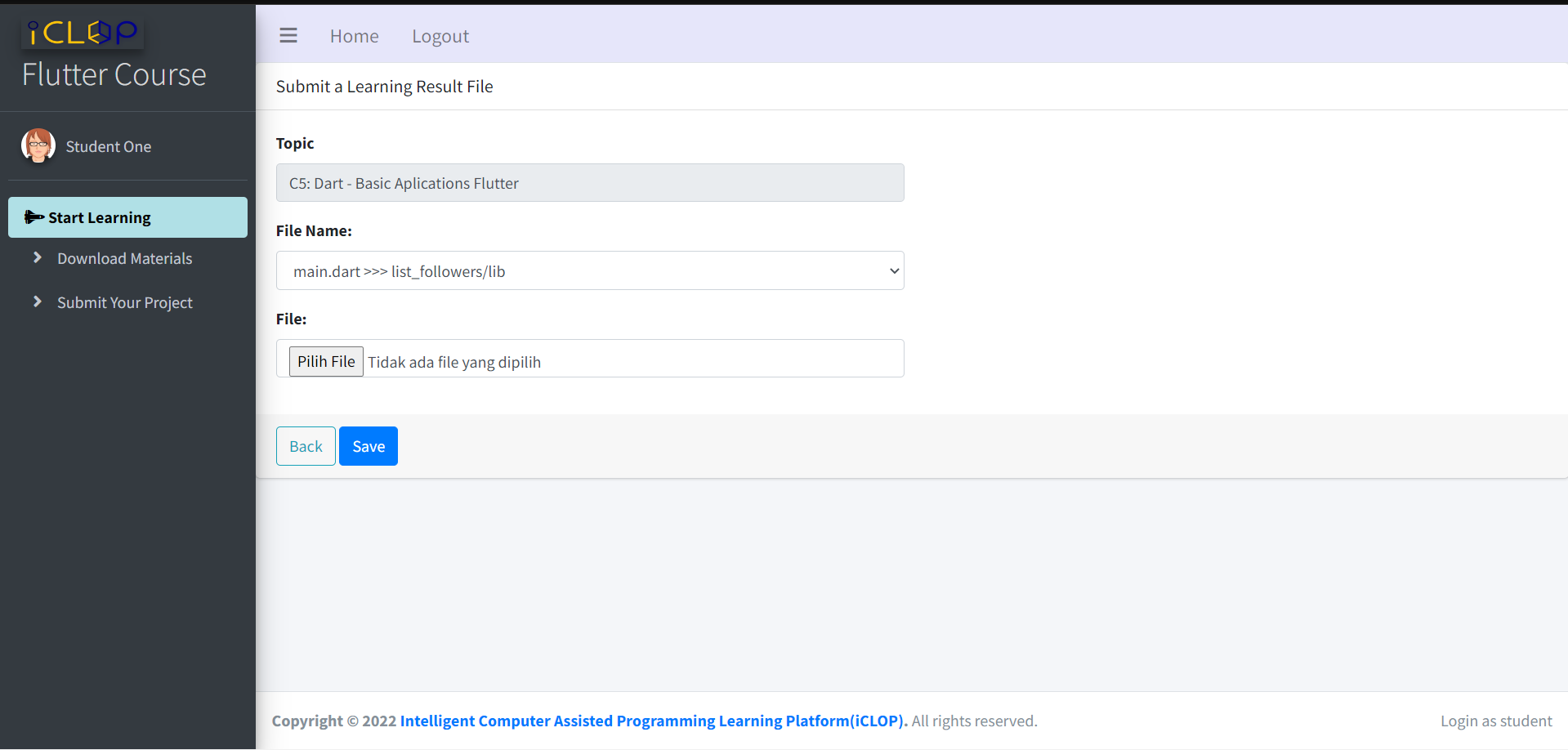
* + - * 1. Halaman submit task result



Gambar 5. 19 Halaman submit task result

Halaman pada gambar 5.19 digunakan mahasiswa untuk mengunggah hasil pekerjaan sesuai task yang dikerjakan meliputi : *Learning Task, Duration, Image* dan *Comment.*

* + - * 1. Halaman submit learning file



Gambar 5. 20 Halaman submit learing file

Halaman pada gambar 5.20 digunakan mahasiswa untuk mengunggah hasil pekerjaan sesuai task yang dikerjakan.

## Implementasi Pembelajaran

Implementasi aplikasi List\_Followers untuk topik pembelajaran *Basic Aplications Dlutter* di iCLOP. Pembelajaran topik dimulai dengan mengunduh file pembelajaran yang terdiri dari 13 *guide file*, 10 *test file*, dan 2 *suplement file*. Dalam menguji guide file, itu akan dicoba dari awal hingga menjadi aplikasi. Dalam memproses aplikasi, aplikasi dibuat seperti yang diharapkan. Jika tidak sesuai, mahasiswa perlu memperbaiki guide file. Jika sesuai guide tes siap digunakan untuk mahasiswa.

### **5.1.1** ***Guide File Basic Aplications Flutter***

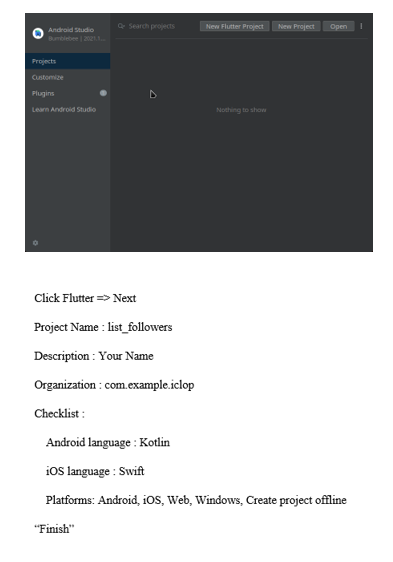
Guide file adalah file yang berisi instruksi untuk membuat aplikasi pembelajaran. Guide fille menyediakan instruksi untuk mengelola antarmuka pengguna aplikasi dan membuat fitur untuk setiap aktivitas aplikasi. Pada pembelajaran basic aplikasi flutter akan diimplementasikan dalam aplikasi Bernama list\_followers*.* Aplikasi *ListFollowers* merupakan aplikasi basic yang mempunyai fitur add. dan remove. Dimana pada pembelejaran basic aplikasi akan di jelaskan pada guide.

Dalam pembuatan aplikasi *ListFollowers* membutuhkan 13 *guide file,10* *test file* dan dua *supplement file.* *Test file* digunakan untuk menguji tugas yang dikerjakan pada setiap *guide file. Guide file* tersebut berisikan tentang:

|  |  |
| --- | --- |
| **NO** | **Deskripsi Guide** |
| 1. | Starting Project, Configure Project and Resource |
| 2. | Desain UI & Widgets TextSpan |
| 3. | Desain UI & Widgets Text.Rich |
| 4. | Desain UI & Widgets TextStyle |
| 5. | Desain UI & Widgets ButtonText |
| 6. | Desain UI & Widgets Image |
| 7. | Desain UI & Widgets Padding, Border |
| 8. | Desain UI & Widgets ListView, Development Models |
| 9. | Desain UI & Widgets Icon |
| 10. | Desain UI & Widgets Notifications Snackbar |
| 11. | Desain UI & Widgets State Management |
| 12. | Desain UI & Widgets Provider |
| 13. | Desain UI & Widgets Route |

Starting Project, Configure Project and Resource

Memulai membuat project baru flutter pada Android Studio dengan configurasi yang sesuai seperti pada gambar 5.21, sehingga program dapat di running dalam berbagai platform seperti Android, Windows, Linux dan Web.

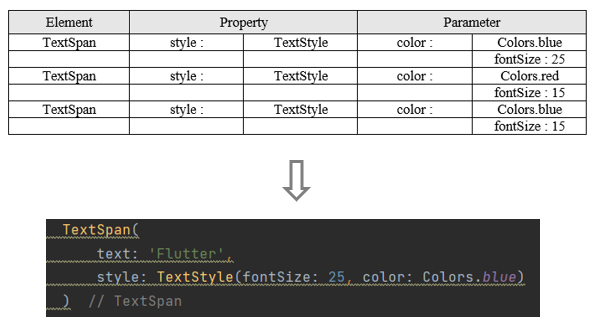


Gambar 5. 21 Configurasi Proyek

Gambar di atas menjelaskan konfigurasi flutter pada android studio yang berisi nama projek, deskripsi, depedency, dan checklist yang berfungsi untuk memilih bahasa serta platform yang akan digunakan dalam pembuatan program aplikasi mobile pada flutter.

Desain UI & Widgets TextSpan

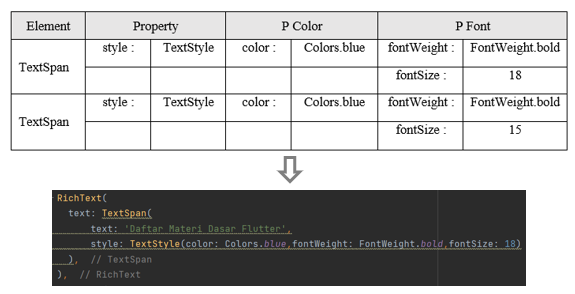
Membuat desain UI untuk menggambarkan text dalam sebuah paragraph, sebagai sub pohon dari widgets TextRich, TextSpan befungsi sebagai object yang mempunyai text biasa. Seperti gambar di bawah ini



Gambar 5. 22 Implementasi guide TextSpan pada dart file

Desain UI & Widgets Text.Rich

Membuat desain text dengan beberapa gaya yang berbeda yang mengambil text dari *textSpan* sebagi subpohon

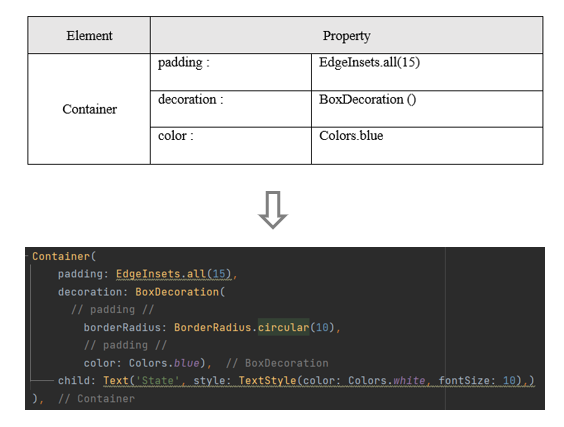


Gambar 5. 23 Implementasi Guide TextRich pada dart file

\

* 1. Desain UI & Widgets TextStyle

Mendesain text pada widgets sehingga viewText tampil interaktif yang di antaranya menambahkan warna, ukuran text dan decorations.



Gambar 5. 24 Implementasi Guide TextStyle pada dart file

* 1. Desain UI & Widgets ButtonText

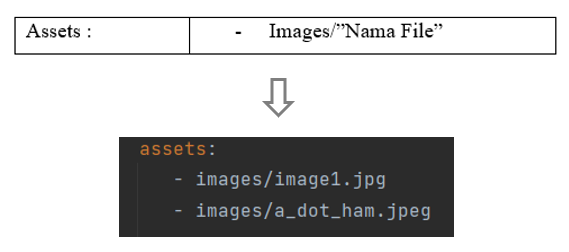
Membuat Button text Flutter dengan menambahkan text decorations yang nantinya akan di gunakan untuk membuka halaman-halaman pada aplikasi list\_followers.



Gambar 5. 25 Implementasi Guide ButtonText pada dart file

* 1. Desain UI & Widgets Image

Membuat gambar pada halaman flutter.dart dan main.dart. Gambar dibuat menggunakan local image dan network image, localImage memanggil gambar pada asset yang sudah di set pada pubspec.yml sedangkan networkImage memanggil gambar berbentuk URL.



Gambar 5. 26 Implementasi Guide Iamge pada dart file

* 1. Desain UI & Widgets Padding, Border

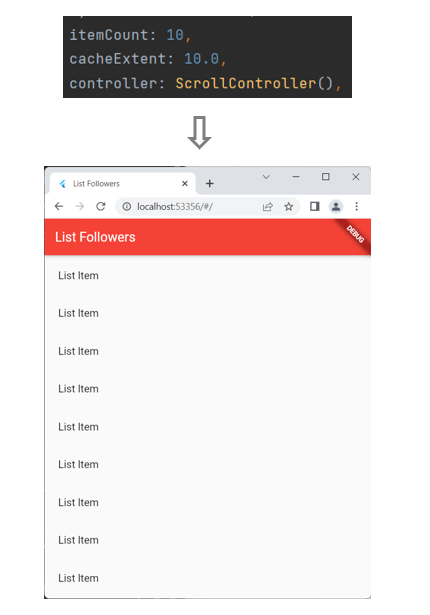
Membuat desain border dan padding pada *TextStyle* dan *ButtonText* untuk mengubah sudut dari persegi pada button dan textstyle.



Gambar 5. 27 Implementasi Guide Padding & Border pada dart file

* 1. Desain UI & Widgets ListView, Development Models

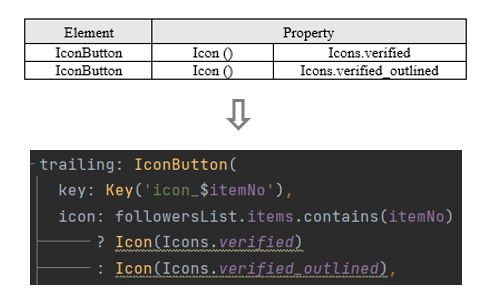
Mendesain tammpilan *listView* yang digunakan untuk menampilkan output dari *textCount*  dari *models* *Followers file .* Tampilan pada view akan sesuai dengan *itemCount* yang di input, jika 10 maka *listView* yang tampil akan 9 dan seterusnya. Models merupakann *basic code* yang berfungsi untuk menambahkan juga menghapus *itemCount*, Juka di tambahkan maka list akan pindah ke *Followers file* pada folder home dan sebaliknya.Pada *listView* juga ditambahkan *scrollingView* yang digunakan *user* untuk menggulirkan layar.



Gambar 5. 28 Implementasi Guide ListView pada view

* 1. Desain UI & Widgets Icon

Membuat icon pada *home file* , *Icons favorite\_outline\_sharp* yang digunakan untuk mengakses halaman *followers* dan *Icons home\_work* yang digunakan untuk kembali ke menu home pada *main file,* Sedangkan pada text *listItem* menggunakan *Icons verified* dan *verified\_outline* yang berfungsi sebagai informasi *listItem* yang sudah di *checklist* dan yang belum di *checklist.*



Gambar 5. 29 Implementasi Guide Icons pada dart file

* 1. Desain UI & Widgets Notifications Snackbar

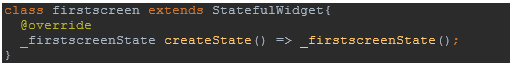
Membuat notifikasi pada *listFollowers* berupa *pop up* yang berisi pesan Tambah ke *Followers* dan Batal Ditambahakn pada *home file*. Sedangkan pada *followers file* pesan berupa tambahkan dulu jika daftar kosong dan Hapus *Followers* untuk memberi pesan bahwa *list item* telah dihapus dari *followers file.*



Gambar 5. 30 Implementasi pada halaman home file

* 1. Desain UI & Widgets State Management

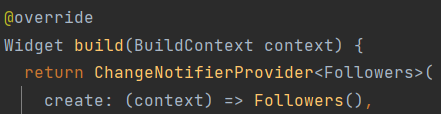
Mendesain relasi *screen* yang mana sebelumnya *screen* hanya bisa di *running* pada *screen* itu sendiri, dengan *state* maka akan memulai membuat *statfullwidgets* yang mana *state* akan bisa di akses oleh semua *screen.*



Gambar 5. 31 Implementasi code state

* 1. Desain UI & Widgets Provider

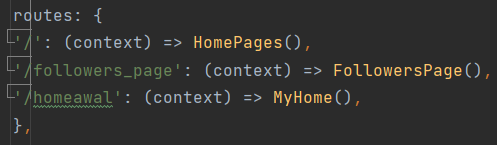
Membuat *provider* yang berfungsi sebagai *inhiritance* yaitu *widgets* yang menurunkan ke anak-anak nya. Setelah di definisikan menjadi *statefullwidgets* pada *state managements*, maka *providers* digunakan untuk memenaggil fungsi dari anak-anak nya. *Providers* di gunakan di *route file*



Gambar 5. 32 Implementasi provider pada route file

* 1. Desain UI & Widgets Route

Membuat navigasi multiscreen menggunakan route ,route digunakan untuk mendefiniksikan sebuah halaman untuk di panggil pada navigasi.pada aplikasi listFollowers terdapat beberapa navigasi diantaranya : Followers Page, Home Page, dan Main Page



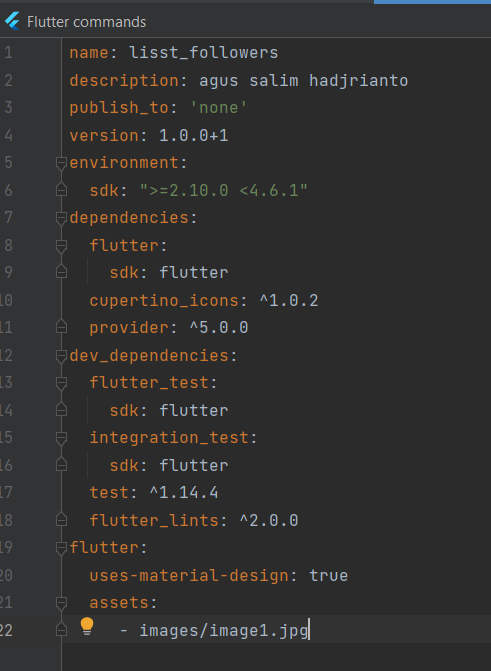
Gambar 5. 33 Implementasi Routing pada route file

### **5.1.2 *Suplement File Basic Aplications Flutter***

*Supplement file* dalam pembuatan aplikasi *ListFollowers* terdiri dari satu gambar dan satu *file*. *Supplement file* tersebut menggunakan ekstensi *file* ‘*.jpg’* dan gambarnya sebagai berikut:

* + - 1. Pubscpec.yml

Pubscpec.yml digunakan untuk mengatur depedensi dan konfigurasi library yang digunakan pada aplikasi listFollowers.



Gambar 5. 34 Gambar pubspec.yml

* + - 1. Gambar.jpg



Gambar 5. 35 Gambar1.jpg

### **5.1.3 *Test File Basic Aplications Flutter***

*Test file* dilakukan untuk menguji hasil pekerjaan dari mahasiswa menggunakan *Android Studio*, pada aplikasi *listFollowers* *flutter basic aplication* menggunakan *flutter test*, terdapat sepuluh *test code* yang akan *memvalidasi* pekerjaan dari mahasiswa. Berikut pengujian untuk membuat *list item* pada *provider state.*



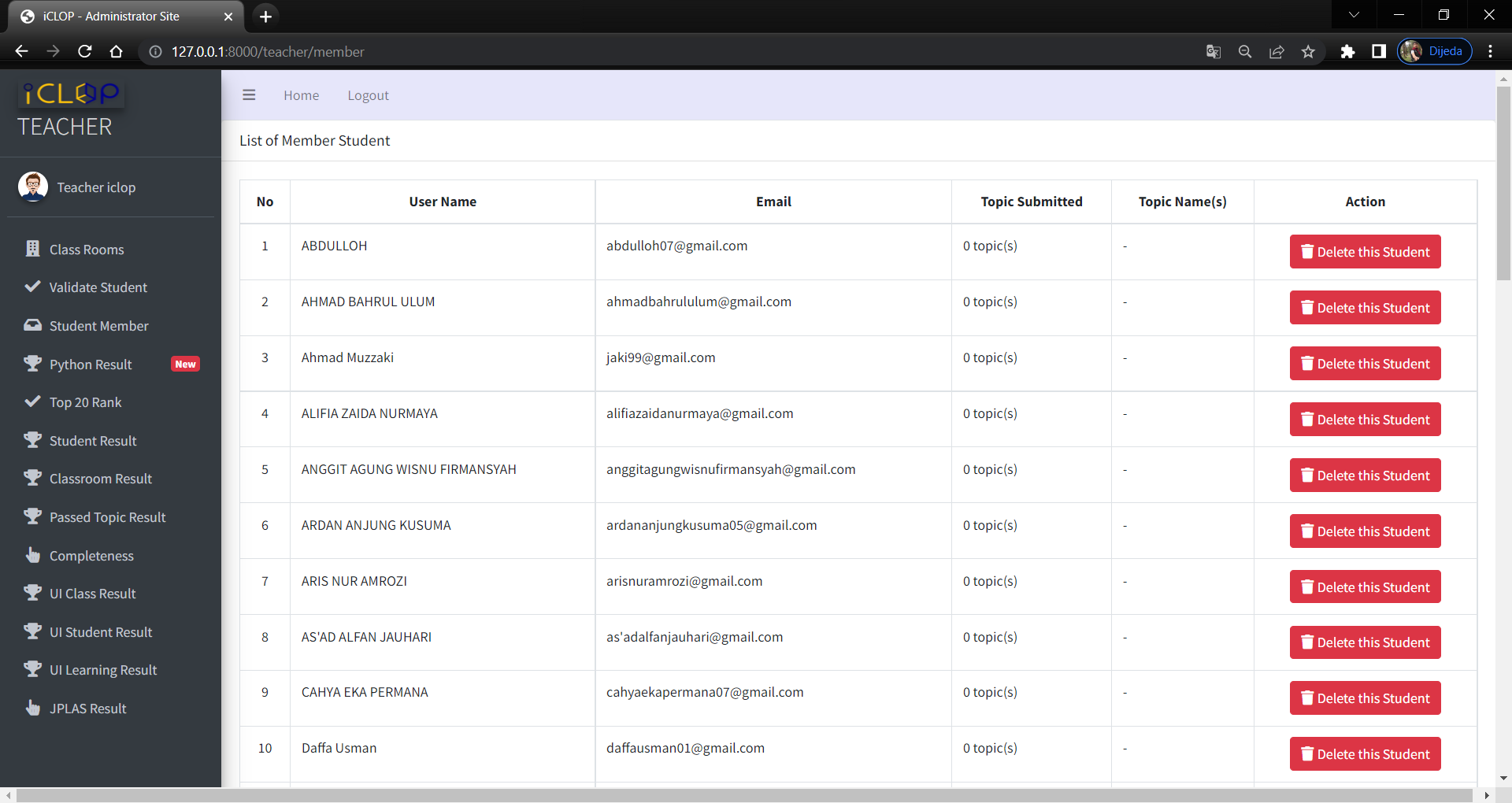
Gambar 5. 36 Pengujian Provider pada List Item

## Pengujian

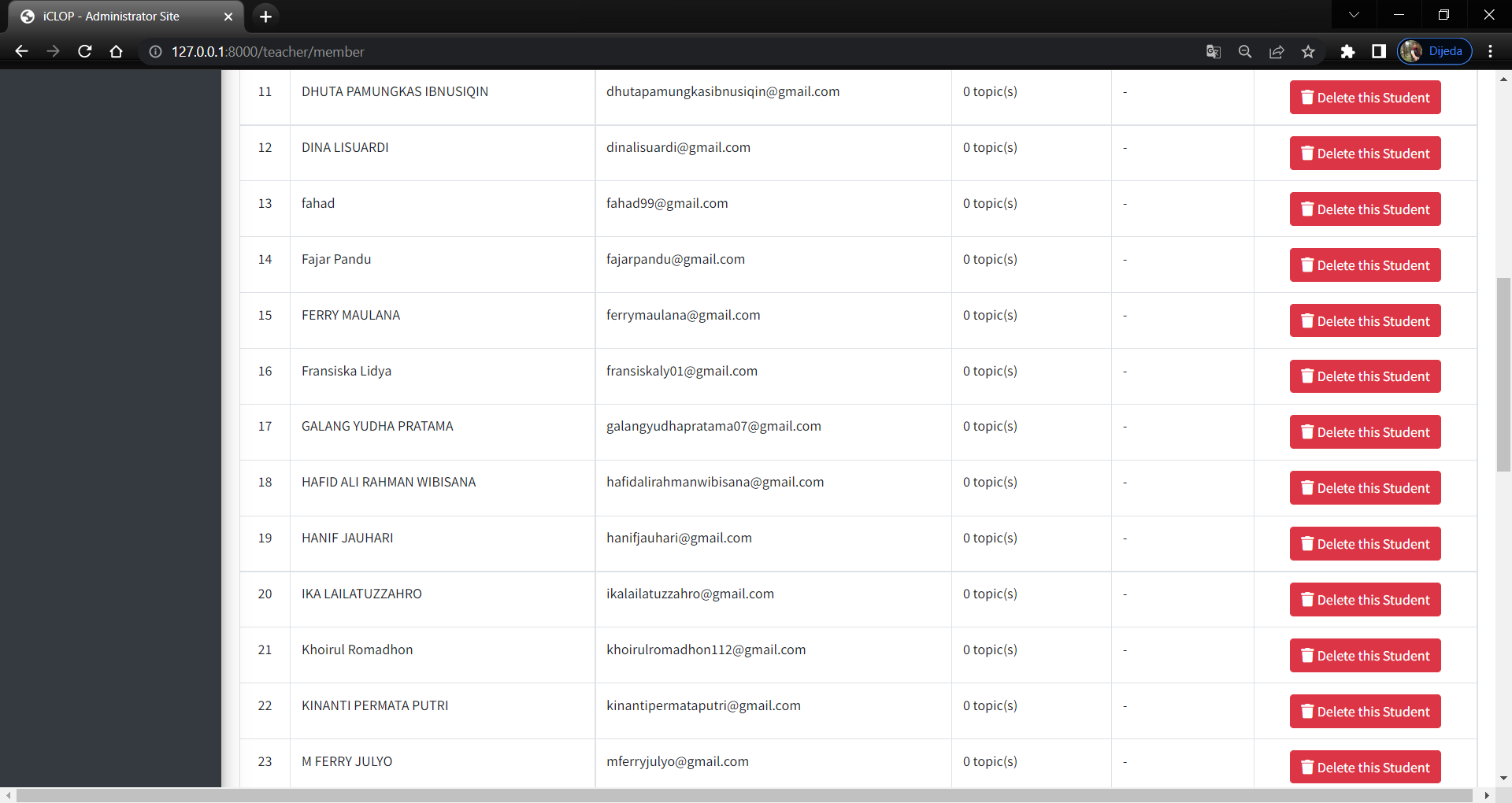
Pengujian pada pengembangan topic basic aplikasi pembelajaran flutter bertujuan untuk mengetahui performa aplikasi yang akan di sajikan ke mahasiswa, selain itu pengujian ini juga berfungsi untuk memvalidasi data mahasiswa yang telah menerima pembelajaran basic aplikasi flutter.

### **Platform Testing**

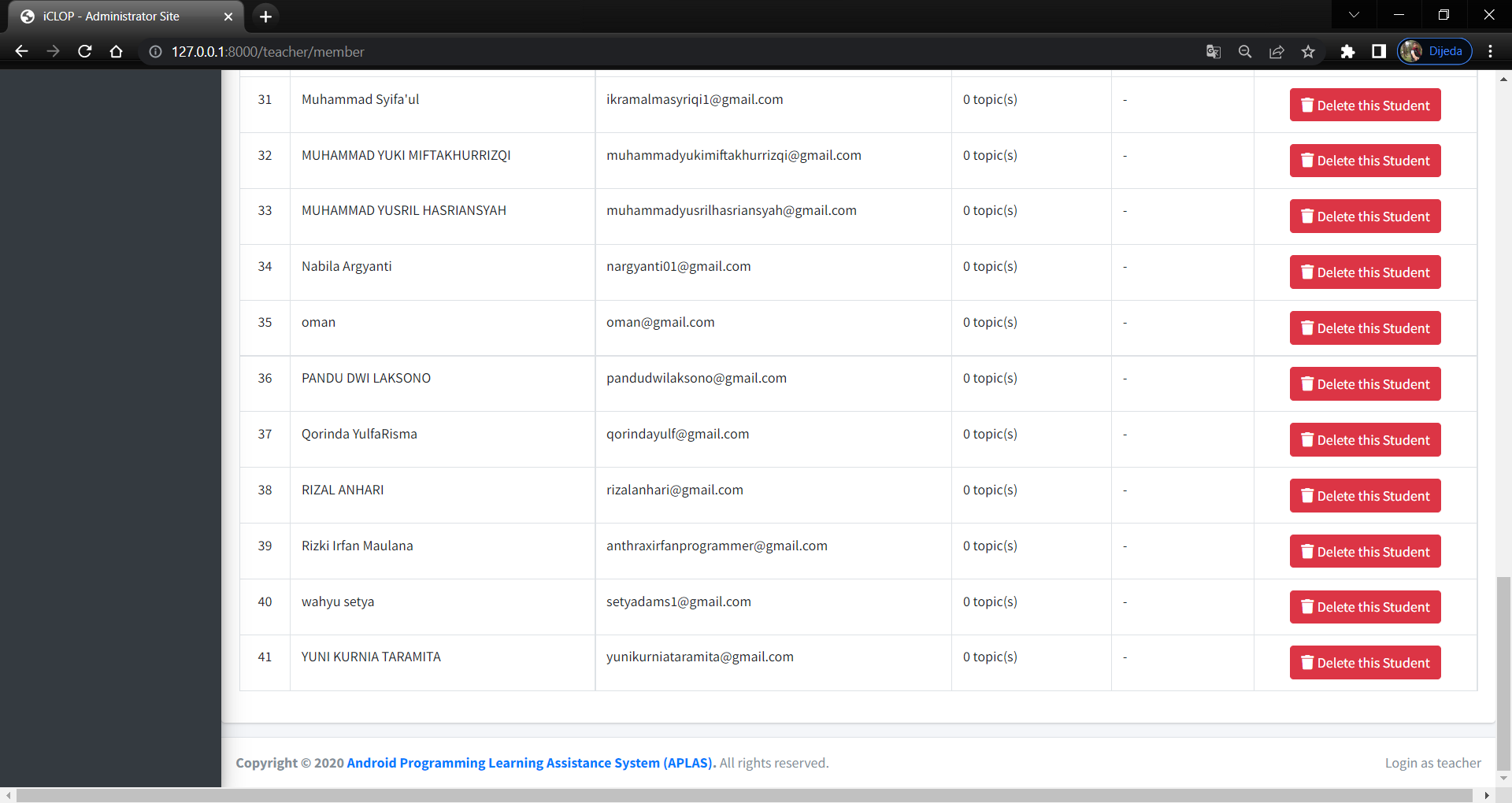
Pengujian topik *Basic Aplication Flutter* ini dilakukan dengan bantuan dari 40 orang mahasiswa Politeknik Negeri Malang. Mahasiswa tersebut merupakan mahasiswa tingkat 4 dari Jurusan Teknik Informatika dari kelas TI 4D dan Basecamp Candi Mendut. Semua mahasiswa sudah mendapatkan pelajaran *Android* sebelumnya. Pengujian dilakukan dalam kurun waktu seminggu, berikut daftar mahasiswa yang menjadi *valunteer.*



Gambar 5. 37 List Valunteer 1-10



Gambar 5. 38 List Valunteer 11-23



Gambar 5. 39 List Valunteer 23-41

Dari gambar 5.36, 5.37 dan 5.38 dipastikan bahwa semua *volunteer* sudah terdaftar sebagai mahasiswa pada platform iCLOP dan memiliki aplikasi *Android Studio* pada perangkat mereka. *Volunteer* melakukan login pada *platform* iCLOP kemudian masuk ke menu *Download Materials* dan memilih topik *Basic Aplications Flutter. 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* atau *passed,* maka *volunteer* telah mengerjakan dengan benar dan *volunteer* bisa mengunggah hasil pengerjaan di platform iCLOP untuk divalidasi.

Mengunggah hasil pengerjaan dilakukan pada menu *Submit your result* dan pastikan topik yang dipilih *Basic Aplications Flutter.* Dalam menggunggah hasil pengerjaan dapat dilakukan dengan tiga acara yaitu *zip file, multi files* dan *github link.* Hasil validasi akan muncul pada platform iCLOP 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, dari penjelasan di atas dapat di ambil data pengujian yang meliputi status,durasi pengerjaan *guide* dan komentar dari setiap *guide* yang dikerjakan, data tersebut akan digunakan dalam membuat grafik presentase pada gambar 6.1 dan 6.2.

### **Performance Testing**

Performance Testing adalah suatu proses untuk menguji batas ketahanan dan kestabilan sebuah sistem termasuk modul aplikasi dan infrastrukturnya, serta menguji bagaimana sistem/aplikasi tersebut dapat bekerja kembali setelah terjadi down dalam kondisi load yang tinggi. Performance Testing adalah suatu proses untuk menguji batas ketahanan dan kestabilan sebuah sistem termasuk modul aplikasi dan infrastrukturnya.

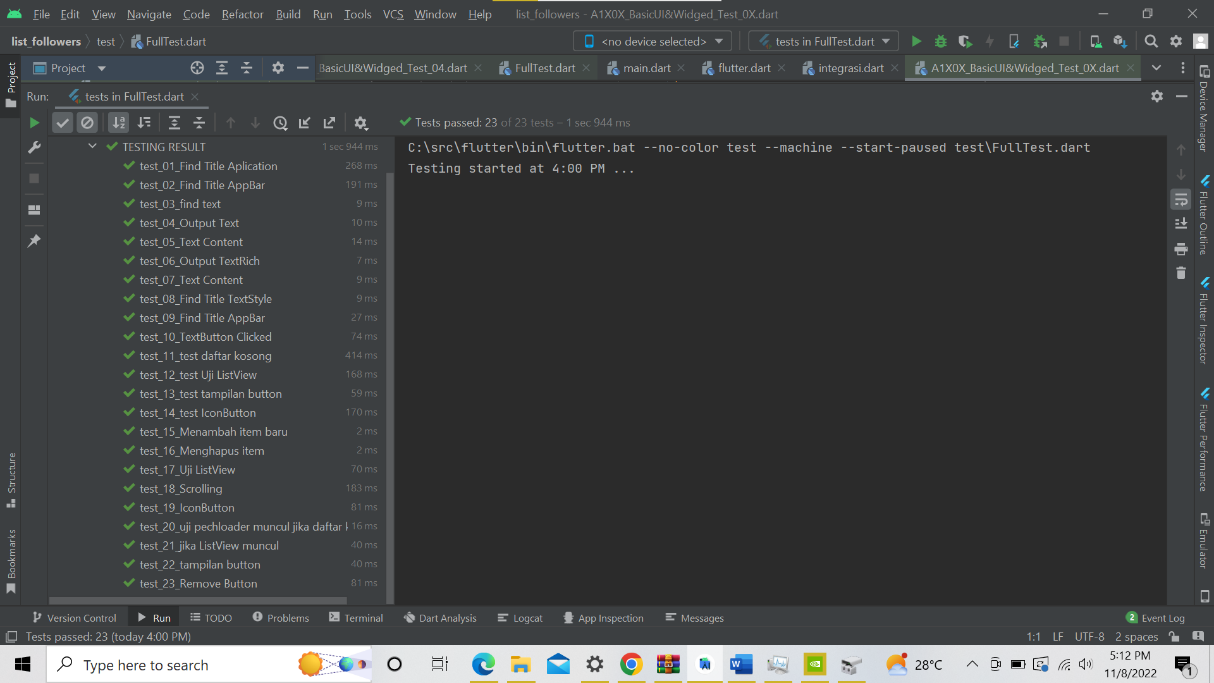
Aplikasi list\_followers di running dalam tiga device yang berbeda yang mana bertujuan untuk mengetahui performa aplikasi jika di jalankan di berbagai device yang berbeda, spesifikasi laptop akan di jelaskan pada tabel berikut,

Pada bagian performance testing akan membandingkan 3 user dengan hardware yang berbeda dengan kualifikasi system pada table berikut:

Tabel 5. 3 Spesifikasi laptop

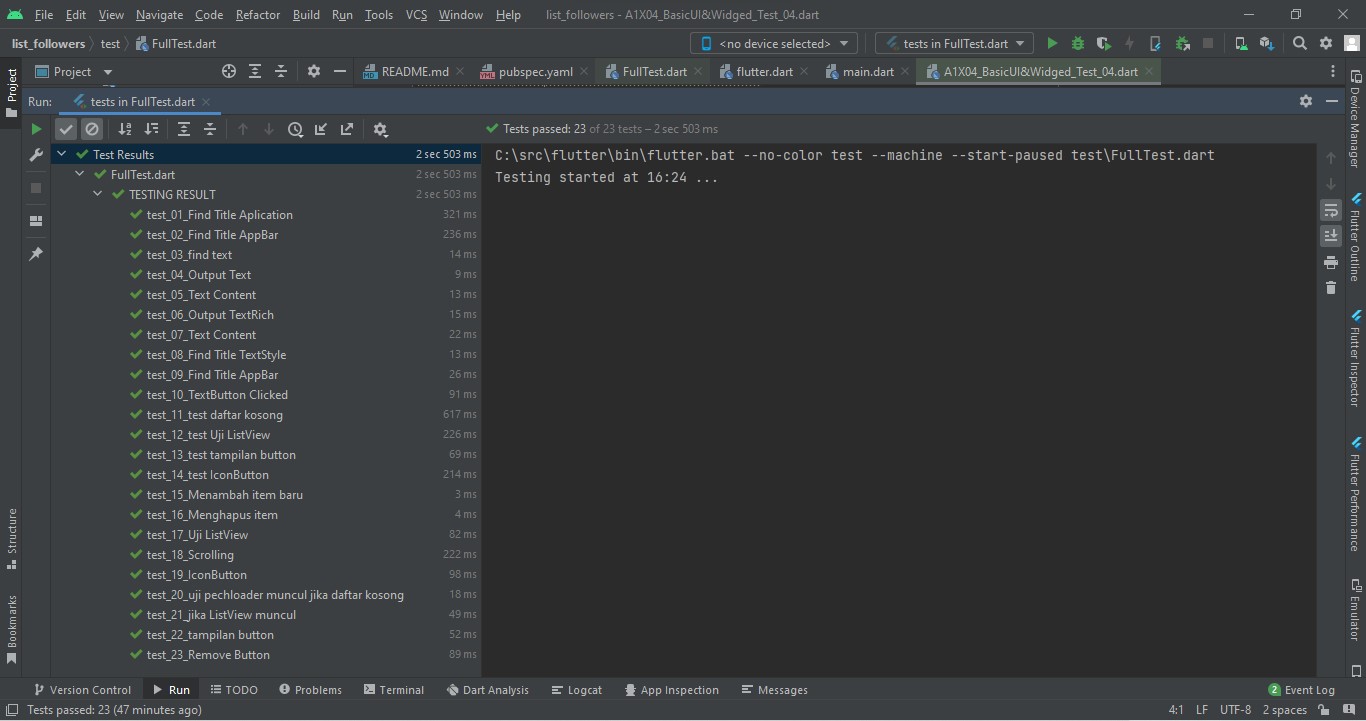
|  |  |
| --- | --- |
| User | Spesifikasi system |
| Asus | * Ram 8 * Windows 10 * Core i7-8265U 8th Gen * Intel UHD + NVIDIA GeForce MX230 * HDD 1 TB |
| Acer | * Ram 8 * Windows 11 * Core i5-10210U 10th Gen * Intel UHD + NVIDIA GeForce MX250 * SSD 512 |
| Lenovo | * Ram 8 * Windows 11 * Core i3-10110U 10th Gen * Intel UHD * SSD 240 |

Tabel di atas terdapat tiga spesifikasi laptop yang berbeda, dengan spesifikasi tersebut diharapkan pengujian performance testing dapat menampilkan performa yang berbeda di setiap device laptop, berikut data yang di ambil dari proses testing dari tiga device yang berbeda :



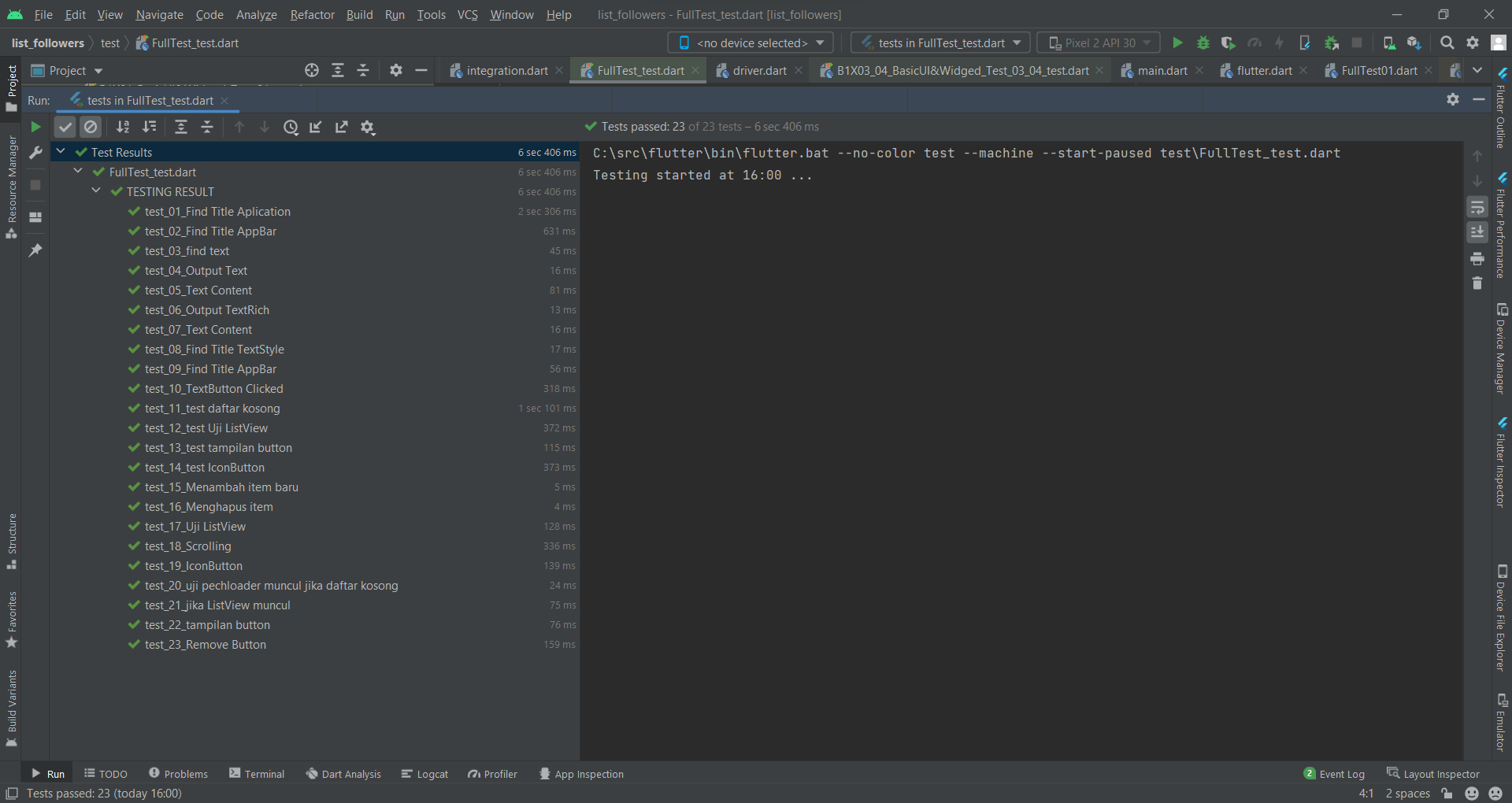
Gambar 5. 40 Testing pada laptop Asus

Gambar di atas menjelaskan proses testing yang dilakukan pada laptop Asus dengan spesifikasi yang terdapat pada tabel 5.3 . Testing tersebut membutuhkan waktu 1,944 detik dari total 23 testing code.



Gambar 5. 41 Testing pada laptop Acer

Gambar di atas menjelaskan proses testing yang dilakukan pada laptop Acer dengan spesifikasi yang terdapat pada tabel 5.3 . Testing tersebut membutuhkan waktu 2,503 detik dari total 23 testing code.



Gambar 5. 42 Testing pada laptop Lenovo

Gambar di atas menjelaskan proses testing yang dilakukan pada laptop Lenovo dengan spesifikasi yang terdapat pada tabel 5.3 . Testing tersebut membutuhkan waktu 6,406 detik dari total 23 testing code.

Dari Gambar di atas akan di rangkum dalam tabel performance testing yang di dalamnya terdapat detail nilai testing dari total 23 testing code, berikut tabel performance testing :

Tabel 5. 4 Perbandingan waktu pengujian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Testing** | **Waktu /Detik** | | |
| **Asus** | **Acer** | **Lenovo** |
| **1** | test\_01\_Find title aplication | 0,268 | 0,321 | 2,306 |
| **2** | test\_02\_Find title appBar | 0,191 | 0,236 | 0,631 |
| **3** | test\_03\_Find Text | 0,009 | 0,014 | 0,045 |
| **4** | test\_04\_Output Text | 0,01 | 0,009 | 0,016 |
| **5** | test\_05\_Text content | 0,014 | 0,013 | 0,081 |
| **6** | test\_06\_Output TextRich | 0,007 | 0,015 | 0,013 |
| **7** | test\_07\_Text Content | 0,009 | 0,022 | 0,016 |
| **8** | test\_08\_Find Title TextStyle | 0,009 | 0,013 | 0,017 |
| **9** | test\_09\_Find Title appBar | 0,027 | 0,026 | 0,056 |
| **10** | test\_10\_TextButton Clicked | 0,074 | 0,091 | 0,318 |
| **11** | test\_11\_Test daftar kosong | 0,414 | 0,617 | 1,101 |
| **12** | test\_12\_Test uji ListView | 0,168 | 0,226 | 0,372 |
| **13** | test\_13\_Test tampilan button | 0,059 | 0,069 | 0,115 |
| **14** | test\_14\_Test icon button | 0,17 | 0,214 | 0,373 |
| **15** | test\_15\_Menambah item baru | 0,002 | 0,003 | 0,005 |
| **16** | test\_16\_Menghapus item | 0,002 | 0,004 | 0,004 |
| **17** | test\_17\_Uji ListView | 0,07 | 0,082 | 0,128 |
| **18** | test\_18\_Scrolling | 0,183 | 0,222 | 0,336 |
| **19** | test\_19\_IconButton | 0,081 | 0,098 | 0,139 |
| **20** | test\_20\_Uji pechloader daftar kosong | 0,016 | 0,018 | 0,024 |
| **21** | test\_21\_Uji ListView muncul | 0,04 | 0,049 | 0,075 |
| **22** | test\_22\_Uji Tampilan button | 0,04 | 0,052 | 0,076 |
| **23** | test\_23\_Remove Button | 0,081 | 0,089 | 0,159 |
| **TOTAL** | | **1,944** | **2,503** | **6,406** |

Tabel perbandingan data testing digunakan untuk mengklasifikasikan data testing berdasarkan jumlah testing dan waktunya, seperti tabel berikut terdapat tiga klasifikasi data testing meliputi : cepat, rata-rata dan lambat. Dari jumlah data pada tabel 5.5 diperoleh dari minimal , maksimal dan avarage dari laptop asus, acer dan lenovo, selengkapnya dapat dilihat dalam tabel dibawah ini.

Tabel 5. 5 Perbandingan data testing / detik

|  |  |  |  |
| --- | --- | --- | --- |
| **Perbandingan Data Testing / Detik** | | | |
| **No.** | **Cepat** | **Rata-Rata** | **Lambat** |
| **1** | 0,268 | 0,965 | 2,306 |
| **2** | 0,191 | 0,352666667 | 0,631 |
| **3** | 0,009 | 0,022666667 | 0,045 |
| **4** | 0,009 | 0,011666667 | 0,016 |
| **5** | 0,013 | 0,036 | 0,081 |
| **6** | 0,007 | 0,011666667 | 0,015 |
| **7** | 0,009 | 0,015666667 | 0,022 |
| **8** | 0,009 | 0,013 | 0,017 |
| **9** | 0,026 | 0,036333333 | 0,056 |
| **10** | 0,074 | 0,161 | 0,318 |
| **11** | 0,414 | 0,710666667 | 1,101 |
| **12** | 0,168 | 0,255333333 | 0,372 |
| **13** | 0,059 | 0,081 | 0,115 |
| **14** | 0,17 | 0,252333333 | 0,373 |
| **15** | 0,002 | 0,003333333 | 0,005 |
| **16** | 0,002 | 0,003333333 | 0,004 |
| **17** | 0,07 | 0,093333333 | 0,128 |
| **18** | 0,183 | 0,247 | 0,336 |
| **19** | 0,081 | 0,106 | 0,139 |
| **20** | 0,016 | 0,019333333 | 0,024 |
| **21** | 0,04 | 0,054666667 | 0,075 |
| **22** | 0,04 | 0,056 | 0,076 |
| **23** | 0,081 | 0,109666667 | 0,159 |
|  | **1,944** | **3,617666667** | **6,406** |

Dari data tebel di atas untuk mengetahui nilai dari cepat, rata-rata dan lambar menggunakan rumus berikut :

* + - * 1. Cepat :

=MIN(Nilai Awal:Nilai Akhir)

Rumus :

|  |  |  |
| --- | --- | --- |
| Asus | Acer | Lenovo |
| 0,268 | 0,321 | 2,306 |

Dari tiga tabel berikut akan diseleksi nilai paling kecil yaitu 0,268

0,268 adalah nilai dari testing\_01 yang dijalankan pada tiga device yang berbeda yaitu asus:0,268, acer:0,321 dan lenovo: 2,306.

* + - * 1. Rata-rata

=AVERAGE(Nilai Awal:Nilai Akhir)

Rumus :

|  |  |  |
| --- | --- | --- |
| Asus | Acer | Lenovo |
| 0,268 | 0,321 | 2,306 |

Dari tiga tabel berikut akan diseleksi nilai rata-rata yaitu 0,965

0,965 adalah nilai dari testing\_01 yang dijalankan pada tiga device yang berbeda yaitu asus:0,268, acer:0,321 dan lenovo: 2,306.

* + - * 1. Lambat

=MAX(Nilai Awal:Nilai Akhir)

Rumus :

|  |  |  |
| --- | --- | --- |
| Asus | Acer | Lenovo |
| 0,268 | 0,321 | 2,306 |

Dari tiga tabel berikut akan diseleksi nilai Maksimal yaitu 2,306

2,306 adalah nilai dari testing\_01 yang dijalankan pada tiga device yang berbeda yaitu asus:0,268, acer:0,321 dan lenovo: 2,306.

Dari data tabel 5.4 dan 5.5 berikutnya akan di olah dalam bentuk grafik seperti gambar berikut :

Gambar 5. 43 Grafik perbandingan device

Gambar di atas menjelaskan perbandingan dari tiga device yang berbeda spesifikasi, dalam grafik di atas device asus mempunyai grafik waktu paling cepat sedangkan lenovo mempunyai waktu paling lama, Dari grafik di atas cepat atau lambatnya proses testing dipengaruhi oleh spesifikasi dari laptop yang digunakan, seperti pada laptop asus mendapatkan waktu tercepat karena mempunyai spesifikasi yang tinggi seperti prosesor core i7 gen 8 dengan ram 8gb dan grafik card intel UHD+NVIDIA MX230 , spesifikasi berikut sangat mempengaruhi kinerja dari proses rendering testcode.

Gambar 5. 44 Garfik perbandingan total testcode

Gambar di atas menjelaskan proses testing dari setiap testcode.

# BAB VI. HASIL PENGUJIAN DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil pengujian dan pembahasan dari pengerjaan topik *Basic Aplications Flutter*  pada iCLOP.

## Hasil Pengujian

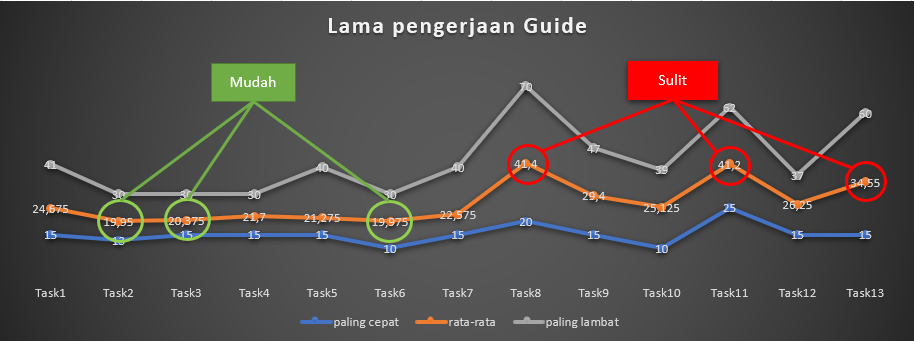
Pengujian dilakukan kepada empat puluh orang *volunteer.* Hasil dari pengujian aplikasi pembelajaran topik *Basic Aplications Flutter* yaitu empat puluh orang *success* mengerjakan topik pembelajaran dengan waktu 7 hari. Dalam mengunggah hasil kerja dapat memilih cara pengunggahan.

Cara pengunggahan yang pertama adalah *multi* *file*s, 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.

## Pembahasan Hasil Pengujian

Berdasarkan hasil uji coba kepada empat puluh orang mahasiswa yang mengerjakan topik *Basic Aplications Flutter* selama tujuh hari. Didapatkan empat puluh orang mahasiswa kelas 4 politeknik negeri malang dapat menyelesaikan dengan berhasil

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



Gambar 6. 1 Hasil pengerjaan topik Basic Aplications Flutter

Dari grafik tersebut terdapat tiga jenis garis. Untuk garis berwarna silver menunjukkan waktu terlama dalam menyelesaikan setiap *task* yang ada pada topik. Dan untuk garis berwarna biru menunjukkan waktu tercepat dalam menyelesaikan setiap *task* pada topik. Sedangkan untuk garis berwarna orange 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, pengambilan data paling mudah dan paling sulit ditentukan menurut jumlah rata-rata waktu pengerjaan seperti gambar 6.2

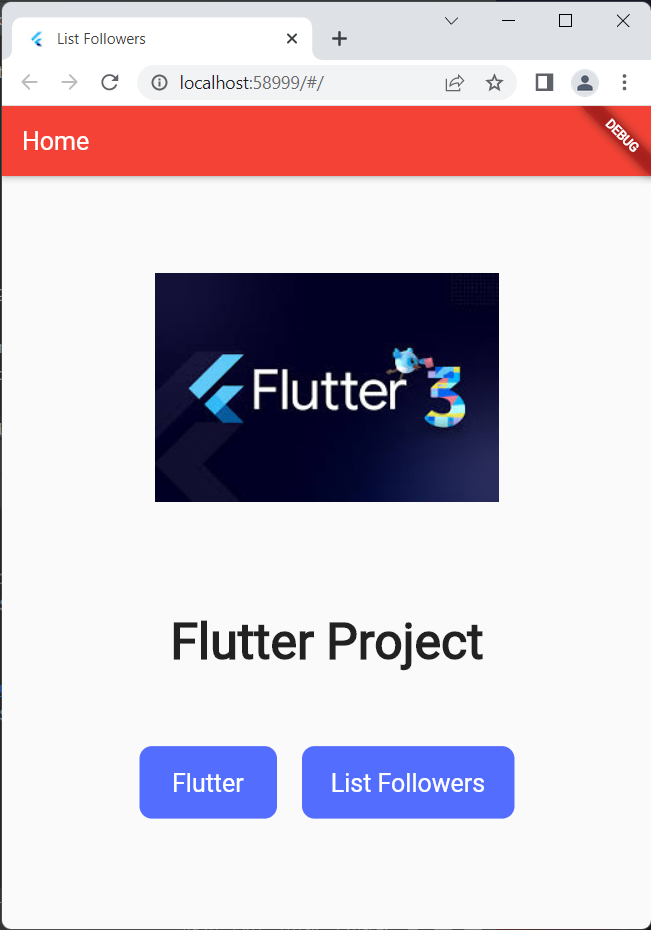
Gambar 6. 2 Klasifikasi rata-rata pengerjaan guide

Dari rata-rata waktu terdapat waktu yang paling cepat bisa disimpulkan sebagai task yang paling mudah yaitu *guide* A1X.06 dan A1X.02. *Guide* A1X.06 merupakan *guide* tentang pembuatan *border* dan *padding*. *Guide* A1X.02 merupakan *guide* tentang membuat *textSpan.* Penyebab pengerjaan *guide* A1X.06 dan A1X.02 dikarenakan pada guide tersebut merupakan basic dan paling mudah dikerjakan oleh mahasiswa.

Sedangkan untuk waktu rata-rata yang paling lambat dapat disimpulkan sebagai *task* paling sulit yaitu *task* B1X.01,B1X.02, B1X.04 dan B1X.06 karena pada guide B1X banyak melakukan perubahan *source code* baik pada halam an maupun pada kerangka aplikasi contohnya migrasi *StatelessWidgets* ke *StatefullWidgets* dan menambahkan *state management provider* pada aplikasi *listFollowers.*

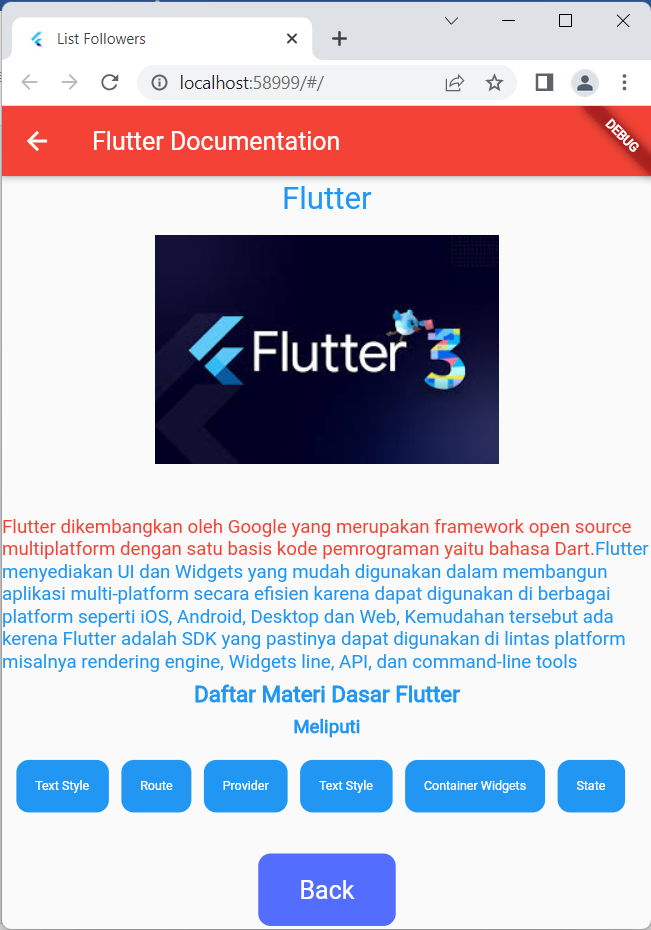
## Contoh Tampilan Aplikasi *List Followers*

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



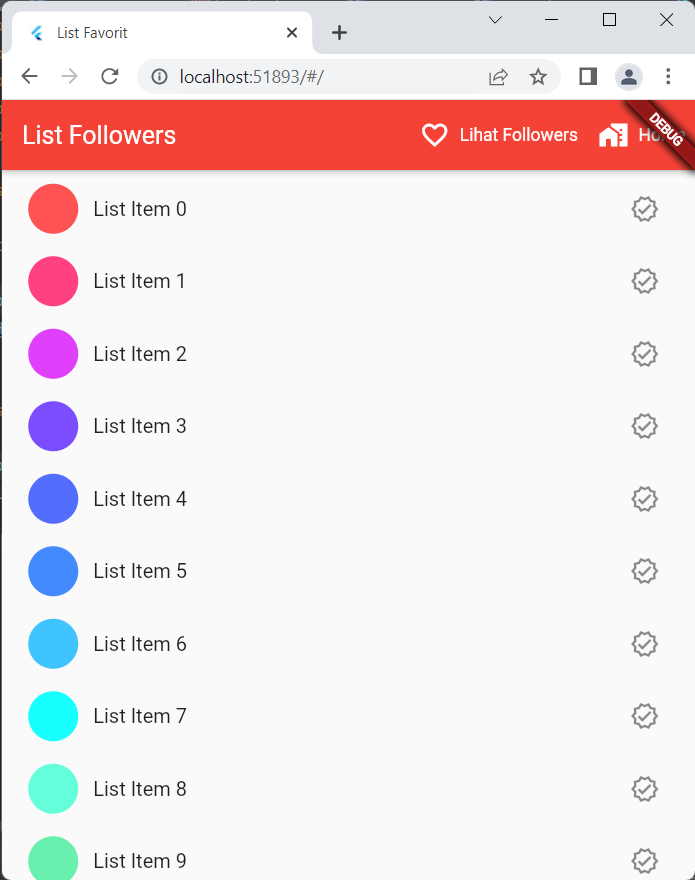
Gambar 6. 3 Home

Halaman Home menampilkan gambar *flutter*  dan dua *button*  yaitu *Flutter* dan  *ListFollowers* yang akan di jelaskan pada gambar 6.4 dan 6.5



Gambar 6. 4 Halaman Flutter

Pada gambar diatas menjelaskan tentang tampilan halaman flutter yang digunakan untuk pembelajaran textWidgets yang meliputi: *TextRich, TextSpan, TextStyle,* dan border.



Gambar 6. 5 Halaman List Followers

Pada gambar di atas menjelaskan tentang halaman List Followers yang meliputi : *IconWidgets, BorderDecorations, ScrollViews, ListItem*  dan *State Management Provider.*

## Umpan Balik Mahasiswa

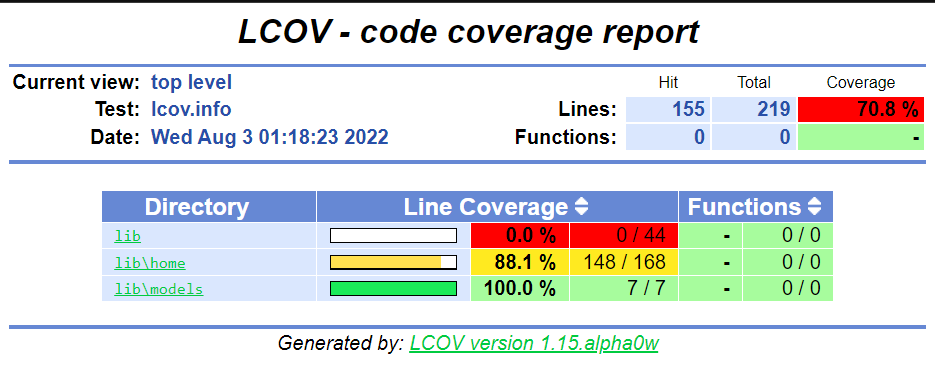
Hasil dari umpan balik mahasiswa yang berhasil dikumpulkan. Komentar tersebut terdiri dari komentar positif dan komentar saran dari setiap task pada topik. Berikut adalah kumpulan umpan balik dari mahasiswa:

Komentar

* + - Mudah
    - Faham
    - Sulit

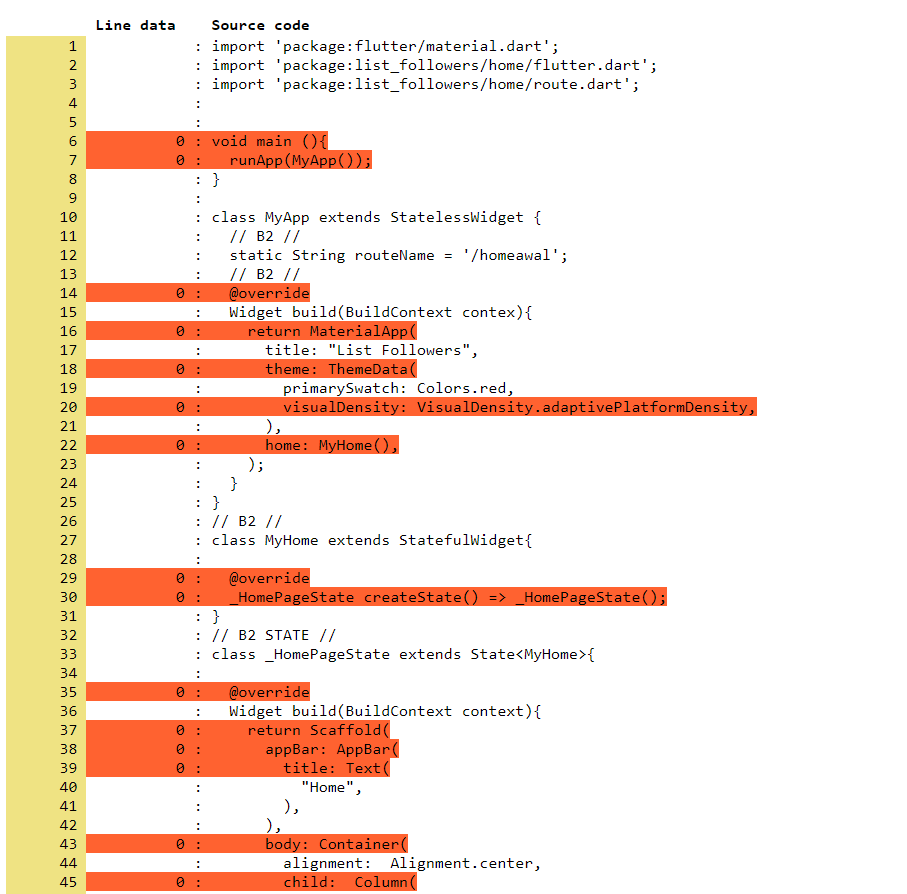
## Pengujian Coverage

Pengujian *coverage* dilakukan untuk menguji sitem aplikasi serta mevalidasi test code aplikasi list followers, *Code coverage* mempunyai tiga pendekatan yaitu : *Requirement Coverage, Code Coverage* dan *Test Coverage*. Hasil pengujian dari *code coverage* ini dapat dijadikan *standart* dalam mevalidasi test code dari aplikasi yang di bangun.Semakin tinggi hasil *coverage* akan semakin bagus dan sebaliknya, perhatikan gambar berikut.



Gambar 6. 6 Hasil Testing Code menggunakan LCOV Code Coverage

Pada gambar di atas menjelaskan hasil pengujian dari code coverage yang menunjukan grafik presentase coverage sebesar 70.8% meliputi 3 directory yaitu lib, lib/home dan lib/models. Dari 3 directory tersebut didapatan hasil Hit (jumlah test code tervalidasi) sebesar 155 dari total lines code sebesar 219, Dari total Hit menunjukan presentase yang berbeda-beda dan dari presenntase tersebut dapat dijelaskan pada gambar 6.7, 6.8 dan 6.9.



Gambar 6. 7 Test Code Coverage Directory Lib

Pada gambar di atas menjelaskan Line Data dan Source Code, Pada line data berisi fungsi yang saling berkaitan dengan line data lainya, jika data 0 maka teks akan berwarna merah karena tidak ada line data yang digunakan dan jika line

# 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.

## Kesimpulan

Setelah berhasil membuat topik pembelajaran *Basic Aplications Flutter* berdasarkan tahap analisis, perancangan serta implementasi. Untuk melakukan evaluasi dari topik pembelajaran ini. Telah dilakukan pengujian kepada empat puluh orang mahasiswa dari Jurusan Teknologi Informasi. Berdasarkan hasil pengujian yang dilakukan selama tujuh hari. Mendapatkan hasil bahwa empat puluh orang mahasiswa mendapatkan hasil *PASSED*.

Seluruh mahasiswa yang telah diuji memberikan umpan balik dengan tiga kategori yaitu mudah, faham dan sulit. Dari hasil tersebut mengkonfirmasi bahwa topik *Basic Aplications Flutter* pada platform iCLOP sangat memudahkan mahasiswa dalam melakukan proses pembelejaran secara mandiri.

## Saran

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

1. Dalam pembelajaran berikutnya dilakukan pembuatan guide secara interaktif dan lebih banyak materi flutter dalam pembelajaran.
2. Disediakan fitur *plagiarism detection* untuk mendeteksi kecurangan dalam proses pengerjaan topik. Yang dapat mendeteksi dari kemiripan *source code*, ukuran *file*.

# DAFTAR PUSTAKA

Haris, Z. A. (2021). *KOMPARASI FLUTTER DAN REACT NATIVE DALAM PENGEMBANGAN PERANGKAT BERGERAK*. 86.

Haryanto, E., & Setiawan, M. A. M. (2021). *Perancangan Sistem Informasi Pembelajaran Praktikum Online Multiplatform Berbasis Framework Flutter*. 8.

Nasution, M. R., & Nasution, L. S. (2021). Konsep Pendidikan Mobile Learning: Layanan Konseling. *Journal of Digital Learning and Education*, *1*(2), 62–71. https://doi.org/10.52562/jdle.v1i2.238

Prayugha, A. W. (2021). *IMPLEMENTASI AUGMENTED REALITY SEBAGAI MEDIA PROMOSI UNIVERSITAS SATYA NEGARA INDONESIA BERBASIS ANDROID MENGGUNAKAN METODE MARKER BASED TRACKING*. 6.

Purba, F. F. (2021). *RANCANG BANGUN APLIKASI PROFIL KARTU GRAFIS BERBASIS ANDROID MENGGUNAKAN FLUTTER*. 8.

Sapari, Y., Suhara, R. B., & Nurhidayat, M. (2021). Pengaruh Penggunaan Smartphone Terhadap Peningkatan Pengetahuan Umum Mahasiswa Prodi Teknik Informatika Universitas Muhammadiyah Cirebon. *Network Media*, *4*(1), 51–58. https://doi.org/10.46576/jnm.v4i1.1153

Satria, S. A., Raya, K. J., & Raya, K. P. (2021). *RANCANG BANGUN APLIKASI CATATAN SEDERHANA ANDROID “CATATAN KAKI” BERBASIS FLUTTER*. 9.

Septianl, A. N. (2021). *IMPLEMENTASI FRAMEWORK FLUTTER UNTUK PENGADUAN MAHASISWA UNIVERSITAS XYZ*. 16.

Sirajuddin, Z., & Dahar, D. (2021). *PENGGUNAAN SMARTPHONE SEBAGAI IMPLEMENTASI TEKNOLOGI INFORMASI DAN KOMUNIKASI DALAM PROSES BELAJAR MENGAJAR MAHASISWA PERTANIAN DI GORONTALO*. 8.

Syaifudin, Y. W., Funabiki, N., & Kuribayashi, M. (2021). *Implementation and Performance Evaluation of Unit Testing for Student’s Answer Validation in Android Programming Learning Assistant System*. 7.

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

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

# LAMPIRAN

**A1X0X\_BasicUI&Widged\_Test\_0X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
  
Widget MyApp(BuildContext context) {  
 return MaterialApp(  
 home: Scaffold(  
 body: Center(  
 ),  
 ),  
 );  
}  
  
void main() {  
  
 group('Kinds of text', () {  
 testWidgets('Find Title Aplication', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 title: 'List Followers',  
 theme: ThemeData(  
 primarySwatch: Colors.*red*,  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 ),  
 ));  
 final Finder title = find.text('List Followers');  
 });  
 testWidgets('Find Title AppBar', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: const Text('Home'),  
 ),  
 )));  
 final Finder title = find.text('Home');  
 });  
 testWidgets('find text', (tester) async{  
 await tester.pumpWidget(Center());  
  
 expect(find.text('Flutter Project'), findsNothing);  
  
 });  
 });  
}

**A1X01\_BasicUI&Widged\_Test\_01X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:list\_followers/home/flutter.dart';  
  
void main() {  
 group('Test Output Text', () {  
 testWidgets('Output Text ', (WidgetTester tester) async {  
 // memanggil pumpwidget pada MyApp//  
 await tester.pumpWidget(firstscreen());  
 // Mencari parameter textspan //  
 var textSpan = find.byType(TextSpan);  
 // nothing yang berarti dinamik/  
 expect(textSpan, findsNothing);  
 });  
 testWidgets('Text Content ', (WidgetTester tester) async {  
 await tester.pumpWidget(  
 const DefaultTextStyle(  
 style: TextStyle(  
 fontSize: 15,  
 ),  
 child: Text.rich(  
 TextSpan(  
 text: 'Flutter',style: TextStyle(fontSize: 25,color: Colors.*blue*),  
 children: <TextSpan>[  
 TextSpan(  
 text: 'Isi text 1',  
 style: TextStyle(fontSize: 15,color: Colors.*red* ),  
 ),  
 TextSpan(  
 text: 'Isi text 2',  
 style: TextStyle(fontSize: 15,color: Colors.*blue* ),  
 ),  
 ],  
 ),  
 textDirection: TextDirection.ltr,  
 ),  
 ),  
 );  
  
 final RichText text = tester.firstWidget(find.byType(RichText));  
 expect(text, isNotNull);  
 expect(text.text.style.fontSize, 15);  
 });  
 });  
}

**A1X02\_BasicUI&Widged\_Test\_02X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:list\_followers/home/flutter.dart';  
  
void main() {  
 group('Test Output Text', () {  
 testWidgets('Output TextRich ', (WidgetTester tester) async {  
 // memanggil pumpwidget pada MyApp//  
 await tester.pumpWidget(firstscreen());  
 // Mencari parameter textRich //  
 var textRich = find.byType(TextSpan);  
 // nothing yang berarti dinamik/  
 expect(textRich, findsNothing);  
 });  
 testWidgets('Text Content ', (WidgetTester tester) async {  
 await tester.pumpWidget(  
 const DefaultTextStyle(  
 style: TextStyle(  
 fontSize: 15,  
 ),  
 child: Text.rich(  
 TextSpan(  
 text: 'Flutter',style: TextStyle(fontSize: 25,color: Colors.*blue*),  
 children: <TextSpan>[  
 TextSpan(  
 text: 'Daftar Materi Dasar Flutter',style: TextStyle(color: Colors.*blue*,fontWeight: FontWeight.*bold*,fontSize: 18)  
 ),  
 TextSpan(  
 children: const <TextSpan>[  
 TextSpan(text: 'Meliputi',  
 style: TextStyle(color: Colors.*blue*,fontWeight: FontWeight.*bold*,fontSize: 15)  
 ),  
 ],  
 ),  
 ],  
 ),  
 textDirection: TextDirection.ltr,  
 ),  
 ),  
 );  
  
 final RichText text = tester.firstWidget(find.byType(RichText));  
 expect(text, isNotNull);  
 expect(text.text.style.fontSize, 15);  
 });  
 });  
}

**A1X03\_BasicUI&Widged\_Test\_03X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter/widgets.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
  
void main() {  
 group('Text Style', () {  
 testWidgets('Find Title TextStyle', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 title: 'Home',  
 theme: ThemeData(  
 primarySwatch: Colors.*red*,  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 ),  
 ));  
 final Finder title = find.text('Home');  
 });  
 testWidgets('Find Title AppBar', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: const Text('List Followers'),  
 ),  
 )));  
 final Finder title = find.text('List Followers');  
 });  
 testWidgets('Test textStyle', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 home: Scaffold(  
 body: Column(  
 children: <Widget>[  
 Container(  
 padding: EdgeInsets.all(25),  
 decoration: BoxDecoration(  
 borderRadius: BorderRadius.circular(10), color: Colors.*red*),  
 child: Text(  
 'Hello World!',  
 style: TextStyle(color: Colors.*white*, fontSize: 20.0),  
 )),  
 ],  
 )),  
 ));  
   
 expect(find.byElementType(BoxDecoration), findsNothing);  
 });  
 });  
}

**A1X04\_BasicUI&Widged\_Test\_04X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter/widgets.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
  
void main() {  
 group('Button Test', () {  
 testWidgets('Find Title TextStyle', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 title: 'List Followers',  
 theme: ThemeData(  
 primarySwatch: Colors.*red*,  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 ),  
 ));  
 final Finder title = find.text('List Followers');  
 });  
 testWidgets('Find Title AppBar', (WidgetTester tester) async {  
 await tester.pumpWidget(MaterialApp(  
 home: Scaffold(  
 appBar: AppBar(  
 title: const Text('Home'),  
 ),  
 )));  
 final Finder title = find.text('Home');  
 });  
 testWidgets('TextButton Clicked', (tester) async {  
 await tester.pumpWidget(new MaterialApp(  
 title: 'Home',  
 theme: ThemeData(  
 primarySwatch: Colors.*red*,  
 visualDensity: VisualDensity.*adaptivePlatformDensity*,  
 ),  
 home: Scaffold(  
 appBar: AppBar(  
 title: const Text('Kinds of text'),  
 ),  
 //Text//  
 body: Column(  
 children: <Widget> [  
 const SizedBox(height:80,width: 5000,),  
 const SizedBox(height:20),  
 Container(  
 padding: EdgeInsets.all(15),  
 decoration: BoxDecoration(  
 borderRadius: BorderRadius.circular(10),  
 color: Colors.*blueAccent*),  
 child: TextButton(  
 onPressed: () {},  
 child: const Text('Flutter'),  
 ),  
 ),: () {},  
 child: const Text('List Followers'),  
 ),  
 ),  
 ],  
 )  
 ),  
 ));  
  
 await tester.pump();  
  
 });  
 });  
}

**B1X01\_BasicUI&Widged\_Test\_01X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:provider/provider.dart';  
import 'package:list\_followers/models/followers.dart';  
import 'package:list\_followers/home/followers.dart';  
import 'package:list\_followers/home/home.dart';  
  
Followers followersList;  
  
Widget createFollowersScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) {  
 followersList = Followers();  
 return followersList;  
 },  
 child: MaterialApp(  
 home: FollowersPage(),  
 ),  
 );  
  
Widget createHomeScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) => Followers(),  
 child: MaterialApp(  
 home: HomePages(),  
 ),  
 );  
  
void main() {  
 group('test widget halaman Followers', () {  
 testWidgets('test daftar kosong',  
 (tester) async {  
 await tester.pumpWidget(createFollowersScreen());  
  
 // verifikasi text pechloader muncul  
 expect(find.text('Tambahkan Dulu'), findsOneWidget);  
 });  
 testWidgets('test Uji ListView', (tester) async {  
 await tester.pumpWidget(createHomeScreen());  
 //menguji listview tampil  
 expect(find.byType(ListView), findsOneWidget);  
 });  
 });  
}

**B1X02\_BasicUI&Widged\_Test\_02X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:provider/provider.dart';  
import 'package:list\_followers/models/followers.dart';  
import 'package:list\_followers/home/followers.dart';  
import 'package:list\_followers/home/home.dart';  
  
Followers followersList;  
  
Widget createFollowersScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) {  
 followersList = Followers();  
 return followersList;  
 },  
 child: MaterialApp(  
 home: FollowersPage(),  
 ),  
 );  
  
Widget createHomeScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) => Followers(),  
 child: MaterialApp(  
 home: HomePages(),  
 ),  
 );  
  
void addItems() {  
 for (var i = 0; i < 5; i++) {  
 followersList.add(i);  
 }  
}  
void main() {  
 group('Test widget halaman Followers & Home ', () {  
 testWidgets('test tampilan button',(tester) async {  
 await tester.pumpWidget(createFollowersScreen());  
  
 addItems();  
 await tester.pumpAndSettle();  
 expect(find.byType(ListView), findsOneWidget);  
 });  
  
 });  
}

**B1X03\_BasicUI&Widged\_Test\_03X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:provider/provider.dart';  
import 'package:list\_followers/models/followers.dart';  
import 'package:list\_followers/home/followers.dart';  
import 'package:list\_followers/home/home.dart';  
  
Followers followersList;  
  
Widget createFollowersScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) {  
 followersList = Followers();  
 return followersList;  
 },  
 child: MaterialApp(  
 home: FollowersPage(),  
 ),  
 );  
  
Widget createHomeScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) => Followers(),  
 child: MaterialApp(  
 home: HomePages(),  
 ),  
 );  
  
void addItems() {  
 for (var i = 0; i < 5; i++) {  
 followersList.add(i);  
 }  
}  
void main() {  
 group('Test widget halaman Followers & Home ', () {  
 testWidgets('test IconButton', (tester) async {  
 await tester.pumpWidget(createHomeScreen());  
  
 // cek favorit.  
 expect(find.byIcon(Icons.*verified*), findsNothing);  
  
 // cek menambahkan icon ke favorit.  
 await tester.tap(find.byIcon(Icons.*verified\_outlined*).first);  
 await tester.pumpAndSettle(Duration(seconds: 1));  
  
 // cek tampilan pesan.  
 expect(find.text('Tambah ke Followers.'), findsOneWidget);  
  
  
 });  
 });  
}

**B1X04\_BasicUI&Widged\_Test\_04X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:provider/provider.dart';  
import 'package:list\_followers/models/followers.dart';  
import 'package:list\_followers/home/followers.dart';  
import 'package:list\_followers/home/home.dart';  
  
Followers followersList;  
  
Widget createFollowersScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) {  
 followersList = Followers();  
 return followersList;  
 },  
 child: MaterialApp(  
 home: FollowersPage(),  
 ),  
 );  
  
Widget createHomeScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) => Followers(),  
 child: MaterialApp(  
 home: HomePages(),  
 ),  
 );  
  
void addItems() {  
 for (var i = 0; i < 5; i++) {  
 followersList.add(i);  
 }  
}  
void main() {  
 group('Test widget halaman Followers & Home ', () {  
 testWidgets('test IconButton', (tester) async {  
 await tester.pumpWidget(createHomeScreen());  
  
 // cek favorit.  
 expect(find.byIcon(Icons.*verified*), findsNothing);  
  
 // cek menambahkan icon ke favorit.  
 await tester.tap(find.byIcon(Icons.*verified\_outlined*).first);  
 await tester.pumpAndSettle(Duration(seconds: 1));  
  
 // cek tampilan pesan.  
 expect(find.text('Tambah ke Followers.'), findsOneWidget);  
  
  
 });  
 });  
}

**B1X05\_BasicUI&Widged\_Test\_05X\_test.dart**

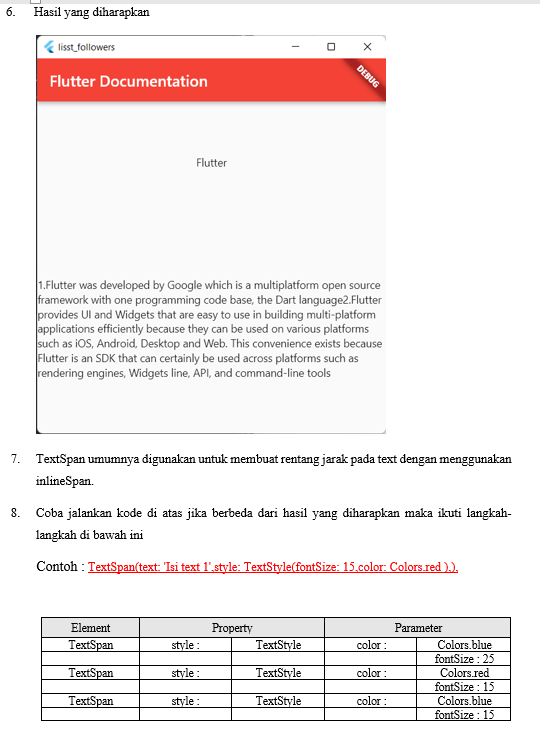
import 'package:test/test.dart';  
import 'package:list\_followers/models/followers.dart';  
  
void main() {  
 group('Test Provider', () {  
  
 var followers = Followers();  
  
 test('Menambah item baru', () {  
 var number = 10;  
  
 // menambah nomor di list  
 followers.add(number);  
  
 // verisikasi telah ditambahkan  
 expect(followers.items.contains(number), true);  
 });  
 test('Menghapus item', () {  
 var number = 15;  
  
 // menambah nomor di list  
 followers.add(number);  
  
 // ferivikasi nomor ditambahkan  
 expect(followers.items.contains(number), true);  
  
 // hapus nomor dari list  
 followers.remove(number);  
  
 // ferivikasi nomor telah di hapus  
 expect(followers.items.contains(number), false);  
 });  
  
 });  
}

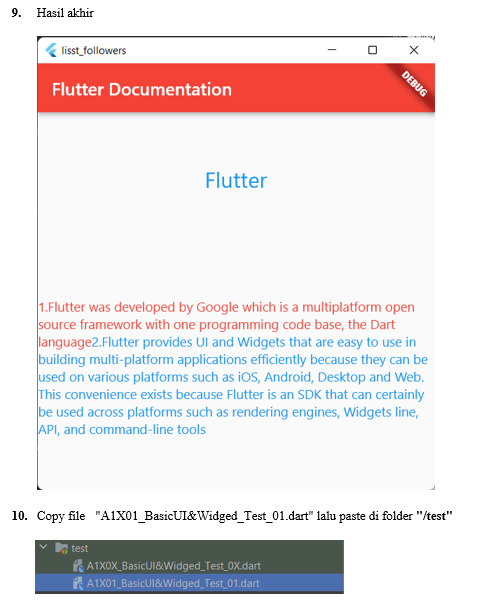
**B1X06\_BasicUI&Widged\_Test\_06X\_test.dart**

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
import 'package:provider/provider.dart';  
import 'package:list\_followers/models/followers.dart';  
import 'package:list\_followers/home/followers.dart';  
import 'package:list\_followers/home/home.dart';  
  
Followers followersList;  
  
Widget createFollowersScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) {  
 followersList = Followers();  
 return followersList;  
 },  
 child: MaterialApp(  
 home: FollowersPage(),  
 ),  
);  
  
Widget createHomeScreen() => ChangeNotifierProvider<Followers>(  
 create: (context) => Followers(),  
 child: MaterialApp(  
 home: HomePages(),  
 ),  
);  
  
void addItems() {  
 for (var i = 0; i < 5; i++) {  
 followersList.add(i);  
 }  
}  
void main() {  
 group('Test widget halaman Followers & Home ', () {  
 testWidgets('test Scrolling', (tester) async {  
 await tester.pumpWidget(createHomeScreen());  
  
 // cek item "0" tampil.  
 expect(find.text('List Item 0'), findsOneWidget);  
  
 // scrolling down.  
 await tester.fling(find.byType(ListView), Offset(0, -200), 3000);  
 await tester.pumpAndSettle();  
  
 // cek item "0" hilang.  
 expect(find.text('List Item 0'), findsNothing);  
 });  
 testWidgets('test Remove Button', (tester) async {  
 await tester.pumpWidget(createFollowersScreen());  
  
 addItems();  
 await tester.pumpAndSettle();  
  
 // get jumlah total item  
 var totalItems = tester.widgetList(find.byIcon(Icons.*close*)).length;  
  
 // hapus satu item.  
 await tester.tap(find.byIcon(Icons.*close*).first);  
 await tester.pumpAndSettle();  
  
 // cek item dihapus  
 expect(tester.widgetList(find.byIcon(Icons.*close*)).length,  
 lessThan(totalItems));  
  
 // Verify if the appropriate message is shown.  
 expect(find.text('Hapus Followers.'), findsOneWidget);  
 });  
 });  
}

testWidgets('test Remove Button', (tester) async {  
 await tester.pumpWidget(createFollowersScreen());  
  
 addItems();  
 await tester.pumpAndSettle();  
  
 // get jumlah total item  
 var totalItems = tester.widgetList(find.byIcon(Icons.*close*)).length;  
  
 // hapus satu item.  
 await tester.tap(find.byIcon(Icons.*close*).first);  
 await tester.pumpAndSettle();  
  
 // cek item dihapus  
 expect(tester.widgetList(find.byIcon(Icons.*close*)).length,  
 lessThan(totalItems));  
  
 // Verify if the appropriate message is shown.  
 expect(find.text('Hapus Followers.'), findsOneWidget);  
 });  
 });  
}

Hasil Guide 1

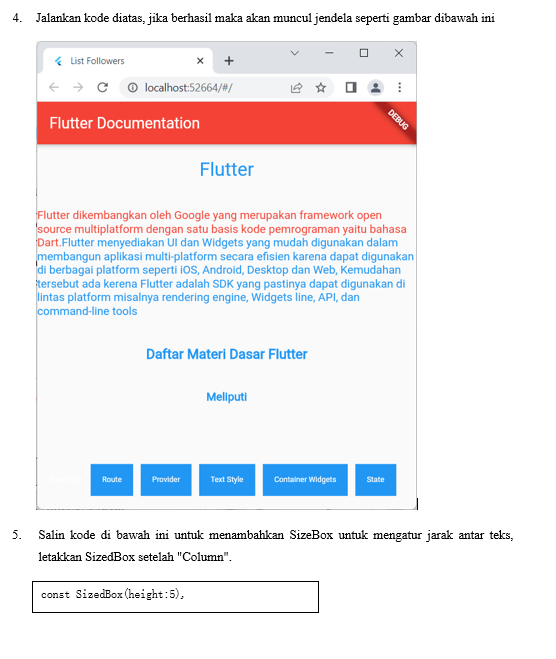


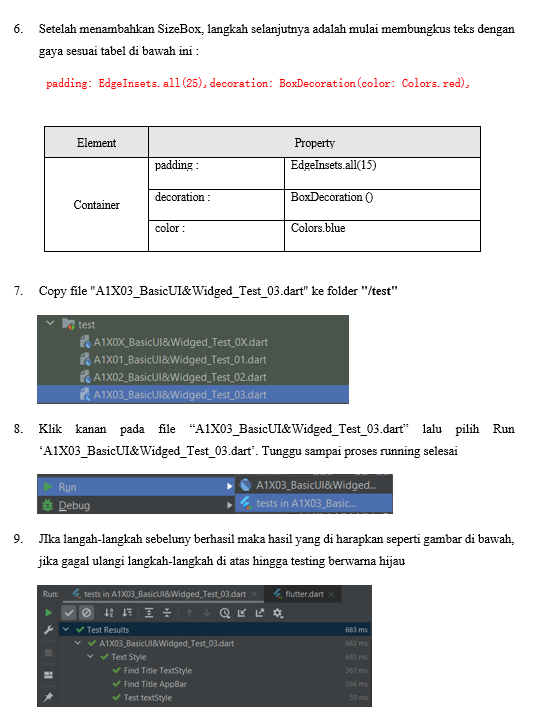


Hasil Guide 2

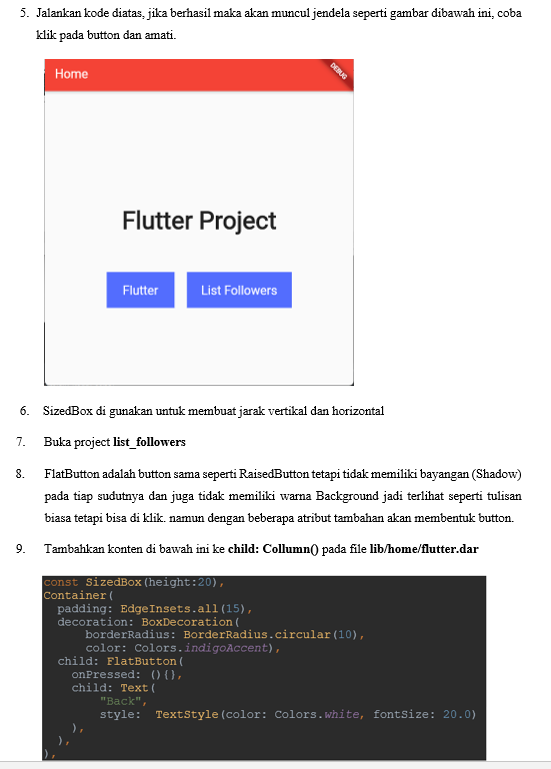


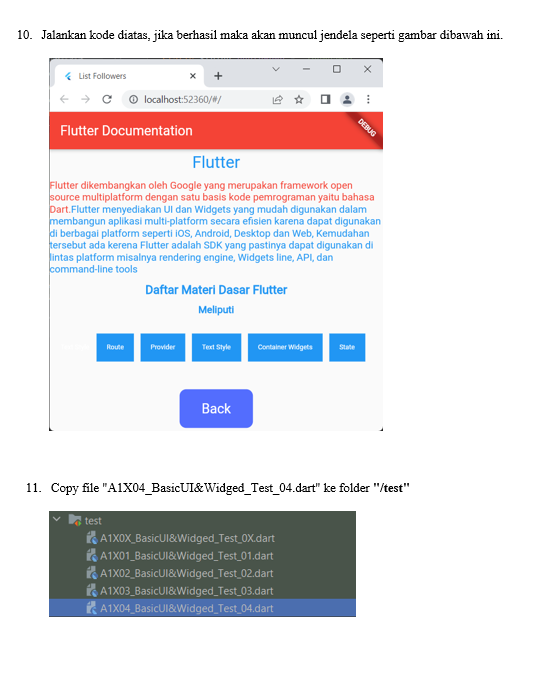
Hasil Guide 3





Hasil Guide 4

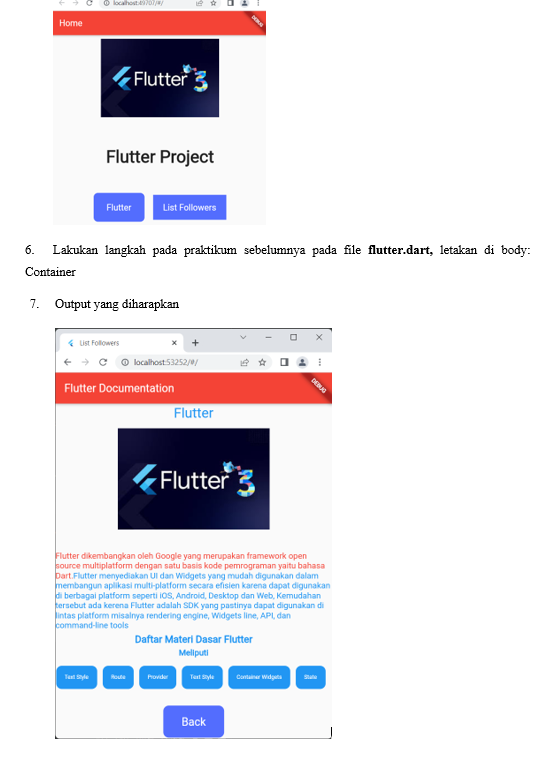




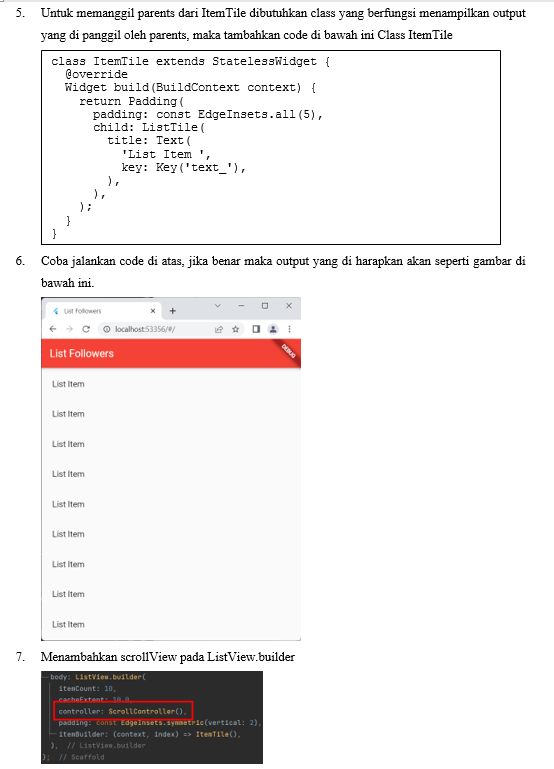
Hasil Guide 5

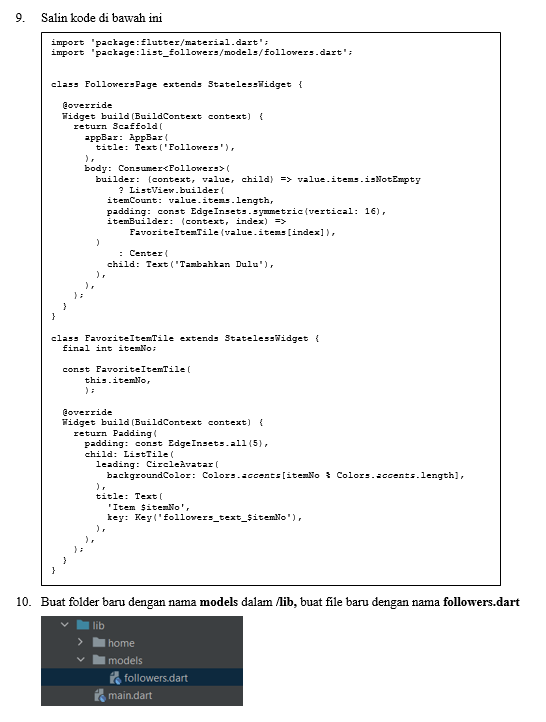


Hasil Guide 6

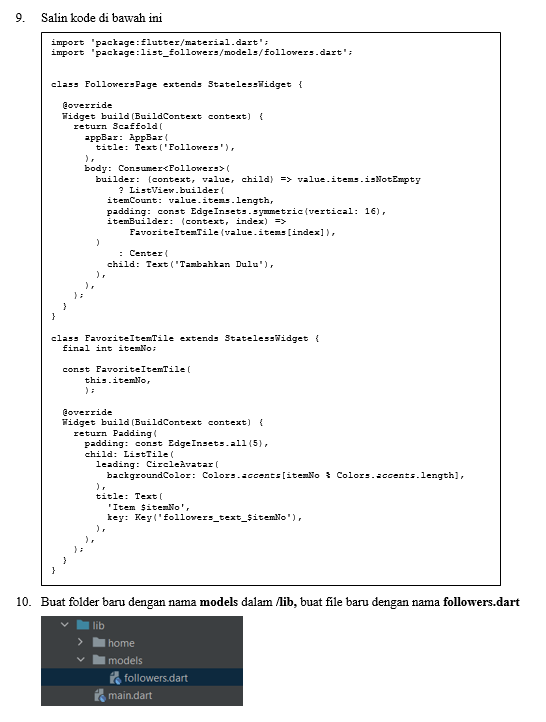


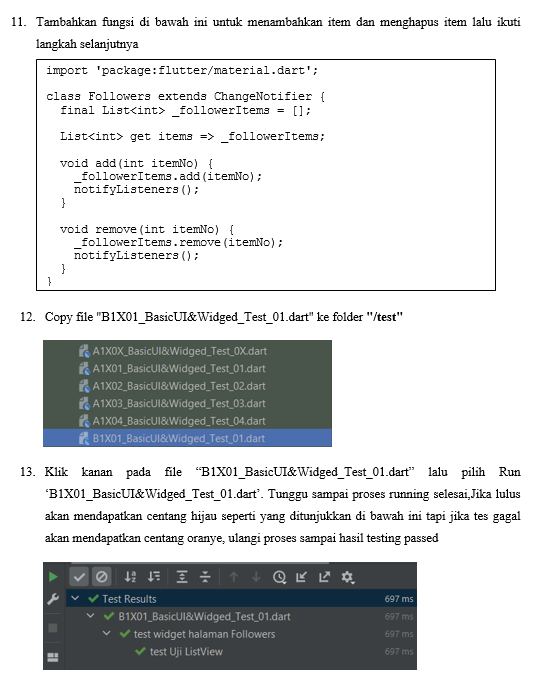
Hasil Guide 7





Hasil Guide 8

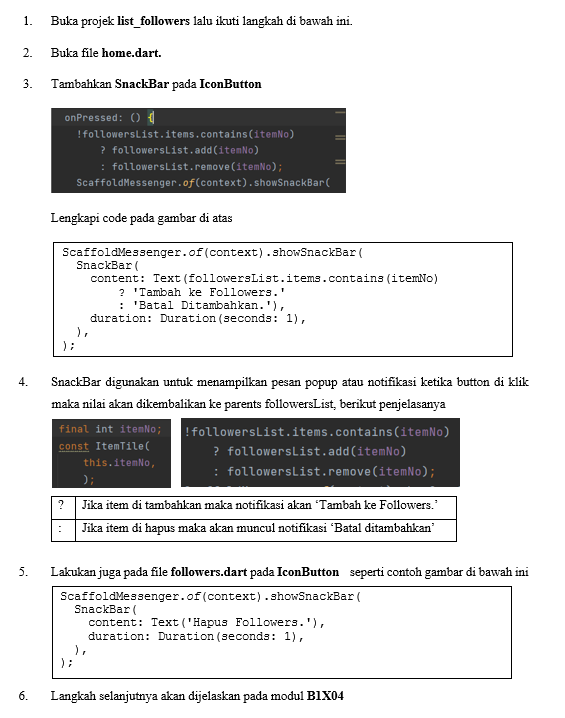




Hasil Guide 9

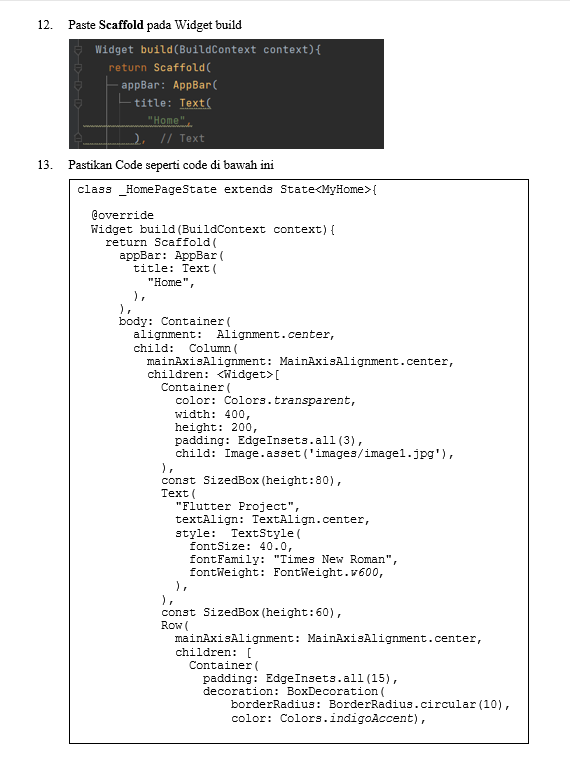


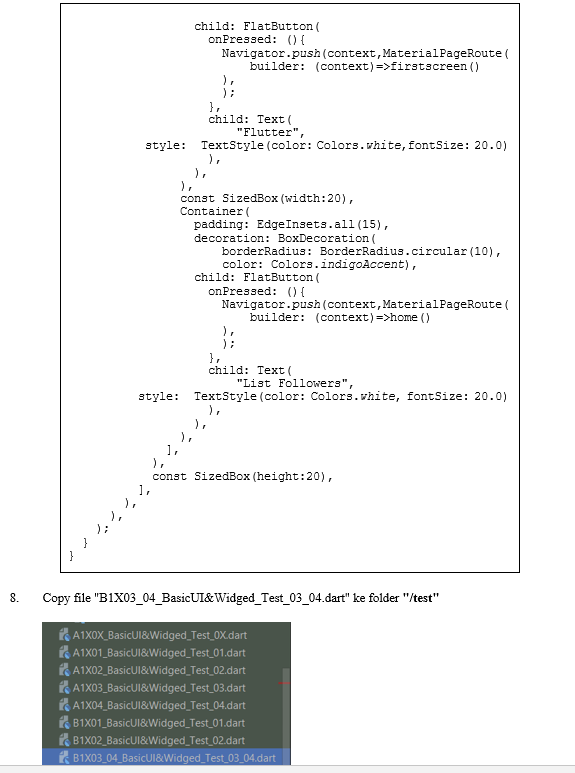
Hasil Guide 10



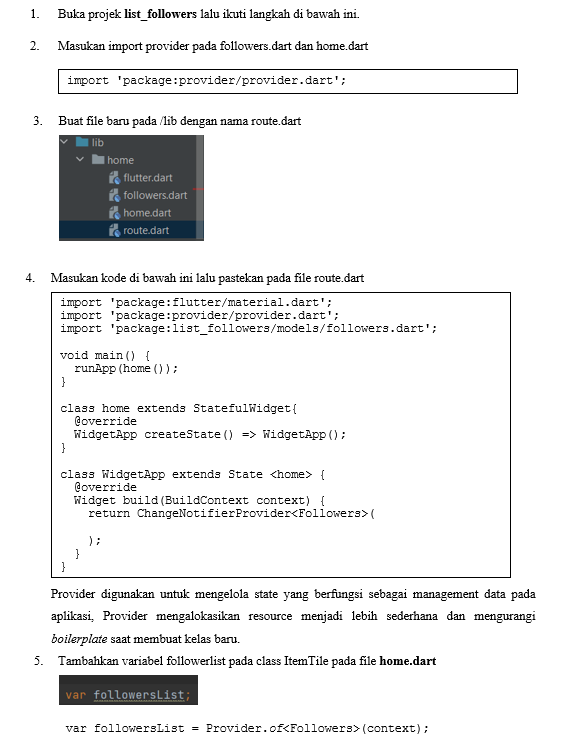
Hasil Guide 11

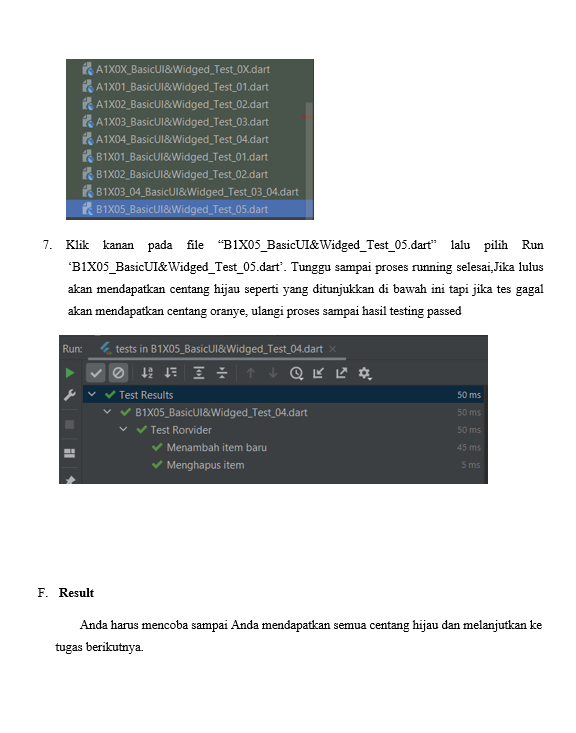






Hasil Guide 12





Hasil Guide 13



