

Pengembangan Aplikasi Kursus Digital menggunakan *Framework Multiplatform* dan *Clean Architecture*

<http://dx.doi.org/10.28932/jutisi.v10i2.8402>

Riwayat Artikel

Received: 16 Februari 2024 | Final Revision: 16 Agustus 2024 | Accepted: 16 Agustus 2024

Creative Commons License 4.0 (CC BY – NC)



Erico Darmawan Handoyo [✉]#1, Sulaeman Santoso ^{*}2, Daniel Jahja Surjawan ^{#3}

[#] *SI Teknik Informatika, Universitas Kristen Maranatha
Jl. Prof. Drg. Suria Sumantri No. 65, Bandung, 40164, Indonesia*

¹erico.dh@it.maranatha.edu

²sulaeman.santoso@it.maranatha.edu

³daniel.js@it.maranatha.edu

[✉]Corresponding author: erico.dh@it.maranatha.edu

Abstrak — Pandemi 2020 mengajarkan banyak orang bagaimana dunia digital dapat berperan besar pada kehidupan. Salah satu bidang yang sangat terpengaruh adalah dunia pembelajaran. Berbagai jenis *Learning Management System* dan perangkat lunak pembelajaran dikembangkan. Namun, banyak aplikasi pembelajaran ini tidak didesain untuk satu jenis pembelajaran namun untuk mengakomodir pembelajaran secara umum. Hal ini membuat LMS dan perangkat lunak tersebut bersifat fleksibel terhadap berbagai tipe pembelajaran namun juga membuat sistem tersebut menjadi rumit, sulit untuk navigasi ataupun diatur. Karena hal inilah LMS pada umumnya tidak tepat digunakan untuk mengakomodir pembelajaran dalam sebuah kursus. Sebuah kursus berbeda dengan sebuah institusi formal pembelajaran karena sifatnya yang lebih sederhana dan kebutuhan penggunaanya juga berbeda. Penelitian ini akan merancang dan mengembangkan perangkat lunak pembelajaran yang khusus dibuat untuk kursus digital yang setiap materinya disampaikan dalam *video tutorial*. Perangkat lunak ini dikembangkan menggunakan *framework multiplatform* sehingga dapat di gunakan di berbagai *platform*.

Kata kunci— edukasi; kursus digital; *multi platform*; *video tutorial*

Digital Course Application Development Using Multiplatform Framework and Clean Architecture

Abstract — The 2020 pandemic taught many people how the digital world can play a big role in life. One area that is affected is the world of learning. Various types of Learning Management Systems (LMS) and learning software were developed. However, many of these learning applications are not designed for one type of learning but to accommodate learning in general. This makes the LMS and software flexible for different types of learning but also makes the system cumbersome and difficult to navigate or manage. Therefore, LMS is generally not appropriate to accommodate learning in a course. A course differs from a formal institution of learning because it is simpler in nature and the needs of its users are also different. This research will design and develop learning software specifically made for digital courses where each material is presented in video tutorials. This software was developed using a multiplatform framework so that it can be used on various platforms.

Keywords— digital course; education; multi-platform; video tutorial

I. PENDAHULUAN

Salah satu bidang yang mengalami banyak sekali perubahan berkenaan dengan kemajuan teknologi adalah bidang pendidikan. Dari penggunaan teknologi sebagai alat bantu ajar seperti mengganti papan tulis dengan layar proyektor ataupun penggunaan perangkat lunak sebagai alat uji, perkembangan teknologi pendidikan selalu terjadi. Salah satu bidang yang juga menjadi akibat perkembangan ini adalah teknologi untuk platform pembelajaran *online*. Dengan ditemukannya *internet*, sebuah cara baru mengakses pengetahuan ditemukan. Pada awal perkembangan *internet*, pengetahuan yang tersedia masih terpisah dan belum terstruktur. Strukturisasi dari pengetahuan yang ada di *internet* itu merupakan sebuah bahan baku yang amat berharga bagi dunia pendidikan. Platform pembelajaran *online* adalah salah satu usaha untuk memanfaatkan pengetahuan yang dapat disimpan di *internet* untuk proses pembelajaran tertentu.

Platform pembelajaran *online* adalah sebuah wadah yang bertujuan membantu proses pembelajaran tanpa memerlukan tatap muka secara langsung. Pada umumnya platform seperti ini menyediakan materi pembelajaran dalam berbagai bentuk yang dapat dipelajari secara asinkronus. Dalam hal ini, platform pembelajaran *online* dapat mengerucut ke dalam dua bentuk besar. Bentuk pertama adalah platform pembelajaran seperti *course management system* atau *learning management system*. Sebuah sistem pengelolaan pembelajaran memiliki 3 bagian utama yaitu alat bantu ajar, alat komunikasi, dan alat produktivitas [1]. Dalam bentuk ini, platform pembelajaran mengakomodir pembelajaran secara menyeluruh. Platform ini menyediakan bantuan untuk semua tahapan pembelajaran yang mungkin dilewati. Mulai dari penyediaan materi pembelajaran dengan berbagai metode dan bentuk, hingga pengujian dan pengaturan nilai. Beberapa produk platform *online* seperti ini contohnya adalah *Google classroom* [2], *moodle* [3], *udemy* [4], *kognity* [5], *odoo* dan sejenisnya. Dengan menyediakan berbagai macam bentuk dan metode pembelajaran, platform sejenis ini memiliki fleksibilitas yang tinggi dalam menyiapkan pembelajaran dan dapat sesuai untuk banyak jenis bentuk pembelajaran dan tingkat pembelajaran. Sayangnya justru karena sistem seperti ini menyediakan semua jenis pembelajaran, sistem ini menjadi cukup rumit untuk dipelajari. Sebuah sistem pengelolaan pembelajaran yang penuh akan memiliki banyak sekali fitur yang membuat penggunaan sistem seperti ini memerlukan pelatihan ataupun penyesuaian. Selain itu, sistem ini juga menyediakan banyak fitur yang mungkin tidak perlu digunakan untuk skenario pembelajaran yang lebih sederhana seperti sebuah kursus. Sebagai contohnya untuk kursus menjahit *online* biasanya tidak membutuhkan sebuah sistem penilaian yang komprehensif. Kelemahan-kelemahan ini membuat sistem pengelolaan pembelajaran yang lengkap umumnya digunakan untuk skenario yang lebih sederhana atau bagi pendidik pemula yang bukan merupakan bagian dari organisasi edukasi.

Bagi pendidik pemula, akan jauh lebih mudah untuk memulai pembelajaran dengan membagikan konten pembelajaran secara langsung pada target pesertanya. Hal ini dapat dilakukan dengan beberapa pendekatan. Pendekatan paling sederhana adalah dengan menyimpan materi pembelajaran pada sebuah penyimpanan *online* misalkan pada *cloud storage* seperti *google drive* atau *onedrive*. Materi ini lalu akan dibagikan dalam bentuk tautan kepada target peserta kursus untuk dapat diunduh dan dipelajari. Kekurangan dari pendekatan seperti ini salah satunya adalah bahwa tidak adanya sebuah struktur alur pembelajaran yang jelas. Karena sifat dari materi hanya penjelasan perlu ada konten lain di tempat lain untuk mestrukturisasikan proses pembelajaran. Belum lagi permasalahan pembajakan dan karya cipta yang dapat lebih mudah timbul pada penggunaan metode seperti ini. Selain daripada itu, peserta juga tidak dapat mengetahui dengan mudah posisi pembelajarannya masing masing karena tidak ada informasi yang dapat digunakan. Seorang pelajar yang baru belajar sebagian materi lalu untuk beberapa waktu akan mengalami kesulitan untuk kembali belajar karena tidak adanya catatan secara implisit posisi pembelajarannya. Selain dari penggunaan *cloud storage*, saat ini juga marak digunakan media sosial sebagai alat untuk pembelajaran bahkan sebagai pengganti platform pembelajaran *online*. Tentunya penggunaan media sosial sebagai pengganti platform pembelajaran juga memiliki kekurangan. Beberapa media sosial tidak memiliki struktur sama seperti pada penggunaan *cloud storage*. Selain itu juga permasalahan dapat pula muncul dari sisi pembuat konten dimana sistem monetisasi konten pembelajaran pada media sosial tidak tepat untuk format kursus.

Aplikasi ini akan dikembangkan dengan menggunakan *flutter framework*, yaitu salah satu *framework* yang memungkinkan pengembangan aplikasi secara *multiplatform* [6]. *Framework* ini menyediakan jembatan penghubung antara perangkat keras dan sistem operasi dengan aplikasi yang dikembangkan. *Flutter* dikembangkan dengan bahasa *dart* yang dapat digunakan baik untuk *frontend* maupun *backend*. Pada bagian *backend*, *dart* dapat diutilisasi menggunakan *library dart-frog* [7] untuk menyediakan *rest-api* yang kemudian dapat dikonsumsi oleh aplikasi yang memintanya. Penelitian ini akan menggunakan *dart-frog* sebagai *backend* dan *flutter framework* sebagai *frontend*.

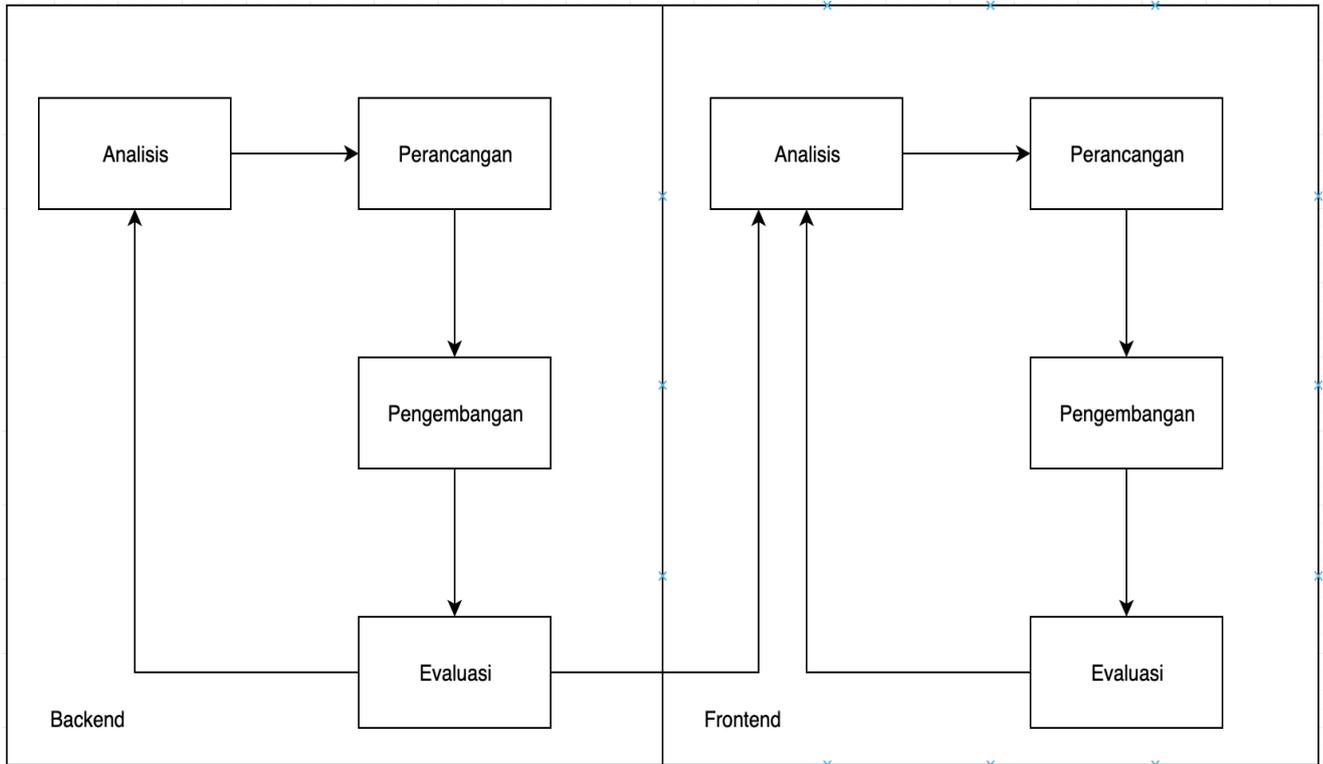
Christian pada tahun 2022 [8] mengembangkan aplikasi kursus *online* berbasis *web* dengan sistem rekomendasi yang membantu siswa mendapatkan pengajaran yang dimiliki namun sistem ini bertujuan untuk keadaan kelas dan bukan kursus. Dimas Purmanto [9] pada tahun 2021 mengembangkan aplikasi kursus *online* berbasis *web service* menggunakan arsitektur *microservices*. Aplikasi ini menggunakan *database* relasional dan tipe pemrograman yang berbeda untuk *backend* dan *frontend* dari aplikasi. Selain dari platform *web* beberapa penelitian juga diarahkan pada platform lain salah satunya adalah jammaludin et al [10] mengembangkan aplikasi untuk kursus pada platform *mobile android*. Kelemahan dari pendekatan ini adalah aplikasi hanya mentarget satu platform saja sehingga tidak fleksibel terhadap perubahan platform yang sering terjadi pada pengguna. Keunggulan dari penelitian ini akan berfokus pada perancangan dan pengembangan perangkat lunak untuk

kursus digital yang bersifat *multiplatform* menggunakan *state management bloc* semua hal yang dapat dilakukan guru maupun pengguna pada aplikasi kursus digital.

II. METODE PENELITIAN

A. Prototyping

Perancangan dan pengembangan menggunakan metode prototyping dengan siklus yang berbeda pada bagian *backend* dan *frontend*. Pada setiap siklus prototipe, akan dilakukan analisis, perancangan, dan pengembangan fitur. Gambar 1 menunjukkan proses pengembangan aplikasi.



Gambar 1. Proses pengembangan aplikasi dan evaluasi

Berikut ini adalah penjelasan apa yang dikerjakan pada setiap tahapan:

Sisi *backend*

1. Analisis
 - a. Identifikasi Kebutuhan: Mengidentifikasi fitur *backend* seperti manajemen pengguna, autentikasi, otorisasi, dan pengelolaan konten kursus.
 - b. Analisis Teknis: Menilai kebutuhan teknis seperti *server*, *database*, dan API.
 - c. Dokumentasi: Menyusun dokumen spesifikasi kebutuhan *backend*.
2. Perancangan
 - a. Desain Arsitektur: Membuat desain arsitektur *backend* dan memilih teknologi.
 - b. Model Data: Merancang model data untuk kursus, pengguna, dan konten.
 - c. Prototipe API: Membuat prototipe API untuk interaksi data.
3. Pengembangan
 - a. Implementasi: Menulis kode dengan dart.
 - b. Manajemen Database: Menggunakan penyimpanan awan dalam bentuk *firebase firestore*.
 - c. Autentikasi dan Otorisasi: Implementasi sistem autentikasi dan otorisasi.
 - d. Pengujian: Melakukan pengujian unit dan integrasi.
4. Evaluasi

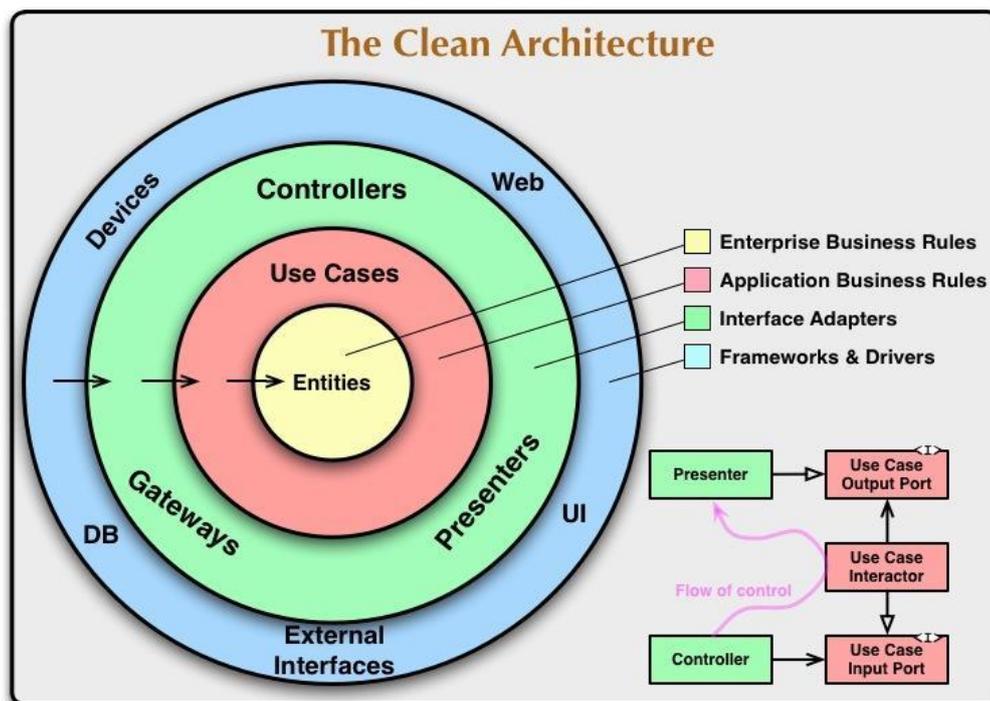
- a. Umpan Balik dan Perbaikan: Mengumpulkan umpan balik, memperbaiki *bug*, dan mengoptimalkan performa.
- b. Dokumentasi Akhir: Menyusun dokumentasi akhir untuk arsitektur *backend*, API, dan skema *database*.

Sisi *frontend*

- 1. Analisis
 - a. Identifikasi Kebutuhan: Menentukan fitur dan kebutuhan pengguna.
 - b. Dokumentasi: Menyusun spesifikasi kebutuhan *frontend*.
- 2. Perancangan
 - a. Desain UI/UX: Membuat desain antarmuka dan pengalaman pengguna.
 - b. Prototipe: Membuat prototipe interaktif.
- 3. Pengembangan
 - a. Implementasi UI: Menulis kode tampilan
 - b. Integrasi API: Menghubungkan *frontend* dengan *backend*.
 - c. Pengujian: Melakukan pengujian fungsional pada komponen UI.
- 4. Evaluasi
 - a. Uji Pengguna: Menguji aplikasi dengan pengguna nyata.
 - b. Feedback: Mengumpulkan dan menerapkan umpan balik.
 - c. Debugging: Memperbaiki *bug* yang ditemukan.
 - d. Dokumentasi Akhir: Menyusun panduan penggunaan dan dokumentasi pengembangan.

B. Entitas dan Use Case Pada Clean Architecture

Sistem kursus digital dengan *framework multiplatform* dikembangkan dengan menggunakan perancangan *clean architecture*. Gambar 2 menunjukkan struktur tersebut.



Gambar 2. Clean architecture

Penggunaan arsitektur ini ditujukan untuk memberikan perangkat lunak. Kemampuan untuk berkembang dengan baik dan mudah. Keuntungan dari *clean architecture* adalah sebagai berikut [11]:

- 1. Sistem tidak terikat oleh *framework* tertentu. Dengan arsitektur seperti ini, *framework* dapat dilepas pasang kepada sistem tanpa melakukan perubahan terlalu besar karena setiap lapisan dari perangkat lunak tidak terikat secara kuat.

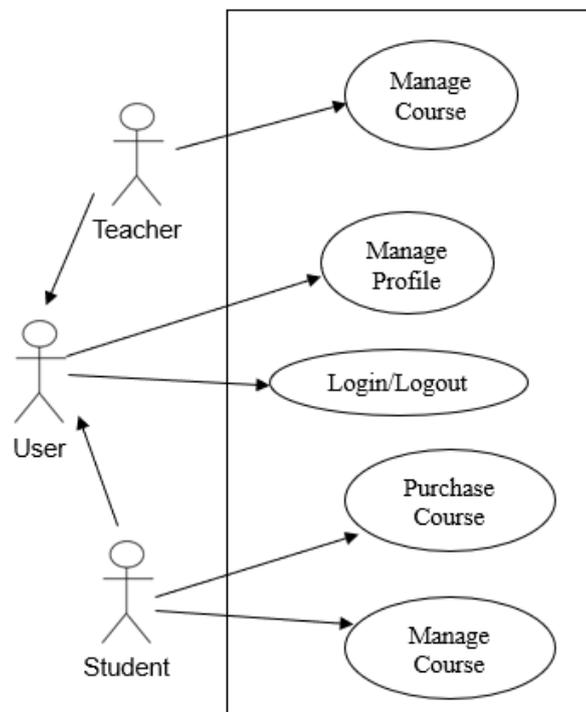
2. Sistem dapat diuji dengan baik. Karena sifatnya yang terbagi bagi, pembuatan test uji dan pelaksanaannya menjadi lebih mudah.
3. Tidak terikat kepada antarmuka pengguna selain dari *framework*. Dengan rancangan seperti ini, antarmuka pengguna pun dapat dilepas pasang sesuai dengan kebutuhan.
4. Tidak terikat kepada *database*. Bahkan untuk *database*, dengan pembagian seperti ini, *database* dapat dilepas pasang.
5. Tidak terikat pada agen eksternal lainnya. Proses bisnis aplikasi pun dapat berubah tanpa perubahan besar pada aplikasi.

Bagian pertama dari sebuah *clean architecture* adalah entitas. Entitas adalah bagian data yang merupakan suatu kesatuan yang terpisah dan berkaitan dengan proses bisnis dari aplikasi. Pada aplikasi kursus digital ini, diterapkan entitas seperti pada tabel 1 berikut:

TABEL 1
ENTITAS APLIKASI KURSUS DIGITAL

No.	Entitas	Keterangan
1	<i>Course</i>	Entitas yang menyimpan sebuah <i>course</i>
2	<i>CourseVideo</i>	Entitas yang mewakili <i>video</i> dalam sebuah <i>course</i>
3	<i>Owned Course</i>	Entitas yang mewakili <i>course</i> yang dimiliki seorang <i>user</i>
4	<i>Payment Method</i>	Entitas yang mewakili metode pembayaran
5	<i>Purchased Course</i>	Entitas yang mewakili <i>course</i> yang hendak dibeli
6	<i>Teacher</i>	Entitas yang mewakili guru kursus
7	<i>Transaction</i>	Entitas yang mewakili transaksi yang terjadi
8	<i>User</i>	Entitas yang mewakili <i>user</i> yang dapat menggunakan kursus

Use case pada sebuah *clean architecture* adalah setiap kemungkinan penggunaan yang dapat dilakukan *user* terhadap perangkat lunak yang dihasilkan. Dalam kasus ini semua hal yang dapat dilakukan guru maupun pengguna di dalam aplikasi kursus digital. Gambar 3 berikut adalah *use case* aplikasi:



Gambar 3. *Use case* diagram aplikasi

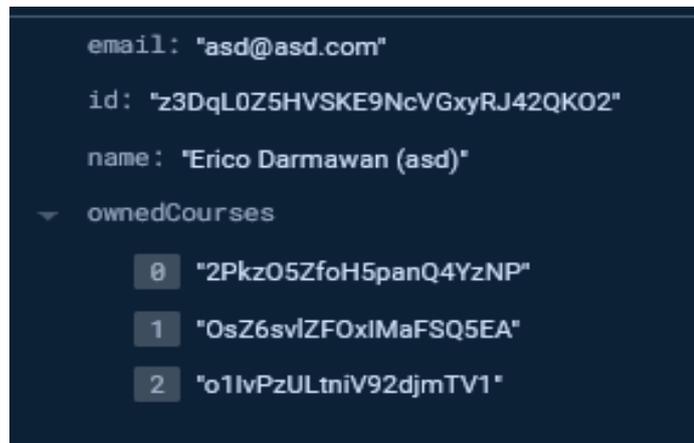
Pada gambar 3 terdapat 3 user yaitu *teacher*, *user*, dan *student*. *Teacher* dapat melakukan *manage course*, *user* dapat melakukan *manage profile* dan *login/logout* aplikasi, *student* dapat melakukan *purchase course* dan *manage course*. Panah dari *teacher* dan *student* ke *user* berarti *user* bisa dibagi menjadi *teacher* dan *student*, beberapa fitur hanya bisa diakses oleh *teacher* dan ada beberapa yang hanya bisa dilakukan *student*.

Controllers memberikan akses dari *use case* kepada *data* yang dibutuhkan. Dalam hal ini, *controller* dibagi menjadi dua bagian yaitu *repository* dan *services*. *Services* menjadi pengantara antara permintaan pengguna melalui *usecase* kepada *repository*. Sedangkan *repository* menjadi jembatan antara *services* dan data asli yang tersimpan. Dengan adanya jembatan jembatan ini maka prinsip *dependency inversion* terpenuhi [12]. Dimana ketergantungan antara bagian dalam perangkat lunak dijembatani sehingga masing masing komponen tidak saling tergantung kepada yang lain tapi melalui jembatan sehingga masing masing dapat berubah tanpa mempengaruhi yang lain.

C. Database

Sistem kursus digital dengan. Menggunakan *clean architecture* berarti *database* yang digunakan untuk perangkat lunak dapat diganti tiap saat. Namun sebagai pengujian diputuskan untuk menggunakan *database* yang bersifat *non-relational* yaitu menggunakan penyimpanan awan dalam bentuk *firebase firestore*. *Database non-relational* ini menyimpan data dalam bentuk kumpulan *key* dan *value* dalam sebuah *collection*. Data pada aplikasi ini dibagi menjadi 4 *collection* besar yaitu *users*, *transactions*, *teacher*, dan *courses*. Masing masing *collection* memiliki struktur sebagai berikut:

1. *Users' collection*: berisi nama, id, dan juga *role* dari *user* tersebut. *Role* ini menentukan apakah *user* tersebut adalah *student* ataupun *teacher*. Pada *student* yang memiliki atau sudah membeli *course*, maka *entry* pada *student* tersebut akan ditambahkan sebuah *field* tambahan yaitu *owned courses* yang adalah sebuah *array* yang berisi semua *course* yang dimilikinya. Contoh *data* pada *user collection* adalah seperti pada gambar 4.
- 2.



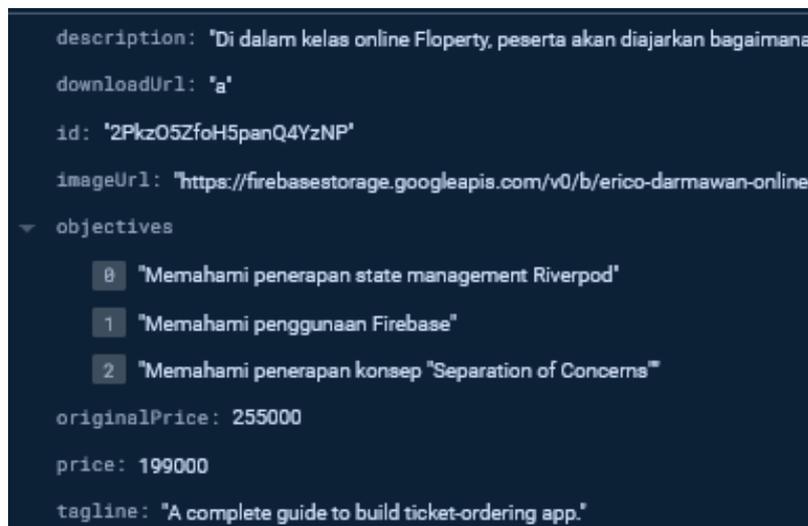
Gambar 4. Data pada *user collection*

3. *Transactions collection*: akan menyimpan semua informasi transaksi yang dibutuhkan. Termasuk di dalamnya adalah informasi pelanggan, informasi pembelian, dan juga *course* yang dibelinya seperti ditunjukkan pada gambar 5.



Gambar 5. Transaction collection

4. *Teachers' collection*: menyediakan informasi dari pengajar, sedangkan *courses* menyediakan informasi kursus yang diberikan. Kursus dipisahkan dari pengajar dengan tujuan agar kursus dapat diajar lebih dari satu pengajar. Gambar 6 menunjukkan informasi yang disimpan di dalam *collection* untuk *courses*.



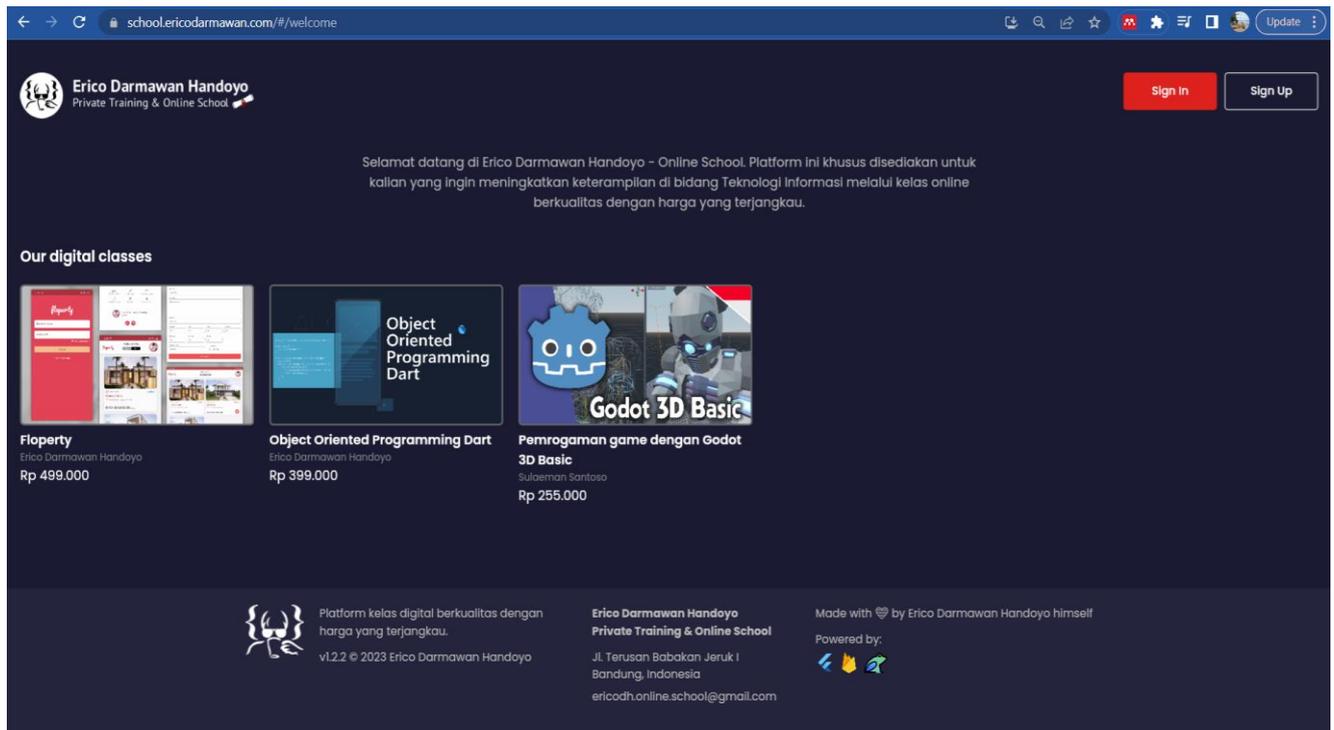
Gambar 6. Courses collection

III. HASIL DAN PEMBAHASAN

A. Tampilan Pengguna

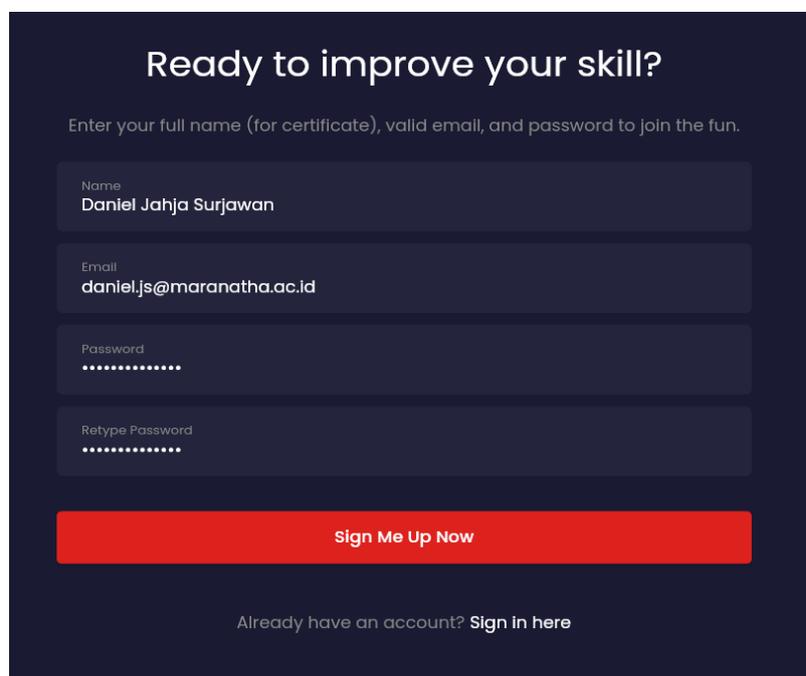
Gambar 7 merupakan tampilan awal dari *website* yang dibuat. Pengguna memasukan tautan <https://school.ericodarmawan.com/> untuk masuk ke halaman ini. Disini pengguna dapat melihat beberapa materi pembelajaran yang ditawarkan dengan harga yang tertera pada setiap materi. Selain itu juga ditampilkan informasi dari kelas pembelajaran seperti nama *online school*, alamat kantor. Untuk pengguna baru diperlukan proses registrasi terlebih dahulu

dengan menekan tombol *Sign up* yang terletak di kanan atas. Dan untuk pengguna yang sudah memiliki akun bisa langsung masuk dengan klik tombol *Sign in* yang terletak di sebelah tombol *Sign up*.



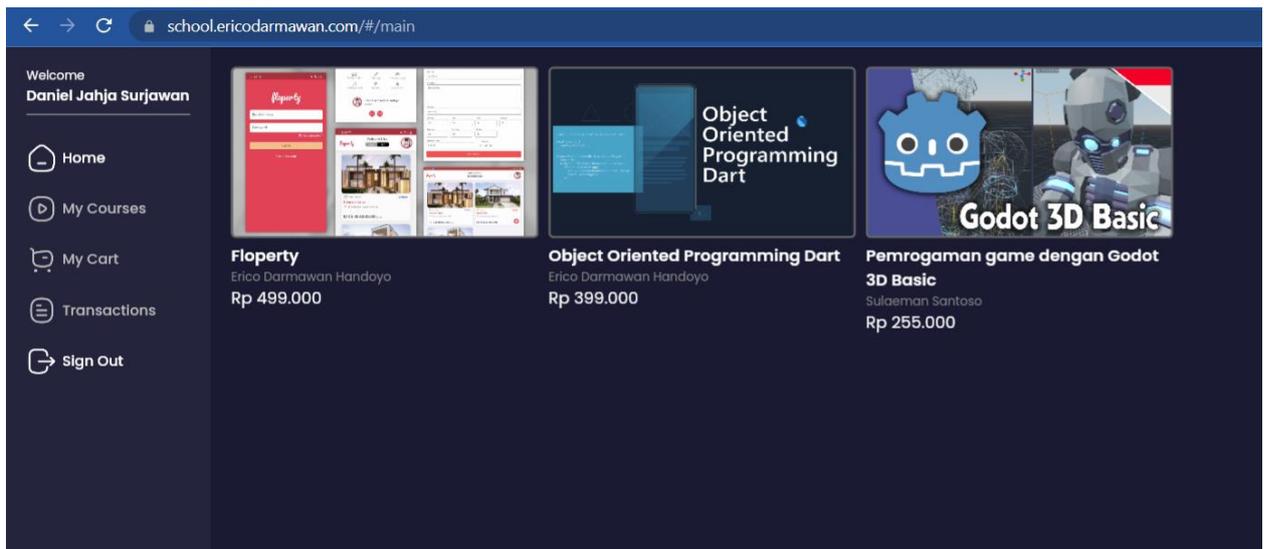
Gambar 7. Tampilan awal aplikasi

Gambar 8 adalah tampilan saat pengguna mendaftarkan akun baru. Data yang harus diisi adalah nama, alamat *email*, dan *password*. Selanjutnya pengguna dapat klik tombol *Sign Me Up Now*.



Gambar 8. Tampilan daftar akun baru

Gambar 9 berikut merupakan tampilan setelah pengguna melakukan berhasil daftar atau login ke website.



Gambar 9. Tampilan menu home

Setelah melakukan sign in, pengguna akan melihat fitur Home, tampilan dari aplikasi adalah seperti pada Gambar 9. Terdapat fitur Home, My Courses, My Chart, Transactions, dan Sign Out setelah pengguna melakukan login ke aplikasi. Pengguna dapat memilih kelas pembelajaran yang hendak diambil dengan cara klik gambar dari kelas yang dimaksud.

Floparty

A complete guide to build rent-sell properties app.
Created by Erico Darmawan Handoyo

Rp 499.000

[Add to cart](#)

[Buy now](#)

Preview this course:

User Entity

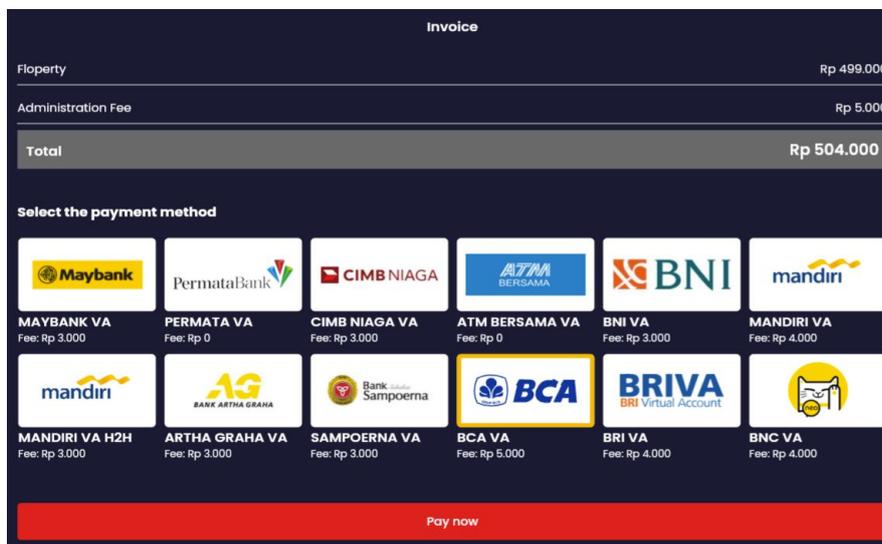
Course's objectives:

- Memahami penggunaan management BloC
- Memahami penggunaan Firebase
- Menerapkan Clean Architecture

Di dalam kelas online Floparty, peserta akan diajarkan bagaimana membuat sebuah aplikasi untuk melakukan transaksi jual/beli dan sewa/menyewa untuk property seperti rumah, apartemen, dan lain sebagainya dengan menggunakan Flutter SDK. Di sini peserta akan belajar bagaimana menerapkan state management, adaptive layout, dan juga clean architecture.

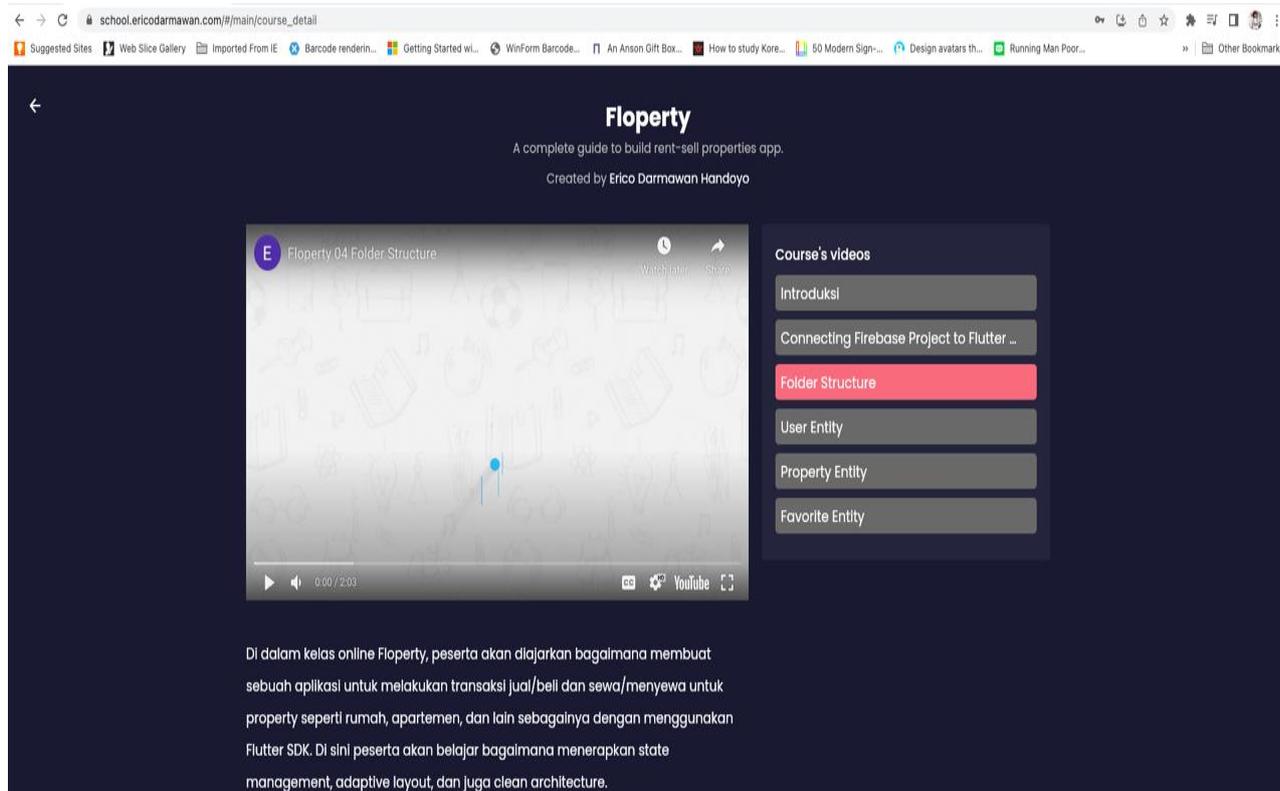
Gambar 10. Tampilan pilih kursus

Gambar 10 merupakan contoh apabila pengguna memilih kelas pembelajaran *Floerty*. Pada tampilan ini akan dijelaskan detail dari materi yang akan diajarkan, selain itu juga diberikan *video* singkat penjelasan yang bisa ditonton oleh pengguna. Jika pengguna hendak mengikuti kelas ini maka bisa dilanjutkan dengan klik tombol “Add to cart” yang berwarna merah. Pengguna juga bisa langsung melakukan pembayaran dengan cara klik “Buy now”.



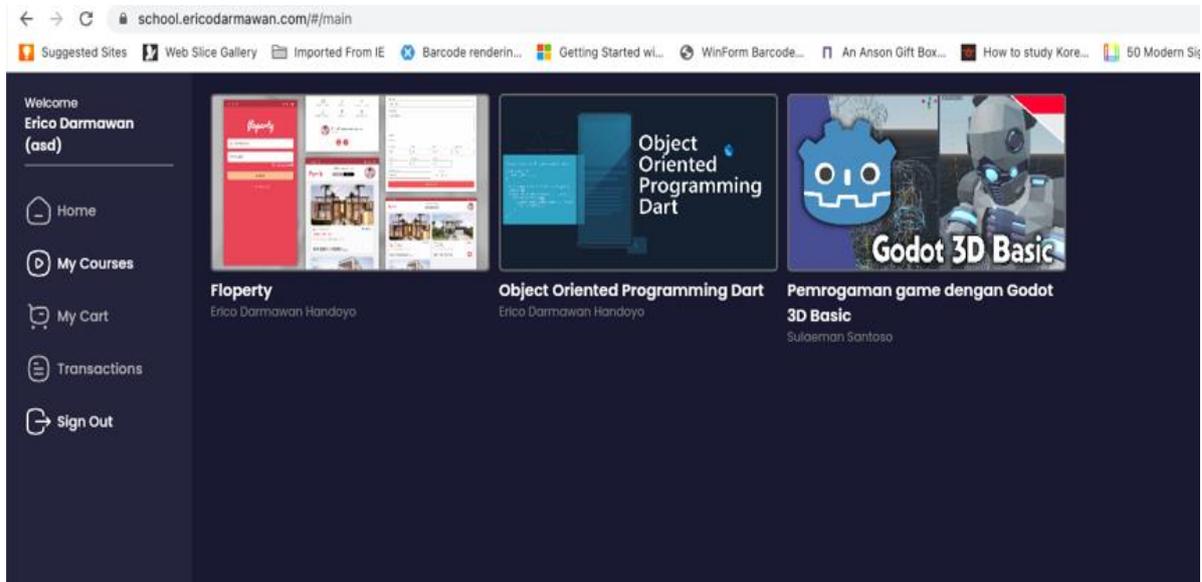
Gambar 11. Tampilan pembayaran kursus

Gambar 11 pengguna akan memilih channel pembayaran melalui beberapa bank yang sudah ditentukan. Untuk contoh pada gambar dipilih *virtual account* BCA. Selanjutnya pengguna bisa klik “Pay now”.



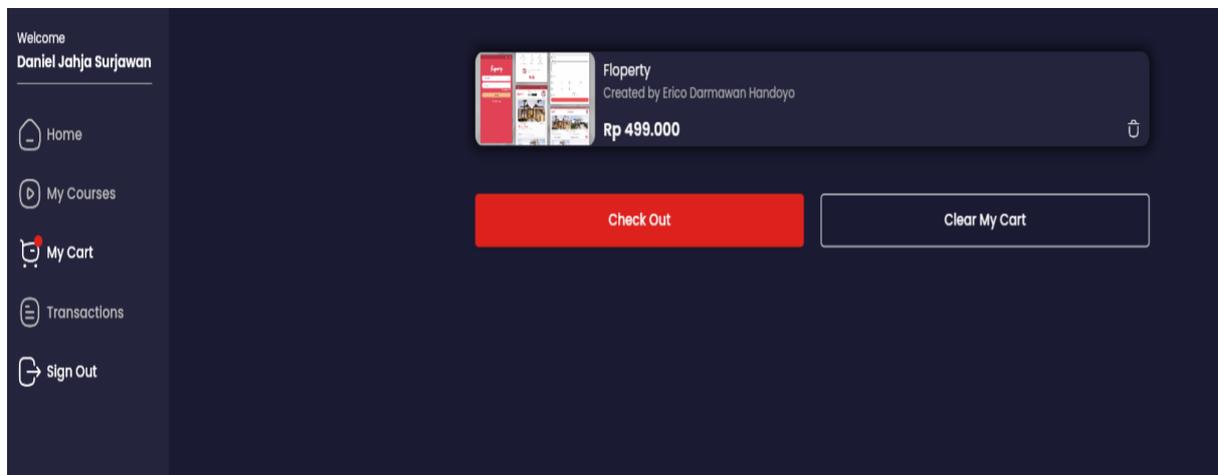
Gambar 12. Tampilan kursus yang dipilih

Gambar 12 pengguna sudah melakukan pembayaran dan masuk ke kursus yang dipilih. Materi dari setiap kursus ditampilkan pada menu sebelah kanan, setiap materi yang di klik akan ditampilkan penjelasan pada *video* yang berada di sebelah kiri. Contoh untuk tampilan gambar 12 adalah kelas pembelajaran *Floperity* dengan materi yang diberikan yaitu *Introduksi*, *Connecting Firebase Project to Flutter*, *Folder Structure*, *User Entity*, *Property Entity*, dan *Favorite Entity*.



Gambar 13. Tampilan my courses

Gambar 13 merupakan tampilan dari setiap kursus yang sudah diambil oleh pengguna dimana sistem menyimpan histori untuk setiap kursus tersebut.



Gambar 14. Tampilan my chart

Gambar 14 menampilkan pengguna yang sudah memilih kursus yang sudah dipilih akan muncul di halaman *My Chart*. Pengguna selanjutnya dapat melanjutkan pembayaran jika akan mengikuti kursus yang dipilih tersebut dengan klik "*Check out*", jika ingin membatalkan pengguna dapat klik tombol "*Clear My Chart*".

Sebagai langkah evaluasi dilakukan dengan melakukan *black box testing* terhadap aplikasi yang dikembangkan guna memastikan setiap fitur berfungsi sebagaimana tujuannya. Pada implementasinya *black box testing* yang digunakan menuju

terhadap bagian bagian terpenting dari aplikasi yaitu pendaftaran, proses *login*, proses pendaftaran kursus, penggunaan kursus, dan lain lain.

Proses pendaftaran dan *login* diuji menggunakan semua kemungkinan skenario *login* dan registrasi pengguna dengan hasil pada tabel 2 berikut.

TABEL 2
HASIL PENGUJIAN PROSES PENDAFTARAN

Skenario	Hasil yang diharapkan	Hasil yang diujikan	Kesimpulan
User memberikan data registrasi yang sesuai	User berhasil mendaftarkan diri dan diarahkan ke halaman <i>dashboard</i>	User berhasil mendaftarkan diri dan diarahkan ke halaman <i>dashboard</i>	Valid
User memberikan data registrasi yang tidak sesuai (kesalahan <i>email</i>)	Sistem mendeteksi kesalahan <i>format email</i> dan memberikan pesan kesalahan	Sistem mendeteksi kesalahan <i>format email</i> dan memberikan pesan kesalahan	Valid
User memberikan data registrasi yang tidak sesuai, <i>password</i> tidak dalam <i>format</i> yang benar	Sistem mendeteksi kesalahan <i>format password</i> lalu memberikan pesan kesalahan	Sistem mendeteksi kesalahan <i>format password</i> lalu memberikan pesan kesalahan	Valid
User memberikan data <i>login</i> yang sesuai	User berhasil mendaftarkan diri dan diarahkan ke halaman <i>dashboard</i>	User berhasil mendaftarkan diri dan diarahkan ke halaman <i>dashboard</i>	Valid
User memberikan data <i>login</i> yang tidak sesuai (kesalahan <i>email</i>)	Sistem mendeteksi kesalahan <i>format email</i> dan memberikan pesan kesalahan	Sistem mendeteksi kesalahan <i>format email</i> dan memberikan pesan kesalahan	Valid
User memberikan data <i>login</i> yang tidak sesuai, <i>password</i> tidak dalam <i>format</i> yang benar	Sistem mendeteksi kesalahan <i>format password</i> lalu memberikan pesan kesalahan	Sistem mendeteksi kesalahan <i>format password</i> lalu memberikan pesan kesalahan	Valid

Proses pendaftaran/pembelian kursus diuji menggunakan semua skenario yang mungkin terjadi pada setiap tahap yang mungkin terjadi. Hasil dapat terlihat pada tabel 3 berikut.

TABEL 3
PROSES PENDAFTARAN KURSUS

Skenario	Hasil yang diharapkan	Hasil yang diujikan	Kesimpulan
User memilih kursus untuk dibeli	Sistem mengarahkan <i>user</i> ke halaman detail kursus	Sistem mengarahkan <i>user</i> ke halaman detail kursus	Valid
User memilih <i>add to cart</i>	Sistem menyimpan pilihan kursus pada keranjang belanja	Sistem menyimpan pilihan kursus pada keranjang belanja	Valid
User memilih <i>buy now</i>	Sistem mengarahkan pengguna ke halaman pembayaran	Sistem mengarahkan pengguna ke halaman pembayaran	Valid
User menambah atau mengurangi kursus pada halaman <i>cart</i>	Sistem menambahkan / mengurangi kursus pada halaman <i>cart</i>	Sistem menambahkan / mengurangi kursus pada halaman <i>cart</i>	Valid
User melakukan pembayaran	Sistem mengarahkan pembayaran pada <i>payment gateway</i>	Sistem mengarahkan pembayaran pada <i>payment gateway</i>	Valid
User memilih <i>checkout</i> pada halaman <i>cart</i>	Sistem mengarahkan pada halaman pembayaran	Sistem mengarahkan pada halaman pembayaran	Valid

Pada dashboard pengguna, ada beberapa bagian utama yang dapat dilakukan dan evaluasi setiap bagian utama ini dapat dilihat pada tabel 4 berikut.

TABEL 4
EVALUASI DASHBOARD PENGGUNA

Skenario	Hasil yang diharapkan	Hasil yang diujikan	Kesimpulan
User memilih kursus yang sudah dimilikinya	Sistem mengarahkan ke halaman detail kursus	Sistem mengarahkan ke halaman detail kursus	Valid

Skenario	Hasil yang diharapkan	Hasil yang diujikan	Kesimpulan
User memilih bagian dalam kursus untuk ditonton	Sistem mengarahkan <i>video</i> ke bagian yang hendak di tonton	Sistem mengarahkan <i>video</i> ke bagian yang hendak ditonton	Valid

IV. SIMPULAN

Dari hasil penelitian tersebut dapat diambil kesimpulan bahwa aplikasi kursus digital memberikan manfaat yang dapat meningkatkan pengalaman belajar dan memudahkan akses ke berbagai materi pembelajaran. Aplikasi yang dibuat dapat memberikan akses yang fleksibel, pilihan materi yang luas, biaya yang lebih terjangkau, pembelajaran seumur hidup karena dikembangkan dengan menggunakan flutter *framework* yang memungkinkan pengembangan aplikasi secara *multiplatform* yang memiliki keunggulan dimana semua hal dapat dilakukan oleh guru maupun pengguna yang menggunakan. Keuntungan yang lain pengembangan aplikasi dengan *clean architecture* yaitu tidak terikat kepada antarmuka pengguna, *database*, dan agen eksternal lainnya. Dengan manfaat-manfaat ini, aplikasi kursus digital dapat menjadi alat yang sangat berguna dalam mendukung pembelajaran dan pengembangan pribadi.

Keunggulan dari aplikasi yang dibuat bersifat *multiplatform* masih bisa dilakukan pengembangan lebih lanjut. Beberapa pengembangan penelitian dari aplikasi yang dapat dilakukan yaitu peningkatan antarmuka siswa pada bagian *feedback visual*, dimana siswa dapat memberikan umpan balik *visual* yang jelas saat siswa menyelesaikan tugas atau mencapai pencapaian tertentu, personalisasi pembelajaran seperti rekomendasi kursus yaitu dapat menggunakan algoritma untuk merekomendasikan kursus berdasarkan minat dan aktivitas sebelumnya, atau fitur penyesuaian konten yang menyesuaikan tingkat kesulitan dan jenis konten berdasarkan kemampuan siswa, dan kemampuan siswa, gamifikasi dengan menerapkan elemen seperti lencana, poin, dan papan peringkat untuk memotivasi siswa, dan analitik dan pelacakan kemajuan dengan membuat *dashboard* kemajuan yang menunjukkan kemajuan belajar siswa, termasuk statistik waktu belajar, pencapaian, dan area yang perlu ditingkatkan, juga bisa dibuat pengiriman notifikasi dan pengingat untuk membantu siswa tetap pada jalur pembelajaran mereka.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada LPPM UK Maranatha yang sudah memberikan dukungan dalam penelitian ini.

DAFTAR PUSTAKA

- [1] N. N. M. Kasim and F. Khalid, "Choosing the Right Learning Management System (LMS) for the Higher Education Institution Context: A Systematic Review," *Int. J. Emerg. Technol. Learn.*, pp. Vol. 11, no. 06, pp. 55–61, Jun. 2016.
- [2] Google, "Fitur Google Classroom," 23 February 2024. [Online]. Available: https://edu.google.com/intl/ALL_id/workspace-for-education/classroom/editions/.
- [3] MoodleDocs, "Features - MoodleDocs," 15 January 2024. [Online]. Available: <https://docs.moodle.org/401/en/Features>.
- [4] Udemy, "Online Courses - Learn Anything, On Your Schedule," 25 March 2024. [Online]. Available: <https://www.udemy.com>.
- [5] Kognity, "Kognity - Education that empowers.," 7 April 2024. [Online]. Available: https://kognity.com/?creative=681060165554&keyword=kognity&matchtype=b&network=g&device=c&gad_source=1.
- [6] Flutter, "Flutter Dev.," 13 April 2024. [Online]. Available: <https://flutter.dev>.
- [7] D. F. 1.1.0, "Pub Dev Packages," 17 May 2024. [Online]. Available: https://pub.dev/packages/dart_frog.
- [8] K. K. Y. Christian, "Rancang Bangun Aplikasi Kursus Online Berbasis Web Dengan Sistem Rekomendasi Metode Content-Based Filtering," 2022. [Online]. Available: <https://ojsbimtek.univrab.ac.id/index.php/rabit/article/view/2181>.
- [9] D. Purwanto, W. Pramusinto and G. P. Utama, "Aplikasi Kursus Online Berbasis Web Service Menggunakan Arsitektur Microservices," in *SENDIU*, Jakarta, 2021.
- [10] Jamaluddin, A. Muliati and A. Hasanuddin, "Program Lao-Kursus (Layanan Aplikasi Online-Kursus) Berbasis Android (Studi Pengembangan Pada Lembaga Kursus Dan Pelatihan Di Kabupaten/Kota Sulawesi Selatan)," in *Prosiding Seminar Nasional Pengabdian kepada Masyarakat 2021 (SNPPM-2021)*, Jakarta, 2021.
- [11] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," in *International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2019.
- [12] R. C. Martin, "Clean Craftsmanship: Disciplines, Standards, and Ethics," 24 July 2024. [Online]. Available: https://books.google.co.id/books?hl=en&lr=&id=qtnPEAAQBAJ&oi=fnd&pg=PA1957&dq=Clean+Craftsmanship:+Disciplines,+Standard+s,+and+Ethics&ots=TWQA42EXCc&sig=yo_4r96cP1QUrfWjUcJgOqV_Bx4&redir_esc=y#v=onepage&q=Clean+Craftsmanship%3ADisciplines%2C+Standards%2.