COURSE BOOK BERBASIS ANDROID

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan

Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Disusun Oleh:

Fitra Dwianto NPM 1062010

POLITEKNIK MANUFAKTUR NEGERI

BANGKA BELITUNG

TAHUN 2024

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

COURSE BOOK BERBASIS ANDROID

Oleh:

Fitra Dwianto/1062010

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1	Pembimbing 2
Linda Fujiyanti,S.T., M.T.I	Sari Mubaroh,S.Pd., M.Pd.
NIP.198109262014042001	NIP. 198501122019032015

Penguji 1	Penguji 2
Riki Afriansyah, M.T	Indah Riezky Pratiwi, M.Pd
NIP. 199004042019031013	NIP.199010082019032018

Dipindai dengan CamScanner

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini:

Nama : Fitra Dwianto NPM : 1062010
Mahasiswa

Dengan Judul : Course Book Berbasis Android

Menyatakan laporan akhir ini merupakan hasil kerja saya sendiri dan bukan plagiat. Oleh karna itu pernyataan ini saya buat dengan sebenarnya dan apabila dikemudian hari saya melanggar pernyataan ini, saya bersedia menerima sanksi yang berlaku.

Sungailiat ,10 Januari 2025

Nama Mahasiswa	Tanda Tangan
1. Fitra Dwianto	.....

Dipindai dengan CamScanner

ABSTRAK

UPTD SDN 13 memiliki masalah dengan ketersediaan buku baru dengan konten terkini. sebagian besar koleksi buku perpustakaan berisi konten yang sudah ketinggalan zaman. Mengingat informasi yang tidak lagi relevan cenderung berubah seiring waktu dan tahun pembelajaran, dan informasi ini umumnya tersedia secara gratis di internet, masalah ini harus ditangani segera. Akibatnya, ada kemungkinan untuk mengembangkan sebuah aplikasi yang dimaksudkan untuk mengumpulkan dan memberikan akses ke data tersebut. Penelitian ini bertujuan untuk meningkatkan minat siswa SDN 13 dalam membaca materi pembelajaran dan memudahkan akses ke materi terbaru. Metode pengembangan yang digunakan dalam penelitian ini adalah metode agile, dan pengujian aplikasi dilakukan dengan pendekatan User Acceptance Test (UAT) dan metode white box testing. Hasil dari pengujian User Acceptance Test (UAT) menunjukkan bahwa nilai yang diperoleh guru adalah 85,66%, sedangkan siswa kelas 6 mencapai 94,13%. Dengan demikian, hasil UAT menunjukkan bahwa aplikasi ini memenuhi kriteria "Sangat Layak" untuk diterapkan di UPTD SDN 13 Sungailiat.

Kata Kunci : Android, Course Book, Digital, Ebook, Perpustakaan

ABSTRACT

UPTD SDN 13 has a problem of not having new books with the latest content available. Most of the library's collection contains outdated content. This problem needs to be addressed immediately, as information that is no longer relevant changes over time and years of research, and such information is generally available for free on the Internet. Therefore, it is possible to develop an application that collects such data and provides access to it. This research aims to increase the interest of SDN 13 students in reading study materials and to facilitate their access to the latest materials. The development methodology adopted in this study is agile methodology and application testing is carried out using UAT (User Acceptance Testing) approach and white box testing methodology. As a result of UAT (User Acceptance Testing) testing, the score obtained by teachers was 85.66% and the 6th grade students obtained 94.13%. Therefore, the UAT results show that this application meets the criteria of being "highly suitable" for implementation in UPTD SDN 13 Sungailiat.

Keyword : Android, Course Book, Digital, Ebook, library

KATA PENGANTAR

Assalamualaiakum warahmatullahi wabarakatuh dengan segala puji Allah SWT yang membimbing penulis dalam menulis laporan akhir ini yang berjudul "COURSE BOOK BERBASIS ANDROID ", yang melimpahkan kelancaran serta hidayahnya kepada penulis agar dapat menyelesaikan program studi Diploma IV Teknologi Rekayasa Perangkat Lunak di Politeknik Manufaktur Negeri Bangka Belitung, Di dalam laporan ini penulis tau bahwa dalam penyusunannya terdapat kekurangan mengingat penulis juga seorang manusia, namun berkat rahmat Allah SWT, serta bimbingan dari dosen dan teman-teman sehingga laporan proyek ini dapat diselesaikan. Harapan saya sebagai penulis agar laporan akhir ini dapar bermanfaat untuk kedepannya. Maka dari itu, sebagai penulis saya ucapkan terima kasih kepada :

1. Bapak I Made Andik Setiawan, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
2. Bapak Irwan, M.Sc, Ph.D selaku Wakil Direktur I Politeknik Manufaktur Negeri Bangka Belitung.
3. Ibu Yang Agita Rindri , S.Kom., M.Eng. selaku Ka. Jurusan Informatika dan Bisnis Politeknik Manufaktur Negeri Bangka Belitung.
4. Bapak Sidhiq Andriyanto , S.T., M.Kom. selaku Koordinator Program Studi Sarjana Terapan Teknologi Rekayasa Perangkat Lunak Politeknik Manufaktur Negeri Bangka Belitung.
5. Ibu Linda Fujiyanti , S.T., M.T.I. selaku Dosen Pembimbing Utama Proyek Akhir.

6. Ibu Sari Mubaroh , S.Pd., M.Pd. selaku Dosen Pembimbing kedua Proyek Akhir.
7. Teman-teman seperjuangan yang telah memberikan support dan dukungannya.
8. Ibu dan Ayah tercinta serta seluruh keluarga yang dengan penuh keikhlasan untuk dukungannya.

Setelah menjalani proses yang panjang dan penuh tantangan, tiba dimana penulis dapat menyelesaikan pembuatan aplikasi beserta laporan proyek akhir ini. Penulis menyadari ada kekurangan dari penulisan makalah proyek akhir ini, oleh karena itu saran, kritik serta petunjuk yang bertujuan membangun dari pembaca yang dapat membantu mengembangkan dan memperbaiki penulisan nantinya. Penulis berharap makalah ini dapat berguna untuk membantu rekan rekan mahasiswa dalam menambah wawasan atau sebagai wacana dalam penulisan, akhir kata Wassalamualaikum Warahmatullahi wabarakatuh.

Sungailiat, 10 Januari 2025

Penulis

Dipindai dengan CamScanner

DAFTAR ISI

LEMBAR PENGESAHAN ii
PERNYATAAN BUKAN PLAGIAT iii
ABSTRAK iv
ABSTRACT v
KATA PENGANTAR vi
DAFTAR ISI viii
DAFTAR GAMBAR xi
DAFTAR TABEL xii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 4
1.3 Tujuan 4
BAB II DASAR TEORI 5
2.1 *State of Art* 5
2.2 Elektronic Book (Ebook) 11
2.3 Aplikasi Mobile Berbasis Android 12
2.4 Metode Pengembangan Sistem 12
2.4.1 Sprint 13
2.5 Software Perancangan Course Book 13
2.5.1 Figma 13
2.5.2 Affinity Designer 2 14
2.5.3 Android Studio 14

2.5.4 Firebase 15
BAB III METODE PELAKSANAAN 16
3.1 Diagram Alir 16
3.2 Pengumpulan Data 16
3.3 Analisi Kebutuhan 17
3.3.1 Hardware 17
3.3.2 Software 17
3.4 Pembuatan 19
3.4.1 Metode Pengembangan Agile 19
3.4.2 Use Case Diagram 19
3.4.3 Perancangan Tampilan Aplikasi 21
3.4.4 Produksi 24
3.4.5 Alpha 24
3.4.6 Beta 24
3.4.7 Rilis 28
3.5 Laporan 28
BAB IV HASIL DAN PEMBAHASAN 29
4.1 Produksi 29
4.1.1 *SPRINT1*: Membuat Halaman *Home* Admin dan *User* 29
4.1.2 SPRINT2: Login Multi Level 32
4.1.3 *SPRINT3*: Menghubungkan *Firebase* ke android studio 36
4.1.4 *SPRINT4*: Membuat Halaman *Search* Admin dan *User* 41
4.1.5 *SPRINT5*: Membuat halaman detail buku dan halaman baca 44
4.1.6 SPRINT6: Add user 47
4.1.7 *SPRINT7*: *Add* book 51

4.1.8 *SPRINT8*: Archive 55
4.1.9 *SPRINT9*: Monitoring Statistics Admin 58
4.2 Alpha 60
4.2.1 Pengujian halaman home 60
4.2.2 Pengujian halaman search 62
4.2.3 Pengujian halaman archive 63
4.2.4 Pengujian halaman baca 64
4.2.5 Pengujian halaman statistik 64
4.2.6 Pengujian halaman add book 65
4.2.7 Pengujian halaman add user 65
4.3 Beta 66
4.3.1 Hasil Kuesioner guru SDN 13 Sungailiat 66
4.3.2 Hasil Kuesioner Siswa Kelas 6 67
4.4 Rilis 69
BAB V KESIMPULAN DAN SARAN 70
5.1 Kesimpulan 70
5.2 Saran 70
DAFTAR PUSTAKA 71
LAMPIRAN 75
LAMPIRAN 2: KODE PROGRAM 76
LAMPIRAN 3: Dokumentasi 82

DAFTAR GAMBAR

Gambar 2.4-1 Metode AGILE 13
Gambar 3.1-1 Diagram alir 16
Gambar 3.4-1 Sprint Agile 19
Gambar 3.4-2 Use Case Diagram 20
Gambar 3.4-3 Desain Aplikasi di software figma 21
Gambar 4.1-1Tampilan home admin dan user 30
Gambar 4.1-2 program halaman home 31
Gambar 4.1-3 Halaman Login 33
Gambar 4.1-4 multilevel login 34
Gambar 4.1-5 Firebase 36
Gambar 4.1-6 tampilan firebase database 37
Gambar 4.1-7 Project settingg firebase 37
Gambar 4.1-8 Penempatan googleservice.json 39
Gambar 4.1-9 Tampilan Halaman Search admin dan user 41
Gambar 4.1-10 program search 42
Gambar 4.1-11 tampilan detail buku dan baca 44
Gambar 4.1-12Program halaman detail buku 45
Gambar 4.1-13 program haklaman baca 46
Gambar 4.1-14 tampilan add user 48
Gambar 4.1-15 program add user 50
Gambar 4.1-16 tampilan add book 52
Gambar 4.1-17 program add book 53
Gambar 4.1-18 Tampilan archive 55
Gambar 4.1-19 program archive 56
Gambar 4.1-20 Tampilan list statistics dan graphic statistics 58
Gambar 4.1-21Tampilan list dan graphic statistics 59
Gambar 4.3-1 Tampilan akses awal playstore 69

DAFTAR TABEL

Table 2.1-1Tabel state of art 5
Table 3.5-1 bobot kuesioner 25
Table 3.5-2 Kuesioner siswa 25
Table 3.5-3 Kuesioner Guru 26
Table 3.5-4 Perhitungan UAT 27
Table 4.2-1 Halaman Home Siswa dan Guru 60
Table 4.2-2Halaman home admin 61
Table 4.2-3 Halaman Search 62
Table 4.2-4 halaman archive 63
Table 4.2-5 halaman baca 64
Table 4.2-6 halaman statistik 64
Table 4.2-7 halaman add book 65
Table 4.2-8 halaman add user 66
Table 4.3-1Kuesioner guru 66
Table 4.3-2 Kuesioner siswa kelas enam 67

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi berkembang pesat di seluruh dunia, membawa keuntungan dan manfaat bagi masyarakat. Saat ini, teknologi digunakan di banyak bidang kehidupan sehari-hari untuk memfasilitasi tugas-tugas yang rumit dan berulang. Salah satu dari dampak teknologi adalah perpustakaan. Perpustakaan berkembang seiring waktu dan seiring kemajuan teknologi, perpustakaan menyediakan kemudahan dan akses mudah ke informasi serta membangun kolaborasi dengan perpustakaan lain untuk berbagi informasi dan mencapai tujuan yang diinginkan[23].

Dukungan teknologi terutama internet sangat memberi manfaat bagi penggunaanya, salah satunya pada suatu sistem berbasis aplikasi yang sangat membantu pengguna dalam memperoleh informasi khusus pada aplikasi perpustakaan dimana hal tersebut dapat memudahkan akses dalam peminjaman buku, akses informasi yang akan meningkatkan kualitas perpustakaan dalam penyimpanan informasi lebih efisien[23]. Manfaat dari sistem perpustakaan adalah sistem dibuat untuk memberikan informasi buku yang tersedia yang bisa diakses oleh siswa secara gratis dan memudahkan monitoring keaktifan pengguna, penambahan buku baru, penambahan pengguna sehingga dapat membantu pengelolahan perpustakaan.

Kepala sekolah dan pustakawan UPTD SDN 13 ingin siswa untuk menjadi lebih aktif dalam membaca buku, namun fasilitas yang disediakan kurang mendukung, dikarenakan sebagian besar koleksi buku di perpustakaan mengandung konten informasi yang sudah ketinggalan zaman yang dapat menurunkan minat siswa dalam membaca. Berdasarkan wawancara penulis dengan pustakawan, total pengujung perpustakaan SDN13 menurun. Hal ini dipengaruhi oleh ketersediaan buku yang kurang menarik, dan konten yang tersedia masih

belum diperbarui, contohnya buku cerita. buku cerita bergambar yang dikemas lebih menarik, dapat menarik perhatian anak, dan dapat mendorong mereka untuk memahami pelajaran di kelas dengan lebih baik[1].

Berdasarkan penelitian sebelumnya mengenai sistem perpustakaan website, SMK Bani Muctar Rancaekek merupakan sebuah sekolah menengah kejuruan swasta yang berada di wilayah Kecamatan Rancaekek. Lebih tepatnya beralamat di Jalan Bojongloa Kp.Pesantren No.27B Desa Bojongloa Kec.Rancaekek Kab.Bandung. SMK Bani Muchtar Rancaekek mulai berdiri pada tahun 2016 dengan keluarnya surat keputusan dengan nomor 421.3/3518- disdikbud. Memiliki 2 jurusan yakni : Tataboga dan Bisnis Daring dan Pemasaran. Saat ini proses dalam transaksi sistem perpustakaan di SMK Bani Muchtar masih berbasis semi komputerisasi yaitu menggunakan pencatatan dibuku dan di salah satu aplikasi dari *Microsoft Office* yaitu *Microsoft Excel*. Data koleksi buku terhitung ada sebanyak 773 buku terhitung sejak tahun 2016 hingga saat ini, lalu ada data pengunjung terhitung minimal 20 orang pengunjung perhari, dengan 1 orang staff pustakawan dan 1 orang kepala pustakawan, jam pelayanan perpustakaan terhitung mulai pukul 08:00 hingga 14:00 dengan jam istirahat perpustakaan pada pukul 12:00-13:00[23].

Terdapat beberapa kendala dari sistem perpustakaan ini, yaitu terbatasnya layanan perpustakaan yang hanya melayani saat jam kerja saja, kesulitan dalam mencari buku yang tersedia, lambatnya proses pembuatan laporan yang tidak bisa dihasilkan secara otomatis, kesulitan dalam mendeteksi buku yang sedang dipinjam, dan membutuhkan proses yang cukup lama dalam penginputan data buku. Adapun tujuan dari penelitian yang dilakukan ini yaitu membuat rancangan sistem aplikasi yang dapat membuat pelayanan perpustakaan menjadi 24jam, merancang sistem agar dapat mempermudah dalam mencari dan mengakses buku yang tersedia, merancang sistem informasi yang mampu memonitoring kegiatan perpustakaan yang akan memudahkan pustakawan dalam meningkatkan proses pembuatan laporan, merancang sistem informasi yang mampu menyimpan buku agar bisa diakses kapan saja, dan merancang sistem informasi untuk mempermudah proses penginputan data buku maupun *user* aplikasi.

Berdasarkan wawancara yang dilakukan oleh penulis dengan kepala sekolah SD 13 Sungailiat, kepala sekolah menceritakan tentang beberapa masalah yang ada berkaitan tentang perpustakaan sekolah, hal ini juga dikonfirmasi oleh pihak pustakawan. Masalah yang sedang dihadapi adalah sebagai berikut: Pertama, terdapat permasalahan terkait persediaan buku baru yang mengandung informasi terkini. Hal ini disampaikan oleh pustakawan, yang menginformasikan bahwa sebagian besar koleksi buku di perpustakaan mengandung konten informasi yang sudah ketinggalan zaman. Problematika ini perlu segera ditangani, mengingat informasi yang tidak lagi mutakhir cenderung mengalami perkembangan seiring berjalannya waktu dan tahun pembelajaran, dan informasi tersebut umumnya dapat diakses secara gratis melalui internet. Oleh karena itu, terdapat potensi untuk mengembangkan sebuah aplikasi yang bertujuan untuk mengumpulkan dan memberikan akses kepada informasi tersebut.

Selain itu, terdapat faktor lain yang menjadi hambatan, seperti faktor ekonomi. Biaya untuk pengadaan buku fisik sangat tinggi, sehingga jika didasarkan pada anggaran sekolah, pengeluaran untuk membeli buku baru menjadi tidak memungkinkan. Faktor lingkungan juga berkontribusi terhadap permasalahan ini; lokalisasi sekolah yang berada di daerah kecil menyulitkan akses terhadap informasi terkini, mengharuskan pihak perpustakaan untuk melakukan perjalanan ke kota, yang tentu akan menambah biaya. Isu seperti, kondisi fisik buku yang sudah using juga berpotensi menurunkan minat siswa untuk membaca buku-buku tersebut.[2]. oleh karena itu perpustakaan dimana seharusnya menjadi tempat bagi siswa untuk menganut ilmu dan menambah wawasan terhalangi[2], hanya dikarenakan kurang mendukungnya fasilitas yang mereka miliki.

Berdasarkan masalah yang dipaparkan maka penting dibuatkan sebuah aplikasi Perpustakaan berbasis digital dengan sistem berbasis android berjudul *Course Book*. Dengan adanya aplikasi tersebut siswa dan guru dapat mengakses buku tersebut melalui perangkat *android* mereka, dimanapun, kapanpun hanya dengan melalui aplikasi tersebut. Oleh karena itu aplikasi ini mampu menyelesaikan masalah yang ada dengan mengatasi faktor-faktor yang menjadi poin utama di penelitian ini.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan, dapat ditarik kesimpulan bahwa pokok permasalahan penelitian ini yaitu :

1. Bagaimana merancang dan membuat aplikasi *Course Book* Bebasis *android*.
2. Bagaimana pengoptimalan fitur yang disediakan di aplikasi *Course Book*.
3. Bagaimana mengetahui tingkat kelayakan dari aplikasi melalui metode pengujian *whitebox* dan *User Acceptance Testing*.

1.3 Tujuan

Berdasarkan rumusan masalah adapun tujuan dari Pembuatan Aplikasi Proyek akhir ini sebagai berikut:

1. Merancang dan membuat aplikasi *Course book* dengan berbasis system android menggunakan bahasa Kotlin dengan perangkat UI *jetpack compose*, beserta *android studio* dengan menggunakan *Firebase NoSQL*.
2. Pengoptimalan fitur yang tersedia di aplikasi *Course Book* dengan memanfaatkan *tools* yang ada.
3. Tingkat kelayakan aplikasi ini diuji menggunakan metode whitebox dan kuesioner UAT yang diberikan ke penguji, siswa dan guru untuk memastikan bahwa aplikasi yang dikembangkan memenuhi kebutuhan pengguna dan siap untuk dirilis.

BAB II
DASAR TEORI

2.1 *State of Art*

Penulisan *state of art* ini merujuk pada penelitian sebelumnya yang selaras dan relevan dengan topik dari penelitian yang dilakukan yang nantinya akan menjadi acuan dan perbandingan dalam melakukan penelitian ini.. Di bawah ini adalah hasil dari penelitian yang telah dilakukan sebelumnya :

Table 2.1-1Tabel state of art

Nama & Tahun Penelitian	Sumber	Judul	Persamaan	Perbedaan
Eni Dwi Wardihani, Bambang Supriyo, Ilham Sayekti (2021)	Ojs.pnb.ac.id	"Penerapan Sistem Informasi dan Manajemen Perpustakaan Berbasis IoT di SDN Kramas,Tembalang,Kota Semarang"	Meneliti sistem informasi untuk manajemen data buku dan *monitoring* perpustakaan	Perbedaan yang ada disini yaitu, penulis menerapkan sistem *monitoring* secara *online*. Fitur ini dapat mempermudah pengawasan *progress* perharian,bulanan,tahu

				nan melalui aplikasi.
Br Ginting, Y. F., & Simamora, H. (2022).	Repository.uinsu.ac.id	"Upaya pemanfaatan buku elektronik berbasis Aplikasi studi kasus dinas perpustakaan dan kearsipan daerah Kabupaten Serdang Bedagai"	Menggunakan *E-book* sebagai salah satu opsi dalam data pembacaaan digital.	Penggunaan Media hanya untuk pendidikan daring dan luring, yang membatasi akses informasi, sedangkan course book bisa diakses kapan saja hanya melalui *smartphone android*
Wilantika Ramadhani (2020)	https://doi.org/10.47709/educendikia.v1i01.1774	"Penggunaan buku digital dalam proses pengajaran di sekolah dasar sangat berpengaruh"	Menggunakan buku digital dalam proses pembelajaran dan pencarian informasi	Materi buku digital disampaikan oleh siswa dengan diceritakan, sedangkan

				course book dapat diakses secara individu.
Wijaya, S. H., & Sayekti, R. (2024).	.https://doi.org/10.32884/ideas.v10i3.1862	"Analisis tingkat ketergantungan aplikasi perpustakaan Universitas Quality dalam mendukung tujuan pembangunan berkelanjutan (SDGs)"	penelitian ini membahas mengenai digitalisasi perpustakaan sangat penting untuk memberikan layanan berkualitas, dalam menanggapi kemajuan teknologi, dan mencapai SDGs	Metode penelitian yang digunakan berbeda, dimana penelitian ini menghitung *System Usability Scale* (SUS) yang dikumpulkan melalui google form sedangkan penulis menggunakan *user acceptance testing (UAT)* untuk mengukur Tingkat

				kelayakan implementasi berdasarkan hasil kuesioner.
Kurniawan, A. (2020).	http://repository.uir.ac.id/id/eprint/12473%0Ahttps://repository.uir.ac.id/12473/1/163510108.pdf	"Pengembangan aplikasi berbasis android untuk menelusuri koleksi dari perpustakaan Universitas Islam Riau (UIR)"	menciptakan perpustakaan digital yang komprehensif serta dapat melayani permintaan informasi	File yang disediakan masih berbentuk file *download* yang belum bisa ditampilkan di aplikasi sedangkan *course book* bisa menampilkan isi file tersebut langsung dari aplikasi.

Berdasakan hasil kajian pustaka yang telah penulis kaji, yakni mencakup berbagai penelitian terkait sistem aplikasi perpustakaan dengan penggunaan ebook sebagai media informasi. Beberapa fitur unggulan dari penelitian-penelitian tersebut adalah pembangunan aplikasi perpustakaan untuk memberikan layanan berkualitas, dalam

menyediakan informasi digital. Namun, ada beberapa kekurangan yang dapat diidentifikasi dari penelitian-penelitian tersebut

a) dimulai dari penelitian yang berjudul "Penerapan Sistem Informasi dan Manajemen Perpustakaan Berbasis IoT di SDN Kramas,Tembalang,Kota Semarang" hasil akhir dari penelitian ini menunjukan masyarakat melalui program program kemitraan masyarakat ini adalah pengembangan situs web dengan subdomain yang digunakan untuk perpustakaan SDN Kramas, Tembalang, yang sekarang beroperasi sebagai perpustakaan sdnkramas.sch.id[3]. Kendala yang terkait dengan pelaksanaan masyarakat ini adalah pandemi COVID-19 yang membatasi pertemuan dan tidak ada siswa yang pergi ke sekolah secara teratur[3]. Akibatnya, pembuatan kartu perpustakaan ditunda[3]. Kekurangan dari penelitian tersebut terletak di monitoring dimana masih belum diwujudkan. Perbedaannya dengan penulis yaitu dalam menyediakan fitur *monitoring* yang dapat dilakukan secara online, adanya fitur ini dapat mempermudah pengawasan progress perharian,bulanan,tahunan melalui aplikasi *course book*, yang dimana menjadi kelebihan tersendiri.
b) Penelitian "Upaya pemanfaatan buku elektronik berbasis aplikasi studi kasus dinas perpustakaan dan kearsipan daerah Kabupaten Serdang Bedagai" di penelitian ini penggunaan media yang efektif sangat penting untuk pendidikan daring dan luring. Guru harus memahami kriteria pemilihan yang sesuai dengan pembelajaran yang akan diberikan kepada siswa[8]. Kriteria ini termasuk relevansi dengan tujuan pembelajaran, dukungan isi materi, kemudahan memperoleh media, ketersediaan berbagai jenis media, kemampuan guru untuk menggunakannya, dan ketersediaan waktu untuk menggunakannya. Ini berlaku juga untuk penggunaan media buku digital di kelas[8]. E-book menjadi salah satu opsi dalam penelitian ini. Oleh karena itu aplikasi *Course book* memiliki kelebihan untuk dapat mengakses buku digital/*Ebook* tersebut. Hadirnya aplikasi ini dapat membantu siswa serta guru dalam mendapatkan materi digital yang efektif dan mudah diakses.

c) Selanjutnya penelitian “Penggunaan buku digital dalam proses pengajaran di sekolah dasar sangat berpengaruh“ berdarkan penelitian sebelumnya penggunaan media yang efektif sangat penting untuk pendidikan daring dan luring. Guru harus memahami kriteria pemilihan yang sesuai dengan pembelajaran yang akan diberikan kepada siswa[7]. Kriteria ini termasuk relevansi dengan tujuan pembelajaran, dukungan isi materi, kemudahan memperoleh media, ketersediaan berbagai jenis media, kemampuan guru untuk menggunakannya, dan ketersediaan waktu untuk menggunakannya. Ini berlaku juga untuk penggunaan media buku digital di kelas[7]. Kesimpulan yang didapat menjadi salah satu pertimbangan dalam pengembangan aplikasi penulis. Oleh karena itu *course book* menjadi salah satu cara agar siswa dapat mengakses buku digital yang dapat membantu siswa serta guru dalam mendapatkan *e-modul* yang efektif dan mudah diakses.

d) Pengukuran “Analisis tingkat ketergantungan aplikasi perpustakaan Universitas Quality dalam mendukung tujuan pembangunan berkelanjutan (SDGs)“ di penelitian ini membahas mengenai digitalisasi perpustakaan sangat penting untuk memberikan layanan berkualitas, menanggapi kemajuan teknologi, dan mencapai SDGs[4]. Dalam penelitian ini mengungkap bagaimana *Quality University Library* memfasilitasi akses ke informasi. *System Usability Scale* (SUS) digunakan untuk penyelidikan ini[4]. Data dikumpulkan melalui survei *Google Forms*[4]. Dalam Perhitungan SUS menunjukkan bahwa aplikasi perpustakaan *Quality University* memiliki skor kegunaan rata-rata 63, yang dianggap cukup[4]. berdasaarkan studi di atas menunjukan pentingnya perpustakaan digital di era *modern* saat ini, maka dari itu perubahan digital perpustakaan SDN 13 akan menjadi langkah yang tepat. aplikasi seperti *Course book* dibutuhkan, untuk memberikan layanan berkualitas kepada siswa siswi beserta guru sekolah.

e) Terakhir yaitu “Pengembangan aplikasi berbasis android untuk menelusuri koleksi dari perpustakaan Universitas Islam Riau (UIR)” dasar penelitian

yang dilakukan perpustakaan UIR bertujuan untuk menciptakan perpustakaan digital yang komprehensif serta dapat melayani permintaan informasi dan menghasilkan lulusan berkualitas dengan daya saing lokal, nasional, dan internasional[6]. Untuk mencapai tujuan ini, perpustakaan UIR telah menerapkan teknologi digital seperti repositori perpustakaan dan katalog akses publik online[6]. Kedua sistem tersebut berbasis web dan dapat diakses melalui internet. untuk menjelajahi atau mencari koleksi perpustakaan, pengguna harus memanfaatkan kedua platform independen ini[6]. Maka dari itu kelemahan website tersebut hanya dapat menyediakan *file source* yang di upload saja. Oleh karena itu peran penting aplikasi *Course Book* dalam hal ini yaitu dapat menyediakan, mengakses serta membaca buku di koleksi hanya dengan menggunakan perangkat android serta memiliki tampilan yang menarik untuk dibaca oleh *user*.

Dari studi literatur di atas dapat disimpulkan bahwa banyak poin penting yang bisa ditimbangkan untuk mengembangkan sebuah aplikasi terutama fleksibilitas dalam penggunaan aplikasi, Keterampilan desain serta promosi untuk menarik banyak orang untuk mecoba dan menggunakan aplikasi dalam keseharian mereka. Selain berfokus pada poin di atas dapat ditimbangkan juga bahwa kemudahan dalam mengakses data tersebut juga menjadi pertimbangan, oleh karena itu penelitian ini akan lebih berfokus dalam fleksibilitas penggunaan aplikasi baik dari sisi *user* maupun admin dalam membaca, menambahkan, menghapus ,memodifikasi sebuah data atau materi yang dapat dilihat dan disediakan.

2.2 Elektronic Book (Ebook)

Ebook merupakan versi elektronik dari buku fisik cetak yang biasa kita gunakan, dengan memanfaatkan Perangkat digital seperti komputer, *Smartphone* atau menggunakan alat baca ebook [7]. *Ebook* memberikan akses bagi siswa untuk mempelajari materi baru dengan penyajian yang lebih bervariasi melalui tayangan suara, gambar, animasi, video pembelajaran, dan dengan bentuknya yang digital masalah seperti, pelapukan, basah akibat hujan atau kehilangan dapat diatasi [7].

Maka dari itu penelitian ini memerlukan ebook dikarenakan bentuk digital tersebut memiliki berbagai kelebihan, terutama dalam pengaksesan yang tidak memerlukan biaya. *ebook* menjadi pilihan utama dalam penelitian ini.

2.3 Aplikasi Mobile Berbasis Android

Aplikasi berbasis android merupakan program yang dibuat untuk berjalan pada sistem operasi android. Aplikasi biasanya digunakan untuk berbagai keperluan, antara lain untuk memelihara dan mengontrol sebuah perusahaan[9], pembuatan alat pendeteksi kebakaran(13), aplikasi mobile pengemudi daring, [13] yang bisa menjadi sarana pembelajaran pemrograman bagi mahasiswa Teknik Informatika[9]

Aplikasi mobile berbasis android dapat membantu meningkatkan keterampilan dan efektivitas pembelajaran, seperti pengembangan aplikasi studi seluler berbasis android untuk meningkatkan keterampilan pemrograman pada mahasiswa Teknik Informatika[13], serta pengembangan media pembelajaran digital berbasis android yang dapat digunakan untuk mendukung pembelajaran matematika dalam penyelesaian persamaan linear satu variabel[13]. Aplikasi berbasis android dikembangkan melalui berbagai metode dan teknik, salah satunya metode *agile sprint*.

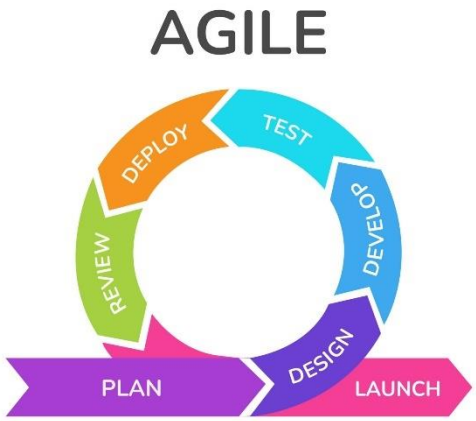
Maka dari itu penelitian yang kami lakukan memudahkan dalam mengakses data tersebut mempertimbangkan penelitian ini lebih berfokus dalam fleksibilitas penggunaan aplikasi baik dari sisi user maupun admin dalam membaca, menambahkan, menghapus ,memodifikasi(Read,Add,Delete,Edit) sebuah data atau materi yang dapat dilihat dari koleksi yang disediakan, oleh karena itu kami memiliki kelebihan dari segi mobilitas dibandingkan penelitian yang telah ada.

2.4 Metode Pengembangan Sistem

Metodologi Agile adalah metodologi SDLC yang didasarkan pada model iteratif inkremental. Agile adalah metodologi pengembangan perangkat lunak yang melibatkan beberapa proses iteratif kecil yang mengutamakan kerja sama antar tim pengembang secara intensif, yang memungkinkan penulis mendapat respons yang

cepat terhadap perubahan dan pembuatan perangkat lunak yang dapat disesuaikan dengan perubahan. Hal ini membuat metodologi agile sangat fleksibel dalam menghadapi perubahan selama proses pengembangan. Penerapan Agile memerlukan kerangka kerja yang mendukung konsep Agile , termasuk Scrum[22].

Scrum berfungsi sebagai kerangka kerja proses yang memandu pengembangan produk yang kompleks, membantu menciptakan produk yang memberikan nilai setinggi mungkin dengan cara yang produktif dan kreatif. [22].



Gambar 2.4-1 Metode AGILE

2.4.1 Sprint

Metode pelaksanaan Sprint menjadi inti dari pengembangan ini, Sprint didefinisikan sebagai periode waktu dalam pengembangan. Selama 1 bulan periode sprint, penulis bekerja untuk menyelesaikan item-item yang ada dalam sprint backlog, dengan tujuan menghasilkan fitur yang fungsional dan mencapai akhir tahap pengembangan(*sprint goal*) mulai dari *plan, design, develop, testing, dan deploy*.

2.5 Software Perancangan Course Book

2.5.1 Figma

Software Figma memungkinkan penggunanya mendesain tampilan secara kolaboratif secara real time [19]. Bisa digunakan di mana saja melalui internet, desktop Windows atau MAC OS [19]. Kelebihan dari Figma yaitu memiliki

kemampuan untuk membuat prototype dan mock-up dengan sangat cepat, dan mampu untuk berkolaborasi dengan orang lain secara *real time*[19].

Berbagai macam pilihan yang sangat bermanfaat untuk desain tersedia, dan untuk orang awam, tidak perlu lagi bersusah payah membuat prototipe aplikasi berbasis web atau smartphone[19]. Dikarenakan hal tersebut penulis memilih figma sebagai salah satu media pembuat desain serta prototype yang baik secara *online*.

2.5.2 Affinity Designer 2

Affinity Designer adalah aplikasi serbaguna dan ramah pengguna yang menawarkan alternatif cepat dan intuitif untuk Adobe Illustrator, Photoshop, dan Sketch. Aplikasi ini dirancang untuk desain vektor dan raster, sehingga cocok untuk berbagai profesi seperti desain web[20]. Antarmuka pengguna intuitif dan mudah dipahami, menjadikannya pilihan ideal bagi mereka yang memiliki pengalaman dengan program vektor[20].

Antarmuka Affinity Designer didasarkan pada interaksi berbasis sentuhan, yang memungkinkan pengguna untuk beralih di antara perangkat alat tanpa mengganggu alur kerja[20]. Aplikasi ini dilengkapi dengan alat pena dan bentuk standar untuk karya seni berbasis vektor dan piksel yang kompleks, dan bilah alat persona membagi alat vektor, piksel, dan ekspor menjadi tiga perangkat terpisah. Pengguna dapat beralih di antara persona berdasarkan tujuan mereka, seperti Draw Persona, Pixel Persona, dan Export Persona[20].

Singkatnya, penulis memilih Affinity Designer untuk digunakan sebagai alternatif Adobe Illustrator yang praktis dan murah, yang menawarkan berbagai fitur menarik untuk aplikasi menggambar vektor tradisional. Aplikasi ini cocok untuk pemula sebagai alat pembelajaran tetapi cukup canggih untuk seniman grafis lepas dengan anggaran terbatas.

2.5.3 Android Studio

Android Studio adalah Integrated Development Environment(IDE) resmi untuk pengembangan aplikasi berbasis android atau mobile development secara

umum[16]. Google menambahkan Kotlin, bahasa yang digunakan di Android Studio, sebagai bahasa pemrograman resmi untuk platform Android bersama Java dan C++.Setiap edisi Android Studio memperkenalkan fungsionalitas baru, Fitur-fiturnya meliputi WYSIWYG, fungsionalitas Seret dan Lepas untuk komponen UI, dan pratinjau tata letak di banyak layar. Android studio memiliki Struktur aplikasi berbasis Gradle yang dibuat untuk menyelesaikan file apk, serta Memproses kode[16]. Lint sebagai penganalisis kode statis yang mengidentifikasi masalah kinerja dan ketidakcocokan versi, Bukan hanya itu proguard yang terintegrasi serta Tata Letak Dasar Android dan Template Komponen yang tersedia[16]. Tujuan utama penulis menggunakan android studio dikarenakan tool resmi untuk membuat aplikasi android, serta android studio menjadi salah satu software yang mensuport bahasa kotlin sebagai salah satu bahasa pemrogramannya.

2.5.4 Firebase

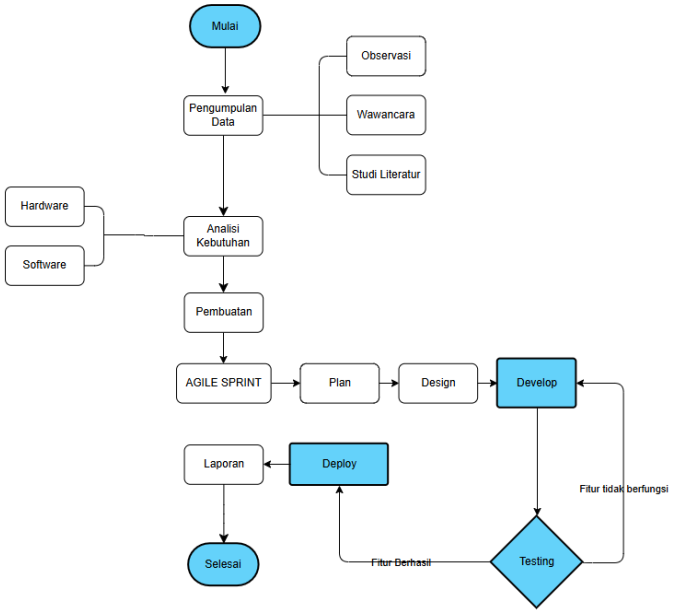
Aplikasi berbasis Android dihosting oleh server seperti Oracle SQL, Microsoft SQL Server, dan MySQL, dengan file PHP yang menghubungkannya. Firebase dikembangkan untuk aplikasi Android yang menyimpan data dalam format JSON, Sedangkan Server lainnya menyimpan data dalam struktur tabel, dengan baris dan kolom. Firebase adalah platform berbasis NoSQL[18]. Firebase dianggap sebagai platform aplikasi web. Firebase juga digunakan untuk membantu pengembangan aplikasi berkualitas tinggi. Data yang keluar akan disimpan dalam format *JavaScript Object Notation* (JSON) dan tidak memerlukan query untuk memasukkan, memperbarui, menghapus, atau menambahkan data. Backend sistem berfungsi sebagai basis data untuk menyimpan data[18].

Dari Penelitian di atas diketahui firebase menjadi pilihan tepat untuk membuat sebuah aplikasi berkualitas tinggi, serta data yang dikeluarkan otomatis tersimpan dalam bentuk JSON yang tidak memerlukan *query* memasukkan, memperbarui, menghapus, atau menambahkan data dan juga system backend yang berfungsi sebagai basis penyimpanan data.

BAB III
METODE PELAKSANAAN

3.1 Diagram Alir

Diagram alir digunakan untuk menjabarkan langkah langkah dalam pembuatan aplikasi *course book* berbasis android pada penelitian ini. Berikut diagram dalam tahapan perancangan alir yang diilustrasikan melalui flowchart di bawah.



Gambar 3.1-1 Diagram alir

3.2 Pengumpulan Data

Berikut metode yang penulis gunakan untuk mengumpulkan data yang diperlukan untuk membuat aplikasi Course Book berbasis android:

- Wawancara

Di metode ini Penulis mewawancarai kepala sekolah dan Pustakawan di UPTD SD 13 Sebelum memasuki ke tahap developing. Wawancara dilakukan secara

berhadapan langsung yang bertujuan untuk memperoleh data dan informasi untuk membantu dalam pengembangan aplikasi Course Book.

- Observasi

Di metode ini penulis mengumpulkan informasi mengenai materi pembelajaran yang berkaitan dengan data informasi yang akan disediakan di aplikasi course book agar dapat membantu guru-guru beserta murid-murid dalam mencari media pembelajaran.

- Studi Literatur

Di metode ini untuk mendapatkan pemahaman yang luas tentang topik penelitian, studi literatur menjadi Langkah dalam mempersiapkan perencanaan penelitian dan membantu menganalisi apa yang dapat dilakukan oleh penulis pada penelitian ini.

3.3 Analisi Kebutuhan

Di penelitian ini ada beberapa hardware dan software yang digunakan dalam pengembangan aplikasi Course Book berbasis android pada UPTD SD 13 sungailiat sebagai berikut.

3.3.1 Hardware

Aplikasi Course Book hanya memerlukan *Smartphone* sebagai *hardware* yang dibutuhkan untuk menjalani aplikasi tersebut, *Smartphone* disini digunakan sebagai media pengembangan aplikasi serta sebagai perangkat digital untuk memonitoring keaktifan dalam pengaksesan konten dalam aplikasi yang disediakan.

3.3.2 Software

Selain itu *software* yang dibutuhkan untuk pengembangan aplikasi Course book terdiri dari *software Graphic designer*, *Database* serta *software* Pengkodingannya antara lain :

a) Figma

Aplikasi figma disini digunakan untuk mendesain UI dari aplikasi Course Book, Figma digunakan untuk mendesain Prototipe awal yang nanti akan dibangun.

b) Affinity Designer

Affinity Designer digunakan sebagai alternatif Adobe Illustrator yang praktis dan murah, yang menawarkan angin segar untuk aplikasi menggambar vektor tradisional.

c) Android Studio

Android Studio adalah Integrated Development Environment(IDE) resmi untuk pengembangan aplikasi berbasis android atau mobile development secara umum. Di penelitian ini penulis menggunakan Bahasa Pemrogramn Kotlin untuk mengembangkan aplikasi Course Book Berbasis Android.

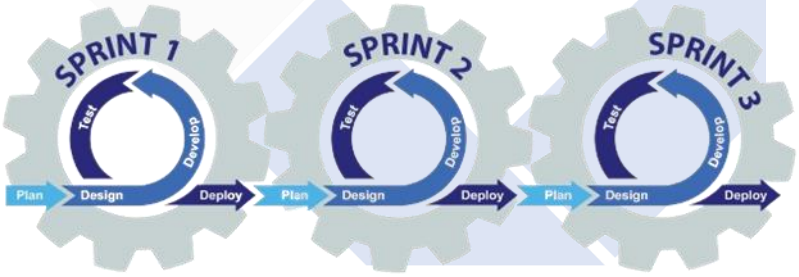
d) Firebase

firebase menjadi pilihan tepat untuk menjadi Database dalam pemembuatan sebuah aplikasi. Dalam hal ini Firebase merupakan Database dengan basis NoSQL

3.4 Pembuatan

3.4.1 Metode Pengembangan Agile

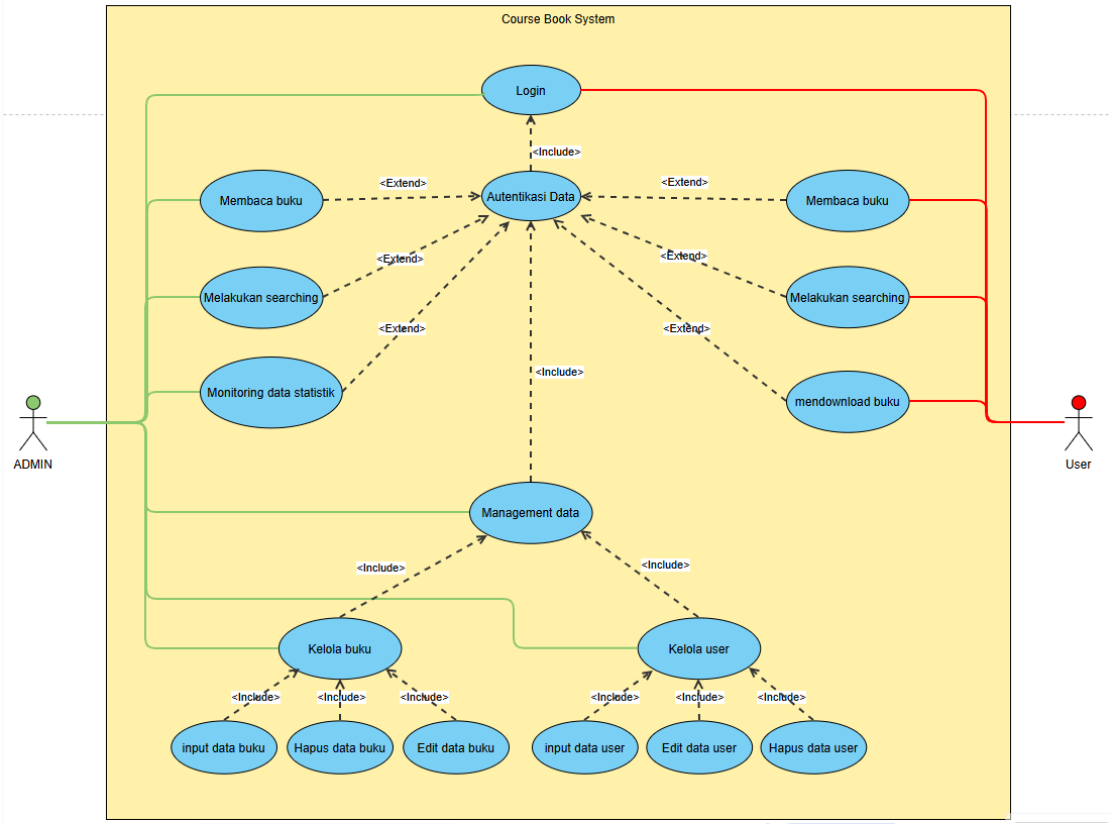
Metodologi pelaksanaan yang digunakan untuk membuat aplikasi dalam penelitian ini didasarkan pada prinsip-prinsip metodologi Agile [20], yang memberikan penekanan khusus pada sifat iteratif dan inkremental pengembangan perangkat lunak. Pendekatan ini menekankan kolaborasi intensif antara tim pengembangan, yang memungkinkan mereka merespons perubahan dengan cepat dan menciptakan perangkat lunak yang beradaptasi terhadap perubahan persyaratan. Tanggung jawab atas dinamika perubahan menjadi bagian integral dari metodologi tangkas, yang memungkinkan penyesuaian fleksibel terhadap prioritas, tugas, dan fokus pengembangan. Proses yang digambarkan dalam diagram berikut memberikan gambaran yang jelas dan sederhana tentang bagaimana perangkat lunak berkembang dari satu fase ke fase berikutnya.



Gambar 3.4-1 Sprint Agile

3.4.2 Use Case Diagram

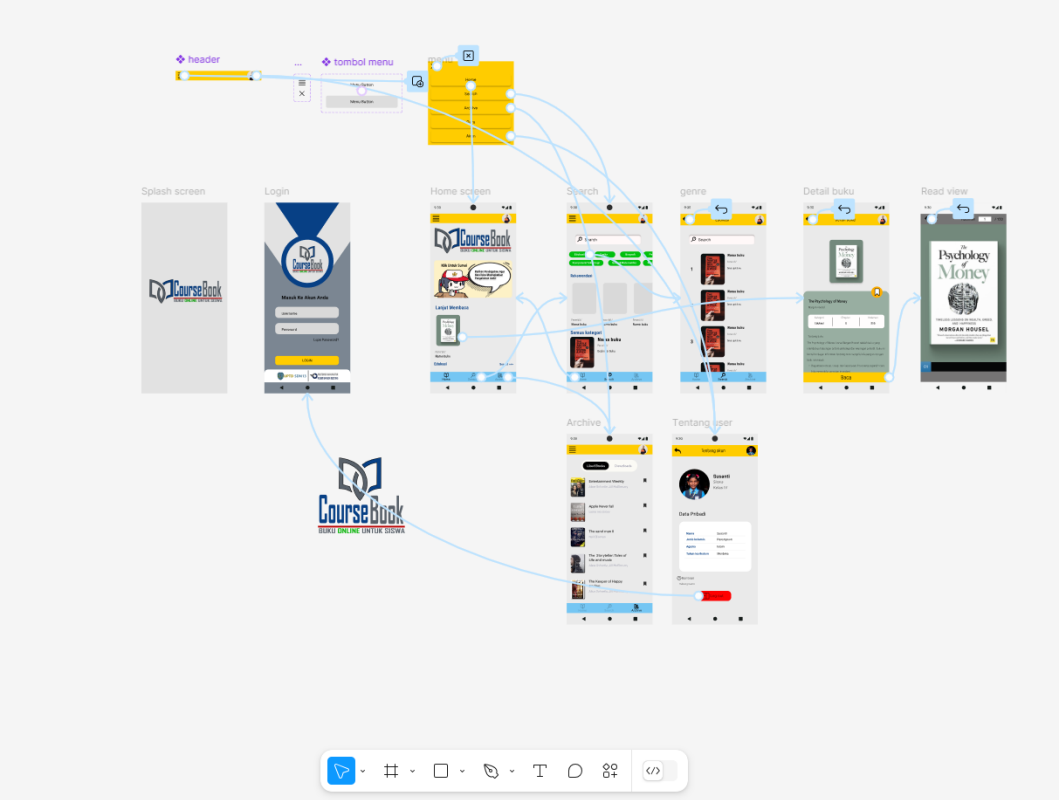
Diagram yang menggambarkan skema use case untuk admin beserta user pada aplikasi Course Book untuk mengidentifikasi interaksi yang akan terjadi antara pengguna dan sistem aplikasi.



Gambar 3.4-2 Use Case Diagram

3.4.3 Perancangan Tampilan Aplikasi

Di tahap ini penulis akan menuliskan metode perancangan yang digunakan untuk membuat tampilan dan fitur aplikasi *Course Book* Berbasis *Android,* penjelasan ditulis berdasarkan penelitian yang telah dilakukan.



Gambar 3.4-3 Desain Aplikasi di software figma

a) halaman *Login*

Desain prototipe halaman login dibuat melalui *software* figma, *resource file* seperti logo aplikasi, *textbox* dan *background* didesain menggunakan *affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box.*

b) Halaman *Home*

Desain prototipe halaman *home* dibuat melalui *software* figma, *resource file* seperti logo aplikasi, gambar *feedback* dan *background* didesain menggunakan

affinity designer 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box.*

c) Halaman *Search*

Desain prototipe halaman search dibuat melalui *software* figma, resource file seperti konten buku, *searchbox* dan *background* didesain menggunakan *affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box*.

d) Halaman *Genre*

Desain prototipe halaman *genre* dibuat melalui *software* figma, *resource file* seperti konten buku, dan background didesain menggunakan *affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box*.

e) Halaman *Archive*

Desain prototipe halaman *archive* dibuat melalui *software* figma, *resource file* seperti tombol aplikasi, icon gambar dan background didesain menggunakan *affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box*.

f) Halaman Detail Buku

Desain prototipe halaman detail buku dibuat melalui software figma, resource file seperti *cover* buku, *textbox* dan *background* didesain menggunakan *affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan

bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box*.

g) Halaman *Read view*

Desain prototipe halaman *read view* dibuat melalui *software* figma, *resource file* seperti tombolscroll dan *background* didesain menggunakan *affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box*.

h) Halaman Tentang *User*

Desain prototipe halaman *user* dibuat melalui *software* figma, *resource file* seperti *icon* tombol , *textbox* dan background didesain *menggunakan affinity designer* 2, kemudian aplikasi dikembangkan di android studio menggunakan bahasa pemrograman Kotlin yang akan diuji oleh guru bidang IT menggunakan metode *white box*.

3.4.4 Produksi

Di tahap ini pengimplementasi serta pengembangan dari seluruh aplikasi, mulai dari Tampilan Menu, fitur yang disediakan akan dibuat sesuai dengan kesepakatan yang disetujui. Di tahap produksi ini akan menggunakan metode *agile* yang mencakup : *Plan,Design,Develop,Testing,Deploy* Di setiap halaman aplikasi *course book*.

3.4.5 Alpha

Di tahap ini aplikasi akan diuji oleh seseorang yang paham dalam bidang IT, pengujian berfokus pada fungsionaltias aplikasi serta kelancaran mekanisme dari fitur yang disediakan, jika terdapat kesalahan di tahap ini, perbaikan akan segera dilakukan, baik dari segi kode atau fitur mekanisme yang berkaitan dengan permasalahan.oleh karena itu ujicoba akan terus dilakukan untuk memperbaiki masalah yang ada.

3.4.6 Beta

Di tahap ini pengujian akan berlanjut dengan mengundang guru-guru dan siswa siswi dari SD 13 untuk mengisi kuesioner UAT yang akan menghitung tingkat kelayakan sebuah aplikasi[17]. Pengujian ini lebih berfokus pada bug atau masalah yang mungkin akan ditemukan, serta dapat menyaring saran dan komentar dari penguji yang dapat meningkatkan pengalaman mereka dalam menggunakan aplikasi dan tahap ini akan diteruskan sebelum memasuki tahap perilisan. Cara pengumpulan data dilakukan

- **Bobot Kuesioner**

Sistem akan diuji dan hasil pengujian akan dinilai menggunakan kuesioner. Penilaian akan menerapkan dengan skala Likert[17]. Kategori nilai pada skala likert sebagai berikut.

Table 3.4-1 bobot kuesioner

Nilai	Kategori
1	STS
2	TS
3	KS
4	S
5	SS

Keterangan : SS = Sangat Setuju

S = Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju

- **Pertanyaan Kuesioner siswa dan guru**

Di bawah ini adalah pertanyaan atau kuesioner yang diisi oleh siswa dan guru UPTD SDN 13, pertanyaan yang dijawab menyangkut bug atau masalah yang mungkin ditemukan saat menggunakan aplikasi, serta dapat menyaring saran dan komentar dari penguji yang dapat menyimpulkan apakah aplikasi layak untuk diterapkan di UPTD SDN 13.

Table 3.4-2 Kuesioner siswa

NO.	Pertanyaan	Skala Penelitian				
		SS	S	KS	TS	STS
1.	Apakah tampilan keseluruhan aplikasi ini menarik?					

2.	Apakah koleksi buku yang disediakan menarik untuk di baca?	
3.	Apakah fitur dan tombol sesuai pada fungsinya?	
4.	Apakah informasi dan isi buku sesuai dengan yang di tampilkan dalam aplikasi?	
5.	Apakah mengalamai gangguan/masalah ketika menggunakan aplikasi?	
6.	Apakah tampilan aplikasi sesuai dengan ukuran layar perangkat anda?	
7.	Apakah Aplikasi berfungsi dengan baik di perangkat?	
8.	Apakah siswa merasa aplikasi ini dapat membantu proses pembelajaran?	
9.	Apakah koleksi buku di aplikasi sesuai dengan sumber yang di gunakan dalam pembelajaran?	
10.	Apakah siswa merasa aplikasi ini dapat membantu memberikan dampak positif untuk kedepannya?	

Table 3.4-3 Kuesioner Guru

NO.	Pertanyaan	Skala Penelitian

		SS	S	KS	TS	STS
1.	Apakah tampilan keseluruhan aplikasi ini menarik?					
2.	Apakah informasi dan isi buku sesuai dengan yang di tampilkan dalam aplikasi?					
3.	Apakah Aplikasi berfungsi dengan baik di perangkat anda?					
4.	Apakah anda merasa aplikasi ini dapat membantu dalam proses pengajaran anda?					
5.	Apakah koleksi buku di aplikasi cukup untuk dijadikan panduan pembelajaran?					
6.	Apakah anda merasa aplikasi ini dapat membantu memberikan dampak positif untuk kedepannya?					

- **Perhitungan UAT**

Di bawah merupakan Rumus yang akan digunakan untuk Perhitungan *User Acceptance Testing* (UAT) dimana hasil dari rentang kriteria yang dikeluarkan akan dikategorikan sesuai tabel di bawah[17].

Table 3.4-4 Perhitungan UAT

Rentang Kriteria	Kategori
0%-20%	Sangat Tidak Layak
21%-40%	Tidak Layak
41%-60%	Kurang Layak
61%-80%	Layak
81%-100%	Sangat Layak

Rumus dari perhitungan UAT sebagai berikut :

Nilai Tertinggi= (bobot tertinggi * jumlah pertanyaan * jumlah responden)

Nilai Akhir = (total skor/ nilai tertinggi)* 100%

Dapat diketahui perhitungan kuesioner nilai akhir yang didapatkan dengan sistem peniliaian dan dapat disimpulkan bahwa aplikasi tersebut layak atau tidak layak[17].

3.4.7 Rilis

Pada tahap ini aplikasi akan dirilis ke Playstore yang nantinya dapat diakses di *android* dan bisa didownload secara langsung. Di tahap ini bertujuan apakah aplikasi dapat berjalan dengan baik ketika didownload melalui *platform* resmi seperti *playstore*, adapun poin di atas menjadi hasil penentu akhir dari proyek akhir yang dikerjakan

3.5 Laporan

Di tahap ini akan dijelaskan secara rinci mengenai hasil akhir proyek penulis. Laporan ini berisikan penjelasan yang berkaitan dengan pembuatan aplikasi *course book* berbasis android. Laporan ini akan diselesaikan ketika proyek akhir telah mencapai tahap akhir.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Produksi

Di tahap ini penulis melakukan pembuatan fitur yang ada pada aplikasi *Course Book*, setelah mengumpulkan data berdasarkan hasil analisa wawancara dan observasi. Maka perlu dibuatkan aplikasi yang dapat mempermudah siswa dan guru dalam mengakses materi edukasi dengan mudah dan tentu saja bervariasi agar siswa tertarik dalam membaca. berikut pengembangan fitur aplikasi menggunakan metode *sprint agile* :

4.1.1 *SPRINT1*: Membuat Halaman *Home* Admin dan *User*

a) Plan

Pada Tahap ini sesuai dengan desain yang disetujui, penulis fokus dalam pembuatan menu *home* untuk admin dan *user*. Halaman *home* ini mencakupi tampilan yang berisikan banyak tombol yang masing masing mempunyai kegunaannya.

1. Admin mempunyai tampilan yang berisikan Riwayat buku dengan icon dari buku tersebut, halaman pada home admin memiliki 6 buah *navigation button* dan 1 *content button*, Dimana 6 *navigation button* ini adalah *home*, *search*, *statistics*, *add book*, *add user*, tentang akun, sedangkan *content button* digunakan untuk mengakses konten yang disediakan yaitu buku digital yang dapat dibaca.
2. User mempunyai tampilan yang hampir sama dengan admin, perbedaanya terletak pada tombol, Dimana tombol yang di akses hanya 4 buah *navigation button* dan 1 *content button*. Dimana 4 *navigation button* tersebut adalah *home, search, archive*, tentang akun sedangkan *content button* digunakan untuk mengakses konten yaitu buku digital yang dapat dibaca.

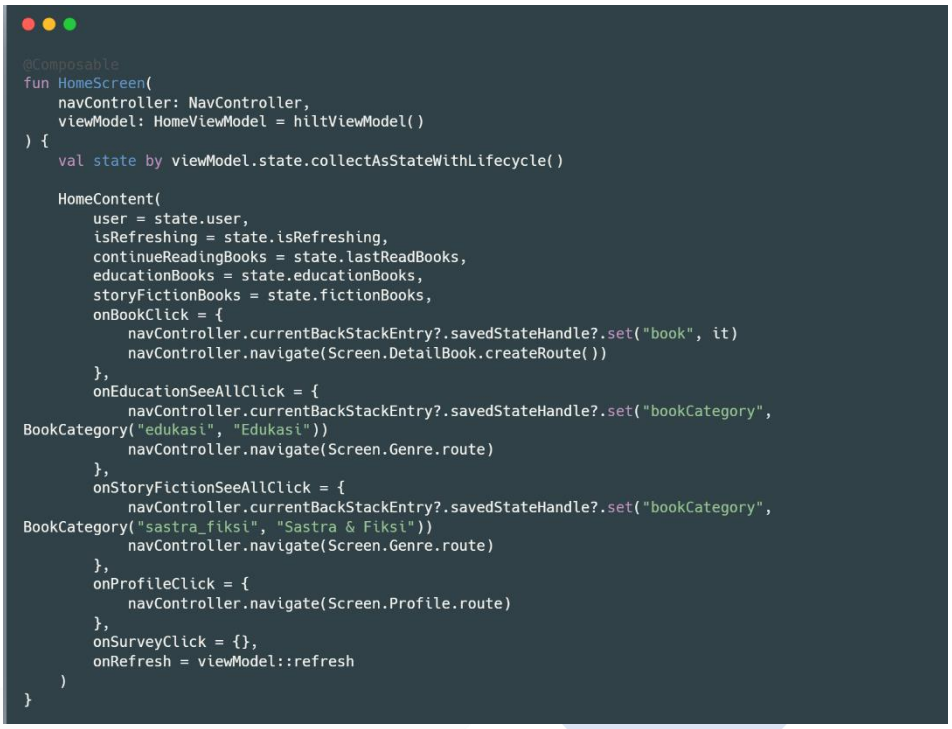
b) Design

Gambar 4.1-1Tampilan home admin dan user

Di tahap design menampilkan antarmmuka dari halaman *home* admin yang memiliki *navigation bar* seperti *home, search, statistics, add book, add user,* tentang akun, dan penempatan serta pengaksesan *content* berada di bagian Tengah yang bisa dilihat ketika *login* ke aplikasi. sedangkan untuk antarmuka *user* memiliki *navigation bar* seperti *home, search, archive,* tentang akun dan penempatan serta pengaksesan *content* dan *resource* sama dengan admin, yang menampilkan konten sesuai dengan *design*.

c) Develop

Di tahap develop atau pengembangan penulis mulai mengkoding aplikasi berdasarkan rancangan desain yang telah dibuat sebelumnya. Penulis menggunakan bahasa pemrograman kotlin untuk membuat antarmuka halaman *home admin* dan *user*, serta menjalankan logika *backend* program yang bertujuan untuk memandu fungsi dari setiap tombol ketika di klik.



```
@Composable
fun HomeScreen(
    navController: NavController,
    viewModel: HomeViewModel = hiltViewModel()
) {
    val state by viewModel.state.collectAsStateWithLifecycle()

    HomeContent(
        user = state.user,
        isRefreshing = state.isRefreshing,
        continueReadingBooks = state.lastReadBooks,
        educationBooks = state.educationBooks,
        storyFictionBooks = state.fictionBooks,
        onBookClick = {
            navController.currentBackStackEntry?.savedStateHandle?.set("book", it)
            navController.navigate(Screen.DetailBook.createRoute())
        },
        onEducationSeeAllClick = {
            navController.currentBackStackEntry?.savedStateHandle?.set("bookCategory",
BookCategory("edukasi", "Edukasi"))
            navController.navigate(Screen.Genre.route)
        },
        onStoryFictionSeeAllClick = {
            navController.currentBackStackEntry?.savedStateHandle?.set("bookCategory",
BookCategory("sastra_fiksi", "Sastra & Fiksi"))
            navController.navigate(Screen.Genre.route)
        },
        onProfileClick = {
            navController.navigate(Screen.Profile.route)
        },
        onSurveyClick = {},
        onRefresh = viewModel::refresh
    )
}
```

Gambar 4.1-2 program halaman home

d) Testing

Di tahap ini penulis memastikan sistem dari aplikasi berfungsi dan tombol bekerja dengan baik dan lancar. Penulis memastikan bahwa tidak terdapat *eror* ketika menggunakan tombol atau menjalankan menu *home* di admin maupun *user*.

e) Deploy

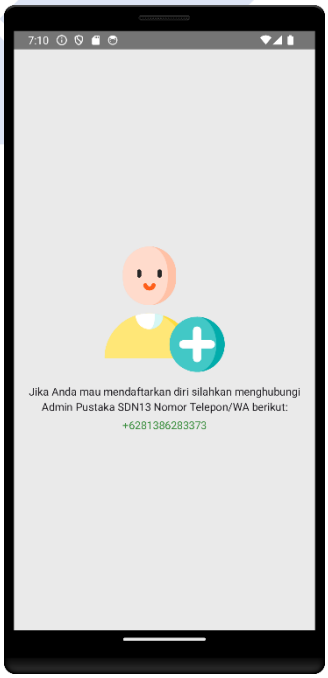
Pada tahap ini penulis memastikan menu admin dan *user* terkoneksi dengan *firebase* agar aplikasi bisa di *develop* lebih lanjut, dan fitur yang disediakan siap digunakan sesuai dengan fungsinya.

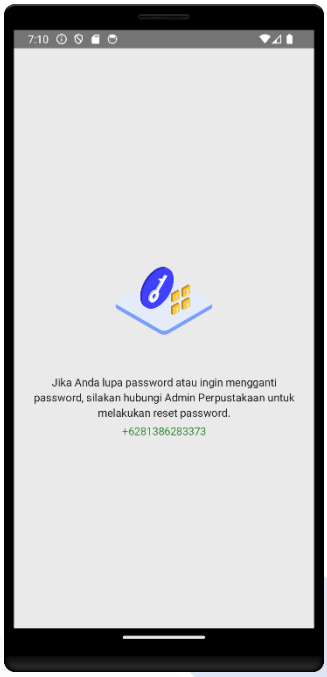
4.1.2 SPRINT2: Login Multi Level

a) Plan

Di tahap ini yaitu *login*, penulis fokus dalam menyelesaikan menu *login* yang bisa memberikan akses pada *user* melalui *multivelel login*, Dimana *user* dapat dikategorikan sesuai kebutuhan, jika *user* yang *login* menggunakan akun admin yang berstatus level 1 maka akan menampilkan menu tampilan admin yang memiliki fitur yang lebih banyak daripada akun lain, dan Ketika *user login* menggunakan akun *user* berstatus level 2 yang ditunjukan untuk siswa dan guru maka akan menampilkan tampilan menu *user* yang memiliki fitur yang dibatasi dan berbeda dengan *user* level 1. Pada tampilan antarmuka *login* terdapat *text box* yang bisa di input dengan *username* dan *password* yang terdaftar di *database*, Ketika selesai input bisa menekan tombol *login* untuk masuk atau tombol *register* ketika mau mendaftarkan akun, kemudian sistem akan menjalankan logika untuk menvalidasi *username* dan *password* apakah akun tersebut terdaftar atau tidak.

b) Design





Gambar 4.1-3 Halaman Login

Di tahap ini penulis mendesain antarmuka login dengan menempatkan *resource* logo aplikasi di atas, selanjutnya meletekan 2 *text box* yang digunakan untuk menginput *username* dan *password*. *Username* dan *password* menggunakan tipe *string* untuk menyimpan teks ke database. Kemudian menempatkan *button* yang bertuliskan login yang dimana *button* ini berfungsi untuk memvalidasi *username* dan *password* yang diinput apakah terdaftar di database, jika tidak valid maka akan muncul notifikasi *usernotfound*, untuk terakhir menambahkan 2 tombol yaitu *register* dan lupa password yang akan mengarahkan user ke halaman *contact support*.

c) *Develop*

Di tahap *develop* atau pengembangan penulis menuliskan kode menggunakan Bahasa pemrograman Kotlin untuk mendesain antarmuka login dan menjalankan logika *backend* dari program, apabila *button login* di tekan akan mengirimkan data *string* yang diinput dan memvalidasi data tersebut apakah sesuai dengan di database, jika sesuai akan melakukan perintah *getuser* untuk menyimpan data login ke *local storage* perangkat, yang akan memudahkan user untuk masuk ke aplikasi tanpa harus login ulang. Terakhir membuat tombol *register* dan lupa password dan logika *backend* program yang mengarahkan *user* ke halaman *contact support* ketika mengklik tombol tersebut.

```
class FirebaseServiceImpl @Inject constructor(
    private val firestore: FirebaseFirestore,
    private val storage: FirebaseStorage,
    private val preferences: SharedPreferences,
) : FirebaseService {

    override suspend fun isAuthenticated(): Boolean {
        return preferences.getString("username", null) != null
    }

    override suspend fun isAdmin(): Boolean {
        return preferences.getString("role", null) == "admin"
    }

    override suspend fun login(username: String, password: String): User? {
        val userSnapshot = firestore.collection("users")
            .document(username)
            .get()
            .await()
        val user = userSnapshot.toObject<User>() ?: throw Exception("User not found")

        val isPasswordValid =
            BCrypt.verifyer().verify(password.toCharArray(), user.hashedPassword).verified

        if (isPasswordValid) {
            preferences.edit {
                putString("username", user.username)
                putString("hashedPassword", user.hashedPassword)
                putString("role", user.role)
                putString("fullName", user.fullName)
                putString("classLevel", user.classLevel)
                putString("gender", user.gender)
                putString("religion", user.religion)
                putString("curriculumYear", user.curriculumYear)
                putString("imageUserUrl", user.imageUserUrl)
                putInt("totalBooksDownloaded", user.totalBooksDownloaded)
                putInt("totalBooksRead", user.totalBooksRead)
                putStringSet("lastReadBooks", user.lastReadBooks.toSet())
                putStringSet("likedBooks", user.likedBooks.toSet())
                putStringSet("downloadedBooks", user.downloadedBooks.toSet())
            }
        } else throw Exception("Invalid password")

        return user
    }

    override suspend fun addUser(imageUserUri: Uri, user: User) {
        val isUserExist = firestore
            .collection("users")
            .document(user.username)
            .get()
            .await()
            .exists()

        if (isUserExist) {
            throw Exception("User already exists")
        }

        val imageUrl = getStorageUrl(imageUserUri, "users/image", user.username, "jpg")
        firestore.collection("users").document(user.username).set(
            user.copy(
                username = user.username,
                imageUserUrl = imageUrl,
                hashedPassword = BCrypt.withDefaults()
                    .hashToString(12, user.hashedPassword.toCharArray())
            )
        ).await()
    }
go(f, seed, [])
}
```

Gambar 4.1-4 multilevel login

d) Testing

Testing atau pengujian dapat dilakukan dengan memverifikasi fungsi dari fitur yang terdapat pada halaman, apakah berfungsi dan beroperasi sesuai dengan plan yang telah ditetapkan. Pengujian ini berdasarkan respon yang diberikan oleh sistem apakah sudah tepat, pertama menguji sistem ketika menginputkan *username* dan *password* yang benar maupun salah apakah memberikan respon sesuai dengan rancangan, kedua apakah sistem memvalidasi dengan tepat sesuai dengan string yang diinput yang mengharuskan kesamaan mutlak dengan data di *firebase*, ketiga apakah *user* yang diinput akan menampilkan halaman sesuai level dari akun tersebut, lvl 1 akan menampilkan halaman *home* admin dan lvl 2 akan menampilkan halaman *home* siswa dan guru. Semua hasil pengujian penulis dokumentasikan untuk memastikan kelancaran serta keefektifan dari aplikasi yang dibuat.

e) Deploy

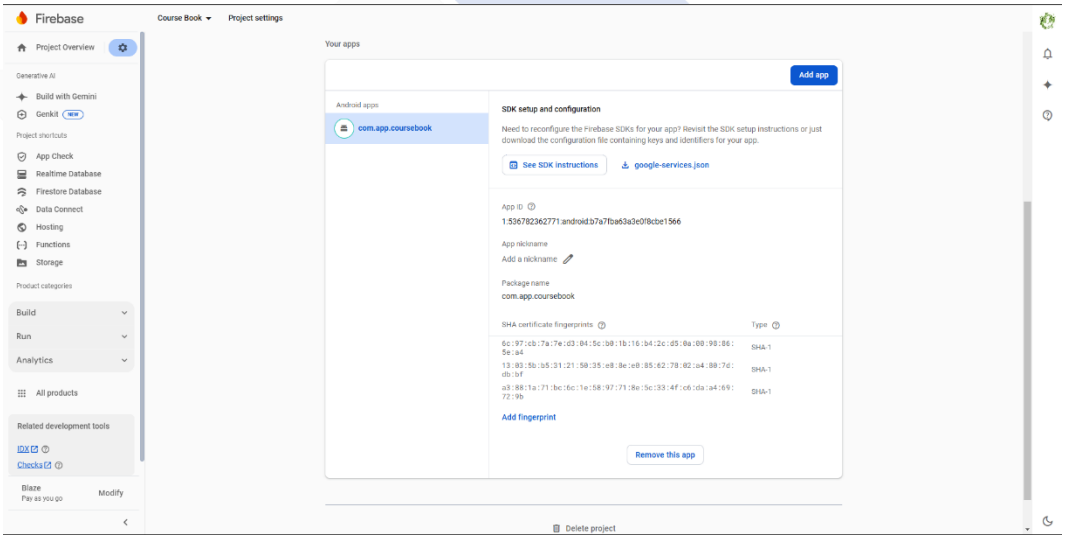
pada tahap ini menandakan fitur *login multilevel* berhasil dibangun dan berfungsi sesuai perencanaan dimana, aplikasi dapat *login* dan memvalidasi akun yang tersimpan di dalam *firebase* serta dapat menampilkan tampilan sesuai dengan lvl *user* yang disimpan. Maka dari itu dipastikan fitur *login* dapat berjalan dengan optimal di aplikasi penulis.

4.1.3 *SPRINT3*: Menghubungkan *Firebase* ke android studio

a) *Plan*

Pada tahap perancangan ini akan berfokus untuk menyambungkan firebase ke android studio, alasan penulis memilih firebase dikarenakan fitur yang tersedia sangat banyak dan mudah digunakan ketika mengkoding di android studio. Firebase disini digunakan untuk menyediakan Real-time database yang dapat sinkronisasi data secara langsung di perangkat pengguna,serta menyediakan layanan authentication yang akan menangani login relatif mudah dan dikarenakan sistem yang disediakan juga tidak memakan biaya.

b) *Design*



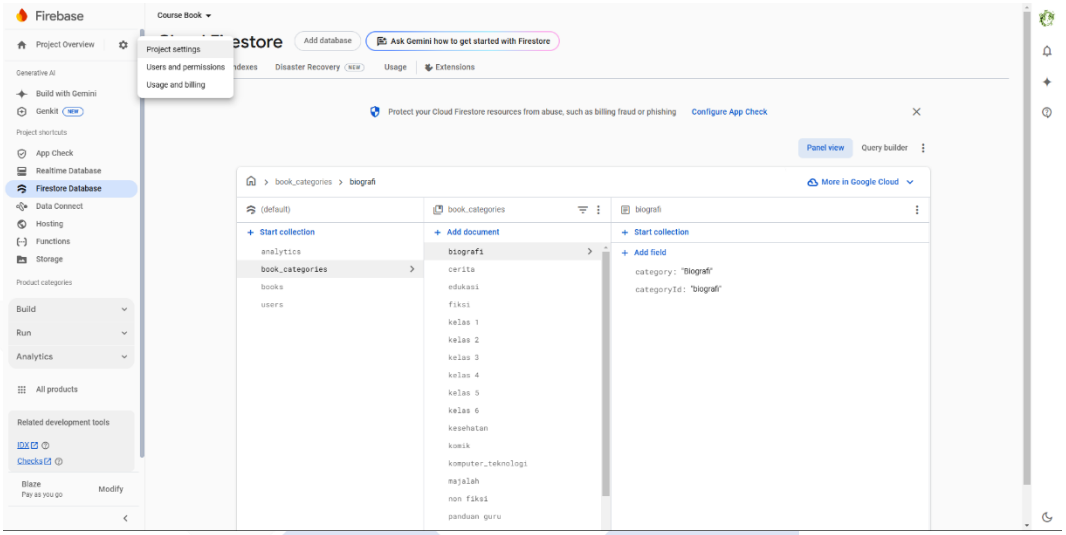
Gambar 4.1-5 Firebase

Desain yang ditampilkan merupakan halaman dari firebase itu sendiri, yang menyediakan banyak fitur yang dapat diakses secara gratis. Dengan mengikuti instruksi panduan yaitu dengan mencopy code SHA certificate fingerprints dari gradle script aplikasi di android studio dan menambahkan code SHA tersebut ke Firebase console.

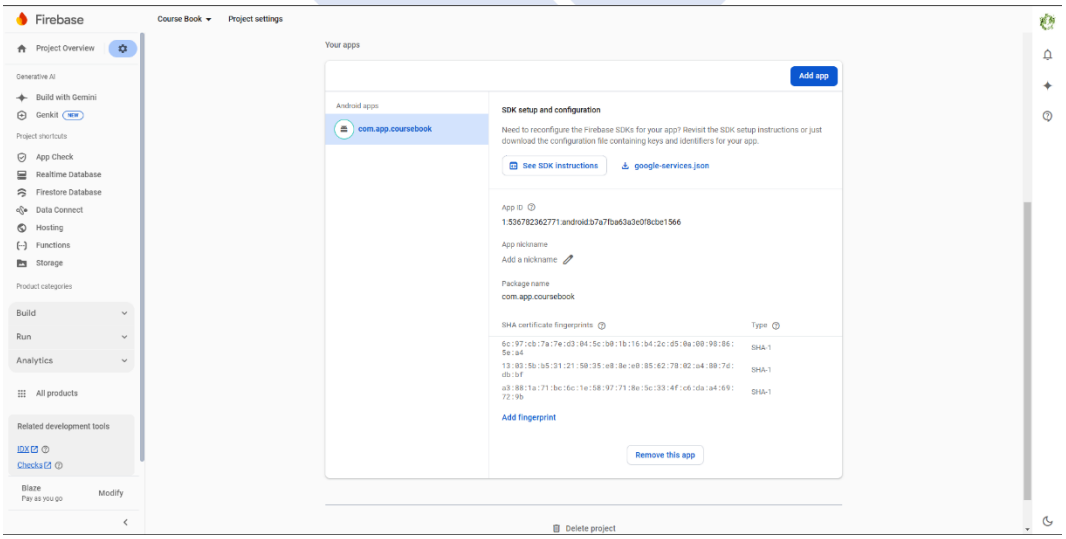
c) *Develop*

Pada tahap develop ini setelah membuat database di firebase penulis pertama membuka firebase console, kemudian membuka setting project dari

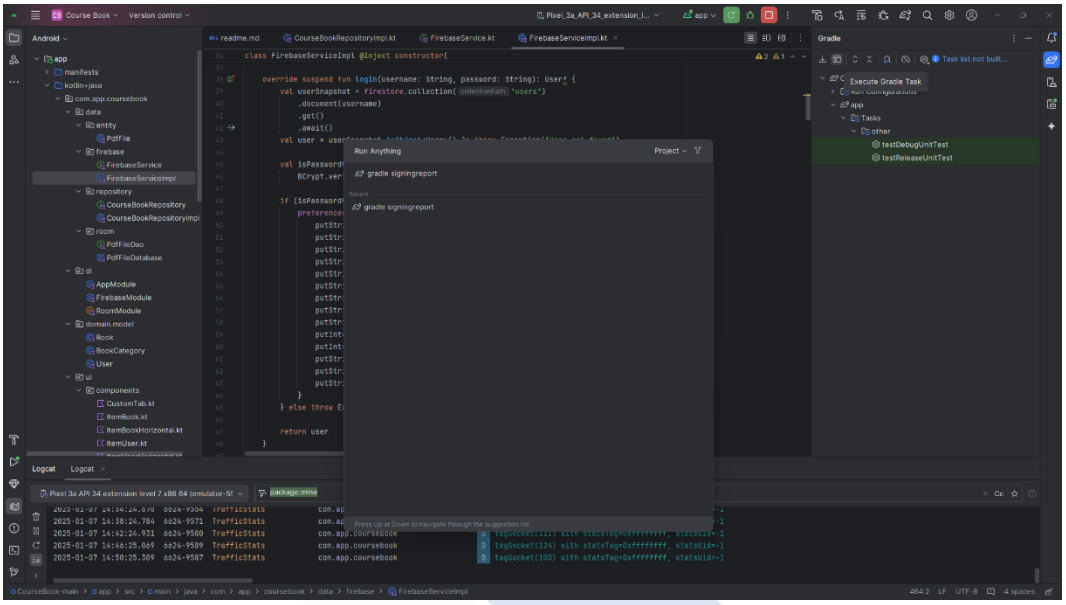
firebase database yang di buat, membuka aplikasi di android studio, execute gradle task dan menuliskan perintah signingreport, maka akan memunculkan SHA *certificate* yang dibutuhkan *firebase console*, copy kode SHA-1 yang tertulis dan paste kode tersebut ke *add fingertips* kemudian simpan. Langkah terakhir yaitu mengunduh file google-service.json dan meletakan file tersebut ke folder app aplikasi *course book* agar *service* berjalan dan terkoneksi jika dipanggil di codingan.

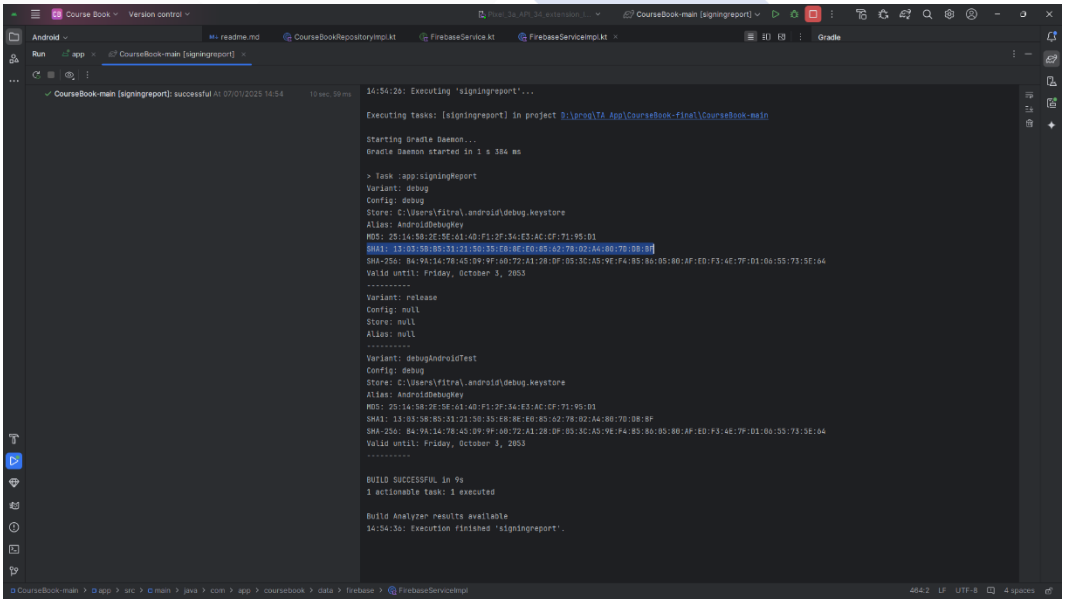


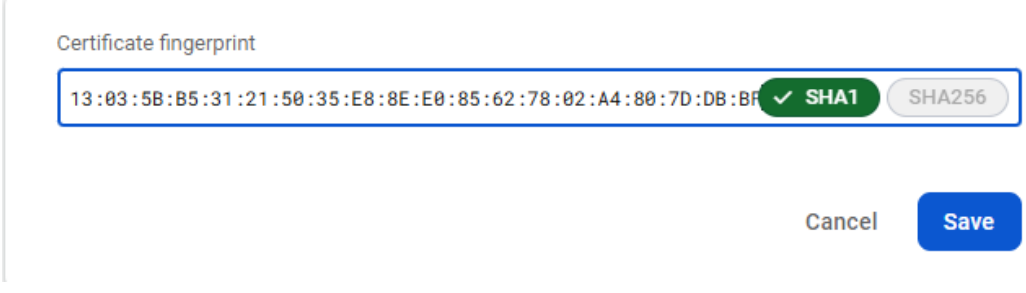
Gambar 4.1-6 tampilan firebase database

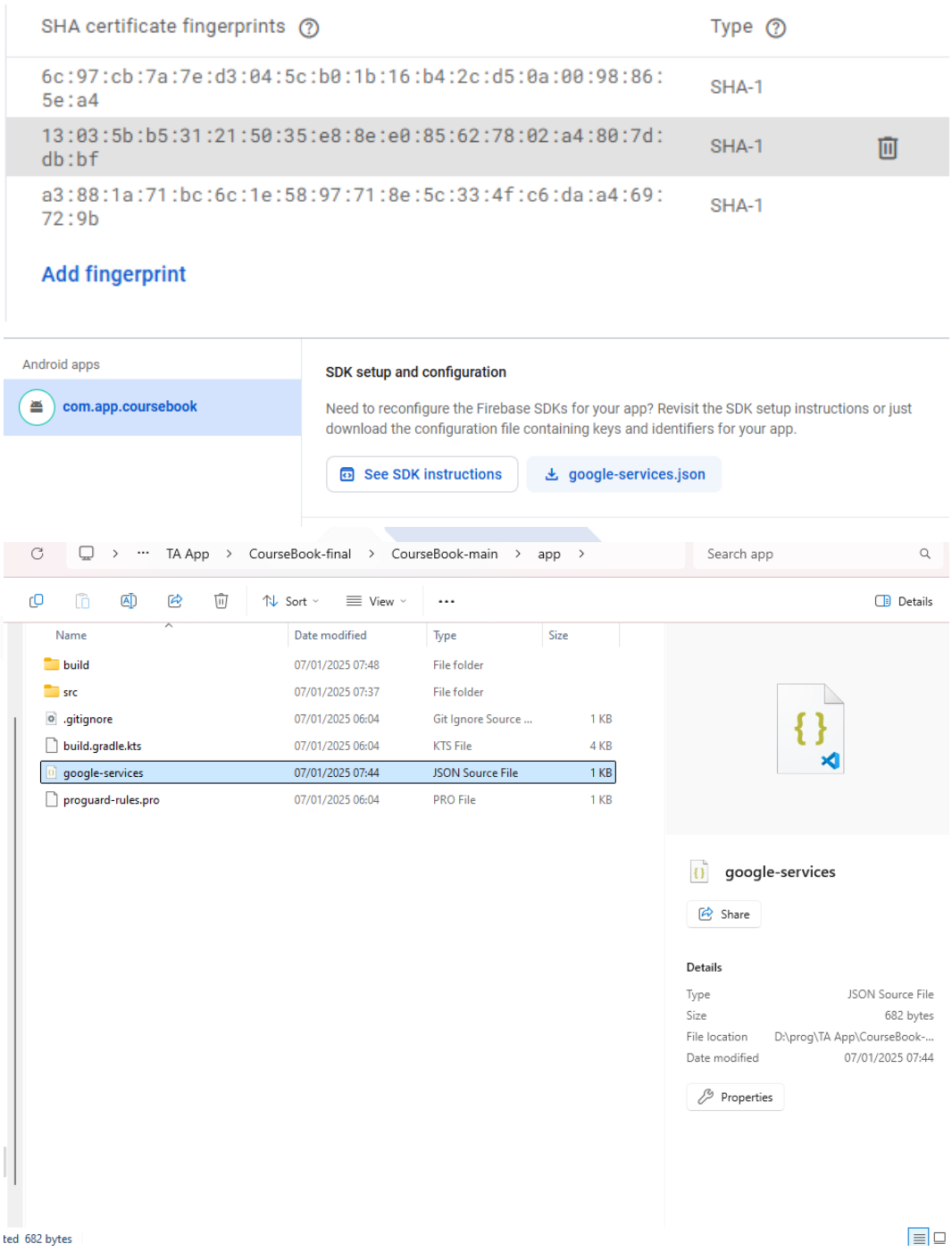


Gambar 4.1-7 Project settingg firebase






Certificate fingerprint
13:03:5B:B5:31:21:50:35:E8:8E:E0:85:62:78:02:A4:80:7D:DB:BF
SHA1
SHA256
Cancel
Save



Gambar 4.1-8 Penempatan googleservice.json

d) Testing

Pada tahap *testing* ada beberapa hal yang dapat diuji, dengan memanggil data yang ada di firebase database, penulis membuat sebuah akun dengan lvl 2 yang terkoneksi dengan memanggil data yang ada di database, kodingan getuser() akan memanggil string data yang ada di firebase kemudian sistem memvalidasi apakah akun tersebut tersedia di database ketika menekan tombol login user akan diarahkan ke tampilan home siswa dan guru, jika tidak maka akan menampilkan usernotfound. Pengujian ini ditujukan untuk mengambil sebuah data pada firebase secara langsung dari android studio.

e) Deploy

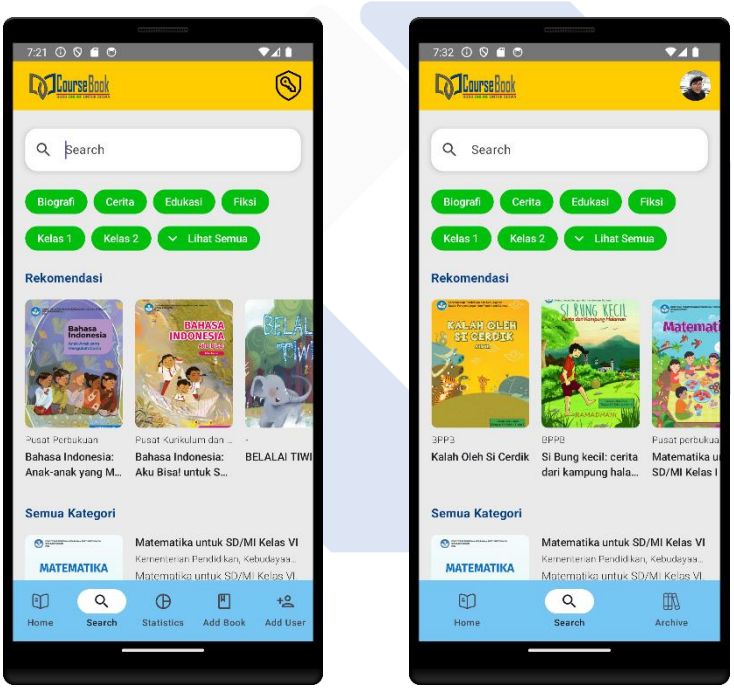
Di tahap ini menunjukan *firebase* berhasil dikoneksi ke android studio, dengan berhasilnya login ke akun *user* membuktikan aplikasi terkoneksi dan dapat mengambil data di *firebase* secara *real time*. Selanjutnya penulis memastikan *firebase* siap untuk digunakan.

4.1.4 *SPRINT4*: Membuat Halaman *Search* Admin dan *User*

a) Plan

Pada tahap perancangan halaman *search* admin dan *user*, penulis fokus dalam menyelesaikan fitur *search* yang menyaring *string* inputan pada *search bar* lalu menampilkan tampilan buku sesuai dengan yang dicari. Agar fitur search bisa ditampilkan,penulis mengambil *list book* dari database kemudian menyimpan *book cache* ke dalam *local storage*, dengan data tersebut bisa *return* data *booktitle* yang tersimpan dan memfilter buku tersebut untuk ditampilkan lagi di halaman *search*.

b) Design



Gambar 4.1-9 Tampilan Halaman Search admin dan user

Di proses ini antarmuka dari halaman *search* admin dan *user* memiliki konten yang sama, memeliki search bar yang bisa menginputkan teks serta memfilter buku sesuai inputan. Adapun pilihan *genre* yang berfungsi untuk mengkategorikan buku sesuai *genre* yang di pilih, tombol tersebut masing masing memiliki fungsi untuk menampilkan buku yang berbeda. Perbedaan halaman *search* admin dan *user* terletak pada *navigation bar* yang

ditampilkan, ketika diperhatikan tombol navigasi dari admin lebih banyak dan mencakup banyak hal dibandingkan *user* biasa.

c) Develop

Pada tahap *develop* ini penulis mulai membuat aplikasi sesuai dengan rancangan yang telah dibuat sebelumnya, menggunakan bahasa pemrograman Kotlin untuk membuat antarmuka dan menulis logika dari *backend program search bar* serta *genre tab* agar sistem dapat mengambil *listbook* dari *database*, mereturn data *booktitle* atau *value* yang tersimpan dan memfilter *value* yang dicari untuk ditampilkan lagi di halaman *search*.

```
override suspend fun searchBooks(query: String): List<Book> {
    return bookCache.filter { book ->
        book.bookTitle.contains(query, ignoreCase = true)
    }
}

override suspend fun getBooksByCategory(bookCategory: String): List<Book> {
    return bookCache.filter { book ->
        book.bookCategory == bookCategory
    }.take(4)
}

override suspend fun getBooks(refresh: Boolean): List<Book> {
    if (bookCache.isEmpty() || refresh) {
            val newBooks = firebaseService.getBooks()

            bookCache.clear()
            bookCache.addAll(newBooks)
        }
        return bookCache
}
```

Gambar 4.1-10 program search

d) Testing

Di tahap pengujian ini, penulis memastikan *search bar* dapat menginput *teks* dan menampilkan data sesuai dengan inputan yang dicari, *genre tab* memfilter konten sesuai dengan *genre* yang di cari, halaman *search* dapat menampilkan koleksi buku dan isi dari buku dapat dibaca. Diharuskan untuk fitur yang disediakan siap untuk digunakan.

e) Deploy

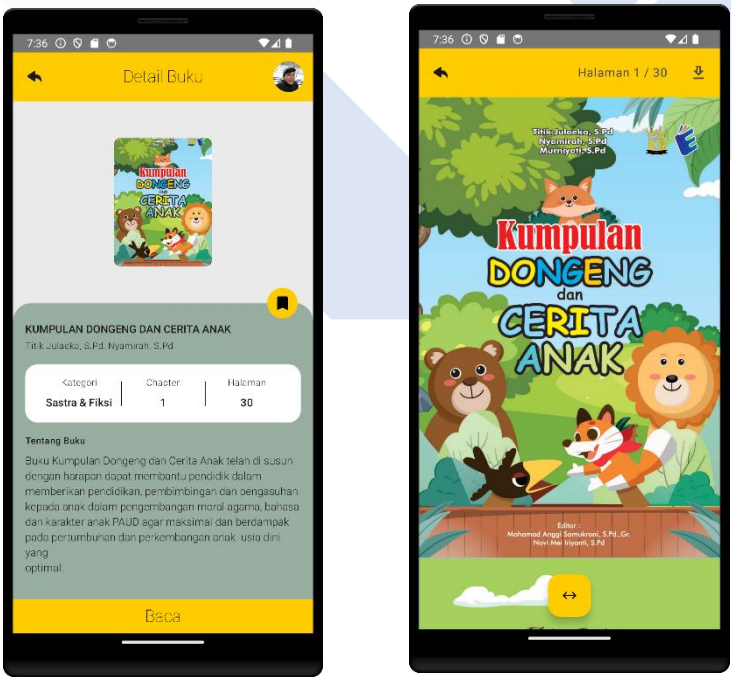
Di tahap ini Halaman *home* fitur, dan tombol di admin dan *user* berfungsi sesuai dengan rancangan dan penulis memastikan untuk fitur dipastikan beroperasional atau siap digunakan

4.1.5 *SPRINT5*: Membuat halaman detail buku dan halaman baca

a) Plan

Pada tahap perancangan halaman detail buku dan halaman baca, file buku menjadi sorotan utama,buku dalam perancangan aplikasi, dikarenakan file tersebut merupakan konten yang akan diakses, jenis tipe file yang akan penulis gunakan adalah pdf. Ketika user memilih sebuah buku, maka buku yang dipilih dapat menampilkan detail dari buku yang diambil dari database Books yang berisikan informasi tentang *author*, *category*, *bookid, bookpdfurl*, *booktitle*, *chapter*, *description*, *imageThumbnailurl*, *lastReadTimeStamp*, *publisher*, *totalDownloadCount*, *totalLikeCount*, *totalPages*, dan *totalReadCount*. Pada tombol baca di halaman detail buku, jika ditekan maka akan mengambil data *bookpdfurl* dan menampilkan isi dari link tersebut ke halaman Baca atau *readview*.

b) Design



Gambar 4.1-11 tampilan detail buku dan baca

Tampilan dari antarmuka halaman detail buku yaitu menempatkan cover buku ditengah dan meampilkan detail dari buku seperti *booktitle*, *author*,

category, *chapter*, *totalPages*, *description* dan dibawah layar terdapat tombol baca jika ditekan akan mengarahkan user ke halaman baca. Halaman baca sendiri menampilkan konten pdf di tengah yang mencakupi hampir seluruh layar, dimana cuma menyisahkan detail halaman konten.

c) Develop

Di tahap *develop* ini penulis membuat aplikasi sesuai dengan rancangan yang dibuat sebelumnya, bahasa pemrograman Kotlin digunakan untuk menjalankan logika *backend program* seperti mengakses *firebase* dan menampilkan detail buku berdarkan data yang disimpan dan tombol baca berfungsi untuk mengarahkan *user* ke halaman baca dan menampilkan isi bookpdfurl dari buku ketika ditekan.

Gambar 4.1-12Program halaman detail buku

```
package com.app.coursebook.ui.screen.book_reader

import android.net.Uri
import android.util.Log
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.app.coursebook.data.repository.CourseBookRepository
import com.app.coursebook.domain.model.Book
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.SharingStarted
import kotlinx.coroutines.flow.onStart
import kotlinx.coroutines.flow.stateIn
import kotlinx.coroutines.flow.update
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class BookReaderViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository,
) : ViewModel() {

    private val _state = MutableStateFlow(BookReaderState())
    val state = _state
        .onStart { getUser() }
        .stateIn(
            viewModelScope,
            SharingStarted.WhileSubscribed(5000L),
            BookReaderState()
        )

    fun onEvent(event: BookReaderEvent) {
        when (event) {
            is BookReaderEvent.OnPdfDownload -> {
                downloadPdf(event.book, event.fileUri)
            }
        }
    }

    private fun getUser() {
        viewModelScope.launch {
            _state.update {
                it.copy(
                    user = courseBookRepository.getUser()
                )
            }
        }
    }

    private fun downloadPdf(book: Book?, fileUri: Uri) {
        viewModelScope.launch {
            if (book != null) {
                try {
                    courseBookRepository.savePdfToRoom(book, fileUri)
                    courseBookRepository.addDownloadedBook(book)
                } catch (e: Exception) {
                    Log.d("DOWNLOAD", "downloadPdf: $e")
                    e.printStackTrace()
                }
            }
        }
    }
}
```

Gambar 4.1-13 program haklaman baca

d) Testing

Pada tahap pengujian atau *testing* penulis memastikan bahwa tampilan, button dapat berfungsi sesuai dengan yang direncanakan. Pengujian dilakukan dengan mengakses buku melalui aplikasi dan melihat apakah penempatan detail dari buku seperti *booktitle*, *author*, *category*, *chapter*, *totalPages*, *description* sesuai dengan rancangan. Tombol baca menampilkan konten buku yang bisa dibaca dengan mengscroll layar atau menggeser layar.

e) Deploy

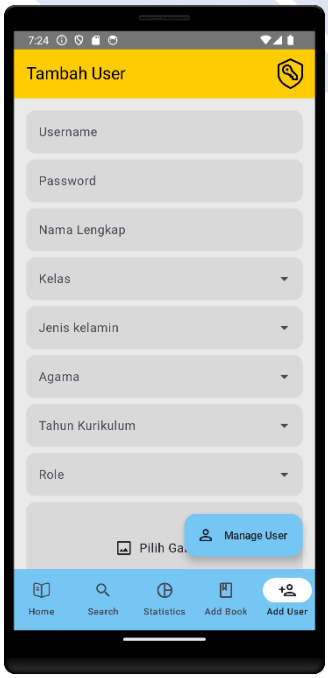
Di tahap deploy aplikasi berfungsi sesuai dengan yang dirancang, menampilkan detail dari buku serta dapat mengakses konten dari buku tersebut

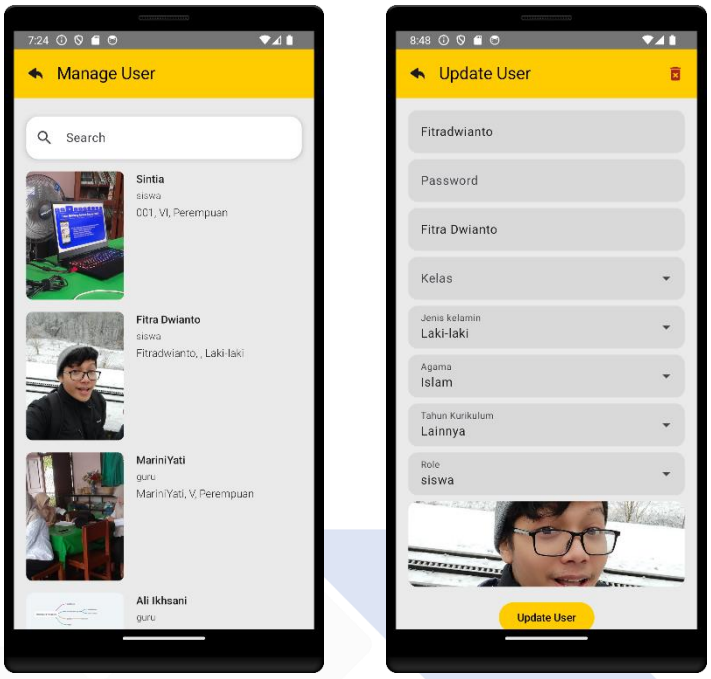
4.1.6 SPRINT6: Add user

a) Plan

Di tahap perancangan *add user*, fitur yang akan dibuat hanya tersedia untuk user level admin, admin dapat mengakses halaman *add user* melalui *navigation bar*, jika admin menekan *navigation bar* bertuliskan *add user* maka akan mengarahkan ke halaman tersebut. Jika user *login* menggunakan akun *user* selain admin maka fitur tersebut tidak ada. *Add user* langsung terkoneksi dengan database, dimana penginputan data seperti *username*, *password*, nama lengkap, kelas, jenis kelamin, agama, tahun kurikulum, role dapat dilakukan sesuai dengan *textbox* yang ditujukan, maka inputan tersebut akan disimpan langsung ke *firebase* ketika admin menekan tombol tambah *user*. Juga dibutuhkan tombol untuk *memanage user* yang diperlukan untuk mengedit, *delete*, *update* data dari *user*.

b) Design





Gambar 4.1-14 tampilan add user

Penulis mendesain antarmuka secara *simplistic* dengan mengikuti rancangan yang dibuat sebelumnya, menambahkan 8 *text box* untuk menginput *username*, *password*, nama lengkap, kelas, jenis kelamin, agama, tahun kurikulum, *role* dan pilih gambar untuk memasukan gambar dari profile user, 2 buah tombol yaitu tambah *user* dan *manage user* yang berfungsi untuk menambahkan, mengedit, *delete*, dan *update user*.

c) *Develop*

Di tahap develop ini penulis membuat aplikasi menggunakan bahasa pemrograman Kotlin untuk membuat antarmuka dan *backend program* yang mengkoneksikan inputan data ke *firebase* setelah admin mengisi data dan menekan tombol tambah user inputan akan langsung tersimpan ke *firebase* dan *user* yang ditambah, bisa langsung digunakan. Tombol *manage user* akan menampilkan deretan *user* yang terdaftar yang dapat diedit, *delete*, *update* oleh admin.

```
@HiltViewModel
class AddUserViewModel @Inject( constructor(
    private val courseBookRepository: CourseBookRepository
) : ViewModel() {

    private val _state = MutableStateFlow(AddUserState())
    val state = _state
        .onStart { getUser() }
        .stateIn(
            scope = viewModelScope,
            started = SharingStarted.WhileSubscribed(5000),
            initialValue = AddUserState()
        )

    fun onEvent(event: AddUserEvent) {
        when (event) {
            is AddUserEvent.OnUsernameChange -> _state.update {
                it.copy(
                    username = event.username
                )
            }
            is AddUserEvent.OnClassLevelChange -> _state.update {
                it.copy(
                    classLevel = event.classLevel
                )
            }
            is AddUserEvent.OnCurriculumYearChange -> _state.update {
                it.copy(
                    curriculumYear = event.curriculumYear
                )
            }
            is AddUserEvent.OnFullNameChange -> _state.update {
                it.copy(
                    fullName = event.fullName
                )
            }
            is AddUserEvent.OnGenderChange -> _state.update {
                it.copy(
                    gender = event.gender
                )
            }
            is AddUserEvent.OnImageUserChange -> _state.update {
                it.copy(
                    imageUserUri = event.imageUserUri
                )
            }
            is AddUserEvent.OnPasswordChange -> _state.update {
                it.copy(
                    hashedPassword = event.password
                )
            }
            is AddUserEvent.OnReligionChange -> _state.update {
                it.copy(
                    religion = event.religion
                )
            }
            is AddUserEvent.OnRoleChange -> _state.update {
                it.copy(
                    role = event.role
                )
            }
            is AddUserEvent.OnSubmit -> onSubmit(
                event.username,
                event.password,
                event.role,
                event.fullName,
                event.classLevel,
                event.gender,
                event.religion,
                event.curriculumYear,
                event.imageUserUri
            )
        }
    }
```

```
    private fun onSubmit(
        username: String,
        password: String,
        role: String,
        fullName: String,
        classLevel: String,
        gender: String,
        religion: String,
        curriculumYear: String,
        imageUserUri: Uri?,
    ) {
        _state.update { it.copy(isLoading = true) }
        viewModelScope.launch {
            val user = User(
                username = username,
                hashedPassword = password,
                role = role,
                fullName = fullName,
                classLevel = classLevel,
                gender = gender,
                religion = religion,
                curriculumYear = curriculumYear,
            )
            if (imageUserUri != null) {
                try {
                    courseBookRepository.addUser(imageUserUri, user)
                    _state.update {
                        it.copy(
                            eventMessage = "Sukses menambahkan user",
                            isLoading = false,
                            isSuccess = true
                        )
                    }
                } catch (e: Exception) {
                    _state.update { it.copy(isLoading = false, eventMessage = e.message) }
                }
            } else {
                _state.update {
                    it.copy(
                        isLoading = false,
                        eventMessage = "Image user is required"
                    )
                }
            }
        }
    }

    private fun getUser() {
        viewModelScope.launch {
            _state.update {
                it.copy(
                    user = courseBookRepository.getUser()
                )
            }
        }
    }
}
```

Gambar 4.1-15 program add user

d) Testing

Pada tahap ini penulis menguji apakah inputan *username*, *password*, nama lengkap, kelas, jenis kelamin, agama, tahun kurikulum, *role* dan pilih gambar dapat tersimpan di *firebase* ketika menekan tombol tambah *user*, jika tersimpan maka akun yang dibuat bisa langsung digunakan. Tombol manage user bisa mengedit,*delete*,*update* data *user* yang tersimpan.

e) Deploy

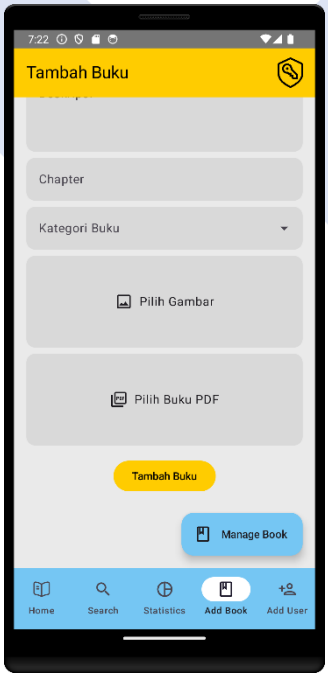
Di tahap *deploy* menunjukan fitur dari *login multilevel* sangat diperlukan untuk pengembangan aplikasi penulis, dimana fitur tersebut dapat membantu admin dalam penginputan data yang praktis tanpa harus membebani admin dalam penginputan data melalui kodingan yang dapat memakan waktu yang lama. Fitur yang digunakan sudah memiliki peforma yang optimal dimana penginputan data akan tersimpan secara langsung dan akun bisa diakses saat *input* tersimpan.

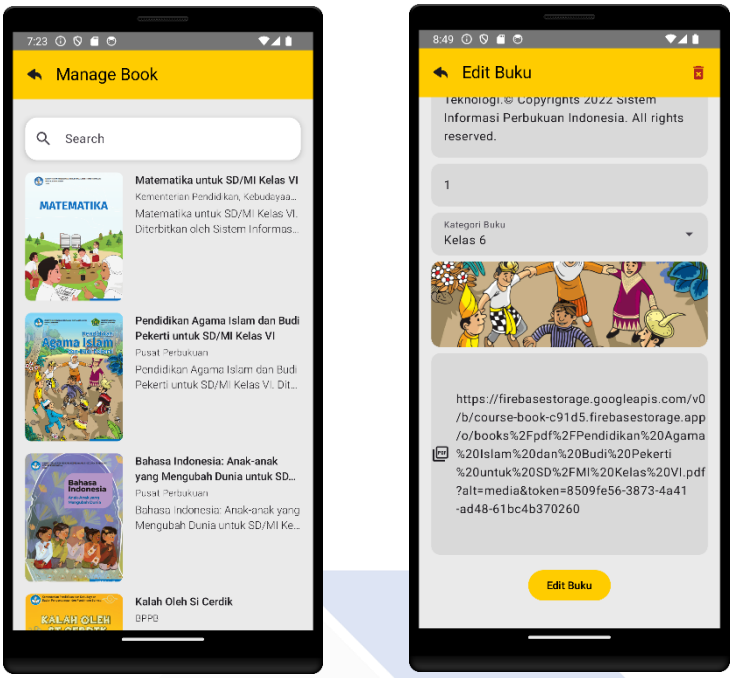
4.1.7 *SPRINT7*: *Add* book

a) Plan

Di tahap perancangan *add book*, fitur yang akan dibuat hanya tersedia untuk user level admin, admin dapat mengakses halaman *add book* melalui *navigation bar*, jika admin menekan *navigation bar* bertuliskan *add book* maka akan mengarahkan ke halaman tersebut. Jika user *login* menggunakan akun *user* selain admin maka fitur tersebut tidak ada. *Add book* langsung terkoneksi dengan database, dimana penginputan data seperti judul buku, pengarang, publisher, deskripsi, chapter, kategori, pilih gambar, pilih buku pdf dapat dilakukan sesuai dengan *textbox* yang ditujukan, maka inputan tersebut akan disimpan langsung ke *firebase* ketika admin menekan tombol tambah buku. Juga dibutuhkan tombol untuk *memanage book* yang diperlukan untuk mengedit, *delete*, *update* data dari buku.

b) Design





Gambar 4.1-16 tampilan add book

Penulis mendesain antarmuka secara *simplistic* dengan mengikuti rancangan yang dibuat sebelumnya, menambahkan 8 *text box* untuk menginput judul buku, pengarang, *publisher*, deskripsi, *chapter*, kategori, pilih gambar, pilih buku pdf dan 2 buah tombol yaitu tambah buku dan *manage book* yang berfungsi untuk menambahkan, mengedit, *delete*, dan *update* buku.

c) Develop

Di tahap *develop* ini penulis membuat aplikasi menggunakan bahasa pemrograman Kotlin untuk membuat antarmuka dan *backend program* yang mengkoneksikan inputan data ke *firebase*, setelah admin mengisi data dan menekan tombol tambah buku inputan, akan langsung tersimpan ke *firebase* dan buku yang ditambah bisa langsung diakses ketika mengunjungi halaman *search*. Tombol *manage book* akan menampilkan deretan buku yang terdaftar yang dapat diedit, *delete*, *update* oleh admin.

```
class AddBookViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository,
) : ViewModel() {

    private val _state = MutableStateFlow(AddBookState())
    val state = _state
        .onStart { getBookCategories() }
        .stateIn(
            scope = viewModelScope,
            started = SharingStarted.WhileSubscribed(),
            initialValue = AddBookState()
        )

    fun onEvent(event: AddBookEvent) {
        when (event) {
            is AddBookEvent.OnBookTitleChange -> _state.update {
                it.copy(bookTitle = event.bookTitle)
            }
            is AddBookEvent.OnImageThumbnailChange -> _state.update {
                it.copy(imageThumbnailUri = event.imageThumbnailUri)
            }
            is AddBookEvent.OnBookPdfChange -> _state.update {
                it.copy(bookPdfUri = event.bookPdfUri)
            }
            is AddBookEvent.OnAuthorChange -> _state.update {
                it.copy(author = event.author)
            }
            is AddBookEvent.OnPublisherChange -> _state.update {
                it.copy(publisher = event.publisher)
            }
            is AddBookEvent.OnDescriptionChange -> _state.update {
                it.copy(description = event.description)
            }
            is AddBookEvent.OnBookCategoryChange -> _state.update {
                it.copy(bookCategory = event.bookCategory)
            }
            is AddBookEvent.OnChapterChange -> _state.update {
                it.copy(chapter = event.chapter)
            }
            is AddBookEvent.OnTotalPagesChange -> _state.update {
                it.copy(totalPages = event.totalPages)
            }
            is AddBookEvent.OnSubmit -> onSubmit(
                bookTitle = event.bookTitle,
                imageThumbnailUri = event.imageThumbnailUri,
                bookPdfUri = event.bookPdfUri,
                author = event.author,
                publisher = event.publisher,
                description = event.description,
                bookCategory = event.bookCategory,
                chapter = event.chapter.toIntOrNull() ?: 1,
                totalPages = event.totalPages,
            )
        }
    }

    private fun getBookCategories() {
        viewModelScope.launch {
            _state.update {
                it.copy(
                    user = courseBookRepository.getUser(),
                    bookCategories = courseBookRepository.getBookCategories()
                )
            }
        }
    }
```

```
    private fun onSubmit(
        bookTitle: String,
        imageThumbnailUri: Uri?,
        bookPdfUri: Uri?,
        author: String,
        publisher: String,
        description: String,
        bookCategory: String,
        chapter: Int,
        totalPages: Int,
    ) {
        _state.update { it.copy(isLoading = true) }
        if (imageThumbnailUri == null || bookPdfUri == null) {
            _state.update {
                it.copy(
                    eventMessage = "image thumbnail and book pdf cannot be empty"
                )
            }
            return
        }
        viewModelScope.launch {
            val book = Book(
                bookTitle = bookTitle,
                author = author,
                publisher = publisher,
                description = description,
                bookCategory = bookCategory,
                chapter = chapter,
                totalPages = totalPages,
            )
            flowOf(courseBookRepository.addBook(bookPdfUri, imageThumbnailUri, book))
                .catch {
                    _state.update {
                        it.copy(
                            eventMessage = it.eventMessage ?: it.toString()
                        )
                    }
                }
                .collect {
                    _state.update {
                        it.copy(
                            eventMessage = "Sukses menambahkan buku",
                            isSuccess = true,
                            isLoading = false
                        )
                    }
                    courseBookRepository.getBooks(refresh = true)
                }
        }
    }
}
```

Gambar 4.1-17 program add book

d) Testing

Pada tahap ini penulis menguji apakah judul buku, pengarang, *publisher*, deskripsi, *chapter*, kategori, pilih gambar, pilih buku pdf, dapat tersimpan di *firebase* ketika menekan tombol tambah buku. Jika tersimpan maka buku yang ditambah bisa langsung diakses di halaman *search*. Kemudian untuk tombol *manage book* bisa digunakan untuk mengedit,*delete*,*update* data *user* yang tersimpan.

e) Deploy

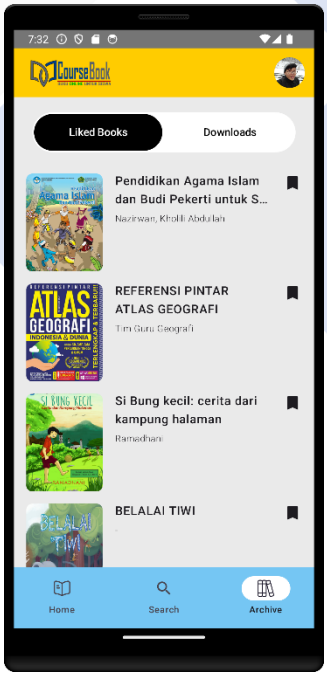
Di tahap *deploy* menunjukan fitur dari *login multilevel* sangat diperlukan untuk pengembangan aplikasi penulis, dimana fitur tersebut dapat membantu admin dalam penginputan data yang praktis tanpa harus membebani admin dalam penginputan data melalui kodingan yang dapat memakan waktu yang lama. Fitur yang digunakan sudah memiliki peforma yang optimal dimana penginputan data akan tersimpan secara langsung dan buku bisa diakses saat input tersimpan.

4.1.8 *SPRINT8*: Archive

a) Plan

Pada tahap perancangan *archive*, file buku menjadi sorotan utama,buku dalam perancangan aplikasi, dikarenakan file tersebut merupakan konten yang akan diakses, jenis tipe file yang akan penulis gunakan adalah pdf. Ketika user memilih sebuah buku, maka buku yang dipilih dapat menampilkan detail dari buku yang diambil. *Archive* berfungsi untuk menandai sebuah buku, agar buku tersebut dapat diakses secara cepat tanpa harus mencari buku tersebut secara manual di halaman *search*. Fitur *download* menjadi kunci dari halaman ini, fitur *download* yang disediakan akan menyimpan buku ke *local storage*, buku yang tersimpan bisa diakses tanpa menggunakan internet (*offline*).

b) Design



Gambar 4.1-18 Tampilan archive

Tampilan antarmuka archive terdiri dari 2 tab yang akan memunculkan data *liked books* dan *download* sesuai yang dipilih. Secara default tampilan yang

dimunculkan ketika membuka halaman *archive* adalah *liked books*, masing masing tab menampilkan buku yang sudah ditandai atau tersimpan ke local storage perangkat.

c) Develop

Di tahap develop ini penulis membuat aplikasi menggunakan bahasa pemrograman Kotlin untuk membuat antarmuka dan *backend program* yang mengkoneksikan inputan data ke *firebase*, tombol *bookmark* tersedia di setiap konten yang tersedia, ketika tombol tersebut ditekan maka akan menambahkan *list liked books* ke akun yang digunakan kemudian koleksi dari *liked book user* akan dibuatkan list di halaman *archive* masing masing. Fitur download akan menyimpan buku ke *local storage* dan membuat list *downloaded book* dan menampilkan list buku tersebut ke halaman *archive*.

```
@HiltViewModel
class ArchiveViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository
) : ViewModel() {

    private val _state = MutableStateFlow(ArchiveState())
    val state = _state
        .onStart { getLikedAndDownloadedBooks() }
        .stateIn(
            viewModelScope,
            SharingStarted.WhileSubscribed(5000),
            ArchiveState()
        )

    fun onEvent(event: ArchiveEvent) {
        when (event) {
            is ArchiveEvent.OnBookmarkChange -> onBookmarkChange(event.book)
            is ArchiveEvent.RefreshBooks -> getLikedAndDownloadedBooks()
        }
    }

    private fun onBookmarkChange(book: Book) {
        viewModelScope.launch {
            try {
                courseBookRepository.deleteLikedBook(book)
            } catch (e: Exception) {
                e.printStackTrace()
            } finally {
                getLikedAndDownloadedBooks()
            }
        }
    }

    private fun getLikedAndDownloadedBooks() {
        viewModelScope.launch {
            _state.update {
                it.copy(
                    user = courseBookRepository.getUser(),
                    likedBooks = courseBookRepository.getLikedBooks(),
                    downloadedBooks = courseBookRepository.getAllPdfFiles().map { file ->
                        Book(
                            bookId = file.bookId,
                            bookTitle = file.bookTitle,
                            imageThumbnailUrl = file.imageThumbnailUrl,
                            bookPdfUrl = file.bookPdfUrl,
                            author = file.author,
                            publisher = file.publisher,
                            description = file.description,
                            bookCategory = file.bookCategory,
                            chapter = file.chapter,
                            totalReadCount = file.totalReadCount,
                            totalPages = file.totalPages,
                            totalDownloadCount = file.totalDownloadCount,
                        )
                    }
                )
            }
        }
    }
}
```

Gambar 4.1-19 program archive

d) Testing

Di tahap ini pengujian dilakukan dengan mengunjungi salah satu konten buku yang disediakan, kemudian menekan tombol *bookmark* yang berada di halaman detail buku tersebut, kemudian memasuki halaman baca buku dan menekan tombol *download*. Testing dianggap berhasil jika tombol berubah warna dan *list liked books* terbuat di halaman *archive*, dan tombol *download* berhasil memunculkan notifikasi *download* ketika ditekan, serta data bisa diakses tanpa internet .

e) Deploy

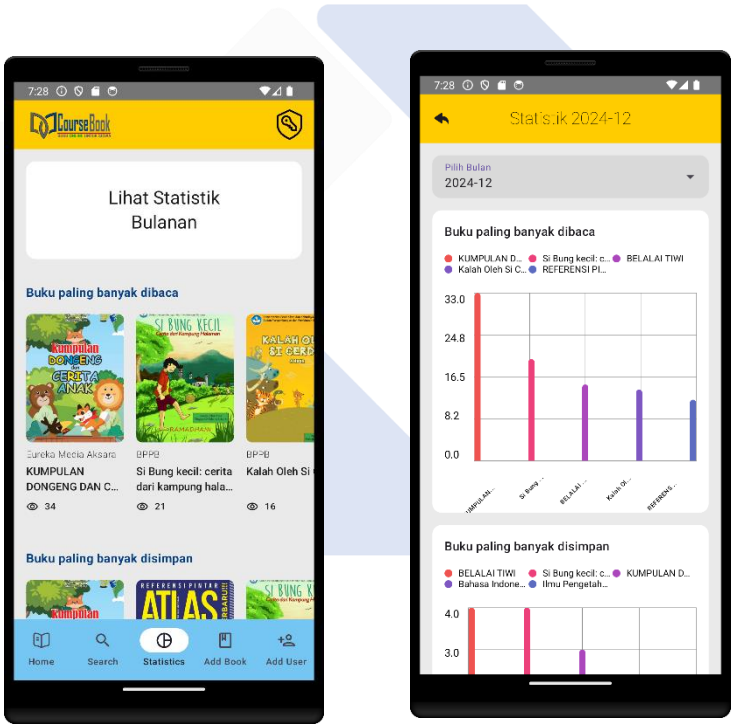
Di tahap ini fitur *bookmark* dan *download* berjalan sesuai dengan yang direncanakan, dalam pembagunan aplikasi ini penulis memastikan bahwa fitur siap untuk digunakan oleh *user*.

4.1.9 *SPRINT9*: Monitoring Statistics Admin

a) *Plan*

Di tahap perancanagan halaman *statistics*, penulis merancang sebuah halaman untuk memonitoring keaktifan user dalam pengaksesan aplikasi. Monitoring ini mencakupi buku paling banyak dibaca, buku paling banyak disimpan, siswa paling banyak membaca, kelas paling banyak membaca serta kategori buku di setiap *genre*. Tujuan utama dari memonitoring ini adalah agar pihak pustakawan dapat melihat *statistic* pengunjungan yang dilakukan oleh *siswa* dan *guru* setiap bulanya.

b) *Design*



Gambar 4.1-20 Tampilan list statistics dan graphic statistics

Tampilan halaman *statistic* yang dibuat berbentuk list statistik yang dimana list konten diurutkan dengan menampilkan statistik dengan gambar dan konten dengan detail, sedangkan list statistik grafik memuat seluruh data statistik menjadi tipe grafik yang simpel dan mudah di mengerti.

c) *Develop*

Di tahap *develop* ini penulis membuat aplikasi menggunakan bahasa pemrograman Kotlin untuk membuat antarmuka dan *backend program* yang mengkoneksikan inputan data ke *firebase*, yang mengurutkan data menjadi bulanan dan kemudian sesuai perhitungan data yang sudah disimpan dijadikan *list statistics* dan grafik *statistics*.

```
    override suspend fun getAllPdfFiles(): List<PdfFile> {
        val username = preferences.getString("username", "") ?: ""
        return dao.getAllFiles(username)
    }

    override suspend fun getAllYearMonths(): List<String> {
        return firebaseService.getAllYearMonths()
    }

    override suspend fun getPdfFromRoom(fileName: String): PdfFile? {
        return dao.getFile(fileName)
    }

    override suspend fun getMostReadBooksThisMonth(month: String): List<Book> {
        return firebaseService.getMostReadBooksThisMonth(month)
    }

    override suspend fun getMostActiveClassesByReadingCountThisMonth(month: String):
List<FirebaseServiceImpl.ClassAnalytics> {
        return firebaseService.getMostActiveClassesByReadingCountThisMonth(month)
    }

    override suspend fun getMostDownloadedBooksThisMonth(month: String): List<Book> {
        return firebaseService.getMostDownloadedBooksThisMonth(month)
    }

    override suspend fun getMostLikedBooksThisMonth(month: String): List<Book> {
        return firebaseService.getMostLikedBooksThisMonth(month)
    }

    override suspend fun getMostStudentReadsThisMonth(month: String): List<User> {
        return firebaseService.getMostStudentReadsThisMonth(month)
    }

    override suspend fun logout() {
        firebaseService.logout()
    }
}
```

Gambar 4.1-21Tampilan list dan graphic statistics

d) Testing

Pada pengujian kali ini penulis menginstruksikan beberapa user untuk membuka buku dan mencoba fitur fitur di aplikasi seperti *bookmark* dan *download*. User yang telah mengakses banyak buku akan muncul di halaman *statistics* admin, dari data tersebut apakah bisa menjadi *list statistics* dan grafik *statistics*

e) Deploy

Di tahap terakhir dari pembuatan halaman statistics ini, fitur yang disediakan dapat menampilkan statistik dalam bentuk list dan juga grafik.

List *statistics* grafik akan mengurutkan data setiap bulannya. Penulis dalam hal ini sudah menyelesaikan tahap monitoring secara optimal dan sesuai fungsi.

4.2 Alpha

Hasil pengujian alpha ini menggunakan metode *whitebox* yang dilakukan oleh orang yang berpengalaman dalam bidang Informasi dan Teknologi(IT).

4.2.1 Pengujian halaman home

Pengujian halaman aplikasi *course book* pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman *home* lolos dari pengujian *alpha*. Berikut tabel halaman akun siswa,guru dan admin:

A. Halaman Akun Siswa dan Guru

Table 4.2-1 Halaman Home Siswa dan Guru

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka aplikasi	Aplikasi di buka dan menampilkan halaman Home	Aplikasi berhasil dibuka	☑	☐
2.	Fungsi tombol home	Kembali menampilkan halaman home	Tampilan home muncul	☑	☐
3.	Fungsi tombol search	Mengarahkan dan menampilkan user ke Halaman Search	Tampilan Search muncul	☑	☐
4.	Fungsi tombol archive	Mengarahkan dan menampilkan user ke Halaman Archive	Tampilan Archive muncul	☑	☐

5.	Membuka halaman baca	Membuka halaman detail buku dan menampilkan deskripsi buku beserta tombol baca.	Tampilan Detail Buku Muncul dan tombol berfungsi	☑	☐
6.	Fungsi tombol detail akun	Membuka halaman detail user dan menampilkan data user beserta tombol log out	Tampilan Detail Akun Muncul dan Tombol Befungsi	☑	☐
7.	Menampilkan buku	Menampilkan Koleksi buku di halaman home.	Koleksi buku Muncul	☑	☐
8.	Akses konten buku	Dapat membaca konten buku yang ditampilkan di laman home	Menampilkan konten buku dan dapat dibaca	☑	☐
9.	Fungsi fitur riwayat membaca	Menampilkan Buku yang sudah pernah dibaca ke halaman home	Tampilan riwayat muncul di halaman home	☑	☐

B. Halaman Admin

Table 4.2-2Halaman home admin

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka aplikasi	Aplikasi di buka dan menampilkan halaman Home	Aplikasi berhasil dibuka	☑	☐
2.	Fungsi tombol home	Kembali menampilkan halaman home	Tampilan home muncul	☑	☐
3.	Fungsi tombol search	Mengarahkan dan menampilkan user ke Halaman Search	Tampilan Search muncul	☑	☐

4.	Fungsi tombol statistik	Mengarahkan dan menampilkan user ke halaman statistil	Tampilan statistik muncul	☑	☐
5.	Fungsi tombol add book	Menampilkan Halaman add book	Tampilan add book muncul	☑	☐
6.	Fungsi tombol add user	Menampilkan Halaman add user	Tampilan add user muncul	☑	☐
7.	Membuka halaman baca	Membuka halaman detail buku dan menampilkan deskripsi buku beserta tombol baca.	Tampilan Detail Buku Muncul dan tombol berfungsi	☑	☐
8.	Fungsi tombol detail akun	Membuka halaman detail user dan menampilkan data user beserta tombol log out	Tampilan Detail Akun Muncul dan Tombol Befungsi	☑	☐
9.	Menampilkan buku	Menampilkan Koleksi buku di halaman home.	Koleksi buku Muncul	☑	☐
10.	Akses konten buku	Dapat membaca konten buku yang ditampilkan di laman home	Menampilkan konten buku dan dapat dibaca	☑	☐
11.	Fungsi fitur riwayat membaca	Menampilkan buku yang sudah pernah dibaca ke halaman home	Tampilan riwayat muncul di halaman home	☑	☐

4.2.2 Pengujian halaman search

Pengujian halaman aplikasi *course book* pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman search lolos dari pengujian alpha.

Table 4.2-3 Halaman Search

NO.	Kesimpulan

	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Diterima	Ditolak
1.	Fungsi Tab Search	Menginput keyword dari keyboard dan menampilkan data sesuai inputan pencarian	Menampilkan data sesuai input	☑	☐
2.	Fungsi tombol genre	Memfilter konten sesuai dengan genre yang di cari	Menampilkan konten sesuai genre	☑	☐
3.	Menampilkan konten Buku	Menampilkan Koleksi buku di halaman search	Tampilan buku muncul	☑	☐
4.	Akses Konten Buku	Dapat membaca konten buku yang di tampilkan di laman search	Buku dapat dibaca	☑	☐

4.2.3 Pengujian halaman archive

Pengujian halaman aplikasi course book pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman archive lolos dari pengujian alpha.

Table 4.2-4 halaman archive

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Fungsi tab Downloads	Menampilkan list buku yang didownload di tabel dan buku yang di download bisa dibaca	Tampilan buku muncul dan buku bisa di baca tanpa loading konten.	☑	☐
2.	Fungsi tab liked book	Menampilkan list buku yang di liked ketika menekan tombol bookmark pada detail buku	Tampilan buku muncul dan tombol berfungsi.	☑	☐

4.2.4 Pengujian halaman baca

Pengujian halaman aplikasi course book pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman baca layak untuk digunakan dan lolos dari pengujian alpha

Table 4.2-5 halaman baca

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka Halaman Detail Buku	Aplikasi dapat menampilkan halaman Detail buku	Tampilan halaman detail buku Muncul	☑	☐
2.	Fungsi Tombol Baca	Membuka tampilan Readview dan menampilkan konten dari buku	Tampilan halaman muncul dan menampilkan konten buku	☑	☐
3.	Fungsi Tombol Change view	Berfungsi menampilkan konten buku menjadi mode scroll atau swipe	Tampilan bisa di scroll dan swipe	☑	☐
4.	Fungsi Tombol Download	Menyimpan data buku ke local storage dan menampilkanya di halaman archive	Data buku tersimpan dan tertampil di halaman	☑	☐

4.2.5 Pengujian halaman statistik

Pengujian halaman aplikasi course book pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman statistik lolos dari pengujian alpha.

Table 4.2-6 halaman statistik

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak

1.	Membuka halaman Statistik	Aplikasi menampilkan halaman Statistik	Tampilan halaman muncul	☑	☐
2.	Fungsi tombol Lihat Statistik Bulanan	Menampilkan grafik statistik berdasarkan data yang tertera di halaman	Tampilan grafik muncul	☑	☐
3.	Menampilkan Statistik	Menampilkan data statistik dengan rincian di halaman	Tampilan statistik muncul	☑	☐

4.2.6 Pengujian halaman add book

Pengujian halaman aplikasi course book pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman add book lolos dari pengujian alpha.

Table 4.2-7 halaman add book

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka Halaman add Book	Menampilkan halaman add book	Tampilan add book muncul	☑	☐
2.	Upload detail buku	Menginput data buku seperti, judul, penulis, penerbit, chapter, kategori, cover dan file buku untuk ditambahkan ke koleksi buku	Data berhasil di input dan diupload	☑	☐
3.	Fungsi Tombol Manage Book	Mengedit dan menghapus data buku dari aplikasi	Data berhasil diubah dan dihapus	☑	☐

4.2.7 Pengujian halaman add user

Pengujian halaman aplikasi course book pada tabel di bawah mendapatkan hasil yang bagus, dimana semua sistem diterima oleh penguji, maka dari itu halaman add user lolos dari pengujian alpha.

Table 4.2-8 halaman add user

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka Halaman Add user	Menampilkan halaman add user	Tampilan Add user muncul	☑	☐
2.	Upload detail user	Menginput data user seperti username,password, nama lengkap, dll untuk ditambahkan ke aplikasi	Data berhasil di input dan diupload	☑	☐
3.	Fungsi Tombol Manage user	Mengedit dan menghapus data dari user aplikasi	Data berhasil diubah dan dihapus	☑	☐

4.3 Beta

Hasil pengujian beta berasal dari kuesioner yang telah dikumpulkan oleh siswa dan guru UPTD SDN 13 Sungailiat yang telah mengenal dan mencoba sistem aplikasi. Di tahap beta ini kueisioner akan dihitung dengan sitem peniliaian *User Acceptance Test* (UAT), hasil perhitungan itu nanti akan menjadi faktor apakah aplikasi *course book* layak dan dibutuhkan di SDN 13 Sungailiat.

4.3.1 Hasil Kuesioner guru SDN 13 Sungailiat

Tabel di bawah menunjukan hasil kuesioner guru SDN 13 Sungailiat :

Table 4.3-1Kuesioner guru

No.	Nama	Pertanyaan						Total
		1	**2**	**3**	**4**	**5**	**6**	
1	Mahyuni, S.Pd	4	4	4	4	4	4	24
2	Hesti Diniarti	4	4	4	4	4	4	24

3	Auliya Rachmawati, S.Pd	4	4	4	4	4	4	24
4	Zuhriya	5	4	4	4	4	4	25
5	Tri Daryanti, S.Pd	4	4	4	4	4	4	24
6	Nuryana, S.Pd	5	4	5	5	4	5	28
7	Sri Muliyati, S.Pd	5	4	4	4	4	4	25
8	Suci Meitonia	4	5	4	4	4	5	26
9	Marini Yati	5	5	5	5	4	5	29
10	Frisca Yunita M, S.Pd	5	5	4	5	4	5	28
		Total						257

Total = 257

Nilai Tertinggi= 5*6*10 = 300

Nilai Akhir = (257/300)* 100% = 85,66%

Dapat diketahui perhitungan kuesioner nilai akhir yang didapatkan dengan sistem peniliaian UAT sebesaar 85,66%, dan dapat disimpulkan bahwa aplikasi course book **sangat layak** untuk diterapkan di UPTD SDN 13 Sungailiat.

4.3.2 Hasil Kuesioner Siswa Kelas 6

Tabel di bawah menunjukan hasil kuesioner siswa kelas 6 SDN 13 Sungailiat :

Table 4.3-2 Kuesioner siswa kelas enam

No.	Nama	Pertanyaan										Total
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
1	Suci Rahmadani	5	5	5	5	4	5	5	5	4	5	48
2	Naura Agustin	5	5	5	5	4	5	5	5	4	5	48
3	Renita	5	5	4	5	3	5	5	5	5	5	47
4	Lira Ana Lia	5	5	5	5	3	5	5	5	4	5	47
5	Aglifan Ramadhan	5	5	5	4	3	5	5	5	4	5	46

6	Nurnaeni	5	5	4	5	2	4	5	5	5	5	45
7	Suci Melinda	5	5	4	3	5	5	5	5	4	5	46
8	Arjun Sahbilui	5	5	5	5	4	5	5	5	5	5	49
9	Rosneni	5	5	5	5	5	5	5	5	5	5	50
10	Andra Pato	5	4	4	5	4	5	5	4	4	5	45
11	Sirutiya Rahayu	5	5	4	3	5	5	5	5	4	5	46
12	M.Deni Ramadhan	5	5	5	5	4	5	5	5	5	5	49
13	Riski Gilang Ramadan	5	5	5	5	4	5	5	5	5	5	49
14	Alvina Embun	5	5	5	5	3	5	5	5	4	5	47
15	Alifa Salsabila	5	5	4	4	2	4	5	5	5	5	44
					Total							706

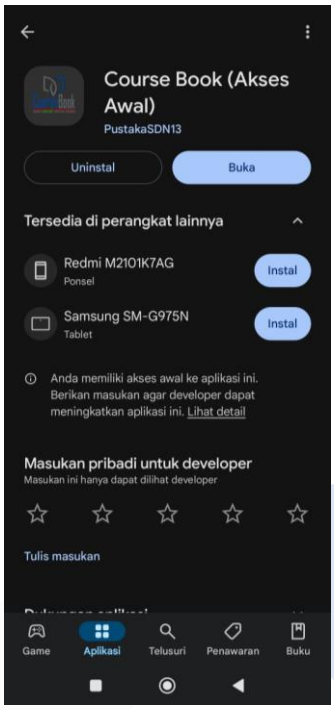
Total = 706

Nilai Tertinggi= 5*10*15 = 750

Nilai Akhir = (706/750)* 100% = 94,13%

Dapat diketahui perhitungan kuesioner nilai akhir yang didapatkan dengan sistem peniliaian UAT sebesaar 94,13%, dan dapat disimpulkan bahwa aplikasi course book **sangat layak** untuk diterapkan di UPTD SDN 13 Sungailiat.

4.4 Rilis



Gambar 4.3-1 Tampilan akses awal playstore

Di tahap ini aplikasi telah selesai diuji dan memasuki tahap penyelesaian, aplikasi diluncurkan dengan *format* apk dan diupload ke google playstore di masa *early access* dengan nama *Course Book*. Instalasi aplikasi dapat dilakukan setelah aplikasi selesai di tahap *early access*, dengan bergabung ke link berikut : https://groups.google.com/g/course-book-tester.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dengan merancang dan membuat aplikasi "Course Book berbasis Android untuk siswa dan guru SDN 13 Sungailiat" maka dapat diambil kesimpullan sebagai berikut:

1. Aplikasi berhasil dirancang dan dibangun menggunakan bahasa pemrograman kotlin melalui *software* android studio yang telah tersambung dengan *firebase console*. Aplikasi berhasil di export menjadi file apk yang siap diinstal di *smartphone* berbasis android.
2. Pengoptimal fitur berhasil diimplementasikan menggunakan bahasa pemrograman kotlin. Dimulai dari tombol yang tersedia befungsi sesuai rancangan yang dibuat, aplikasi dapat menampilkan isi buku untuk dibaca, fitur admin dapat menambahkan, mengedit,menghapus buku atau user melalui aplikasi dan memonitoring statistik dalam mendokumentasikan keaktifan user perbulannya.
3. Hasil dari pengujian alpha menyatakan semua aspek dan fitur pada aplikasi berfungsi dan disetujui oleh pihak penguji sedangkan hasil kuesioner UAT dapat diketahui perhitungan nilai akhir yang didapatkan dengan sistem peniliaian UAT sebesaar 85,66% untuk guru, dan 94,13% untuk siswa, dapat disimpulkan bahwa aplikasi course book mendapatkan rentang kriteria di atas 81% yang menyatakan aplikasi sangat layak untuk diterapkan di UPTD SDN 13 Sungailiat..

5.2 Saran

Masih terdapat kekurangan yang bisa dilengkapi agar aplikasi dapat menjadi lebih bagus dan sempurna.Aplikasi dapat ditambah dengan fitur yang dapat memainkan jenis file *audio book*, yang akan meningkatkan variasi buku yang diakses, Menambahkan animasi agar fitur yang dilihat lebih menarik dan bagus,.*Layout* dari aplikasi bisa dibuat lebih simpel dan *compact*

DAFTAR PUSTAKA

[1] Ngura, E. T., Go, B., & Rewo, J. M. (2020). Pengaruh Media Pembelajaran Buku Cerita Bergambar Terhadap Perkembangan Emosional Anak Usia Dini. Jurnal Ilmiah Pendidikan Citra Bakti, 7(2), 118–124. https://doi.org/10.38048/jipcb.v7i2.94

[2] Sari, M. P., Ahyani, N., & Nurlina, N. (2024). Peranan Perpustakaan dalam Meningkatkan Minat Baca di SD Islam Khalifah Annizam Palembang. AKADEMIK: Jurnal Mahasiswa Humanis, 4(3), 881–891. https://doi.org/10.37481/jmh.v4i3.1000

[3] Eni Dwi Wardihani, Bambang Supriyo, Ilham Sayekti, " Penerapan Sistem Informasi dan Manajemen Perpustakaan Berbasis IoT di SDN Kramas,Tembalang,Kota Semarang" pp. 1-10.

[4] Wijaya, S. H., & Sayekti, R. (2024). Analisis Tingkat Ketergunaan Aplikasi Perpustakaan Universitas Quality dalam Mendukung Tujuan Pembangunan Berkelanjutan (SDGs. Ideas: Jurnal Pendidikan, Sosial, Dan Budaya, 10(3), 569. https://doi.org/10.32884/ideas.v10i3.1862

[5] Wardihani, E. D., Supriyo, B., & Sayekti, I. (2021). Penerapan Sistem Informasi dan Manajemen Perpustakaan Berbasis IoT di SDN Kramas, Tembalang, Kota Semarang. Bhakti Persada, 7(1), 1–10. https://doi.org/10.31940/bp.v7i1.2160

[6] Kurniawan, A. (2020). Pengembangan Aplikasi Mobile Penelusuran Koleksi Perpustakaan Universitas Islam Riau (UIR) Berbasis Android. Repository Uir. Uneversitas Islam Riau, 31. http://repository.uir.ac.id/id/eprint/12473%0Ahttps://repository.uir.ac.id/12473/1/163510108.pdf

[7] (Br Ginting & Simamora, 2022; Kurniawan, 2020; Wardihani et al., 2021; Wijaya & Sayekti, 2024)Br Ginting, Y. F., & Simamora, H. (2022). Penggunaan E-Book Dalam Pembelajaran di Sekolah Dasar. Edu Cendikia: Jurnal Ilmiah Kependidikan, 1(01), 36–39. https://doi.org/10.47709/educendikia.v1i01.1774

[8] Wilantika Ramadhani (2020) “Upaya Pemanfaatan Buku Elektronik Berbasis Aplikasi: Studi Kasus Dinas Perpustakaan Dan Kearsipan Daerah Kabupaten Serdang Bedagai” pp. 1-78 http://repository.uinsu.ac.id/11124/1/Skripsi%20Wilantika%20Ramadhani.pdf

[9] R. Putri, and R. Theis, “EFFECT OF PSYCHOLOGICAL FACTORS IN LEARNING TO LEARNING OUTCOMES ON REAL ANALYSIS,” Prosiding SEMIRATA 2017 Bidang MIPA BKS-PTN-Barat, pp. 534-541, 2017.

[10] R. Setiawan, “SISTEM IDENTIFIKASI SIDIK JARI PADA DINAS KEPOLISIAN WILAYAH SIDOARJO DENGAN KOMBINASI METODE GALTON HENRY DAN TRANSFORMASI FOURIER,” Kurawal-Jurnal Teknologi, Informasi dan Industri, vol. 4, no. 1, pp. 31-42, 2021.

[11] H. T. Saputra, A. Muhaimin, and B. Kurniawan, “Sistem Kontrol Kunci Pintu Rumah Menggunakan Fingerprint Smartphone Android Berbasis Arduino Uno,” Jurnal Ilmu Komputer, vol. 11, no. 1, pp. 5-9, 2022.

[12] M. Arifin, “Sistem Pengamanan Mesin Atm Dengan Menggunakan Pengenalan Sidik Jari Dan Wajah Face Recognition Untuk Meminimalisir Cyberbanking Crime,” OISAA Journal of Indonesia Emas, vol. 5, no. 1, pp. 35-42, 2022.

[13] F.H.Ramadhan, I. E. Maulani, N. F. Zuhriyah et al., “Pengembangan Aplikasi Mobile Learning Berbasis Android untuk Meningkatkan Keterampilan Pemrograman pada Mahasiswa Teknik Informatika,” Jurnal Sosial dan Teknologi, vol. 3, no. 2, pp. 108-113, 2023

[14] (Spatioti et al., 2022)Spatioti, A. G., Kazanidis, I., & Pange, J. (2022). A Comparative Study of the ADDIE Instructional Design Model in Distance Education. Information (Switzerland), 13(9). https://doi.org/10.3390/info13090402

[15] (Wardhanie & Lebdaningrum, 2023)Wardhanie, A., & Lebdaningrum, K. (2023). Pengenalan Aplikasi Desain Grafis Figma pada Siswa-Siswi Multimedia SMK PGRI 2 Sidoarjo. Yumary: Jurnal Pengabdian Kepada Masyarakat, 3(3), 165–174. https://doi.org/10.35912/yumary.v3i3.1536

[16] (Ergash et al., 2023; Khawas & Shah, 2018; Wardhanie & Lebdaningrum, 2023)Ergash, Q. F. o'g'li, G'ulodimin, S. N. o'g'li, & Ergash, N. S. qizi. (2023). Features Of The Android Studio Software Package. May 2017, 130–146. https://doi.org/10.5281/zenodo.8050543

[17] (Anjasmara & Hadi Sumitro, 2023)Anjasmara, V. M., & Hadi Sumitro, A. (2023). Pengembangan Sistem Informasi Masjid Darul Arham Menggunakan Metode V-Model dan UAT (User Acceptance Testing). INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS : Journal of Information System, 8(1), 47. https://doi.org/10.51211/isbi.v8i1.2443

[18] Khawas, C., & Shah, P. (2018). Application of Firebase in Android App Development-A Study. International Journal of Computer Applications, 179(46), 49–53. https://doi.org/10.5120/ijca2018917200

[19] Wardhanie, A., & Lebdaningrum, K. (2023). Pengenalan Aplikasi Desain Grafis Figma pada Siswa-Siswi Multimedia SMK PGRI 2 Sidoarjo. Yumary: Jurnal Pengabdian Kepada Masyarakat, 3(3), 165–174. https://doi.org/10.35912/yumary.v3i3.1536

[20] (Секция Принттехнологий и Медиакоммуникаций, n.d.)Секция принттехнологий и медиакоммуникаций.(Bagian Teknologi Cetak dan Media Komunikasi, n.d.)Bagian Teknologi Cetak dan Media Komunikasi. (2017). 394–396.

[21] R. P. Dewi, "Studi Kasus-Metode Penelitian Kualitatif," 2019

[22] (Edrina Christine et al., 2024)Edrina Christine, N., Priyatna, B., Lia Hananto, A., & Shofiah Hilabi, S. (2024). Penerapan Metode Agile Scrum Pada Sistem E-Posyandu Berbasis Web. JATI (Jurnal Mahasiswa Teknik Informatika), 8(2), 2013–2019. https://doi.org/10.36040/jati.v8i2.9257

[23] (Alfiah & Indriani, 2021)Alfiah, N., & Indriani, H. (2021). Sistem Informasi Perpustakaan Berbasis website. *Jurnal Teknologi Dan Bisnis*, *3*(2), 188–198. https://doi.org/10.37087/jtb.v3i2.65

LAMPIRAN

LAMPIRAN 1: DAFTAR RIWAYAT HIDUP PENULIS

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap	:	Fitra Dwianto
Tempat & Tanggal Lahir	:	Sungailiat, 07 Desember 2002
Alamat Rumah	:	Jalan amarta no.5 RT07 , Sungailiat.
		Hp: 081273949731
		Email: fitradwianto07@gmail.com
Jenis Kelamin	:	Laki-Laki
Agama	:	Islam

2. Riwayat Pendidikan

SDN 3 SUNGAILIAT	:	2008 – 2014
SMPN 5 SUNGAILIAT	:	2014 – 2017
SMK YAPENSU SUNGAILIAT	:	2017 – 2020

Sungailiat,10 Januari 2025

Fitra Dwianto

LAMPIRAN 2: KODE PROGRAM

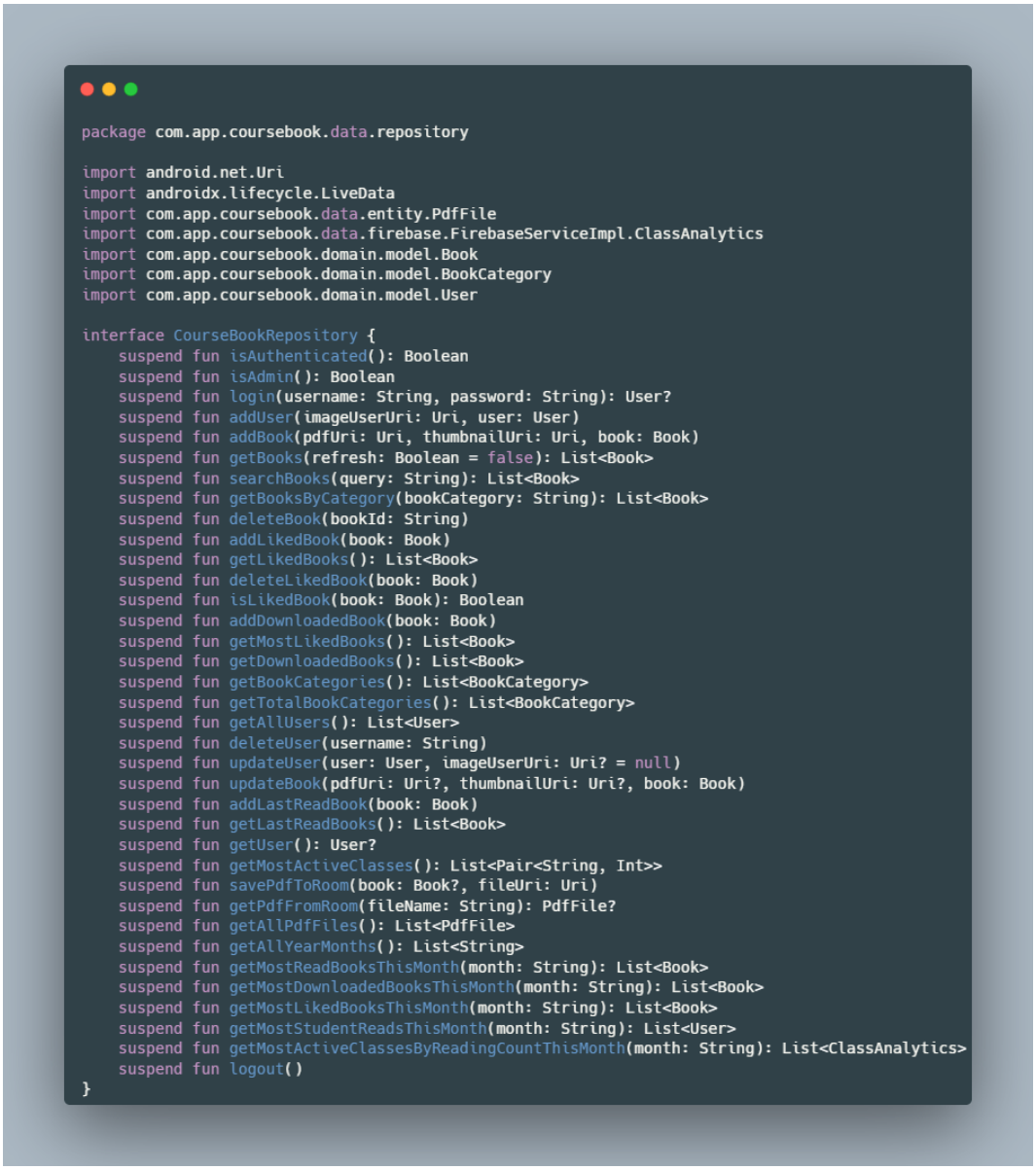
FireBaseService.kt

```kotlin
package com.app.coursebook.data.firebase

import android.net.Uri
import com.app.coursebook.data.firebase.FirebaseServiceImpl.ClassAnalytics
import com.app.coursebook.domain.model.Book
import com.app.coursebook.domain.model.BookCategory
import com.app.coursebook.domain.model.User

interface FirebaseService {
    suspend fun login(username: String, password: String): User?
    suspend fun isAdmin(): Boolean
    suspend fun isAuthenticated(): Boolean
    suspend fun addUser(imageUserUri: Uri, user: User)
    suspend fun getUser(): User?
    suspend fun deleteUser(username: String)
    suspend fun getAllUsers(): List<User>
    suspend fun updateUser(user: User, imageUserUri: Uri?)
    suspend fun addBook(pdfUri: Uri, thumbnailUri: Uri, book: Book)
    suspend fun getBooks(): List<Book>
    suspend fun updateBook(pdfUri: Uri?, thumbnailUri: Uri?, book: Book)
    suspend fun deleteBook(bookId: String)
    suspend fun addLikedBook(book: Book)
    suspend fun deleteLikedBook(book: Book)
    suspend fun getLikedBooks(): List<Book>
    suspend fun getMostLikedBooks(): List<Book>
    suspend fun isLikedBook(book: Book): Boolean
    suspend fun addDownloadedBook(book: Book)
    suspend fun getDownloadedBooks(): List<Book>
    suspend fun addLastReadBook(book: Book)
    suspend fun getLastReadBooks(): List<Book>
    suspend fun getBookCategories(): List<BookCategory>
    suspend fun getTotalBookCategories(): List<BookCategory>
    suspend fun getMostActiveClasses(): List<Pair<String, Int>>
    suspend fun getAllYearMonths(): List<String>
    suspend fun getMostReadBooksThisMonth(month: String): List<Book>
    suspend fun getMostDownloadedBooksThisMonth(month: String): List<Book>
    suspend fun getMostLikedBooksThisMonth(month: String): List<Book>
    suspend fun getMostStudentReadsThisMonth(month: String): List<User>
    suspend fun getMostActiveClassesByReadingCountThisMonth(month: String): List<ClassAnalytics>
    suspend fun logout()
}l totalPages: Int = 1,
    val totalReadCount: Int = 0,
    val totalDownloadCount: Int = 0,
    val username: String = "",
)
```

CourseBookRepository.kt



```kotlin
package com.app.coursebook.data.repository

import android.net.Uri
import androidx.lifecycle.LiveData
import com.app.coursebook.data.entity.PdfFile
import com.app.coursebook.data.firebase.FirebaseServiceImpl.ClassAnalytics
import com.app.coursebook.domain.model.Book
import com.app.coursebook.domain.model.BookCategory
import com.app.coursebook.domain.model.User

interface CourseBookRepository {
    suspend fun isAuthenticated(): Boolean
    suspend fun isAdmin(): Boolean
    suspend fun login(username: String, password: String): User?
    suspend fun addUser(imageUserUri: Uri, user: User)
    suspend fun addBook(pdfUri: Uri, thumbnailUri: Uri, book: Book)
    suspend fun getBooks(refresh: Boolean = false): List<Book>
    suspend fun searchBooks(query: String): List<Book>
    suspend fun getBooksByCategory(bookCategory: String): List<Book>
    suspend fun deleteBook(bookId: String)
    suspend fun addLikedBook(book: Book)
    suspend fun getLikedBooks(): List<Book>
    suspend fun deleteLikedBook(book: Book)
    suspend fun isLikedBook(book: Book): Boolean
    suspend fun addDownloadedBook(book: Book)
    suspend fun getMostLikedBooks(): List<Book>
    suspend fun getDownloadedBooks(): List<Book>
    suspend fun getBookCategories(): List<BookCategory>
    suspend fun getTotalBookCategories(): List<BookCategory>
    suspend fun getAllUsers(): List<User>
    suspend fun deleteUser(username: String)
    suspend fun updateUser(user: User, imageUserUri: Uri? = null)
    suspend fun updateBook(pdfUri: Uri?, thumbnailUri: Uri?, book: Book)
    suspend fun addLastReadBook(book: Book)
    suspend fun getLastReadBooks(): List<Book>
    suspend fun getUser(): User?
    suspend fun getMostActiveClasses(): List<Pair<String, Int>>
    suspend fun savePdfToRoom(book: Book?, fileUri: Uri)
    suspend fun getPdfFromRoom(fileName: String): PdfFile?
    suspend fun getAllPdfFiles(): List<PdfFile>
    suspend fun getAllYearMonths(): List<String>
    suspend fun getMostReadBooksThisMonth(month: String): List<Book>
    suspend fun getMostDownloadedBooksThisMonth(month: String): List<Book>
    suspend fun getMostLikedBooksThisMonth(month: String): List<Book>
    suspend fun getMostStudentReadsThisMonth(month: String): List<User>
    suspend fun getMostActiveClassesByReadingCountThisMonth(month: String): List<ClassAnalytics>
    suspend fun logout()
}
```

BookReaderViewModel.kt

```kotlin
package com.app.coursebook.ui.screen.book_reader

import android.net.Uri
import android.util.Log
import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.app.coursebook.data.repository.CourseBookRepository
import com.app.coursebook.domain.model.Book
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.SharingStarted
import kotlinx.coroutines.flow.onStart
import kotlinx.coroutines.flow.stateIn
import kotlinx.coroutines.flow.update
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class BookReaderViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository,
) : ViewModel() {

    private val _state = MutableStateFlow(BookReaderState())
    val state = _state
        .onStart { getUser() }
        .stateIn(
            viewModelScope,
            SharingStarted.WhileSubscribed(5000L),
            BookReaderState()
        )

    fun onEvent(event: BookReaderEvent) {
        when (event) {
            is BookReaderEvent.OnPdfDownload -> {
                downloadPdf(event.book, event.fileUri)
            }
        }
    }

    private fun getUser() {
        viewModelScope.launch {
            _state.update {
                it.copy(
                    user = courseBookRepository.getUser()
                )
            }
        }
    }

    private fun downloadPdf(book: Book?, fileUri: Uri) {
        viewModelScope.launch {
            if (book != null) {
                try {
                    courseBookRepository.savePdfToRoom(book, fileUri)
                    courseBookRepository.addDownloadedBook(book)
                } catch (e: Exception) {
                    Log.d("DOWNLOAD", "downloadPdf: $e")
                    e.printStackTrace()
                }
            }
        }
    }
}
```

HomeViewModel.kt

```kotlin
package com.app.coursebook.ui.screen.home

import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.app.coursebook.data.repository.CourseBookRepository
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.SharingStarted
import kotlinx.coroutines.flow.onStart
import kotlinx.coroutines.flow.stateIn
import kotlinx.coroutines.flow.update
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class HomeViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository
) : ViewModel() {

    private val _state = MutableStateFlow(HomeState())
    val state = _state
        .onStart { getLastReadBooks() }
        .stateIn(
            viewModelScope,
            SharingStarted.WhileSubscribed(5000),
            HomeState(),
        )

    fun refresh() {
        _state.update { it.copy(isRefreshing = true) }
        getLastReadBooks()
    }

    private fun getLastReadBooks() {
        viewModelScope.launch {
            try {
                courseBookRepository.getBooks(refresh = true)
                val user = courseBookRepository.getUser()
                val lastReadBooks = courseBookRepository.getLastReadBooks()
                val educationBooks = courseBookRepository.getBooksByCategory("Edukasi")
                val fictionBooks = courseBookRepository.getBooksByCategory("Sastra & Fiksi")

                _state.update {
                    it.copy(
                        user = user,
                        lastReadBooks = lastReadBooks,
                        educationBooks = educationBooks,
                        fictionBooks = fictionBooks,
                        isRefreshing = false
                    )
                }
            } catch (e: Exception) {
                e.printStackTrace()
            }
        }
    }
}
```

LogimViewModel.kt

```kotlin
package com.app.coursebook.ui.screen.login

import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.app.coursebook.data.repository.CourseBookRepository
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.SharingStarted
import kotlinx.coroutines.flow.onStart
import kotlinx.coroutines.flow.stateIn
import kotlinx.coroutines.flow.update
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class LoginViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository
) : ViewModel() {

    private val _state = MutableStateFlow(LoginState())
    val state = _state
        .onStart { checkAuthentication() }
        .stateIn(
            viewModelScope,
            SharingStarted.WhileSubscribed(),
            LoginState()
        )

    fun onEvent(event: LoginEvent) {
        when (event) {
            is LoginEvent.OnUsernameChange -> {
                _state.update {
                    it.copy(
                        username = event.username
                    )
                }
            }
            is LoginEvent.OnLogin -> doLogin(event.username, event.password)
            is LoginEvent.OnPasswordChange -> {
                _state.update {
                    it.copy(
                        password = event.password
                    )
                }
            }
        }
    }

    private fun checkAuthentication() {
        viewModelScope.launch {
            _state.update {
                it.copy(
                    isAuthenticated = courseBookRepository.isAuthenticated()
                )
            }
        }
    }

    private fun doLogin(username: String, password: String) {
        viewModelScope.launch {
            _state.update { it.copy(isLoading = true, eventMessage = null) }
            try {
                val user = courseBookRepository.login(username, password)
                _state.update {
                    it.copy(
                        isLoading = false,
                        isAuthenticated = user != null,
                        eventMessage = if (user == null) "User not found" else "Login Sukses"
                    )
                }
            } catch (e: Exception) {
                _state.update {
                    it.copy(
                        isLoading = false,
                        eventMessage = e.message
                    )
                }
            }
        }
    }
}
```

StatisticViewModel.kt

```kotlin
package com.app.coursebook.ui.screen.statistics

import androidx.lifecycle.ViewModel
import androidx.lifecycle.viewModelScope
import com.app.coursebook.data.repository.CourseBookRepository
import dagger.hilt.android.lifecycle.HiltViewModel
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.SharingStarted
import kotlinx.coroutines.flow.onStart
import kotlinx.coroutines.flow.stateIn
import kotlinx.coroutines.flow.update
import kotlinx.coroutines.launch
import javax.inject.Inject

@HiltViewModel
class StatisticsViewModel @Inject constructor(
    private val courseBookRepository: CourseBookRepository,
) : ViewModel() {

    private val _state = MutableStateFlow(StatisticsState())
    val state = _state
        .onStart { getStatistics() }
        .stateIn(
            scope = viewModelScope,
            started = SharingStarted.WhileSubscribed(5_000),
            initialValue = StatisticsState()
        )

    private fun getStatistics() {
        viewModelScope.launch {
            try {
                val books = courseBookRepository.getBooks(refresh = true)
                val users = courseBookRepository.getAllUsers()
                val user = courseBookRepository.getUser()

                _state.update { currentState ->
                    currentState.copy(
                        user = user,
                        mostDownloadedBooks = books.sortedByDescending { it.totalDownloadCount },
                        mostReadBooks = books.sortedByDescending { it.totalReadCount },
                        mostLikedBooks = courseBookRepository.getMostLikedBooks(),
                        mostActiveReadUser = users.sortedByDescending { it.lastReadBooks.size },
                        mostActiveClasses = courseBookRepository.getMostActiveClasses(),
                        bookCategories = courseBookRepository.getTotalBookCategories()
                    )
                }
            } catch (e: Exception) {
                e.printStackTrace()
            }
        }
    }
}
```

LAMPIRAN 3: Dokumentasi

Nama Penguji : JAKA MARTADIPURA, S.Kom

Tanggal Pengujian : 18-12-2024

A. Halaman Akun Siswa dan Guru

Tabel 4.1 Halaman Home

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka aplikasi	Aplikasi di buka dan menampilkan halaman Home	Aplikasi berhasil dibuka	✓	
2.	Fungsi tombol home	Kembali menampilkan halaman home	Tampilan home muncul	✓	
3.	Fungsi tombol search	Mengarahkan dan menampilkan user ke Halaman Search	Tampilan Search muncul	✓	
4.	Fungsi tombol archive	Mengarahkan dan menampilkan user ke Halaman Archive	Tampilan Archive muncul	✓	
5.	Membuka halaman baca	Membuka halaman detail buku dan menampilkan deskripsi buku beserta tombol baca.	Tampilan Detail Buku Muncul dan tombol berfungsi	✓	
6.	Fungsi tombol detail akun	Membuka halaman detail user dan menampilkan data user beserta tombol log out	Tampilan Detail Akun Muncul dan Tombol Befungsi	✓	
7.	Menampilkan buku	Menampilkan Koleksi buku di halaman home.	Koleksi buku Muncul	✓	
8.	Akses konten buku	Dapat membaca konten buku yang ditampilkan di laman home	Menampilkan konten buku dan dapat dibaca	✓	
9.	Fungsi fitur riwayat membaca	Menampilkan Buku yang sudah pernah dibaca ke halaman home	Tampilan riwayat muncul di halaman home	✓	

Tabel 4.2 Halaman Search

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Fungsi Tab Search	Menginput keyword dari keyboard dan menampilkan data sesuai inputan pencarian	Menampilkan data sesuai input	✓	
2.	Fungsi tombol genre	Memfilter konten sesuai dengan genre yang di cari	Menampilkan konten sesuai genre	✓	
3.	Menampilkan konten Buku	Menampilkan Koleksi buku di halaman search	Tampilan buku muncul	✓	
4.	Akses Konten Buku	Dapat membaca konten buku yang di tampilkan di laman search	Buku dapat dibaca	✓	

Tabel 4.3 Halaman Archive

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Fungsi tab Downloads	Menampilkan list buku yang didownload di tabel dan buku yang di download bisa dibaca	Tampilan buku muncul dan buku bisa di baca tanpa loading konten.	✓	
2.	Fungsi tab liked book	Menampilkan list buku yang di liked ketika menekan tombol bookmark pada detail buku	Tampilan buku muncul dan tombol berfungsi.	✓	

Tabel 4.4 Halaman Baca

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka Halaman Detail Buku	Aplikasi dapat menampilkan halaman Detail buku	Tampilan halaman detail buku Muncul	✓	
2.	Fungsi Tombol Baca	Membuka tampilan Readview dan menampilkan konten dari buku	Tampilan halaman muncul dan menampilkan konten buku	✓	
3.	Fungsi Tombol Change view	Berfungsi menampilkan konten buku menjadi mode scroll atau swipe	Tampilan bisa di scroll dan swipe	✓	
4.	Fungsi Tombol Download	Menyimpan Data Buku ke local storage dan menampilkanya di halaman archive	Data buku tersimpan dan tertampil di halaman	✓	

B. Halaman Admin

Tabel 4.5 Halaman Home Admin

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka aplikasi	Aplikasi di buka dan menampilkan halaman Home	Aplikasi berhasil dibuka	✓	
2.	Fungsi tombol home	Kembali menampilkan halaman home	Tampilan home muncul	✓	
3.	Fungsi tombol search	Mengarahkan dan menampilkan user ke Halaman Search	Tampilan Search muncul	✓	
4.	Fungsi tombol statistik	Mengarahkan dan menampilkan user ke halaman statistil	Tampilan statistik muncul	✓	
5.	Fungsi tombol add book	Menampilkan Halaman add book	Tampilan add book muncul	✓	
6.	Fungsi tombol add user	Menampilkan Halaman add user	Tampilan add user muncul	✓	
7.	Membuka halaman baca	Membuka halaman detail buku dan menampilkan deskripsi buku beserta tombol baca.	Tampilan Detail Buku Muncul dan tombol berfungsi	✓	
8.	Fungsi tombol detail akun	Membuka halaman detail user dan menampilkan data user beserta tombol log out	Tampilan Detail Akun Muncul dan Tombol Befungsi	✓	
9.	Menampilkan buku	Menampilkan Koleksi buku di halaman home.	Koleksi buku Muncul	✓	
10.	Akses konten buku	Dapat membaca konten buku yang ditampilkan di laman home	Menampilkan konten buku dan dapat dibaca	✓	
11.	Fungsi fitur riwayat membaca	Menampilkan buku yang sudah pernah dibaca ke halaman home	Tampilan riwayat muncul di halaman home	✓	

Tabel 4.2 Halaman Search Admin

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Fungsi Tab Search	Menginput keyword dari keyboard dan menampilkan data sesuai inputan pencarian	Menampilkan data sesuai input	✓	
2.	Fungsi tombol genre	Memfilter konten sesuai dengan genre yang di cari	Menampilkan konten sesuai genre	✓	
3.	Menampilkan konten Buku	Menampilkan Koleksi buku di halaman search	Tampilan buku muncul	✓	
4.	Akses Konten Buku	Dapat membaca konten buku yang di tampilkan di laman search	Buku dapat dibaca	✓	

Tabel 4.7 Halaman Statistik Admin

NO.	Aktifitas yang diuji	Hasil yang dinginkan	Hasil yang Diuji	Kesimpulan	
				Diterima	Ditolak
1.	Membuka halaman Statistik	Aplikasi menampilkan halaman Statistik	Tampilan halaman muncul	✓	
2.	Fungsi tombol Lihat Statistik Bulanan	Menampilkan grafik statistik berdasarkan data yang tertera di halaman	Tampilan grafik muncul	✓	
3.	Menampilkan Statistik	Menampilkan data statistik dengan rincian di halaman	Tampilan statistik muncul	✓	

Kuesioner Guru

Nama lengkap : Hesti Diniarti

Tanggal Kuisoner : 18 desember 2024

NO.	Pertanyaan	Skala Penelitian				
		SS	S	KS	TS	STS
1.	Apakah tampilan keseluruhan aplikasi ini menarik?		✓			
2.	Apakah informasi dan isi buku sesuai dengan yang di tampilkan dalam aplikasi?		✓			
3.	Apakah Aplikasi berfungsi dengan baik di perangkat anda?		✓			
4.	Apakah anda merasa aplikasi ini dapat membantu dalam proses pengajaran anda?		✓			
5.	Apakah koleksi buku di aplikasi cukup untuk dijadikan panduan pembelajaran?		✓			
6.	Apakah anda merasa aplikasi ini dapat membantu memberikan dampak positif untuk kedepannya?		✓			

Keterangan :

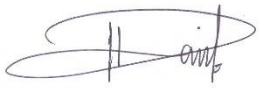
SS = Sangat Setuju

S = Setuju

KS = Kurang Setuju

TS = Tidak Setuju

STS = Sangat Tidak Setuju



HESTI DINIARTI, S.Pd

Tertanda Tangan