

BAB 2

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Aplikasi

Aplikasi adalah sebuah program yang menjalankan proses untuk memenuhi permintaan pengguna dalam mencapai tujuan tertentu (Anendya, 2023). Sementara itu, dalam Kamus Besar Bahasa Indonesia, aplikasi didefinisikan sebagai penerapan rancangan sistem untuk pengolahan data dengan ketentuan atau aturan bahasa pemrograman tertentu. Saat ini, jutaan aplikasi tersedia di *App Store* dan *Google Play Store*, masing-masing menawarkan layanan yang bervariasi. Sejak peluncuran iPhone pada tahun 2007 dan *App Store* pada tahun 2008 (Stephanie & Nistanto, 2023), aplikasi telah menjadi dasar ekonomi digital, serta media utama bagi pengguna dalam memasuki era *revolusi smartphone*.

2.1.2 Keuangan

Keuangan adalah ilmu yang mempelajari tentang cara mengelola, memperoleh, dan menginvestasikan uang baik oleh individu, perusahaan, maupun pemerintah. Bidang ini mencakup kegiatan perencanaan, pengalokasian, dan pengendalian sumber daya keuangan untuk mencapai tujuan yang diinginkan (Wadiyo, 2024). Sektor keuangan memainkan peran krusial dalam mendukung pertumbuhan ekonomi dan meningkatkan kesejahteraan masyarakat, yang diwujudkan melalui berbagai lembaga seperti perbankan, dan pasar modal (Hanggoro, 2024).

2.1.3 Kotlin

Kotlin merupakan bahasa pemrograman berbasis Java Virtual Machine (JVM) yang mendukung konsep object oriented dan functional programming. Kotlin juga memiliki interoperabilitas dengan Java sehingga dapat digunakan dalam satu proyek yang sama. Selain untuk pengembangan aplikasi Android, Kotlin dapat dimanfaatkan dalam pengembangan aplikasi desktop, web, dan backend. Beberapa keunggulan Kotlin antara lain mampu mengurangi terjadinya `NullPointerException`, penulisan kode yang lebih ringkas dan mudah dibaca,

mudah dipelajari, serta didukung oleh IDE yang mempermudah proses pengembangan aplikasi (Sibarani dkk., 2018).

2.1.4 *Android Studio*

Android Studio adalah *Integrated Development Environment* (IDE) resmi untuk membuat aplikasi Android yang dibangun dengan dasar IntelliJ IDEA (Wibowo, 2019). Setiap tahun, Android terus berinovasi dan berkembang menjadi platform yang semakin maju, didorong oleh Google sebagai pengembang utama setelah akuisisi Android. Google menciptakan platform khusus ini agar lebih fleksibel dan memungkinkan perkembangan ekosistem Android secara terbuka.

Platform Android mencakup sistem operasi berbasis Linux, GUI (*Graphic User Interface*), web browser, serta berbagai aplikasi pengguna yang bisa diunduh. Selain itu, platform ini memungkinkan pengembang bebas berkreasi, menciptakan aplikasi berkualitas, dan kompatibel dengan berbagai perangkat.

2.1.5 *Visual Studio Code*

Visual Studio Code (VS Code) merupakan editor kode sumber yang tersedia secara gratis dan dikembangkan oleh Microsoft. Editor ini mendukung berbagai bahasa pemrograman serta menawarkan fitur-fitur unggulan seperti debugging, integrasi kontrol versi Git, penyorotan sintaksis, serta ekstensi yang dapat dikonfigurasi sesuai kebutuhan pengguna. Berkat fleksibilitasnya, dukungan komunitas yang luas, dan kompatibilitas dengan sistem operasi seperti Windows, Linux, serta mac OS, VS Code menjadi pilihan utama bagi banyak pengembang dalam menulis dan mengelola kode secara efisien (Patria, 2023).

2.1.6 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah teknik visualisasi yang digunakan untuk memodelkan dan perancangan sistem berorientasi objek. UML juga didefinisikan sebagai bahasa baku untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak, sering kali disebut cetak biru pengembangan aplikasi (Intern, 2021). Penggunaan UML diharapkan dapat menyederhanakan proses pengembangan perangkat lunak (RPL) sekaligus memenuhi kebutuhan pengguna dengan tepat, termasuk dari segi keamanan,

skalabilitas, dan ketahanan. Jenis diagram UML:

1. *Use Case Diagram*

Diagram ini menggambarkan interaksi antara sistem dan pengguna atau aktor. Use Case Diagram menjelaskan jenis interaksi yang terjadi antara pengguna dengan sistem, termasuk aksi yang dilakukan aktor serta respons dari sistem (Intern, 2021).

2. *Class Diagram*

Diagram ini menunjukkan struktur dan hubungan antar kelas, atribut, dan metode dalam sistem. Diagram ini bersifat statis, sehingga lebih fokus pada relasi antar kelas daripada perilaku dinamis (Setiawan, 2021). *Class Diagram* sangat relevan untuk proyek yang menggunakan pendekatan berorientasi objek, karena memberikan gambaran yang mudah diimplementasikan.

2.1.7 *Laravel*

Laravel merupakan framework PHP bersifat *open-source* yang mengadopsi arsitektur *Model-View-Controller* (MVC) guna menyederhanakan proses pembuatan aplikasi web. *Framework* ini menghadirkan berbagai fitur unggulan, seperti *Eloquent ORM* untuk mengelola database secara efisien, *Blade Templating Engine* untuk menyusun tampilan dinamis, serta sistem *routing* dan *middleware* yang fleksibel untuk mengontrol alur kerja aplikasi (Novia, 2024).

2.1.8 *Metode Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah metode dalam proses pengembangan aplikasi yang dirancang untuk menghasilkan produk secara lebih cepat dengan tetap menjaga kualitasnya. Pendekatan ini menjadi solusi atas keterbatasan model tradisional seperti waterfall, dengan menggabungkan berbagai teknik, seperti *Joint Application Development* (JAD), *prototyping*, dan teknik terstruktur guna mempercepat proses pembangunan sistem. Dengan percepatan ini, organisasi dapat menekan biaya pengembangan serta pemeliharaan perangkat lunak (Delima et al., 2017).

Sementara itu, menurut Kendall & Kendall (2010), RAD merupakan pendekatan berbasis pengembangan sistem berorientasi objek yang

menggabungkan metode dan perangkat lunak tertentu. Tujuan utamanya adalah untuk mempercepat keseluruhan siklus hidup dan implementasi aplikasi, terutama dalam merespons kebutuhan bisnis yang terus berubah dengan cepat. Kecepatan dalam merilis produk atau layanan baru bisa menjadi keunggulan kompetitif, dan RAD memungkinkan pengembang untuk segera meluncurkan aplikasi berbasis, memberikan bantuan bagi mahasiswa dan mengatur keuangan yang sehat.

Proses penting dalam pengembangan aplikasi seperti evaluasi, perancangan, dan penerapan, menjadi bagian tak terpisahkan dari RAD. Metodologi ini dibagi menjadi tiga tahap utama yang melibatkan para pemangku kepentingan dalam proses analisis kebutuhan dan pemanfaatan aplikasi.



Gambar 2.1 Metode Rapid Application Development

Sesuai dengan metodologi RAD menurut (Kendall & Kendall, 2010), berikut adalah tahapan pengembangan aplikasi dari tiap-tiap fase:

1. *Requirements Planning (Perencanaan syarat-syarat)*

Pada tahap ini, partisipasi aktif antara pengguna dan pengembang sistem memiliki peran krusial dalam metode RAD. Kedua pihak berkolaborasi untuk mengidentifikasi tujuan sistem serta merumuskan kebutuhan yang diperlukan guna mencapai tujuan bersama.

2. *RAD Design Workshop (Workshop Desain RAD)*

Diagram ini menunjukkan struktur dan hubungan antar kelas, atribut, dan metode dalam sistem. Diagram ini bersifat statis, sehingga lebih fokus pada relasi antar kelas daripada perilaku dinamis (Setiawan, 2021). Class Diagram sangat relevan untuk proyek yang menggunakan pendekatan berorientasi

objek, karena memberikan gambaran yang mudah diimplementasikan.

3. Implementation (Implementasi)

Pada tahap ini, pengembangan sistem mulai dilakukan berdasarkan desain dan prototipe yang telah disepakati sebelumnya. Prosesnya mencakup penulisan kode program, pengembangan fitur utama, serta pembuatan antarmuka pengguna yang interaktif. Selain pengkodean, pengembang juga melakukan serangkaian pengujian, seperti unit testing (pengujian setiap komponen kecil dalam sistem), integration testing (pengujian bagaimana komponen bekerja bersama), dan system testing (pengujian keseluruhan sistem untuk memastikan bahwa sistem berjalan sesuai dengan yang diharapkan).

2.1.9 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan proses penilaian terhadap suatu aplikasi atau sistem untuk memastikan bahwa perangkat lunak tersebut berfungsi sesuai dengan kebutuhan pengguna serta terbebas dari bug atau kesalahan. Tujuan utama dari pengujian ini adalah meningkatkan kualitas perangkat lunak dengan mengidentifikasi serta memperbaiki masalah selama tahap pengembangan sebelum digunakan secara luas. Pengujian yang efektif dapat meningkatkan pengalaman pengguna serta meminimalkan risiko kegagalan dalam penggunaan nyata. Terdapat dua jenis utama pengujian perangkat lunak:

1. White-box

White-box adalah pengujian ini berfokus pada struktur internal kode program, termasuk alur logika, jalur eksekusi, dan fungsi tertentu. Metode ini bertujuan untuk memastikan bahwa setiap bagian kode dapat berjalan dengan benar dan sesuai dengan spesifikasi yang ditentukan (Zidniryi, 2022).

2. Black-box

Black-box adalah pengujian ini mengevaluasi fungsionalitas perangkat lunak tanpa memperhatikan detail kode yang digunakan. Penguji hanya memverifikasi apakah input yang diberikan menghasilkan output yang diharapkan tanpa mengetahui cara kerja sistem di dalamnya (Zidniryi, 2022). Teknik yang digunakan pengujian dilakukan dengan memberikan input

tertentu dan memeriksa output yang dihasilkan. Contohnya, menguji apakah fitur pencatatan transaksi dapat menyimpan data dengan benar dan menampilkannya sesuai dengan yang diharapkan.

2.1.10 Usability Testing

Usability Testing adalah suatu metode evaluasi dalam pengembangan perangkat lunak yang bertujuan untuk mengukur sejauh mana suatu aplikasi atau sistem dapat digunakan dengan mudah, efisien, dan efektif oleh pengguna. Pengujian ini dilakukan dengan melibatkan pengguna nyata yang diminta untuk menyelesaikan serangkaian tugas dalam sistem, sehingga dapat diidentifikasi hambatan atau kesulitan yang mereka alami selama interaksi (Supriyono & Ardillah, 2023).

Menurut penelitian yang dilakukan oleh (Sharfina & Santoso, 2016), instrumen pengukuran dengan metode *System Usability Scale* (SUS) pada awalnya disusun menggunakan bahasa Inggris. Daftar pertanyaan pada SUS versi asli kemudian diterjemahkan serta disesuaikan ke dalam bahasa Indonesia melalui proses adaptasi lintas budaya. Penyesuaian ini bertujuan agar kuesioner dapat dipahami dengan baik oleh responden lokal dan dapat digunakan dalam proses pengumpulan data, sebagaimana ditunjukkan pada tabel berikut:

Tabel 2.1 Pertanyaan SUS dalam (Sharfina & Santoso, 2016)

No.	Item
1.	Saya berpikir akan menggunakan sistem ini lagi.
2.	Saya merasa sistem ini rumit untuk digunakan.
3.	Saya merasa sistem ini mudah untuk digunakan.
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8.	Saya merasa sistem ini membingungkan.
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Berdasarkan penjelasan dari (Sauro & Lewis, 2012), proses penilaian SUS diawali dengan menentukan nilai kontribusi untuk setiap butir pertanyaan yang

berada pada rentang 0 hingga 4. Pada pertanyaan bernada positif, yaitu nomor ganjil, nilai kontribusi dihitung dengan mengurangi posisi skala yang dipilih responden dengan angka satu ($x_i - 1$). Sementara itu, pada pertanyaan bernada negatif yang terdapat pada nomor genap, nilai kontribusi diperoleh dengan mengurangkan posisi skala dari angka lima ($5 - x_i$). Setelah seluruh nilai kontribusi dijumlahkan, hasilnya dikalikan dengan 2,5 untuk memperoleh skor SUS. Dengan metode tersebut, nilai akhir SUS berada pada rentang 0 sampai 100 dengan interval kenaikan sebesar 2,5.

Melalui usability testing, pengembang dapat memperoleh wawasan mengenai pengalaman pengguna (UX), menemukan aspek yang perlu diperbaiki, serta meningkatkan kualitas antarmuka dan fungsionalitas sistem agar lebih intuitif. Beberapa aspek utama yang diuji meliputi kemudahan navigasi, efisiensi dalam menyelesaikan tugas, serta tingkat kepuasan pengguna terhadap aplikasi. Hasil dari pengujian ini dapat digunakan untuk mengoptimalkan desain dan fitur perangkat lunak sebelum diluncurkan secara luas.

2.2 Penelitian Terdahulu

Tinjauan penelitian terdahulu berfungsi untuk memberikan masukan dan ide yang berkaitan dengan penelitian yang sedang dilakukan. Dalam penyusunan tugas akhir ini, terdapat lima penelitian sebelumnya yang dijadikan referensi dan bahan perbandingan.

Penelitian yang dilakukan oleh Hastedi & Sejati (2024), dengan judul penelitian Rancang Bangun Aplikasi Keuangan Untuk Mengatur Jumlah Pengeluaran Pribadi Berbasis Android menghasilkan aplikasi mengatur jumlah pengeluaran, yang dapat berjalan secara *offline*. Juga aplikasi ini menampilkan grafik berdasarkan pengeluaran yang dimana dapat membantu pengguna dalam pemantauan. Namun, aplikasi ini terbatas pada penggunaan *offline* dengan antarmuka sederhana, sehingga pengembangan lebih lanjut mencakup fitur *online* serta peningkatan tampilan antarmuka.

Penelitian yang kedua yang dilakukan oleh Juhardi & Khairullah (2019), dalam penelitiannya berjudul Sistem Pencatatan dan Pengelolaan Keuangan Pada Aplikasi Manajemen Keuangan E-Dompet Berbasis Android menghasilkan

aplikasi yang memudahkan pencatatan pemasukan dan pengeluaran secara *offline* di perangkat Android, serta mencakup fitur utama untuk mencatat transaksi dan mengelola kategori keuangan. Riset ini menunjukkan bahwa desain yang sederhana mempermudah pengguna dalam mencatat dan mengevaluasi keuangan tanpa memerlukan koneksi internet. Namun, aplikasi ini terbatas pada penggunaan *offline* dengan antarmuka sederhana, sehingga pengembangan lebih lanjut mencakup fitur *online* serta peningkatan tampilan antarmuka.

Penelitian ketiga yang dilakukan oleh Saputra dkk. (2024), dengan judul Pembangunan Aplikasi *Money Tracker* untuk Meningkatkan Literasi Keuangan dan Pengelolaan Keuangan Pribadi. Aplikasi *Money Tracker* berbasis web ini dirancang untuk membantu masyarakat mengelola keuangan secara efektif dan efisien melalui fitur utama input data keuangan dan rekam jejak keuangan. Berdasarkan hasil pengujian, aplikasi ini memudahkan pengguna dalam memantau arus keuangan, namun masih memiliki keterbatasan, seperti tidak adanya fitur notifikasi untuk pengeluaran berlebihan, belum mendukung pengaturan anggaran berdasarkan kategori yang diatur oleh sistem, serta belum terintegrasi dengan layanan lain. Selain itu, antarmuka aplikasi masih perlu ditingkatkan agar lebih menarik dan intuitif bagi pengguna.

Penelitian keempat yang dilakukan oleh Adeva dkk. (2024), dengan judul Analisis dan Implementasi *FinanceKu*: Sebuah Aplikasi Manajemen Keuangan Pribadi. Aplikasi *FinanceKu* menawarkan pengelolaan pemasukan, pengeluaran, pelacakan anggaran bulanan, dan penetapan tujuan finansial. Berdasarkan hasil pengujian, aplikasi ini efektif dalam memenuhi kebutuhan pengelolaan keuangan pribadi, khususnya dengan antarmuka intuitif dan fitur analisis yang membantu pengguna memahami pola pengeluaran. Namun, beberapa keterbatasan ditemukan, seperti ketiadaan notifikasi untuk pengeluaran berlebihan dan keterbatasan integrasi dengan transaksi otomatis.

Penelitian kelima yang dilakukan oleh Rosidi & Afriyudi (2023), dengan judul Aplikasi Pencatatan Keuangan Pribadi Berbasis Web Mobile Penelitian ini menghasilkan aplikasi web mobile untuk mencatat pemasukan dan pengeluaran dengan antarmuka sederhana yang mudah digunakan, serta dilengkapi laporan

keuangan harian, mingguan, dan bulanan. Pengujian usability menghasilkan skor rata-rata 82, yang masuk kategori *good*. Untuk pengembangan lebih lanjut, aplikasi dapat ditingkatkan dengan fitur integrasi API perbankan untuk pencatatan otomatis, pengingat transaksi, peningkatan keamanan melalui otentikasi dua faktor, dan pengembangan aplikasi mobile native untuk meningkatkan pengalaman pengguna.

Pada Tabel 2.2 menyajikan ringkasan penelitian terdahulu yang relevan. Penelitian dengan pengembangan aplikasi Personal Finance. Studi-studi ini menjadi dasar referensi untuk mengidentifikasi kesenjangan dan peluang dalam penelitian ini.

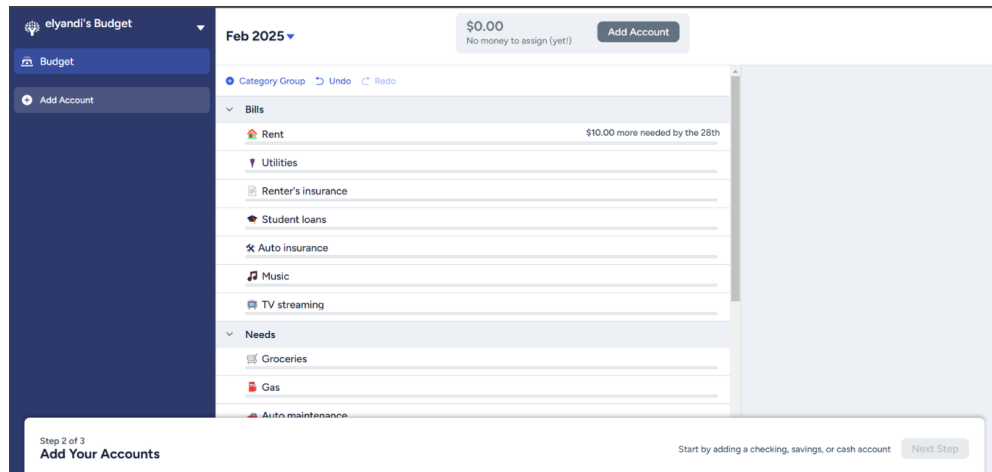
Tabel 2.2 Perbandingan Penelitian

Judul Penelitian	Metodologi	Tujuan	Hasil	Gap
Dimas Rian Hastedi dan Rr. Hajar Puji Sejati	Metode <i>Waterfall</i>	Mengembangkan aplikasi untuk mencatat pemasukan dan pengeluaran, dengan laporan keuangan.	Aplikasi yang dapat Membantu pengguna pencatatan pengeluaran, laporan pengeluaran.	Aplikasi yang dapat Membantu pengguna pencatatan pengeluaran, laporan pengeluaran.
Ujang Juhardi dan Khairullah	Metode <i>Waterfall</i>	Mengembangkan aplikasi untuk memudahkan dalam mencatat pemasukan dan pengeluaran, dengan laporan keuangan.	Aplikasi ini memudahkan pengguna mencatat pemasukan dan pengeluaran, Fitur-fitur utamanya pencatatan transaksi, laporan	Fokus pada pencatatan pengeluaran saja dan tidak, akun pribadi serta tidak bisa mengatur budget.

Judul Penelitian	Metodologi	Tujuan	Hasil	Gap
			keuangan.	
Achmad Dani Saputra, Ananda Hawa Oktavia, Davi Pramudya Putra, Rahmadi Wijaya, dan Erna Hikmawati	Metode <i>Waterfall</i>	Mengembangkan aplikasi untuk mencari alur dari keuangan dari mencatat pemasukan dan pengeluaran, dengan laporan keuangan.	Aplikasi money tracker membantu masyarakat mengelolah keuangan, dengan fitur utama input data keuangan dan rekam jejak keuangan.	Fokus pada inputan pencatatan pengeluaran dan pengeluaran saja untuk melakukan tracking pada keuangan.
Muhammad Adeva, Hapsoro Wisnu Murti, Lailiatul Qudriyah, Rayhan Gading Andri Purwanto, dan Fawwaz Ali Akbar	Metode <i>Design Thinking</i>	Mengembangkan aplikasi untuk membantu pengguna dalam mengatur budget pengeluaran.	Aplikasi menyediakan fitur-fitur utama, menambah budget bulanan, pengelolaan pemasukan dan pengeluaran, serta laporan keuangan.	Fokus utama pada budget yang hanya dapat di atur oleh pengguna.

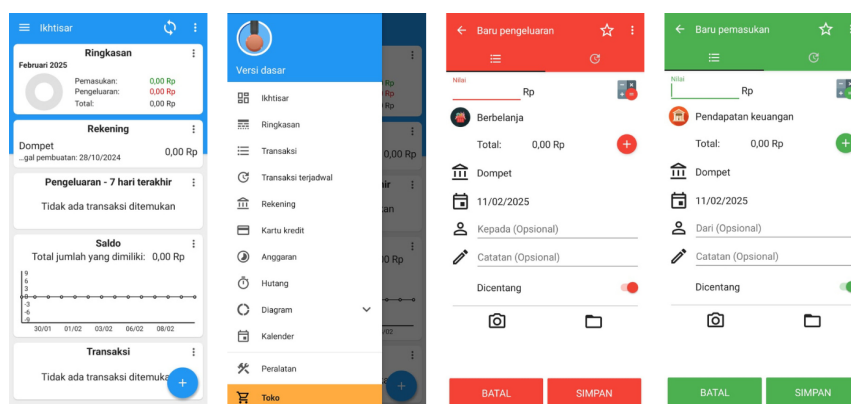
Judul Penelitian	Metodologi	Tujuan	Hasil	Gap
Alvin Rosidi dan Afriyudi	Metode <i>Object Oriented Analyses Design Gap (OOAD)</i>	Mengembangkan aplikasi untuk dengan laporan keuangan berdasarkan periode.	Aplikasi pencatatan keuangan pribadi yang mencatat pemasukan dan pengeluaran, Laporan keuangan harian, mingguan, dan bulanan, yang dapat berjalan di system operasi website dan mobile.	Fokus pada pencatatan pengeluaran, pemasukkan dan pelaporan keuangan, serta tidak fitur budget.

Selain beberapa penelitian tersebut, terdapat juga beberapa aplikasi yang sudah pernah dibuat sebelumnya, salah satunya adalah YNAB (*You Need a Budget*). Aplikasi ini dipilih sebagai referensi karena merupakan salah satu aplikasi pengelolaan keuangan terbaik dengan rating 4.8 dari skala 5, yang menunjukkan tingkat kepuasan pengguna yang sangat tinggi. YNAB telah terbukti efektif dalam membantu pengguna mengelola keuangan dengan sistem budgeting, memungkinkan pengguna mengatur anggaran, melacak pengeluaran, mengurangi utang, serta menabung dengan lebih disiplin. Pemilihan YNAB sebagai pembandingan dilakukan untuk mempelajari fitur-fitur unggulannya yang telah diakui oleh pengguna, serta mengidentifikasi peluang pengembangan aplikasi yang lebih sesuai dengan kebutuhan mahasiswa sebagai target pengguna.



Gambar 2.2 Tampilan aplikasi YNAB (*You Need a Budget*)

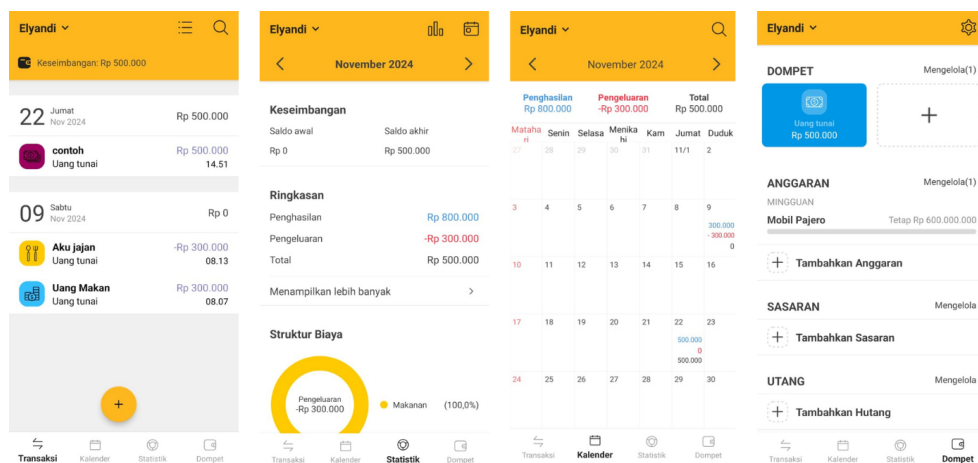
Aplikasi selanjutnya yang dijadikan referensi adalah Anggaran Cepat, yang memiliki rating penilaian 4.8 dari skala 5, menunjukkan tingkat kepuasan pengguna yang sangat tinggi. Aplikasi ini memiliki fitur utama yang relevan dengan penelitian ini, yaitu Anggaran dan Transaksi. Berdasarkan dokumentasinya, fitur Anggaran memungkinkan pengguna untuk mengetahui jumlah uang yang tersedia dan memilih lebih dari satu kategori pengeluaran. Sementara itu, fitur Transaksi memungkinkan pengguna untuk mencatat pemasukan dan pengeluaran secara terperinci. Pemilihan Anggaran Cepat sebagai referensi dilakukan karena aplikasi ini telah terbukti efektif dalam membantu pengguna mengelola keuangan sehari-hari, serta memiliki fitur-fitur yang dapat diadaptasi dan dikembangkan lebih lanjut untuk memenuhi kebutuhan sebagai target pengguna.



Gambar 2.3 Tampilan aplikasi Anggaran Cepat

Aplikasi Smart Budget juga dijadikan referensi dalam penelitian ini.

Aplikasi ini memiliki rating penilaian 4.9 dari skala 5, yang menunjukkan tingkat kepuasan pengguna yang sangat tinggi. Fitur utama yang dimiliki oleh Smart Budget meliputi Transaksi, Kalender, Statistik, dan Dompot. Fitur Kalender memungkinkan pengguna untuk melacak pengeluaran berdasarkan waktu, sementara fitur Statistik memberikan visualisasi data keuangan yang membantu pengguna memahami pola pengeluaran mereka. Selain itu, aplikasi ini dikenal mudah digunakan, sehingga cocok untuk pengguna dengan berbagai tingkat pemahaman tentang pengelolaan keuangan. Pemilihan *Smart Budget* sebagai referensi dilakukan karena aplikasi ini telah terbukti efektif dalam membantu pengguna mengelola keuangan secara komprehensif, serta memiliki antarmuka yang *user-friendly* yang dapat dijadikan inspirasi dalam pengembangan aplikasi kami.



Gambar 2.4 Tampilan aplikasi *Smart Budget*

Pada Tabel 2.2 diperlihatkan beberapa aplikasi yang sudah ada. Analisis terhadap aplikasi-aplikasi tersebut dilakukan untuk mengidentifikasi kelebihan dan kekurangannya, sehingga dapat menjadi acuan dalam menentukan pendekatan pengembangan yang lebih efektif. Berikut adalah Tabel Aplikasi Kompetitor:

Tabel 2.3 Aplikasi Kompetitor

Nama Aplikasi	Platform	Teknologi yang digunakan	Output
YNAB (<i>You Need a Budget</i>)	<i>Web, Mobile Android & Mobile IOS</i>	Penggunaan algoritma dengan basis nya adalah aturan yang sudah ditentukan.	Menghasilkan Aplikasi mengelola keuangan, memudahkan pelacakan pengeluaran, pengurangan utang, dan peningkatan tabungan.
Anggaran Cepat	<i>Web, Mobile Android & Mobile IOS</i>	Penggunaan algoritma dengan basis nya adalah aturan yang sudah ditentukan.	Menghasilkan Aplikasi mengelola keuangan, dan pencatatan yang simple.
<i>Smart Budget</i>	<i>Mobile Android</i>	Penggunaan algoritma dengan basis nya adalah aturan yang sudah ditentukan.	Menghasilkan Aplikasi yang mudah digunakan, mengelola keuangan, memudahkan pelacakan pengeluaran dan pencatatan yang simple.