

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Tinjauan pustaka adalah rangkuman dari penelitian sebelumnya yang memiliki keterkaitan atau relevansi dengan studi yang akan dilakukan. Penelitian sebelumnya berfungsi sebagai titik perbandingan atau referensi bagi penelitian yang sedang direncanakan. Tujuan dari pengkajian pustaka ini adalah untuk memperluas pemahaman terhadap studi yang sedang disusun.

Penelitian dengan topik pembangunan aplikasi mobile berbasis Android telah banyak dibuat, beberapa metode populer digunakan untuk pembangunan aplikasi mobile yaitu Software Development Life-Cycle (SDLC), Metodologi Waterfall, Metodologi Agile, dan Rapid Application Development (RAD). Adapun beberapa tinjauan pustaka untuk membantu pembuatan penelitian ini yaitu:

Table 2.1 Penelitian Sebelumnya

No	Title	Comparing	Contrasting	Critize	Synthesize	Summarize
1	Rancang Bangun Sistem Informasi Persediaan Obat Berbasis Web Menggunakan Model RAD[11]	Penelitian ini membuat sistem pengecekan dan persediaan obat berbasis web yang menggunakan metodologi RAD dengan pengujian menggunakan metode <i>blackbox</i>	Penelitian ini hanya berbasis Web dan menggunakan server lokal	Untuk penelitian selanjutnya dapat dicoba untuk platform lain	Penelitian ini dapat membuat sistem persediaan dan pengecekan yang cukup menarik namun untuk penelitian selanjutnya akan memakai server <i>cloud</i>	Penelitian ini menghasilkan sistem berbasis web yang dapat menyimpan dan mengecek ketersediaan obat di gudang yang diuji menggunakan metode <i>blackbox</i> .
2.	Analisa Dan Perancangan Aplikasi Sistem <i>Invetory</i> (Studi Kasus: PT. Cakra Medika Utama)[12]	Penelitian ini merancang sistem informasi <i>inventory</i> barang yang akan berbasis di web menggunakan metodologi RAD	Penelitian ini hanya mencapai tahap analisa dan perancangan	Untuk penelitian selanjutnya dapat diteruskan ke tahap pembuatan	Penelitian ini membuat perancangan sistem <i>inventory</i> barang yang cukup detail namun tidak sampai di tahap pembuatan.	Penelitian ini menghasilkan rancangan <i>mockup</i> sistem <i>inventory</i> yang dapat mengecek stok barang, detail transaksi dan mengecek laporan barang menggunakan metodologi RAD

No	Title	Comparing	Contrasting	Critize	Synthesize	Summarize
3.	Aplikasi Sistem Persediaan Barang Gudang PT. Berkah Pena Ilmu Menggunakan Android Studio Dan Metode <i>Rapid Application Development</i> (RAD)[7]	Penelitian ini membuat aplikasi sistem penambahan dan pengecekan ketersediaan barang berbasis <i>mobile</i> yang mempunyai fitur barcode dan menggunakan metodologi RAD serta menggunakan pengujian metode <i>blackbox</i>	Penelitian ini belum terintegrasi dengan <i>server cloud</i> dan hanya menggunakan penyimpanan lokal	Untuk penelitian selanjutnya dapat di integrasikan dengan <i>server cloud</i>	Penelitian ini membuat sistem aplikasi yang dapat <i>meng-scan barcode</i> untuk memasukkan barang ke dalam database serta mengecek stok barang yang ada di database namun penyimpanan masih memakai penyimpanan lokal di perangkat.	Penelitian ini menghasilkan aplikasi berbasis <i>mobile</i> yang dapat mengecek, menambah dan <i>scan barcode</i> untuk menambahkan barang ke <i>database</i> .

No	Title	Comparing	Contrasting	Critize	Synthesize	Summarize
4.	Penerapan Metode RAD Untuk Pengembangan Sistem Inventori Obat Berbasis Web Menggunakan Framework <i>Laravel</i> (Studi Kasus Pada Klinik Sukaraya Medika Kabupaten Bekasi).[9]	Penelitian ini membuat sistem inventory berbasis web yang menggunakan metode RAD dan pengujian menggunakan metode <i>blackbox</i>	Penelitian ini hanya membuat sistem berbasis web	Untuk penelitian selanjutnya dapat dibuat di platform lain agar lebih fleksibel	Penelitian ini membuat sistem inventori obat yang di mana peneliti dapat mengetahui berapa banyak obat yang ada dan tanggal kadaluarsanya namun tidak ada fitur untuk penjualan obatnya.	Penelitian ini menghasilkan sistem berbasis <i>web</i> yang dapat mengecek inventori obat serta detailnya seperti laporan data obat, data masuk serta laporan <i>expired</i> obat.
5	Implementasi model <i>Rapid Application Development</i> Pada Sistem Informasi Persediaan Barang Dengan metode FIFO [14]	Penelitian ini membuat sistem persediaan barang berbasis <i>desktop</i> menggunakan metode RAD dan FIFO	Penelitian ini hanya memakai aplikasi berbasis <i>desktop</i> serta memakai penyimpanan lokal	Untuk penelitian selanjutnya dapat dibuat di platform lain dan memakai <i>server cloud</i> agar lebih fleksibel	Penelitian ini membuat sistem <i>inventory</i> barang dan persediaannya namun UI yang dipakai sudah usang dan perlu diperbaharui	Penelitian ini menghasilkan aplikasi berbasis <i>desktop</i> yang berguna untuk mengecek persediaan barang dan pembelian barang.

Penelitian berjudul “Rancang Bangun Sistem Informasi Persediaan Obat Berbasis Web Menggunakan Model RAD”. Penelitian ini dilakukan oleh Biktra Rudianto, Yuni Eka Achyani, dan Indah Ariyati. Jurnal ini membahas mengenai masalah proses pengelolaan data dan penyediaan informasi khususnya data laporan keluar masuk barang, laporan stok akhir barang. Jurnal ini menghasilkan sistem berbasis web yang dapat mengelola data barang yang lebih efisien dan akurat dan melakukan pengujian menggunakan blackbox.

Penelitian berjudul “Analisa Dan Perancangan Aplikasi Sistem Inventory (Studi Kasus: PT. Cakra Medika Utama)”. Penelitian ini dilakukan oleh Sitiani Zalukhu dan Inge Handriani, jurnal ini berasal dari Universitas Mercu Buana Jakarta Selatan. Jurnal ini membahas mengenai bagaimana mitra masih menggunakan pencatatan di buku besar untuk proses pendataan dan pelaporan persediaan barang masuk dan keluar. Jurnal ini menghasilkan perancangan sistem informasi inventory menggunakan metode Rapid Application Development (RAD) yang akan meningkatkan efisiensi pelaporan barang.

Penelitian berjudul “Aplikasi Sistem Persediaan Barang Gudang PT. Berkah Pena Ilmu Menggunakan Android Studio Dan Metode Rapid Application Development (RAD)”. Penelitian ini dilakukan oleh Iman Alfajri, NM Faizah dan Ryan Rakryan WP, jurnal ini berasal Universitas Jagakarsa, Kota Jakarta Selatan. Jurnal ini membahas mengenai kurangnya keakuratan dalam pencatatan serta efisiensi pelaporan barang gudang. Jurnal ini menghasilkan aplikasi mobile berbasis Android yang berguna untuk pencatatan persediaan barang.

Penelitian berjudul “Penerapan Metode RAD Untuk Pengembangan Sistem Inventori Obat Berbasis Web Menggunakan Framework Laravel (Studi Kasus Pada Klinik Sukaraya Medika Kabupaten Bekasi)”. Penelitian ini dibuat oleh Widi Anggraeni Puspitasari, Ahmad Turmudizy dan Edora, jurnal ini berasal dari Universitas Pelita Bangsa. Jurnal ini membahas mengenai masalah mitra masih menggunakan buku untuk melakukan pencatatan barang tanpa sistem informasi yang terintegrasi. Jurnal ini menghasilkan sistem informasi inventory berbasis web yang bisa menjadi solusi dari permasalahan mitra. Pengujian menggunakan metode blackbox

Penelitian berjudul “Implementasi model Rapid Application Development Pada Sistem Informasi Persediaan Barang Dengan metode FIFO” Penelitian ini dibuat oleh Iqbal Kamil Siregar, jurnal ini berasal dari sekolah tinggi manajemen informatika dan komputer royal. Jurnal ini membahas mengenai masalah mengenai bagaimana cara mengawasi barang agar tidak kadaluwarsa dan barang yang pertama dibeli akan pertama dijual. Jurnal ini menghasilkan aplikasi berbasis desktop yang bisa mengelola informasi persediaan barang menggunakan metode RAD dan FIFO (First In First Out)

Dalam pembaruan ini, penelitian memperbarui pada dua aspek utama. Pertama, terjadi pergeseran dari penggunaan *server database* lokal ke penggunaan *server database* berbasis *cloud*. peneliti menggunakan server database berbasis cloud yang menawarkan keuntungan dalam hal skalabilitas dan aksesibilitas data. Hal ini memungkinkan pengelolaan yang lebih baik tanpa mengorbankan kinerja, sambil memungkinkan akses data dari berbagai lokasi melalui koneksi internet. Selain itu, yang kedua, terjadi penyempurnaan antarmuka pengguna atau *User Interface* (UI) dengan fokus pada responsivitas tampilan. Antarmuka pengguna direvitalisasi menggunakan *framework* yang memastikan tampilan responsif, menyesuaikan diri dengan berbagai perangkat dan ukuran layar. Kombinasi dari penggunaan *server database* berbasis *cloud* dan pembaruan UI ini diharapkan akan memberikan pengalaman pengguna yang lebih lancar dan terhubung dalam aplikasi tersebut.

2.2. Landasan Teori

Adapun teori umum yang digunakan sebagai landasan dan pendukung dalam pembuatan penelitian ini, yaitu:

2.2.1. Aplikasi Mobile

Aplikasi *mobile* adalah sebuah *software* atau kumpulan program yang berjalan atau tampil di *handphone* untuk melakukan rangkaian tugas yang sudah di program, aplikasi *mobile* gampang digunakan, fleksibel serta digunakan diunduh dan dijalankan di sebagian besar *handphone*. Aplikasi *mobile* dapat melakukan berbagai macam hal seperti menelfon, *chatting*, sosial media (social media), bermain *game* dan sebagainya [15]

2.2.2. Android

Android adalah sistem operasi sumber terbuka (*open source*) dan berbasis linux yang digunakan untuk *handphone* dan tablet. Android dikembangkan oleh Aliansi Open Handset (*Open Handset Alliance*) yang di mana terdapat lebih dari 30 perusahaan teknologi dan perusahaan telfon seluler yang bekerjasama untuk membuat Android [16].

2.2.3. Profil Apotek Habibi

Apotek Habibi adalah apotek yang terletak didepan Puskesmas Bontomatene di Jl. Pahlawan – Bontosinde, Kelurahan Batangmata, Kecamatan Bontomatene, Kabupaten Kepulauan Selayar, <https://goo.gl/maps/sTnCpYZtBMxgd35E6>. Apotek ini memasok beberapa Puskesmas dan Puskesmas pembantu, Apotek Habibi juga adalah satu satunya Apotek di tiga Kecamatan sekitar Apotek.



Gambar 2.1 Palang Tanda Apotek Habibi

2.2.4. Farmasi

Farmasi adalah profesi dan ilmu didalam industri kesehatan yang bertanggung jawab dalam memastikan keamanan dan penggunaan obat yang efektif [17]. Dalam profesinya farmasi terlibat dengan pengembangan, produksi, pemasaran serta distribusi obat-obatan. Ini menunjukkan betapa pentingnya peran farmasi dalam bidang kesehatan dan bagaimana ilmu dan profesi farmasi mempunyai peran penting dalam memastikan bahwa obat-obatan yang digunakan oleh masyarakat aman dan efektif [18].

2.2.5. Apotek

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 35 Tahun 2014 Pasal 1 Nomor 1, Apotek adalah sarana pelayanan kefarmasian tempat dilakukan praktik kefarmasian oleh Apoteker [19]. Apotek adalah salah satu industri pelayanan kefarmasian di bidang kesehatan yang berfungsi untuk melayani masyarakat dengan menjual obat medikasi demi meningkatkan derajat kesehatan yang optimal [20].

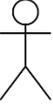
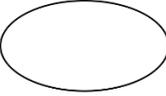
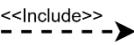
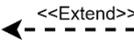
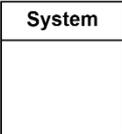
2.2.6. *Unified Modelling Language (UML)*

Bahasa Pemodelan Terpadu atau *Unified Modelling Language (UML)* adalah bahasa pemodelan visual untuk sistem perangkat lunak. Dikembangkan oleh Jim Rumbaugh, Grady Booch, dan Ivar Jacobson sebagai alat untuk pemodelan sistem berorientasi objek. UML menyediakan cara visual untuk menggambarkan, membangun, dan mendokumentasikan sistem berorientasi objek, serta memudahkan pemahaman tentang arsitektur sistem dan pengelolaan kompleksitas dari sistem yang besar [21].

Ada beberapa jenis diagram pada UML seperti Diagram *Use Case*, Diagram *Activity*, Diagram *Sequence*, Diagram *Class*, Diagram *State Machine*, Diagram *Communication*, Diagram *Deployment*, Diagram *Component*, Diagram *Object*, Diagram *Composite Structure*, Diagram *Interaction Overview*, Diagram *Package*, Diagram *Timing* [22]. Pada penelitian ini peneliti memakai dua jenis diagram yaitu diagram aktivitas (*Activity*) dan diagram kasus pengguna (*Use Case*):

1. Diagram *Use Case*

Diagram kasus pengguna menggambarkan interaksi antara aktor (orang atau sistem eksternal) dengan sistem. Aktor digambarkan sebagai gambar orang berdiri. Kasus pengguna merupakan urutan transaksi terkait yang dilakukan oleh aktor untuk mencapai fungsionalitas atau nilai tertentu. Kasus pengguna digambarkan sebagai oval dan didokumentasikan dengan urutan peristiwa, dimulai dari sudut pandang aktor. Urutan peristiwa mencakup skenario "hari-hari bahagia" yang tipikal dan skenario alternatif untuk penanganan kesalahan. Diagram kasus pengguna memberikan gambaran tingkat tinggi tentang sistem, menunjukkan apa yang berada di luar sistem (aktor) dan fungsionalitas yang harus diberikan oleh sistem (kasus pengguna) [23]. Adapun Komponen dalam diagram *use case* yang dipakai dalam penelitian ini yaitu:

	Actor	Entitas yang berinteraksi dengan sistem
	Use Case	Aktivitas yang dapat dilakukan actor pada sistem
	Association	Hubungan Antara Actor
	Include	Suatu use case termasuk bagian dari use case lain
	Extend	Suatu use case dapat diperluas dengan use case lain
	System	Sistem yang dikembangkan

Gambar 2.2 Komponen Diagram *Use Case*

Pada Gambar 2.1 Komponen Diagram *Use Case*, terdapat 6 komponen yang dipakai dalam penelitian ini [24].

a. Aktor

Bagian pertama dari diagram use case adalah aktor, yang merujuk pada entitas atau individu yang berinteraksi dengan sistem. Dalam diagram use case, aktor digambarkan sebagai manusia atau objek, bergantung pada sifat sistem yang sedang diilustrasikan. Dalam konteks nyata, aktor bisa mencakup pelanggan, administrator, dan pemilik situs web[24].

b. *Use Case*

Use case menggambarkan tindakan atau kegiatan yang dapat dilakukan oleh aktor terhadap sistem. Dalam diagram *use case*, elemen ini umumnya direpresentasikan dalam bentuk oval. Contoh-contoh dari *use case* dalam situasi nyata meliputi langkah-langkah seperti proses *login*, pendaftaran akun, dan pembayaran produk saat melakukan *checkout*[24].

c. *Association*

Asosiasi menunjukkan kaitan antara aktor dan use case tanpa harus menyertakan interaksi yang spesifik. Sebagai contoh dalam konteks e-commerce, aktor "pelanggan" memiliki asosiasi dengan use case "pencarian produk". Dalam hal ini, pelanggan memiliki kemampuan untuk mencari produk yang tersedia di platform e-commerce tersebut[24].

d. *Include*

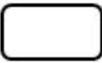
Sebuah *use case* disebut bagian (atau bagian dari) use case lain ketika itu tidak dapat beroperasi secara independen. Sebagai contoh, fitur "pilih tempat duduk" merupakan sub-bagian dari *use case* "pilih kereta" ketika seorang pelanggan menggunakan aplikasi pemesanan tiket kereta[24].

e. *Extend*

Sebuah *use case* dapat diperluas (*extended*) ke satu atau lebih *use case* lain untuk menambah fungsionalitasnya. Sebagai ilustrasi, ketika seorang pelanggan menggunakan opsi "pengembalian dana", ini bisa memicu *use case* lain yang disebut "cek status pengembalian dana" di dalam website toko online tersebut[24].

2. Diagram *Activity*

Diagram aktivitas menggambarkan alur pengendalian dalam sebuah sistem. Aktivitas digambarkan sebagai persegi panjang berbulat dan biasanya merupakan keadaan aksi yang otomatis berpindah ke keadaan berikutnya setelah aksi selesai dilakukan. Permulaan dari diagram aktivitas ditandai dengan lingkaran yang diisi. Anak panah menunjukkan perpindahan antar aktivitas, sementara garis sinkronisasi menggambarkan aktivitas yang berlangsung secara paralel. Diagram aktivitas digunakan untuk menggambarkan alur kerja bisnis, alur antar kasus pengguna (*use case*), dan alur antar metode dalam sebuah kelas. *Swimlanes* dapat digunakan untuk memisah dan menunjukkan kepemilikan serta tanggung jawab terhadap berbagai aktivitas [23]. Adapun Komponen dalam diagram *use case* yang dipakai dalam penelitian ini yaitu:

Simbol	Nama	Keterangan
	Status awal	Sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
	Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Gambar 2.3 Komponen Diagram *Activity*

Pada Gambar 2.2 Komponen Diagram *Activity*, terdapat 6 komponen yang dipakai dalam penelitian ini [24]

a. Status Awal

Ini adalah titik awal dari alur aktivitas. Ini menunjukkan di mana aktivitas dimulai. Biasanya direpresentasikan dengan simbol lingkaran yang berisi tanda panah masuk. Contoh: Dalam sebuah sistem manajemen tugas, "Pengguna masuk ke sistem" bisa menjadi titik awal aktivitas[24].

b. Aktivitas

Aktivitas mewakili tindakan atau langkah-langkah konkret yang diambil dalam alur kerja. Ini bisa berupa tugas-tugas, operasi, atau proses yang dilakukan dalam proses. Contoh: Dalam proses

pemesanan online, "Tambahkan produk ke keranjang belanja" adalah aktivitas yang mewakili tindakan yang harus dilakukan pengguna[24].

c. Percabangan

Percabangan adalah komponen yang digunakan untuk menggambarkan cabang dalam alur aktivitas berdasarkan kondisi tertentu. Ini memungkinkan untuk memilih jalur yang berbeda berdasarkan pilihan yang diambil. Contoh: Setelah menambahkan produk ke keranjang belanja, "Apakah Anda ingin melanjutkan ke *checkout*?" adalah sebuah keputusan yang mengarah pada dua jalur yang berbeda: "Ya" menuju proses checkout dan "Tidak" menuju aktivitas lanjutan seperti menambah produk lainnya[24].

d. Status Akhir

Ini adalah titik akhir dari alur aktivitas. Ini menunjukkan di mana aktivitas berakhir atau berhenti. Contoh: Dalam proses pemesanan *online*, "Pemesanan selesai" bisa menjadi titik akhir aktivitas[24].

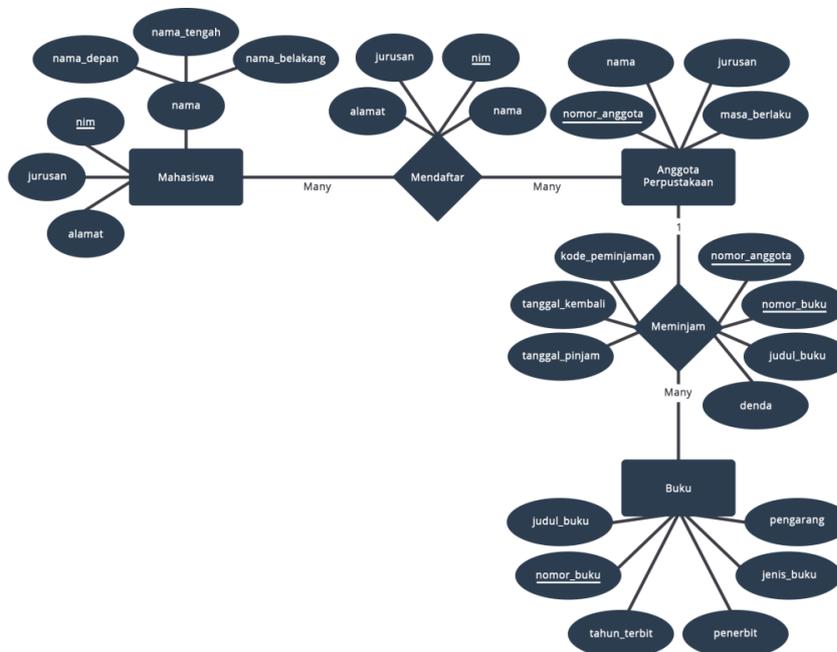
e. *Swimlane*

Swimlane adalah komponen yang digunakan untuk mengelompokkan dan membagi aktivitas dalam beberapa kelompok atau entitas yang berbeda, seperti peran atau departemen dalam organisasi. Ini membantu dalam memvisualisasikan siapa yang bertanggung jawab atas setiap aktivitas. Contoh: Dalam proses manajemen proyek, *swimlane* dapat digunakan untuk mengelompokkan aktivitas berdasarkan tim atau departemen yang bertanggung jawab, seperti "Tim Pengembangan" dan "Tim Desain"[24].

2.2.7. *Entity Relationship Diagram (ERD)*

Diagram ERD atau Diagram Hubungan Entitas adalah alat grafis yang digunakan dalam perancangan basis data untuk menggambarkan secara rinci hubungan serta keterkaitan antara berbagai objek atau entitas beserta atribut-atribut yang dimilikinya. Dengan memanfaatkan ERD, struktur sistem basis data yang

sedang dikembangkan dapat diilustrasikan dengan lebih teratur dan terorganisir [25].



Gambar 2.4 Contoh Gambar ERD Perpustakaan

Pada contoh diagram ER di atas, dijelaskan entitas, atribut, dan proses yang terlibat ketika seorang mahasiswa ingin meminjam buku dari perpustakaan. Diagram ER tersebut dapat dijadikan panduan untuk membuat sistem basis data perpustakaan[26].

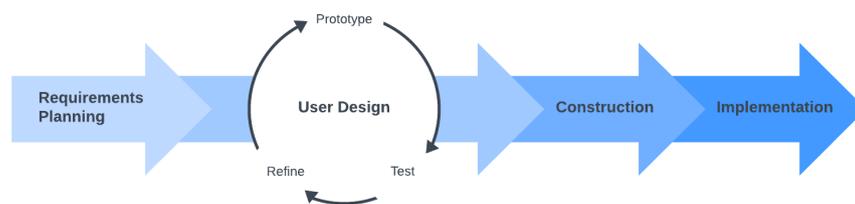
Dalam diagram perpustakaan tersebut, terdapat empat entitas. Keempat entitas tersebut adalah mahasiswa, anggota perpustakaan, buku, dan denda. Setiap entitas memiliki atributnya sendiri[26].

2.2.8. *Rapid Application Development (RAD)*

Metode *Rapid Application Development (RAD)* yang berfokus pada pengembangan cepat yang melalui iterasi atau perulangan dengan *feedback* dari pengguna. RAD diperkenalkan oleh James Martin pada tahun 1991 [27]. Keterlibatan pengguna sepanjang siklus RAD memastikan aplikasi sesuai kebutuhan dan harapan pengguna. *Prototyping* digunakan untuk membantu pengguna memvisualisasikan dan meminta perubahan pada aplikasi jika ada kesalahan.

Struktur metode RAD dirancang untuk memastikan bahwa pengembang membangun sistem yang benar-benar dibutuhkan pengguna. Siklus RAD melewati empat tahapan, termasuk semua kegiatan yang diperlukan untuk mendefinisikan, merancang, mengembangkan, dan mengimplementasikan sistem aplikasi itu mendukung kebutuhan tersebut [28].

Rapid Application Development (RAD)



Gambar 2.5 Fase dalam metode *Rapid Application Development* [29]

Gambar 2.1 di atas merupakan fase dalam metode Rapid Application development yang mempunyai 4 tahap yaitu *requirement planning*, *user design*, *construction* dan *implementation*.

a. *Requirements Planning*

Di tahap ini peneliti mendefinisikan semua kebutuhan untuk membuat aplikasi di penelitian ini, adapun kebutuhan yang dimaksud adalah kebutuhan mitra akan fitur yang diperlukan serta kebutuhan software maupun hardware. Contohnya:

1. Definisi kebutuhan mitra: Mitra dalam proyek ini adalah sebuah perusahaan penyedia jasa pengiriman barang. Mereka membutuhkan fitur pelacakan *real-time* yang memungkinkan pelanggan untuk melacak posisi paket mereka dengan akurat.
2. Kebutuhan software: Aplikasi harus dapat berjalan di platform Android, memiliki antarmuka yang *user-friendly*, dan terintegrasi dengan sistem pelacakan logistik perusahaan.
3. Kebutuhan hardware: Aplikasi harus dapat berjalan lancar pada smartphone dengan minimal spesifikasi RAM 2GB dan sistem operasi Android 7.0 ke atas.

b. *User Design*

Pada tahap ini peneliti masuk ke pengembangan prototype dari aplikasi yang di mana akan peneliti berikan kepada mitra agar mendapatkan feedback seperti fitur fitur tambahan atau penemuan kesalahan di fitur, setelah itu peneliti akan mengulangi proses ini lagi hingga pengguna merasa aplikasi cukup memuaskan. Melalui fase pengembangan prototype, peneliti dapat dengan mudah mengevaluasi kelayakan fitur dan komponen aplikasi serta meminimalisir kesalahan di fase selanjutnya. Contohnya:

1. Pengembangan *prototype*: Sebuah mockup aplikasi dengan antarmuka pelacakan paket sederhana dibuat. Pengguna dapat mengklik tombol "Lacak Paket" dan melihat tampilan status paket pada peta.
2. *Feedback* mitra: Mitra memberikan masukan bahwa mereka ingin menambahkan notifikasi push ketika status paket berubah dan meminta peningkatan dalam visualisasi peta.
3. Pengembangan ulang *prototype*: *Mockup* diperbarui dengan tambahan fitur notifikasi dan peningkatan tampilan peta.

c. *Construction*

Pada tahap ini peneliti mulai masuk ke pengembangan aplikasi serta pengujian sistem, di tahap ini peneliti mengubah prototype yang peneliti buat di fase sebelumnya menjadi model aplikasi yang berfungsi. Pada tahap ini aplikasi diuji secara menyeluruh, pengembang bekerja sama dengan pengguna untuk mengumpulkan *feedback* mengenai antarmuka dan fungsionalitas untuk memastikan hasil akhir aplikasi memuaskan. Contohnya:

1. Pengembangan aplikasi: Pengembang mulai mengimplementasikan fitur-fitur yang telah dirancang dalam fase sebelumnya. Antarmuka pelacakan paket dikembangkan menjadi fungsi yang beroperasi penuh, menghubungkan data pelacakan yang sebenarnya.

2. Pengujian sistem: Aplikasi diuji secara menyeluruh di berbagai perangkat dan kondisi jaringan untuk memastikan kinerja yang baik. Pengujian khusus dilakukan untuk memastikan notifikasi push berfungsi seperti yang diharapkan.
3. Pengumpulan *feedback*: Pengguna dari berbagai latar belakang diundang untuk menguji aplikasi. Beberapa masukan diterima mengenai keterbacaan tampilan peta pada layar kecil.

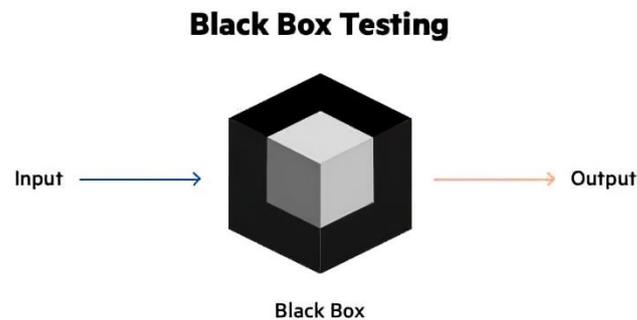
d. *Implementation*

Pada tahap ini aplikasi sudah hampir siap digunakan dan diluncurkan, semua perubahan akhir dan pengujian terakhir akan dilakukan di tahap ini sebelum aplikasi diluncurkan dan diberikan kepada pengguna. Contohnya:

1. Pembenahan akhir: Berdasarkan *feedback* dari pengujian, beberapa perbaikan kecil dilakukan pada tampilan peta dan antarmuka notifikasi.
2. Pengujian akhir: Aplikasi dijalankan melalui serangkaian pengujian akhir untuk memastikan semua fitur berjalan dengan baik dan tidak ada masalah yang kritis.
3. Peluncuran: Aplikasi diunggah Google Play Store, dan diberikan kepada pengguna. Mitra perusahaan pengiriman mengumumkan fitur pelacakan baru kepada pelanggan mereka.

2.2.9. Blackbox

Pengujian *BlackBox* melibatkan pengujian suatu sistem tanpa pengetahuan sebelumnya tentang cara kerjanya secara internal. Seorang pengujian memberikan masukan, dan mengamati keluaran yang dihasilkan oleh sistem yang sedang diuji. Ini memungkinkan untuk mengidentifikasi bagaimana sistem merespons tindakan pengguna yang diharapkan maupun yang tak terduga, waktu responsnya, masalah ketergunaan, dan masalah kehandalan[30].



Gambar 2.6 Fase *Blackbox Testing*

Pengujian dapat mensimulasikan aktivitas pengguna dan melihat apakah sistem memenuhi kinerja yang di inginkan. Dalam proses ini, pengujian *Blackbox* mengevaluasi semua subsistem yang relevan, termasuk *User Interface / User Experience* (UI/UX), *server web* atau *server* aplikasi, basis data, dependensi, dan sistem terintegrasi [30].

2.2.10. *Post-Study System Usability Questionnaire (PSSUQ)*

Pada akhir tahun 1980-an, Jim Lewis mengembangkan PSSUQ sebagai bagian dari proyek riset awal yang berfokus pada pengalaman pengguna atau *user experience* di perusahaan IBM. Kedua kuesioner ini telah menjadi populer untuk mengevaluasi persepsi tentang ketergunaan dan kemudian dipublikasikan pada tahun 1995. Seperti halnya dengan SUS, PSSUQ memiliki karakteristik teknologi-agnostik, yang artinya ia dapat diterapkan pada berbagai antarmuka, baik itu perangkat keras, perangkat lunak, aplikasi seluler, atau pun situs web [31].

PSSUQ (*Post-Study System Usability Questionnaire*) merupakan kuesioner standar yang terdiri dari 19 poin, kuesioner ini juga memiliki 3 sub-skala, yaitu *System Usefulness* atau kemanfaatan sistem, *Information Quality* atau kualitas informasi, dan *Interface Quality* atau kualitas antarmuka. Kuesioner ini umumnya dimanfaatkan untuk mengukur tingkat kepuasan yang dirasakan oleh para pengguna terhadap suatu situs web, perangkat lunak, sistem, atau produk setelah selesainya suatu studi [32]. Kuesioner ini memanfaatkan skala respons tujuh poin, di mana nilai yang lebih kecil mengindikasikan kinerja yang lebih unggul. (Kuesioner ini

juga menyertakan pilihan N/A jika responden merasa tidak dapat menjawab item tersebut.) [31]. Cara perhitungan PSSUQ dengan merata-ratakan skor dari item-item yang sesuai untuk mendapatkan skor skala dan sub-skala dimana nilai 1 paling bagus dan 7 paling buruk dan 4 tidak bagus namun tidak buruk [33].

- *Overall*: the average scores of questions 1 to 19
- *System Usefulness (SYSUSE)*: rata-rata skor dari pertanyaan 1 to 8
- *Information Quality (INFOQUAL)*: rata-rata skor dari pertanyaan 9 to 15
- *Interface Quality (INTERQUAL)*: rata-rata skor dari pertanyaan 16 to 19

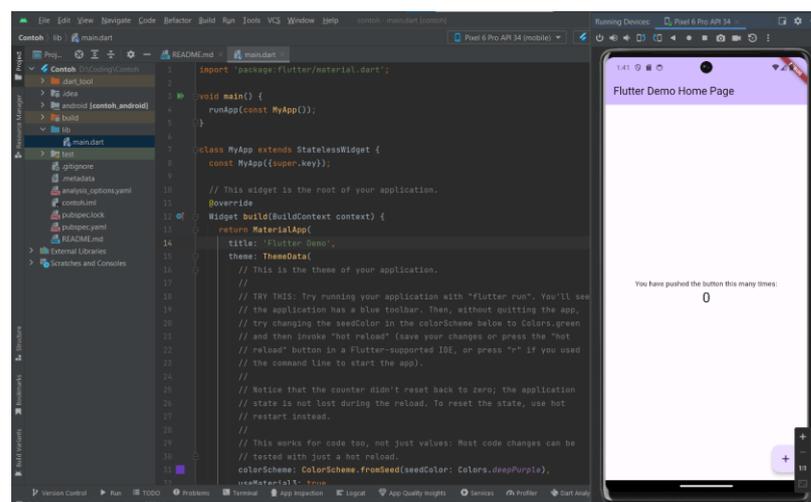
Adapun pertanyaan dalam kuisioner PSSUQ yang akan dipakai dalam penelitian ini yaitu :

1. Secara keseluruhan, saya puas dengan seberapa mudahnya menggunakan aplikasi ini.
2. Sangat mudah menggunakan aplikasi ini.
3. Saya dapat menyelesaikan pekerjaan dengan efektif menggunakan aplikasi ini.
4. Saya dapat menyelesaikan pekerjaan dengan cepat menggunakan aplikasi ini.
5. Saya dapat menyelesaikan pekerjaan dengan efisien menggunakan aplikasi ini.
6. Saya merasa nyaman menggunakan aplikasi ini.
7. Sangat mudah untuk belajar menggunakan aplikasi ini.
8. Saya yakin saya dapat menjadi produktif dengan cepat menggunakan aplikasi ini.
9. Aplikasi memberikan pesan kesalahan yang jelas tentang bagaimana memperbaiki masalah.
10. Setiap kali saya melakukan kesalahan menggunakan aplikasi ini, saya dapat pulih dengan mudah dan cepat.
11. Informasi (seperti bantuan daring, pesan di layar, dan dokumentasi lainnya) yang disediakan dengan aplikasi ini sangat jelas.
12. Sangat mudah menemukan informasi yang saya butuhkan.

13. Informasi yang disediakan untuk aplikasi ini mudah dipahami.
14. Informasi tersebut efektif dalam membantu saya menyelesaikan pekerjaan.
15. Penyusunan informasi pada layar sistem sangat jelas.
16. Antarmuka dari aplikasi ini menyenangkan.
17. Saya suka menggunakan antarmuka dari aplikasi ini.
18. Aplikasi ini memiliki semua fungsi dan kemampuan yang saya harapkan.
19. Secara keseluruhan, saya puas dengan aplikasi ini [33].

2.2.11. Flutter

Flutter adalah *toolkit* antarmuka yang dirancang untuk lintas platform di mana ini akan memungkinkan kode untuk digunakan ulang pada sistem operasi lain seperti IOS dan Android [34], Flutter bersifat *open source* yang dikembangkan oleh google untuk membuat aplikasi android dan IOS [35]. Beberapa kelebihan flutter yang paling mencolok adalah Flutter hanya memerlukan satu *codebase* untuk membuat aplikasi yang mempunyai sistem operasi yang berbeda, Flutter juga mempunyai performa yang tinggi, banyaknya dokumentasi dan bahan pembelajaran dari komunitas Flutter. Flutter mempunyai beberapa fitur seperti fitur *hot reload* yang di mana jika peneliti menemukan *bug* saat menguji aplikasi peneliti bisa memperbaiki *bug* tersebut tanpa harus mengulang pembuatan aplikasi, fitur lainnya adalah Flutter mempunyai berbagai macam *widget* yang bisa dipakai untuk membuat aplikasi lebih mudah [36].

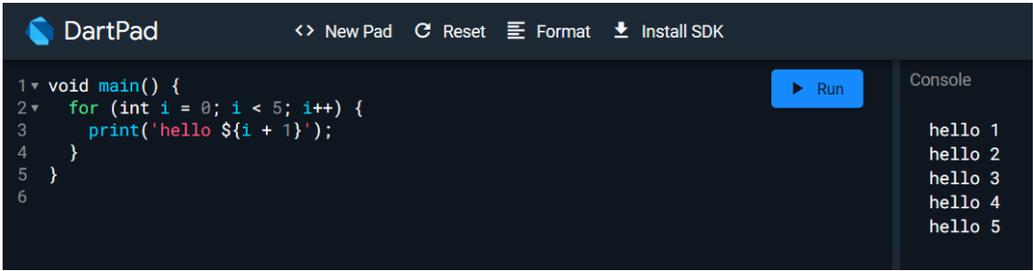


Gambar 2.7 Simulasi Flutter

Gambar 2.7 adalah contoh aplikasi Flutter yang dapat dicoba saat menggunakan Flutter

2.2.12. Dart

Dart adalah bahasa pemrograman sumber terbuka, terstruktur, dan fleksibel yang dikembangkan oleh Google, khususnya ditujukan untuk pengembangan web, namun tidak secara eksklusif. Bahasa pemrograman baru ini hadir untuk mempermudah kehidupan para programmer dengan memungkinkan mereka mengembangkan aplikasi web yang lebih kompleks dengan pemeliharaan yang lebih baik dan performa yang lebih baik [37].



The image shows the DartPad web editor interface. At the top, there are navigation buttons: '<> New Pad', 'Reset', 'Format', and 'Install SDK'. The main area contains a code editor with the following Dart code:

```

1 void main() {
2   for (int i = 0; i < 5; i++) {
3     print('hello ${i + 1}');
4   }
5 }
6

```

To the right of the code editor is a 'Run' button. Below the code editor is a 'Console' panel displaying the output of the program:

```

hello 1
hello 2
hello 3
hello 4
hello 5

```

Gambar 2.8 Contoh Program Dasar Dart

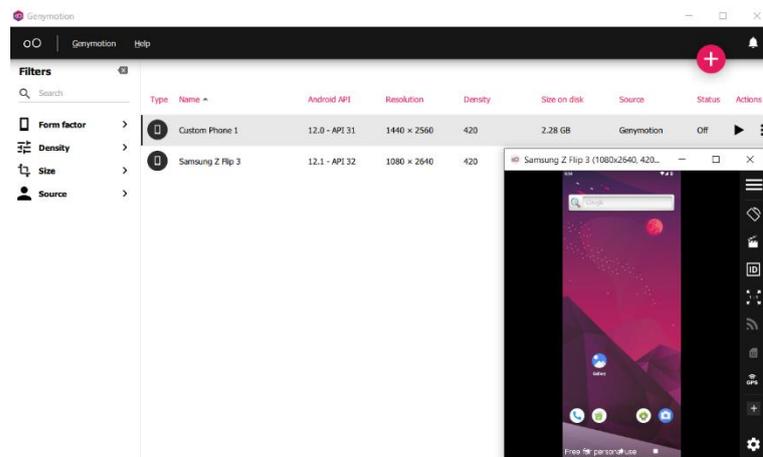
2.2.13. Firebase

Firebase adalah layanan BaaS (Backend as a Service) yang dikembangkan oleh Google untuk memudahkan pengembang aplikasi *mobile* yang di mana peneliti bisa menghubungkan aplikasi dengan layanan penyimpanan *cloud* [38]. Firebase dipilih karena beberapa fitur yang ditawarkan yaitu:

1. Firebase menyediakan penyimpanan gratis hingga 5GB penyimpanan *cloud* untuk media seperti gambar, video, dan lain lain.
2. *Cloud Firestore* yang di mana *database* dapat digunakan secara *offline*, data yang peneliti buat secara lokal akan di sinkronisasikan saat terhubung lagi dengan *database*. *Cloud Firestore* juga menyediakan penyimpanan *database* gratis hingga 1GiB.
3. Firebase menyediakan autentikasi demi keamanan yang bisa peneliti lakukan melalui *handphone* [39].

2.2.14. Genymotion

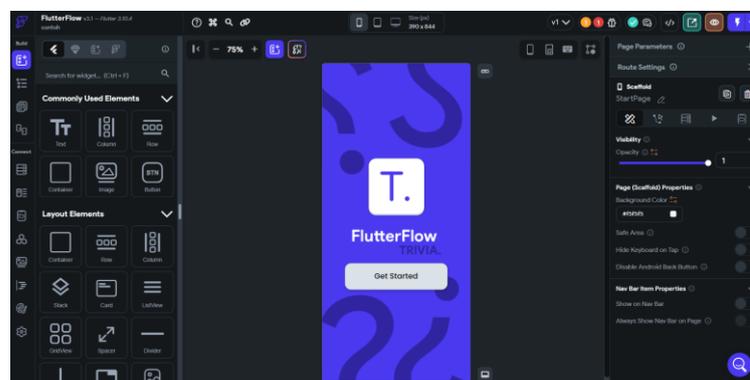
Genymotion adalah emulator android yang didirikan oleh Arnaud Dupuis pada tahun 2011, Genymotion dibuat untuk melayani developer yang ingin mengembangkan aplikasi android tanpa harus membeli perangkat asli untuk menguji aplikasi mereka. Kelebihan dari Genymotion antara lain adalah banyaknya pilihan perangkat untuk menguji aplikasi, untuk pemakaian pribadi layanan gratis serta performa yang tinggi [40].



Gambar 2.9 Salah Satu Emulator Genymotion

2.2.15. FlutterFlow

FlutterFlow adalah pembuat aplikasi berbasis web yang dapat diakses melalui browser yang memungkinkan pengguna untuk membuat aplikasi native lintas platform. FlutterFlow merupakan alat pembuat aplikasi visual pihak ketiga untuk kerangka kerja Flutter, yang memungkinkan pembuat aplikasi untuk mempercepat proses pembuatan aplikasi secara signifikan [41].



Gambar 2.10 Salah Satu Template Di FlutterFlow