

Financial and Information Dashboard Design in GMIM Southwest Manado Region

Desain Dashboard Keuangan dan Informasi di GMIM Wilayah Manado Barat Daya

Teofilus G. V. Pangemanan¹⁾, Arie S. M. Lumenta²⁾, Yaulie D. Y. Rindengan³⁾

Dept. of Electrical Engineering, Sam Ratulangi University Manado, Kampus Bahu St., 95115, Indonesia
e-mails : 18021106101@student.unsrat.ac.id¹⁾, al@unsrat.ac.id²⁾, rindengan@unsrat.ac.id³⁾

Received: 15 June 2023; revised: 08 August 2023; accepted: 15 October 2023

Abstract — The presence of ICT in the current era of information development has an impact on society. The financial centralization process within the scope of the Evangelical Christian Church in Minahasa (GMIM) is still carried out conventionally. To answer the problems above and meet user needs, it is deemed necessary to provide an innovation in the form of designing an integrated congregational information system. This research will be carried out using Agile development with the SCRUM methodology model. The design of this information system is in the form of a website in the form of a Dashboard using the JavaScript programming language with the ReactJS framework for the Front-End and NodeJS for the Back-End and then integrated into financial applications in every congregation in the Southwest Manado region (E-JIB) using the API. The Monitoring Dashboard can run well but requires improvements to other features to make it more optimal. Suggestions that can be made are integrating a financial information system in the form of income and expenditure data for each congregation to make it easier to carry out financial reporting, but you must also consider security aspects when you have entered the stage of using a banking account.

Key words— API; Dashboard; JavaScript; NodeJS; ReactJS.

Abstrak — Kehadiran TIK dalam era perkembangan informasi saat ini memberikan dampak bagi masyarakat. Proses sentralisasi keuangan dalam lingkup pelayanan Gereja Masehi Injili di Minahasa (GMIM) masih dilakukan secara konvensional. Untuk menjawab permasalahan di atas dan memenuhi kebutuhan pengguna maka dianggap perlu untuk memberikan suatu inovasi berupa merancang sistem informasi keuangan jemaat yang terintegrasi. Penelitian ini akan dilakukan dengan menggunakan Agile development dengan model SCRUM methodology. Rancangan sistem informasi ini berupa website dalam bentuk Dashboard dengan menggunakan bahasa pemrograman JavaScript dengan framework ReactJS untuk bagian Front-End dan NodeJS untuk bagian Back-End untuk kemudian diintegrasikan ke aplikasi keuangan yang ada di setiap jemaat di wilayah Manado Barat Daya (E-JIB) menggunakan API. Dashboard Monitoring sudah dapat berjalan dengan baik namun memerlukan perbaikan pada bagian fitur lainnya agar menjadi lebih optimal. Saran yang dapat yaitu mengintegrasikan sistem informasi keuangan berupa data pemasukan dan pengeluaran setiap jemaat memberikan kemudahan dalam melakukan pelaporan keuangan, tetapi juga harus mempertimbangkan aspek *security* apabila sudah masuk pada tahap menggunakan rekening perbankan.

Kata kunci — API; Dashboard; JavaScript; NodeJS; ReactJS.

I. PENDAHULUAN

Di era Teknologi Informasi dan Komunikasi (TIK) saat ini

yang semakin pesat, manusia disuguhkan dengan kemudahan-kemudahan dalam beraktifitas. Kehadiran TIK dalam era perkembangan informasi saat ini memberikan dampak bagi masyarakat. Literasi TIK menjadi acuan kesiapan bagi masyarakat dalam pengimplementasian teknologi informasi di kehidupan sosial. Teknologi informasi dan komunikasi (TIK) juga meliputi teknologi komunikasi untuk mengirim pesan dan tidak hanya sebatas pada teknologi komputer yang digunakan untuk memproses dan menyimpan informasi [1].

Ketersediaan data keuangan yang berlimpah dari berbagai sistem dan sumber telah menjadi tantangan tersendiri bagi banyak perusahaan. Keterpisahan data dari sistem yang berbeda dapat menyebabkan kesulitan dalam menganalisis informasi keuangan secara menyeluruh dan *real-time*. Oleh karena itu, integrasi data dan desain *dashboard* keuangan yang efektif menjadi semakin penting sebagai solusi untuk mengatasi permasalahan ini.

Integrasi data adalah proses menggabungkan data dari berbagai sumber atau sistem menjadi satu set data terpadu. Pendekatan ini memungkinkan organisasi untuk memiliki akses terhadap data yang konsisten dan terpercaya dari berbagai departemen atau unit bisnis. Seiring dengan integrasi data, desain *dashboard* keuangan berperan penting dalam menyajikan data keuangan secara visual dan informatif. *Dashboard* keuangan adalah alat yang memberikan pandangan ringkas tentang kinerja keuangan organisasi melalui grafik, diagram, dan visualisasi data lainnya. Sebuah *dashboard* keuangan yang efektif harus menyajikan metrik keuangan yang relevan dan penting secara jelas dan mudah dipahami.

Sistem di dalam suatu organisasi dikenal sebagai sistem informasi. Sistem ini menggabungkan kebutuhan pengolahan transaksi harian untuk mendukung operasi manajerial dan kegiatan strategis serta memberikan laporan yang diperlukan oleh pihak luar [2].

Namun, meskipun integrasi data dan desain *dashboard* keuangan memiliki manfaat yang signifikan, belum banyak penelitian yang menggabungkan keduanya secara holistik dalam konteks organisasi. Oleh karena itu, tujuan penelitian ini untuk menyelidiki dan mengidentifikasi bagaimana integrasi data yang baik dapat mempengaruhi desain *dashboard* keuangan yang efektif dalam membantu pengambilan keputusan bisnis yang lebih baik.

Untuk menjawab permasalahan di atas dan memenuhi kebutuhan pengguna maka dianggap perlu untuk memberikan suatu inovasi berupa merancang sebuah *dashboard* keuangan

yang terintegrasi. Rancangan sistem informasi ini berupa Dashboard informasi keuangan dengan menggunakan bahasa pemrograman JavaScript dengan *framework* ReactJS untuk bagian *Front-End* dan NodeJS untuk bagian *Back-End* untuk kemudian di integrasikan ke aplikasi keuangan yang ada di setiap jemaat di wilayah Manado Barat Daya (E-JIB).

A. Penelitian Terkait

Heri et al. melakukan penelitian tentang Sistem Informasi Pengolahan Data Keuangan Berbasis Web yang bertujuan untuk membantu proses pengambilan keputusan di bidang keuangan dengan menyediakan informasi yang lebih mudah dan lebih cepat untuk diakses. Hasilnya menunjukkan bahwa data akan terintegrasi dan terstruktur dengan baik [16].

Penelitian yang dilakukan oleh Excelsis dkk berjudul Rancang Bangun aplikasi Pengelola Keuangan gereja Berbasis Android bertujuan untuk merancang aplikasi pengelola keuangan gereja yang dapat mencatat transaksi dan memberikan informasi keuangan kepada jemaat. Aplikasi ini bermanfaat sebagai alat bantu pengelolaan keuangan dalam organisasi gereja [17].

Billy dkk. melakukan penelitian tentang Rancang Bangun Web Service Sistem Informasi Keuangan GMIM Wilayah Tomohon 3 dengan tujuan untuk mengetahui perkembangan di bidang keuangan gereja karena pengelolaan data keuangan di Wilayah Tomohon 3 GMIM masih dilakukan secara manual, yang tidak efektif dan efisien [18].

Studi yang dilakukan oleh Yanuardi dkk. tentang Rancang Bangun Sistem Informasi Keuangan pada PT. Secret Discoveries Travel and Leisure Berbasis Web menemukan bahwa karena banyaknya laporan yang ada membuat pegawai kewalahan untuk menatanya, aplikasi ini diharapkan dapat membantu perusahaan travel mengelola data dengan baik [19].

Tujuan dari penelitian yang dilakukan oleh Hafiz et al. adalah untuk menggunakan model Rapid Application Development (RAD) untuk merancang sistem informasi keuangan untuk PT. Asuransi Mega Pratama. Sebuah sistem informasi yang dihasilkan dari penelitian ini dapat membantu perusahaan mengelola keuangan mereka dengan lebih efektif [20].

Zaen dkk. melakukan penelitian berjudul Sistem Informasi Keuangan pada Dinas Perhubungan Kabupaten Lombok Tengah Berbasis Web. Tujuan dari penelitian ini adalah untuk merancang sistem informasi berbasis web dengan melihat masalah saat ini seperti pembuatan kwitansi, pengelolaan data, dan perencanaan penggunaan uang [21].

Studi kasus yang dilakukan oleh Happy et al. tentang Perancangan Sistem Informasi Manajemen Keuangan Berbasis Web pada PT. Karya Swadaya Abadi bertujuan untuk membuat sistem informasi manajemen berbasis web yang akan meningkatkan efisiensi proses bisnis dengan menyediakan laporan keuangan, data asset perusahaan, dan data karyawan. [22].

Penelitian ini dilakukan oleh I Nyoman et al. tentang Rancang Bangun Sistem Informasi Laporan Keuangan pada SMP Nasional Berbasis Web. Tujuan dari penelitian ini adalah untuk melakukan analisis sistem untuk memenuhi kebutuhan sistem informasi di SMP Nasional Denpasar, dengan hasil

membantu sekolah dalam mengelola dan mengawasi aktivitas keuangan mereka. [23].

B. Sistem Informasi

Sistem informasi adalah kombinasi prosedur kerja, informasi, orang, dan teknologi informasi yang dirancang untuk mencapai tujuan organisasi [1]. Sistem di dalam suatu organisasi dikenal sebagai sistem informasi. Sistem ini menggabungkan kebutuhan pengolahan transaksi harian untuk mendukung operasi manajerial dan kegiatan strategis serta memberikan laporan yang diperlukan oleh pihak luar [2].

Komponen utama dari sistem informasi meliputi: Orang (*People*), Merupakan pengguna atau pelaku dalam sistem informasi. Orang adalah pihak yang berinteraksi dengan data dan teknologi untuk mencapai tujuan tertentu. Teknologi (*Technology*), Meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan untuk mengumpulkan, menyimpan, dan memproses data. Teknologi juga mencakup infrastruktur jaringan dan sistem operasi yang mendukung pengoperasian sistem informasi. Proses (*Process*), Merupakan alur atau rangkaian kegiatan yang dilakukan untuk mengubah data mentah menjadi informasi yang berguna. Proses mencakup aktivitas pengumpulan data, pemrosesan data, analisis, dan presentasi informasi. Data (*Data*), Merupakan fakta atau kumpulan fakta yang diorganisir dan direpresentasikan dalam bentuk angka, teks, gambar, atau suara. Data menjadi bahan dasar untuk menghasilkan informasi dalam sistem informasi. Kebijakan (*Policy*), Merupakan aturan, prosedur, dan pedoman yang mengatur penggunaan sistem informasi. Kebijakan bertujuan untuk memastikan keamanan, integritas, dan privasi data, serta mengatur hak akses pengguna dalam sistem.

C. Integrasi

Dalam konteks ilmiah, integrasi seringkali mengacu pada penggabungan berbagai konsep, data, atau sistem yang berbeda menjadi satu entitas yang lebih kompleks dan terorganisir. Integrasi dalam teknologi informasi merujuk pada menggabungkan sistem-sistem atau aplikasi-aplikasi yang berbeda agar dapat beroperasi bersama dan berbagi data secara efektif [3]. Pada dasarnya, integrasi melibatkan proses penggabungan dan konsolidasi elemen-elemen yang berbeda untuk menciptakan hubungan atau koneksi yang lebih kokoh dan terarah. Tujuan utama dari integrasi adalah untuk menciptakan keselarasan dan kesatuan di antara elemen-elemen yang sebelumnya berdiri sendiri.

D. Basis Data

Basis data adalah suatu kumpulan data yang terstruktur dan terorganisir dengan baik, yang disimpan secara elektronik dalam sistem komputer. Basis data dirancang untuk menyimpan, mengelola, dan mengakses data dengan efisien dan aman, sehingga memungkinkan pengguna untuk melakukan berbagai operasi dan analisis.

Basis data adalah kumpulan data yang saling terkait yang disimpan di dalam komputer secara terstruktur dan dirancang

untuk diakses dengan menggunakan bahasa *query* (kueri) yang spesifik [4].

Basis data adalah kumpulan data yang terhubung yang disimpan secara bersama-sama pada media, diatur dengan cara tertentu, dan dapat dimanipulasi dengan software untuk tujuan tertentu [9].

Basis data adalah kumpulan data yang terintegrasi secara bersama-sama dengan deskripsi data yang relevan, yang direpresentasikan dalam bentuk independen dari aplikasi yang menggunakan basis data tersebut [5].

Basis data adalah kumpulan data yang saling terkait, yang diorganisir agar data dapat diakses dan dikelola dengan efisien [6].

Secara umum, basis data adalah kumpulan data yang terstruktur, diorganisir, dan terintegrasi dengan baik, yang disimpan dalam sistem komputer untuk memfasilitasi akses, pengelolaan, dan analisis data.

E. Integrasi Data

Integrasi data adalah proses yang kompleks dan penting dalam dunia teknologi informasi yang menggabungkan dan menyatukan data dari berbagai sumber, sistem, atau platform menjadi satu kesatuan yang terpadu dan konsisten. Dalam konteks ilmiah, integrasi data melibatkan berbagai tahapan, termasuk pengumpulan data dari berbagai sumber, transformasi data ke dalam format yang sesuai dan seragam, serta penggabungan data ke dalam satu repositori sentral.

Integrasi data mengacu pada proses menggabungkan dan mengkonsolidasikan data dari berbagai sumber atau sistem menjadi satu set data yang terpadu dan kohesif.

Ada berbagai teknik dan pendekatan integrasi data, antara lain: *Integrasi Data Manual*, Metode ini melibatkan pengumpulan data secara manual dari berbagai sumber dan menggabungkannya menjadi satu set data. Pendekatan ini dapat memakan waktu, rentan terhadap kesalahan, dan mungkin tidak cocok untuk proyek integrasi data dalam skala besar. *Integrasi Data Batch*, Pada integrasi data batch, data dari berbagai sumber diekstraksi, diubah, dan dimuat (ETL) ke dalam gudang data sentral atau data mart pada interval yang terjadwal. Pendekatan ini umumnya digunakan saat menghadapi volume data yang besar, karena memungkinkan konsolidasi dan pemrosesan data selama jam-jam *non-peak*. *Integrasi Data Real-time*, Metode ini melibatkan integrasi data secara real-time atau hampir *real-time* seiring dengan dihasilkannya atau pembaruan data. Biasanya digunakan dalam skenario di mana informasi yang terbaru sangat penting, seperti dalam perdagangan keuangan, transaksi *online*, atau sistem pemantauan. *Virtualisasi Data*, Virtualisasi data mengintegrasikan data dari berbagai sumber secara virtual, tanpa perlu memindahkan atau menggandakan data menjadi repositori sentral. Ini menyediakan pandangan data yang terpadu tanpa perlu menggunakan data warehouse penuh, sehingga memungkinkan akses data yang lebih cepat dan fleksibel. *Enterprise Service Bus (ESB)*, adalah alat *middleware* yang memfasilitasi komunikasi antara sistem-sistem yang berbeda dengan bertindak sebagai tulang punggung pengiriman pesan. Ini memungkinkan integrasi data dengan memungkinkan berbagai aplikasi dan layanan untuk saling

bertukar data secara mulus. Integrasi Melalui API (*Application Programming Interface*), memungkinkan berbagai aplikasi perangkat lunak untuk berkomunikasi dan berbagi data satu sama lain. Integrasi data dapat dicapai dengan memanfaatkan API untuk mengekstrak dan menyinkronkan data antara berbagai sistem.

F. Application Programming Interface (API)

Konsep fungsi antarmuka pemrograman aplikasi, atau API, adalah cara agar suatu aplikasi dapat diakses dan digunakan oleh orang lain tanpa mengubah struktur kode utama atau database sistem. API juga memudahkan komunikasi antar sistem di platform yang berbeda [7], [8]. Bahasa pemrograman antarmuka (API) adalah kumpulan alat pemrograman, protokol, dan aturan yang memungkinkan berbagai perangkat lunak atau aplikasi berkomunikasi dan berinteraksi satu sama lain. API menyediakan antarmuka yang jelas dan terdefinisi dengan fungsi-fungsi atau layanan yang dapat diakses oleh pengembang perangkat lunak untuk mengakses fungsionalitas yang telah disediakan oleh sistem atau platform tertentu.

G. Representational State Transfer (REST) API

REST API (*Representational State Transfer Application Programming Interface*) adalah salah satu jenis antarmuka aplikasi yang digunakan untuk berkomunikasi dan bertukar data antara sistem atau perangkat lunak yang berbeda melalui protokol HTTP (*Hypertext Transfer Protocol*). REST API menyediakan cara yang terstandarisasi dan terdefinisi untuk mengakses dan mengelola sumber daya (*resources*) yang disediakan oleh server. Setiap sumber daya direpresentasikan oleh URL (*Uniform Resource Locator*), dan REST API menggunakan metode HTTP seperti GET, POST, PUT, DELETE, dll, untuk berkomunikasi dengan sumber daya tersebut.

REST API beroperasi pada prinsip-prinsip sebagai berikut: *Stateless*, Setiap permintaan ke server harus berisi semua informasi yang diperlukan, dan tidak menyimpan keadaan (*state*) antara permintaan. Artinya, *server* tidak menyimpan informasi tentang klien antara permintaan-permintaan yang berbeda. *Resources*, Sumber daya adalah entitas yang dapat diakses melalui URL, seperti data, gambar, dokumen, atau layanan lainnya. Setiap sumber daya memiliki URL yang unik. *Methods*, Metode HTTP digunakan untuk berinteraksi dengan sumber daya. Metode yang umum digunakan dalam REST API adalah GET (mengambil data), POST (membuat data baru), PUT (memperbarui data), dan DELETE (menghapus data). *Representations*, Data dari sumber daya dapat direpresentasikan dalam berbagai format, seperti JSON (*JavaScript Object Notation*) atau XML (*Extensible Markup Language*), yang memungkinkan aplikasi untuk memahami dan memproses data tersebut.

H. Website

Website merupakan kumpulan dari halaman web yang telah dipublikasikan di jaringan internet dan memiliki URL (*Uniform Resource Locator*) yang bisa diakses oleh semua pengguna internet dengan cara mengetik alamatnya. Halaman website

terdiri dari berbagai komponen seperti gambar, video, suara, teks dll [10]. Website adalah kumpulan halaman yang saling terkait yang menampilkan informasi seperti teks, gambar diam atau gerak, animasi, suara, dan gabungan dari semua ini. Website ini dapat bersifat statis atau dinamis dan terhubung satu sama lain melalui jaringan halaman [11].

I. Konsep Model View Controller (MVC)

MVC adalah singkatan dari Model View Controller. Sebenarnya, MVC adalah sebuah pola pemrograman yang membedakan logika bisnis (alur pikir), logika data (penyimpanan data), dan logika presentasi (antarmuka aplikasi). Secara sederhana, MVC membedakan antara desain, data, dan proses [12].

J. JavaScript

JavaScript adalah bahasa pemrograman web yang pemrosesannya dilakukan di sisi klien dan digunakan untuk aplikasi berbasis web yang berbasis client serta script tampilan pendukung yang memungkinkan halaman web melakukan fungsi tambahan yang tidak dapat dilakukan oleh script HTML [13]. JavaScript adalah bahasa script dinamis yang dapat digunakan untuk menciptakan interaksi pada halaman HTML statis karena berjalan di sisi client. [14].

K. ReactJS

React atau biasa dinamakan juga React.js atau ReactJS adalah *library* JavaScript yang semula dikemukakan dari Facebook untuk memberikan fasilitas pembuatan dari pada sebuah komponen antarmuka yang interaktif, stateful, dan gampang untuk digunakan kembali. ReactJS sangat cocok digunakan untuk *rendering* antarmuka yang kompleks dengan performa tinggi [15]. ReactJS adalah library yang dapat dibagi menjadi beberapa komponen saat membuat antarmuka pengguna. Ini adalah bagian dari konsep Model View Controller (MVC), yang berarti React hanya menangani aspek tampilan antarmuka dengan pengguna tanpa menangani cara mendapatkan data atau menghubungkan ke basis data [15].

L. NodeJS

NodeJS (Node) adalah platform pengembangan source terbuka yang memungkinkan pengeksekusi sisi server kode pada JavaScript. Ini sangat membantu dalam pembuatan aplikasi yang membutuhkan koneksi terus-menerus dari browser ke server. Aplikasi waktu nyata seperti chat, feeds berita, dan notifikasi push web adalah beberapa contoh aplikasi yang menggunakan NodeJS. Node.js memproses permintaan secara asynchronous, yang berarti setiap permintaan yang diterima akan dimasukkan ke dalam tumpukan event yang konstan dan akan mengirimkan permintaan kecil satu demi satu tanpa menunggu respons. Ini adalah pergeseran dari model utama, yang dikenal sebagai sinkron, yang menjalankan proses yang lebih besar dan rumit dan menunggu respon yang sesuai untuk setiap perintah sebelum melanjutkan permintaan berikutnya.

II. METODE

Penelitian ini akan dilakukan dengan menggunakan Agile development dengan model SCRUM methodology. Berikut alur dari Agile:

A. Requirement Analysis

Analisis adalah kegiatan untuk mengidentifikasi permasalahan yang ada dan solusi serta kebutuhan yang diperlukan dalam penyelesaian masalah tersebut.

B. Design

Desain adalah sebuah proses penggambaran solusi dari hasil analisis yang dilakukan pada tahap sebelumnya dalam bentuk representasi perangkat lunak sebelum dilakukan proses *coding*.

C. Development

Tahap ini adalah tahap implementasi gambaran sistem yang telah dibuat sebelumnya ke dalam suatu program. Tahap ini dilakukan dengan cara pemrograman atau *coding* program dengan bantuan perangkat komputer. Desain keseluruhan *dashboard* akan dibuat dalam tahap ini.

D. Testing

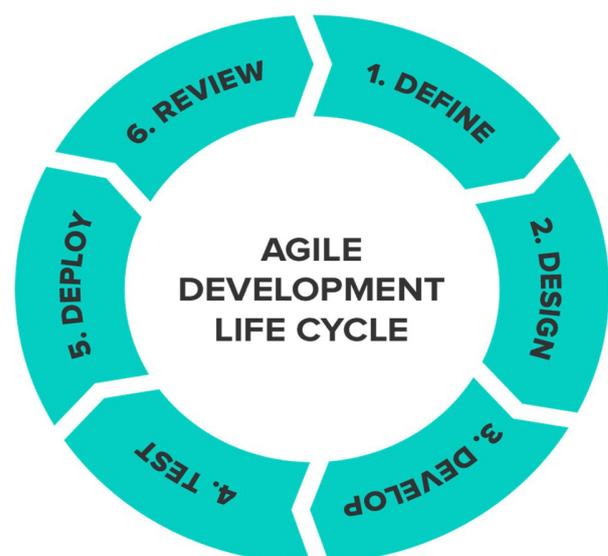
Testing adalah proses pengujian hasil pembuatan aplikasi yang telah dibuat. Proses pengujian ini dilakukan untuk menilai keberhasilan aplikasi yang dibuat berdasarkan tujuan awal pembuatan.

E. Deployment

Proses ini dilakukan untuk memastikan kualitas perangkat lunak yang dibuat melalui pengujian kualitas sistem. Jika sistem memenuhi syarat, perangkat lunak tersebut sudah siap untuk dikembangkan di masa mendatang.

F. Review

Tahap selanjutnya yaitu review pada perangkat lunak yang telah di uji sebelumnya. Jika telah sesuai dengan yang dibutuhkan maka produk berhasil. Namun jika ada keluhan atau ketidaknyamanan saat menggunakan aplikasi, maka akan dilakukan pengembangan ulang.



Gambar 1. Metode Agile

III. HASIL DAN PEMBAHASAN

A. Requirement Analysis

1) Analisis Sistem yang Berjalan

Langkah awal yang dilakukan oleh penulis adalah mengidentifikasi dan menganalisa permasalahan yang dihadapi serta bagaimana proses yang sedang berjalan dalam sentralisasi keuangan di jemaat. Penulis melakukan wawancara dan observasi proses kerja yang sedang berjalan kepada beberapa pegawai gereja di jemaat GMIM Imanuel Bahu untuk mendapatkan gambaran penyelesaian masalah yang dihadapi.

2) Analisis Sistem yang Diusulkan

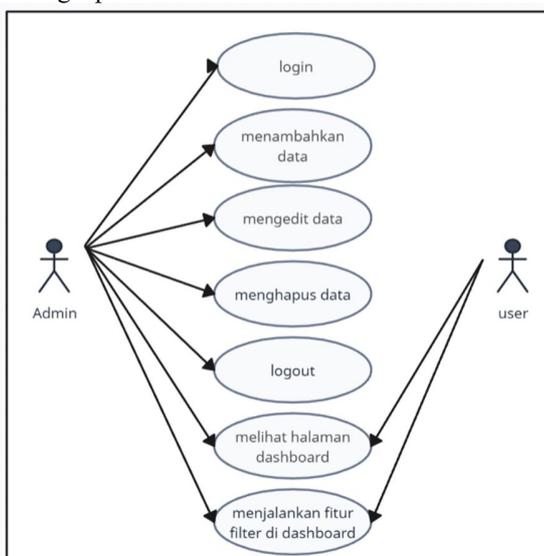
Berdasarkan hasil analisa terhadap sistem yang berjalan sebelumnya maka bisa diketahui bahwa sistem yang berjalan pada tingkat jemaat sampai pada tingkat wilayah masih bersifat manual. Efektivitas proses bisnis yang berjalan seperti ini dianggap kurang efisien dan perlu adanya inovasi terbaru. Dalam Penelitian ini penulis akan mengusulkan sebuah sistem informasi keuangan berupa *Dashboard* Wilayah yang dapat membantu proses pelaporan keuangan ke wilayah menjadi lebih efisien dan transparan dengan mengintegrasikan ke setiap aplikasi keuangan yang ada di jemaat-jemaat dalam lingkup Wilayah Manado Barat Daya.

B. Design

1) Use Case Diagram

Login, Admin harus melakukan *login* dengan menginputkan *username* dan *password* untuk dapat mengakses E-JIB. Admin dapat menambahkan data pada E-JIB yang nantinya akan di tampilkan pada *dashboard* keuangan.

Admin dapat mengedit data pada E-JIB yang nantinya akan di tampilkan pada *dashboard* keuangan. Admin dapat menghapus data pada E-JIB dan kemudian secara otomatis akan di *update* pada *dashboard* keuangan. Admin Menampilkan secara garis besar setiap data yang di input dan tersimpan pada *database* dan diampilkkan pada *dashboard* keuangan. Untuk *user* dapat melihat/*view dashboard* tanpa bisa menginput, mengedit atau menghapus data.



Gambar 2. Use Case Diagram

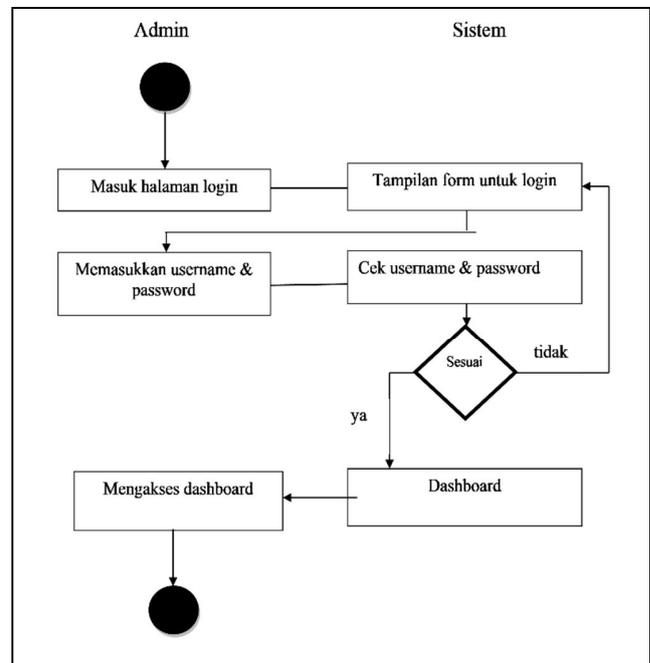
Admin dan user dapat menjalankan fitur filter di *dashboard* keuangan. Keluar dari *dashboard* (lihat Gambar 1).

2) Activity Diagram

Admin mengunjungi halaman login E-JIB untuk memasukkan username dan password pada form login, kemudian melakukan input data pada fitur penginputan data. Data berhasil tersis akan di simpan dalam sebuah *database*. Kemudian data yang sudah tersimpan di *database* akan di tampilkan pada *Dashboard* Wilayah yang telah rancang sesuai dengan *request* dari sistem. *Website Dashboard* hanya bersifat *view* data tanpa bisa melakukan penginputan data sehingga user hanya dapat melakukan *view* saja. Sementara itu, hubungan antara E-JIB dan *Dashboard* hanya bersifat searah sehingga diperlukan adanya API untuk menjadi penghubung antara 2 aplikasi tersebut (lihat Gambar 2).

3) Desain tampilan

Desain Tampilan awal antar muka dari *Dashboard* dapat di lihat pada Gambar 3.

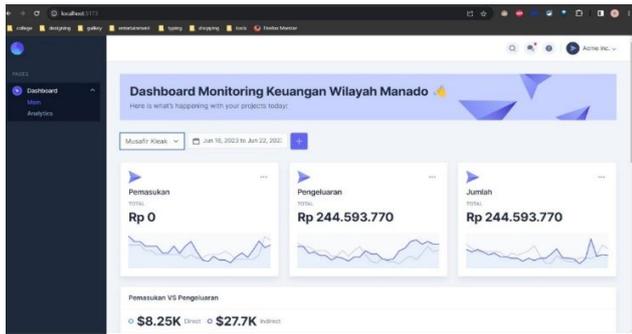


Gambar 3. Activity Diagram

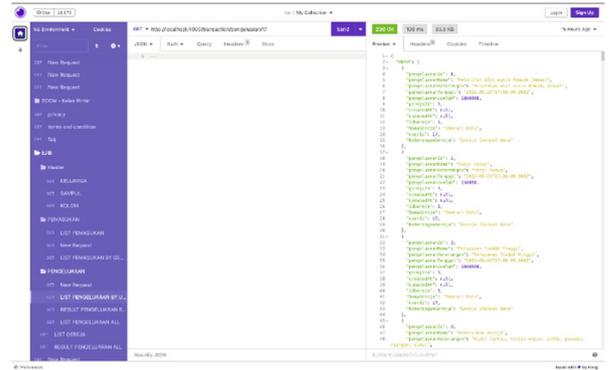


Gambar 4. Tampilan Desain awal

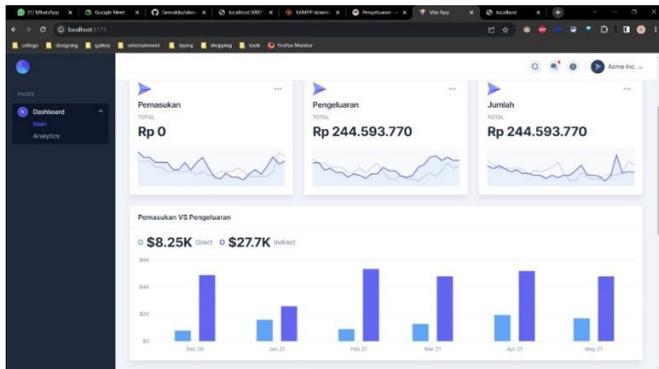
C. Development



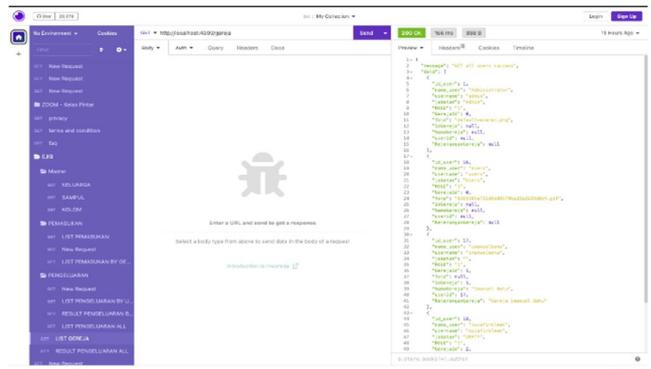
Gambar 5. Tampilan Halaman Dashboard



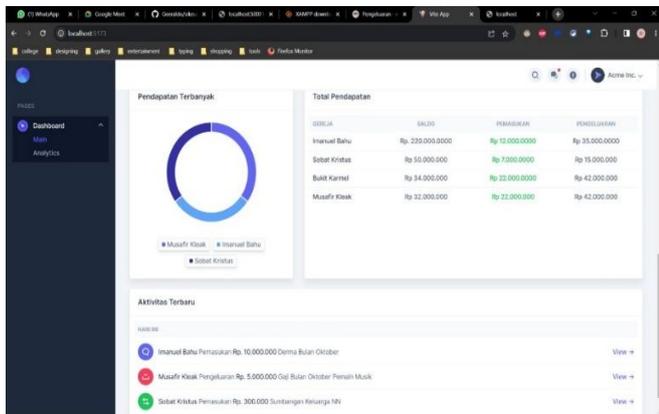
Gambar 9. Tampilan API Pengeluaran



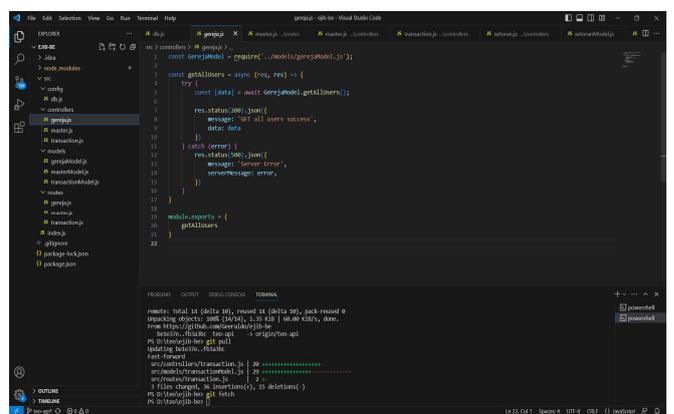
Gambar 6. Tampilan Halaman Dashboard (2)



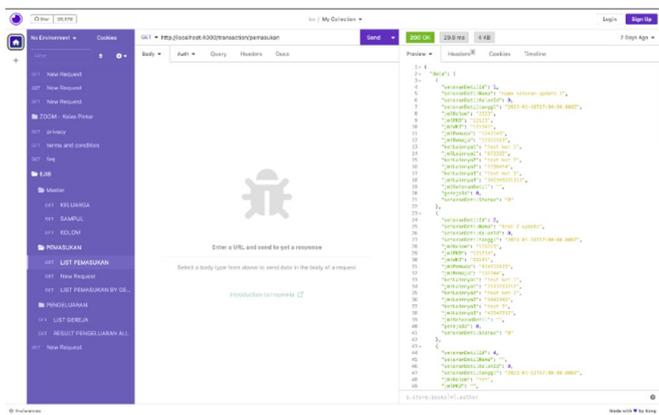
Gambar 10. Tampilan API List Gereja



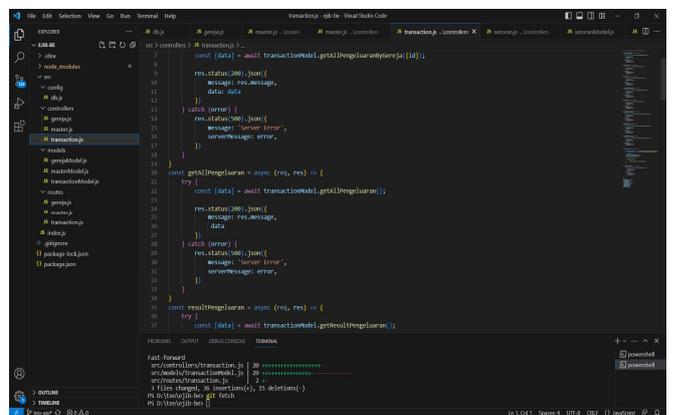
Gambar 7. Tampilan Halaman Dashboard (3)



Gambar 11. Tampilan koneksi code API



Gambar 8. Tampilan API Pemasukan



Gambar 12. Tampilan koneksi code API (2)

1) *Halaman Dashboard*

Untuk dapat masuk ke halaman *dashboard*, *user* masuk dengan mengakses URL yang telah di tentukan. *Dashboard* yang di rancang tanpa menggunakan *username* dan *password* sehingga siapa saja bisa mengaksesnya. Pada halaman *dashboard* terdapat fitur yang menampilkan pemasukan dan pengeluaran serta jumlah saldo dari setiap jemaat yang ada di Wilayah Manado Barat Daya. Ada fitur untuk memfilter jemaat mana saja yang hanya akan di tampilkan pada *dashboard* (Lihat Gambar 4-6).

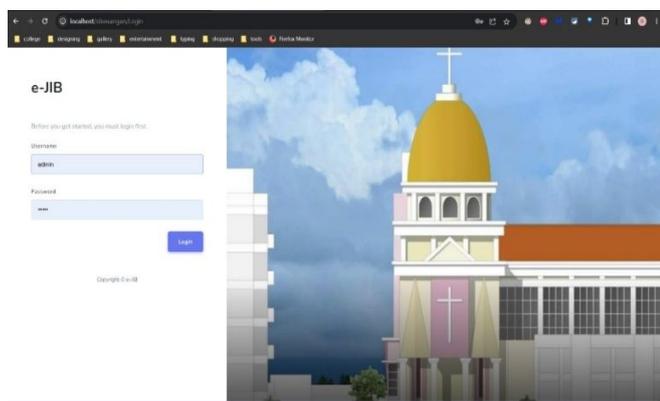
Tampilan API dan Koneksi code API dapat di lihat pada Gambar 7-11.

Data yang di tampilkan pada *Dashboard* Wilayah di input dari aplikasi E-JIB di setiap jemaat. Kemudian hasil penginputan tersebut ditampilkan pada *Dashboard*.

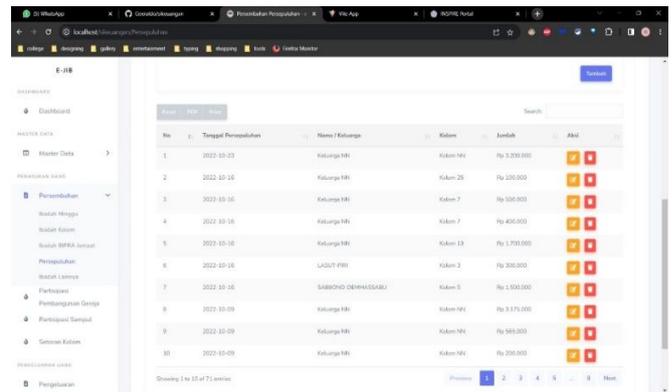
Halaman utama E-JIB berisikan rincian pemasukan dari setiap aspek pelayanan dan juga pengeluaran serta jumlah saldo yang ada. Untuk dapat mengakses E-JIB setiap jemaat diberikan akun seperti *username* dan *password* masing-masing. Admin dapat mengolah data *user* yang ada pada halaman ini seperti menghapus, menambahkan dan mengubah data (lihat Gambar 12-13).

Salah satu fitur pemasukan yaitu persembahan persepuluhan (Gambar 14). Setelah data input, data dikumpulkan dalam bentuk tabel hasil penginputan. (Gambar 15).

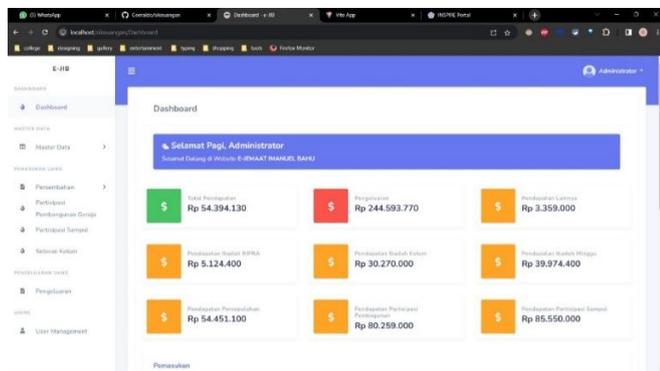
Sama seperti penginputan data pemasukan, hasil daripada penginputan data pengeluaran dikumpulkan dan ditampikan dalam bentuk tabel rekapan. (Gambar 17).



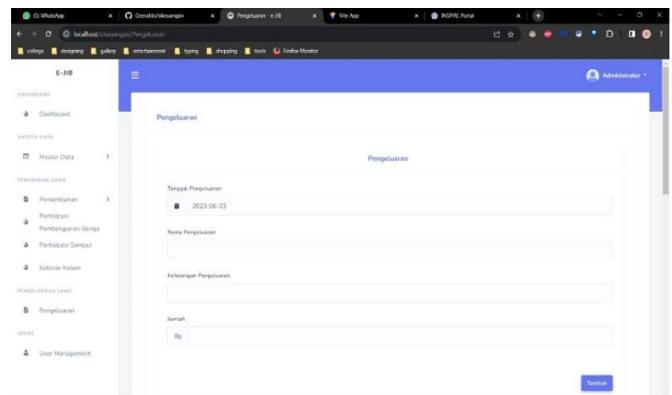
Gambar 13. Tampilan Halaman Login E-JIB



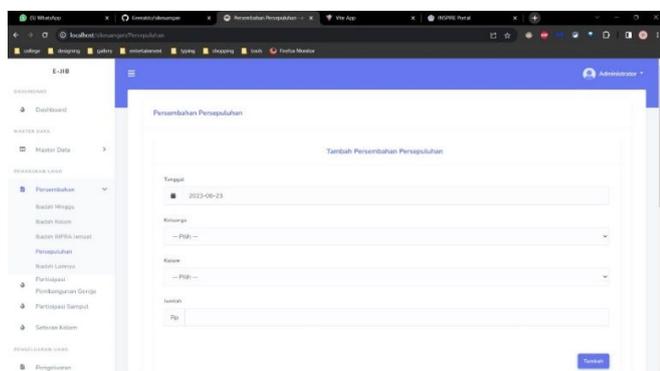
Gambar 16. Tampilan Halaman Tabel Hasil Penginputan Data



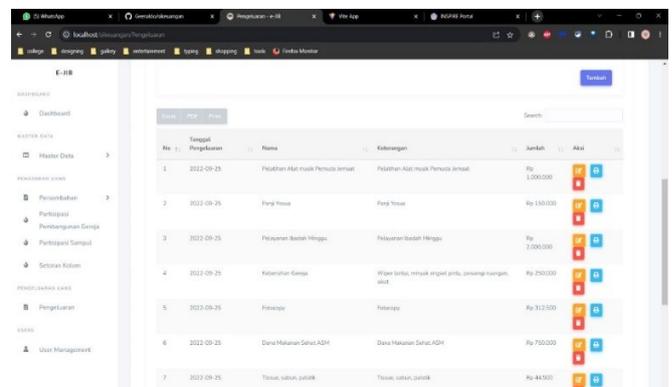
Gambar 14. Tampilan Halaman Utama E-JIB



Gambar 17. Tampilan Halaman Penginputan Data Pengeluaran



Gambar 15. Tampilan Halaman Penginputan Data Pemasukan



Gambar 18. Tampilan Halaman Tabel Hasil Penginputan Data

D. Testing

1) Hasil Uji Black Box

Pengujian *Black Box* dilakukan untuk melihat kinerja sistem yang dibuat apakah sudah sesuai dengan tujuan awal perancangan. *Black box* testing adalah pengujian yang mendefinisikan berbagai kondisi *input* yang dilakukan pada sistem untuk mendapatkan suatu *output* yang dapat dinilai. Pengujian dilakukan berulang kali untuk mendapatkan hasil yang akurat dan dapat dievaluasi (lihat Tabel I).

E. Deployment

Setelah melakukan pengujian *black box*, penulis melakukan uji coba langsung aplikasi bersama *user*. Pengujian ini bertujuan untuk mendapatkan hasil data yang dapat dijadikan sebagai acuan tingkat keberhasilan aplikasi yang dikembangkan untuk dapat diterima oleh pengguna. Penulis mengumpulkan sebanyak 20 orang untuk melakukan uji coba aplikasi.

Berikut ini adalah keterangan pertanyaan-pertanyaan yang diberikan kepada *user*:

P1 : Apakah tampilan dari desain *dashboard* menarik?

P2 : apakah *user* memahami setiap fitur dengan mudah?

P3 : apakah menu pada *dashboard* berjalan dengan baik?

P4 : apakah terdapat *error* saat menjalankan fitur di *dashboard*?

P5 : apakah proses *loading* data sampai ditampilkan di *dashboard* terasa cepat?

TABEL I
HASIL UJI BLACK BOX

No.	Pengujian	Input	Hasil
1.	Login dengan <i>password</i> salah	Memasukkan <i>password</i> salah	Sesuai
2.	Login dengan <i>password</i> kosong	Tidak memasukkan <i>password</i>	Sesuai
3.	Login dengan <i>username</i> salah	Memasukkan <i>username</i> salah	Sesuai
4.	Login dengan <i>username</i> kosong	Tidak memasukkan <i>username</i>	Sesuai
5.	Login dengan user name dan <i>password</i> benar	Memasukkan <i>username</i> dan <i>password</i> yang benar	Sesuai
6.	Admin menambahkan data	menambahkan data	Sesuai
7.	Admin menghapus data	Menghapus data	Sesuai
8.	Admin mengedit data	Mengubah data	Sesuai
9.	User meng – click filter Jemaat Imanuel Bahu	Klik filter Jemaat Imanuel Bahu	Sesuai
10.	User meng – click filter Jemaat Pniel Bahu	Klik filter Jemaat Pniel Bahu	Sesuai
11.	User meng – click filter Jemaat Musafir Kleak	Klik filter Jemaat Musafir Kleak	Sesuai
12.	Admin akan melakukan logout	Klik logout	Sesuai

Hasil jawaban pertanyaan kepada *user* dinilai dengan 5 kategori, yaitu SS (Sangat Setuju), S (Setuju), N (Netral), TS (Tidak Setuju) dan STS (Sangat Tidak Setuju). (lihat Tabel II).

Data hasil wawancara dihitung dengan menggunakan rumus persamaan di bawah ini.

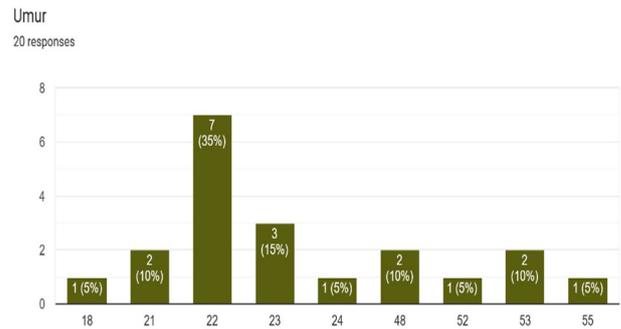
$$\text{Skor tertinggi (Smax)} = 4 \times n = 4n \text{ (SS)}$$

$$\text{Skor terendah (Smin)} = 0 \times n = 0 \text{ (STS)}$$

$$\text{Skor (S)} = \Sigma(\text{Jumlah Responden} \times \text{Bobot Jawaban})$$

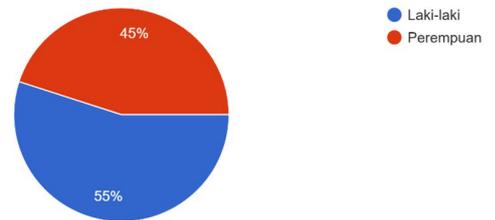
$$\text{Presentase interpretasi } p = \frac{S}{S_{max}} \times 100\%$$

(1)



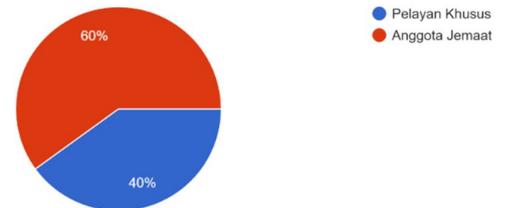
Gambar 19. Grafik Umur Responden

Jenis Kelamin
20 responses



Gambar 20. Grafik Jenis Kelamin Responden

Status/Posisi
20 responses



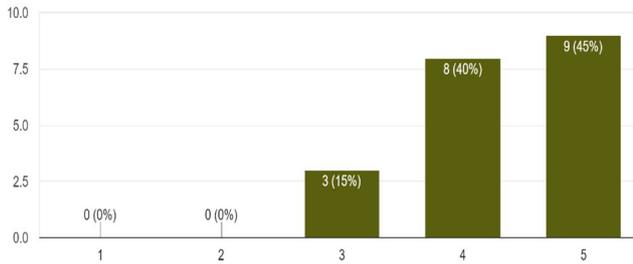
Gambar 21. Grafik Status/Posisi Responden

TABEL II
HASIL TES WAWANCARA

Pertanyaan	Total Nilai	Presentase
P1	66	82.5%
P2	71	88.75%
P3	71	88.75%
P4	65	81.25%
P5	68	85%

Apakah tampilan dari desain dashboard menarik?

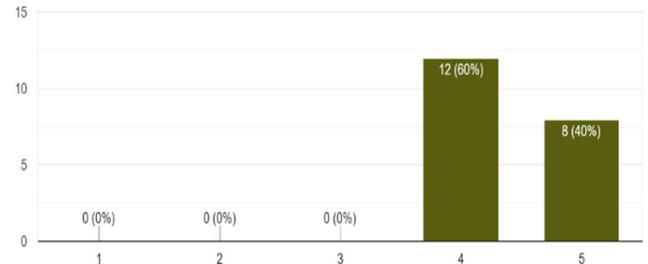
20 responses



Gambar 22. Hasil Jawaban P1

apakah proses loading data sampai ditampilkan di dashboard terasa cepat?

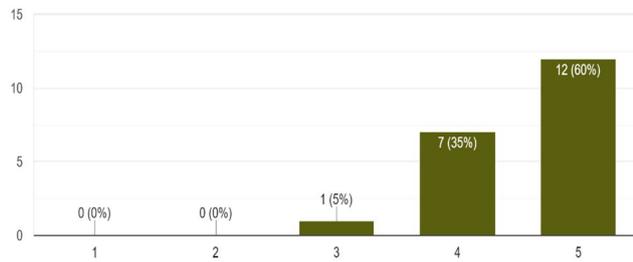
20 responses



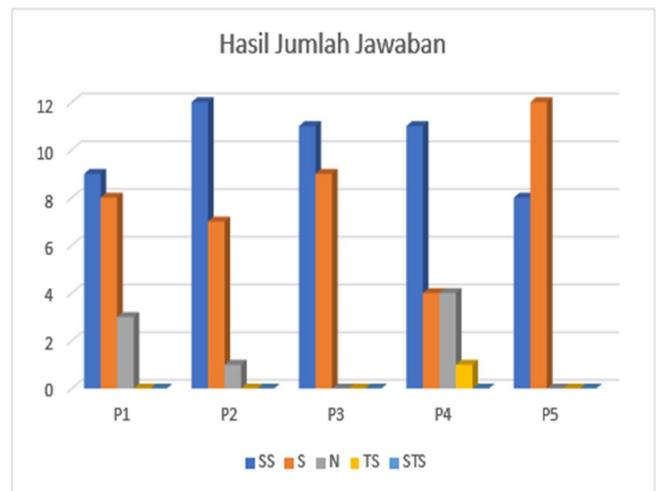
Gambar 26. Hasil Jawaban P5

apakah user memahami setiap fitur dengan mudah?

20 responses



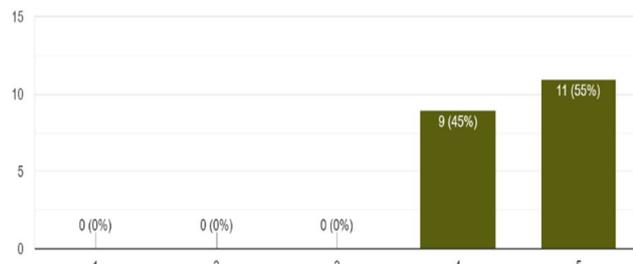
Gambar 23. Hasil Jawaban P2



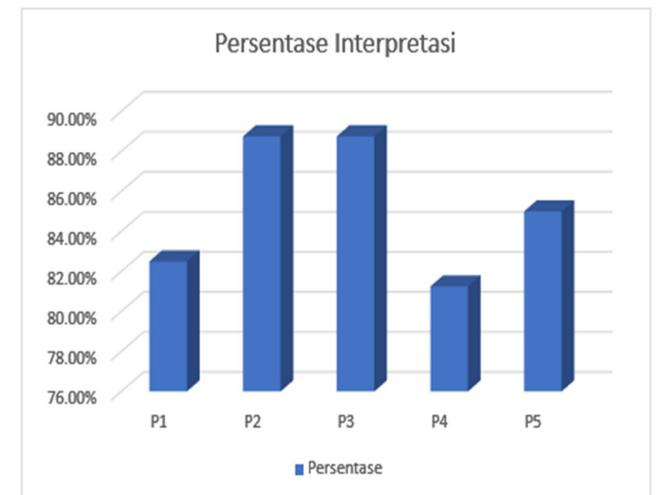
Gambar 27. Grafik Jumlah Jawaban

apakah menu pada dashboard berjalan dengan baik ?

20 responses



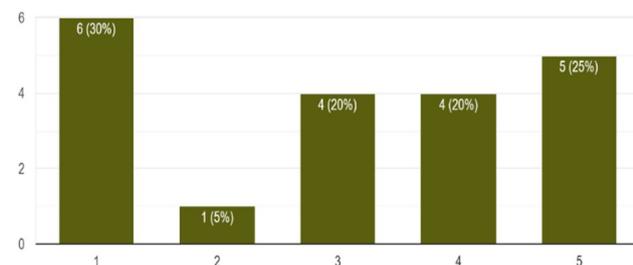
Gambar 24. Hasil Jawaban P3



Gambar 28. Grafik Presentase Interpretasi

apakah terdapat error saat menjalankan fitur di dashboard?

20 responses



Gambar 25. Hasil Jawaban P4

$$S_{max} = 4 \times 20 = 80$$

$$S_{min} = 0 \times 20 = 0$$

(2)

Wawancara diberikan kepada 20 orang *user* yang melakukan uji coba pada tahap ini. Hasil data jawaban dapat disimpulkan pada Tabel II.

Tingkat *Presentase Interpretasi* (P) diukur dengan skala interval berikut ini:

- Nilai 0% - 20% = Sangat Kurang
- Nilai 21% - 40% = Kurang
- Nilai 41% - 60% = Cukup
- Nilai 61% - 80% = Baik
- Nilai 81% - 100% = Sangat Baik

F. Review

Berdasarkan hasil data wawancara kepada *user* dapat disimpulkan bahwa tampilan desain *Dashboard* cukup menarik, penggunaannya cukup mudah, menu di dalamnya berfungsi sesuai dengan harapan, tidak terdapat *error* saat menjalankan aplikasi dan penggunaan *website* dianggap bermanfaat bagi *user*.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian ini yaitu Desain *Dashboard* Keuangan dan Informasi GMIM Wilayah Manado Barat Daya, maka dapat disimpulkan sebagai bahwa *Dashboard* Monitoring Wilayah Manado Barat Daya berhasil dibangun dengan menggunakan bahasa pemrograman JavaScript. Tampilan antar muka aplikasi dapat menjadi lebih menarik dengan menggunakan library ReactJS untuk *Front-end* sedangkan untuk *Back-end* menggunakan NodeJS. Berdasarkan pengujian yang dilakukan, *Dashboard* Monitoring sudah dapat berjalan dengan baik namun memerlukan perbaikan pada bagian fitur lainnya agar menjadi lebih optimal. *Dashboard* Monitoring dapat Menpermudah untuk mengetahui proses pemasukan dan pengeluaran setiap jemaat.

Adapun saran yang dapat diberikan yaitu mengintegrasikan sistem informasi keuangan berupa data pemasukan dan pengeluaran setiap jemaat memberikan kemudahan dalam melakukan pelaporan keuangan, tetapi juga harus mempertimbangkan aspek *security* apabila sudah masuk pada tahap menggunakan rekening perbankan. Disarankan agar dapat mengembangkan aspek *security* agar kinerja sistem lebih optimal. *Dashboard* Wilayah dapat dikembangkan hingga ke tingkat sinode GMIM dan juga ke rana komersial.

V. KUTIPAN

- [1] Abdul, K. (2003). Pengenalan sistem informasi. Penerbit Andi, Yogyakarta, 1, 1.
- [2] Sutabri, T. (2012). Konsep sistem informasi. Penerbit Andi.
- [3] Bourgeois, D. T., Smith, J. L., Wang, S., & Mortati, J. (2019). *Information systems for business and beyond*. Saylor Academy.
- [4] Connolly, T. M., & Begg, C. E. (2016). Database systems: a practical approach to design, implementation, and management. Sixth Edition. Pearson Education.
- [5] Elmasri, R., & Navathe, S. (2014). *Fundamentals of database systems* (Vol. 7). Pearson.
- [6] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Database system concepts.
- [7] Wijonarko, D., & Mulya, B. W. R. (2018). Pengembangan Antarmuka Pemrograman Aplikasi Menggunakan Metode RESTful pada Sistem Informasi Akademik Politeknik Kota Malang. *SMATIKA JURNAL*. vol. 8, no. 02, pp. 63–66, 2018, doi: 10.32664/smatika.v8i02.202.
- [8] Muri, M. F. A., Utomo, H. S., & Sayyidati, R. (2019). Search Engine Get Application Programming Interface. *Jurnal Sains dan Informatika*, 5(2),

- 88-97.
- [9] Pamungkas, C. A. (2017). *Pengantar dan Implementasi Basis Data*. Deepublish.
- [10] Arief, M. R. (2011). Pemrograman web dinamis menggunakan php dan mysql. *Yogyakarta: Andi*, 7-19.
- [11] Bekt, B. H. (2015). Mahir Membuat Website dengan Adobe Dreamweaver CS6, CSS dan JQuery. *Yogyakarta: Andi Offset*.
- [12] Daqiqil, I. (2011). *Framwork Codeigniter*. Pekanbaru: *Elex media komputindo*.
- [13] Abdulloh, R. (2018). *7 in 1 Pemrograman Web Untuk Pemula*. Elex Media Komputindo.
- [14] Siahaan, V., & Sianipar, R. H. (2020). *Buku Pintar JavaScript*. Balige Publishing.
- [15] Kumar, A., & Singh, R. K. (2016). Comparative analysis of angularjs and reactjs. *International Journal of Latest Trends in Engineering and Technology*, 7(4), 225-227.
- [16] Purnomo, H., & Makhunah, J. (2018). Sistem Informasi Pengolahan Data Keuangan Berbasis Web. *JIMP (Jurnal Informatika Merdeka Pasuruan)*, 3(3).
- [17] Lukow, E. D., Najoran, X. B., & Sentinuwo, S. R. (2022). Rancang Bangun Aplikasi Pengelola Keuangan Gereja Berbasis Android: Design and Development of Android-Based Church Financial Management Application. *Jurnal Teknik Informatika*, 17(4), 307-314.
- [18] Marentek, B., Lumenta, A. S., & Lantang, O. A. (2017). Rancang Bangun Web Service Sistem Informasi Keuangan GMIM Wilayah Tomohon 3. *Jurnal Teknik Informatika*, 12(1).
- [19] Yanuardi, Y., & Permana, A. A. (2019). Rancang Bangun Sistem Informasi Keuangan Pada Pt. Secret Discoveries Travel and Leisure Berbasis Web. *JIKA (Jurnal Informatika)*, 2(2).
- [20] Riyadli, H., Arliyana, A., & Saputra, F. E. (2020). Rancang Bangun Sistem Informasi Keuangan Berbasis WEB. *Jurnal Sains Komputer Dan Teknologi Informasi*, 3(1), 98-103.
- [21] Zaen, M. T. A., Julkarnaen, J., & Saleh, M. (2018). Sistem Informasi Keuangan Pada Dinas Perhubungan Kabupaten Lombok Tengah Berbasis Web. *Jurnal Manajemen Informatika Dan Sistem Informasi*, 2(1), 50-56.
- [22] Margaretha, H. A., & Nababan, M. N. (2020). Perancangan Sistem Informasi Manajemen Keuangan Berbasis Web Studi Kasus PT. Karya Swadaya Abadi. *Jurnal Sains dan Teknologi*, 1(2), 24-31.
- [23] Arsana, I. N. A., & Lestari, A. S. (2021). Rancang Bangun Sistem Informasi Laporan Keuangan Pada SMP Nasional Berbasis Web. *Jurnal Krisnadana*, 1(1), 47-56.

TENTANG PENULIS



Penulis bernama lengkap **Teofilus Gratiano Victor Pangemanan** lahir di Manado, 17 Agustus 2000. Merupakan anak pertama dari dua bersaudara dari pasangan Alm. Daniel H. Pangemanan (Ayah) dan Agnest M. Wullur M.Th (Ibu). Penulis memulai Pendidikan di TK GMIM Kasih Ibu Pinaras (2003-2005), kemudian melanjutkan pendidikan sekolah dasar di SD GMIM Pinaras (2005-2008), SD Negeri Kolongan (2008-2011), selanjutnya menempuh pendidikan di SMP Negeri 1 Kalawat (2011-2014), melanjutkan pendidikan di SMA Manado Independent School di Kolongan (2014-2017). Pada tahun 2018 penulis melanjutkan pendidikan tingkat sarjana (S1) di salah satu perguruan tinggi negeri di Sulawesi Utara, yaitu Universitas Sam Ratulangi Manado, dengan mengambil Jurusan Teknik Elektro, Program Studi Informatika.