BAB II

LANDASAN TEORI

2.1 Tinjauan Teori

2.1.1 Unified Modelling Language (UML)

UML atau *Unified Modeling Language*, adalah bahasa pemodelan standar yang terdiri dari serangkaian diagram terintegrasi, yang dikembangkan untuk membantu pengembang sistem dan perangkat lunak untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan sistem non-perangkat lunak lainnya. UML mewakili kumpulan praktik rekayasa terbaik yang telah terbukti berhasil dalam pemodelan sistem yang besar dan kompleks. UML adalah bagian yang sangat penting dalam pengembangan perangkat lunak berorientasi objek dan proses pengembangan perangkat lunak. UML sebagian besar menggunakan notasi grafis untuk mengekspresikan desain proyek perangkat lunak. Menggunakan UML membantu tim proyek berkomunikasi, mengeksplorasi desain potensial, dan memvalidasi desain arsitektur perangkat lunak. Ada beberapa cara untuk memecah diagram model UML untuk menjadi perspektif atau tampilan yang dapat menangkap aspek tertentu dari sistem[1].

- **Logical View**

  Menjelaskan deskripsi abstrak dari bagian-bagian sistem. Digunakan untuk memodelkan apa sistem terdiri dari dan bagaimana bagian-bagian tersebut berinteraksi satu sama lain. Jenis diagram UML yang biasanya membentuk ini view termasuk class, object, state machine, dan diagram interaksi.

- **Process View**

  Menjelaskan proses dalam sistem. Ini sangat membantu ketika memvisualisasikan apa yang harus terjadi dalam sistem. Tampilan ini biasanya berisi diagram aktivitas.

- **Development View**

  Menjelaskan bagaimana bagian-bagian sistem diatur ke dalam modul dan komponen. Ini sangat berguna untuk mengelola lapisan dalam arsitektur sistem. Tampilan ini biasanya berisi paket dan komponen diagram.

- **Physical View**

  Menjelaskan bagaimana desain sistem, seperti yang dijelaskan dalam tiga tampilan sebelumnya, kemudian dihidupkan sebagai satu set entitas dunia nyata. Diagram dalam tampilan ini menunjukkan bagaimana bagian-bagian abstrak dipetakan ke dalam final sistem yang dikerahkan. Tampilan ini biasanya berisi diagram penerapan.

- **Use Case View**

  Menjelaskan fungsionalitas sistem yang dimodelkan dari perspektif dunia luar. Pandangan ini diperlukan untuk menggambarkan apa yang seharusnya dilakukan oleh sistem. Semua pandangan lain bergantung pada gunakan tampilan kasus untuk memandu merekaitu sebabnya model ini disebut 4+1. Tampilan ini biasanya berisi use case diagram, deskripsi, dan diagram *overview*.

2.1.2 Bahasa Pemrograman Javascript

JavaScript adalah bahasa pemrograman dinamis yang digunakan untuk pengembangan *web*, dalam aplikasi *web*, untuk pengembangan *game*, dan banyak lagi. Ini memungkinkan para penggunanya untuk menerapkan fitur dinamis pada halaman *web* yang tidak dapat dilakukan hanya dengan *HTML* dan *CSS*. Banyak *browser* menggunakan *JavaScript* sebagai bahasa *scripting* untuk melakukan hal-hal dinamis di *web*. Tampilan menu tarik-turun klik untuk menampilkan, konten tambahan ditambahkan ke halaman, dan warna elemen yang berubah secara dinamis pada halaman, untuk menyebutkan beberapa fitur, merupakan efek dari *JavaScript*.

2.1.3 Framework React Native

React Native adalah kerangka kerja *JavaScript* untuk menulis *Mobile Application* yang nyata dan asli untuk *iOS* dan *Android*. Ini didasarkan pada *React*, perpustakaan *JavaScript Facebook* untuk membangun antarmuka pengguna, tetapi alih-alih menargetkan browser, ini menargetkan platform seluler. Dengan kata lain: pengembang *web* kini dapat menulis *Mobile Application* yang terlihat dan terasa benar-benar "asli", semuanya dari kenyamanan pustaka *JavaScript* yang sudah kita kenal dan sukai. Plus, karena sebagian besar kode yang ditulis dapat dibagikan antar platform, React Native memudahkan pengembangan secara bersamaan untuk Android dan iOS[2].

Mirip dengan *React* untuk *Web*, aplikasi *React Native* ditulis menggunakan campuran *JavaScript* dan *markup XML*, yang dikenal sebagai *JSX*. Kemudian, di bawah tenda, "jembatan" *React Native* memanggil API *rendering* asli di *Objective-C* (untuk *iOS*) atau *Java* (untuk *Android*). Dengan demikian, aplikasi akan dirender menggunakan komponen *UI* seluler nyata, bukan tampilan *web*, dan akan terlihat dan terasa seperti *Mobile Application* lainnya. *React Native* juga mengekspos antarmuka *JavaScript* untuk *API platform*, sehingga aplikasi *React Native* dapat mengakses fitur platform seperti kamera ponsel, atau lokasi pengguna. *React Native* saat ini mendukung *iOS* dan *Android*, dan memiliki potensi untuk berkembang ke *platform* masa depan juga.

2.1.4 Airtable Database

Airtable adalah sebuah *platform* yang memudahkan untuk membangun sebuah *custom application* yang kuat. Alat-alat yang disediakan sudah di desain agar dapat mempermudah hampir semua proses dalam perancangan aplikasi seperti, alur kerja, atau proyek. Keuntungan utama yang dimiliki *Airtable* adalah *platformnya* yang bersifat *Low Code/No Code* yang berarti dapat digunakan dengan kemampuan *coding* yang minimal dan bahkan tidak ada pengalaman *coding*. Beberapa hal yang dapat dilakukan dengan menggunakan

Airtable adalah seperti menggunakan untuk melacak wawancara kerja hingga mengelola produksi video skala besar, digunakan oleh perusahaan untuk menjalankan proses bisnis mereka setiap hari.

Pada intinya, *Airtable* memungkinkan para penggunanya untuk dengan mudah membangun sebuah *database* yang menyimpan informasi yang penting untuk pekerjaan, kemudian menggunakannya untuk mendukung visualisasi, proses, dan integrasi yang membentuk aplikasi khusus yang benar-benar unik bagi para penggunanya. Berikut merupakan beberapa hal tentang apa yang mungkin dilakukan dengan Airtable:

- **Pengelolaan pekerjaan "*End-to-end*":**

  Pengguna *Airtable* dapat menggunakannya untuk melacak semua informasi seputar sasaran dan tujuan dan juga dapat menghubungkannya ke alat yang lain untuk mengotomatiskan tindakan yang sangat taktis seperti menerbitkan *tweet* atau mengirim pembaruan *email*.

- **Menghasilkan visualisasi dengan cepat dan jelas:**

  Airtable dapat dengan mempermudah proses visualisasikan data pengguna yang terdapat dalam *database*, seperti menggunakan bagan Gantt untuk melihat bagaimana pencapaian proyek sudah sejajar atau memutar grafik cepat untuk menganalisis kinerja kampanye pemasaran para pengguna.

- **Memberikan konteks yang diperlukan kepada *Stakeholders*:**

  Pengumpulan informasi dari berbagai sumber untuk memastikan agar semua pihak dapat bekerja dengan versi yang terbaru. Kemudian, urutkan, filter, dan atur ulang data untuk membuat *custom entry point* untuk dibagikan dengan rekan-rekan dan pihak yang berkaitan sehingga mereka masing-masing memiliki informasi yang mereka butuhkan, dengan cara yang paling sesuai untuk mereka.

Biasanya, dalam membuat aplikasi secara independen, diperlukan banyak *Software Engineer* dan banyak waktu. Tetapi dengan membangun di *Airtable*, pengguna dapat memanfaatkan *database* yang fleksibel dan kuat yang dibangun dari bawah ke atas untuk memenuhi alur kerja yang paling kompleks dan spesifik. Airtable memberikan para penggunanya kontrol penuh atas aplikasi yang sedang dibangun, *Software Engineer* atau tidak.

2.1.5 Sistem Operasi Android

Gambar 2. 1 Android Operating System

Android OS adalah sistem operasi seluler berbasis *Linux* yang terutama berjalan di *smartphone* dan tablet. Platform *Android* mencakup sistem operasi berdasarkan *kernel Linux*, GUI, *browser web*, dan aplikasi pengguna akhir yang dapat diunduh. Meskipun demonstrasi awal Android menampilkan smartphone QWERTY generik dan layar VGA besar, sistem operasi ini ditulis untuk berjalan pada handset yang relatif murah dengan *keypad* numerik konvensional.

Android dirilis di bawah lisensi *open source Apache* v2; ini memungkinkan banyak variasi *OS* dikembangkan untuk perangkat lain, seperti konsol *game* dan kamera digital. *Android* didasarkan pada perangkat lunak *open-source*, tetapi sebagian besar perangkat *Android* telah diinstal sebelumnya

dengan rangkaian perangkat lunak berpemilik, seperti *Google Maps*, *YouTube*, *Google Chrome, dan Gmail*.

Android memulai hidupnya sebagai perusahaan rintisan berbasis Palo Alto yang disebut *Android Inc*., pada tahun 2003. Awalnya, perusahaan tersebut ingin mengembangkan sistem operasi untuk kamera digital, tetapi upaya tersebut ditinggalkan untuk menjangkau pasar yang lebih luas. *Google* mengakuisisi *Android Inc*. dan karyawan utamanya pada tahun 2005 dengan nilai sedikitnya $50 juta. *Google* memasarkan *platform* seluler awal ke produsen *handset* dan operator seluler dengan manfaat utamanya sebagai fleksibilitas dan kemampuan untuk ditingkatkan[3].

Pada akhir 2007, *Open Handset Alliance* (OHA) mengumumkan pembentukannya. OHA adalah koalisi lebih dari 30 perusahaan perangkat keras, perangkat lunak dan telekomunikasi, termasuk *Google, Qualcomm, Broadcom, HTC, Intel, Samsung, Motorola, Sprint, Texas Instruments* dan operator nirkabel Jepang KDDI dan NTT DoCoMo. Tujuan aliansi adalah untuk berkontribusi pada pengembangan *platform* sumber terbuka pertama untuk perangkat seluler.

Google merilis versi *beta* publik *Android* 1.0 untuk pengembang sekitar waktu yang sama dengan pengumuman aliansi, pada November 2007. Tidak sampai *Google* merilis Android 1.5 pada April 2009 *Google* memperkenalkan skema penamaan bertema makanan penutup khas Android; nama Android 1.5 adalah "*Cupcake*." Sekitar waktu rilis *Android* 4.4 *KitKat*, *Google* merilis pernyataan resmi untuk menjelaskan penamaan: "Karena perangkat ini membuat hidup kita begitu manis, setiap versi *Android* dinamai makanan penutup." Namun, pada tahun 2019, *Google* meninggalkan nama makanan penutup dalam rebranding *Android*; *Android* 10 hanya dikenal sebagai *Android Q*.

2.1.6 Mobile Application

Mobile Application, paling sering disebut sebagai *App*, adalah jenis *Software* yang dirancang untuk berjalan di perangkat seluler, seperti *smartphone* atau komputer tablet. *Mobile Application* sering kali memberikan layanan yang serupa kepada pengguna dengan yang diakses di PC. *App* umumnya kecil, unit perangkat lunak individual dengan fungsi terbatas. Penggunaan *Software app* ini awalnya dipopulerkan oleh *Apple Inc*. dan *App Store*-nya, yang menawarkan ribuan aplikasi untuk *iPhone*, *iPad*, dan *iPod Touch*.

Mobile Application menjauh dari sistem perangkat lunak terintegrasi yang umumnya ditemukan di PC. Sebagai gantinya, setiap aplikasi menyediakan fungsionalitas terbatas dan terisolasi seperti *game*, kalkulator, atau penjelajahan *web* seluler. Meskipun aplikasi mungkin menghindari *multitasking* karena sumber daya perangkat keras yang terbatas dari perangkat seluler awal, kekhususannya sekarang menjadi bagian dari keinginan mereka karena memungkinkan konsumen untuk memilih sendiri apa yang dapat dilakukan perangkat mereka.

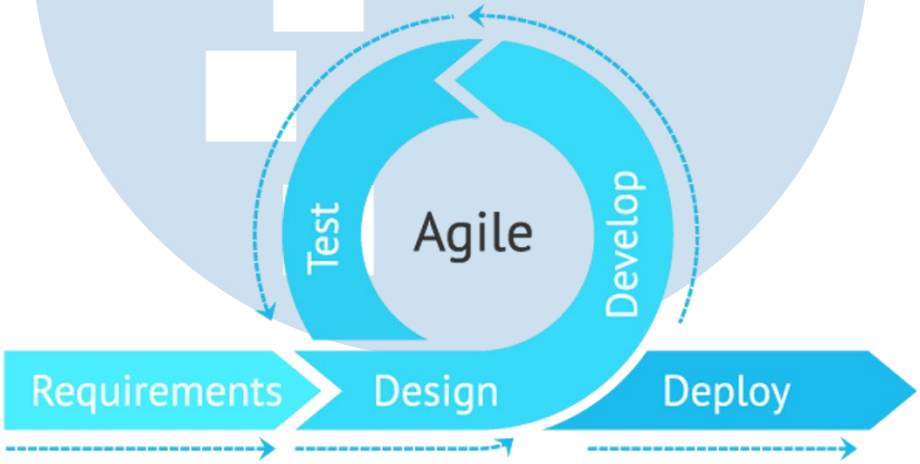
Mobile Application paling sederhana mengambil aplikasi berbasis PC dan memindahkannya ke perangkat seluler. Karena *Mobile Application* menjadi lebih kuat, teknik ini agak kurang. Pendekatan yang lebih canggih melibatkan pengembangan khusus untuk lingkungan seluler, mengambil keuntungan dari keterbatasan dan kelebihannya. Misalnya, aplikasi yang menggunakan fitur berbasis lokasi secara inheren dibangun dari bawah ke atas dengan memperhatikan seluler mengingat pengguna tidak terikat pada suatu lokasi, seperti pada PC.

Aplikasi dibagi menjadi dua kategori besar: aplikasi asli dan aplikasi *web*. Aplikasi asli dibuat untuk sistem operasi seluler tertentu, biasanya *iOS* atau *Android*. Aplikasi asli menikmati kinerja yang lebih baik dan antarmuka

pengguna (UI) yang lebih halus, dan biasanya harus melewati proses pengembangan dan jaminan kualitas yang jauh lebih ketat sebelum dirilis.

Aplikasi *web* digunakan dalam *HTML5* atau *CSS* dan memerlukan memori perangkat minimum karena dijalankan melalui *browser*. Pengguna diarahkan ke halaman *web* tertentu, dan semua informasi disimpan di *database* berbasis *server*. Aplikasi *web* memerlukan koneksi yang stabil untuk digunakan[4].

2.1.7 Agile Software Development Method



Gambar 2. 2 Agile Software Development

Agile Software Development, juga disebut hanya sebagai *Agile* adalah jenis metodologi pengembangan yang mengantisipasi kebutuhan akan fleksibilitas dan menerapkan tingkat pragmatisme pada pengiriman produk jadi. Agile Software Development memerlukan perubahan budaya di banyak perusahaan karena berfokus pada pengiriman yang bersih dari bagian-bagian atau bagian-bagian perangkat lunak dan bukan pada keseluruhan aplikasi.

Manfaat *Agile* termasuk kemampuannya untuk membantu tim dalam lanskap yang berkembang sambil mempertahankan fokus pada penyampaian nilai bisnis yang efisien. Budaya kolaboratif yang difasilitasi oleh *Agile* juga meningkatkan efisiensi di seluruh organisasi saat tim bekerja bersama dan

memahami peran spesifik mereka dalam proses tersebut. Terakhir, perusahaan yang menggunakan pengembangan perangkat lunak Agile dapat merasa yakin bahwa mereka merilis produk berkualitas tinggi karena pengujian dilakukan selama pengembangan, memberikan kesempatan untuk membuat perubahan sesuai kebutuhan dan memperingatkan tim terhadap potensi masalah apa pun[5].

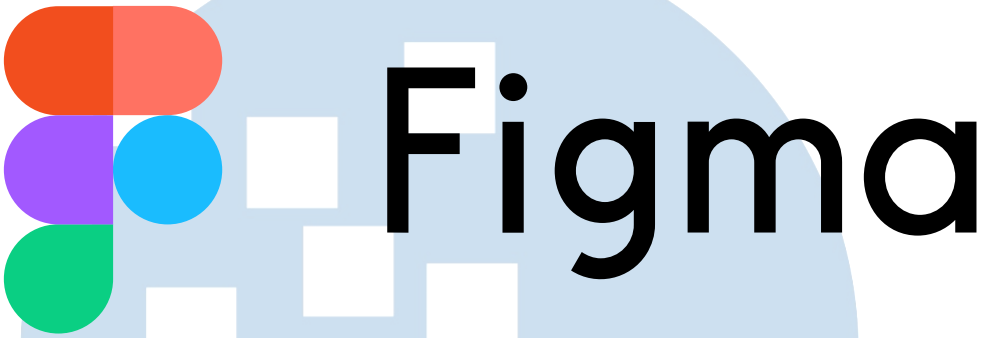
2.1.8 Bravo Studio

Gambar 2. 3 Bravo Studio

Bravo Studio adalah pembuat aplikasi untuk desainer. Ini memungkinkan untuk mengubah prototipe Figma menjadi aplikasi *iOS* dan *Android* asli secara instan tanpa menulis kode apa pun. *Bravo Studio* menggunakan tindakan/komponen seluler asli untuk menjadikan desain *UI* aplikasi untuk menjadi aplikasi asli melalui apa yang disebut Tag Bravo.

Tag Bravo memungkinkan Bravo untuk mengenali perilaku asli dalam desain aplikasi seperti berbagi, menu *slide*, layar *splash*, animasi *Lottie*, dll. untuk membuat komponen menjadi nyata saat mengonversinya menjadi aplikasi nyata di *Bravo Studio*. Bagian penting lainnya dari *Bravo Studio* adalah dapat menghubungkan prototipe dengan *data* waktu nyata. Ini memungkinkan aplikasi untuk menarik dan menampilkan data dari API apa pun atau layanan pihak ketiga seperti *Airtable*. Setiap perubahan pada data akan secara otomatis disinkronkan dan akan ditampilkan langsung di aplikasi dalam hitungan detik. Buat prototipe aplikasi di Figma dan beri nama setiap layer agar kompatibel dengan Bravo[6].

2.1.9 Figma



Gambar 2. 4 Figma

Figma adalah salah satu *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain. Figma bisa digunakan di sistem operasi windows, linux ataupun mac dengan terhubung ke internet. Umumnya *Figma* banyak digunakan oleh seseorang yang bekerja dibidang *UI/UX*, *web design* dan bidang lainnya yang sejenis.

Selain mempunyai kelengkapan fitur layaknya *Adobe XD*, *Figma* memiliki keunggulan yaitu untuk pekerjaan yang sama dapat dikerjakan oleh lebih dari satu orang secara bersama-sama walaupun ditempat yang berbeda. Hal tersebut bisa dikatakan kerja kelompok dan karena kemampuan aplikasi figma tersebutlah yang membuat aplikasi ini menjadi pilihan banyak *UI/UX designer* untuk membuat *prototype website* atau aplikasi dengan waktu yang cepat dan efektif[7].

2.1.10 User Experience

User Experience berarti persepsi manusia dan tanggapan yang dihasilkan dari penggunaan atau penggunaan yang diharapkan dari suatu produk, sistem, atau layanan. Persepsi pengguna meliputi emosi, preferensi, kenyamanan, perilaku, dan pencapaian, yang terjadi sebelum, selama dan setelah penggunaan. Interaksi dengan pengguna adalah hasil dari citra merek, presentasi, fungsionalitas, produktivitas sistem, perilaku interaktif, dan kapasitas tambahan

sistem, produk, atau layanan. Itu juga merupakan hasil dari kondisi *internal* dan fisik pengguna, yang muncul dari pengalaman sebelumnya, sikap, keterampilan seseorang, dari konteks penggunaan [8].

2.1.11 User Interface

User Interface mencakup semua komponen dari sistem interaktif (perangkat lunak atau perangkat keras), yang menyediakan informasi dan cara untuk mengelolanya, yang memungkinkan pengguna untuk melakukan tugas-tugas tertentu dengan bantuan sistem interaktif. Dengan kata lain, antarmuka pengguna adalah perantara antara manusia dan komputer, mendorong interaksi; kegunaan adalah alat yang menyediakan kegiatan untuk interaksi yang efektif untuk mencapai tujuan yang pasti[8].

2.1.12 Native Application

Native Application mengacu pada aplikasi yang khusus ditulis dan dikembangkan untuk ponsel tertentu sistem operasi. Tiga operasi seluler terkemuka sistem adalah *Google Android, Apple iOS*, dan *Windows Phone*. Untuk menciptakan yang benar, asli aplikasi, bahasa pemrograman Java harus digunakan untuk Android, bahasa pemrograman *Objective C* untuk *iOS*, dan kerangka kerja *.NET* untuk *Windows Phone*. Karakteristik utama yang umum dari aplikasi asli adalah bahwa aplikasi ini memiliki akses tanpa hambatan ke perangkat perangkat keras dan mendukung semua antarmuka dan interaksi pengguna tersedia di lingkungan operasi seluler masing-masing[9].

2.1.13 Firebase Database

Gambar 2. 5 Google Firebase

Firebase dianggap sebagai *platform* aplikasi *web*. Itu membantu *developer-developer* untuk membuat aplikasi berkualitas tinggi. Firebase menyimpan data di Format Notasi Objek *JavaScript* (JSON) yang tidak menggunakan permintaan untuk memasukkan, memperbarui, menghapus, atau menambahkan data ke dalamnya. *Firebase* merupakan *backend* dari sistem yang digunakan sebagai database untuk menyimpan data[10]. *Firebase* menyediakan beberapa layanan, antara lain:

- **Firebase Analytics**

  Ini memberikan wawasan tentang penggunaan aplikasi. Ini adalah aplikasi berbayar solusi pengukuran yang juga menyediakan keterlibatan pengguna. Fitur unik ini memungkinkan pengembang aplikasi untuk memahami bagaimana pengguna menggunakan aplikasi. SDK memiliki fitur menangkap peristiwa dan properti sendiri dan juga memungkinkan mendapatkan data khusus

- **Firebase Cloud Messaging**

  Ini sebelumnya dikenal sebagai Google Clouds Messaging (GCM), FCM adalah layanan berbayar yang merupakan solusi lintas platform untuk pesan dan pemberitahuan untuk Android, Aplikasi Web, dan iOS.

- **Firebase Authentication**

  Firebase Auth mendukung penyedia login sosial seperti Facebook, Google GitHub, dan Twitter. Ini adalah layanan yang dapat mengautentikasi pengguna hanya menggunakan kode sisi klien dan ini berbayar melayani. Ini juga mencakup sistem manajemen pengguna dimana pengembang dapat mengaktifkan otentikasi pengguna dengan email dan login kata sandi disimpan dengan Firebase

- **Real-time Database**

  Firebase menyediakan layanan seperti database waktu nyata dan bagian belakang. API disediakan untuk pengembang aplikasi yang memungkinkan data aplikasi disinkronkan di seluruh klien dan disimpan di cloud Firebase. Pustaka klien adalah disediakan oleh perusahaan yang memungkinkan integrasi dengan Aplikasi Android, iOS, dan JavaScript.

- **Firebase Storage**

  Ini memfasilitasi transfer file yang mudah dan aman terlepas dari jaringan kualitas untuk aplikasi Firebase. Ini didukung oleh Google Cloud Storage yang merupakan layanan penyimpanan objek hemat biaya. Itu pengembang dapat menggunakannya untuk menyimpan gambar, audio, video, atau lainnya konten buatan pengguna

- **Firebase Test Lab for Android**

  Ini menyediakan infrastruktur berbasis cloud untuk menguji Android aplikasi. Dengan satu operasi, pengembang dapat memulai pengujian aplikasi mereka di berbagai perangkat dan perangkat konfigurasi. Berbagai hasil pengujian seperti tangkapan layar, video dan log tersedia di konsol Firebase.. Meskipun pengembang belum menulis kode pengujian apa pun untuk aplikasi mereka, Test Lab dapat menjalankan aplikasi secara otomatis, mencari kerusakan

- **Firebase Crash Reporting**

  Laporan terperinci tentang kesalahan dibuat di aplikasi. Itu kesalahan dikelompokkan ke dalam kelompok jejak tumpukan serupa dan diprioritaskan berdasarkan tingkat keparahannya. Fitur lainnya adalah: pengembang dapat mencatat peristiwa khusus untuk membantu menangkap langkah-langkah yang mengarah terjadinya *crash*.
- **Firebase Notifications**

  Ini memungkinkan pemberitahuan pengguna yang ditargetkan untuk *Mobile Application* pengembang dan layanan tersedia secara bebas.

2.2 Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu 1

Penulis	NA Dewananto, H Tolle, HM Az-Zahra
Nama Jurnal	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer
Judul Artikel	Perancangan User Experience Menggunakan Metode Human Centered Design Pada Aplikasi Mobile Portal Berita Tabloidjubi[11]
Permasalahan	Permasalahan yang ditemukan dalam aplikasi pada evaluasi awalnya adalah kategorisasi yang rumit dan tidak adanya fungsi pengguna.
Metode	Heuristic Evaluation
Kesimpulan	Hasil dari penelitian adalah identifikasi dan analisis masalah usability dan rekomendasi rancangan untuk perbaikan lebih lanjut berupa high-fidelity prototype sebagai gambaran solusi.

Tabel 2. 2 Penelitian Terdahulu 2

Penulis	Inda Dwi Rahmadani, Cristopher, M. Rizki Ramadhan, Swadexi Istiqphara
Nama Jurnal	Jurnal ELECTRON
Judul Artikel	Rancang Bangun Aplikasi Virtual Reality Bersepeda Berbasis Android Dengan Menggunakan Metodologi Pengembangan Perangkat Lunak Agile[12]
Permasalahan	Olahraga merupakan kegiatan yang penting untuk menjaga kesehatan badan, namun banyak orang tidak memiliki cukup waktu dan tempat untuk berolahraga sepeda diluar ruangan.
Metode	Agile Software Development
Kesimpulan	Hasil yang diperoleh menunjukan bahwa perangkat lunak yang dibangun dapat bekerja dengan baik baik dalam kecepatan gambar dan fungsional sistem.

Tabel 2. 3 Penelitian Terdahulu 3

Penulis	Rifqi Fahrudin, Reza Ilyasa
Nama Jurnal	Jurnal Ilmiah Teknologi Informasi Terapan
Judul Artikel	Perancangan Aplikasi "Nugas" Menggunakan Metode Design Thinking Dan Agile Development[13]
Permasalahan	Kurangnya minat mahasiswa dalam mengerjakan tugas karna dianggap hal yang membebankan dalam proses pembelajaran.
Metode	Design Thinking dan Agile Software Development
Kesimpulan	Menghasilkan sebuah aplikasi yang dapat membantu mahasiswa dalam mengelola dan mengerjakan tugas.

Tabel 2. 4 Penelitian Terdahulu 4

Penulis	Sri Astuti
Nama Jurnal	Universitas Mercu Buana Jakarta
Judul Artikel	Perancangan Aplikasi Mobile Android Untuk Pendaftaran Kegiatan Berbasis Android[14].
Permasalahan	Kendala dari proses registrasi dalam Universitas Mercu Buana secara manual yaitu salah satunya memakan banyak waktu untuk proses memasukkan data.
Metode	Metode Scrum Agile
Kesimpulan	Aplikasi Pendaftaran Kegiatan berbasis Android untuk mempermudah Mahasiswa dan Panitia dalam melakukan proses registrasi sehingga Mahasiswa tidak perlu repot-repot mengantri dalam melakukan suatu proses registrasi pendaftaran kegiatan di Universitas Mercu Buana.

Tabel 2. 5 Penelitian Terdahulu 5

Penulis	Edgar Winata, Johan Setiawan
Nama Jurnal	UltimaInfoSys: Jurnal Ilmu Sistem Informasi
Judul Artikel	Analisis dan Perancangan Prototipe Aplikasi Tracking Bis Universitas Multimedia Nusantara pada Platform Android[15].
Permasalahan	Kemacetan dan kepadatan penduduk menjadi suatu kendala sehingga bis tidak dapat mencapai titik keberangkatan dengan tepat waktu sesuai dengan jadwal yang sudah dibuat.
Metode	Observasi, perancangan system menggunakan diagram-diagram UML, perancangan user interface menggunakan pedoman dan prinsip perancangan dari Ben Schneiderman
Kesimpulan	• Prototipe aplikasi tracking bis kampus ini dirancang untuk bagian Marketing yang mengurus bis kampus dan mahasiswa pengguna jasa transportasi bis kampus dan menggunakan smartphone dengan sistem operasi Android versi 2.2 (Froyo). • Prototipe aplikasi tracking bis ini merupakan fasilitator untuk pengguna agar dapat melihat dan mengetahui rute bis kampus, posisi bis kampus, titik pemberhentian bis kampus dan jadwal bis kampus.

Dapat dilihat pada tabel-tabel yang diatas merupakan penelitian terdahulu yang digunakan sebagai referensi dalam penulisan penelitian ini. Dapat dilihat dari beberapa metode yang digunakan dalam penelitian terdahulu merupakan metode Agile Software Development yang merupakan metode yang juga digunakan dalam penelitian ini dalam perancangan sebuah aplikasi yang berbasis Android. Perbedaan yang dimiliki oleh penelitian ini adalah tools yang digunakan dalam proses perancangan aplikasi Android. Sebagian besar dari penelitian yang digunakan sebagai referensi menggunakan program seperti Android Studio untuk merancang

aplikasinya, sedangkan penelitian ini menggunakan *Bravo Studio* dalam proses perancangannya. Perbedaan yang paling besar tentang *Android Studio* dengan *Bravo Studio* adalah proses perancangan dengan *Android Studio* memerlukan kemampuan dasar dalam *coding* untuk digunakan sedangkan *Bravo Studio* tidak memerlukan kemampuan dasar dalam *coding* karena *Bravo Studio* merupakan sebuah *no-code app builder*.