

Perancangan Sistem *E-reporting* Menggunakan ReactJS dan Firebase

<http://dx.doi.org/10.28932/jutisi.v7i1.3098>

Riwayat Artikel

Received: 1 November 2020 | Final Revision: 9 Maret 2021 | Accepted: 12 Maret 2021

Jeremy Panjaitan^{✉#1}, Andrew Fernando Pakpahan^{*2}

[#]Teknik Informatika, Universitas Advent Indonesia
Jl. Kolonel Masturi No.288, Kabupaten Bandung Barat, Jawa Barat

¹1881027@unai.edu

²andrew@unai.edu

Abstract — The head office of west region of Seventh Day Adventist Indonesia in Indonesia is located in Jakarta. Every month employees must report every transaction they have made. The report can be in the form of proof of purchase, shopping receipt, notes and other proof of transactions. The reporting system of the office is done by submitting a hardcopy proof of the transaction to the finance department. The reporting process takes a long time and allows for inaccurate proof of transactions. Another problem that is often found is the unknown status of reports. Employees often do not know whether the reports that have been received by the financial department have been approved or not. The purpose of this study is to design an E-reporting system that will make it easier for employees to report to the financial department so that reporting can be done quickly and accurately. The system to be built uses ReactJS and Firebase using an Agile Software development method. The result of this research is the creation of an E-reporting application that can help to report on the financial department every month.

Keywords— E-reporting, Firebase, ReactJS, Web based information system

I. PENDAHULUAN

Sistem E-reporting merupakan sebuah sistem yang menggunakan jaringan komputer untuk melakukan proses pelaporan informasi secara elektronik [1]. Dengan adanya sistem E-reporting maka dapat membantu melakukan monitoring terhadap laporan baik oleh penerima maupun pelapor. Sistem ini dapat mengatasi sistem pelaporan secara manual yang memakan banyak waktu, tenaga dan juga biaya sehingga dapat menghasilkan alur kerja dan transmisi informasi yang baik [2]–[4]. Oleh karena itu Sistem E-reporting menjadi sesuatu yang dibutuhkan oleh organisasi yang efisien dalam sistem pelaporannya.

Gereja Masehi Advent Hari Ketujuh (GMAHK) adalah sebuah kelompok atau organisasi orang-orang kristen yang menganut aliran penginjilan atau *evangelical*. GMAHK merupakan sebuah organisasi yang bertujuan untuk

membantu tugas, visi dan misi dalam pelayanan GMAHK yaitu untuk memberitakan injil yang merupakan aliran yang dianut yaitu aliran *evangelical* [5].

Kantor pusat GMAHK memiliki 15 departemen dimana tiap-tiap departemen memiliki tanggung jawab masing-masing. Berdasarkan hasil wawancara dengan salah satu pegawai kantor pusat GMAHK yang bertanggung jawab pada salah satu departemen, disebutkan bahwa setiap bulannya pegawai pada setiap departemen akan melakukan laporan terhadap transaksi yang terkait dengan pekerjaan pelayanan yang mereka lakukan; misalnya: bukti pembayaran, kwitansi, dsb. Sistem pelaporan yang dilakukan masih secara manual dimana setiap pegawai mengisi formulir laporan yang telah diberikan oleh bagian keuangan dengan menggunakan Microsoft Excel, serta harus melampirkan setiap bukti-bukti transaksi dengan menempelkan pada kertas menggunakan lem. Tentu sistem ini sangat tidak efisien, karena memungkinkan resiko yang sangat besar akan terjadi bukti-bukti transaksi yang disimpan hilang karena tercecer. Sistem pelaporan seperti ini akan mengakibatkan laporan yang tidak akurat dan sulit untuk dipercaya yang mengakibatkan sulitnya pengambilan keputusan pada bagian keuangan. Permasalahan lain yang sering kali terjadi adalah pada saat pegawai mengumpulkan laporan pada bagian keuangan, pegawai tidak mengetahui status dari laporan tersebut. Sehingga pegawai tidak mengetahui apakah laporan telah diterima atau tidak. Pegawai juga tidak mengetahui sudah berapa banyak *budget* yang sudah terpakai.

Dari permasalahan di atas, mengusulkan sebuah sistem E-reporting berbasis web yang dapat diakses oleh perangkat apa saja yang terhubung pada internet sehingga memudahkan setiap pegawai yang ada di kantor pusat GMAHK untuk melakukan pelaporan terhadap setiap transaksi yang mereka lakukan setiap bulannya dan memudahkan bagian keuangan untuk mengambil keputusan

terhadap laporan yang telah diterima. Adapun kegunaan dari sistem yang diharapkan dari penelitian ini adalah: 1) Mengurangi resiko bukti-bukti transaksi tercecer, karena dapat difoto lalu dilampirkan dengan laporan dengan cara mengupload foto tersebut. 2) Memberikan informasi kepada pegawai dan bagian keuangan status dari laporan. 3) Membatasi pengeluaran dari setiap pegawai. 4) Memberikan informasi kepada setiap pegawai dan bagian keuangan jumlah anggaran yang telah terpakai. 5) Meningkatkan monitoring bagian keuangan terhadap laporan dari setiap pegawai.

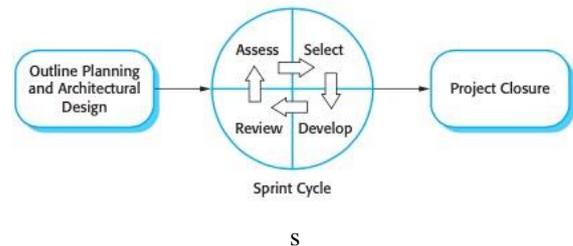
II. METODE PENELITIAN

Seiring berkembangnya zaman, kebutuhan manusia semakin meningkat dan berubah dengan sangat cepat. Dalam mengembangkan sebuah sistem, sulit sekali untuk memprediksi hal-hal yang mungkin terjadi selama proses pengembangan salah satunya yaitu adanya perubahan atau penambahan kebutuhan sistem. Kegagalan yang muncul dalam adaptasi merupakan salah satu penyebab kegagalan pengembangan sistem, oleh karena itu diperlukan sebuah metode pengembangan yang dapat menangani hal-hal tersebut. Salah satu metode yang dapat menangani perubahan adalah metode pengembangan *Agile* dengan model *scrum* [6] dan metode ini dijadikan sebagai metode penelitian.

Dari segi filosofinya sendiri, *agile* memiliki arti diantaranya: mendorong demi terciptanya kepuasan pelanggan; meningkatkan *delivery* perangkat lunak secara bertahap (*incremental*); tim proyek yang ramping dan memiliki motivasi yang sangat tinggi; minimasi pekerjaan; serta menyederhanakan (birokrasi) keseluruhan proses dalam membangun perangkat lunak [7]. Konsep *Agile Software Development* diperkenalkan secara umum pada tahun 2001 oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa *agile Software development* merupakan sebuah metode membangun software dengan melakukannya dan membantu orang lain membangunnya sekaligus [8]. Sementara menurut [9], metode *agile* merupakan pengembangan *incremental* yang berfokus pada pengembangan secara cepat, perangkat lunak yang dirilis secara bertahap, mengurangi overhead proses, menghasilkan kode dengan kualitas yang tinggi dan melibatkan pelanggannya secara langsung dalam proses pengembangannya.

Terdapat beberapa model dalam metode *agile software development*, yaitu 1) *Extreme Programming*, 2) *Adaptive Software Development*, 3) *Dynamic Systems Development Method*, 4) *Model Scrum*, dan 5) *Agile Modeling* [10]. Dalam penelitian ini, memilih menggunakan model *scrum*. *Scrum* merupakan *software development method* yang dilakukan oleh Jeff Sutherland bersama rekan nya pada awal tahun 1990 an. Prinsip yang ada pada model *scrum* sejalan dengan prinsip-prinsip yang terdapat pada metode pengembangan secara cepat yang digunakan dalam menuntun kegiatan pengembangan perangkat lunak,

kebutuhan, analisa, dan pengiriman (*delivery*). Rangkaian kegiatan dalam model *scrum* diantaranya 1) Aktivitas backlog, 2) aktivitas sprints, 3) aktivitas Scrum Meeting dan 4) Demo [12]. Rangkaian kegiatan dalam model *scrum* digambarkan pada Gambar 1.



Gambar 1. Proses scrum [9]

III. PERANCANGAN SISTEM

Dalam membangun sistem E-reporting berbasis web, penelitian ini menggunakan dua komponen utama yang mendukung sistem ini, yaitu Firebase dan ReactJS.

A. Firebase

Firebase merupakan Baas (*Backend as a Service*) yang dikembangkan oleh Google. Firebase mempermudah para *developer* perangkat lunak untuk mengembangkan aplikasinya [11]. Layanan firebase yang digunakan pada sistem yang dirancang antara lain *authentication* agar memungkinkan user untuk *login* dan terkoneksi secara langsung dengan server Firebase [12], *firebase storage* yang memungkinkan pengembang untuk mengunduh maupun mengunggah sebuah berkas [13] dan *firebase cloud function* sebagai framework tanpa server yang memungkinkan kode dijalankan pada *back-end* secara otomatis sebagai respons terhadap yang dipicu oleh fitur maupun permintaan *HTTPS* [14]. Firebase memiliki library (pustaka) yang lengkap dan dapat digabungkan dengan berbagai framework lain seperti NodeJS, Java, Javascript, ReactJS, dan lain-lain [15]. Firebase memberikan kemudahan dan mempercepat pengembangan dari sistem E-reporting ini.

B. ReactJS

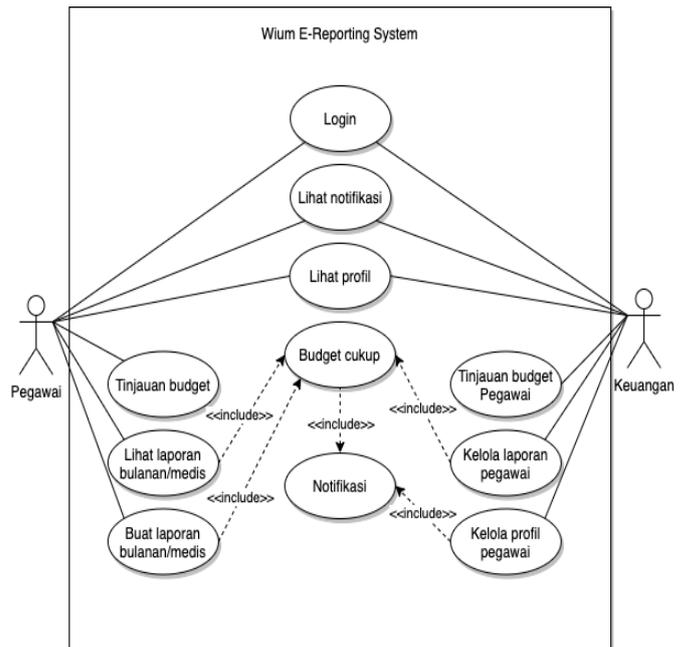
ReactJS adalah front-end library yang dikembangkan oleh Facebook. ReactJS digunakan sebagai pendukung dari web-framework [16]. ReactJS memiliki beberapa keunggulan diantaranya memberikan kecepatan, *simplicity*, dan *scalability*. ReactJS memungkinkan pengembang dapat membangun sebuah komponen UI yang lebih interaktif, *stateful*, & *reusable*. Di dalam kaidah MVC (*Model View Control*), ReactJS bertanggung jawab pada bagian *view* saja [17].

C. Unified Modeling Language

Dalam Perancangan dan pembuatan sistem ini, menggunakan UML (Unified Modeling Language). Unified Modeling language atau UML adalah salah satu “bahasa” ataupun notasi yang saat ini telah menjadi standar dokumentasi, perancangan serta pembangunan perangkat lunak [18]. Sistem yang dikembangkan ini memiliki 2 bagian utama sisi user, yaitu antarmuka pada sisi pegawai dan antarmuka pada sisi bagian keuangan. Antarmuka pada sisi pegawai merupakan penggunaan sistem untuk melakukan pelaporan terhadap transaksi yang telah dilakukan tiap bulannya. Sedangkan antarmuka pada sisi bagian keuangan merupakan penggunaan sistem untuk mengelola laporan pegawai dan mengelola pegawai.

D. Use Case Diagram

Use case diagram merupakan sebuah diagram yang berfungsi untuk memvisualisasikan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Gambar 2 adalah use case diagram dari penelitian ini. Terdapat 2 aktor atau pengguna yang akan menggunakan sistem E-reporting yaitu pegawai dan bagian keuangan. Aktor pegawai dan bagian keuangan memiliki 3 fungsi yang sama. Untuk menggunakan sistem maka mereka harus login terlebih dahulu. Mereka mampu melihat notifikasi. Notifikasi ini berisi pesan untuk memberitahu interaksi apa saja yang dilakukan antara keuangan dan pegawai sebagai contoh apabila pegawai telah membuat laporan bulanan/medis maka sistem akan menyimpan notifikasi ini untuk memberitahu kepada bagian keuangan bahwa ada laporan yang baru saja dibuat. Aktor pegawai dapat melihat tinjauan *budget* mereka. Pegawai mampu melihat maupun membuat laporan bulanan maupun medis. Namun, disaat mereka hendak menyimpan laporan tersebut sistem akan mengkalkulasi apakah *budget* yang tersisa cukup. Aktor bagian keuangan mampu untuk melihat tinjauan *budget* seluruh pegawai. Bagian keuangan memiliki kemampuan untuk mengelola laporan dari pegawai dalam hal ini bagian keuangan mampu untuk menolak atau menyetujui atau menghapus laporan yang telah diserahkan. Selain itu, keuangan juga mampu untuk mengelola profil pegawai dalam hal ini keuangan dapat membuat atau menghapus atau mengubah profil dari pegawai.

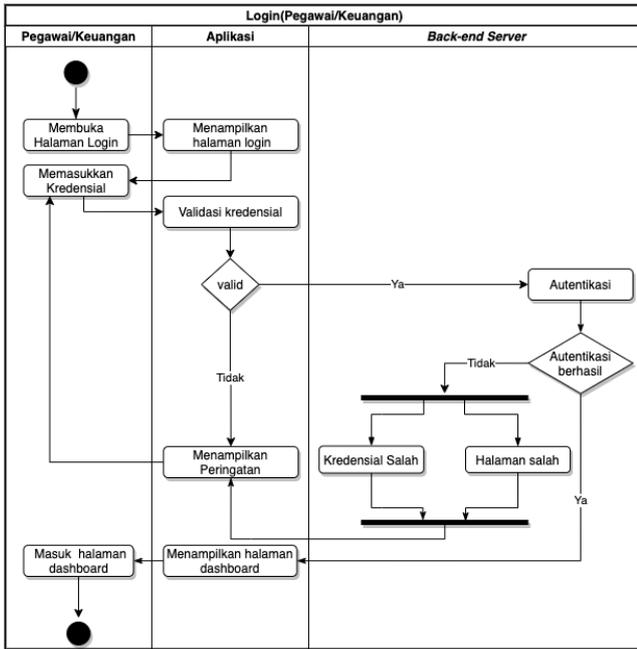


Gambar 2. Use case diagram

E. Activity Diagram

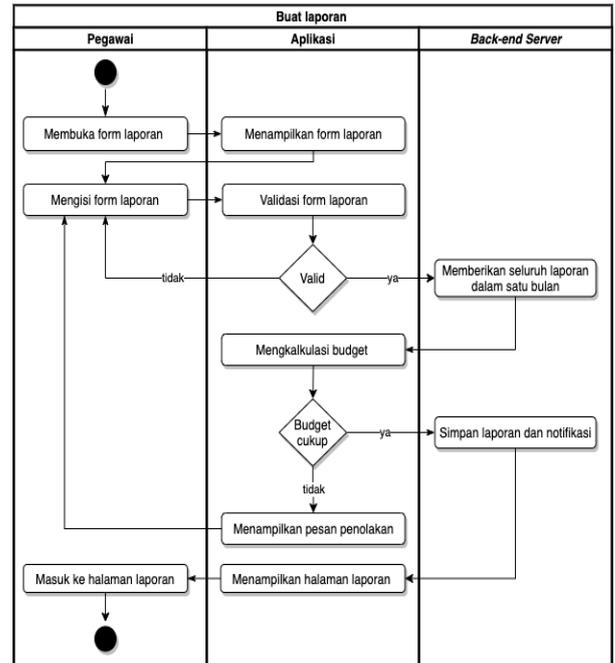
Activity Diagram merupakan diagram yang menunjukkan aktivitas sistem berupa bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai hingga berakhir [19]. Pada sistem ini terdapat beberapa activity diagram yaitu *login*, *buat laporan*, *menyetujui laporan*, dan *buat pegawai*

Gambar 3 menjelaskan hal yang harus dilakukan pertama-tama sebelum pegawai atau bagian keuangan hendak masuk ke dalam sistem yaitu *login*. Untuk *login* ke dalam sistem, user diminta untuk memasukkan kredensial yakni email dan password pada input form yang telah disediakan. Setelah user memasukkan kredensial, sistem akan melakukan validasi apakah kredensial yang telah diinput sudah valid. Apakah kredensial sudah valid, sistem akan mengirimkannya kepada *back-end*. *Back-end* akan melakukan autentikasi terhadap kredensial user. Apabila autentikasi berhasil maka sistem akan menampilkan halaman *dashboard* user. Apabila autentikasi tidak berhasil maka sistem akan menampilkan pesan *error* dimana terdapat 2 kemungkinan pesan yaitu user memasukkan kredensial yang salah atau user masuk pada halaman yang salah.



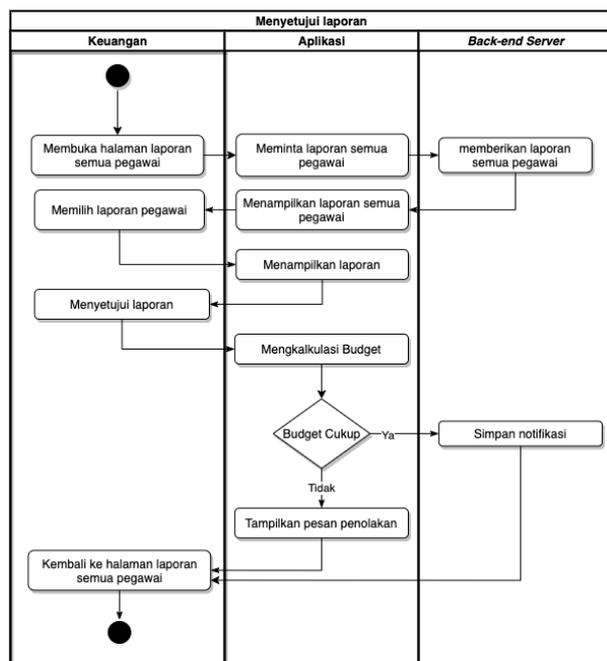
Gambar 3 Activity diagram login

Gambar 4 menjelaskan gambaran bagaimana seorang pegawai yang akan membuat laporan. Untuk membuat laporan, pegawai akan membuka form laporan. Setelah aplikasi menampilkan form laporan, pegawai akan mengisi form tersebut. Form tersebut akan divalidasi oleh aplikasi sebelum dikirimkan ke *back-end*. Pegawai diminta untuk mengisi ulang laporan apabila form tidak valid. Namun apabila valid, *back-end* akan memberikan seluruh laporan dalam jangka waktu satu bulan kepada aplikasi. Lalu aplikasi akan mengkalkulasi *budget*. Apabila *budget* tidak mencukupi, aplikasi menolak pembuatan laporan. Apabila *budget* mencukupi, maka laporan dan notifikasi dibuatnya laporan tersebut akan disimpan pada *back-end*. Apabila semua rangkaian aktivitas berjalan dengan baik, maka aplikasi akan membawa pegawai ke halaman laporan, dimana pada halaman ini pegawai akan melihat laporan yang telah mereka buat dalam jangka waktu satu bulan.

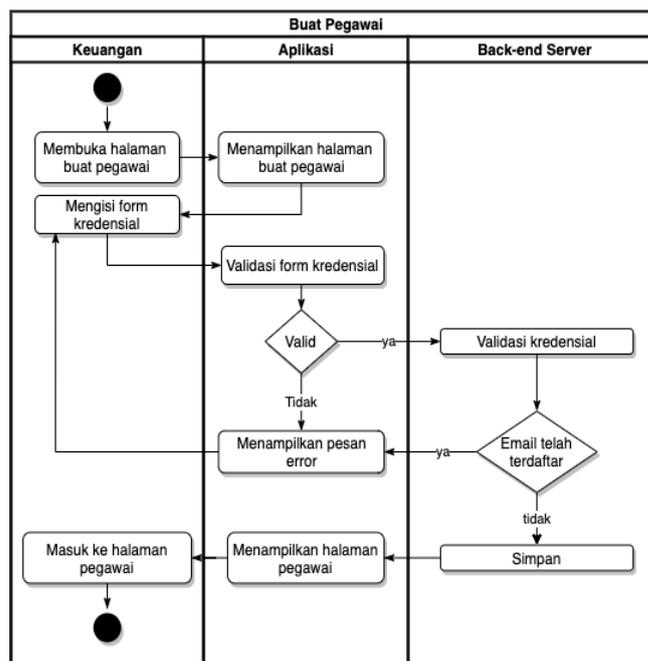


Gambar 4. Activity diagram membuat laporan

Gambar 5 menjelaskan rangkaian aktivitas yang terjadi saat bagian keuangan menyetujui laporan yang telah dibuat oleh pegawai. Pertama-tama user akan membuka halaman yang menampilkan semua laporan pegawai. Kemudian sistem akan meminta semua laporan dari *back-end* dan kemudian menampilkannya kepada user. User kemudian memilih laporan mana yang akan disetujui. Setelah laporan dipilih, sistem akan menampilkan laporan tersebut. Saat ini user dapat menyetujui laporan. Namun sebelum sistem memproses ke tahap selanjutnya, sistem akan mengkalkulasi apakah *budget* mencukupi. Apabila *budget* tidak mencukupi maka sistem akan menampilkan pesan penolakan. Namun apabila *budget* mencukupi, maka sistem akan menyimpan laporan disertai dengan notifikasi.



Gambar 5. Activity diagram menyetujui laporan



Gambar 6. Activity diagram buat pegawai

Gambar 6 menjelaskan gambaran user (bagian keuangan) saat membuat pegawai. Untuk membuat pegawai pertama-tama user akan membuka halaman buat pegawai. Setelah aplikasi menampilkan halaman buat pegawai, user akan mengisi kredensial pada form di halaman tersebut. Kemudian form tersebut akan divalidasi oleh aplikasi. Apabila form tidak valid, maka aplikasi meminta user untuk memperbaiki kredensial yang telah dimasukkan. Namun apabila form valid, aplikasi akan mengirimkannya kepada *back-end*. Sebelum *back-end* menyimpan kredensial tersebut sebagai pegawai baru, *back-end* memeriksa apakah email yang ada pada kredensial sudah terdaftar atau belum. Apabila sudah terdaftar, *back-end* akan menolak untuk membuat pegawai baru. Namun apabila email belum terdaftar, maka *back-end* akan menyimpan kredensial tersebut. Kemudian aplikasi akan membawa user ke halaman pegawai.

F. Perancangan Database Firebase

Firebase memiliki 2 layanan database yaitu *Cloud firestore* dan *Realtime database*. Sistem yang dirancang pada penelitian ini menggunakan layanan *Cloud firestore* oleh karena layanan tersebut merupakan layanan terbaru yang melanjutkan keberhasilan dari layanan database *Realtime database*. *Cloud firestore* juga memiliki fitur kueri yang lebih lengkap dan lebih cepat, serta penskalaan yang lebih mendalam dibandingkan dengan *Realtime database* [20]. *Cloud firestore* merupakan sebuah penyimpanan yang berbasis data non-SQL. Ada beberapa tipe data yang dapat disimpan pada database ini, antara lain string, number, boolean, map, array, null, timestamp, geopoint, dan reference. Namun pada sistem ini, hanya digunakan 3 tipe data yaitu string, number, dan timestamp. Berbeda dengan basis data SQL, database basis non-SQL tidak memiliki tabel dan baris [13]. *Cloud firestore* menyimpan data pada sebuah dokumen yang berisi pemetaan *field* dengan nilai. Kemudian dokumen tersebut akan disimpan pada sebuah *collection*. Melalui *collection* ini kita dapat mengatur data dan membangun kueri. Sistem yang dibangun memiliki 4 *collection* diantaranya, *users* untuk menyimpan data user, *monthlyexpense* untuk menyimpan data laporan bulanan, *medicalexpense* untuk menyimpan data laporan medis, dan *notifications* untuk menyimpan data notifikasi. Setiap tabel yang disampaikan pada penjelasan berikut ini merupakan *field* yang ada pada setiap *collection* beserta tipe datanya.

Pada Tabel I dijelaskan bahwa setiap dokumen user yang tersimpan pada *collection* ini memiliki beberapa *field* dan tipe data. Diantaranya adalah *id*, *address*, *email*, *name*, *phone*, *phoneUrl*, *role*, *totalMedicalBudget*, dan

totalMonthlyBudget. Field id memiliki tipe data string yang digunakan untuk menyimpan id dari dokumen, address memiliki tipe data string yang digunakan untuk menyimpan alamat dari user, email memiliki tipe data string yang digunakan untuk menyimpan email dari user, name memiliki tipe data string yang digunakan untuk menyimpan nama dari user, phone memiliki tipe data string yang digunakan untuk menyimpan nomor telfon user, photoUrl yang memiliki tipe data string digunakan untuk menyimpan url foto user, role yang memiliki tipe data string digunakan untuk menyimpan jabatan pegawai., totalMedicalBudget yang memiliki tipe data number digunakan untuk menyimpan total maksimal pengeluaran user yang berhubungan dengan medis, dan totalMonthlyBudget yang memiliki tipe data number digunakan untuk menyimpan total maksimal pengeluaran user berhubungan dengan laporan bulanan.

TABEL I
USERS COLLECTION

Nama Field	Tipe Field
id	string
address	string
email	string
name	string
phone	string
photoUrl	string
role	string
totalMedicalBudget	number
totalMonthlyBudget	number

Pada Tabel II dijelaskan bahwa medicalexpenditure dan monthlyexpense memiliki field dan tipe data yang sama. Diantaranya adalah amount, createdAt, currency, date, expense, owner, receipt, dan type. Amount digunakan untuk menyimpan jumlah pengeluaran untuk laporan tersebut, createdAt digunakan untuk menyimpan waktu dibuatnya laporan tersebut, currency digunakan untuk menyimpan mata uang yang digunakan. Sistem ini masih mendukung 2 mata uang, yaitu rupiah dan dollar. Date digunakan untuk menyimpan tanggal transaksi pada laporan tersebut dilakukan. Expense digunakan untuk menyimpan deskripsi dari laporan. Owner digunakan untuk menyimpan uid dari user yang membuat laporan. Receipt digunakan untuk menyimpan url foto dari receipt, dimana url ini mengacu pada foto receipt yang disimpan pada *firebase storage*. Approver digunakan untuk menyimpan nama finance yang menyetujui atau menolak laporan tersebut. Type digunakan untuk menyimpan tipe laporan. Untuk medicalexpenditure dibagi menjadi 3 tipe yaitu dental, regular, dan hospitalization expense sementara untuk monthlyexpense dibagi menjadi 2 tipe yaitu travel dan other expense.

TABEL II
MEDICAL DAN MONTHLY EXPENSE COLLECTION

Nama Field	Tipe Field
amount	number
createdAt	timestamp
currency	string
date	string
expense	string
owner	string
receipt	array
approver	string
type	string

Pada Tabel III dijelaskan bahwa *notifications collection* memiliki beberapa field dan tipe datanya. Diantaranya adalah createdAt, expenseId, from, message, to, dan type. createdAt digunakan untuk menyimpan tanggal dibuatnya notifikasi, expenseId digunakan untuk menyimpan id dari laporan yang terkait, from digunakan untuk menyimpan user yang melakukan aksi. message digunakan untuk menyimpan isi pesan dari notifikasi, to digunakan untuk menyimpan penerima notifikasi, dan type digunakan untuk menyimpan tipe notifikasi.

TABEL III
NOTIFICATIONS COLLECTION

Nama Field	Tipe Field
createdAt	timestamp
expenseId	string
from	string
message	string
to	string
type	string

G. Perancangan Desain Antar Muka

Rancangan desain sebuah sistem dibuat agar bertujuan sebagai acuan dalam perancangan sistem. Perancangan interface sistem E-reporting ini menggunakan aplikasi *Balsamiq Mockups 3*. *Balsamiq Mockups 3* dipilih dalam perancangan sistem ini karena aplikasi ini berbasis cloud disertai dengan aplikasi desktop sehingga pengguna dapat dengan mudah dan cepat membuka rancangan tampilan interface pengguna sistem E-reporting berbasis web [21]. Dalam merancang sebuah tampilan antarmuka, penelitian ini perlu memperhatikan kebutuhan dari pengguna sistem ini. Oleh karena itu, dalam merancang tampilan antarmuka, penelitian ini menggunakan metode *user centered design*. *User centered design* adalah sebuah proses desain interface dimana hanya berfokus terhadap tujuan kegunaan, karakteristik pengguna, lingkungan, tugas, dan alur kerja di dalam desainnya [22]. Adapun prinsip-prinsip yang perlu diperhatikan dalam UCD adalah [23]:

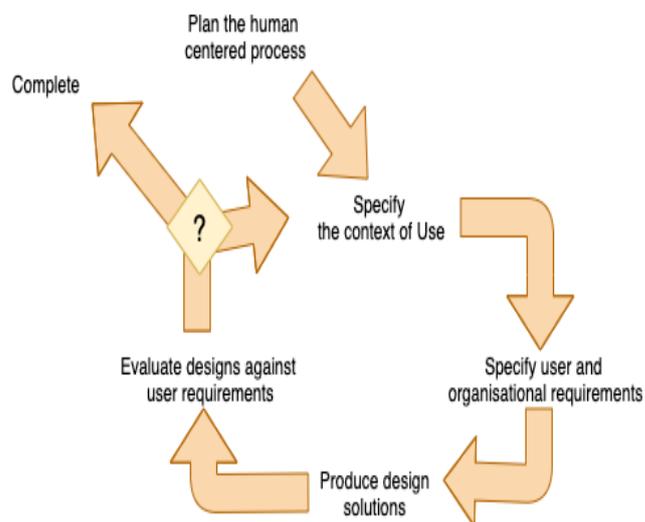
- *Fokus pada pengguna:* Perancangan harus dilakukan dengan hubungan langsung terhadap pengguna sesungguhnya dengan melalui interview, survey, dan partisipasi dalam workshop perancangan. Hal ini dilakukan agar dapat memahami kognisi, karakter, dan sikap pengguna serta karakteristik anthropometric. Aktivitas utamanya yaitu mencakup pengambilan data, analisis dan integrasinya ke dalam informasi perancangan dari pengguna tentang karakteristik tugas, lingkungan teknis dan organisasi. *Perancangan terintegrasi:* Perancangan mencakup antarmuka pengguna, sistem bantuan, dukungan teknis serta prosedur instalasi dan konfigurasi
- *Pengujian pengguna:* Pendekatan yang dilakukan yaitu observasi tentang kelakuan pengguna, evaluasi umpan-balik yang cermat, wawasan pemecahan terhadap masalah yang ada, dan motivasi yang kuat untuk mengubah rancangan.
- *Perancangan interaktif:* Sistem harus didefinisikan, dirancang, dan di test secara berulang-ulang. Berdasarkan hasil test tersebut maka didapat fungsi, antarmuka, sistem bantuan, dokumentasi pengguna, dan pendekatan pelatihannya.

Terdapat 4 langkah yang dilakukan secara iterasi dalam UCD. Diantaranya adalah [24]:

- *Specify the context of use:* pada langkah ini dilakukan identifikasi terhadap orang yang akan menggunakan sistem. Akan dijelaskan untuk apa dan dalam kondisi seperti apa mereka akan menggunakan sistem ini. Berdasarkan hasil observasi dan wawancara dengan pegawai menunjukkan bahwa proses pelaporan masih dilakukan secara manual dimana pegawai akan mengisi form excel yang telah dibagikan oleh bagian keuangan dan untuk bukti transaksi di tempelkan pada kertas HVS menggunakan lem.
- *Specify user and organizational requirements:* pada langkah ini dilakukan identifikasi kebutuhan pengguna dan kebutuhan organisasi. Berdasarkan hasil wawancara dengan pegawai kantor menunjukkan bahwa nantinya yang akan menggunakan sistem ini adalah pegawai kantor yang nantinya akan membuat dan mengumpulkan laporan beserta bukti transaksi lalu pihak lain yang akan menggunakan sistem ini adalah bagian keuangan yang nantinya mereka akan mengelola laporan.
- *Produce design solutions:* pada langkah ini akan dimulai untuk membangun desain yang diharapkan sebagai solusi dari sistem yang sedang dianalisis. Adapun tampilan antarmuka pengguna yang dibuat diantaranya *login* pegawai dan keuangan, halaman *dashboard* pegawai dan keuangan, halaman laporan, halaman lihat laporan, halaman buat laporan, halaman profil, halaman notifikasi pegawai dan keuangan, halaman laporan seluruh pegawai, halaman laporan pegawai, halaman lihat laporan pegawai, halaman buat pegawai.

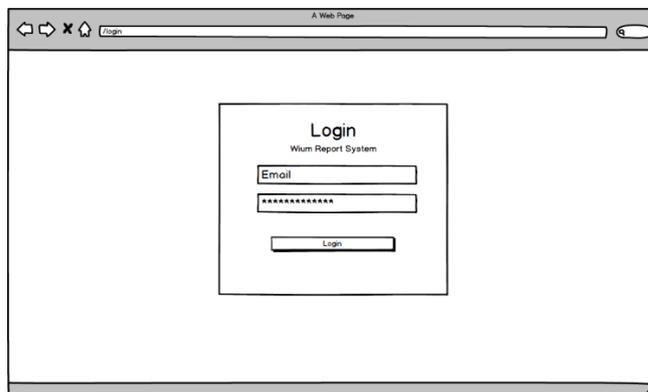
- *Evaluate Design:* pada langkah ini dilakukan evaluasi terhadap sistem yang telah didesain.

Gambar 7 menjelaskan langkah yang dilakukan dalam UCD.



Gambar 7. User centered design

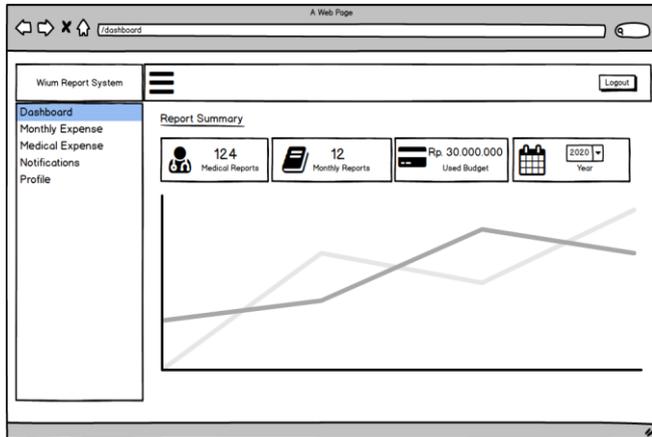
Gambar 8 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *login*. Halaman *login* ini berada pada sisi pegawai dan keuangan. Pada halaman ini terdapat 2 input diantaranya adalah input untuk memasukkan email dan input untuk memasukkan password. Selain itu terdapat tombol yang berfungsi untuk melakukan proses autentikasi.



Gambar 8. Halaman login

Gambar 9 menunjukkan hasil rancangan antarmuka pengguna untuk halaman *dashboard* pegawai. Halaman yang dirancang ini berada pada sisi pegawai yang bertujuan untuk memberikan tinjauan/ringkasan terhadap laporan dan jumlah *budget* yang telah terpakai dalam periode waktu 1 tahun. Pada bagian atas halaman ini terdapat 4 bagian diantaranya, 1) *Medical reports* yang memberikan informasi terhadap jumlah seluruh laporan medis yang telah disetujui

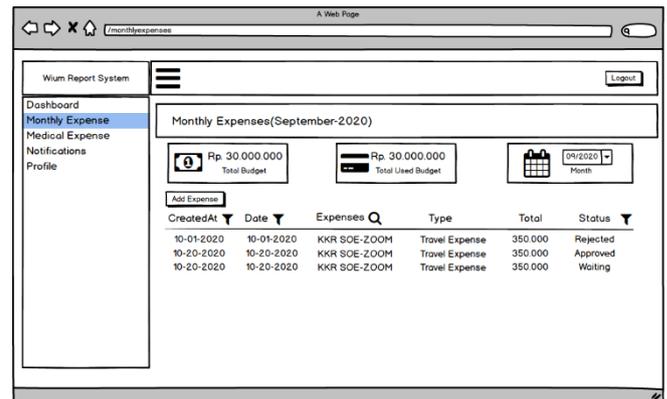
2) *Monthly reports* yang memberikan informasi terhadap seluruh jumlah laporan bulanan yang telah disetujui 3) *Used budget* memberikan informasi terhadap jumlah *budget* yang telah terpakai 4) *Year* memungkinkan pegawai untuk memilih tahun yang akan dilihat tinjauan atau ringkasan laporannya. Lalu pada bagian bawah halaman ini terdapat grafik yang memberikan gambaran terhadap penggunaan *budget* pegawai dalam periode waktu 1 tahun.



Gambar 9. Halaman *dashboard* pegawai

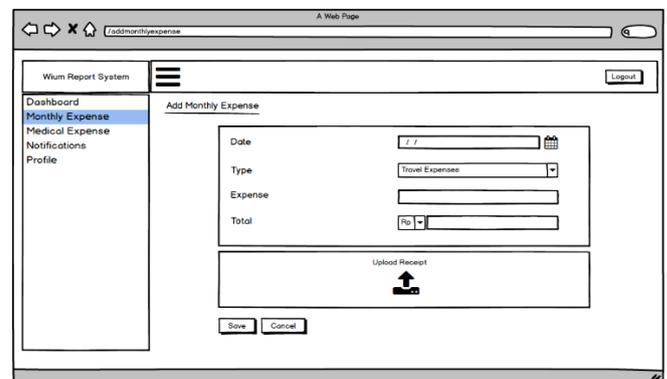
Gambar 10 menunjukkan hasil rancangan antarmuka pengguna untuk halaman laporan. Halaman yang dirancang ini berada pada sisi pegawai yang bertujuan untuk menampilkan laporan yang telah pegawai buat dalam jangka waktu satu bulan, baik itu laporan bulanan maupun laporan medis. Namun, pada gambar tersebut yang ditunjukkan hanya halaman laporan untuk laporan bulanan saja. Pada bagian atas halaman tersebut terdapat keterangan yang menunjukkan bahwa laporan yang sedang dibuka oleh pegawai adalah halaman laporan bulanan atau laporan medis. Lalu dibawah nya terdapat 3 bagian yang memberikan informasi diantaranya, 1) *Total budget* memberikan informasi total maksimal penggunaan *budget* bulanan 2) *Total used budget* memberikan informasi jumlah budget dalam jangka waktu satu bulan yang telah terpakai 3) *Month* memungkinkan pegawai untuk memilih bulan yang akan ditampilkan laporannya. Lalu pada bagian bawah dari halaman ini terdapat tabel yang berisi laporan pegawai dalam jangka waktu satu bulan. Tabel tersebut terdiri dari 6 kolom. Kolom-kolom tersebut memberikan deksripsi terhadap setiap laporan diantaranya 1) *Created At* memberikan informasi kapan dibuatnya atau dikumpulnya laporan tersebut 2) *Date* memberikan informasi kapan transaksi pada laporan tersebut terjadi 3) *Expense* memberikan deskripsi singkat terhadap laporan 4) *Type* memberikan tipe laporan tersebut. 5) *Total* memberikan informasi *budget* yang terpakai pada laporan tersebut 6) *Status* memberikan informasi apakah laporan tersebut disetujui atau ditolak oleh bagian keuangan. Pada halaman ini juga terdapat satu tombol yang berfungsi untuk

membawa pegawai ke halaman buat laporan. Pada halaman ini disediakan fungsi pencari dan penyaring.



Gambar 10. Halaman laporan

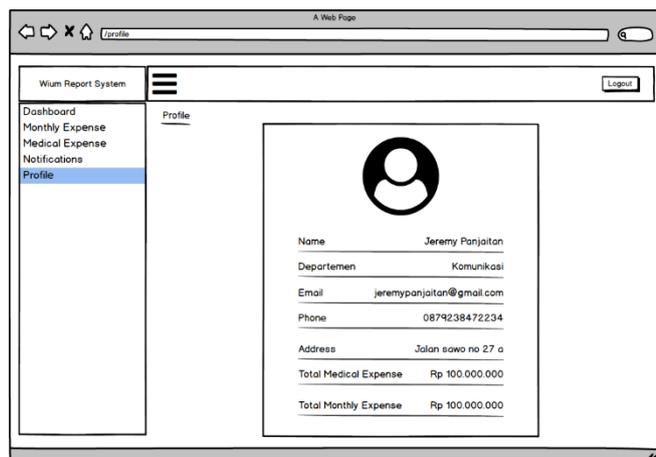
Gambar 11 menunjukkan hasil rancangan antarmuka pengguna untuk halaman buat laporan. Halaman yang dirancang ini berada pada sisi pegawai yang bertujuan untuk menampilkan form untuk buat laporan. Form yang ada pada halaman ini memiliki beberapa field diantaranya 1) field untuk memasukkan tanggal transaksi dilakukan 2) field untuk memberikan tipe laporan bulanan. Terdapat 2 tipe laporan yaitu *Travel expense* untuk laporan perjalanan dan *Other expense* untuk laporan selain dari laporan perjalanan. 3) input untuk memasukkan total *budget* yang terpakai. Mata uang yang didukung pada sistem ini ada dua yaitu dollar dan rupiah. Pada bagian akhir form ini terdapat field untuk melampirkan bukti-bukti transaksi.



Gambar 11 Halaman buat laporan

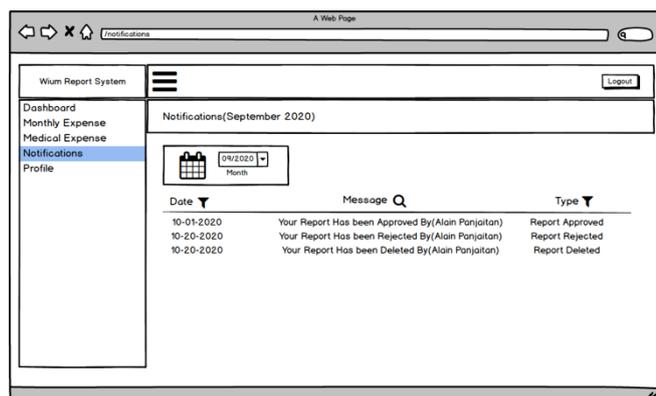
Gambar 12 menunjukkan hasil rancangan antarmuka pengguna untuk halaman profil. Halaman yang dirancang ini berada pada sisi pegawai yang bertujuan untuk memberikan informasi tentang pegawai. Pada bagian atas halaman ini terdapat foto dari pegawai lalu dibawah foto tersebut terdapat field-field yang akan memberikan informasi pegawai yang ditampilkan pada field-field diantaranya nama, departemen, email, nomor telfon, alamat,

total *budget* untuk laporan bulanan, dan total *budget* untuk laporan medis.



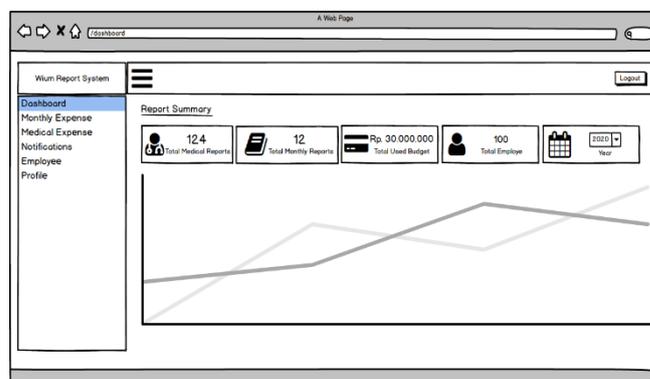
Gambar 12. Halaman profil

Gambar 13 menunjukkan hasil rancangan antarmuka pengguna untuk halaman notifikasi. Halaman yang dirancang ini berada pada sisi pegawai dan keuangan yang bertujuan untuk memberikan informasi setiap aktivitas yang dilakukan pada sistem ini. Pada bagian atas halaman ini terdapat bagian yang menunjukkan bahwa halaman ini adalah halaman notifikasi yang disertai dengan bulan notifikasi ini dibuat. Lalu dibawahnya terdapat pilihan yang memungkinkan user untuk memilih bulan. Lalu pada bagian bawah halaman ini terdapat tabel yang berisi notifikasi-notifikasi. Tabel tersebut terdiri dari beberapa kolom dimana tiap-tiap kolomnya memberikan informasi untuk setiap notifikasi. Kolom tersebut diantaranya 1) *Date* untuk memberikan informasi tanggal notifikasi tersebut dibuat 2) *Message* berisi pesan dari notifikasi tersebut 3) *Type* memberikan informasi tipe dari notifikasi. Pada halaman ini disediakan fungsi pencari dan penyaring.



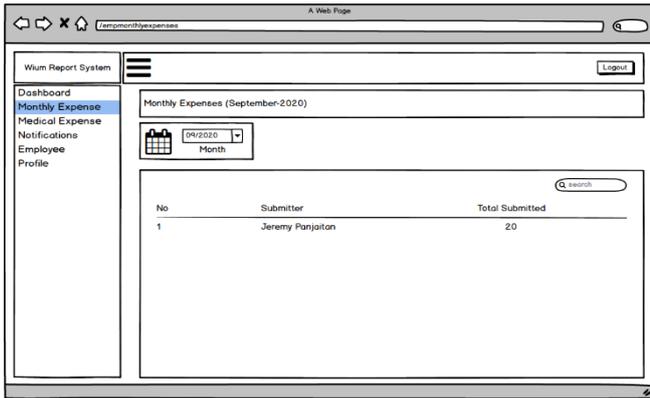
Gambar 13. Halaman notifikasi

Gambar 14 menunjukkan halaman hasil rancangan antarmuka pengguna untuk halaman *dashboard* keuangan. Halaman yang dirancang ini berada pada sisi keuangan yang bertujuan untuk memberikan tinjauan/ringkasan terhadap laporan dan jumlah *budget* yang telah terpakai oleh seluruh pegawai yang ada pada kantor tersebut dalam periode 1 tahun. Pada bagian atas halaman ini terdapat 4 bagian diantaranya, 1) *Total medical reports* memberikan informasi terhadap jumlah seluruh laporan medis dari seluruh pegawai yang telah disetujui 2) *Total monthly reports* memberikan informasi terhadap jumlah seluruh laporan bulanan dari seluruh pegawai yang telah disetujui 3) *Total used budget* memberikan informasi terhadap jumlah budget yang telah terpakai oleh seluruh pegawai 4) *Total employee* memberikan informasi jumlah pegawai 5) *Year* memungkinkan pegawai untuk memilih tahun yang akan dilihat tinjauan dan ringkasan laporannya. Lalu pada bagian bawah halaman ini terdapat grafik yang memberikan gambaran terhadap penggunaan *budget* seluruh pegawai dalam periode waktu 1 tahun.



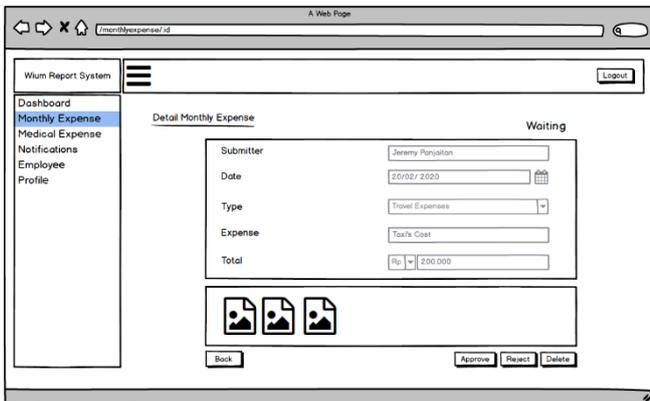
Gambar 14. Halaman *dashboard* keuangan

Gambar 15 menunjukkan hasil rancangan antarmuka pengguna untuk halaman laporan seluruh pegawai. Halaman yang dirancang ini berada pada sisi keuangan yang bertujuan untuk menampilkan seluruh pegawai yang telah membuat laporan dalam jangka waktu 1 bulan beserta jumlah laporan yang telah dibuat oleh tiap-tiap pegawai. Pada halaman ini terdapat menu untuk memungkinkan user untuk memilih bulan. Lalu pada bagian bawah halaman ini terdapat tabel yang akan menampilkan data pegawai yang telah membuat laporan beserta jumlah laporan yang telah dibuat. Pada halaman ini disediakan fungsi untuk mencari.



Gambar 15. Halaman laporan seluruh pegawai

Gambar 16 menunjukkan hasil rancangan antarmuka pengguna untuk halaman lihat laporan pegawai. Halaman yang dirancang ini berada pada sisi keuangan yang bertujuan untuk melihat laporan pegawai baik laporan bulanan maupun laporan medis. Pada halaman ini bagian keuangan dapat melihat dari detail dari laporan yang ditampilkan dalam sebuah form begitu juga dengan bukti-bukti transaksi yang telah dilampirkan oleh pegawai. Pada bagian kanan atas terdapat bagian yang memberikan informasi terhadap status dari laporan. Status terdiri dari 3, antara lain *Waiting* yang berarti sedang menunggu bagian keuangan untuk memeriksa laporan, *Rejected* yang berarti laporan yang dibuat oleh pegawai telah ditolak oleh bagian keuangan dan *Approved* yang berarti laporan yang telah dibuat oleh pegawai telah diterima oleh bagian keuangan. Pada halaman ini terdapat juga tombol *delete* yang memungkinkan bagian keuangan untuk menghapus laporan.



Gambar 16. Halaman lihat laporan pegawai

Gambar 17 menunjukkan hasil rancangan antarmuka pengguna untuk halaman pegawai. Halaman yang dirancang ini berada pada sisi keuangan yang bertujuan untuk menampilkan seluruh pegawai yang terdaftar dalam sistem ini. Setiap pegawai ditampilkan dalam sebuah *cards* (kartu), dimana di dalam kartu tersebut terdapat nama,

departemen, dan foto dari pegawai. Apabila kartu tersebut di klik oleh user maka akan muncul halaman profil dari pegawai tersebut. Terdapat tombol *Create employee* yang berfungsi untuk mengarahkan user kepada halaman buat pegawai. Pada halaman ini disediakan fungsi untuk mencari berdasarkan nama atau departemen pegawai.



Gambar 17. Halaman pegawai

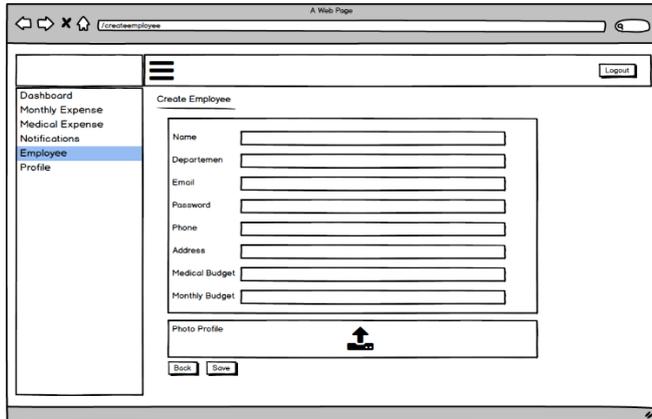
Gambar 18 menunjukkan hasil rancangan antarmuka pengguna untuk halaman profil pegawai. Halaman yang dirancang ini berada pada sisi keuangan yang bertujuan untuk melihat detail profil dari pegawai. Pada halaman ini bagian keuangan memungkinkan untuk mengubah maupun menghapus dari profil pegawai.



Gambar 18. Halaman profil pegawai

Gambar 19 menunjukkan hasil rancangan antarmuka pengguna untuk halaman buat pegawai. Halaman yang dirancang ini berada pada sisi keuangan yang bertujuan untuk membuat pegawai baru. Pada halaman ini terdapat field-field yang akan diisi oleh user dimana field-field ini adalah informasi yang dibutuhkan untuk membuat pegawai baru. User juga diminta untuk memasukkan foto

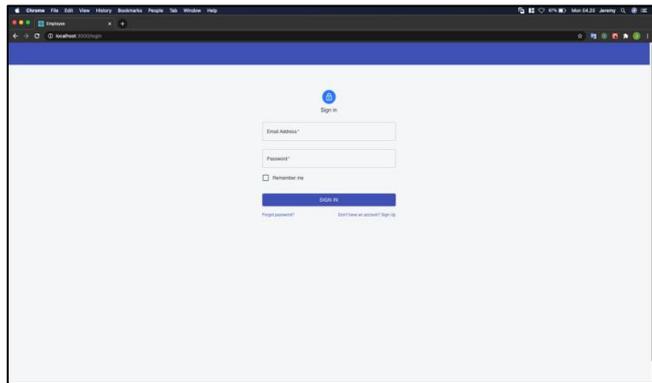
dari pegawai yang akan dibuat. Terdapat 2 tombol di bagian bawah halaman ini yaitu tombol *back* untuk kembali dan tombol *save* untuk memproses pembuatan pegawai baru.



Gambar 19. Halaman buat pegawai

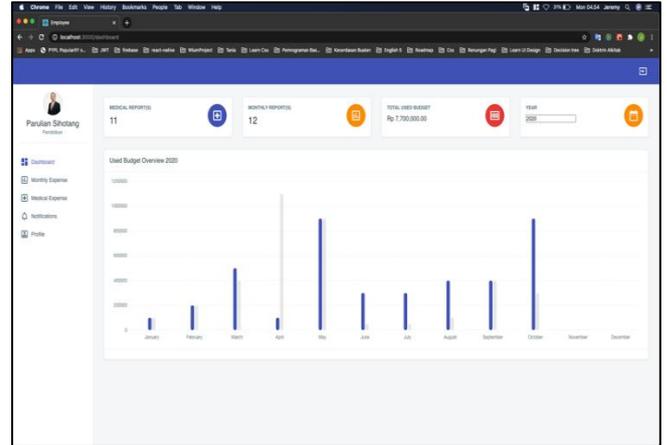
IV. IMPLEMENTASI

A. Halaman login



Gambar 20. Tampilan halaman login

B. Halaman dashboard pegawai



Gambar 21. Tampilan halaman dashboard pegawai

C. Halaman laporan

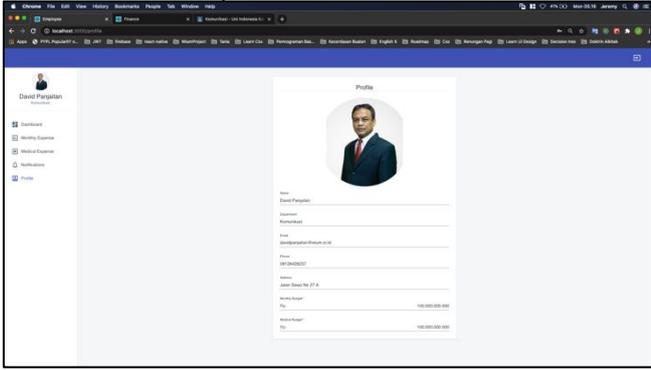
Date	Expense	Type	Total	Status
2020-10-08 09:27:33	My travel to Jakarta	Travel	Rp 200,000.00	Approved
2020-10-23 17:28:46	My travel to Bandung	Travel	Rp 100,000.00	Approved

Gambar 22. Tampilan halaman laporan

D. Halaman buat laporan

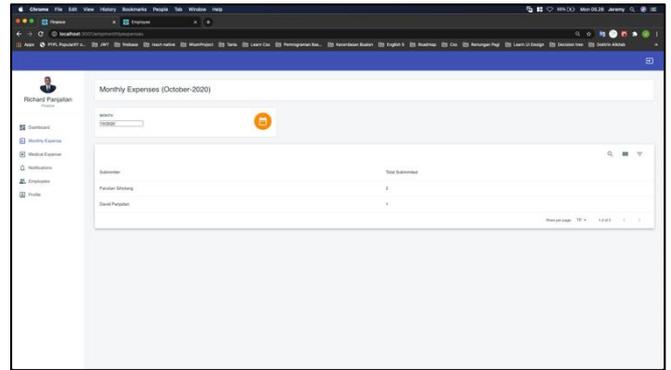
Gambar 23. Tampilan halaman buat laporan

E. Halaman buat laporan



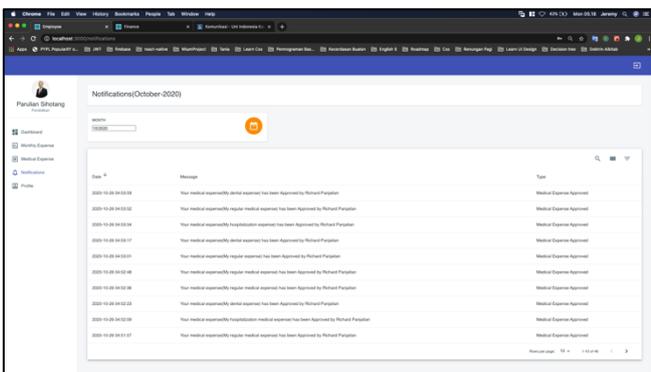
Gambar 24. Tampilan halaman profil

H. Halaman laporan seluruh pegawai



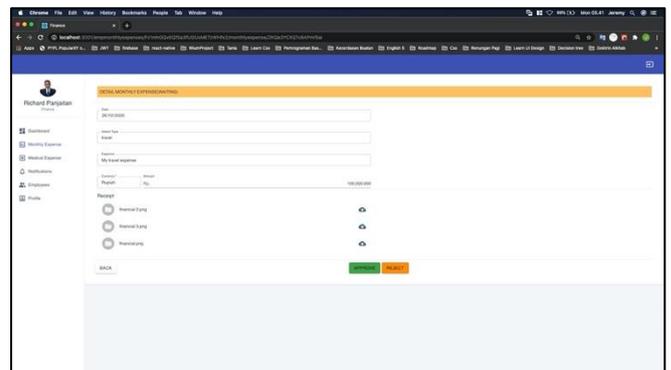
Gambar 27 Tampilan laporan seluruh pegawai

F. Halaman notifikasi



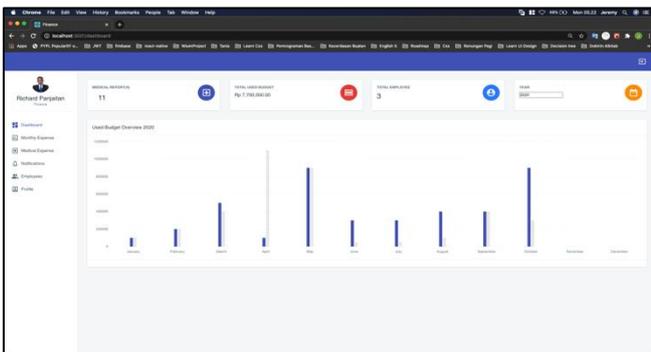
Gambar 25. Tampilan halaman notifikasi

I. Halaman lihat laporan pegawai



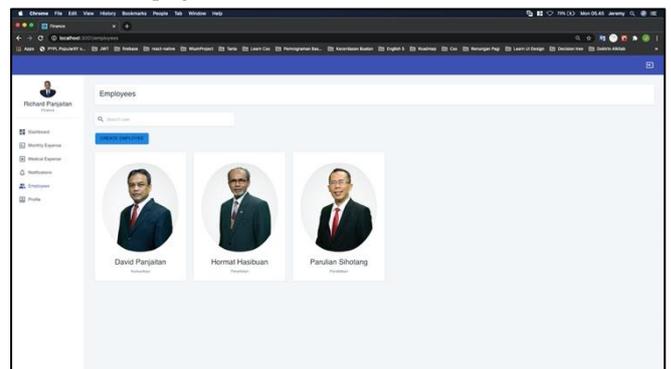
Gambar 28. Tampilan halaman lihat laporan pegawai

G. Halaman dashboard keuangan



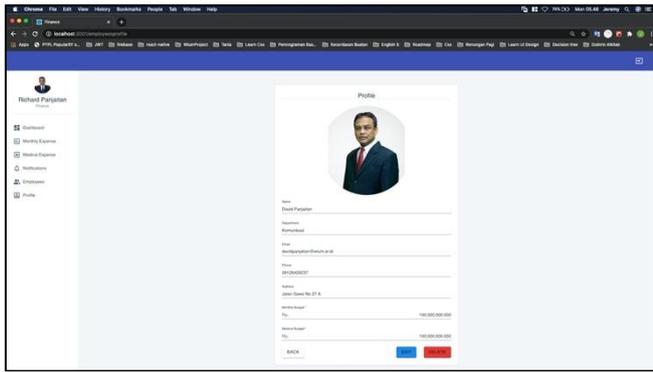
Gambar 26. Tampilan halaman dashboard keuangan

J. Halaman pegawai



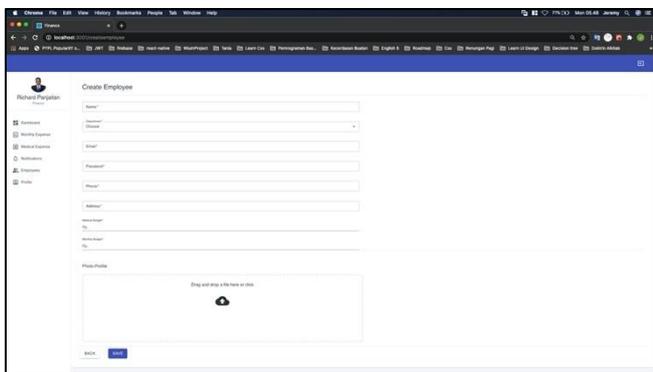
Gambar 29. Tampilan halaman pegawai

K. Halaman profil pegawai



Gambar 30. Tampilan halaman profil pegawai

L. Halaman buat pegawai



Gambar 31. Tampilan halaman buat pegawai

V. HASIL DAN PEMBAHASAN

Untuk melakukan pengujian terhadap sistem, meminta dua orang pegawai dari kantor tersebut untuk melakukan pengujian. Metode pengujian yang digunakan untuk menguji sistem yang telah dirancang adalah metode *Blackbox Testing*. Sistem ini merupakan sebuah metode pengujian yang digunakan untuk menguji fungsionalitas dari sistem yang telah dirancang [17]. Hasil dari pengujian dijelaskan pada tabel berikut.

TABEL IV
HASIL PENGUJIAN SISTEM

Sistem yang diuji	Luaran yang diharapkan	Hasil Pengujian
Login	Jika otentikasi berhasil, user diarahkan ke halaman dashboard	[✓] Valid [] Invalid
Logout	User diarahkan kepada halaman login	[✓] Valid [] Invalid
Pegawai melihat budget overview	Halaman dashboard pegawai menampilkan budget overview dalam	[✓] Valid [] Invalid

Sistem yang diuji	Luaran yang diharapkan	Hasil Pengujian
	jangka waktu satu tahun	
Pegawai Menambahkan monthly expense	Monthly expense tersimpan	[✓] Valid [] Invalid
Pegawai Menambahkan medical expense	Medical expense tersimpan	[✓] Valid [] Invalid
Pegawai menambahkan monthly expense melebihi budget	Sistem akan menolak penambahan monthly expense	[✓] Valid [] Invalid
Pegawai menambahkan medical expense melebihi budget	Sistem akan menolak penambahan medical expense	[✓] Valid [] Invalid
Lihat total budget yang telah terpakai pada monthly expense	Pada halaman monthly expense sistem akan memberikan informasi terhadap total monthly expense dalam jangka waktu satu bulan	[✓] Valid [] Invalid
Lihat notifikasi laporan	Pegawai dapat melihat notifikasi terhadap status laporan yang telah disimpan	[✓] Valid [] Invalid
Lihat total budget yang telah terpakai pada medical expense	Pada halaman medical expense sistem akan memberikan informasi terhadap total medical expense dalam jangka waktu satu bulan	[✓] Valid [] Invalid
Lihat status dari monthly expense yang telah ditambahkan	Pada halaman detail monthly expense akan terlihat status laporan	[✓] Valid [] Invalid
Lihat status dari medical expense yang telah ditambahkan	Pada halaman detail medical expense akan terlihat status laporan	[✓] Valid [] Invalid
Bagian keuangan melihat budget overview seluruh pegawai	Halaman dashboard keuangan menampilkan budget overview dari seluruh pegawai dalam jangka waktu satu tahun	[✓] Valid [] Invalid
Bagian keuangan menyetujui monthly expense pegawai	Pada halaman monthly expense, bagian keuangan dapat menyetujui monthly expense pegawai	[✓] Valid [] Invalid
Bagian keuangan menyetujui medical expense pegawai	Pada halaman medical expense, bagian keuangan dapat menyetujui medical expense pegawai	[✓] Valid [] Invalid
Sistem menolak menyetujui medical expense pegawai yang melebihi budget	Sistem menolak menyetujui medical expense	[✓] Valid [] Invalid
Sistem menolak menyetujui	Sistem menolak menyetujui monthly	[✓] Valid [] Invalid

Sistem yang diuji	Luaran yang diharapkan	Hasil Pengujian
monthly expense pegawai yang melebihi budget	expense	
Bagian keuangan membuat buget dari pegawai	Sistem menyimpan budget dari setiap pegawai	[✓] Valid [] Invalid
Bagian keuangan melihat notifikasi terhadap status laporan	Pada halaman notifikasi bagian keuangan dapat melihat notifikasi terhadap status laporan	[✓] Valid [] Invalid

VI. SIMPULAN

Berdasarkan hasil penelitian dan pembahasan dapat disimpulkan bahwa sistem E-reporting memiliki peranan penting untuk efisiensi pelaporan transaksi pekerjaan pelayanan pada kantor GMAHK. Transaksi yang dilaporkan oleh pegawai dapat disetujui atau ditolak oleh bagian keuangan berdasarkan aturan yang ada. Disamping itu sistem akan mengawasi setiap pengeluaran yang dilakukan oleh pegawai. Berdasarkan uji coba yang telah dilakukan maka pegawai dan bagian keuangan dapat menggunakan berbagai fitur untuk melakukan proses pelaporan pada satu sistem yang terintegrasi.

Beberapa saran pengembangan yang dapat dilakukan adalah mengembangkan tampilan menggunakan ReactJS agar sistem terlihat lebih *responsive* jika digunakan pada perangkat *mobile*, menggunakan layanan firebase agar fitur notifikasi dapat dilakukan secara *real time*, mengembangkan sistem agar dapat menggunakan berbagai macam mata uang asing, menambahkan fitur untuk mencetak bukti bahwa laporan telah diterima/ditolak

DAFTAR PUSTAKA

- [1] A. E. D. A. N. E. Kerusakan *et al.*, "Kerja Pelaksanaan Jalan Wilayah Iii Palembang Dengan Metode V-Model Berbasis Website," no. 31.
- [2] N. Nazir and G. Darmawati, "Perancangan Pencatatan Dan Pelaporan Terpadu Puskesmas Berbasis E-Report Untuk Meningkatkan Kesehatan Masyarakat," *Jurnal Sains dan Teknologi: Jurnal Keilmuan dan Aplikasi Teknologi Industri*, vol. 18, no. 2, p. 75, 2018, doi: 10.36275/stsp.v18i2.109.
- [3] W. J. Rudman, J. H. Bailey, C. Hope, P. Garrett, and C. A. Brown, "The Impact of a Web-based Reporting System on the Collection of Medication Error Occurrence Data," *Advances in Patient Safety: From Research to Implementation (Volume 3: Implementation Issues)*, pp. 195–206, 2005, [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/21249994>.
- [4] R. H. Dai, "Rancang Bangun Aplikasi E-Report Pengaduan Masyarakat Design Public Complaint E-Report Application," *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 2, no. 1, pp. 64–73, 2017, doi: 10.21831/elinvo.v2i1.16421.
- [5] Y. Yuliawan, M. J. D. Sunarto, and T. Soebijono, "Pengembangan Sistem Informasi Pendataan Jemaat Gereja Masehi Advent Hari Ketujuh Konferens Jawa Kawasan Timur Berbasis Web," *JSIKA*, vol. 2, no. 2, p. 25, 2017, [Online]. Available: [https://repository.widyatama.ac.id/xmlui/bitstream/handle/123456789/8241/Bab 2.pdf?sequence=8](https://repository.widyatama.ac.id/xmlui/bitstream/handle/123456789/8241/Bab%202.pdf?sequence=8).
- [6] A. Rama Febrianto, A. Wulansari, and L. Latipah, "Pengembangan Sistem Pengelolaan dan Pemantauan Proyek dengan Metode Agile Pola Scrum," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, pp. 206–221, 2020, doi: 10.28932/jutisi.v6i2.2592.
- [7] F. Nurzaman, "Pengembangan Sistem Otomatisasi Tagihan Menggunakan Metode Agile Software," vol. 4, no. 1, pp. 46–57, 2020, [Online]. Available: https://en.wikipedia.org/wiki/Knowledge-based_systems.
- [8] R. S. Pressman, *Software engineering: a practitioner's approach*. 2015.
- [9] P. S. Ganney, S. Pisharody, and E. Claridge, *Software Engineering*. 2013.
- [10] I. Mahendra and deny tresno eby Yanto, "Agile Development Methods Dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus : Bank Bri Unit Kolonel Sugiono)," *Jurnalteknologi Dan Open Source*, vol. 1, no. proses pengajuan kredit pada Bank BRI Unit Kolonel Sugiono masih dilakukan secara manual., pp. 13, 14, 15, 2018.
- [11] H. P. Ramadhan, C. Kartiko, and A. Prasetiadi, "Monitoring Kualitas Air Tambak Udang Menggunakan NodeMCU, Firebase, dan Flutter," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 1, pp. 102–114, 2020, doi: 10.28932/jutisi.v6i1.2365.
- [12] H. M. Christon, E. B. Wagiu, and Y. Palopak, "Perancangan Sistem E-Learning Berbasis Android Dengan Menggunakan Firebase Pada Universitas Advent Indonesia," *TeIka*, vol. 8, no. 2, pp. 97–105, 2018, doi: 10.36342/teika.v8i2.755.
- [13] L. A. Sandy, R. J. Akbar, and R. R. Hariadi, "Rancang Bangun Aplikasi Chat pada Platform Android dengan Media Input Berupa Canvas dan Shareable Canvas untuk Bekerja dalam Satu Canvas Secara Online," *Jurnal Teknik ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23782.
- [14] "Cloud Functions for Firebase." <https://firebase.google.com/docs/functions>.
- [15] E. A. W. Sanad, "Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire," *Jurnal Penelitian Enjiniring*, vol. 22, no. 1, pp. 20–26, 2019, doi: 10.25042/jpe.052018.04.
- [16] M. Wali and L. Ahmad, "Perancangan Access Open Journal System (AOJS) dengan menggunakan Framework Codeigniter dan ReactJs," *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 2, no. 1, p. 48, 2018, doi: 10.35870/jtik.v2i1.53.
- [17] F. F. Nursaid, A. Hendra Brata, and A. P. Kharisma, "Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri)," *J-Ptiik.Ub.Ac.Id*, vol. 4, no. 1, pp. 46–55, 2020, [Online]. Available: <http://j-ptiik.ub.ac.id>.
- [18] F. J. Kaunang, "Rancang Bangun Sistem Reservasi Kamar pada Manadia Guest House Manado berbasis Web," *TeIka*, vol. 10, no. 01, pp. 15–23, 2020, doi: 10.36342/teika.v10i01.2253.
- [19] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan)," *Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 1, pp. 1–9, 2018, [Online]. Available: <http://jurnal.uinsu.ac.id/index.php/algorithm/article/download/3148/1871>.
- [20] "Memilih Database: Cloud Firestore atau Realtime Database." <https://firebase.google.com/docs/firestore/>.
- [21] Z. Munawar, M. I. Fudsyi, and D. Z. Musadad, "Perancangan Interface Aplikasi Pencatatan Persediaan Barang Di Kios Buku Palasari Bandung Dengan Metode User Centered Design Menggunakan Balsamiq Mockups," *Jurnal Informatika*, vol. 6, no. 2, pp. 10–20, 2019.
- [22] S. L. Henry, "Notes on User Centered Design Process (UCD)," 2004. <https://www.w3.org/WAI/redesign/ucd> (accessed Oct. 25, 2020).
- [23] I. A. Astuti, M. Suyanto, and Sukoco, "Penerapan Metode User Centered Design Pada Game Based," *Jurnal Informatika*

- [24] *Interaktif*, vol. 2, no. 1, pp. 10–20, 2017.
I. S. Yatana Saputri, M. Fadhli, and I. Surya, “Penerapan Metode UCD (User Centered Design) Pada E-Commerce Putri Intan

Shop Berbasis Web,” *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 3, no. 2, pp. 269–278, 2017, doi: 10.25077/teknosi.v3i2.2017.269-278.