**PERANCANGAN SISTEM PEMINJAMAN BUKU PERPUSTAKAAN KAMPUS BERBASIS WEBSITE DENGAN PHP**

**Daud Ibrahim**

**Syaibatul Hamdi**

**Anjeng Puspita Ningrum, S.Kom**



**JUDUL PROYEK**

**Penulis :** Daud Ibrahim

Syaibatul Hamdi

**Editor** :

**Desain Sampul :**

**Tata Letak :**

**ISBN :**

# KATA PENGANTAR

Segala puji dan syukur kami panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, kami dapat menyelesaikan proyek ini. Laporan ini berjudul "PERANCANGAN SISTEM PEMINJAMAN BUKU PERPUSTAKAAN KAMPUS BERBASIS WEBSITE DENGAN PHP" yang merupakan hasil dari pemikiran para penulis. Kami menyadari bahwa penyelesaian laporan ini tidak lepas dari bimbingan berbagai pihak, untuk itu kami mengucapkan terima kasih kepada Ketua STTI NIIT ITECH, Kepala Program Studi, Dosen Pembimbing STTI NIIT ITECH, dan semua pihak yang telah mendukung kami dalam proses penyusunan laporan ini.

Dalam era digital ini, hampir semua layanan telah bertransformasi ke platform digital, sehingga menjadi lebih efisien dan lebih mudah diakses. Salah satunya adalah layanan perpustakaan yang beralih dari sistem manual ke sistem berbasis website. Kami melihat adanya potensi untuk merancang sistem peminjaman buku perpustakaan secara online, karena saat ini masih dilakukan secara konvensional, sehingga kami terdorong untuk merancang sistem peminjaman buku perpustakaan kampus berbasis website. Layanan ini diharapkan dapat memudahkan mahasiswa dan civitas akademika dalam mengakses informasi buku serta melakukan peminjaman tanpa harus datang langsung ke perpustakaan.

Namun, kami juga menyadari bahwa laporan ini masih memiliki kekurangan dan ruang untuk perbaikan. Oleh karena itu, kami sangat mengharapkan saran dan masukan yang membangun dari pembaca untuk perbaikan laporan ini di masa depan. Semoga laporan ini dapat memberikan manfaat serta kontribusi positif bagi pengembangan ilmu pengetahuan dan kemajuan teknologi.

Tim Penyusun

# DAFTAR ISI

[KATA PENGANTAR iii](#_30j0zll)

[DAFTAR ISI iv](#_3znysh7)

[**BAB 1** 5](#_2et92p0)

[**BAB 2** 7](#_tyjcwt)

[A. Perancangan 7](#_3dy6vkm)

[**BAB 3** 9](#_2s8eyo1)

[**BAB 4** 10](#_17dp8vu)

[**BAB 5** 11](#_3rdcrjn)

[**BAB 6** 12](#_26in1rg)

[**BAB 7** 13](#_lnxbz9)

[**BAB 7** 14](#_35nkun2)

[**BAB 8** 15](#_2jxsxqh)

[DAFTAR PUSTAKA 16](#_1ksv4uv)

[LAMPIRAN 17](#_z337ya)

[TENTANG PENULIS 18](#_44sinio)

**BAB 1**

**PENDAHULUAN**

Perkembangan teknologi informasi dan digitalisasi telah membawa berbagai perubahan signifikan serta manfaat di berbagai bidang, termasuk di bidang pendidikan. Transformasi digital telah membuat berbagai layanan menjadi lebih cepat, efisien, dan mudah diakses. Salah satu contoh implementasi teknologi ini adalah sistem peminjaman buku perpustakaan di dalam kampus berbasis website yang memanfaatkan teknologi World Wide Web(WWW) atau Internet sebagai media utama sebagai penyedia layanan.

Dengan teknologi tersebut, sistem perpustakaan dapat dikembangan menjadi berbasis website yang memungkinkan pengguna untuk dengan lebih mudah melihat informasi terkait dengan referensi buku, memeriksa ketersediaan buku, serta melakukan reservasi secara online. Sistem ini juga memberikan kemudahan bagi pengelola perpustakaan dalam memantau peminjaman buku, meningkatkan akurasi pencatatan, serta meminimalisir potensi "human error". Selain itu, data yang didapatkan dari sistem ini dapat digunakan untuk mempermudah dalam menganalisis terkait dengan tren peminjaman buku sehingga memberikan dasar yang lebih kuat untuk mengambil keputusan dalam penambahan koleksi buku yang relevan bagi pengunjung ataupun Mahasiswa.

Dengan adanya sistem ini, pengunjung atau pengguna perpustakaan dapat menikmati layanan yang lebih fleksibel dalam melakukan reservasi, sementara pengelola perpustakaan dapat dengan mudah memantau proses peminjaman dan lebih mudah dalam menganalisis buku-buku yang paling banyak dipinjam.

**BAB 2**

**KONSEP DAN ANALISIS SISTEM**

## Perancangan

Perancangan menurut KBBI adalah proses, perbuatan merencanakan, atau program. Kata perancangan berasal dari kata "rancang" yang berarti mengatur segala sesuatu sebelum bertindak. Namun perancangan dalam konteks software design adalah proses menyusun, menggambarkan, dan mengorganisasi elemen-elemen menjadi sebuah sistem yang terintegrasi dan berfungsi secara utuh. Roger S. Pressman menjelaskan dalam bukunya “Software engineering : A Practitioner’s Approach” pada edisi ke-9. Roger S. Pressman menjelaskan bahwa perancangan perangkat lunak adalah proses mengubah kebutuhan pengguna menjadi sebuah rancangan awal atau blueprint yang merepresentasikan gambaran umum dari perangkat lunak yang akan dikembangkan. Proses ini masih berada pada tingkat abstraksi yang tinggi, dengan fokus pada sasaran sistem, kebutuhan data, serta pola perilaku sistem secara menyeluruh.

Tujuan dari Perancangan perangkat lunak adalah memastikan bahwa program atau sistem yang dirancang sudah sesuai dengan kebutuhan pengguna, diawali dengan analisis kebutuhan, lalu dilanjutkan dengan pembuatan desain konseptual dan desain rinci yang mencakup elemen-elemen seperti input, output, serta hubungan dalam sistem. Dengan kata lain, perancangan adalah untuk menjembatani kebutuhan dengan implementasi, dan memastikan bahwa semua sistem berjalan selaras untuk mencapai tujuan yang telah ditetapkan.

B. Sistem Informasi

Sistem informasi (SI) adalah sistem yang terstruktur yang mengintegrasikan aspek teknis, sosial, dan organisasional untuk mendukung pengumpulan, pemrosesan, penyimpanan, dan distribusi informasi. Dalam perspektif sosioteknis, SI terdiri atas empat elemen utama : tugas, manusia, struktur organisasi, dan teknologi. Sistem ini dirancang sebagai fondasi pengambilan keputusan berbasis data dalam berbagai domain, seperti bisnis, Pendidikan, pemerintahan, dan Kesehatan.

SI berbasis komputer menggabungkan interaksi antara manusia dan perangkat teknologi dalam interpretasi informasi, mencakup perangkat lunak dan infrastruktur pendukungnya dan SI menekankan pada keterkaitan antara sistem, pengguna, proses, dan jaringan komunikasi dalam suatu kerangka terintregasi. Dan SI dibedakan dari teknologi informasi (TI) dan proses bisnis. TI lebih menekankan pada aspek teknologi, sedangkan SI fokus pada bagaimana teknologi digunakan secara strategis dalam mendukung proses organisasi. Proses bisnis adalah rangkaian aktivitas, sementara SI berperan sebagai alat kontrol terhadap aktivitas tersebut.

Tujuan utama SI adalah menghasilkan informasi yang relevan, tepat waktu, dan akurat. Informasi yang tidak memenuhi ketiga kriteria tersebut dianggap tidak bernilai. Pengembangan SI mencakup empat tahap utama: identifikasi masalah, formulasi solusi alternatif, pemilihan solusi terbaik, dan implementasi solusi terpilih. Tiga tahap awal merupakan inti dari proses analisis sistem.

C. Website

Situs web merujuk pada sekumpulan halaman digital yang saling terkoneksi dan umumnya disimpan dalam satu sistem peladen yang sama. Konten yang ditampilkan dapat berasal dari individu, kelompok, maupun entitas organisasi, dan dirancang untuk menyampaikan informasi secara daring. Akses terhadap situs ini dilakukan melalui jaringan komputer seperti Internet atau jaringan lokal (LAN), menggunakan alamat unik yang dikenal sebagai URL (Uniform Resource Locator).

Halaman dalam situs web pada dasarnya adalah dokumen teks biasa yang dibentuk menggunakan struktur markup seperti HTML. Tak jarang, komponen ini turut dilengkapi oleh skrip pemrograman guna memperkaya interaktivitas. Dokumen tersebut kemudian ditafsirkan oleh aplikasi peramban (browser) dan ditampilkan sebagai antarmuka visual bagi pengguna.

Cikal bakal situs web diciptakan oleh Sir Tim Berners-Lee, dengan tujuan awal untuk memfasilitasi pertukaran data dan pemutakhiran informasi antar peneliti di lembaga tempat ia bekerja, yaitu CERN. Tonggak sejarah penting terjadi pada 30 April 1993, saat CERN secara resmi mengumumkan bahwa teknologi World Wide Web dapat diakses dan digunakan secara bebas oleh masyarakat luas. Sebelumnya, situs web pertama yang terhubung ke jaringan mulai beroperasi sejak tahun 1991.

Dalam praktiknya, situs web dapat dibedakan menjadi dua tipe utama: statis dan dinamis. Situs statis dirancang dengan konten tetap yang tidak diperbaharui secara rutin. Proses pengubahan informasinya dilakukan secara langsung pada file sumber. Sementara itu, situs dinamis dikembangkan dengan struktur yang memungkinkan isi dapat diperbarui secara berkala tanpa harus melakukan modifikasi manual terhadap kode. Kontennya bersifat adaptif, menyesuaikan dengan perkembangan data atau waktu. Sebagai contoh, platform berita daring biasanya mengadopsi pendekatan situs dinamis karena kebutuhan akan pembaruan informasi yang cepat dan kontinu.

D. Internet (pengertian umum internet)

Internet merupakan kependekan dari interconnected network, yang secara harfiah berarti “jaringan yang saling terhubung.” Dalam pengertian sederhana, internet adalah sebuah sistem besar yang menghubungkan jutaan bahkan miliaran komputer di seluruh dunia agar bisa saling bertukar informasi. Komputer-komputer ini saling terhubung menggunakan seperangkat aturan yang disebut protokol TCP/IP agar bisa "berkomunikasi" satu sama lain.

Awal mula internet berasal dari proyek penelitian di Amerika Serikat pada tahun 1960-an. Saat itu, pemerintah ingin membangun sistem komunikasi antar komputer yang tetap bisa berjalan meskipun terjadi gangguan, seperti dalam situasi perang. Dari situlah lahir jaringan pertama yang disebut ARPANET, yaitu jaringan komputer eksperimental yang menghubungkan beberapa universitas dan lembaga penelitian di Amerika. Seiring waktu, teknologi ini dikembangkan lebih lanjut dan mulai digunakan di luar lingkup militer dan akademik. Pada tahun 1980-an, proyek pengembangan jaringan didukung oleh lembaga pemerintah dan sektor swasta, sehingga banyak jaringan komputer regional mulai saling terhubung. Masuknya perusahaan-perusahaan komersial pada awal 1990-an menjadi titik penting dalam sejarah, karena sejak saat itu internet mulai berkembang pesat dan terbuka untuk masyarakat umum.

Peran Internet dalam Kehidupan Modern

Internet telah membawa perubahan besar dalam berbagai aspek kehidupan. Banyak sarana komunikasi tradisional seperti surat, telepon kabel, radio, dan televisi telah bertransformasi menjadi layanan digital. Contohnya, kini kita bisa mengirim pesan melalui email, melakukan panggilan video, menonton televisi secara daring, hingga mendengarkan musik melalui platform streaming

Internet tidak memiliki satu lembaga pusat yang mengatur seluruh sistemnya. Masing-masing jaringan dan penyedia layanan internet menetapkan kebijakan operasionalnya sendiri. Namun, untuk menjaga keteraturan dan kompatibilitas teknis antar perangkat dan jaringan, terdapat beberapa organisasi internasional yang berperan penting. Seperti ICANN (Internet Corporation for Assigned Names and Numbers), yang bertanggung jawab mengatur sistem penamaan domain dan alamat IP agar setiap situs web memiliki alamat unik. IETF (Internet Engineering Task Force), yang bertugas merancang dan mengembangkan standar teknis agar berbagai perangkat dan sistem bisa saling terhubung dengan lancar.

Secara umum, internet adalah infrastruktur global yang memungkinkan komunikasi dan pertukaran informasi antar komputer di seluruh dunia. kini internet telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari, baik dalam bidang pendidikan, bisnis, hiburan, maupun interaksi sosial. Internet telah membawa perubahan besar dalam berbagai aspek kehidupan. Banyak sarana komunikasi tradisional seperti surat, telepon kabel, radio, dan televisi telah bertransformasi menjadi layanan digital

E. Uniform Resource Locator (URL)

URL (Uniform Resource Locator), atau dalam bahasa sehari-hari sering disebut sebagai alamat web, adalah cara untuk menunjukkan lokasi suatu sumber daya (seperti halaman web, gambar, atau file) di jaringan komputer, serta bagaimana cara mengaksesnya. Dengan kata lain, URL berfungsi seperti alamat rumah di dunia maya, agar komputer tahu ke mana harus pergi untuk menemukan sesuatu di internet. URL merupakan bagian dari kelompok yang lebih luas yang disebut URI (Uniform Resource Identifier). URL digunakan untuk mengakses halaman web dengan protokol HTTP atau HTTPS. Namun, URL juga digunakan dalam berbagai layanan lain, seperti, mengunduh file melalui FTP, Mengirim email, dan Mengakses basis data menggunakan JDBC.

URL pertama kali didefinisikan secara resmi dalam dokumen RFC 1738 pada tahun 1994. Standar ini dikembangkan oleh Tim Berners-Lee, pencipta World Wide Web, bersama kelompok kerja URI dari Internet Engineering Task Force (IETF). Sebelumnya, ide tentang pengidentifikasi dokumen ini sudah dibahas sejak tahun 1992 dalam forum-forum diskusi teknis. Format URL sendiri menggabungkan sistem nama domain (yang sudah diperkenalkan pada tahun 1985) dengan struktur jalur file, di mana simbol garis miring (/) digunakan untuk memisahkan folder dan nama file. Pada awal pengembangannya, istilah yang diusulkan adalah Universal Document Identifier (UDI) atau Universal Resource Locator, namun akhirnya istilah tersebut diubah menjadi Uniform Resource Locator, yang kemudian lebih dikenal luas hingga sekarang. Berners-Lee sendiri mengaku lebih menyukai kata “universal” karena menggambarkan jangkauan global dari sistem ini.

Secara singkat, URL adalah alat penting dalam dunia internet yang memungkinkan pengguna untuk mengakses berbagai sumber daya secara cepat dan terstruktur. Meskipun sederhana dalam penggunaannya, URL memiliki sejarah teknis dan evolusi yang cukup panjang dalam dunia pengembangan web dan jaringan komputer.

G. JavaScript

JavaScript adalah salah satu bahasa pemrograman utama yang digunakan dalam pengembangan situs web. Bahasa ini berperan dalam membuat halaman web menjadi interaktif dan responsif, seperti menampilkan notifikasi, memvalidasi formulir, atau memperbarui isi halaman tanpa harus memuat ulang seluruh laman.

JavaScript berjalan langsung di dalam web browser, seperti Google Chrome atau Mozilla Firefox, sehingga pengguna bisa berinteraksi dengan halaman web secara dinamis. Selain itu, JavaScript juga bisa digunakan di luar browser, misalnya untuk membangun aplikasi server melalui platform seperti Node.js. Berbeda dengan bahasa pemrograman lain, JavaScript bersifat Berorientasi objek, artinya dapat digunakan untuk berbagai pendekatan pemrograman JavaScript juga menjadi bagian dari standar global yang disebut ECMAScript, yang terus diperbarui agar sesuai dengan kebutuhan zaman dan hampir semua situs web modern menggunakan JavaScript, menjadikannya salah satu teknologi cukup penting dalam dunia digital saat ini.

H. jQuery

jQuery adalah sebuah pustaka (library) dari bahasa pemrograman JavaScript yang dirancang untuk memudahkan proses pengembangan halaman web, khususnya dalam mengelola dan memanipulasi struktur HTML (DOM), menangani interaksi pengguna (event), mengatur animasi CSS, serta melakukan komunikasi data secara asinkron (Ajax).

jQuery pertama kali dikembangkan pada tahun 2006 oleh John Resig, dan telah menjadi salah satu teknologi web yang paling banyak digunakan dan memiliki pengaruh besar terhadap perkembangan pustaka JavaScript lainnya. Hingga kini, jQuery tetap menjadi pilihan utama dalam menciptakan halaman web yang dinamis dan responsif, terutama bagi pengembang yang ingin bekerja lebih efisien tanpa harus menulis kode JavaScript dari awal.

Dengan sintaks yang ringkas dan mudah dipahami, jQuery memungkinkan pengembang untuk dengan cepat mencari elemen dalam halaman, mengubah tampilannya, serta merespons berbagai aksi pengguna seperti klik atau hover. Keunggulan lainnya adalah kemampuan untuk membuat plugin, sehingga pengembang dapat memperluas fungsionalitas jQuery sesuai kebutuhan.

I. Framework Laravel

Laravel adalah sebuah framework berbasis PHP yang digunakan untuk membangun aplikasi web secara efisien dan terstruktur. Laravel menerapkan pendekatan Model-View-Controller (MVC), yang membantu pengembang memisahkan logika, tampilan, dan pengelolaan data dalam aplikasi agar lebih rapi dan mudah dikembangkan.

Framework ini dibuat oleh Taylor Otwell, dengan tujuan memberikan pengalaman pengembangan yang lebih modern dan lengkap dibandingkan dengan framework PHP lainnya, seperti CodeIgniter. Laravel memiliki banyak fitur unggulan, seperti Routing agar mudah dikelola, Autentikasi pengguna secara bawaan, Manajemen database yang fleksibel, Sistem templating Blade. Serta dukungan penuh untuk email, queue, storage, dan task scheduling.

Laravel juga dikenal karena sintaksnya yang bersih dan ekspresif, sehingga cocok untuk pemula maupun profesional. Framework ini bersifat gratis dan open-source, serta didukung oleh komunitas yang besar dan aktif. Laravel terus diperbarui secara berkala untuk mengikuti perkembangan teknologi web, dan saat ini telah mencapai versi Laravel 11.

J. Database

Database adalah sekumpulan data yang saling berkaitan dan disusun untuk mewakili informasi nyata dari dunia sekitar. Data tersebut bisa berupa nama, alamat, nomor telepon, atau informasi lainnya yang memiliki arti dan bisa disimpan untuk digunakan kembali. Contohnya, daftar kontak yang kita simpan di ponsel atau spreadsheet Excel merupakan bentuk sederhana dari database.

Umumnya, database digunakan untuk menggambarkan suatu bagian dari dunia nyata (sering disebut miniworld), dan data di dalamnya akan diperbarui setiap kali ada perubahan dalam dunia nyata tersebut. Sebuah database tidak hanya sekadar kumpulan data acak, tapi harus memiliki struktur yang logis, tujuan yang jelas, dan dapat dimanfaatkan oleh pengguna tertentu untuk keperluan tertentu. Ukuran database bisa sangat beragam, dari yang kecil seperti daftar pelanggan toko, hingga yang sangat besar seperti basis data perpajakan milik pemerintah atau data produk dan transaksi di situs besar seperti Amazon. Database besar semacam itu bisa berisi ratusan juta entri, yang harus dapat diakses, dicari, dan diperbarui dengan cepat dan efisien.

Untuk mengelola database secara efisien, kita menggunakan perangkat lunak yang disebut Database Management System (DBMS). DBMS adalah sistem perangkat lunak yang memungkinkan pengguna untuk membuat struktur database (definisi data dan aturan), menyimpan dan menyusun data, mengakses dan memanipulasi data (misalnya mencari, memperbarui, atau menghapus), dan Berbagi akses ke database di antara banyak pengguna atau aplikasi, Serta menjaga keamanan dan integritas data.

K. CRUD

CRUD adalah singkatan dari Create, Read, Update, dan Delete, yang merupakan empat fungsi dasar dalam pengelolaan data secara permanen di berbagai jenis basis data, baik itu basis data relasional seperti MySQL dan PostgreSQL, maupun basis data NoSQL seperti MongoDB dan DynamoDB. Konsep ini sangat penting dalam pengembangan aplikasi karena mencerminkan cara sistem menyimpan, menampilkan, mengubah, dan menghapus data sesuai kebutuhan pengguna.

Sebagai contoh sederhana, ketika seseorang mendaftarkan akun dalam sebuah aplikasi, sistem akan membuat data pengguna baru—itulah proses Create. Saat pengguna membuka halaman daftar produk atau layanan, sistem melakukan proses Read untuk menampilkan informasi tersebut. Jika kemudian pengguna mengubah alamat atau data pribadi lainnya, maka sistem melakukan proses Update. Dan ketika pengguna memutuskan untuk menghapus akun, sistem menjalankan perintah Delete agar data tersebut dihilangkan dari penyimpanan.

Keempat proses ini biasanya diatur melalui perintah SQL dalam basis data relasional atau melalui fungsi-fungsi khusus dalam sistem NoSQL. Dalam praktiknya, perintah CRUD ini tidak selalu dijalankan secara manual oleh pengembang atau administrator, melainkan telah ditanamkan ke dalam kode program agar dapat berjalan otomatis ketika pengguna melakukan interaksi tertentu di dalam aplikasi.

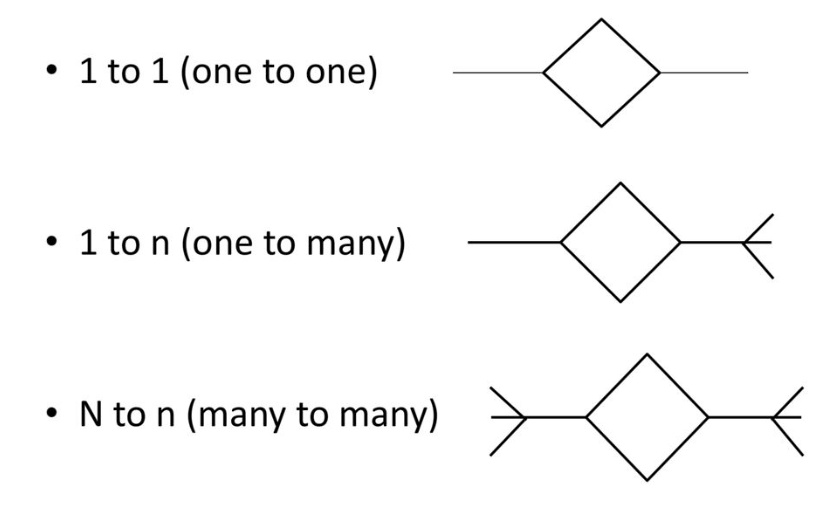
Secara keseluruhan, CRUD menjadi pondasi penting dalam pengelolaan data karena seluruh siklus hidup data dalam sebuah aplikasi digital bergantung padanya. Pemahaman yang baik terhadap konsep ini sangat membantu dalam merancang sistem yang efisien, aman, dan mampu beroperasi secara optimal, terutama ketika harus menangani banyak pengguna dalam waktu bersamaan. Desain basis data yang buruk dapat menghambat proses CRUD dan berdampak langsung pada performa sistem secara keseluruhan.

L. ERD

ERD, atau Entity Relationship Diagram, adalah sebuah diagram yang digunakan untuk merancang struktur basis data secara visual. Diagram ini menggambarkan objek-objek utama (disebut entitas) dalam sistem dan bagaimana hubungan antar objek tersebut (relationship) terbentuk. Karena itu, diagram ini dinamakan "Entity Relationship". Yaitu hubungan antar entitas.

Berikut adalah simbol-simbol dari ERD :

1. Simbol Kardinalitas



Gambar 1. Simbol-simbol Kardinallitas

One to One (1:1)

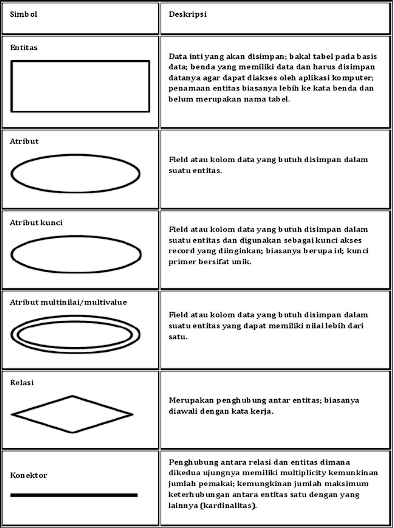
Setiap elemen dalam entitas A terhubung hanya dengan satu elemen di entitas B, dan hal tersebut juga berlaku sebaliknya.

One to Many (1:M / Many)

Satu elemen dari entitas A bisa memiliki relasi dengan beberapa elemen dari entitas B, namun elemen di entitas B hanya terhubung ke satu elemen di entitas A.

Many to Many (M:M)

Setiap elemen dalam entitas A bisa memiliki relasi dengan banyak elemen di entitas B, dan begitu pula sebaliknya — elemen di entitas B dapat terhubung dengan banyak elemen di entitas A.



Gambar 2. Simbol ERD

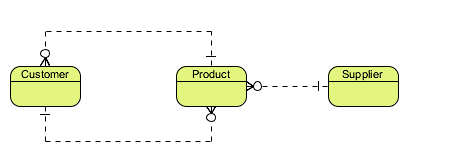
Gambar 3. Contoh ERD konseptual

Diagram ini menggambarkan hubungan antara tiga entitas utama dalam sebuah sistem, yaitu: Customer (Pelanggan), Product (Produk), dan Supplier (Pemasok). Setiap entitas digambarkan sebagai persegi panjang berwarna hijau muda. Hubungan antar entitas ditunjukkan melalui garis dengan simbol khusus yang mewakili jenis dan kardinalitas (jumlah relasi) antara entitas.

Pada diagram ERD tersebut, terdapat 3 entitas :

1. Entitas: Customer

Entitas ini mewakili pelanggan dalam sistem dan terhubung ke entitas Product dengan garis putus-putus dan simbol “lingkaran kaki gagak” di kedua sisi. Ini menunjukkan bahwa seorang Customer dapat membeli banyak Product, dan sebuah Product dapat dibeli oleh banyak Customer. Artinya, hubungan antara Customer dan Product adalah Many-to-Many (banyak ke banyak).

2. Entitas: Product

Entitas ini mewakili produk yang tersedia di dalam sistem. Produk menjadi penghubung antara pelanggan dan pemasok. Produk dapat memiliki banyak relasi dengan Customer (seperti dijelaskan di atas). Satu hubungan opsional ke Supplier, yang artinya sebuah produk boleh memiliki satu pemasok, atau tidak memiliki pemasok sama sekali. Dan sebaliknya, satu Supplier bisa menyuplai banyak Product.

3. Entitas: Supplier

Mewakili pemasok atau penyuplai barang. Dihubungkan ke Product dengan garis putus-putus dan simbol “lingkaran” di sisi Product dan “kaki gagak” di sisi Supplier. Ini menunjukkan hubungan Optional-to-Many yang mana satu supplier dapat menyuplai banyak produk, tetapi tidak semua produk harus memiliki supplier (misalnya produk hasil buatan internal sendiri).

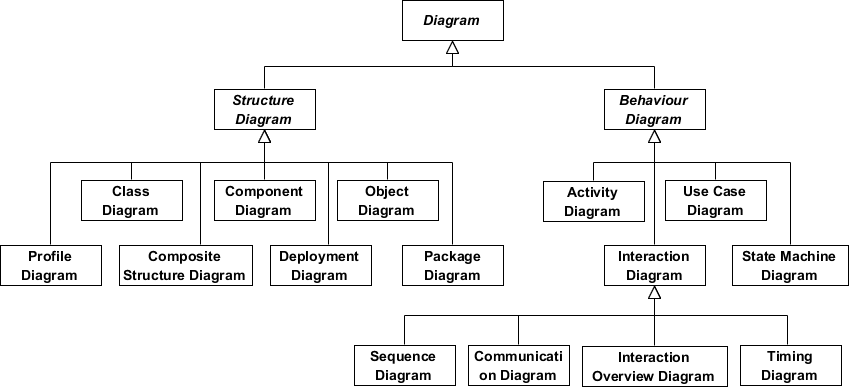
M. UML (Use Case, Activity, Sequence, Class)

Unified Modeling Language (UML) merupakan bahasa pemodelan yang dirancang sebagai standar untuk memvisualisasikan, mendokumentasikan, mendefinisikan, dan membangun sistem perangkat lunak secara menyeluruh. Tujuan utamanya adalah untuk menciptakan pemahaman yang selaras di antara seluruh anggota tim pengembang mengenai cara kerja suatu sistem. Berbeda dengan flowchart yang lebih sederhana dan cocok untuk menggambarkan alur proses dasar, UML memberikan gambaran yang lebih rinci dan terstruktur, sehingga lebih sesuai untuk sistem yang kompleks dan membutuhkan spesifikasi teknis yang jelas.

Jika ditelusuri dari sejarahnya, UML lahir dari kolaborasi tiga tokoh besar dalam dunia rekayasa perangkat lunak. Dimulai pada tahun 1994, ketika Jim Rumbaugh meninggalkan General Electric dan bergabung dengan Grady Booch di Rational Corporation untuk menyatukan metode pemodelan yang mereka kembangkan secara terpisah. Proyek awal ini dikenal sebagai "Unified Method." Setahun kemudian, pada tahun 1995, Ivar Jacobson ikut bergabung dan menyumbangkan konsep penting mengenai "Use Case." Kolaborasi ini kemudian melahirkan UML seperti yang kita kenal saat ini. Karena kontribusi besar mereka, Rumbaugh, Booch, dan Jacobson dijuluki sebagai “Three Amigos.”

Dalam praktik pengembangan sistem, berbagai pihak akan menggunakan UML dengan cara yang berbeda sesuai dengan kebutuhan mereka. Seorang programmer, misalnya, akan menggunakan diagram UML sebagai acuan saat menulis kode, sedangkan seorang technical writer akan lebih fokus pada representasi sistem secara umum untuk keperluan dokumentasi. Oleh karena itu, UML menyediakan beragam jenis diagram yang terbagi ke dalam dua kategori utama: Diagram Struktur dan Diagram Perilaku (Behavioural Diagram). Diagram Struktur menampilkan aspek statis sistem, seperti hubungan antar kelas dan komponen. Sementara Diagram Perilaku lebih menekankan pada dinamika interaksi dan alur kerja antar elemen dalam sistem.

Berikut ini adalah ilustrasi hierarki berbagai jenis diagram dalam UML:



Gambar 4. Hierarki dari berbagai jenis UML diagram

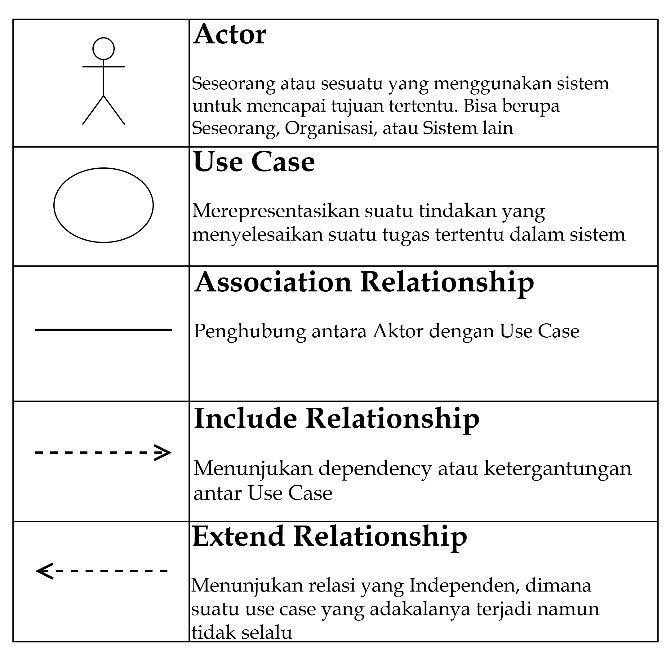
Pada gambar di atas dapat dilihat bahwa UML memiliki berbagai jenis diagram yang diklasifikasikan ke dalam dua kategori utama, yaitu Structure Diagram dan Behaviour Diagram. Masing-masing jenis diagram memiliki fungsi dan tujuan tersendiri dalam proses pengembangan perangkat lunak.

1. Use Case Diagram

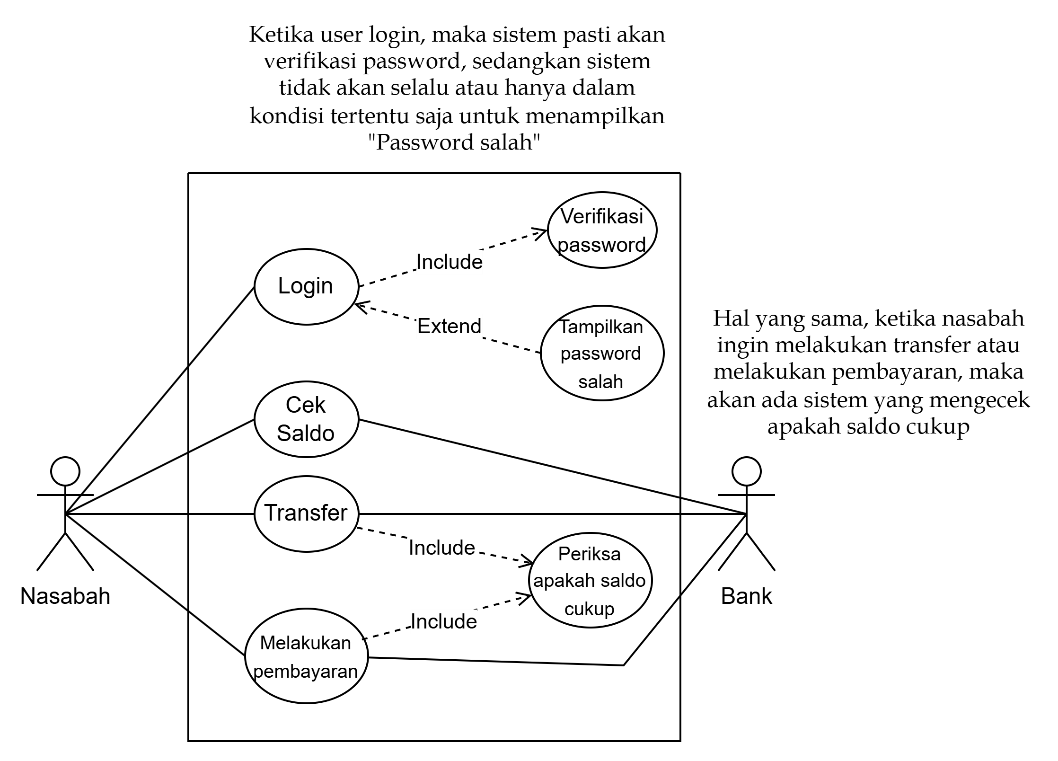
Use Case Diagram adalah salah satu diagram pada UML yang berfungsi untuk memvisualisasikan bagaimana pengguna (disebut aktor) akan berinteraksi dengan sistem melalui berbagai aktivitas atau fungsi yang tersedia (disebut use case). Fokus utama dari use case diagram bukan pada bagaimana sistem bekerja secara teknis, tetapi lebih kepada perilaku sistem yang bisa dilihat dan digunakan oleh pihak luar.

Tujuan dari use case diagram adalah untuk menyampaikan bagaimana sistem akan berperilaku dalam menjawab kebutuhan pengguna, dengan cara yang sederhana dan mudah dipahami. Diagram ini biasanya tidak terlalu kompleks, cukup menunjukkan siapa saja yang berinteraksi dengan sistem, fungsi apa saja yang tersedia, serta hubungan antar elemen tersebut. Perlu dicatat bahwa use case diagram tidak menjelaskan langkah-langkah rinci dari tiap fungsi, dan tidak menunjukkan urutan proses secara detail.

Gambar dibawah ini adalah symbol-simbol pada Use Case Diagram :



Gambar 5. Contoh sederhana Use Case Diagram



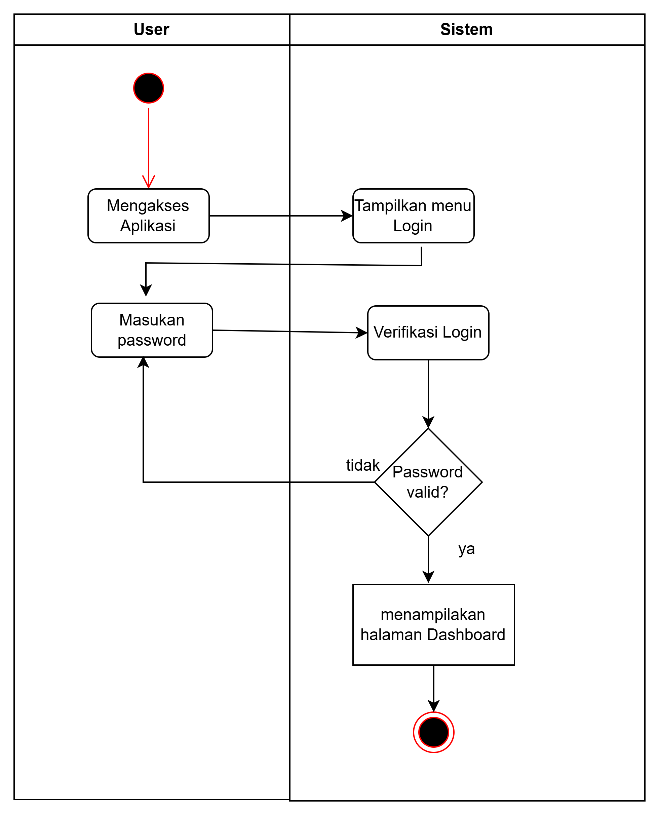
Gambar 6.

Pada gambar di atas, ada dua aktor, yaitu Primary Actor (Nasabah) dan Secondary Actor (Bank). Pada gambar tersebut menjelaskan Nasabah bisa melakukan login, cek saldo, transfer, dan melakukan pembayaran.

2. Activity Diagram

Activity Diagram merupakan salah satu jenis diagram penting dalam Unified Modeling Language (UML) yang berfungsi untuk merepresentasikan aspek dinamis dari suatu sistem. Diagram ini digunakan untuk menunjukkan alur aktivitas, arus data, kendali proses, serta urutan langkah-langkah yang terjadi dalam sistem atau aplikasi. Dengan diagram ini, kita dapat memahami bagaimana berbagai aktivitas saling berkaitan dan berkontribusi terhadap pencapaian suatu output atau hasil akhir yang diinginkan.

Berikut ini disajikan contoh sederhana dari Activity Diagram untuk menggambarkan konsep tersebut dengan lebih jelas.



Gambar 7. Contoh sederhana Activity Diagram

Activity Diagram di atas menggambarkan alur proses login pada sebuah aplikasi, dimulai dari interaksi antara pengguna (user) dan sistem. Proses dimulai ketika pengguna mengakses aplikasi. Setelah itu, sistem secara otomatis menampilkan halaman login yang berisi form untuk masuk ke dalam sistem.

Langkah berikutnya, pengguna diminta untuk memasukkan kata sandi (password). Setelah itu, sistem melakukan proses verifikasi terhadap data login yang telah dimasukkan. Sistem kemudian memeriksa apakah kata sandi yang diberikan valid atau tidak. Jika kata sandi benar, maka sistem akan mengarahkan pengguna ke halaman dashboard, yang menjadi tujuan utama setelah berhasil login.

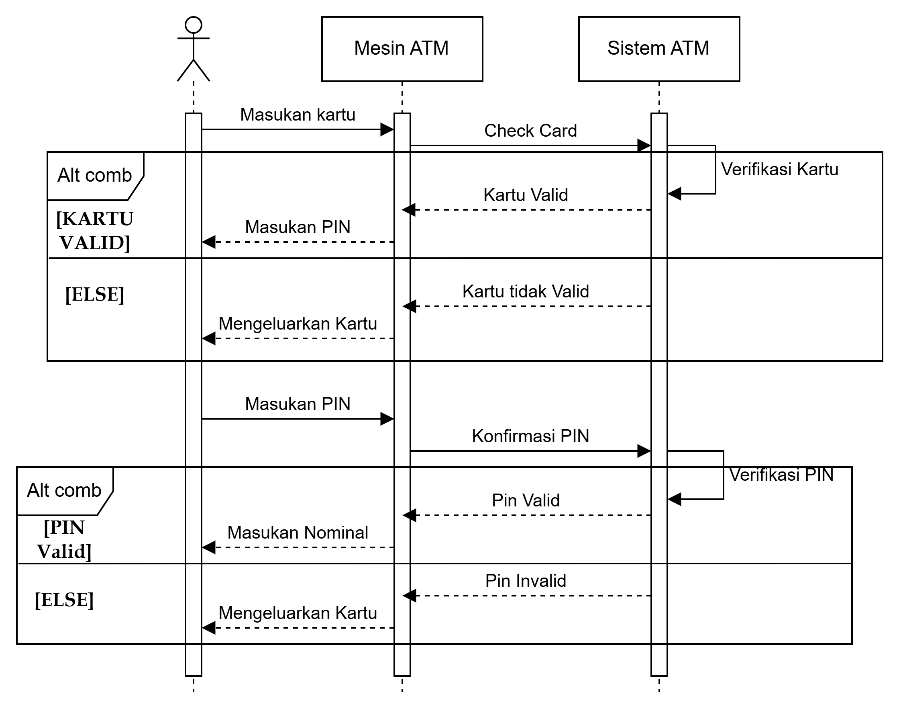
Diagram ini tidak hanya memperlihatkan urutan aktivitas yang terjadi, namun juga membedakan peran antara pengguna dan sistem dengan jelas. Jalur aktivitas dibagi menjadi dua bagian, yaitu, satu untuk pengguna dan satu lagi untuk sistem. Sehingga mempermudah pembacaan dan pemahaman peran masing-masing pihak dalam proses login.

Secara keseluruhan, diagram ini membantu menjelaskan bagaimana interaksi antara user dan sistem terjadi dalam proses masuk ke dalam aplikasi, serta memperlihatkan logika pengambilan keputusan berdasarkan validitas data yang dimasukkan.

3. Sequence Diagram

Sequence Diagram merupakan salah satu bentuk dari Interaction Diagram yang berfungsi untuk menggambarkan alur komunikasi antar objek secara berurutan dalam suatu sistem. Dengan memanfaatkan diagram ini, kita dapat melihat secara jelas bagaimana objek-objek di dalam sistem saling berinteraksi untuk menyelesaikan sebuah proses. Diagram ini sangat berguna, baik bagi pengembang perangkat lunak maupun kalangan profesional di bidang bisnis, karena membantu dalam memahami cara kerja sistem serta mendokumentasikan skenario atau proses bisnis secara terstruktur dan visual.

Sebagai ilustrasi, berikut ini disajikan contoh sederhana dari implementasi Sequence Diagram pada sistem ATM.



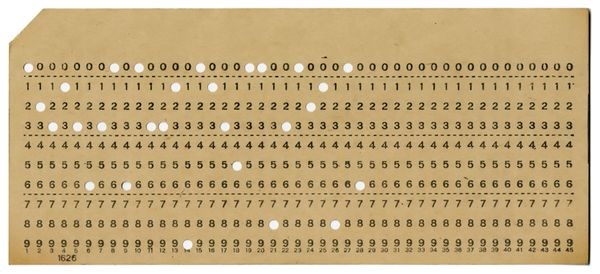
Gambar 8. Contoh sederhana Sequence Diagram

Pada ilustrasi diagram di atas, proses dimulai ketika nasabah memasukkan kartu ke dalam mesin ATM. Selanjutnya, sistem secara otomatis menjalankan proses verifikasi untuk memastikan keabsahan kartu tersebut. Apabila kartu dinyatakan valid, sistem akan meminta pengguna untuk memasukkan PIN sebagai langkah keamanan berikutnya. Namun, apabila kartu tidak valid, maka sistem akan segera mengeluarkan kartu dari mesin sebagai respons terhadap ketidaksesuaian data.

N. User Interface

User Interface (UI) adalah antarmuka atau perantara antara manusia dan mesin. Termasuk komputer, yang memungkinkan terjadinya interaksi. Tujuannya adalah agar manusia dapat memberikan perintah kepada mesin atau komputer, dan mesin tersebut dapat memberikan umpan balik atau respons terhadap perintah tersebut dalam bentuk yang dapat dimengerti manusia.

Komputer pada dasarnya hanya memahami bahasa biner, yaitu angka 0 dan 1. Bahasa ini sangat tidak intuitif bagi manusia. Misalnya, untuk menyampaikan satu instruksi sederhana kepada komputer, dibutuhkan serangkaian angka biner yang sangat panjang dan sulit dipahami oleh pengguna biasa. Oleh karena itu, diperlukan suatu jembatan komunikasi yang mengubah interaksi manusia yang kompleks menjadi instruksi yang bisa dimengerti oleh mesin, dan sebaliknya. Di sinilah peran User Interface untuk menerjemahkan keinginan dan perintah manusia menjadi sinyal-sinyal yang dapat diproses oleh komputer, serta menerjemahkan hasil dari komputer menjadi tampilan, suara, atau bentuk lain yang dapat dimengerti oleh manusia.



Gambar 5. Punch Card (Bentuk antarmuka paling awal)

Seiring perkembangan teknologi, user interface mengalami evolusi yang sangat panjang dan menarik. Pada masa awal komputer, interaksi dilakukan menggunakan batch interface yang sangat terbatas. Pengguna harus memasukkan instruksi melalui kartu berlubang (punch card) dan menunggu prosesnya dalam waktu yang cukup lama, tanpa adanya interaksi langsung.



Gambar 6. Command line Interface (Antarmuka pada tahun 1980an)

Kemudian, muncul command-line interface yang menggunakan teks. Sistem ini lebih cepat, namun tetap sulit digunakan oleh orang awam karena membutuhkan hafalan sintaks perintah yang cukup kompleks.

Setelah itu, mulai muncul antarmuka berbasis teks yang lebih standar dan konsisten, seperti sistem menu dengan shortcut umum, sehingga lebih mudah diakses oleh pengguna biasa. Revolusi besar terjadi saat graphical user interface atau GUI diperkenalkan, yang menghadirkan ikon, tombol, jendela, dan mouse sebagai alat bantu visual. Perkembangan ini membuat komputer jauh lebih mudah digunakan oleh semua kalangan. Sampai akhirnya, dunia menyaksikan kemunculan UI modern yang semakin interaktif dan canggih, seperti antarmuka sentuh di smartphone, perintah suara melalui asisten digital, hingga realitas virtual dan augmented reality yang membawa pengalaman pengguna ke level baru. Bahkan sekarang, teknologi sudah memungkinkan otak manusia berinteraksi langsung dengan mesin melalui brain-computer interface, tanpa perantara fisik seperti keyboard atau layar.

**BAB 3**

**ANALISA KEBUTUHAN SISTEM**

A. Spesifikasi Kebutuhan Sistem dan Analisis Kebutuhan Sistem

Dalam proses penentuan kebutuhan perangkat lunak, menyusun daftar kebutuhan untuk suatu proyek merupakan langkah yang sangat penting, bahkan bisa dikatakan vital. Mungkin terdengar seolah-olah pelanggan sudah memahami apa yang mereka inginkan, atau bahwa pengguna akhir sudah mengetahui fitur dan fungsi yang dibutuhkan. Namun, kenyataannya, dalam banyak situasi, hal tersebut tidak sepenuhnya benar. Oleh karena itu, penting untuk terlebih dahulu mendefinisikan kebutuhan perangkat lunak, agar dapat memahami dengan jelas apa yang diinginkan oleh pelanggan sebelum proses perancangan dan pembangunan perangkat lunak dimulai.

Proses rekayasa kebutuhan biasanya diawali dengan tahap untuk mendefinisikan ruang lingkup serta karakteristik masalah yang akan diselesaikan. Setelah itu dilanjutkan dengan tahap yang berfokus pada upaya menggali dan mengidentifikasi kebutuhan dari para pemangku kepentingan. Tahap berikutnya adalah di mana kebutuhan yang telah dikumpulkan akan diperinci dan disempurnakan.

Agar proses analisis kebutuhan menjadi lebih terstruktur, kebutuhan sistem umumnya dibagi ke dalam dua kategori utama, yaitu Kebutuhan Fungsional dan Kebutuhan Non-Fungsional.

B. Analisis Kebutuhan Fungsional

Kebutuhan fungsional yang dirancang untuk perancangan website sistem perpustakaan dengan php menggunakan Greenfoot, adalah sebagai berikut :

1. Memberikan fitur bagi user untuk mencari dan melihat daftar buku yang tersedia di perpustakaan.
2. Memungkinkan user untuk melakukan peminjaman buku berdasarkan kategori atau jenis buku yang diinginkan.
3. Menyediakan fitur pemberian notifikasi kepada user terkait status peminjaman, seperti pengingat pengembalian
4. Mengatur hak akses admin untuk menambah, mengedit, atau menghapus data buku di sistem.
5. Menyediakan fitur untuk admin memantau dan mengatur status peminjaman buku (dipinjam, dikembalikan, terlambat, dan lainya).
6. Mengatur ketersediaan buku secara dinamis berdasarkan aktivitas peminjaman dan pengembalian.
7. Menyediakan fitur pengaturan notifikasi atau reminder kepada user dalam pengembalian buku

C. Kebutuhan Non Fungsional

Non-Functional Requirements atau persyaratan non-fungsional adalah batasan yang menentukan kualitas sistem, seperti keamanan, kinerja, dan skalabilitas, yang memastikan sistem memenuhi kebutuhan pengguna. Diantaranya sebagai berikut

1. Perangkat Lunak (Software)

Perangkat lunak atau software yang kami gunakan dalam perancangan sistem manajemen perpustakaan dengan PHP adalah sebagai berikut:

1. Microsoft Windows 11 (64-bit)
2. Visual Studio Code
3. Laravel 12
4. Browser Google Chrome atau Mozilla Firefox
5. Software lainya yang mendukung perancangan game ini.

2. Perangkat Keras (Hardware)

Spesifikasi dari perangkat keras yang digunakan dalam peembuatan game FOODVENTURE “The Healthy Maze berbasis java menggunakan Greenfoot adalah sebagai berikut:

1) Asus TUF A15 Processor AMD Ryzen 5 4600H with Radeon Graphics 3 GHz

2) Memory: 16 GB

3) SSD: 512 GB

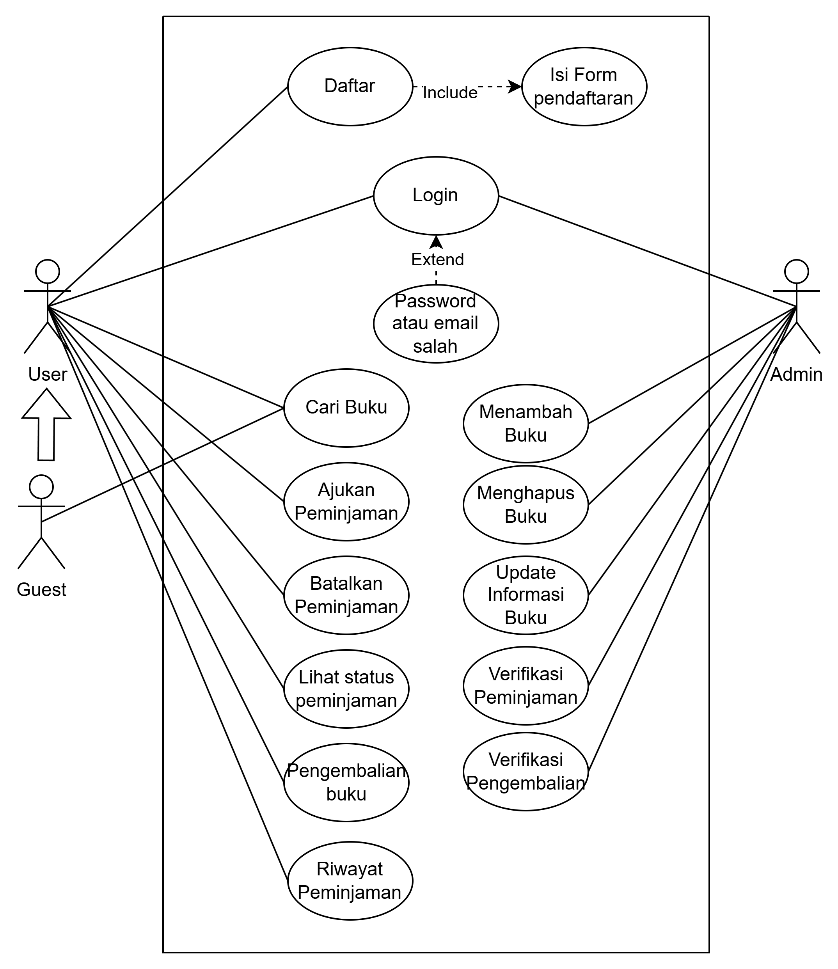
4) Monitor 15.6-inch FHD (1920 x 1080)

**BAB 4**

**PERANCANGAN SISTEM**

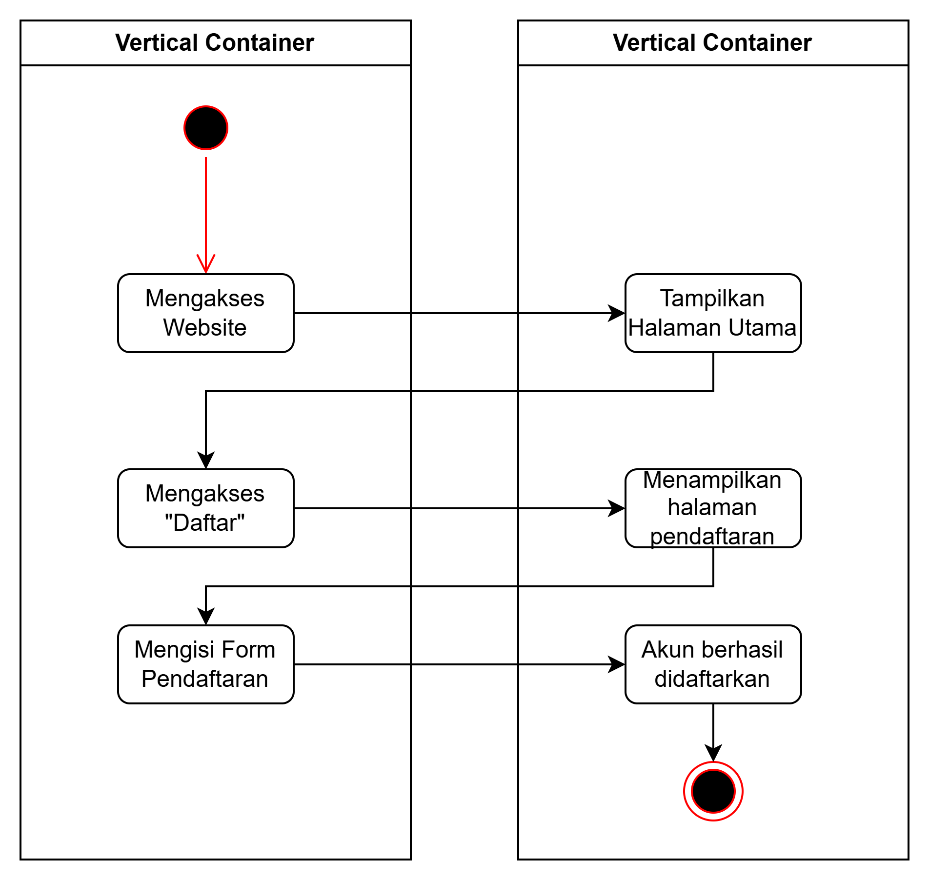
A. Rancangan Use Case Diagram

Dibawah ini adalah rancangan Use Case Diagram Sistem Perpustakaan



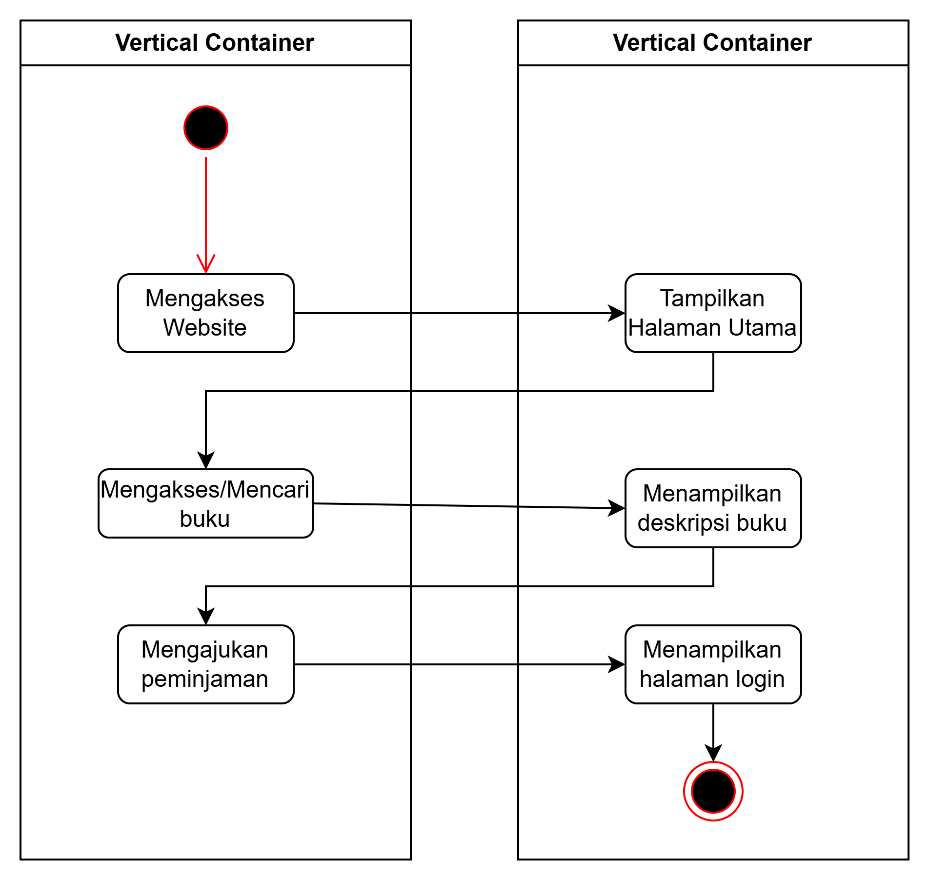
Gambar 7. Rancangan use case diagram

Pada use case diagram tersebut, ada 2 tokoh utama, yaitu user dan admin, Admin dapat login, meng-edit, menambah, ataupun delete buku, dan dapat memverifikasi peminjaman, maupun verifikasi pengembalian, Untuk user sendiri dapat daftar, login, dan mencari buku, mengajukan peminjaman, batalkan peminjaman sebelum disetujui, maupun pengembalian buku dan lain-lain. Namun guest sendiri adalah child dari user, yang melihat buku, namun jika ingin meminjam diharuskan untuk login terlebih dahulu



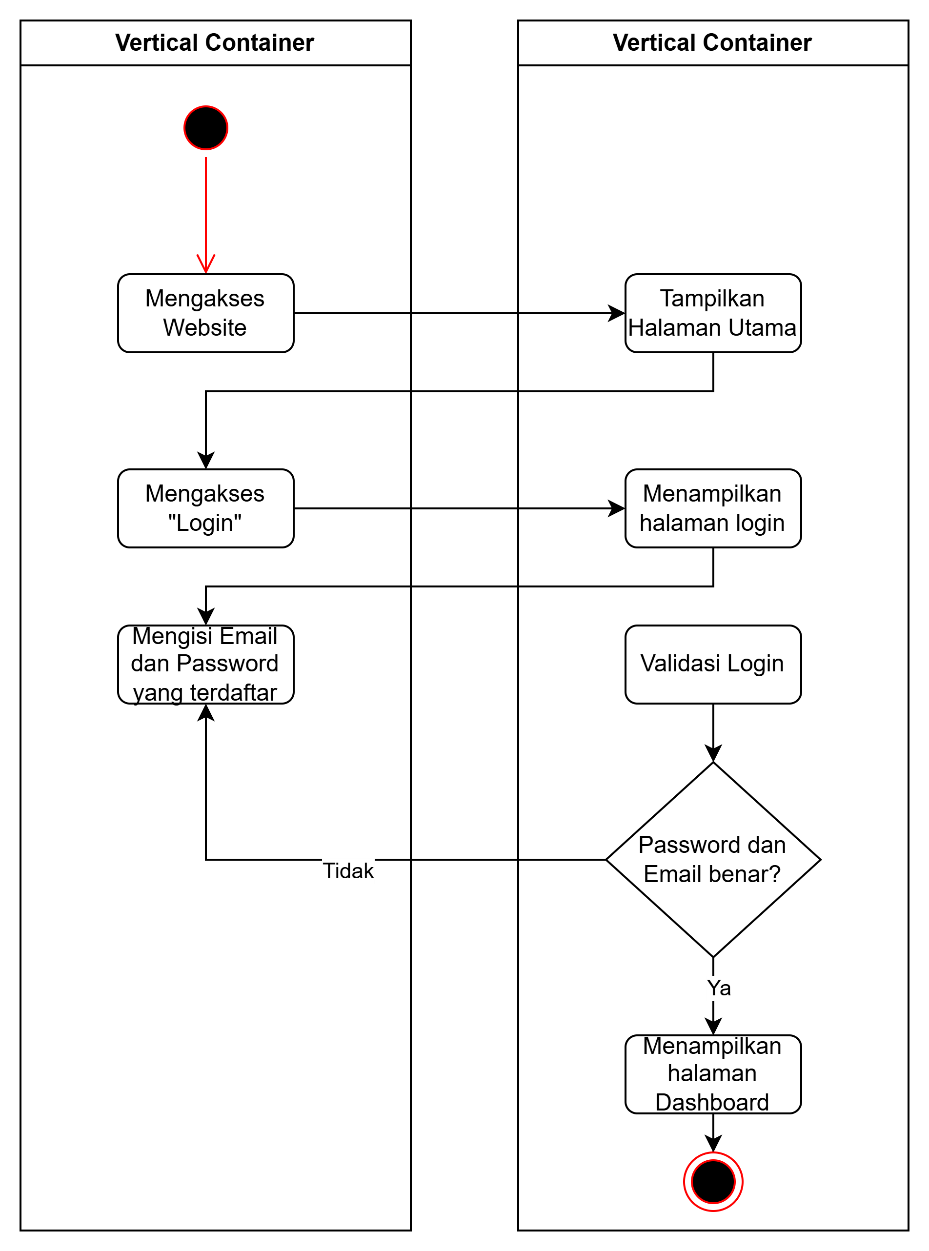
Gambar 8. Rancangan Activity diagram Pendaftaran

Rancangan Activity diagram diatas adalah Activity Diagram untuk sistem pendaftaran pada laman registrasi. Dimulai dari user yang mengakses website, kemudian sistem akan menampilkan halaman utama, dan user mengakses daftar dan mengisi form pendaftaran kemudian data user berhasil disimpan



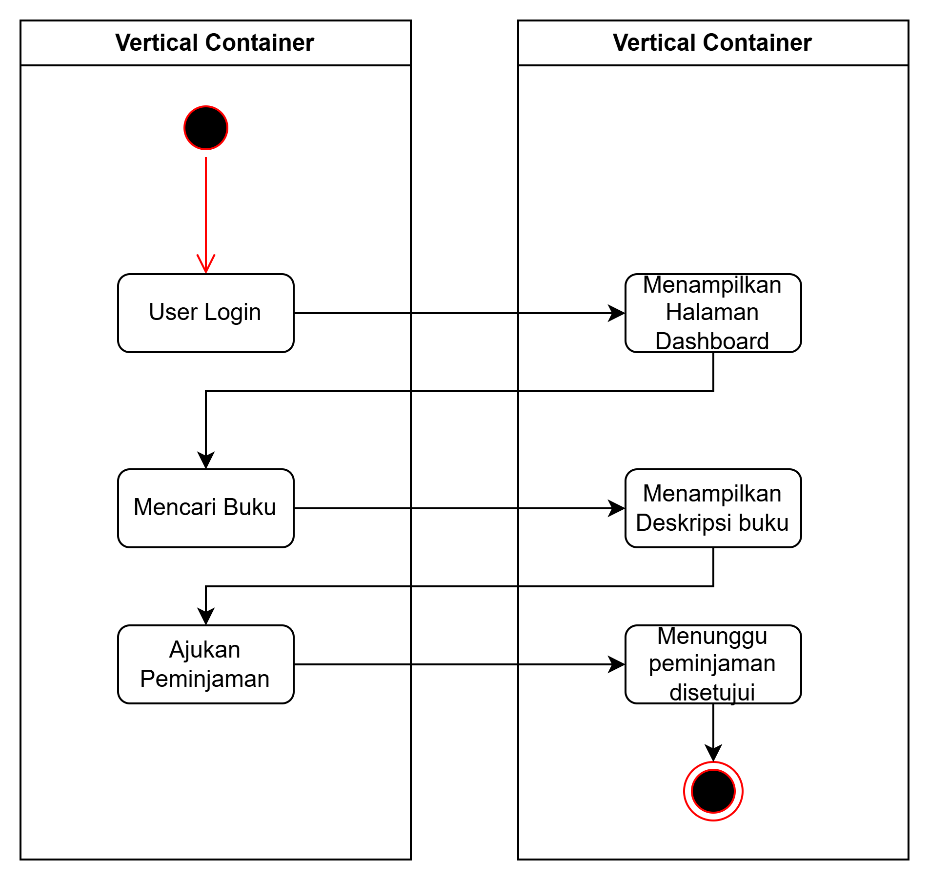
Gambar 9. Rancangan Activity Diagram pada user yang belum mendaftarkan akun

Pada Activity Diagram tersebut, User yang belum mendaftarkan akun dapat melihat dan mengakses buku-buku, namun jika ingin meminjam buku, maka harus login atau daftar terlebih dahulu.



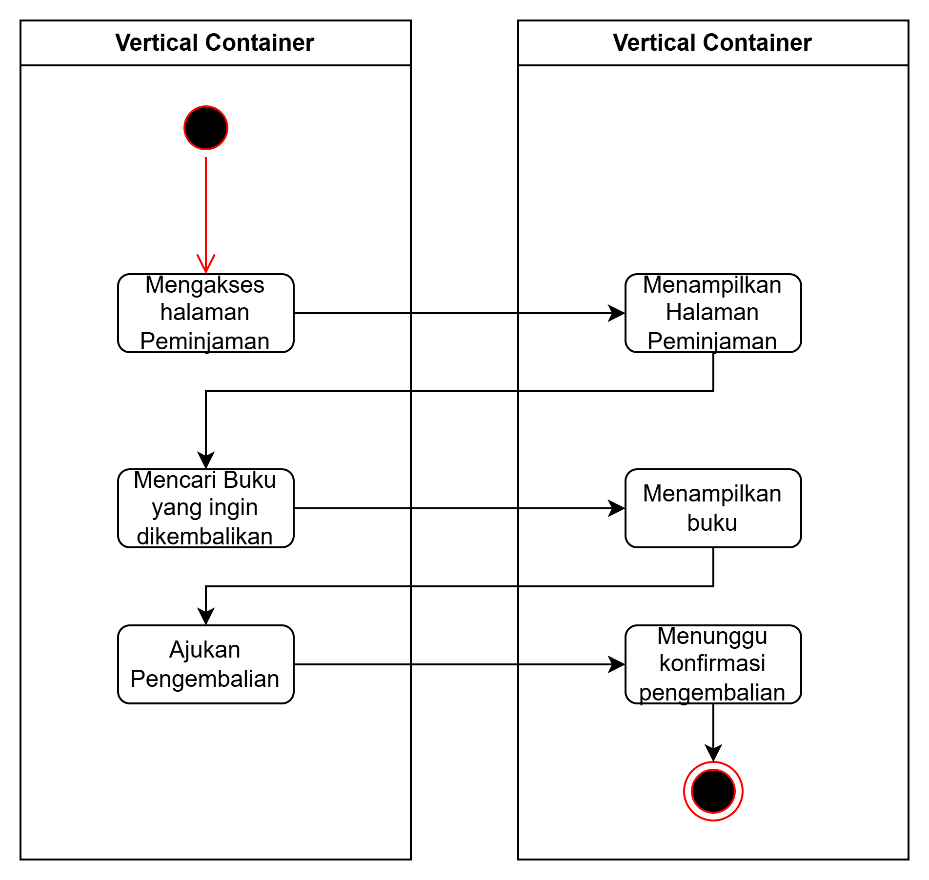
Gambar 10. Rancangan Activity Diagram Ketika user mencoba untuk login

Pada rancangan Activity Diagram tersebut, jika user memasukan email dan password, sistem akan mengecek apakah email dan password tersebut sesuai dengan yang ada pada database, dan khusus untuk password sudah melalui proses hashing terlebih dulu, sehingga sistem akan hashing password dan membandingkan hasil hashing yang diketikan user, dan kondisinya harus kedua-nya benar, dengan kata lain email dan password harus terisi kedua nya dan sesuai.



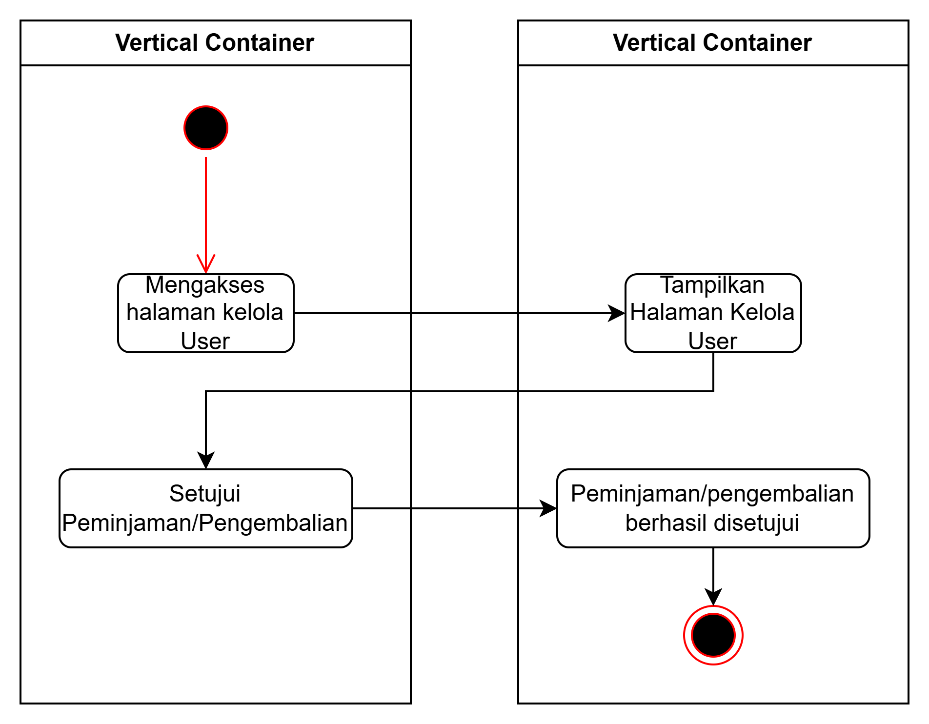
Gambar 11. Rancangan Activity Diagram Ketika user yang sudah ter-registrasi ingin mengajukan peminjaman

Pada rancangan Activity Diagram tersebut jika user sudah login, user dapat mencari buku dan mengajukan peminjaman dan menunggu konfirmasi dari admin yang mengelola.



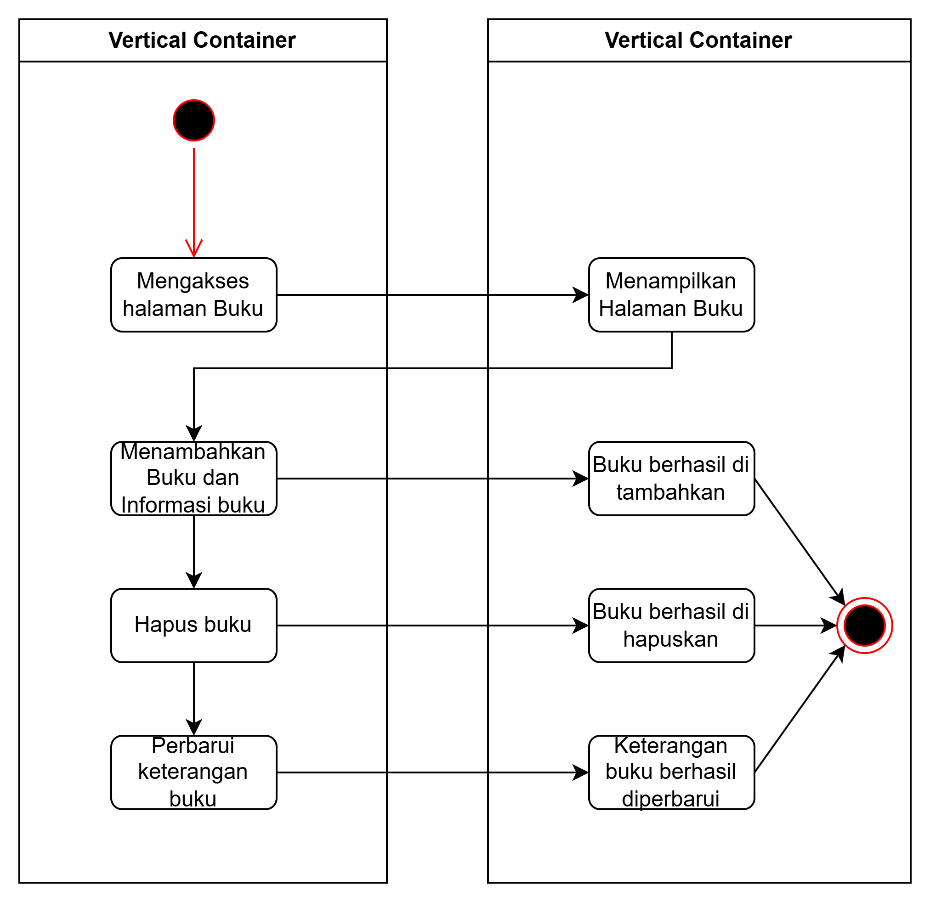
Gambar 12. Rancangan Activity Diagram pengembalian buku

Pada diagram tersebut, user mengakses halaman peminjaman, dan memilih buku yang ingin dikembalikan, dan klik tombol ajukan pengembalian, dan setelah dikonfirmasi oleh admin maka status peminjaman menjadi “selesai”.



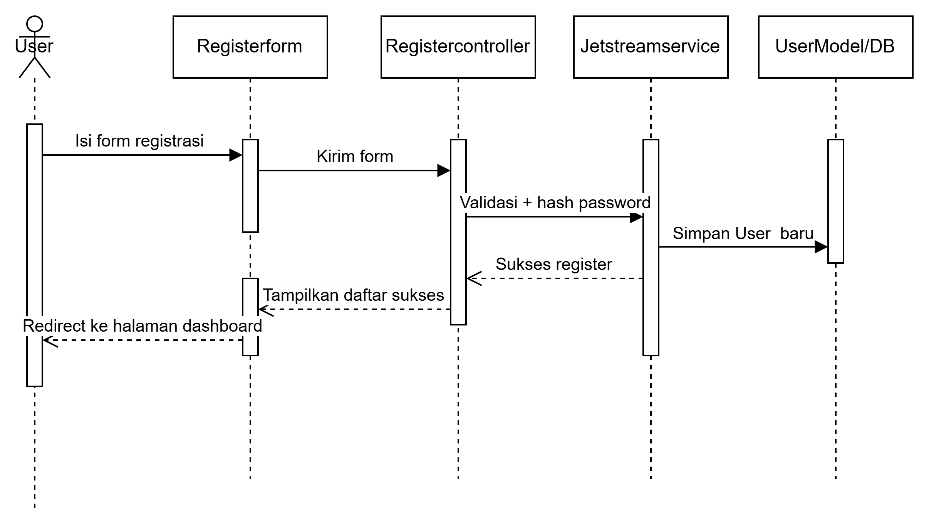
Gambar 13. Rancangan Activity diagram pada admin untuk Kelola peminjama user.

Pada rancangan Activity Diagram ini, kita beralih ke admin dalam mengelola peminjaman maupun pengembalian buku para user, admin dapat pergi ke halaman Kelola user, dan dapat melihat request dari semua user, dan dapat menyetujui peminjaman maupun mengkonfirmasi pengembalian buku.



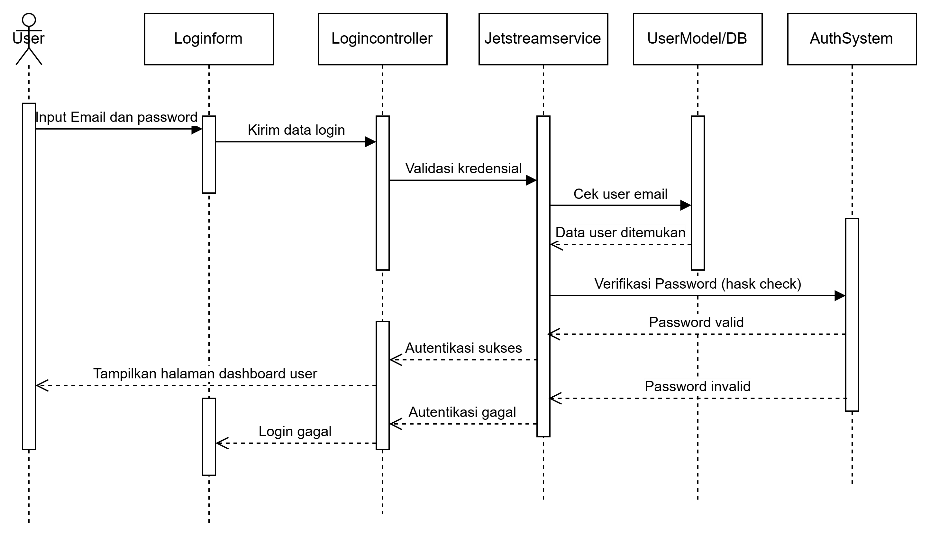
Gambar 14. Rancangan Activity Diagram untuk admin dalam pengelolaan buku

Admin dapat mengakses halaman buku, dan dapat menambahkan buku baru, melihat buku yang sudah ditambahkan, meng-edit informasi buku, maupun menghapus buku yang sudah terdaftar.



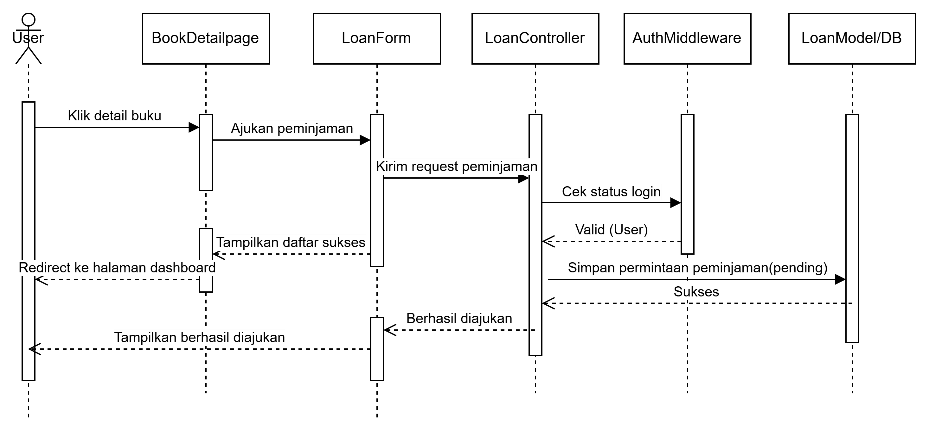
Gambar 15. Rancangan Sequence Diagram pada form registrasi akun

User mengisi form resgistrasi pada laman registrasi, dan mengirim data form pada Registercontroller, kemudian password di hash dan jika berhasil, sistem akan menyimpan data pada database



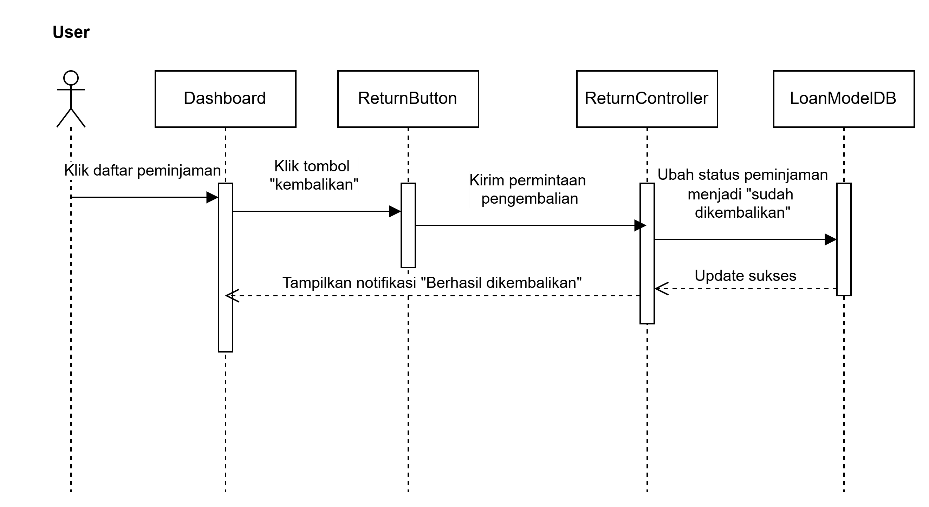
Gambar 16. Rancangan Sequence Diagram pada sistem autentika login

User input email dan password pada lama login, dan data dikirim, kemudian Jetstreamservice akan mem-validasi dengan mengecek pada database apakah email sudah terdaftar, jika data ditemukan maka selanjutnya sistem akan memverifikasi password yang dimasukan user(hash check) jika valid, maka user akan masuk ke halaman dashboard, dan jika tidak login gagal



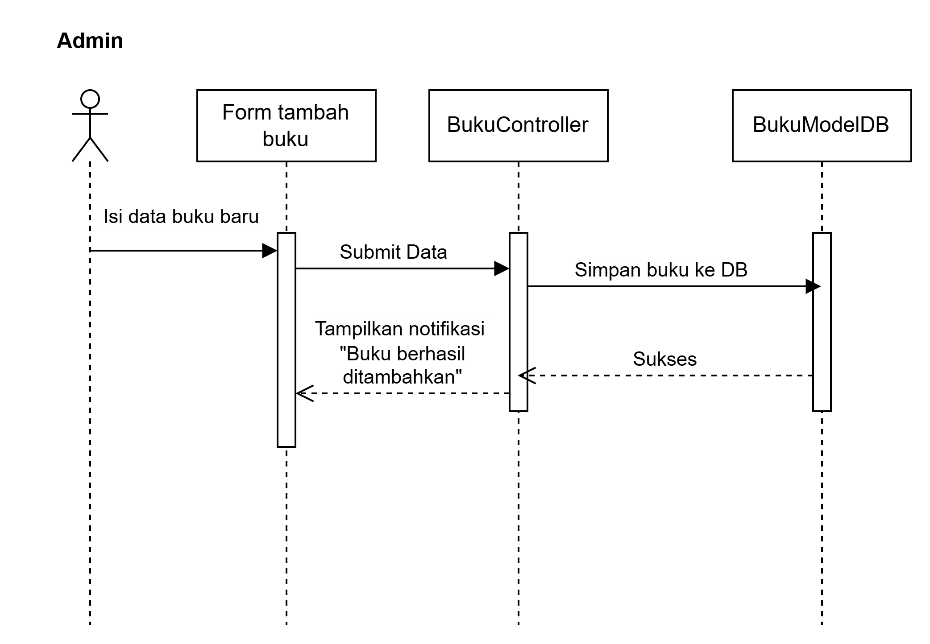
Gambar 17. Rancangan Sequence Diagram pada sistem peminjaman

User mengajukan peminjaman terlebih dahulu, kemudian sistem akan mengirim request peminjaman, dan sistem juga akan mengecek apakah user tersebut sudah login, jika ya, maka peminjaman akan tersimpan ke database dan peminjaman buku sudah berhasil diajukan.



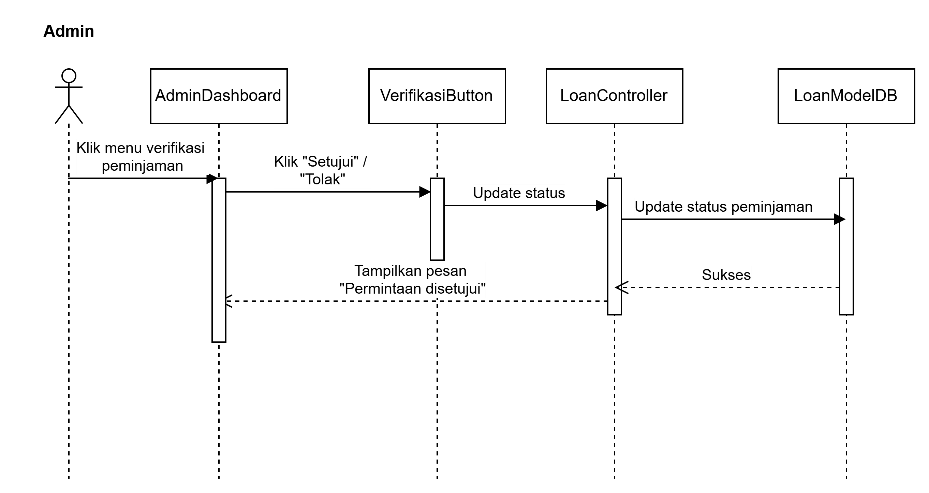
Gambar 18. Rancangan Sequence Diagram sistem pengembalian

Rancangan Sequence Diagram tersebut adalah sistem untuk pengembalian buku, user mengakses peminjaman buku terlebih dahulu, kemudian klik tombol “kembalikan”, sistem akan mengirimkan request dan mengubah status peminjaman menjadi "sudah dikembalikan"



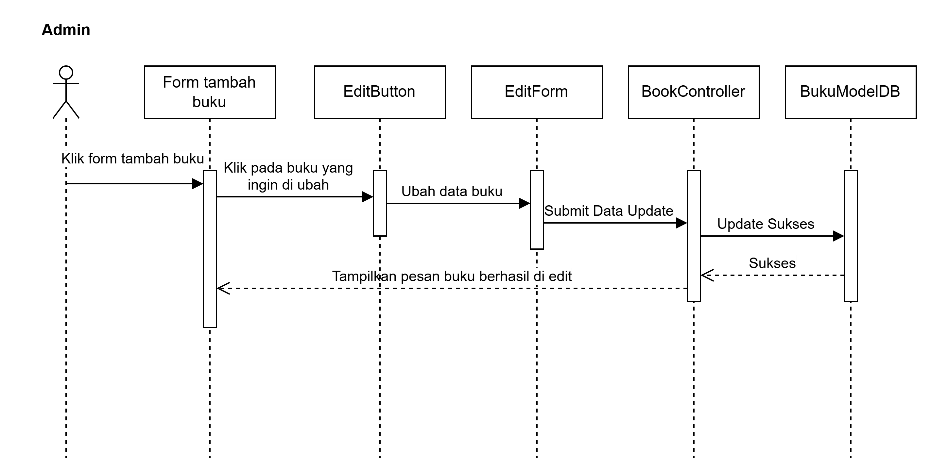
Gambar 18. Rancangan Sequence Diagram tambah buku

Admin dapat login terlebih dahulu, kemudian masuk ke halaman tambah buku, dan mengisi data buku baru, kemudian buku akan disimpan ke dalam Database, dan sistem akan menampilkan notifikasi “Buku berhasil ditambahkan”



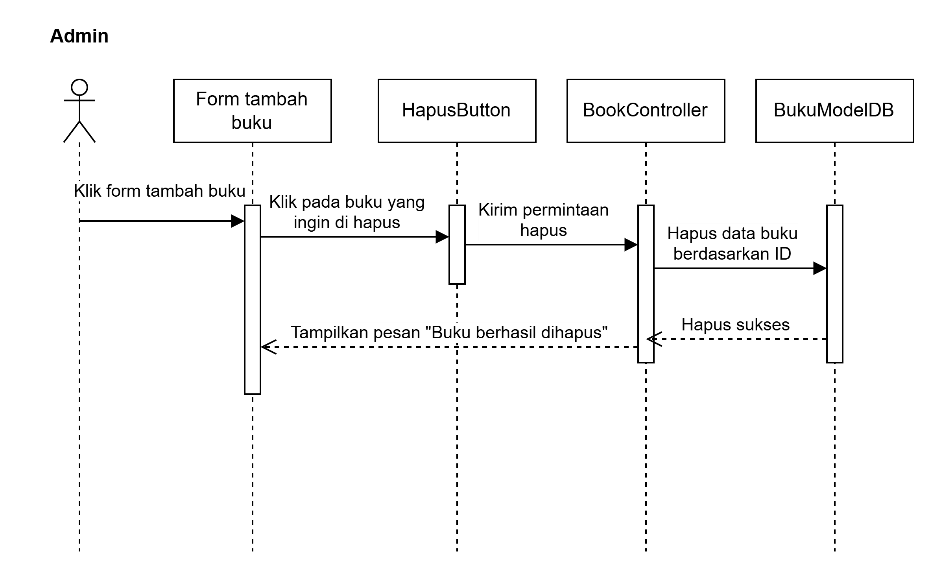
Gambar 19. Rancangan Seqeunce Diagram verifikasi peminjaman

Sequence Diagram diatas adalah untuk admin mengkonfirmasi peminjaman oleh user, admin dapat login terlebih dahulu dan akses halaman menu verifikasi peminjaman (kelola user) dan klik “setuju” atau “tolak”. Kemudian status peminjaman akan ter-update pada Database.



Gambar 20. Rancangan Sequence Diagram kelola buku(edit)

Rancangan Sequence Diagram diatas adalah untuk admin mengelola buku. Khususnya untuk meng-edit detail buku. Pertama admin mengakses halaman kelola buku, kemudian mengisi form tambah buku, klik kirim, dan data akan berubah pada database, dan sistem akan mengirimkan pesan buku berhasil di edit



Gambar 21. Rancangan Sequence Diagram kelola buku

Rancangan Sequence Diagram diatas adalah untuk admin mengelola buku. Khususnya untuk meng-hapus buku. Admin mengakses halaman kelola buku, kemudian pilih buku yang ingin dihapus dan klik tombol “hapus buku”, klik kirim, dan data akan dihapus pada database, dan sistem akan mengirimkan pesan buku berhasil di hapus.

**BAB 5**

**PERANCANGAN BASIS DATA**

1. **Tabel User**

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | users |
| Fungsi | : | Untuk menyimpan data pengguna dan admin |
| Media | : | Hardisk |
| Primary key | : | id |
| Foreign key | : | - |

**Tabel 1. users**

| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT | Primary Key |
| name | VARCHAR | Nama pengguna |
| email | VARCHAR (Unique) | Email pengguna (unik) |
| phone | VARCHAR (Nullable) | Nomor telepon |
| address | VARCHAR (Nullable) | Alamat (opsional) |
| usertype | VARCHAR | Role user (default: user) |
| password | VARCHAR | Password terenkripsi |
| created\_at | TIMESTAMP | Waktu dibuat |
| updated\_at | TIMESTAMP | Waktu diubah |

1. **Tabel kategori**

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | kategori |
| Fungsi | : | Menyimpan data kategori buku (Seperti contoh: Programming, Jaringan, Fiksi, dll) |
| Media | : | Hardisk |
| Primary key | : | id |
| Foreign key | : | - |

**Tabel 2. kategori**

| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT (Auto) | Primary Key |
| nama | VARCHAR | Nama kategori buku |
| created\_at | TIMESTAMP | Waktu dibuat |
| updated\_at | TIMESTAMP | Waktu diubah |
| deleted\_at | TIMESTAMP | jika dihapus |

1. **Tabel buku**

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | buku |
| Fungsi | : | Menyimpan data buku yang tersedia di perpustakaan |
| Media | : | Hardisk |
| Primary key | : | id |
| Foreign key | : | kategori\_id > kategori(id) |

**Tabel 3. buku**

| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT (Auto) | Primary Key |
| judul | VARCHAR (Nullable) | Judul buku |
| penulis | VARCHAR (Nullable) | Nama penulis |
| penerbit | VARCHAR (Nullable) | Nama penerbit |
| deskripsi | LONGTEXT (Nullable) | Deskripsi detail buku |
| isbn | VARCHAR (Nullable) | Nomor ISBN |
| tahun | VARCHAR (Nullable) | Tahun terbit |
| jumlah | VARCHAR (Nullable) | Jumlah stok buku |
| book\_img | VARCHAR (Nullable) | Path gambar buku |
| kategori\_id | BIGINT | Foreign Key ke tabel kategori.id, onDelete CASCADE |
| created\_at | TIMESTAMP | Waktu dibuat |
| updated\_at | TIMESTAMP | Waktu diubah |

1. **Tabel pinjams**

|  |  |  |
| --- | --- | --- |
| Nama Tabel | : | pinjams |
| Fungsi | : | Menyimpan data peminjaman buku oleh pengguna |
| Media | : | Menyimpan data peminjaman buku oleh pengguna |
| Primary key | : | id |
| Foreign key | : | buku\_id > buku(id), user\_id > users(id) |

**Tabel 4. pinjams**

| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| --- | --- | --- |
| id | BIGINT (Auto) | Primary Key |
| buku\_id | BIGINT | Foreign Key ke tabel buku.id, onDelete CASCADE |
| user\_id | BIGINT | Foreign Key ke tabel users.id, onDelete CASCADE |
| status | VARCHAR | Status peminjaman (default: Ditolak) |
| created\_at | TIMESTAMP | Waktu dibuat |
| updated\_at | TIMESTAMP | Waktu diubah |

**BAB 6**

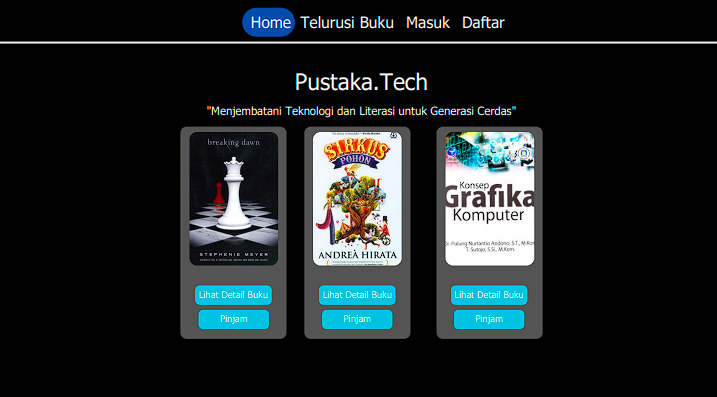
**PERANCANGAN ANTAR MUKA (USER INTERFACE)**

1. **Perancangan Halaman Dashboard Pengguna ketika belum memiliki akun atau belum masuk**

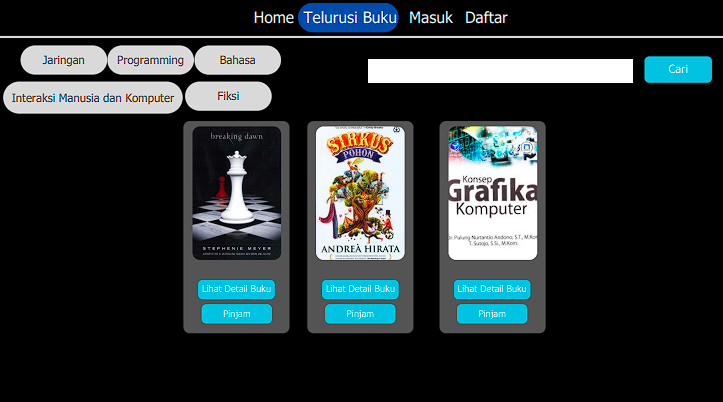
Pada saat pengguna mengunjungi halaman utama pada website, dibawah ini adalah rancangan tampilan utama ketika pengguna belum memiliki akun atau belum masuk dan mengakses website.

1. Perancangan halaman utama atau Beranda ketika pengguna belum memiliki akun.

Pada Rancangan dibawah ini, akan ada opsi untuk masuk (Jika sudah memiliki akun) atau Daftar (Jika belum memiliki akun).



2. Perancangan halaman telusuri buku



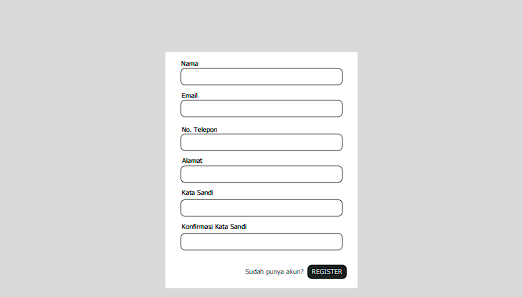
1. **Perancangan Halaman Login dan Register**

Pada saat pengguna belum memiliki akun, dan ingin membuatnya (agar dapat meminjam buku) atau jika pengguna sudah memiliki akun dan ingin melakukan login untuk masuk ke dalam akun (Sebagai user maupun admin), maka akan dihadapkan pada halaman login, maka pengguna diharuskan mendaftar terlebih dahulu jika belum memiliki akun.

1. Perancangan halaman login



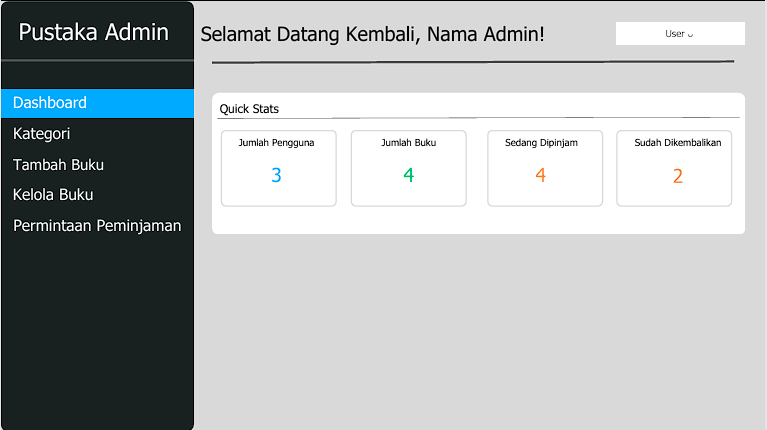
2. Perancangan halaman register



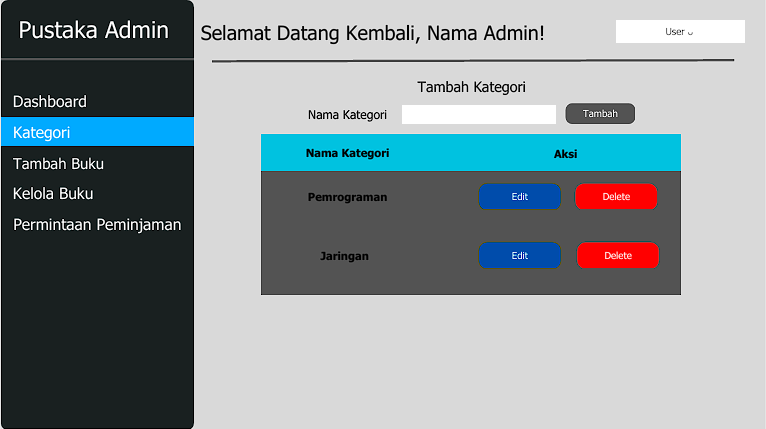
1. **Perancangan Halaman beranda Admin**

Saat pengguna dengan peran admin ingin masuk ke sistem, Pengguna harus memiliki akun admin yang valid untuk dapat mengakses halaman dashboard admin. Setelah berhasil login, pengguna akan dibawa ke tampilan dashboard admin yang berfungsi sebagai pusat kontrol untuk mengelola sistem. Berikut adalah rancangan dari tampilan dashboard admin.

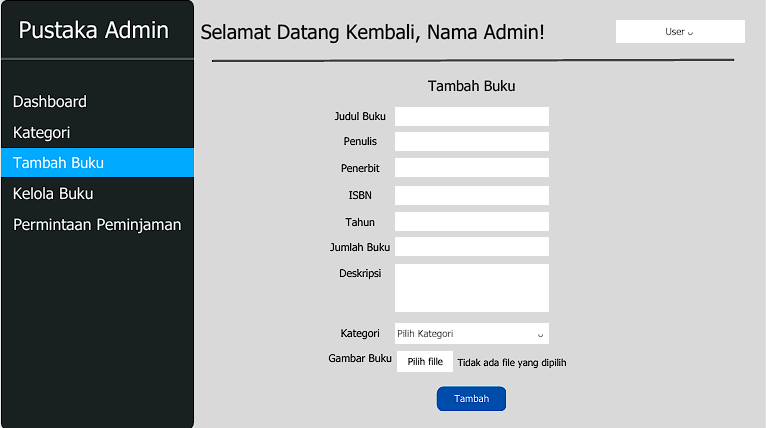
1. Perancangan halaman dashboard admin.



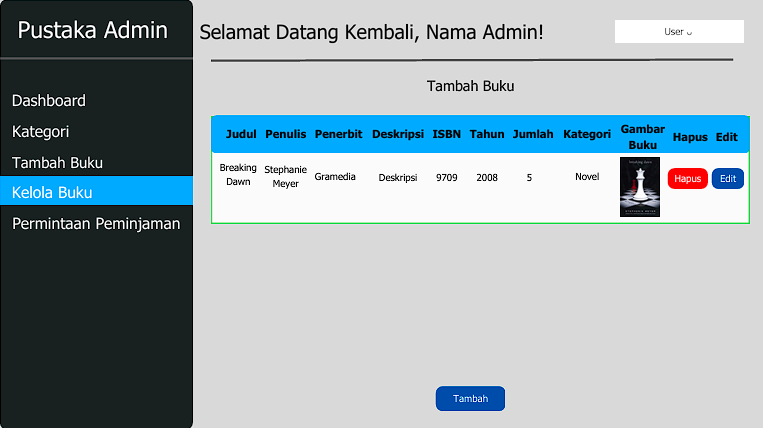
2. Perancangan halaman Kategori pada Admin



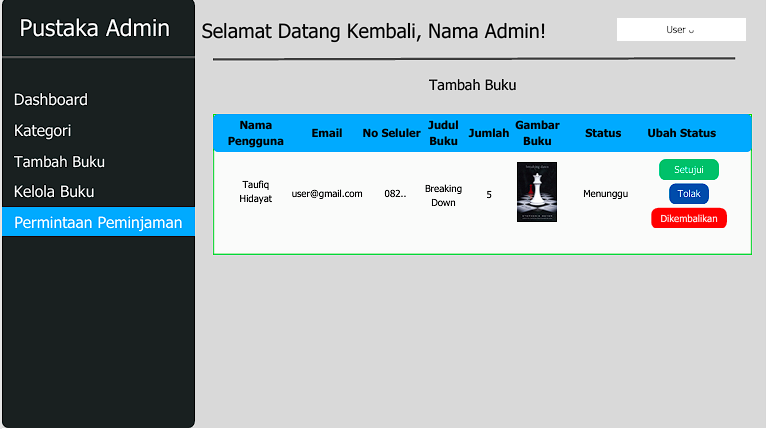
3. Perancangan halaman Tambah Buku admin.



4. Perancangan halaman Kelola Buku Admin.



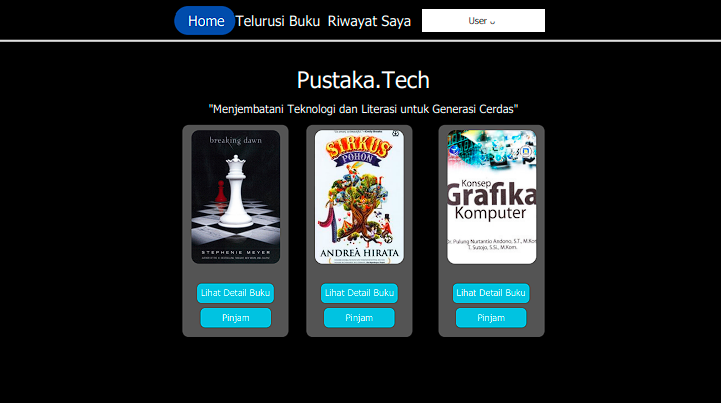
5. Perancangan halaman Permintaan Peminjaman Admin.



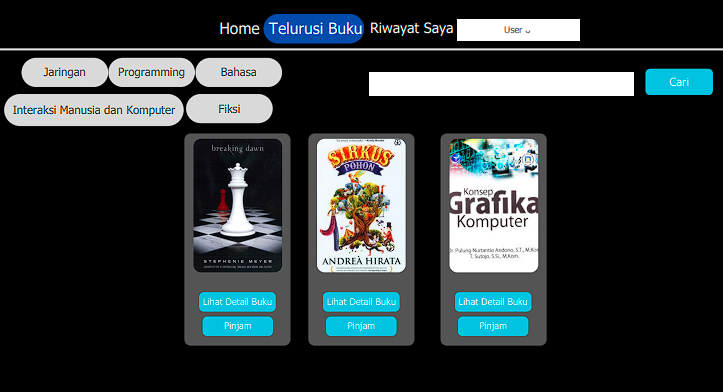
1. **Perancangan Halaman Pengguna login**

Berikut ini adalah rancangan Ketika pengguna sudah memiliki akun dan sudah berhasil login.

1. Perancangan halaman utama pengguna ketika berhasil masuk

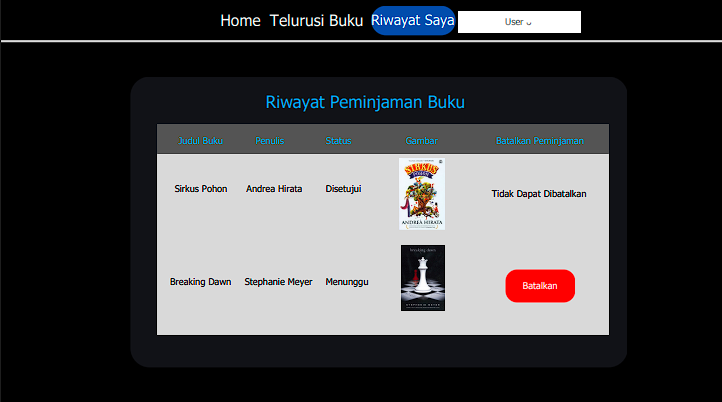


2. Perancangan halaman telusuri buku.



3. Perancangan halaman Riwayat Peminjaman Buku.

Halaman ini hanya dapat diakses oleh pengguna yang sudah memiliki akun, dan pada halaman ini pengguna dapat melihat buku yang sedang menunggu konfirmasi, sedang dipinjam, maupun buku yang sudah pernah dipinjam, dan pengguna juga dapat membatalkan peminjaman apabila status peminjaman belum disetujui oleh admin



**BAB 7**

**IMPLEMENTASI SISTEM**

**BAB 8**

**PENUTUP**

# DAFTAR PUSTAKA

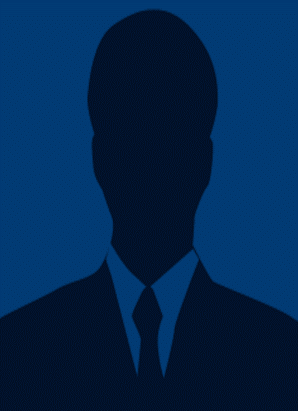
Abdul Kadir. (2017). Pemogramanan Arduino dan Procesing. Jakarta: Penerbit kompas gramedia.

Mr. Dic'sr. (2017). Unified Modelling Language (UML). *Available* *at:* http://www.dicsr-qnt.com/2017/09/uml-unified-modelling-language.html. Diakses tanggal 3 Desember 2022.

# TENTANG PENULIS

**PENULIS 1**

**(nama penulis)**

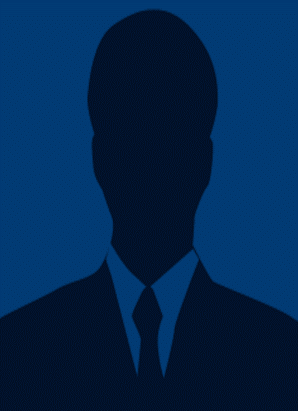
(nama ….) adalah laki laki / perempuan kelahiran …… pada tanggal ….. Merupakan anak pertama/kedua/ketiga dari …..bersaudara. Menghabiskan masa kecil di kota ……, tinggal bersama ……hingga lulus Sekolah Menengah Atas/Kejuruan.

Riwayat pendidikan dimulai dari TK ……(tahun …), SDN ….Pagi (tahun….), SMP (….), SMA/SMKK (tahun….), dan Sekolah Tinggi Teknologi Informasi NIIT Jakarta (tahun ….).

Saat ini aktif dalam kegiatan Organisasi ……, membuat saya semakin cinta akan hidup sosial, membuat saya semakin yakin kalau hidup harus berguna untuk orang lain. Setelah menyelesaikan pendidikan wajib saya melanjutkan kuliah di Sekolah Tinggi Teknologi Informasi NIIT pada Program Studi Teknik Informatika pada tahun …... Selama berkuliah saya aktif mengikuti berbagai organisasi kemahasiswaan, salah satu nya Komunitas programming web dan Android. Dalam mengisi waktu luang selama perkuliahan, saya sering mencari kesibukan untuk menambah ilmu dan pengalaman saya. Sesekali juga saya dipercaya untuk mengerjakan suatu proyek magang (PKL) untuk membuat Aplikasi pada suatu perusahaan. **Silahkan ditambahkan sesuai dengan keperluan.**

**PENULIS 2**

**(nama penulis)**

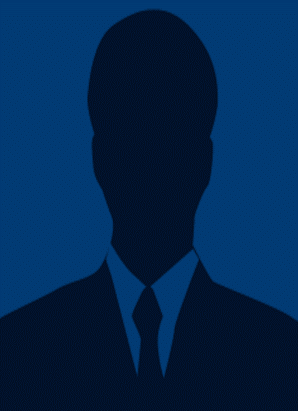
(nama ….) adalah laki laki / perempuan kelahiran …… pada tanggal ….. Merupakan anak pertama/kedua/ketiga dari …..bersaudara. Menghabiskan masa kecil di kota ……, tinggal bersama ……hingga lulus Sekolah Menengah Atas/Kejuruan.

Riwayat pendidikan dimulai dari TK ……(tahun …), SDN ….Pagi (tahun….), SMP (….), SMA/SMKK (tahun….), dan Sekolah Tinggi Teknologi Informasi NIIT Jakarta (tahun ….).

Saat ini aktif dalam kegiatan Organisasi ……, membuat saya semakin cinta akan hidup sosial, membuat saya semakin yakin kalau hidup harus berguna untuk orang lain. Setelah menyelesaikan pendidikan wajib saya melanjutkan kuliah di Sekolah Tinggi Teknologi Informasi NIIT pada Program Studi Teknik Informatika pada tahun …... Selama berkuliah saya aktif mengikuti berbagai organisasi kemahasiswaan, salah satu nya Komunitas programming web dan Android. Dalam mengisi waktu luang selama perkuliahan, saya sering mencari kesibukan untuk menambah ilmu dan pengalaman saya. Sesekali juga saya dipercaya untuk mengerjakan suatu proyek magang (PKL) untuk membuat Aplikasi pada suatu perusahaan. **Silahkan ditambahkan sesuai dengan keperluan.**

**PENULIS 3**

**(nama penulis)**

(nama ….) adalah laki laki / perempuan kelahiran …… pada tanggal ….. Merupakan anak pertama/kedua/ketiga dari …..bersaudara. Menghabiskan masa kecil di kota ……, tinggal bersama ……hingga lulus Sekolah Menengah Atas/Kejuruan.

Riwayat pendidikan dimulai dari TK ……(tahun …), SDN ….Pagi (tahun….), SMP (….), SMA/SMKK (tahun….), dan Sekolah Tinggi Teknologi Informasi NIIT Jakarta (tahun ….).

Saat ini aktif dalam kegiatan Organisasi ……, membuat saya semakin cinta akan hidup sosial, membuat saya semakin yakin kalau hidup harus berguna untuk orang lain. Setelah menyelesaikan pendidikan wajib saya melanjutkan kuliah di Sekolah Tinggi Teknologi Informasi NIIT pada Program Studi Teknik Informatika pada tahun …... Selama berkuliah saya aktif mengikuti berbagai organisasi kemahasiswaan, salah satu nya Komunitas programming web dan Android. Dalam mengisi waktu luang selama perkuliahan, saya sering mencari kesibukan untuk menambah ilmu dan pengalaman saya. Sesekali juga saya dipercaya untuk mengerjakan suatu proyek magang (PKL) untuk membuat Aplikasi pada suatu perusahaan. **Silahkan ditambahkan sesuai dengan keperluan.**