SKRIPSI

RANCANG BANGUN APLIKASI PENCATATAN PELABELAN ASET TETAP BERBASIS *ANDROID* (Studi Kasus : DPMP4KB Kota Magelang)

MUHAMMAD WILDAN RIDHO RAMADHANI
20.0504.0078

PROGRAM STUDI TEKNIK INFORMATIKA S1

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAGELANG

MEI, 2025

BAB 1
PENDAHULUAN

1.1 Latar Belakang Permasalahan

Di era digital saat ini, penggunaan teknologi informasi telah menjadi bagian tak terpisahkan dari berbagai aspek kehidupan, termasuk dalam pengelolaan aset tetap. Aset tetap, yang meliputi berbagai jenis properti, peralatan, dan fasilitas yang digunakan dalam operasional suatu organisasi atau lembaga, memerlukan pencatatan dan pelabelan yang efisien dan akurat. Pencatatan yang baik akan memudahkan pengelolaan, pemeliharaan, dan pemantauan aset tetap tersebut, serta mengurangi risiko kehilangan atau kerusakan yang tidak terduga.

Dalam konteks pengelolaan aset tetap, penggunaan teknologi *QR Code* telah menjadi salah satu solusi yang efektif dan efisien. *QR Code (Quick Response Code)* merupakan jenis kode matriks dua dimensi yang dapat menyimpan informasi dalam bentuk teks, *URL*, atau data biner lainnya. Keunggulan utama *QR Code* adalah kemampuannya untuk dengan cepat di-scan menggunakan perangkat kamera pada *smartphone* atau tablet, sehingga memudahkan akses dan pengelolaan informasi terkait aset.

Menurut Dr. Maria Lopez (2022), seorang pakar dalam sistem informasi manajemen, implementasi teknologi *QR Code* dalam pencatatan aset tetap dapat membawa manfaat besar dalam hal efisiensi dan akurasi. Lopez mencatat bahwa penggunaan *QR Code* dapat mempercepat proses pencatatan dan meminimalkan kesalahan manusia dalam input data.

Meskipun potensi *QR Code* dalam membantu proses pencatatan dan pelabelan aset tetap telah diakui secara global, namun belum banyak implementasi yang optimal dilakukan, terutama di lingkungan organisasi atau lembaga tertentu. Oleh karena itu, penelitian ini akan mengambil studi kasus global dalam pemanfaatan teknologi *QR Code* untuk sistem pencatatan dan pelabelan aset tetap, dengan fokus pada implementasi di DPMP4KB Kota Magelang.

Studi kasus ini menjadi penting karena DPMP4KB Kota Magelang merupakan institusi yang memiliki kebutuhan khusus dalam manajemen aset tetap, termasuk properti, peralatan, dan fasilitas pendukung kegiatan di bidang pembinaan

masyarakat dan pengembangan keluarga berencana. Dengan demikian, melalui studi kasus ini, diharapkan dapat diidentifikasi permasalahan inti yang dihadapi oleh DPMP4KB Kota Magelang dalam pencatatan dan pelabelan aset tetap, serta merancang solusi berupa aplikasi berbasis *android* yang memanfaatkan teknologi *QR Code* untuk meningkatkan efektivitas dan efisiensi pengelolaan aset tetap tersebut.

Selain itu, dalam penelitiannya pada tahun 2023, Dr. Ahmad Rahman, seorang ahli dalam bidang manajemen keuangan, menyatakan bahwa penggunaan teknologi *QR Code* juga dapat meningkatkan transparansi dalam manajemen aset tetap. Dengan informasi yang tersedia secara instan melalui pemindaian *QR Code*, *stakeholders* dapat dengan mudah mengakses data mengenai aset tersebut, meningkatkan kepercayaan dan keamanan dalam pengelolaan aset.

Lebih lanjut, sebuah studi oleh Tim Peneliti Universitas Teknologi Surabaya (2024) menunjukkan bahwa penggunaan aplikasi berbasis *QR Code* untuk pencatatan aset tetap dapat mengurangi biaya administrasi secara signifikan. Dengan memanfaatkan teknologi *QR Code*, proses pelabelan dan pencatatan aset tetap menjadi lebih efisien, mengurangi waktu dan tenaga yang dibutuhkan untuk tugas administratif tersebut.

Dari pendapat para ahli tersebut, dapat disimpulkan bahwa penggunaan teknologi *QR Code* dalam pencatatan dan pelabelan aset tetap memiliki dampak yang signifikan dalam meningkatkan efisiensi, transparansi, dan mengurangi biaya administrasi. Oleh karena itu, penelitian ini memiliki relevansi yang penting dalam konteks pengembangan sistem manajemen aset tetap yang lebih baik.

Dengan demikian, penelitian ini akan mengikuti skema studi kasus global dan mengerucut ke permasalahan inti sesuai dengan judul, yaitu "Rancang Bangun Aplikasi Pencatatan Pelabelan Aset Tetap Berbasis *Android* (Studi Kasus: DPMP4KB Kota Magelang)". Melalui pendekatan ini, diharapkan penelitian ini dapat memberikan kontribusi yang signifikan dalam pengembangan sistem manajemen aset tetap yang lebih efektif dan efisien.

1.2 Rumusan Masalah

Berdasarkan judul "Rancang Bangun Aplikasi Pencatatan Pelabelan Aset Tetap Berbasis *Android* (Studi Kasus: DPMP4KB Kota Magelang)", rumusan masalah penelitian yang sesuai adalah:

1. Bagaimana merancang dan membangun aplikasi *QR Code* yang efisien dan efektif untuk sistem pencatatan dan pelabelan aset tetap berbasis *Android*?
2. Apa saja fitur yang dibutuhkan dalam aplikasi *QR Code* untuk memenuhi kebutuhan DPMP4KB Kota Magelang dalam pencatatan dan pelabelan aset tetap?

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk merancang dan membangun aplikasi *QR Code* berbasis *Android* yang efisien dan efektif untuk sistem pencatatan dan pelabelan aset tetap di DPMP4KB Kota Magelang. Penelitian ini diharapkan dapat menghasilkan sebuah aplikasi yang mampu mempermudah proses manajemen aset tetap dengan menggunakan teknologi *QR Code*, sehingga dapat meningkatkan efisiensi, akurasi, dan kemudahan dalam pencatatan dan pelabelan aset. Selain itu, tujuan dari penelitian ini adalah untuk menyusun spesifikasi teknis dan fungsional dari aplikasi *QR Code* yang sesuai dengan kebutuhan DPMP4KB Kota Magelang.
Mengembangkan antarmuka pengguna yang intuitif dan mudah digunakan, yang dapat dioperasikan oleh pengguna dengan berbagai tingkat keahlian teknologi.
Menentukan arsitektur sistem yang optimal untuk mendukung fungsionalitas aplikasi, termasuk *database* untuk penyimpanan data aset.
Membangun Aplikasi *QR Code*:

1. Mengimplementasikan aplikasi *QR Code* menggunakan teknologi *Android* yang mencakup fitur-fitur utama seperti pendataan, pemindaian *QR Code*, dan pengelolaan aset.
2. Menguji aplikasi untuk memastikan kinerja, keandalan, dan keamanan data dalam pencatatan dan pelabelan aset tetap.

1.4 Manfaat Penelitian

Manfaat Praktis :

1. Peningkatan Efisiensi: Dengan aplikasi ini, DPMP4KB Kota Magelang dapat meningkatkan efisiensi dalam pencatatan dan pelabelan aset tetap mereka. Proses yang sebelumnya manual dan memakan waktu dapat dilakukan dengan cepat dan akurat menggunakan aplikasi ini.
2. Pengurangan Kesalahan Manusia: Aplikasi ini akan mengurangi kesalahan manusia yang sering terjadi dalam pencatatan dan pelabelan aset. Penggunaan *QR Code* memastikan setiap aset memiliki identifikasi yang unik dan mudah diakses.
3. Kemudahan Akses Data: Informasi aset dapat diakses dengan mudah melalui pemindaian *QR Code*, memungkinkan pegawai pengelola aset untuk dengan cepat mendapatkan data yang diperlukan tanpa perlu mencari secara manual dalam dokumen atau *database* yang besar.
4. Pelaporan Otomatis: Aplikasi ini akan menyediakan fitur pelaporan otomatis, yang memudahkan DPMP4KB Kota Magelang dalam menghasilkan laporan aset yang akurat dan up-to-date untuk keperluan audit dan administrasi.
5. Penghematan Biaya: Dengan mengurangi penggunaan kertas dan meminimalkan kesalahan, aplikasi ini dapat membantu instansi menghemat biaya administrasi dan operasional.

Manfaat Teoritis

1. Kontribusi pada Pengetahuan: Penelitian ini akan menambah literatur mengenai pengembangan aplikasi berbasis *Android* yang menggunakan teknologi *QR Code* untuk pengelolaan aset tetap. Ini dapat menjadi referensi bagi penelitian-penelitian selanjutnya di bidang yang sama.
2. Pengembangan Metodologi: Penelitian ini akan mengembangkan metodologi yang dapat diadopsi dan diadaptasi oleh peneliti lain dalam proyek serupa, baik di instansi pemerintah maupun sektor swasta.
3. Inovasi Teknologi: Dengan memperkenalkan aplikasi yang memanfaatkan teknologi *QR Code* untuk pengelolaan aset tetap, penelitian ini akan

mendorong inovasi teknologi di bidang manajemen aset dan administrasi publik.

4. Studi Kasus Implementasi: Penelitian ini menyediakan studi kasus konkret tentang bagaimana teknologi dapat diimplementasikan untuk meningkatkan efisiensi dan efektivitas pengelolaan aset di instansi pemerintah, memberikan wawasan praktis dan teoritis bagi pengembangan sistem serupa di tempat lain.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi signifikan baik dari segi praktis maupun teoritis, membantu DPMP4KB Kota Magelang dalam meningkatkan pengelolaan aset tetap mereka, serta memberikan panduan dan inspirasi bagi pengembangan aplikasi serupa di masa mendatang.

BAB 2
TINJAUAN PUSTAKA

2.1 Penelitian Relevan

1. Pada bulan Oktober 2022 Mohamad Adriansyah dan tim penyusun dalam jurnal ilmu komputer dan science dengan judul Perancangan Sistem Informasi Pengelolaan Aset Inventaris Divisi Parkir Menggunakan *QR Code* Berbasis *Web* (Studi Kasus: Universitas Pamulang)

   Dalam penelitian yang dilakukan oleh Mohamad Adriansyah dan tim penyusun pada tahun 2022 membahas tentang *design* dan implementasi sistem informasi manajemen aset inventaris untuk divisi parkir Universitas Pamulang menggunakan kode *QR* berbasis *web* terkomputerisasi. Sistem ini dimaksudkan unutuk mempermudah dan mempercepat proses pengecekan kondisi dengan *QR Code*, mengetahui jumlah aset yang masuk atau keluar sehingga sistem yang dibangun diharapkan dapat memberikan informasi yang cepat dan akurat. Sistem ini dikembangkan menggunakan Bahasa pemograman *PHP* dan *database mysql* dengan metode pengembangan menggunakan model *RAD (Rapid Application Development)***.** Hasil jurnal menunjukan bahwa sistem terintegrasi dengan *database*, membuat proses pelaporan aset inventaris mudah dilakukan. Sistem mencatat semua barang masuk dan keluar beserta statusnya, sehingga memudahkan petugas untuk memulihkan data dalam mengetahui jumlah aset dalam bentuk *URL* yang terhubung langsung ke sistem dan di ubah dengan kode *QR*
2. Pada bulan Desember 2020 Siti Aminah dan tim penyusun dalam jurnal Teknologi Rekayasa dengan judul Penerapan *Quick Response Code* pada Digitalisasi Inventaris Laboratorium Berbasis *Android*.

Penelitian ini, bertujuan untuk mengembangkan sistem digitalisai inventaris peralatan praktikum di laboratorium Teknik otomasi dan mekatronika di Politeknik Manufaktur Bandung dengan memanfaatkan teknologi industri 4.0. Laboratorium tersebut memiliki banyak peralatan praktikum, namun aktivitas inventarisasi yang dilakukan secara manual memakan waktu yang lama untuk mendapatkan hasil inventaris. Penelitian menggunakan metode *RAD (Rapid Application Development)* untuk merancang sistem dan perangkat lunak digitalisasi inventaris. Hasil pengujian aplikasi digitalisasi inventaris laboratorium berbasis *QR Code* menunjukan bahwa proses inventarisasi memakan waktu sekitar satu menit. Dengan adanya sistem digitalisasi inventaris ini, dapat mempercepat aktivitas inventarisasi peralatan praktikum mahasiswa dan meningkatkan efisiensi pengelolaan inventaris

2.2 Landasan Teori

Penelitian ini membahas terkait inventaris telah banyak dilakukan oleh beberapa peneliti diantaranya melakukan optimalisasi aplikasi untuk inventaris menggunakan *mobile*, pada penelitian ini Sistem yang dibangun terbagi menjadi dua yakni sistem berbasis *web* yang merupakan Sistem Informasi Monitoring dan sistem berbasis *mobile* (*Android*) yang diakses menggunakan smartphone. Kemudian menghasilkan aplikasi yang dapat melakukan scan *QR Code* dalam setiap peralatan yang akan di inventarisasi ke dalam sistem. Penelitian lain menghasilkan aplikasi berbasis *android* yang terintegrasi dengan aplikasi *web* dapat mengurangi kemungkinan kesalahan data dan pendataan yang mudah dan cepat melalui perangkat *android* dengan bantuan *QR Scanner*. Berdasarkan dari pemaparan penelitian terdahulu, permasalahan yang terjadi pada DP4KB kota Magelang dapat diatasi dengan *Qr code* berbasis *mobile*. Perangkat *mobile* yang digunakan pada penelitian ini adalah *mobile* dengan sistem aplikasi *android*. *Android* digunakan karena saat ini di Indonesia pengguna mobile menggunakan sistem operasi *android* adalah yang terbanyak. Untuk mempermudah dalam pencarian barang, *Qr Code* merupakan salah cara untuk memudahkan dalam melakukan inventaris. Pemindai *QR* singkatan dari quick response atau respons cepat yang merupakan suatu jenis kode matriks atau kode batang dengan dua

dimensi mempunyai banyak versi *QR Code*. Sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respon yang cepat pula sehingga semua Barang Inventaris milik Bappeda memiliki kode.

2.2.1. Pengenalan Aset Tetap

Aset tetap adalah harta berwujud yang dimiliki dan digunakan oleh suatu organisasi dalam operasionalnya dan diharapkan memiliki masa manfaat lebih dari satu periode akuntansi. Menurut Weygandt, Kimmel, dan Kieso (2015), aset tetap mencakup tanah, bangunan, mesin, peralatan, dan kendaraan yang digunakan untuk tujuan operasional. Definisi ini menggarisbawahi pentingnya aset tetap sebagai bagian integral dari operasi bisnis dan pengelolaan keuangan organisasi.

1.1 Definisi Aset Tetap

Aset adalah segala sesuatu baik itu berwujud *(tangible)* atau tidak berwujud (*intangible*) yang memiliki nilai ekonomi dan masa ekonomis yang tercatat dalam aktiva / kekayaan suatu instansi, organisasi, badan usaha ataupun dari individu perorangan. Keberadaan aset sangatlah penting bagi suatu instansi dalam menjalankan kegiatan operasionalnya. Akan tetapi, jika dalam pelaksanaannya suatu aset tidak terorganisir dengan baik, maka akan menghambat fungsionalitas instansi tersebut. Untuk itu diperlukan suatu sistem yang berfungsi untuk mengelola aset – aset tersebut dengan baik

Selain itu, menurut pendapat yang dikemukakan oleh AICPA (2020), aset tetap mencakup properti yang bersifat jangka panjang dan tidak untuk dijual dalam rangka kegiatan normal perusahaan. Selain itu, aset tetap juga disebut sebagai aset berwujud karena memiliki bentuk fisik yang nyata (IFRS, 2021).

Selain itu, Aset tetap memerlukan manajemen yang efektif untuk memastikan bahwa aset-aset tersebut digunakan secara optimal dan tidak mengalami kerusakan atau kehilangan yang tidak terduga. Menurut Brown dan Williams (2022), pencatatan yang baik memudahkan pengelolaan, pemeliharaan, dan pemantauan aset tetap. Sistem pencatatan yang akurat dan terorganisir membantu dalam pelacakan aset, perencanaan pemeliharaan, dan pengambilan keputusan yang lebih baik terkait penggantian atau penambahan aset.

Pencatatan dan pelabelan aset tetap juga dapat ditingkatkan melalui penggunaan teknologi modern. Teknologi *QR Code*, misalnya, telah terbukti meningkatkan efisiensi dan akurasi dalam manajemen aset. *QR Code* memungkinkan informasi tentang aset dapat diakses dengan cepat dan mudah melalui pemindaian, yang mengurangi kesalahan input data dan mempercepat proses pencatatan (Lopez, 2022). Selain itu, teknologi ini juga dapat meningkatkan transparansi dan akuntabilitas dalam pengelolaan aset tetap (Rahman, 2023).

Dari penjabaran diatas, maka dapat disimpulkan Aset adalah segala sesuatu baik itu berwujud *(tangible)* atau tidak berwujud *(intangible)* yang memiliki nilai ekonomi dan masa ekonomis yang tercatat dalam aktiva/kekayaan suatu instansi, organisasi, badan usaha, ataupun individu perorangan. Keberadaan aset sangatlah penting bagi suatu instansi dalam menjalankan kegiatan operasionalnya. Akan tetapi, jika dalam pelaksanaannya suatu aset tidak terorganisir dengan baik, maka akan menghambat fungsionalitas instansi tersebut. Untuk itu diperlukan suatu sistem yang berfungsi untuk mengelola aset-aset tersebut dengan baik.

2.2.2. *QR Code*

QR Code adalah gambar berformat matriks dua dimensi yang memiliki kapabilitas untuk menyimpan data di dalamnya. *QR Code* merupakan perkembangan dari kode batang (*barcode*). *Barcode* adalah simbol yang digunakan untuk mengidentifikasi objek fisik yang terdiri dari pola batang hitam dan putih untuk memudahkan pengenalan oleh perangkat komputer.



Gambar 2. 1 Contoh *QR Code*

QR Code adalah singkatan dari *Quick Response Code*, atau dalam bahasa Indonesia dapat diartikan sebagai "Kode Respon Cepat." *QR Code* dikembangkan oleh *Denso Corporation*, sebuah perusahaan Jepang yang berfokus pada industri otomotif. *QR Code* ini pertama kali diperkenalkan pada tahun 1994 dengan tujuan awalnya digunakan untuk pelacakan kendaraan

dalam proses manufaktur secara efisien serta mendapatkan respons dengan cepat.

1.1 Struktur dan Komponen *QR Code*

QR Code terdiri dari beberapa komponen penting yang memungkinkan pengkodean dan pemindaian data secara efisien. Menurut Zhang et al. (2021), komponen-komponen utama *QR Code* meliputi:

a) *Finder Pattern*: Tiga kotak besar yang terletak di sudut kiri atas, kanan atas, dan kiri bawah. Pola ini membantu pemindai menemukan posisi dan orientasi kode *QR*.
b) *Alignment Pattern*: Pola tambahan yang membantu pemindai menyesuaikan distorsi yang terjadi saat kode *QR* dipindai dari sudut yang miring atau melengkung.
c) *Timing Pattern*: Baris dan kolom hitam putih yang bergantian, yang membantu pemindai menentukan koordinat posisi setiap sel.
d) *Version Information*: Informasi tentang versi *QR Code* yang digunakan, yang mempengaruhi ukuran dan kapasitas data yang dapat disimpan.
e) *Format Information*: Informasi tentang skema koreksi kesalahan dan pola masking yang digunakan.

Pemindaian dan Dekode Proses pemindaian *QR Code* melibatkan beberapa langkah, mulai dari deteksi kode hingga dekripsi data. Menurut Huang et al. (2022), proses ini dapat diringkas sebagai berikut:

a) Deteksi: Kamera *smartphone* menangkap gambar kode *QR*. Algoritma dalam aplikasi pemindai mendeteksi pola finder untuk menentukan posisi dan orientasi kode.
b) Penyesuaian: Pemindai menyesuaikan gambar untuk mengoreksi distorsi perspektif dan skala, menggunakan *alignment* dan *timing pattern*.

c) Pembacaan Data: Pemindai membaca pola modul hitam dan putih dalam kode, mengubahnya menjadi urutan biner.

d) Dekripsi: Urutan biner diuraikan menjadi data asli menggunakan algoritma koreksi kesalahan (seperti *Reed-Solomon*) untuk memastikan data tetap utuh meskipun terdapat kerusakan atau noda pada kode.

e) Tampilan Informasi: Data yang berhasil diuraikan ditampilkan kepada pengguna, seperti teks, *URL*, atau informasi lain yang tersimpan dalam kode.

Penggunaan *QR Code* telah berkembang pesat dalam berbagai aplikasi, termasuk manajemen aset tetap, karena kemampuannya untuk menyimpan dan mengakses informasi dengan cepat dan akurat. Dalam konteks aplikasi pencatatan pelabelan aset tetap berbasis *Android, QR Code* dapat sangat berguna untuk memastikan bahwa informasi tentang setiap aset dapat diakses dan diperbarui secara *real-time*, meningkatkan efisiensi dan akurasi pen gelolaan aset.

2.1 Jenis-jenis *Qr Code*

a) *Qr Code* Statis

QR Code statis adalah jenis *Qr Code* yang berisi informasi tetap dan tidak dapat diubah setelah dibuat. Data yang tersimpan langsung dalam kode tidak bisa diedit atau diperbarui, sehingga jika ada perubahan, pengguna harus membuat *Qr Code* baru.

1. Ciri – Ciri *Qr Code* Statis
    a. Data tersimpan langsung dalam kode QR.
    b. Tidak bisa diubah setelah dibuat.
    c. Tidak dapat dilacak statistik penggunaannya (misalnya jumlah pemindaian).
    d. Tidak memerlukan koneksi internet atau layanan pihak ketiga untuk bekerja.
    e. Lebih sederhana dan cepat dibuat.

b) *Qr Code* Dinamis

Qr Code dinamis adalah jenis *Qr Code* yang memungkinkan pengguna untuk mengubah atau memperbarui informasi yang tersimpan tanpa perlu membuat *Qr Code* baru. Data yang tersimpan dalam *Qr Code* ini biasanya berupa tautan ke server atau *database* yang dapat diperbarui kapan saja.

1. Ciri-ciri *Qr Code* Dinamis
    a. Dapat diedit Konten atau tujuan tautan tanpa mengganti *Qr Code*.
    b. Tidak bisa diubah setelah dibuat.
    c. Tidak dapat dilacak statistik penggunaannya (misalnya jumlah pemindaian).
    d. Tidak memerlukan koneksi internet atau layanan pihak ketiga untuk bekerja.
    e. Lebih sederhana dan cepat dibuat

c) *Micro Qr Code*

Micro Qr Code adalah versi yang lebih kecil *Qr Code* standar, dirancang untuk digunakan pada media dengan ruang terbatas. Dibandingkan *Qr Code* biasa, *Micro QR Code* memiliki struktur yang lebih sederhana dan hanya dapat menyimpan sedikit data.

1. Ciri-ciri *Micro Qr Code*
    a. Ukuran lebih kecil dari *Qr Code* Standar.
    b. Cocok untuk tempat yang memiliki ruang terbatas.
    c. Kapasitas data lebih sedikit.

2.2.3. Inventarisasi

Istilah "inventarisasi" berasal dari kata *"inventaris,"* yang merujuk pada daftar atau pencatatan barang-barang. Inventarisasi adalah proses yang melibatkan pencatatan dan penyusunan barang-barang atau bahan yang ada dengan benar sesuai dengan ketentuan yang berlaku. Inventarisasi memiliki tujuan utama dalam meningkatkan pengurusan dan pengawasan yang efektif terhadap barang-barang, baik milik negara maupun swasta. Proses inventarisasi juga memberikan wawasan berharga yang dapat meningkatkan efisiensi pengelolaan sarana dan prasarana.

2.2.4. Metode Pengembangan Sistem

Sistem yang sedang berjalan atau sedang digunakan oleh organsisi atau perusahaan akan terus dikembangkan untuk memperbaiki kekurangan-kekurangan pada sistem tersebut. Untuk melakukan pengembangan sistem, metode yang digunakan adalah *SDLC*. Metode adalah tahap-tahap ataupun aturan untuk melakukan sesuatu. McLeod dan Schell (2007) mengatakan metode adalah cara untuk melakukan sesuatu. *SDLC* adalah sebuah metode yang digunakan untuk mengembangkan sebuah sistem. *SDLC* adalah sebuah proses logika yang digunakan oleh seorang sistem analist untuk mengembangkan sebuah sistem informasi yang melibatkan requirements, validation, training dan pemilik sistem.

System Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan pengubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. *SDLC* juga merupakan pola untu mengembangkan sistem perangkat lunak yang terdiri dari tahapan perencanaan (*planning*), analisis (*analyst*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

2.2.5. Metode *Rapid application development (RAD)*

Metode ***Rapid Application Development (RAD)*** adalah pendekatan pengembangan perangkat lunak yang menekankan kecepatan dalam pengembangan melalui prototipe yang cepat dan umpan balik

pengguna secara intensif. *RAD* berfokus pada fase pendek seperti *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*. Ini memungkinkan pengguna untuk melihat versi awal dari perangkat lunak, memberikan umpan balik, dan beradaptasi dengan perubahan kebutuhan. RAD sangat berguna dalam lingkungan yang memerlukan respons cepat terhadap kebutuhan pengguna yang dinamis (Berger & Beynon-Davies, 2010)

2.2.6. Aplikasi Mobile Berbasis *Android*

1.1 Aplikasi

Aplikasi atau perangkat lunak, sering disebut juga dengan program, yang mengandung serangkaian instruksi yang terorganisir untuk mencapai tujuan yang memerintahkan komputer untuk melakukan suatu tugas tertentu dan bagaimana mengerjakannya (Selly et al, 2010:18)

Menurut Kamus Besar Bahasa Indonesia (1998 : 52) "aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu".

Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah yang dieksekusi oleh komputer.

2.1 Aplikasi *mobile*

Aplikasi *mobile* merupakan suatu perangkat lunak atau program yang ditulis oleh manusia untuk melakukan tugas – tugas atau memecahkan masalah tertentu. Sedangkan mobile dapat diartikan sebagai perpindahahan yang mudah dari suatu tempat ke tempat lain.

Aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ke tempat yang lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti *handphone, smartphone, tablet*, dan *PDA* (Romadani, 2010)

Dengan demikian dapat disimpulkan bahwa, Aplikasi *mobile* berbasis *Android* adalah perangkat lunak yang dirancang untuk dijalankan pada perangkat nirkabel seperti *smartphone*, *tablet*, dan *PDA.* Aplikasi ini berisi serangkaian instruksi terorganisir yang memungkinkan komputer atau perangkat mobile untuk melakukan tugas tertentu. Menurut Kamus Besar Bahasa Indonesia, aplikasi adalah penerapan dari rancangan sistem untuk mengolah data menggunakan aturan atau bahasa pemrograman tertentu. Aplikasi *mobile* memberikan fleksibilitas bagi pengguna untuk tetap dapat menggunakan aplikasi tersebut meskipun berpindah tempat tanpa terjadi pemutusan komunikasi, menjadikannya sangat berguna dalam berbagai situasi *mobile*.

3.1 *Android*

Android adalah sistem operasi berbasis *Linux* yang dirancang terutama untuk perangkat mobile layar sentuh seperti smartphone dan tablet. *Android* dikembangkan oleh *Google* dan pertama kali diluncurkan pada tahun 2008. Sistem operasi ini terkenal karena sifatnya yang *open-source* dan dukungannya terhadap berbagai perangkat keras dan aplikasi.

Menurut Hidayat (2012), *Android* adalah sistem operasi *mobile* yang menggunakan *kernel Linux* dan dirancang untuk perangkat *mobile* layar sentuh seperti *smartphone* dan *tablet*. Keunggulan utama dari *Android* adalah sifatnya yang *open-source*, yang memungkinkan pengembang untuk memodifikasi dan mendistribusikan perangkat lunak sesuai kebutuhan mereka. Selain itu, *Android* mendukung berbagai aplikasi yang dapat diunduh melalui *Google Play Store*, memberikan fleksibilitas dan pilihan yang luas bagi pengguna.

Sistem operasi *Android* memiliki beberapa versi yang terus diperbarui untuk meningkatkan kinerja dan menambah fitur baru. Setiap versi *Android* memiliki nama kode yang diambil dari nama makanan manis, seperti *Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop, Marshmallow, Nougat, Oreo, Pie,* dan yang terbaru adalah *Android* 10, 11, 12, dan seterusnya.

Android juga mendukung berbagai bahasa pemrograman, namun yang paling umum digunakan adalah *Java dan Kotlin*. Penggunaan Kotlin sebagai bahasa pemrograman resmi untuk *Android* diumumkan oleh Google pada tahun 2017. Menurut Yulianto (2018), Kotlin menawarkan sintaks yang lebih bersih dan fitur-fitur yang lebih canggih dibandingkan Java, sehingga memudahkan pengembangan aplikasi yang lebih efisien dan minim *bug*.

4.1 Aplikasi *Mobile* Berbasis *Android*

Aplikasi *mobile* berbasis *Android* adalah perangkat lunak yang dirancang untuk dijalankan pada perangkat *Android*. Aplikasi ini memanfaatkan kemampuan perangkat *mobile* untuk menyediakan berbagai layanan kepada pengguna. Menurut Romadani (2010), aplikasi *mobile* adalah perangkat lunak yang dapat digunakan oleh pengguna meskipun mereka berpindah tempat dengan mudah tanpa terjadi pemutusan komunikasi.

Aplikasi *mobile* berbasis *Android* dikembangkan menggunakan *Android SDK (Software Development Kit)* yang menyediakan berbagai *tools* dan *API (Application Programming Interface)* untuk membantu pengembang dalam membuat aplikasi. Menurut Setiawan (2019), *Android SDK* menyediakan *emulator*, *debugger*, dan *libraries* yang diperlukan untuk mengembangkan, menguji, dan mendistribusikan aplikasi *Android*.

Selain itu, aplikasi *Android* harus mematuhi pedoman desain yang ditetapkan oleh *Google* untuk memastikan konsistensi dan kenyamanan pengguna. Menurut Prasetyo (2015), pedoman desain ini mencakup prinsip-prinsip seperti navigasi yang intuitif, penggunaan ikon yang jelas, dan tampilan antarmuka yang responsif.

2.2.7. UML *(Unified Modeling Language)*

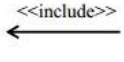
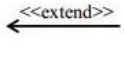
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis Objek. *Unified Modeling Language* (UML) bukanlah merupakan bahasa pemprograman tetapi model-model yang

tercipta berhubungan langsung dengan berbagai macam bahasa pemprograman, sehingga memungkinkan melakukan pemetaan (mapping) langsung dari model-model yang dibuat dengan *Unified Modeling Language* (UML) dengan bahasa-bahasa pemprograman berorientasi obyek. UML *(Unified Modeling Language)* adalah salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorintasi objek. UML *(Unified Modeling Language)* memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya :

a. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk sistem informai yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case.*

Tabel 2.1 Simbol *Use case diagram*

Simbol	**Keterangan**
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan *use case*
	*Use case* : Abstraksi dan interaksi antara sistem dan aktor
	Association : Abstraksi dari penghubung antara aktor dengan use case
	Generalisasi : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

b. *Activity Diagram*

Activity Diagram menggambarkan aliran kerja *(workflow)* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas

Tabel 2.2 Simbol *Activity diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memilih sebuah status awal.
Aktivitas 	Aktivitas yang di lakukan sistem aktivitas biasanya di awali dengan kata kerja
Percabangan / decision 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari
Penggabungan / join 	Asosiasi penggabungan dimana lebih satu aktivitas di gabung menjadi satu
Status akhir 	Status akhir yang di lakukan sistem sebuah diagram aktivitas memiliki sebuah status akhir

c. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Maka dari itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* serta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Sequence diagram mendokumentasikan komunikasi antar kelas-kelas. Diagram ini menunjukkan sejumlah obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek didalam *use case.*

Dalam diagram ini, kelas dan aktor ditempatkan secara berurutan dari kiri ke kanan di bagian atas diagram, dengan garis diposisikan secara vertikal di depan kelas dan aktor. Berikut adalah simbol-simbol yang ada pada Sequence diagram:

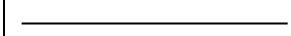
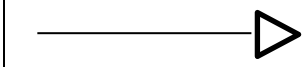
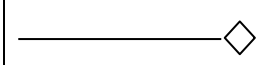
Tabel 2.3 Simbol *Sequence diagram*

Simbol	Deskripsi
Aktor  Nama Aktor	Orang, proses atau sistem yang lain yang berinteraksi dengan sistem informasi yang akan di buat diluar sistem yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tetapi aktor belum tentu menggunakan orang; biasanya di nyatakan menggunakan kata bemda di awal frase nama aktor
Garis hidup / *lifeline*	Menyatakan hidup suatu objek
Objek Nama Objek : Nama Kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe *create* <<create>>	Objek yang lain, arah panah mengarah pada objek yang di buat

d. *Class diagram*

Class Diagram menunjukan interaksi antara kelas dalam sistem. Kelas mengandung informasi dan tingkah laku yang berkaitan dengan informasi tersebut. Sebuah kelas pada diagram kelas dibuat untuk setiap tipe objek pada diagram sekuensial atau diagram kolaborasi . *Class diagram* menggambarkan struktur sistem dari segi pendefinisian yang akan dibuat untuk membangun sistem. Berikut adalah simbol-simbol yang ada pada diagram kelas.

Tabel 2.4 Simbol *Class diagram*

Simbol	Deskripsi
Kelas	Kelas pada struktur sistem
Antarmuka/ / *interface*  Nama_*interface*	Sama dengan konsep interface dalam pemograman berorientasi objek
Asosiasi / *association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesifikasi (umum-khusus)
Keberuntung/*dependency* 	Relasi antar kelas dengan makna ketergantungan antar kelas
Agregasi / *aggregation* 	Relasi antar kelas dengan makna semua bagian

2.2.8. *Visual Studio Code*

Visual Studio Code (VS Code) adalah editor kode sumber yang dikembangkan oleh *Microsoft* untuk *Windows, Linux,* dan *macOS. VS Code* dirilis pertama kali pada April 2015 dan telah berkembang menjadi salah satu editor kode paling populer di dunia.

Fitur-fitur utama dari *visual studio code* antara lain :

a. *Extensions*: *VS Code* mendukung ribuan ekstensi yang memungkinkan pengguna untuk menambahkan fungsionalitas sesuai kebutuhan.
b. *Integrated Terminal:* Fitur ini memungkinkan pengguna untuk menjalankan perintah dari dalam *editor*.
c. *Source Control*: Integrasi dengan *Git* memungkinkan pengguna untuk mengelola versi kode mereka langsung dari *VS Code*.
d. *Debugging*: Fitur debugging yang kuat mendukung berbagai bahasa pemrograman dan konfigurasi.
e. *IntelliSense*: Fitur ini memberikan saran kode yang cerdas, autocompletion, dan parameter info yang membantu dalam penulisan kode.

2.2.9. *Dart*

Dart merupakan Bahasa pemrograman umum yang dibuat oleh *Google* pada tahun 2011, bahasa ini digunakan dalam membuat aplikasi *Flutter*. Mirip seperti *java dan C#*, Dart memiliki sintaks yang mirip dengan *C*. Tidak seperti Bahasa pemrograman lainnya, Bahasa ini dioptimasi untuk dikembangkan dan dijalankan di berbagai platform :

- Di *Web browser* sebagai *JavaScript*
- Sebagai Aplikasi *Native*

Dart dirilis secara *open source* sehingga dapat digunakan secara bebas , berbasis kelas dan orientasi object serta menggunakan sintaks Bahasa perograman *C*.

2.2.10. Flutter

Flutter adalah sebuah teknologi yang dibuat oleh Google untuk membangun aplikasi *Mobile* (*Android*, *iOS*), *web* dan *desktop* (*Windows*, *Mac, Linux*) sekaligus dengan hanya melalui satu kode sumber saja, kode ini akan *di-compile* secara *native*. Poin ini yang menjadi perbedaan antara Google *Flutter* dan *Framework cross- platform* lainnya. *Flutter* menggunakan bahasa pemrograman *Dart*.

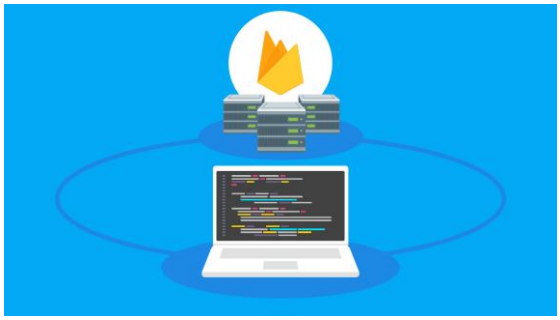
Penulis menyimpulkan *Flutter* merupakan *Framework* atau kerangka kerja untuk membuat aplikasi *Mobile, web* dan *desktop* dengan satu *codebase* yaitu dengan bahasa pemrograman *Dart* .

2.2.11. *Firebase*

Firebase adalah penyedia layanan *cloud* dengan *backend* sebagai servis yang berbasis di San Fransisco, California. *Firebase* membuat sejumlah produk untuk pengembangan aplikasi *Mobile* ataupun *web*. *Firebase* didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara *realtime* di tahun 2012. Produk utama dari *Firebase* yakni suatu *database* yang menyediakan *API* untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat *multiple client*. Perusahaan ini diakusisi oleh *Google* pada Oktober 2014. *Firebase* adalah penyedia layanan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi yang memungkinkan pengembang membuat *API* untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud. Firebase* memiliki banyak *library* yang memungkinkan untuk mengintegrasikan dengan *Android, Ios, Javacript, Java, Objective-C dan Node.JS* . *Database Firebase* juga bisa diakses lewat *REST API. REST API* tersebut menggunakan *protokol Server-Sent Event* dengan membuat koneksi *HTTP* untuk menerima *push notification* dari *server*. Pengembang menggunakan *REST API.* untuk *post* data yang selanjutnya *Firebase client library* yang sudah diterapkan pada aplikasi yang dibangun yang akan mengambil data *secara realtime.*

Dari pengertian diatas penulis menyimpulkan *Firebase* adalah penyedia layanan *cloud* yang memungkinkan pengembang untuk membangun aplikasi *Mobile* dan *web* dengan menyediakan berbagai produk, termasuk *database real-time*. Produk ini memungkinkan sinkronisasi data melalui berbagai klien dan tersedia dengan berbagai *library* untuk berbagai *platform*. *Firebase* juga dapat diakses melalui *REST API* dan *Server-Sent Event* untuk pembaruan data *real-time*.

Berikut gambaran dari topologi *Firebase* :



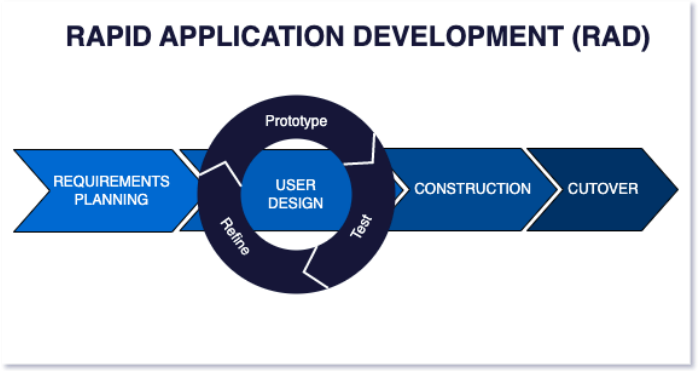
Gambar 2.2 *Firebase*

2.2.12. *Blackbox Testting*

Metode *Blackbox Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan *Detail software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing- masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai (Latif, 2015). Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan.

BAB 3
METODOLOGI PENELITIAN

3.1 Prosedur Penelitian



Gambar 3.1 Metode *Rapid Apllication Development (RAD)*

Penulis menggunakan metode *RAD (Rapid Application Development)* karena salah satu model yang fleksibel untuk digunakan. *RAD* merupakan model yang simpel dan mudah untuk di pahami maupun di gunakan dalam pembangunan suatu sistem. Penulis menggunakan tahapan-tahapan metode *RAD* dalam pembangunan sistem. Berikut tahapan-tahapan yang dilakukan :

1. *Requirement Planning* (Perencanaan Kebutuhan)
   - Tahap awal bertujuan untuk mengidentifikasi kebutuhan sistem.
   - Melibatkan diskusi antara pengembang dan pengguna untuk menentukan tujuan proyek,batasan dan fitur utama.
   - Fokus pada pemahaman masalah yang akan diselesaikan oleh aplikasi.
2. *User Design* (Perancangan oleh Pengguna)
   - Pembuatan prototipe awal berdasarkan kebutuhan yang telah disepakati
   - Pengguna terlibat aktif dalam memberikan masukan dan melakukan evaluasi terhadap desain
   - Dilakukan secara iteratif,artinya perancangan dan revisi dilakukan berulang-ulang hingga sesuai kebutuhan

3. *Construction* (Pembangunan Sistem)
    - Pengembangan mulai menulis kode dan membangun sistem berdasarkan prototipe yang telah disempurnakan
    - Proses ini tetap fleksibel sehingga perbaikan atau penyesuaian dapat dilakukan dengan cepat.
    - Pengujian dilakukan secara paralel untuk memastikan aplikasi berjalan dengan baik.
4. *Cutover* (Implementasi dan Penyempurnaan)
    - Sistem yang sudah dikembangkan diterapkan ke lingkungan nyata dan mulai digunakan oleh pengguna.
    - Migrasi data,pelatihan pengguna,serta pemantauan awal untuk memsatikan sitem berjalan secara optimal.
    - Jika ada masalah,perbaikan dilakukan secara cepat agar aplikasi berjalan dengan baik.

3.2 Instrumen Penelitian

Perangkat yang digunakan dalam penelitian ini adalah sebagai berikut :

1. *Software* (Perangkat Lunak)

   Dalam melakukan penelitian,peneliti menggunakan beberapa perangkat lunak sebagai berikut :

a. Sistem Operasi yang digunakan adalah *Windows 11 Home Single Language 64-bit (10.0, Build 22635).*
b. Aplikasi Visual Studio Code yang digunakan untuk melakukan pengkodingan dari sistem yang akan dibuat.
c. Aplikasi Figma digunakan untuk membuat prototipe antarmuka aplikasi.
d. *Smartphone Android* digunakan untuk melakukan *debugging* untuk mencari *bug* atau eror di perangkat *andorid.*
e. *Firebase* digunakan untuk menjadi layanan *Backand* untuk mendukung layanan aplikasi dengan memanfaatkan beberapa fitur yaitu *Firebase Fire Store* sebagai media penyimpanan data secara *real time* dan dapat diakses

secara cepat dan sinkron dan *Firebase cloud Storage* untuk menyimpan file foto yang di unggah ke *database.*

2. *Hardware* (Perangkat Keras)

   Beberapa *Hardware* (perangkat keras) yang digunakan peneliti dalam penelitian ini adalah :

   a. Laptop MSI Modern 14
   b. Processor Intel i3-10110U with Intel UHD Graphics
   c. RAM dengan ukuran 8 GB
   d. SSD dengan ukuran 256 GB
   e. Layar Monitor 14”
   f. Smartphone Samsung S22 dan Samsung J7 2016

3.2.1. Analisa Sistem Yang Berjalan

Dalam analisa sistem yang berjalan di Dinas Pemberdayaan Masyarakat,Perempuan,Perlindungan Anak,Pengendalian Penduduk Dan Keluarga Berencana (DPMP4KB) penulis mengidentifikasi beberapa permasalahan yang memerlukan pembenahan. yang pertama, kurangnya efisiensi pencatatan aset yang dilakukan dikarenakan pencatatan ditulis secara manual mengakibatkan tingginya resiko kesalahan pencatatan. Kedua pemindahan lokasi aset yang tidak tercatat lengkap mengakibatkan banyaknya aset yang hilang. Ketiga pelaporan aset dilakukan secara manual belum ada moderenisasi. Terakhir tidak ada bukti pemindahan aset secara jelas sehingga menyulitkan monitoring data aset dari setiap lokasi aset tersebut.

3.2.2. Analisa Sistem yang Diusulkan

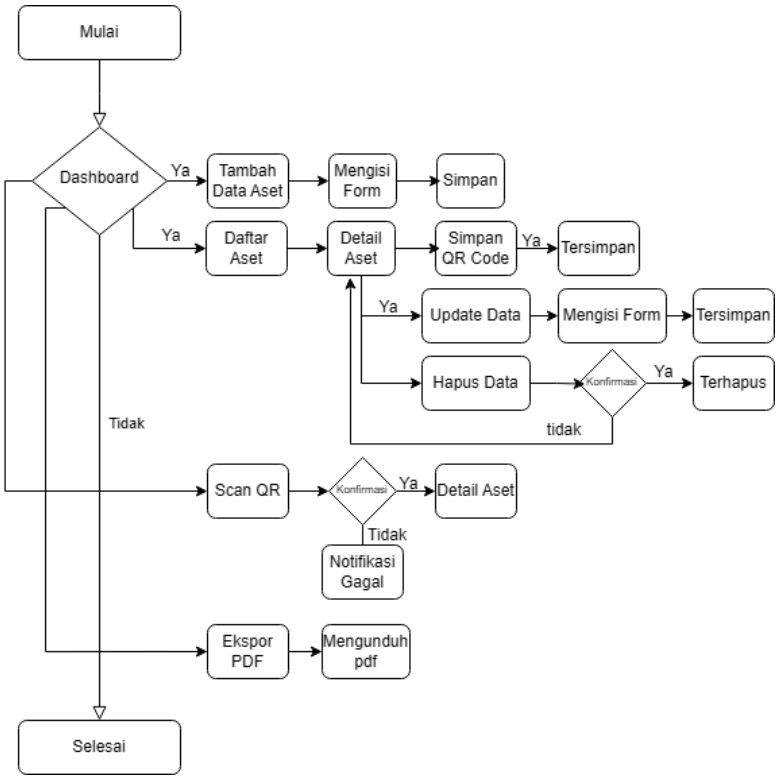
Berdasarkan hasil observasi dan analisa masalah pada sistem yang berjalan, penulis merancang sistem berikut untuk menyelesaikan permasalahan tersebut.

a. Sistem yang dibuat adalah aplikasi Mobile berbasis *Android* .
b. Terdapat beberapa fitur dalam aplikasi tersebut yaitu : Tambah aset, foto aset, daftar aset, pencarian aset, simpan *QR Code* aset,update data aset, hapus data aset, pindai *QR Code* aset, ekspor pdf keseluruhan aset.

c. Setiap aset mempunyai informasi didalamnya yaitu : *QR Code,* foto aset, kode aset, nama aset, tahun pembelian, jenis aset, nomor registrasi, ruangan aset, merek aset, kondisi aset.

d. Data aset dan foto di simpan di *Firebase.*

1. *Flowchart* Program Aplikasi



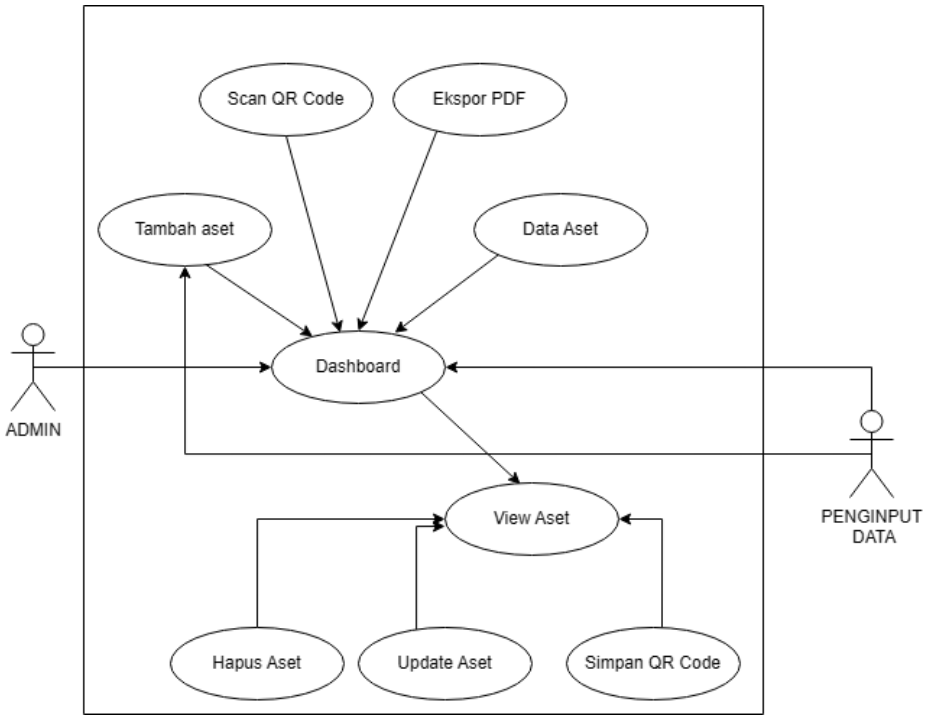
Gambar 3.2 *Flowchart* Program Aplikasi

3.3 Desain Pengembangan Sistem

1. Desain UML Sistem

   Dalam pengembangan sistem, penulis memanfaatkan UML (Unified Modeling Language) untuk mendefinisikan kebutuhan, melakukan analisis dan perancangan, serta menggambarkan arsitektur. Penulis menggunakan beberapa jenis diagram UML, antara lain sebagai berikut :

a. Use Case Diagram



Gambar 3.3 *Use Case Diagram* Pengembangan sistem

Penjelasan :

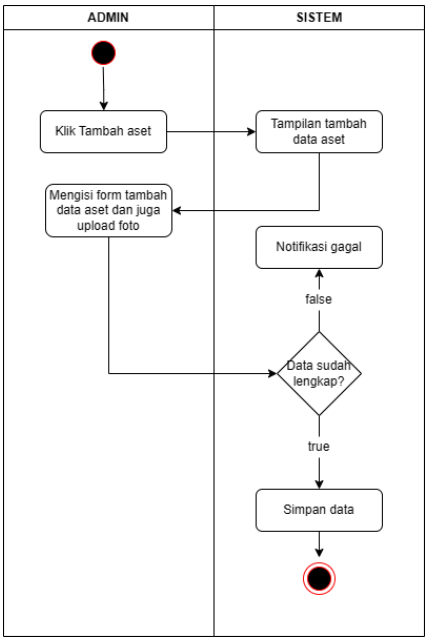
Admin : Aktor yang mempunyai hak akses penuh pada sistem untuk mengelola data aset.

Penginput Data : Aktor yang mempunyai hak untuk menginputkan data aset.

b. Activity Diagram

Melalui *activity diagram* ini, penulis akan menjelaskan rangkaian aktivitas serta interaksi pengguna yang di ilustrasikan melalui aktor dalam sistem ini. Berikut adalah tampilannya :

Tampilan *activity diagram* pada halaman tambah data aset :



Gambar 3.4 *Activity diagram* halaman tambah data aset

1. Narasi *activity diagram* halaman tambah data aset

Activity diagram ini menggambarkan proses yang terjadi ketika seorang Admin ingin menambahkan data aset baru ke dalam sistem. Berikut adalah langkah-langkah dalam diagram:

a. Mulai

   Proses dimulai oleh Admin, yang ditandai dengan simbol titik hitam di bagian kiri diagram.

b. Klik Tambah Aset

   Admin mengklik tombol "Tambah Aset" di dalam sistem. Sistem kemudian menampilkan tampilan formulir tambah data aset.

c. Mengisi Form Tambah Data Aset dan Upload Foto

   Admin mengisi formulir dengan informasi aset yang diperlukan, termasuk mengunggah foto aset jika diperlukan.

d. Validasi Kelengkapan Data

Sistem memeriksa apakah semua data yang diperlukan telah diisi dengan benar.

e. Jika Data Tidak Lengkap

   Jika data yang dimasukkan tidak lengkap, sistem menampilkan notifikasi gagal, dan Admin harus kembali ke formulir untuk melengkapi data.

f. Jika Data Lengkap

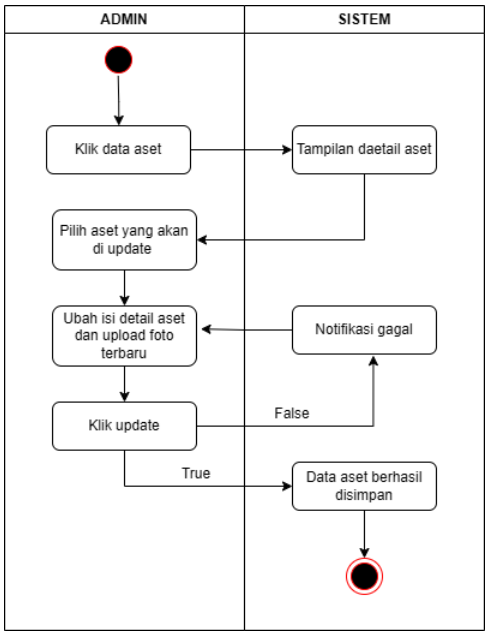
   Jika semua data telah terisi dengan benar, sistem akan menyimpan data aset ke dalam *database*.

g. Proses Selesai

   Setelah data berhasil disimpan, proses selesai, yang ditandai dengan simbol titik hitam dengan lingkaran merah.

Diagram ini menunjukkan bagaimana sistem memastikan bahwa hanya data aset yang valid yang dapat disimpan, serta bagaimana Admin dapat mengoreksi kesalahan sebelum menyelesaikan proses input.

Tampilan *activity diagram* pada halaman *update* data aset :



Gambar 3.5 *Activity* diagram halaman *update* data aset

2. Narasi *activity diagram* halaman *update* data aset

Activity diagram ini menggambarkan alur proses saat seorang Admin melakukan pembaruan atau update data aset dalam sistem. Berikut adalah langkah-langkahnya:

a. Mulai

   Proses dimulai oleh Admin, yang ditandai dengan simbol titik hitam di awal diagram.

b. Klik Data Aset

   Admin memilih menu atau tombol "Data Aset" untuk melihat daftar aset yang sudah terdaftar dalam sistem.

c. Sistem Menampilkan Detail Aset

   Sistem akan menampilkan detail aset yang tersedia, sehingga Admin dapat melihat informasi lengkap sebelum melakukan perubahan.

d. Memilih Aset yang Akan Diperbarui

   Admin memilih aset tertentu yang akan diperbarui dari data yang di tampilkan.

e. Mengubah Detail Aset dan Mengunggah Foto Terbaru

   Admin melakukan perubahan pada informasi aset dan juga melakukan upload foto aset terbaru.

f. Klik Update

   Setelah melakukan perubahan, Admin mengklik tombol "*Update"* untuk menyimpan perubahan.

g. Validasi Kelengkapan Data

   Sistem memeriksa apakah semua data yang diperbarui sudah lengkap dan valid.

h. Jika Data Tidak Lengkap

   Jika data yang diperbarui tidak memenuhi persyaratan, sistem akan menampilkan notifikasi gagal, dan admin harus mengoreksi data.

i. Jika Data Lengkap

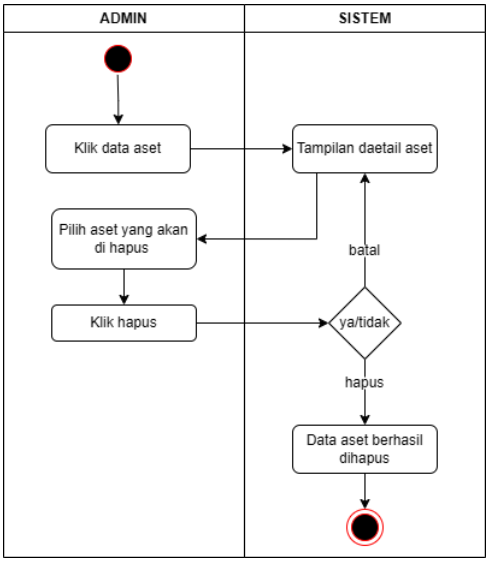
Jika semua data sudah benar dan lengkap, sistem akan menyimpan perubahan data aset ke dalam *database*.

j. Proses Selesai

Setelah data berhasil diperbarui, proses selesai ditandai dengan simbol titik hitam dengan lingkaran merah.

Diagram ini menunjukkan bagaimana sistem memastikan bahwa hanya perubahan aset yang valid yang dapat disimpan, serta bagaimana Admin dapat memperbarui informasi aset dengan mudah dan terorganisir.

Tampilan *activity diagram* halaman hapus data aset



Gambar 3.6 *Activity diagram* halaman hapus data aset

3. Narasi *activity diagram* halaman hapus data aset

Activity diagram ini menggambarkan alur proses saat seorang Admin melakukan penghapusan data aset dalam sistem. Berikut adalah langkah-langkahnya:

a. Mulai

Proses dimulai oleh Admin, yang ditandai dengan simbol titik hitam di awal diagram.

b. Akses Dashboard

   Admin mengakses halaman dashboard, yang merupakan tampilan utama sistem.

c. Klik Data Aset

   Admin memilih menu atau tombol "Data Aset" untuk melihat daftar aset yang sudah terdaftar dalam sistem.

d. Sistem Menampilkan Detail Aset

   Sistem akan menampilkan detail aset yang tersedia, sehingga Admin dapat melihat informasi lengkap sebelum melakukan penghapusan.

e. Memilih Aset yang Akan Dihapus

   Admin memilih aset tertentu yang akan dihapus dari daftar aset yang ditampilkan.

f. Klik Hapus

   Sistem menampilkan dialog konfirmasi yang meminta Admin untuk mengonfirmasi apakah benar ingin menghapus aset tersebut. Jika admin memilih tidak maka proses dibatalkan dan aset tetap tersimpan dalam sistem.

g. Data Aset Berhasil Dihapus

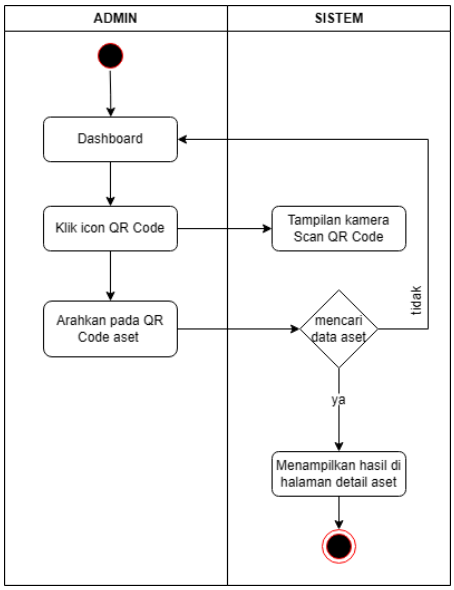
   Jika Admin mengonfirmasi penghapusan, sistem akan menghapus data aset dari *database* dan memastikan aset tersebut tidak lagi tersedia dalam daftar.

h. Proses Selesai

   Setelah data berhasil dihapus, proses selesai ditandai dengan simbol titik hitam dengan lingkaran merah.

Diagram ini menunjukkan bahwa proses penghapusan data aset dilakukan dengan aman melalui tahap konfirmasi sebelum eksekusi, guna menghindari kesalahan dalam penghapusan data yang tidak disengaja.

Tampilan *activity diagram* halaman pindai *qr code*



Gambar 3.7 *Activity diagram* halaman pindai *qr code*

4. Narasi *activity diagram* halaman pindai *qr code*

Activity diagram ini menggambarkan alur proses saat seorang admin menggunakan fitur pemindaian *qr code* untuk mencari data aset dalam sistem. Berikut adalah tahapan yang terjadi dalam proses ini:

a. Mulai

Proses dimulai oleh Admin, yang ditandai dengan simbol titik hitam di awal diagram.

b. Akses Dashboard

Admin mengakses halaman dashboard, yang merupakan tampilan utama sistem.

c. Klik Icon *qr code*

Admin memilih fitur pemindaian dengan menekan ikon *qr code* yang tersedia dalam sistem.

d. Sistem Menampilkan Kamera untuk memindai *qr code*

Sistem mengaktifkan kamera perangkat dan menampilkan tampilan pemindaian *qr code***.**

e. Admin Mengarahkan Kamera ke QR Code Aset

Admin mengarahkan kamera ke *qr code* yang tertera pada aset yang ingin dipindai.

f. Sistem Mencari Data Aset

Sistem memproses hasil pemindaian dan mencari data aset yang sesuai di dalam *database*. Jika sistem tidak menemukan data aset, sistem akan menampilkan notifikasi gagal.

g. Jika Data Aset Ditemukan

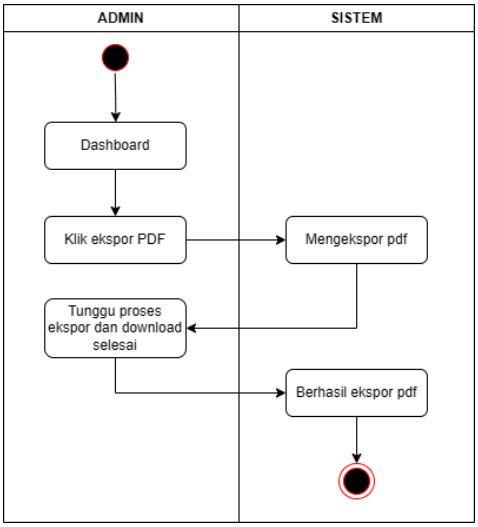
Jika sistem berhasil menemukan data aset yang sesuai, sistem akan menampilkan detail aset pada layar.

h. Proses Selesai

Setelah data berhasil dihapus, proses selesai ditandai dengan simbol titik hitam dengan lingkaran merah.

Diagram ini menggambarkan bagaimana fitur pemindaian QR Code memungkinkan Admin untuk mengakses informasi aset dengan cepat dan efisien**.**

Tampilan *activity diagram* halaman ekspor *pdf*



Gambar 3.8 *Activity diagram* halaman ekspor *pdf* data aset

5. Narasi *activity diagram* halaman ekspor *pdf* data aset

Activity diagram ini menggambarkan alur proses saat seorang admin mengekspor data aset dalam bentuk file *pdf* melalui sistem. Berikut adalah tahapan dalam proses ini:

a. Mulai

Proses dimulai oleh Admin, yang ditandai dengan simbol titik hitam di awal diagram.

b. Akses Dashboard

Admin mengakses halaman dashboard, yang merupakan tampilan utama sistem.Sistem Menampilkan Detail Aset

c. Klik Ekspor *pdf*

Admin memilih fitur "*Ekspor pdf*" untuk memulai proses pembuatan dokumen dalam format *pdf*.

d. Sistem Mengekspor *pdf*

Sistem mulai memproses data dan mengonversinya ke dalam format *pdf* sesuai dengan struktur yang telah ditentukan.

e. Tunggu Proses Ekspor dan Unduh Selesai

Proses ini bisa memerlukan waktu tergantung pada jumlah data yang harus dikonversi ke PDF.

f. Sistem Berhasil Mengekspor PDF

Setelah proses ekspor selesai, sistem menyimpan file *pdf.*

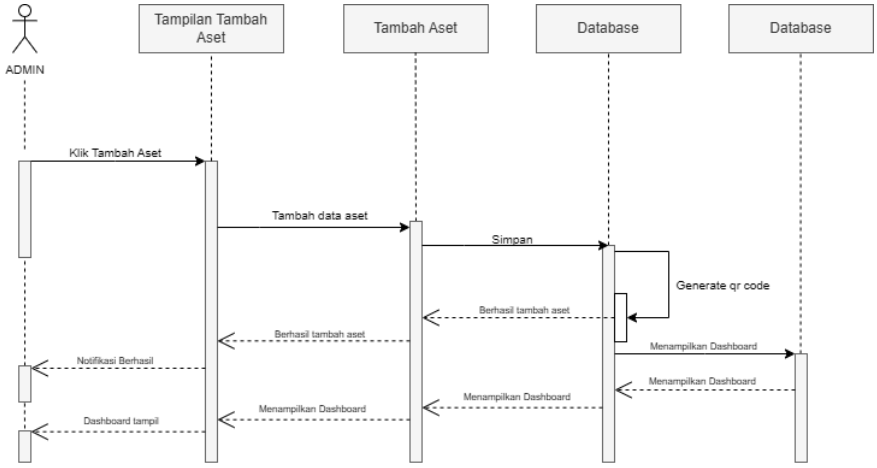
g. Proses Selesai

Setelah file *pdf* berhasil diekspor dan diunduh, proses berakhir, yang ditandai dengan simbol titik hitam dengan lingkaran merah.

Diagram ini menunjukkan bagaimana sistem memungkinkan admin untuk menghasilkan laporan dalam format *pdf* secara otomatis, sehingga mempermudah proses dokumentasi dan penyimpanan data.

c. *Sequence Diagram*

Sequence diagram ini menjelaskan proses *Generate QR Code* dan tambah aset



Gambar 3.9 *Sequence Diagram generate qr code* dan tambah aset

1. Narasi *Sequence Diagram generate qr code* dan tambah aset

Sequence diagram ini menggambarkan alur proses pembuatan *qr code* setelah admin menambahkan data aset dalam sistem. Berikut adalah tahapan dalam proses ini:

a. Admin Klik "Tambah Aset"

Admin memulai proses dengan menekan tombol "Tambah Aset" pada sistem.

b. Menampilkan Halaman Tambah Aset

Sistem menampilkan form tambah aset, yang memungkinkan Admin untuk mengisi data aset baru.

c. Admin Mengisi Form dan Mengirim Data

Admin mengisi form tambah aset dengan informasi aset yang diperlukan dan mengirimkan data ke sistem.

d. Sistem Menyimpan Data ke *Database*

Data aset yang dikirim oleh Admin diterima oleh sistem dan kemudian disimpan dalam *database*.

e. Sistem Menghasilkan QR Code

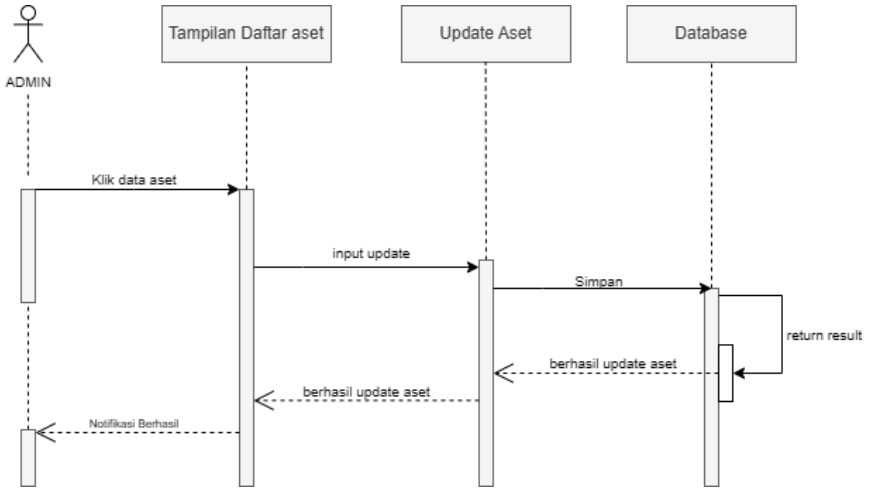
Setelah data aset berhasil disimpan, sistem secara otomatis menghasilkan QR Code yang unik untuk aset tersebut.

f. Menampilkan Hasil

Setelah proses selesai, sistem mengarahkan kembali Admin ke halaman daftar aset.

Diagram ini menunjukkan bagaimana sistem secara otomatis menghasilkan *qr code* setiap kali aset baru ditambahkan. Hal ini memungkinkan setiap aset memiliki identitas unik yang dapat dipindai untuk akses data lebih cepat dan efisien.

Sequence diagram ini menjelaskan proses *update* aset



Gambar 3.10 *Sequence Diagram update* aset

2. Narasi *Sequence Diagram update* aset

Sequence diagram ini menggambarkan bagaimana admin memperbarui data aset dalam sistem. Berikut adalah tahapan dalam proses ini:

a. Admin Memilih Data Aset

Admin memulai proses dengan mengklik "Data Aset" untuk melihat daftar aset yang tersedia.

b. Sistem Menampilkan Daftar Aset

   Sistem menampilkan halaman daftar aset, di mana admin dapat memilih aset yang ingin diperbarui.

c. Admin Memasukkan Perubahan Data

   Admin melakukan input update dengan mengubah detail aset sesuai kebutuhan, seperti nama aset, atau informasi lainnya.

d. Sistem Menyimpan Perubahan ke *Database*

   Data yang telah diperbarui dikirim ke sistem, kemudian disimpan dalam *database***.**

e. *Database* Mengonfirmasi Perubahan

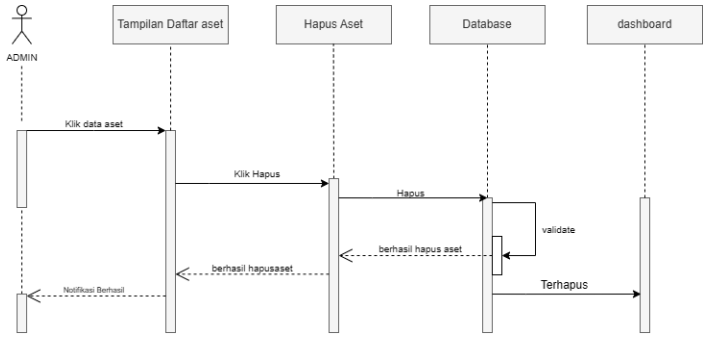
   Database memproses dan menyimpan perubahan yang dikirim oleh sistem. Setelah data berhasil diperbarui, *database* mengirimkan hasil konfirmasi kembali ke sistem.

f. Sistem Memberikan Notifikasi Berhasil

   Sistem mengirimkan notifikasi sukses kepada admin bahwa proses update telah berhasil.

Diagram ini menunjukkan bagaimana sistem menangani pembaruan data aset dengan memastikan perubahan tersimpan di *database* dan memberikan notifikasi kepada Admin. Proses ini membantu dalam memastikan integritas data aset dalam sistem.

Sequence diagram ini menjelaskan proses hapus aset



Gambar 3.11 *Sequence Diagram* hapus aset

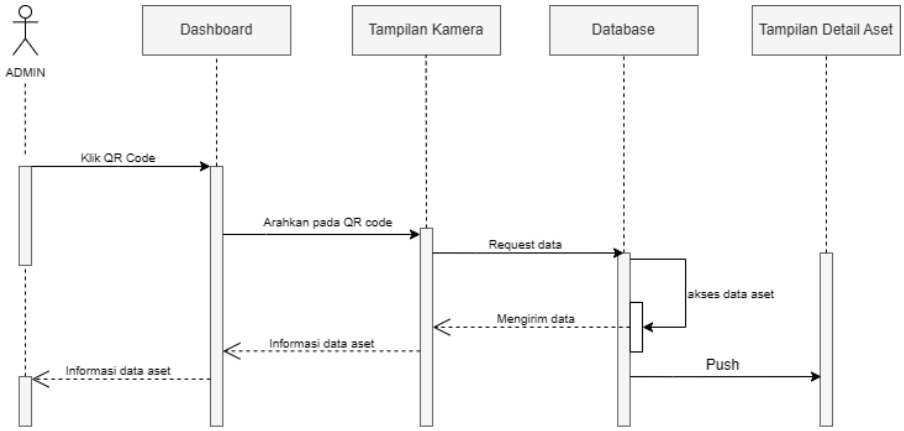
3. Narasi *Sequence Diagram* hapus aset

Sequence diagram ini menggambarkan bagaimana admin menghapus aset dalam sistem. Berikut adalah tahapan dalam proses ini:

a. Admin Memilih Data Aset
   Admin memulai proses dengan mengklik "Data Aset" untuk melihat daftar aset yang tersedia.
b. Sistem Menampilkan Daftar Aset
   Sistem menampilkan halaman daftar aset, di mana admin dapat memilih aset yang ingin dihapus.
c. Admin Menghapus Aset
   Admin memilih aset yang ingin dihapus dan mengklik tombol "Hapus".
d. Sistem Mengirim Permintaan Hapus ke *Database*
   Sistem meneruskan permintaan penghapusan ke *database* untuk diproses.
e. *Database* Mengonfirmasi Penghapusan data
   Database melakukan validasi apakah aset dapat dihapus. Jika validasi berhasil, aset dihapus dari *database*.
f. Sistem Memberikan Notifikasi Berhasil
   Sistem mengirimkan notifikasi sukses kepada admin bahwa proses hapus telah berhasil.

Diagram ini menjelaskan alur penghapusan aset secara sistematis dengan melibatkan validasi dari *database* sebelum aset benar-benar dihapus. Dengan adanya notifikasi keberhasilan, Admin dapat memastikan bahwa aset telah dihapus dari sistem.

Sequence diagram ini menjelaskan proses *Scan QR Code*



Gambar 3.12 *Sequence Diagram scan qr code*

4. Narasi *Sequence Diagram* scan qr code

Sequence diagram ini menjelaskan alur kerja sistem dalam proses pemindaian *qr code* aset untuk menampilkan detail informasi aset. Berikut adalah langkah-langkah dalam proses ini:

a. Admin Memulai Pemindaian QR Code

   Admin mengakses dashboard dan mengklik ikon *qr code* untuk memulai proses pemindaian..

b. Sistem Menampilkan Tampilan Kamera

   Sistem mengaktifkan kamera dan menampilkan antarmuka pemindaian *qr code*.

c. Admin Memindai QR Code

   Admin mengarahkan kamera ke *qr code* aset untuk melakukan pemindaian..

d. Sistem Meminta Data ke *Database*

   Setelah *qr code* terdeteksi, sistem mengirimkan permintaan data aset ke *database* berdasarkan informasi yang terkandung dalam *qr code.*

e. *Database* Mengakses dan Mengirim Data Aset

   Database mencari informasi aset yang sesuai. Setelah ditemukan, *database* mengirimkan data aset kembali ke sistem.

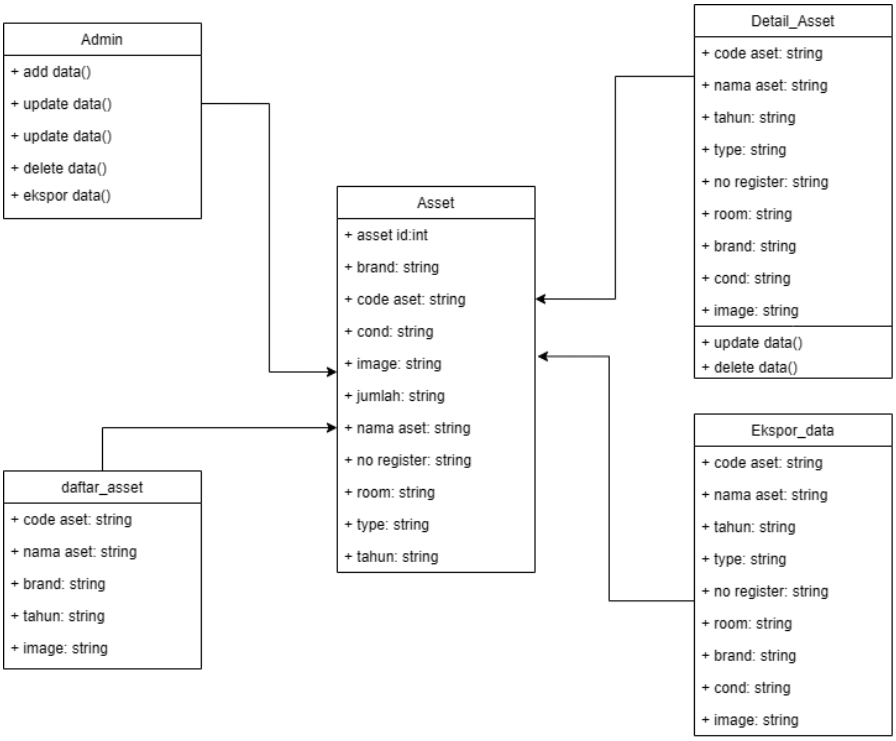
f. Sistem Menampilkan Detail Aset

Data aset yang diterima dikirim ke tampilan detail aset, sehingga admin dapat melihat informasi lengkap mengenai aset tersebut. Sistem juga mengirimkan informasi data aset kembali ke dashboard sebagai konfirmasi bahwa pemindaian berhasil.

Proses ini memungkinkan Admin untuk dengan cepat mendapatkan informasi aset hanya dengan memindai QR Code menggunakan kamera. Alur ini memastikan bahwa sistem secara otomatis mengambil dan menampilkan data yang relevan dari *database* tanpa perlu pencarian manual.

d. *Class diagram*

Class diagram menunjukan relasi antar kelas serta memberikan detail deskripsi tiap kelas dalam model desain sistem. Setiap kelas dapat dilihat pada ilustrasi yang disajikan dibawah ini :



Gambar 3.13 *Class Diagram*

3. Perancangan Struktur *Database* Sistem

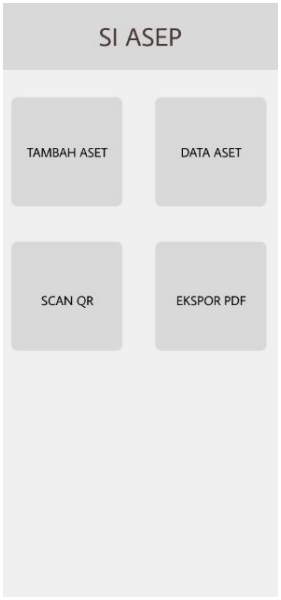
   Perancangan struktur *database* menguraikan jenis file yang digunakan dalam pengolahan data pada sistem yang diusulkan, adalah sebagai berikut:

   a. Tabel *database* aset

Tabel 2.5 Tabel database Aset

Nama	Tipe Data
assetId	int
brand	string
code	string
cond	string
image	string
jumlah	string
name	string
reg	string
room	string
type	string
year	string

4. Perancangan Antarmuka



Gambar 3.14 Desain antarmuka *Dashboard*

Antarmuka ini merupakan tampilan dashboard utama dari sistem SI ASEP**,** yang dirancang untuk mempermudah pengguna dalam mengelola aset.

a. Bagian Header

   Menampilkan judul aplikasi “SI ASEP” sebagai identitas sistem.

b. Bagian Konten Utama

   Memiliki empat tombol utama, yang masing-masing memiliki fungsi sebagai berikut:

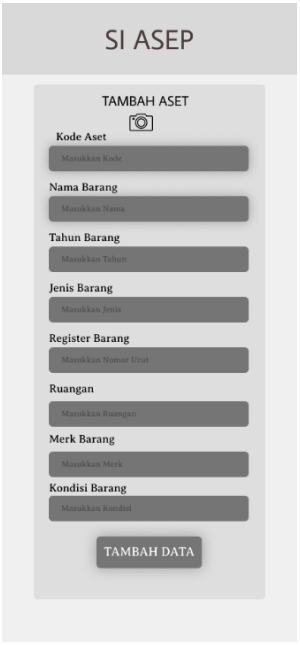
   1. Tambah Aset, untuk menambahkan aset baru ke dalam sistem.
   2. Data Aset, untuk mengakses dan mengelola daftar aset yang telah tersimpan.
   3. *Scan qr*, untuk memindai *qr code* aset guna menampilkan informasi detail aset secara langsung.
   4. Ekspor *pdf*, untuk mengekspor data aset dalam format *pdf* guna keperluan dokumentasi atau laporan.

c. Tata Letak

   Setiap tombol disusun dalam format grid 2x2, sehingga mudah diakses oleh pengguna.

Kesimpulan :

Dashboard ini dirancang dengan antarmuka yang sederhana dan fungsional**,** sehingga memudahkan pengguna dalam mengelola aset dengan fitur-fitur utama yang tersedia.



Gambar 3. 15 Desain antarmuka Tambah aset

Antarmuka ini merupakan tampilan halaman tambah aset dalam sistem “SI ASEP”, yang memungkinkan pengguna untuk menambahkan data aset baru dengan mudah.

a. Bagian Header

   Menampilkan judul aplikasi "SI ASEP", sebagai identitas sistem.

b. Bagian Formulir Tambah Aset

   Berisi beberapa kolom input yang diperlukan untuk mecatat aset yaitu:

   1. Kode Aset, kolom untuk memasukkan kode unik aset.
   2. Nama Barang, kolom untuk mengisi nama aset yang akan didaftarkan.
   3. Tahun Barang, kolom untuk mencatat tahun perolehan aset.

4. Jenis Barang, kolom untuk menentukan kategori atau jenis aset.
5. Register Barang, kolom untuk mencatat nomor register barang.
6. Ruangan, kolom untuk mencatat lokasi atau ruangan tempat aset disimpan.
7. Merk Barang, kolom untuk mencatat merek atau brand aset.
8. Kondisi Barang, kolom untuk mencatat kondisi fisik aset (baik, rusak, dll.).
9. Ikon kamera di bagian atas memungkinkan pengguna untuk mengambil gambar aset.

c. Tombol Tambah Data

Berfungsi untuk mengirimkan data yang telah diisi ke sistem.

d. Tata Letak

Formulir disusun secara vertikal untuk memudahkan input data secara bertahap. Tombol "TAMBAH DATA" dibuat mencolok agar mudah dikenali oleh pengguna.

Kesimpulan :

Halaman ini dirancang dengan antarmuka yang sederhana dan intuitif, sehingga pengguna dapat menambahkan aset dengan cepat dan akurat ke dalam sistem “SI ASEP”.



Gambar 3.16 Desain antarmuka Data aset

Antarmuka ini merupakan halaman daftar aset dalam sistem “SI ASEP”, yang berfungsi untuk menampilkan data aset yang telah terdaftar secara sistematis.

a. Bagian Header

   Menampilkan judul aplikasi "SI ASEP" sebagai identitas sistem.

b. Bagian Daftar Aset

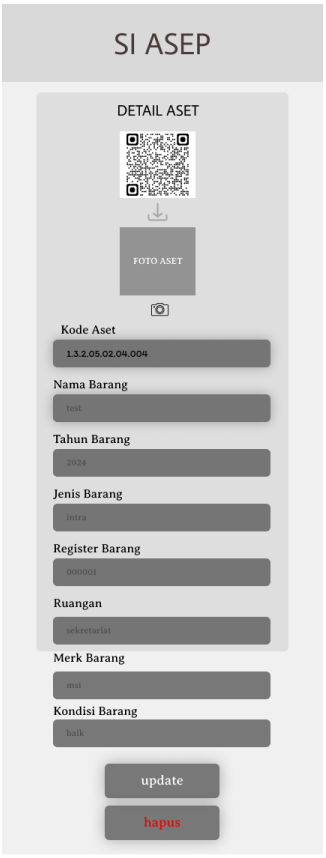
   Berisi daftar aset yang tersimpan dalam sistem. Setiap aset ditampilkan dalam bentuk kartu informasi yang berisikan kode aset,nama,merek,tahun dan *qr code.*

c. Tata Letak

   Daftar aset ditampilkan dalam format vertikal, memudahkan pengguna untuk melihat informasi secara sekilas.

Kesimpulan :

Halaman ini dirancang untuk mempermudah pengguna dalam melihat daftar aset secara cepat dan terstruktur.



Gambar 3.17 Desain antarmuka Detail aset

Antarmuka ini merupakan halaman detail aset dalam sistem “SI ASEP”, yang berfungsi untuk menampilkan informasi lengkap mengenai aset tertentu.

a. Bagian Header

   Menampilkan judul aplikasi "SI ASEP" sebagai identitas sistem.

b. Bagian Informasi Aset

   1. *QR Code* Aset, ditempatkan di bagian atas sebagai representasi unik dari aset, yang dapat unduh.
   2. Foto Aset, menyediakan ikon kamera**,** yang berfungsi untuk menambahkan atau memperbarui foto aset.
   3. Detail Informasi Aset, ditampilkan dalam bentuk kolom isian

c. Aksi

Tombol "Update"**,** memungkinkan pengguna untuk memperbarui data aset.

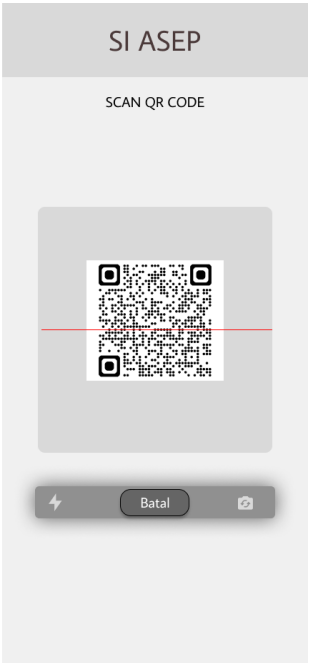
Tombol "Hapus" digunakan untuk menghapus aset dari sistem.

d. Tata Letak

Desain sederhana dan minimalis dan tombol aksi diletakkan dibagian bawah agar mudah dikenali.

Kesimpulan :

Halaman ini dirancang untuk memberikan tampilan detail aset yang lengkap, dengan fitur interaktif seperti unduh *QR*, dan menampilkan foto aset. Adanya tombol update dan hapus memungkinkan pengguna untuk mengelola data aset dengan lebih mudah dan efisien**.**



Gambar 3.18 Desain antarmuka *Scan qr code*

1. Antarmuka ini merupakan halaman pemindaian *qr code* dalam sistem “SI ASEP”, yang memungkinkan pengguna untuk memindai kode *qr* aset guna mengakses informasi aset dengan cepat dan akurat.

a. Bagian Header

   Menampilkan judul aplikasi "SI ASEP" sebagai identitas sistem.

b. Area Pemindaian QR Code

   Bagian utama halaman terdiri dari kotak pemindaian *qr code***,** yang menampilkan *qr code* dengan garis merah di tengahnya. Garis ini berfungsi sebagai indikator pemindaian**.**

c. Bagian Tombol Aksi

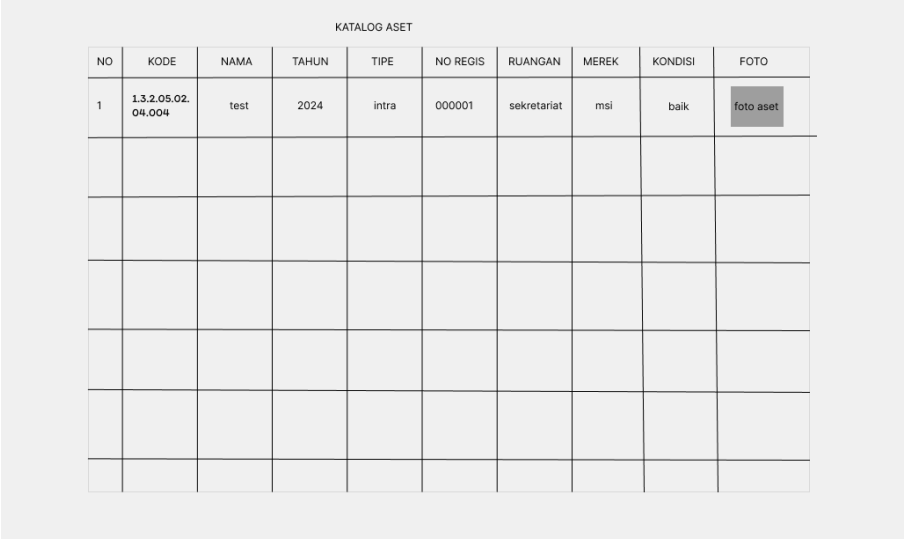
   Berisikan tombol batal,lampu kilat dan kamera.

d. Tata Letak

   Kotak pemindaian dengan garis merah memberikan indikasi visual yang jelas bagi pengguna.

Kesimpulan :

Halaman ini dirancang untuk memfasilitasi pemindaian *qr code* aset secara cepat dan efisien**.** Dengan adanya indikator pemindaian, tombol flash, dan opsi kamera**,** pengguna dapat dengan mudah menyesuaikan kondisi pemindaian sesuai kebutuhan.


KATALOG ASET

NO	KODE	NAMA	TAHUN	TIPE	NO REGIS	RUANGAN	MEREK	KONDISI	FOTO
1	1.3.2.05.02.04.004	test	2024	intra	000001	sekretariat	msi	baik	foto aset

Gambar 3.19 Desain antarmuka hasil Ekspor *pdf*

Antarmuka ini menampilkan halaman hasil ekspor *pdf* dari katalog aset dalam sistem “SI ASEP”. Halaman ini digunakan untuk

mendokumentasikan aset dalam format tabel, yang nantinya dapat disimpan atau dicetak dalam bentuk file *pdf*.

a. Bagian Header

   Terdapat judul "KATALOG ASET" di bagian atas, memberikan identitas dokumen yang dihasilkan.

b. Bagian Tabel Katalog Aset

   Tabel berisi beberapa kolom yang mencantumkan informasi penting tentang aset**,** yaitu nomor urut, kode aset,nama,tahun,tipe,nomor registrasi,ruangan,merek,kondisi,foto.

c. Struktur dan Tampilan

   Tabel memiliki format yang rapi dengan garis pemisah antar kolom dan baris, sehingga data dapat dibaca dengan jelas.

d. Fungsi dan Manfaat

   Memudahkan pengguna dalam menyimpan dan membagikan informasi aset dalam bentuk *pdf***.**

Kesimpulan :

Halaman ini berfungsi sebagai hasil ekspor katalog aset dalam format *pdf***,** yang dirancang dengan tabel informatif dan tampilan profesional**.** Dokumentasi ini dapat digunakan untuk keperluan laporan, pencetakan, atau penyimpanan arsip aset.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan pada bab sebelumnya, dapat dirangkum beberapa poin penting sebagai berikut:

1. Sistem informasi berbasis *Android* yang dikembangkan menggunakan metode *Rapid Application Development* (RAD), dengan pemanfaatan teknologi *QR Code* sebagai media identifikasi aset serta *Firebase* sebagai media penyimpanan data dan gambar secara daring, mampu membantu DPMP4KB Kota Magelang dalam memaksimalkan proses pelacakan lokasi, kondisi, pelaporan dan perpindahan aset secara efektif.
2. Penggunaan *QR Code* dalam sistem ini terbukti meningkatkan ketepatan identifikasi aset secara langsung (*real-time*) dan turut meminimalisir risiko kerugian akibat kehilangan aset atau ketidaktepatan informasi.
3. Melalui pengujian menggunakan metode black box, dapat disimpulkan bahwa sistem telah berfungsi sesuai dengan harapan. Seluruh fitur utama berjalan dengan baik, yang menandakan bahwa sistem sudah mampu memenuhi kebutuhan dasar dalam pengelolaan aset secara digital.

5.2 Saran

Sebagai bahan masukan untuk pengembangan sistem ke depan agar lebih optimal dan berkelanjutan, berikut beberapa hal yang dapat dipertimbangkan:

1. Sebelum sistem ini diterapkan secara menyeluruh, disarankan untuk melakukan kegiatan sosialisasi dan pelatihan kepada pengguna agar mereka memahami cara penggunaan serta alur kerja aplikasi dengan baik.

2. Untuk pengembangan selanjutnya, sistem sebaiknya dirancang agar bersifat *multiplatform*, sehingga dapat berjalan tidak hanya di *Android*, tetapi juga di perangkat *desktop* maupun sistem operasi *iOS*.
3. Dianjurkan agar sistem dilengkapi dengan fitur pencatatan riwayat atau log aktivitas, sehingga setiap perubahan terhadap data aset dapat dilacak secara terperinci.
4. Untuk peningkatan performa, dapat ditambahkan fitur pencarian aset secara offline agar aplikasi tetap dapat digunakan saat tidak terhubung dengan internet.
5. Evaluasi berkala terhadap sistem juga penting dilakukan untuk menyesuaikan dengan perubahan kebutuhan organisasi serta perkembangan teknologi yang terus berubah.

DAFTAR PUSTAKA

AICPA. (2020). Accounting for Fixed Assets. American Institute of Certified Public Accountants.

Coding Studio. (2023). Black Box Testing Adalah: Tipe Dan Contoh Pengujiannya. Diakses dari https://www.codingstudio.id/black-box-testing

Hidayat, R. (2012). Pengantar Sistem Operasi Android. Jakarta: Elex Media Komputindo.

Huang, Y., Wang, X., & Liu, Z. (2022). Enhancing the efficiency of QR Code recognition through machine learning techniques. Journal of Information and Telecommunication, 6(3), 250-267. https://doi.org/10.1080/24751839.2022.1234567

IFRS. (2021). International Financial Reporting Standards: Property, Plant and Equipment. IFRS Foundation.

Gupta, P., & Singh, R. (2020). QR Codes: Historical Development, Technological Advancements, and Emerging Trends. International Journal of Computer Applications, 176(36), 15-22. doi:10.5120/ijca2020920903

Kieso, D. E., Weygandt, J. J., & Warfield, T. D. (2019). Intermediate Accounting. Wiley.

Lopez, M. (2022). Implementing QR Code Technology in Asset Management Systems. Journal of Information Systems, 30(3), 223-235.

Lopez, M. (2022). The Role of QR Code Technology in Fixed Asset Recording. Journal of Management Information Systems, 25(2), 45-58.

M. Adriansyah and E. Sutrisna, “Aset Inventaris Divisi Parkir Menggunakan QR Code Berbasis Web (Studi Kasus : Universitas Pamulang),” vol. 1, no. 10, pp. 1756–1765, 2022.

Park, J., Lee, S., & Han, K. (2021). Evolution and Applications of QR Codes: A Literature Review. Journal of Digital Information Management, 19(4), 221-230. doi:10.1234/jdim.2021.0403

Prasetyo, A. (2015). Desain Antarmuka Pengguna pada Aplikasi Mobile. Bandung: Informatika.

Rahman, A. (2023). Enhancing Transparency in Fixed Asset Management Through QR Code Technology. Journal of Financial Management, 18(3), 112-125.

Romadani, S. (2010). Aplikasi Mobile: Konsep dan Implementasi. Yogyakarta: Graha Ilmu.

Sekawan Media. (2023). Black Box Testing: Definisi, Cara Kerja, dan 8 Tekniknya. Diakses dari https://www.sekawanmedia.co.id/black-box-testing

Setiawan, D. (2019). Panduan Lengkap Pengembangan Aplikasi Android. Surabaya: Penerbit Andi.

Serupa.id. (2022). Black Box Testing - Pengertian, Ciri, Jenis, Kategori & Langkah. Diakses dari https://www.serupa.id/black-box-testing

Tim Peneliti Universitas Teknologi Surabaya. (2024). Optimizing Fixed Asset Management with QR Code Application: A Case Study. Journal of Technology and Management, 30(1), 78-91.

Yulianto, A. (2018). Kotlin untuk Pemula: Panduan Praktis Pemrograman Android. Jakarta: Elex Media Komputindo.

Zhang, L., Li, H., & Chen, J. (2021). A study on the structure and application of QR Codes in modern information systems. International Journal of Computer Science and Network Security, 21(4), 175-182. https://doi.org/10.1111/ijcsns.2021.175