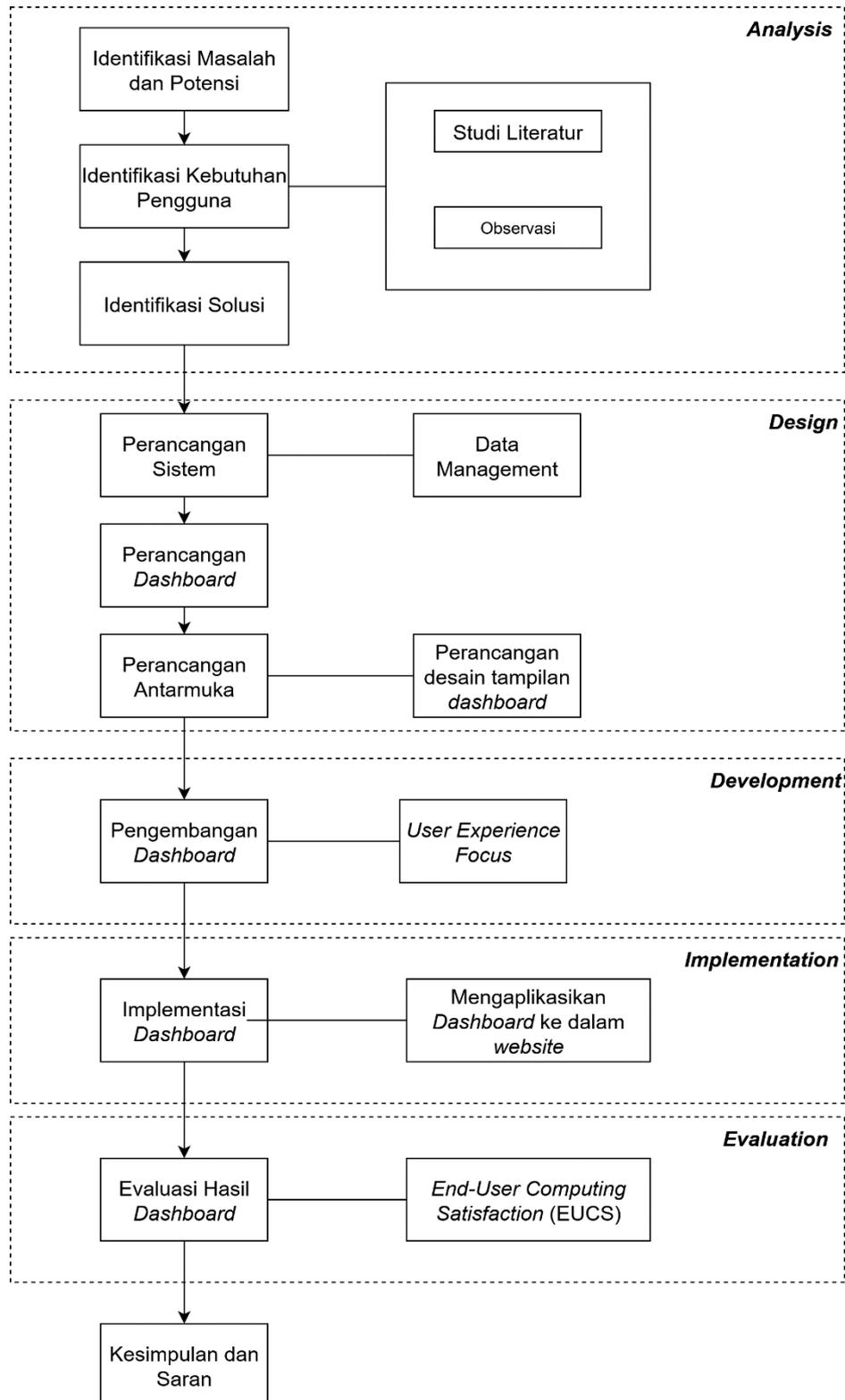


## **BAB III**

### **METODOLOGI PENELITIAN**

Pada bab ini, akan dijelaskan secara komprehensif mengenai metodologi penelitian yang telah dipilih untuk memandu langkah-langkah penelitian. Analisis dalam tahap ini mencakup pemilihan metode penelitian yang tepat, teknik yang sesuai untuk mengumpulkan dan menganalisis data, serta alat-alat yang digunakan dalam proses perancangan dan implementasi *dashboard* operasional untuk perolehan donasi beasiswa di tingkat perguruan tinggi yang menjunjung tinggi prinsip akuntabilitas. Tahapan analisis ini sangat penting untuk memastikan bahwa penelitian ini dilaksanakan dengan tepat dan menghasilkan solusi yang relevan serta dapat diimplementasikan secara efektif dalam konteks yang bersangkutan. Dengan demikian, metodologi penelitian yang dijelaskan akan memberikan landasan yang kokoh untuk merancang *dashboard* operasional yang tidak hanya memenuhi kebutuhan pengguna, tetapi juga memastikan perolehan donasi beasiswa berjalan secara transparan dan akuntabel.

### 3.1 Desain Penelitian



Gambar 3.1 Desain Penelitian

Rasyid Andriansyah, 2025

RANCANG BANGUN DASHBOARD PEROLEHAN DONASI BEASISWA MENGGUNAKAN PENDEKATAN COGNITIVE LOAD THEORY UNTUK MEMBANTU AKUNTABILITAS INFORMASI

Universitas Pendidikan Indonesia | repository.upi.edu | perpustakaan.upi.edu

Desain penelitian adalah gambaran inti dari penelitian yang akan dilakukan dan tujuan desain penelitian adalah untuk mendapatkan gambaran luas dari penelitian yang akan diteliti. Pada Gambar 3.1 di atas dijelaskan desain penelitian yang akan dilakukan oleh penulis.

### 3.1.1 *Analysis*

Tujuan dari tahap *analysis* adalah untuk mengidentifikasi dan mengevaluasi permasalahan yang timbul dalam sistem perolehan donasi beasiswa yang sedang berjalan guna pengembangan *dashboard* yang baik. Langkah pertama dalam analisis adalah mengidentifikasi masalah yang dihadapi dalam perolehan donasi beasiswa di tingkat perguruan tinggi. Ini termasuk masalah-masalah seperti kurangnya transparansi dalam penggunaan dana donasi, kesulitan dalam pelacakan dan pelaporan pengeluaran, serta tantangan dalam memastikan akuntabilitas dalam pengelolaan dana. Di sisi lain, potensi solusi juga perlu dipertimbangkan, seperti penggunaan teknologi *dashboard* operasional untuk meningkatkan transparansi, efisiensi, dan akuntabilitas dalam pengelolaan donasi. Penelitian ini menggunakan teknik pengumpulan data dengan studi literatur, observasi.

#### a. Studi literatur

Studi literatur dilakukan untuk memahami secara mendalam tentang praktik-praktik terbaik dalam perolehan donasi beasiswa serta konsep-konsep yang relevan seperti transparansi dan akuntabilitas. Dari studi literatur ini, dapat ditemukan wawasan yang berharga yang akan membantu dalam mengidentifikasi kebutuhan pengguna.

#### b. Observasi

Melalui observasi langsung terhadap proses perolehan donasi beasiswa di beberapa perguruan tinggi, kami dapat mengidentifikasi secara langsung tantangan-tantangan yang dihadapi oleh para pengelola dan bagaimana proses tersebut berlangsung secara nyata. Observasi ini memberikan wawasan yang berharga untuk memahami kebutuhan pengguna dengan lebih baik.

### c. Identifikasi Solusi

Setelah masalah dan kebutuhan pengguna teridentifikasi dengan baik, langkah selanjutnya adalah mengidentifikasi solusi yang tepat. Ini melibatkan penggunaan teknologi *dashboard* operasional untuk memenuhi kebutuhan dan mengatasi masalah yang ada. Dengan memperhatikan input dari studi literatur dan observasi, solusi yang diusulkan akan dirancang sedemikian rupa sehingga dapat memberikan manfaat maksimal bagi para pengguna dan membantu akuntabilitas dalam perolehan donasi beasiswa di tingkat perguruan tinggi.

### 3.1.2 Design

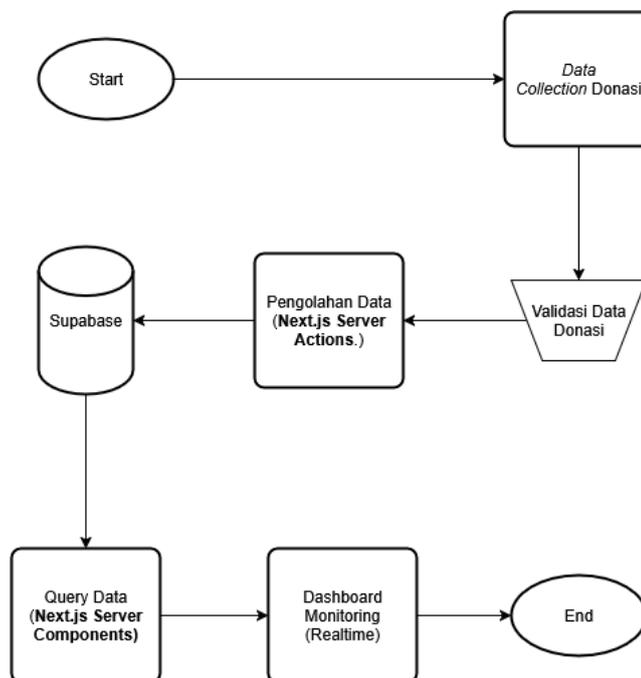
Setelah melalui fase analisis untuk memahami masalah dan kebutuhan pengguna, penelitian beralih ke tahap desain. Pada fase ini, fokus utama adalah menerjemahkan temuan-temuan dari tahap analisis menjadi sebuah rancangan sistem dan antarmuka yang konkret dan fungsional. Proses perancangan ini didasarkan pada riset user experience (UX) yang mendalam, yang telah mengidentifikasi kebutuhan serta preferensi esensial pengguna melalui observasi.

Pembuatan persona pengguna membantu dalam memvisualisasikan profil pengguna utama, memastikan bahwa setiap keputusan desain berpusat pada pengguna (user-centered). Dengan menggunakan landasan ini, perancangan dashboard diarahkan untuk menciptakan alur interaksi yang logis dan penyajian informasi yang jelas. Tujuannya adalah untuk menjawab kebutuhan fungsional sekaligus memastikan bahwa sistem yang dihasilkan mudah digunakan, responsif, dan pada akhirnya mampu meningkatkan adopsi serta kepuasan pengguna secara keseluruhan.

#### 3.1.2.1 Perancangan Sistem

Perancangan sistem adalah tahapan yang merinci secara mendalam tentang bagaimana sistem akan bekerja dan beroperasi (Nopriandi, 2018). Perancangan ini

akan membentuk kerangka kerja bagi setiap aspek yang diimplementasikan dalam sistem.



Gambar 3.2 Desain Sistem

Untuk memberikan gambaran umum yang terstruktur mengenai arsitektur ini, tabel berikut memetakan setiap komponen dalam Gambar 3.2 ke teknologi spesifik yang digunakan, beserta fungsi dan justifikasi teknisnya.

Tabel 3.1 Gambaran umum desain sistem

Komponen (Diagram)	Teknologi yang Digunakan	Fungsi dan Justifikasi Teknis
Data Collection Donasi	Next.js Client Components, HTML <form>	Menyediakan antarmuka pengguna untuk input data donasi. Penggunaan elemen <form> standar yang terikat pada Server Action memastikan progressive enhancement, sehingga fungsionalitas tetap terjaga meskipun JavaScript gagal dimuat.
Pengolahan & Validasi Data	Next.js Server Actions, Zod	Menangani mutasi data (pembuatan donasi baru) di sisi server secara aman dan efisien, mengadopsi paradigma Remote Procedure Call (RPC) yang menyederhanakan

Komponen (Diagram)	Teknologi yang Digunakan	Fungsi dan Justifikasi Teknis
		arsitektur dengan menghilangkan kebutuhan endpoint API manual. Validasi skema dengan Zod menjamin integritas dan keamanan data sebelum interaksi dengan basis data.
Supabase (Database)	Supabase (PostgreSQL), Supabase Auth, Row-Level Security (RLS)	Berfungsi sebagai Backend-as-a-Service (BaaS) yang menyediakan basis data PostgreSQL, manajemen otentikasi, dan fitur keamanan krusial Row-Level Security (RLS) untuk menegakkan kebijakan akses data yang ketat, yang merupakan fondasi dari sistem yang akuntabel.
Query Data	Next.js Server Components (RSC)	Melakukan akuisisi data awal untuk render pertama dashboard. Proses ini terjadi sepenuhnya di server, meningkatkan performa dan keamanan dengan menghilangkan lapisan API untuk operasi baca (read operations) dan menjaga kredensial basis data agar tidak terekspos ke klien.
Dashboard Monitoring (Realtime)	Next.js Client Components, Supabase Realtime Subscriptions	Menampilkan data awal yang diterima dari Server Components, kemudian berlangganan pembaruan data secara real-time melalui koneksi WebSocket. Ini memungkinkan pemantauan berkelanjutan dan instan terhadap aktivitas donasi, memenuhi kebutuhan "dashboard operasional".

Tahapan perancangan sistem ini meliputi langkah-langkah awal hingga pengujian validasi untuk memastikan keakuratan dan kualitas data yang digunakan.

#### 1. Pengumpulan Data Donasi (Client-Side)

Proses pengelolaan donasi dimulai dari interaksi pengguna dengan antarmuka aplikasi yang dibangun menggunakan Next.js. Komponen yang

bertanggung jawab untuk pengumpulan data, seperti formulir donasi, kemungkinan besar akan diimplementasikan sebagai Client Component untuk memungkinkan interaktivitas yang kaya, seperti validasi input di sisi klien secara langsung.

Komponen ini akan merender sebuah elemen `<form>` HTML standar. Salah satu keunggulan utama dari arsitektur Next.js modern adalah kemampuan untuk mengikat formulir ini secara langsung ke fungsi yang dieksekusi di server, yang dikenal sebagai Server Action, melalui atribut `action`. Pendekatan ini merupakan implementasi dari prinsip progressive enhancement. Artinya, formulir dapat berfungsi penuh dan data dapat dikirimkan ke server bahkan jika JavaScript di sisi klien gagal dimuat atau dinonaktifkan oleh pengguna. Hal ini memberikan tingkat keandalan yang lebih tinggi dibandingkan dengan pendekatan *Single Page Application* (SPA) tradisional yang sepenuhnya bergantung pada eksekusi JavaScript di klien untuk menangani pengiriman formulir.

## 2. Pengolahan dan Validasi Data di Sisi Server (Next.js Server Actions)

Setelah pengguna mengirimkan formulir, data tersebut tidak lagi ditangani oleh skrip di sisi klien, melainkan langsung diproses oleh sebuah Next.js Server Action. Dalam diagram alur, tahapan "Pengolahan Data" dan "Validasi Data Donasi" secara konseptual digabungkan ke dalam satu fungsi asinkron tunggal yang aman dan dieksekusi sepenuhnya di lingkungan server. Fungsi ini diidentifikasi dengan direktif "use server" di bagian atasnya, yang menginstruksikan Next.js untuk memperlakukannya sebagai endpoint server.

Penggunaan Server Actions mengubah paradigma pengembangan dari yang sebelumnya mengharuskan pembuatan dan pengelolaan endpoint API (misalnya, REST atau GraphQL) secara manual, menjadi model Remote Procedure Call (RPC) yang lebih terintegrasi. Hal ini menyederhanakan arsitektur dan meningkatkan pengalaman pengembang dengan memungkinkan logika mutasi data ditempatkan berdekatan dengan komponen antarmuka yang memicunya. Namun, abstraksi yang kuat ini

juga membawa tanggung jawab keamanan yang besar. Pengembang harus menyadari bahwa setiap Server Action pada dasarnya adalah sebuah endpoint API publik yang dibuat secara otomatis oleh Next.js, meskipun tidak terlihat secara eksplisit. Oleh karena itu, memperlakukannya sebagai fungsi biasa tanpa pengamanan yang memadai dapat menciptakan "rasa aman yang palsu" dan membuka celah keamanan.

Untuk memitigasi risiko ini, setiap Server Action dalam sistem ini harus menerapkan dua lapisan keamanan fundamental:

- a. Validasi Input Wajib: Sebelum data diproses lebih lanjut, formData yang diterima oleh Server Action harus divalidasi secara ketat terhadap skema yang telah ditentukan. Pustaka seperti Zod sangat direkomendasikan untuk tujuan ini. Validasi skema memastikan bahwa data yang masuk memiliki format yang benar (misalnya, format email yang valid), memenuhi aturan bisnis (misalnya, jumlah donasi tidak boleh negatif), dan bersih dari potensi input berbahaya. Prinsip "jangan pernah mempercayai input dari klien" (*never trust client input*) harus ditegakkan secara mutlak pada lapisan ini.
- b. Otorisasi Pengguna: Server Action harus memverifikasi identitas dan hak akses pengguna yang memanggilmnya. Ini dilakukan dengan memeriksa sesi otentikasi pengguna yang aktif, yang dapat dikelola secara mulus melalui integrasi dengan Supabase Auth. Hanya pengguna yang terotentikasi dan memiliki izin yang sesuai yang boleh melanjutkan proses penyimpanan donasi.

Setelah proses validasi dan otorisasi berhasil dilewati, Server Action akan melanjutkan ke tahap interaksi dengan basis data.

### 3. Interaksi dengan Basis Data (Integrasi Supabase)

Interaksi antara logika server Next.js dan basis data Supabase difasilitasi oleh pustaka klien yang dirancang khusus untuk lingkungan server. Di dalam Server Action, pustaka "@supabase/ssr" akan digunakan untuk membuat klien Supabase yang aman untuk operasi di sisi server.

Pada tahap ini, salah satu fitur paling kuat dari Supabase yang menjadi fondasi akuntabilitas sistem mulai berperan adalah **Row-Level Security (RLS)**. RLS adalah fitur asli dari PostgreSQL yang memungkinkan administrator basis data untuk mendefinisikan kebijakan keamanan yang sangat granular langsung pada baris-baris data di dalam tabel.

Dengan menerapkan RLS, kontrol akses tidak lagi hanya bergantung pada logika di dalam kode aplikasi, tetapi ditegakkan secara fundamental di tingkat basis data dan ini merupakan implementasi teknis yang solid dari prinsip "privasi" dan "akuntabilitas" yang menjadi inti dari penelitian ini.

#### 4. *Query Data*

Next.js Server Components (RSC) melakukan akuisisi data awal untuk render pertama dashboard. Proses ini terjadi sepenuhnya di server, meningkatkan performa dan keamanan dengan menghilangkan lapisan API untuk operasi baca (read operations) dan menjaga kredensial basis data agar tidak terekspos ke klien.

#### 5. Visualisasi dan Pemantauan Real-time (*Dashboard Monitoring*)

*Dashboard monitoring* akan menjadi antarmuka yang digunakan untuk memvisualisasikan data donasi beasiswa yang telah terkumpul dan divalidasi. *Dashboard* ini akan menampilkan berbagai informasi tentang donasi yang telah diterima, penggunaan dana, dan dampak yang telah dicapai. Visualisasi data yang disajikan dalam *dashboard* akan membantu pemangku kepentingan untuk memahami informasi dengan lebih baik dan mengambil keputusan yang tepat terkait perolehan donasi beasiswa.

Dengan demikian, tahapan perancangan sistem ini menjadi landasan yang kokoh dalam pengembangan *dashboard* operasional untuk perolehan donasi beasiswa yang akuntabel di tingkat perguruan tinggi.

### 3.1.2.2 Perancangan *Dashboard*

Perancangan *dashboard* ini adalah merinci desain konsep dari *dashboard* yang akan dibangun. Dalam tahap ini, akan disusun tata letak yang optimal, komponen-komponen utama yang akan disertakan, serta fitur-fitur tambahan yang

mendukung fungsionalitas dan keterjangkauan informasi yang dibutuhkan untuk memastikan akuntabilitas dalam perolehan donasi beasiswa. Tata letak yang dipilih akan mempertimbangkan kebutuhan pengguna, dengan fokus pada penyajian informasi yang jelas dan mudah dimengerti. Komponen utama, seperti grafik dan visualisasi data lainnya, akan dipilih dengan hati-hati untuk memastikan bahwa mereka efektif dalam menyampaikan informasi yang relevan tentang pengelolaan dana donasi beasiswa dan dampaknya. Selain itu, fitur-fitur tambahan seperti kemampuan filter dan pencarian data akan disertakan untuk meningkatkan utilitas *dashboard* dalam memonitor dan mengevaluasi kinerja perolehan donasi beasiswa secara lebih detail, serta memastikan transparansi dalam pengelolaan dana. Dengan memperhatikan semua aspek ini, konsep dasar dari *dashboard* yang akan dibangun akan dirancang untuk memenuhi kebutuhan pengguna dan mendukung prinsip-prinsip akuntabilitas dalam perolehan donasi beasiswa di tingkat perguruan tinggi.

### 3.1.2.3 Perancangan Antarmuka

Perancangan antarmuka merupakan langkah penting dalam merancang sistem *dashboard* operasional untuk perolehan donasi beasiswa yang akuntabel di tingkat perguruan tinggi. Pada tahap ini, akan dirancang bagaimana tampilan *dashboard* akan disajikan kepada pengguna. Peneliti akan mengembangkan desain antarmuka *dashboard* dan sistem berdasarkan hasil perancangan sistem yang telah dipersiapkan sebelumnya. Melalui desain antarmuka ini, pengguna akan mendapatkan gambaran tentang cara menggunakan *dashboard* untuk melihat dan mengelola informasi terkait dengan donasi beasiswa. Selain itu, desain antarmuka juga akan menjelaskan bagaimana setiap komponen yang dibentuk akan mendukung prinsip-prinsip manajemen data yang telah diatur dalam perancangan sistem sebelumnya. Dengan demikian, perancangan antarmuka ini akan memastikan bahwa pengguna dapat dengan mudah dan efektif berinteraksi dengan *dashboard* operasional untuk perolehan donasi beasiswa yang akuntabel di tingkat perguruan tinggi.

### 3.1.2.4 Integrasi *Cognitive Load Theory* dan *Information Dashboard Design*

Pada tahap perancangan antarmuka, prinsip *Cognitive Load Theory* (CLT) dan *Information Dashboard Design* (berdasarkan prinsip Gestalt) diintegrasikan secara sistematis untuk memastikan dashboard tidak hanya estetik, tetapi juga efektif secara kognitif. Tujuannya adalah untuk meminimalkan beban kognitif ekstrinsik dan memaksimalkan beban kognitif germane. Langkah-langkah operasionalnya meliputi:

1. Simplifikasi Visual untuk Mengurangi Beban Ekstrinsik:
  - a. Prinsip *Signaling*: Menggunakan isyarat visual (seperti warna, tebal huruf, atau ikon) untuk menyoroti informasi paling kritis. Contohnya, angka KPI (total donasi) akan ditampilkan dengan ukuran font terbesar. Ini mengarahkan perhatian pengguna dan mengurangi waktu pencarian informasi.
  - b. Menghindari Efek *Split-Attention*: Informasi yang saling menjelaskan akan ditempatkan berdekatan secara fisik. Misalnya, legenda grafik tidak akan ditempatkan jauh dari grafik itu sendiri. Hal ini mencegah pengguna membagi perhatian mereka antara dua lokasi visual yang berbeda.
  - c. Prinsip Redundansi: Menghindari penyajian informasi yang sama dalam format yang berbeda secara bersamaan (misalnya, menampilkan tabel data mentah tepat di samping grafik yang sudah mewakili data tersebut), kecuali jika diperlukan untuk tujuan perbandingan spesifik.
2. Strukturisasi Informasi untuk Mengelola Beban Intrinsik:
  - a. Prinsip *Chunking*: Informasi akan dikelompokkan ke dalam "potongan" atau modul yang logis (misalnya, modul Pemasukan, modul Pengeluaran, modul Demografi Donatur). Setiap modul menyajikan 3-5 metrik kunci, sesuai dengan kapasitas memori kerja.

- b. Hierarki Informasi yang Jelas: Menempatkan informasi paling penting dan ringkas (KPI utama) di bagian atas dashboard (sesuai *F-pattern reading*). Informasi yang lebih detail atau analitis ditempatkan di bagian bawah, memungkinkan pengguna melakukan *drill-down* sesuai kebutuhan.
3. Mendorong Beban Germane melalui Interaktivitas:
    - a. Interaktivitas Terpandu: Menyediakan fitur interaktif yang bermakna, seperti filter berdasarkan periode waktu atau kategori donasi. Interaksi ini dirancang untuk tidak bersifat bebas (yang bisa membingungkan), tetapi untuk menjawab pertanyaan spesifik yang mungkin dimiliki pengguna (misalnya, "Bagaimana tren donasi pada kuartal lalu?").
    - b. Prinsip *Worked Example*: Beberapa visualisasi data akan berfungsi sebagai "contoh kerja". Misalnya, anotasi pada titik data penting di sebuah grafik tren dapat menjelaskan "Mengapa ada lonjakan donasi di sini?", secara langsung membantu pengguna membangun pemahaman yang lebih dalam.

### 3.1.3 Development

Tahap pengembangan merupakan fase realisasi teknis dari arsitektur sistem yang telah dirancang pada sub-bab 3.1.2.1. Pada tahap ini, peneliti menerjemahkan cetak biru konseptual menjadi sebuah aplikasi fungsional. Pengembangan sistem ini sepenuhnya mengadopsi tumpukan teknologi modern yang terdiri dari kerangka kerja Next.js untuk logika aplikasi dan antarmuka, serta Supabase sebagai platform Backend-as-a-Service (BaaS). Pilihan teknologi ini didasarkan pada kemampuannya untuk secara langsung menjawab rumusan masalah penelitian, terutama dalam hal penyajian data real-time, penegakan keamanan data yang granular, dan penyederhanaan arsitektur pengembangan secara keseluruhan.

Pengembangan sistem dibagi menjadi beberapa area utama yang saling terintegrasi yaitu pembangunan fondasi backend dengan supabase dimana supabase berfungsi sebagai tulang punggung (*backbone*) dari sistem ini, lalu menyediakan

seluruh layanan backend yang diperlukan dalam satu platform terpadu. dan pengembangan aplikasi full-stack dengan Next.js. Kolaborasi antara Supabase sebagai backend yang kuat dan Next.js sebagai kerangka kerja full-stack yang fleksibel memungkinkan peneliti untuk mewujudkan desain sistem yang telah direncanakan secara efisien. Pendekatan ini menghasilkan sebuah dasbor operasional yang tidak hanya fungsional, tetapi juga aman, responsif, dan mampu menyajikan data secara real-time, yang merupakan inti dari sistem pengelolaan donasi yang akuntabel.

### 3.1.4 Implementation

Setelah berhasil merancang *dashboard* dan sistem, tahap berikutnya adalah melakukan uji coba langsung dengan menerapkan hasil dari *dashboard* dan sistem tersebut di lingkungan perguruan tinggi, khususnya di Universitas Pendidikan Indonesia. Ada dua hal terkait dengan donatur beasiswa mahasiswa yang dapat digunakan sebagai sarana untuk mengimplementasikan hasil dari *dashboard* tersebut, diantaranya:

1. IKA UPI

Ikatan Keluarga Alumni (IKA) UPI. IKA UPI merupakan sebuah organisasi alumni yang aktif berkontribusi melalui program arisan untuk menyediakan beasiswa bagi mahasiswa yang berprestasi namun kurang mampu secara finansial. IKA UPI ini untuk mengetahui apakah *dashboard* yang dibentuk telah memenuhi kebutuhan donasi yang akuntabel atau belum. Peneliti akan memfokuskan pada implementasi *dashboard* dengan data hasil donasi untuk beasiswa mahasiswa.

2. Partisipasi Dosen

Dosen di UPI seringkali menjadi donatur yang peduli terhadap pendidikan dan kesejahteraan mahasiswa. Dengan menerapkan hasil dari *dashboard* dan sistem ini, diharapkan proses pengelolaan dan pemantauan donasi beasiswa dari dosen dapat lebih terstruktur dan mudah dipantau, serta memberikan dukungan yang lebih efektif kepada mahasiswa penerima beasiswa.

### 3. Partisipasi Masyarakat Umum

Selain dari lingkungan internal UPI, sistem ini juga akan diimplementasikan untuk menjangkau masyarakat umum yang memiliki kepedulian terhadap pendidikan. Tujuan dari pelibatan masyarakat umum adalah untuk memperluas sumber donasi dan memberikan kemudahan bagi siapa saja yang ingin berkontribusi dalam mendukung mahasiswa berprestasi. Pada tahap ini, uji coba akan difokuskan pada kemudahan akses, kejelasan informasi, dan transparansi proses donasi bagi pengguna eksternal yang tidak memiliki afiliasi langsung dengan universitas. Dengan demikian, *dashboard* ini diharapkan tidak hanya menjadi alat internal, tetapi juga jembatan filantropi antara masyarakat luas dengan mahasiswa UPI yang membutuhkan.

#### 3.1.5 Evaluation

Tahap *evaluation* dilakukan dengan melakukan penilaian terhadap efektifitas *dashboard* dan sistem perolehan donasi beasiswa mahasiswa yang telah diimplementasikan sebelumnya. Peneliti akan melakukan bentuk pengukuran yaitu:

##### 1. *End-User Computing Satisfaction*

Penelitian ini akan menggunakan metode *End-User Computing Satisfaction* (EUCS) sebagai kerangka evaluasi. EUCS adalah pendekatan yang digunakan untuk mengevaluasi kepuasan pengguna terhadap penggunaan dan kinerja sistem informasi atau perangkat lunak yang digunakan dalam konteks pekerjaan atau kegiatan sehari-hari. Tahap pertama dalam EUCS adalah identifikasi dimensi-dimensi kepuasan pengguna yang relevan dengan sistem yang diteliti. Dimensi-dimensi tersebut mencakup aspek *content*, *accuracy*, *format*, *ease of use*, dan *timeliness* dari *dashboard* operasional. Dengan demikian, penggunaan metode *End-User Computing Satisfaction* diharapkan dapat memberikan pemahaman yang komprehensif tentang sejauh mana *dashboard* operasional tersebut memenuhi kebutuhan dan harapan pengguna dalam perolehan donasi beasiswa yang akuntabel di

tingkat perguruan tinggi serta memperoleh umpan balik langsung dari pengguna agar dapat mengidentifikasi kekurangan dalam desain atau fungsionalitas *dashboard* yang mungkin tidak terdeteksi melalui metode evaluasi teknis.

### 3.1.6 Kesimpulan dan Saran

Setelah semua langkah telah diselesaikan, langkah terakhir dalam penelitian ini adalah mengevaluasi apakah *dashboard* dan sistem perolehan donasi beasiswa mahasiswa yang telah dikembangkan telah menjawab masalah yang dirumuskan dan mencapai tujuan penelitian. Akhirnya, disusul dengan pemberian saran yang mengandung berbagai masukan konstruktif untuk meningkatkan kualitas penelitian di masa mendatang.

### 3.2 Alat Penelitian

Alat yang digunakan dalam perancangan *dashboard* pengelolaan donasi mahasiswa adalah laptop pribadi. Perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini yaitu,

a. Perangkat Keras:

- Processor: Intel Core i5-9300H CPU 2.40GHz
- Kapasitas RAM: 16 GB
- Kapasitas SSD: 1000GB
- Layar laptop: 15.6” Full HD display

b. Perangkat Lunak

- Sistem Operasi: Windows 10 Pro
- Framework: Next.js 15
- Web Browser: Google Chrome v118.0.5993.118 (Official Build) (64-bit)
- Text editor: Visual Studio Code