

Implementasi Fitur Sinkronasi Basis Data Pada Aplikasi Monitoring Keuangan Pada Wilayah dengan Keterbatasan Jaringan Internet

Aldi Setiawan¹, Danny Sebastian², Kristian Adi Nugraha³

Informatika, Universitas Kristen Duta Wacana

Jl. Dr. Wahidin Sudirohusodo No.5-25, Yogyakarta

¹aldi.setiawan@ti.ukdw.ac.id

²danny.sebastian@staff.ukdw.ac.id

³adinugraha@staff.ukdw.ac.id

Abstract— Tobacco farmers in Central Java often have commercial relationships with many parties. One of the actions of the process is the process of accounts payable between farmers and traders. The difficulty that arises from this process is when recording debt data every day. Often farmers in these areas still use paper as a medium to store debt data. Of course, this can be a problem when the paper is lost and the debt data that is owned is also lost. So it is proposed to make a debt recording application on a mobile device so that it is easy to carry and the stored data is not easily lost. However, a new problem arises, namely when the mobile device is not connected to the internet network whether debt data can still be stored, seeing that the area where the farmer is located does not have an adequate internet network.

Based on the problems encountered, it is proposed to create a synchronization algorithm that can handle when the mobile device is not connected to the internet network. The data will be synchronized on the website application created, so that the data entered by farmers when they are not connected to the internet network can be stored properly without fear of losing the debt data.

Tests are carried out using white box testing and actual testing. For the white box testing that was tested on the algorithm that has been made, the results show that the algorithm created has a low risk level and a 5 percent improvement probability. For the actual tests carried out, based on the results obtained, each test scenario can be tested well and produce good results. So it can be concluded that the algorithm that has been made is feasible to be applied to mobile devices.

Intisari— Petani tembakau pada Jawa Tengah sering melakukan hubungan komersial terhadap banyak pihak. Salah satu dari aksi proses tersebut adalah proses hutang piutang antara petani dengan pedagang. Kesulitan yang muncul dari proses tersebut adalah ketika mencatat data hutang setiap harinya. Seringkali petani pada daerah tersebut masih menggunakan kertas sebagai media untuk menyimpan data hutang. Tentunya hal ini dapat menjadi masalah ketika kertas tersebut hilang dan data hutang yang dimiliki juga hilang. Sehingga diusulkan untuk membuat aplikasi pencatatan hutang pada perangkat bergerak sehingga mudah dibawa dan data yang disimpan tidak mudah hilang. Tetapi muncul permasalahan baru yaitu ketika perangkat bergerak tidak terhubung dengan jaringan internet apakah data hutang masih tetap dapat disimpan melihat bahwa daerah tempat petani berada tidak memiliki jaringan internet yang memadai.

Berdasarkan pada masalah yang dihadapi, maka diusulkan untuk membuat algoritma sinkronisasi yang dapat menangani ketika perangkat bergerak tidak terhubung dengan jaringan internet. Data tersebut akan dilakukan sinkronisasi pada

aplikasi situs web yang dibuat, sehingga data yang dimasukkan oleh petani ketika tidak terhubung dengan jaringan internet dapat tersimpan dengan baik tanpa takut kehilangan data hutang tersebut.

Pengujian dilakukan dengan menggunakan pengujian kotak putih dan pengujian aktual. Untuk pengujian kotak putih yang diujikan pada algoritma yang telah dibuat, didapatkan hasil bahwa algoritma yang dibuat memiliki tingkat resiko yang rendah dan probabilitas perbaikan sebesar 5 persen. Untuk pengujian aktual yang dilakukan, berdasarkan hasil yang didapat tiap skenario pengujian dapat teruji dengan baik dan menghasilkan hasil yang baik. Sehingga dapat disimpulkan bahwa algoritma yang telah dibuat layak untuk diterapkan pada perangkat bergerak.

Kata Kunci— Perangkat bergerak, sinkronisasi, jaringan internet.

I. PENDAHULUAN

Seiring dengan perkembangan zaman, hubungan interaksi yang terjadi antar individu semakin beragam dimana salah satunya adalah hubungan yang bersifat ekonomis dan komersial. Dalam hal ini seringkali terjadi proses jual beli untuk barang maupun jasa dan tidak luput juga dari sistem pencatatan transaksi. Sistem ini digunakan oleh penjual untuk mengetahui riwayat transaksi yang telah terjadi sebelumnya. Sama halnya dengan perniagaan yang dilakukan oleh pedagang dan petani.

Pedagang dan petani memiliki hubungan ekonomis dan komersial. Hubungan tersebut dapat diartikan dengan pedagang akan menjual kepada petani dan petani akan membeli barang dari pedagang. Tetapi berbeda dengan pedagang dan petani lainnya, terdapat pedagang dan petani tembakau di daerah Jawa Tengah yang menggunakan sistem hutang piutang. Pada awal tahun sekitar bulan Februari hingga April, terkadang petani tidak memiliki modal untuk bertanam. Karena hal ini maka petani akan melakukan hutang modal kepada pedagang lalu pada saat musim panen tiba yakni di antara bulan Agustus hingga Oktober petani dapat mengembalikan hutang yang dipinjam dengan menggunakan hasil penjualan yang didapat.

Ketika transaksi hutang piutang terjadi, pedagang pasti memerlukan pencatatan. Pencatatan ini diperlukan oleh pedagang untuk menentukan jumlah modal yang dapat dipinjam oleh petani. Pencatatan yang dilakukan oleh

pedagang memiliki cara yang beragam, yaitu dapat ditulis di media cetak dan dapat diinputkan pada perangkat komputer. Ketika transaksi dituliskan pada media cetak maka hal ini akan mengharuskan pedagang membawa media cetak tersebut dimanapun dan kapanpun pedagang melakukan transaksi lalu jika media cetak tersebut hilang atau rusak maka pedagang akan kehilangan data transaksi yang dimiliki. Lalu saat data transaksi disimpan pada perangkat komputer, maka hal ini akan membuat data transaksi tidak dapat dibawa karena perangkat komputer tidak dapat dibawa kemana saja. Muncul suatu permasalahan dari keterbatasan pada masing-masing cara penyimpanan, yaitu pada saat data transaksi yang disimpan tidak aktual akan membuat pedagang memberikan pinjaman modal yang terlalu besar kepada petani sehingga membuat hasil penjualan tembakau tidak bisa menutup hutang yang dipinjam oleh petani. Diperlukan suatu perangkat yang dapat dibawa kemana saja serta memiliki data aktual. Hal ini didukung dengan data penggunaan perangkat bergerak di Indonesia sejumlah 345 juta pengguna yang terhubung dengan koneksi internet [1].

Perangkat bergerak merupakan perangkat yang dapat dibawa kemana saja dan memiliki fitur yang lengkap. Fitur yang ditawarkan adalah dapat terhubung dengan jaringan internet serta dapat menyimpan data dimanapun dan kapanpun ketika digunakan. Ketika data transaksi disimpan di sebuah penyimpanan di cloud maka diperlukan jaringan internet untuk mengambil data tersebut. Ketika perangkat sedang kesulitan dalam menangkap jaringan internet maka data transaksi tidak dapat diambil dan tidak dapat digunakan oleh pedagang yang mana hal ini akan menjadi masalah baru bagi petani. Sehingga diperlukan suatu solusi yang dapat menyelesaikan permasalahan dalam jaringan internet dan perangkat yang memiliki sifat portable bagi petani.

Berdasarkan dengan pemaparan masalah yang telah dijelaskan sebelumnya, penelitian ini akan membuat suatu aplikasi yang dapat difungsikan dalam keadaan ketika sedang terhubung dan tidak terhubung dengan jaringan internet. Serta aplikasi ini akan memungkinkan untuk melakukan sinkronisasi data saat mengalami kedua kondisi sebelumnya. Sinkronisasi dilakukan pada dua perangkat berbeda dengan masing-masing basis data bertipe SQL dan JSON. Aplikasi akan dikembangkan dengan metode prototyping lalu akan diujikan dengan pengujian kotak putih dengan rangkaian pengujian. Ditambah lagi data yang ditampilkan untuk pedagang akan dibuat mudah untuk dibaca dan dipahami sehingga akan mempermudah pengguna ketika menggunakan aplikasi. Dengan menggabungkan fitur-fitur yang ada maka pedagang akan dapat membaca data transaksi secara keseluruhan ketika tidak terhubung dengan jaringan internet. Sehingga masalah yang sebelumnya sudah dijelaskan dapat teratasi dengan baik.

Pembuatan aplikasi pada perangkat bergerak berbasis Android dan iOS memiliki berbagai cara. Untuk pembuatan aplikasi pada perangkat bergerak berbasis Android dapat dibuat dengan menggunakan bahasa pemrograman khusus seperti Java sedangkan untuk aplikasi cross-platform dapat menggunakan framework seperti React Native dan Flutter.

Pada tahun 2017, Kocakoyun membuat aplikasi Android berbasis Java untuk diteliti [2]. Beliau mengatakan bahwa pembuatan aplikasi Android dengan menggunakan metode ini cocok untuk pemula dan mudah untuk diterapkan. Pada 2016, Hansson & Vidhall membuat penelitian tentang pembuatan aplikasi Android berbasis React Native. Beliau mengatakan bahwa dengan menggunakan pembuatan aplikasi dengan metode ini maka terdapat beberapa hal yang harus diperhatikan antara lain adalah penggunaan CPU yang berlebih karena React Native karena dapat dijalankan juga di iOS tetapi penggunaan metode ini masih dianjurkan [3]. Pengembangan penelitian tentang aplikasi Android juga dilakukan oleh Payne, 2019 yang dituliskan dalam bukunya tentang membuat aplikasi dengan Flutter. Beliau mengatakan bahwa Flutter memiliki pengembangan yang cepat dan memiliki teknik pengembangan yang mudah untuk dipelajari jika dibandingkan dengan framework cross-platform lainnya. Untuk pembuatan aplikasi pada perangkat bergerak berbasis iOS dengan menggunakan bahasa pemrograman khusus seperti Swift sedangkan untuk aplikasi cross-platform dapat menggunakan framework seperti React Native dan Flutter [4]. Pada 2015, Wells melakukan evaluasi tentang bahasa pemrograman Swift. Bahasa ini merupakan bahasa pemrograman yang dikhususkan untuk membuat aplikasi iOS dan OS X. Sehingga kecepatan, performa, dan kualitas dari aplikasi yang dibuat memiliki performa yang baik dan cepat saat digunakan. Karena pembuatan aplikasi iOS dan OS X hanya dapat dilakukan pada iOS maka hal ini dapat menjadi pertimbangan ketika mengembangkan aplikasi tersebut [5].

Pembuatan aplikasi yang dapat dijalankan pada kedua sistem operasi dapat dilakukan dengan menggunakan framework seperti React dan Flutter. Seperti yang dijelaskan sebelumnya bahwa aplikasi yang dibangun dengan React Native memiliki penggunaan CPU yang besar, hal ini dikarenakan React Native dijalankan dengan menggunakan Javascript sehingga memerlukan optimasi performa untuk ke depannya. Kelebihan Flutter dibandingkan dengan framework cross-platform lainnya adalah konsistensi pada frame per second (FPS) yang diberikan, dapat menggunakan fungsi-fungsi yang disediakan dari Firebase dengan mudah. Sehingga pengembangan aplikasi dengan Flutter layak untuk dipertimbangan dan dicoba [6].

Pada tahun 2016, Setiyawati dkk membuat penelitian tentang pembuatan aplikasi web untuk manajemen laundry dengan integrasi data dan web service. Penelitian ini dibuat dengan menggunakan PHP, HTML, CSS dan Javascript untuk pengembangan situs webnya. Untuk media penyimpanan digunakannya MySQL sebagai basis data. Hasil akhir yang didapat pada penelitian ini adalah aplikasi web yang dikembangkan berhasil digunakan dengan metode pengujian black box, dimana setiap skenario yang diujikan memiliki nilai yang valid [7].

Pada tahun 2017 Wicaksana, membuat suatu penelitian tentang sinkronisasi basis data untuk basis data SQL dan NoSQL dengan menggunakan Data Adapter dengan metode Access Query Approach. Data Adapter berfungsi untuk menerima berbagai query yang dikirimkan pengguna untuk

diteruskan ke basis data. Pengujian dilakukan dengan melakukan rangkaian pengujian seperti melakukan operasi CRUD terhadap basis data, kemudian hasil menunjukkan bahwa dengan metode ini sinkronisasi bekerja seratus persen berhasil [8].

Pada tahun 2019 Pamuji, dkk membuat suatu penelitian tentang sinkronisasi basis data berbasis RESTful Web Services untuk aplikasi presensi. Penelitian ini melibatkan pembuatan web service dan juga aplikasi berbasis android untuk perangkat bergerak. Ketika perangkat tidak terhubung dengan jaringan internet maka perangkat akan menyimpan data dalam basis data lokal dan ketika tombol sinkronisasi pada aplikasi ditekan maka akan dilakukan pengecekan terlebih dahulu apakah sudah terhubung dengan internet atau tidak. Ketika sudah terhubung maka tahap selanjutnya adalah memperbaharui basis data yang berada di server dengan yang di lokal. Hasil dari penelitian yang dilakukan menunjukkan keberhasilan sebesar seratus persen dengan percobaan sebanyak dua puluh lima kali, tetapi terdapat sedikit saran untuk pengembangan agar tidak perlu menekan tombol sinkronisasi pada aplikasi sehingga sinkronisasi dilakukan secara otomatis [9].

Pada tahun 2016, Kekgathetse & Letsholo melakukan survey penelitian tentang tentang sinkronisasi basis data untuk perangkat bergerak. Pada penelitian ini terdapat Synchronization Algorithms based on Message Digests, Generic Data Synchronization Approaches, dan XML Based Synchronization yang merupakan algoritma untuk melakukan sinkronisasi basis data. Metode yang pertama adalah Synchronization Algorithms based on Message Digests merupakan metode sinkronisasi basis data dengan membuat dua tabel penampang pada masing-masing penyimpanan. Lalu terdapat Generic Data Synchronization Approaches merupakan metode sinkronisasi data berdasarkan jejak waktu pada tiap data yang berubah, ketika terdapat perubahan data maka hanya kolom tersebut yang dilakukan sinkronisasi. Terakhir terdapat XML Based Synchronization merupakan metode sinkronisasi yang menggunakan pertukaran data bertipe XML untuk proses sinkronisasinya [10]. Pada tahun selanjutnya dilakukan penelitian tentang sinkronisasi antara dua basis data yang berbeda dilakukan oleh Jha & Sebastian, 2017. Penelitian ini membahas tentang teknik dan algoritma untuk menghubungkan basis data yang berada di penyimpanan lokal dan penyimpanan cloud. Secara sederhana langkah dari algoritma yang dibuat adalah memeriksa apakah sudah terhubung dengan server atau belum, jika belum maka akan dibuat basis data sementara untuk data yang ada, lalu jika sudah berhasil menghubungkan dengan server maka dilakukan proses pengambilan data dari basis data cloud lalu akan dilakukan perbandingan data dengan data yang berada di lokal jika ada perbedaan maka akan dilakukan penulisan data pada basis data lokal. Hasil akhir yang didapat adalah basis data dapat terhubung antara satu dengan yang lain tetapi perlu dilakukan pengujian akan akurasi dari datanya [11].

Table 1. Tabel perbedaan penelitian sebelumnya

PERBEDAAN	NILAI		
	SAMD	Generic Data Synchronization	XML Based Synchronization
Metode Sinkronisasi	Bertipe SQL	Bertipe SQL atau XML	Bertipe XML
Format Data	Membandingkan basis data lokal dengan basis data server, jika ada perbedaan maka akan dilakukan sinkronisasi	Membandingkan basis data lokal dengan basis data server, jika ada perbedaan maka akan dilakukan sinkronisasi	Membandingkan basis data lokal dengan basis data server, jika ada perbedaan maka akan dilakukan sinkronisasi
Cara Kerja	Perangkat mobile dan Server	Sistem pada aplikasi mobil atau desktop	Perangkat mobile dan Server

Dengan melihat pada tabel 1 yang berisi tentang perbandingan metode untuk sinkronisasi pada basis data di perangkat bergerak dan server maka dalam penelitian ini akan membuat suatu metode dengan menggunakan tinjauan pada metode yang sudah ada untuk dilakukan suatu sinkronisasi antara basis data yang memiliki tipe SQL dengan berbasis JSON. Basis data dengan tipe SQL akan digunakan sebagai basis data pada perangkat bergerak, sedangkan untuk basis data dengan tipe JSON akan digunakan sebagai basis data pada server. Metode sinkronisasi akan ditulis dengan menggunakan bahasa pemrograman Dart dengan menggunakan framework Flutter. Sehingga penelitian ini akan menghasilkan suatu aplikasi berbasis Android untuk perangkat bergerak dengan menggunakan menggunakan bahasa pemrograman Flutter. Lalu akan dilakukan implementasi fitur sinkronisasi data pada basis data lokal yang terdapat pada perangkat bergerak dengan basis data berbasis Cloud.

II. LANDASAN TEORI

A. Basis Data

Merupakan kumpulan informasi yang disimpan dalam suatu ruang penyimpanan secara sistematis untuk mempermudah mengatur data yang disimpan. Satu basis data menunjukkan kumpulan informasi data yang digunakan dalam suatu lingkup. Suatu record terdiri dari beberapa field yang berhubungan dimana kumpulan record akan membentuk suatu tabel atau biasa disebut entity [12].

Basis data yang digunakan adalah:

1. SQLite

SQLite adalah suatu library yang mesin database self-contained, server-less, zero-configuration, dan transactional. Serverless berarti SQLite dalam mengakses database baik itu read atau write dapat secara langsung dari file database tanpa melalui server [13].

2. Firebase

Merupakan suatu framework yang berguna untuk membangun aplikasi portabel dan website untuk hal yang

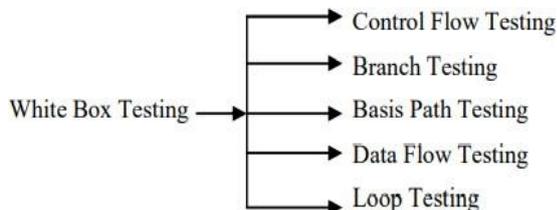
membutuhkan data secara real-time ketika satu pengguna memperbarui catatan dalam basis data maka pembaharuan akan terjadi seketika data diperbaharui. Data yang disimpan pada basis data akan disimpan dalam format JSON. Dengan menggunakan format penyimpanan tipe ini maka akan mempermudah proses komunikasi dan pertukaran data [14].

B. Application Programming Interface (API)

Merupakan sekumpulan fungsi yang membuat suatu aplikasi atau program dapat berinteraksi dengan aplikasi atau program lainnya. Melalui API suatu sistem dapat berinteraksi dan melakukan pertukaran data antar dua aplikasi yang berbeda dapat terhubung secara bersamaan. Terdapat request dan response, request merupakan permintaan akan data kepada suatu aplikasi sedangkan response merupakan jawaban dari permintaan yang biasanya berupa data yang diterima oleh aplikasi.

C. Pengujian Kotak Putih (White Box Testing)

Merupakan teknik pengujian aplikasi untuk menguji fungsionalitas dari suatu program atau aplikasi dengan menggunakan rangkaian skenario pengujian yang dibentuk dengan menggunakan algoritma pada aplikasi. Pengujian dilakukan dengan menggunakan satu atau beberapa jalur pada algoritma dengan menggunakan parameter tertentu. Pengujian ini lebih berfokus kepada logika atau alur kerja pada aplikasi serta untuk mengetahui alur data yang berjalan selama aplikasi berlangsung [15].



Gambar 1. Macam jenis pengujian pada pengujian kotak putih

Terdapat beberapa teknik yang digunakan dalam pengujian ini yaitu:

1. Control Flow Testing

Merupakan teknik yang menggunakan alur dari program sebagai control untuk menampilkan logika yang kompleks serta memudahkan melihat alur program.

2. Branch Testing

Teknik ini digunakan untuk melakukan pengujian terhadap percabangan yang ada.

3. Basis Path Testing

Digunakan untuk menghasilkan pengukuran alur program dimana pengukuran ini akan digunakan untuk membuat pengujian pada program.

4. Data Flow Testing

Teknik ini digunakan untuk melakukan perincian terhadap variabel yang digunakan serta hasilnya dalam diagram alir yang ada. Dengan perlakuan seperti ini maka variabel akan dapat diketahui kegunaannya dan dimana saja variabel tertentu digunakan.

5. Loop Testing

Teknik pengujian yang fokus kepada eksekusi dari perulangan yang ada. Tiap perulangan akan diuji untuk

dilihat apakah outputnya sudah sesuai dengan yang diharapkan.

D. Sinkronisasi

Metode sinkronisasi adalah metode untuk menyesuaikan data yang disimpan pada basis data lokal dengan data yang tersimpan pada basis data server. Hal ini bertujuan untuk membangun basis data yang memiliki konsistensi antara satu dengan yang lain.

Pengertian menurut beberapa ahli:

1. Sinkronisasi data adalah proses membangun konsistensi antara data dari sumber ke target penyimpanan data atau sebaliknya, serta merupakan harmonisasi data yang berkesinambungan dari waktu ke waktu [16].

2. Sinkronisasi data merupakan proses pembentukan konsistensi data antar basis data dengan menggunakan basis data utama sebagai target penyimpanan data dan database cabang sebagai target tujuan atau sebaliknya (Kaur, A, 2014).

E. Synchronization Algorithms based on Message Digests (SAMD)

Merupakan metode yang digunakan untuk basis data pada server dan pada lokal memiliki konsistensi di dalamnya. Metode ini membandingkan dua tabel pada basis data yang berbeda untuk dilakukan sinkronisasi, ketika terdapat satu atau lebih data yang berbeda maka akan dilakukan sinkronisasi menggunakan metode ini. Metode pengujian dilakukan dengan menjalankan berbagai perintah CRUD pada basis data, dan melakukan analisis apakah data setelah dilakukansinkronisasi dan sebelum sinkronisasi memiliki perbedaan atau tidak. Sinkronisasi hanya dilakukan pada tiap baris yang memiliki perubahan data [17].

Metode ini menggunakan SQL standard yang digunakan untuk melakukan sinkronisasi. Berikut merupakan langkah dalam algoritma ini:

- Pada server memiliki dua basis data yaitu basis data utama dan basis data cadangan untuk melakukan sinkronisasi
- Masing-masing kolom memiliki baris flag untuk memberi tanda bahwa terdapat baris yang memiliki perubahan
- Ketika terjadi perubahan pada basis data lokal, flag yang sebelumnya bernilai nol (nilai default) akan diubah menjadi satu.
- Lalu akan dibandingkan dengan basis data yang berada pada server apakah mengalami perubahan, jika iya maka pada basis data cadangan akan berisi nilai pada basis data lokal yang mengalami perubahan lalu nantinya juga akan mengambil primary key dari baris yang mengalami perubahan.
- Ketika sudah dipindahkan ke basis data cadangan, selanjutnya adalah mengubah nilai pada basis data cloud.
- Kemudian flag pada basis data yang menjadi satu akan diubah menjadi 0.
- Proses sinkronisasi selesai.

2	Pengguna Situs Web	Aktor ini memiliki wewenang untuk melakukan menambah, mengubah, menghapus, membaca data, dan melakukan perintah sinkronisasi.
3	Sistem	Aktor ini memiliki wewenang untuk melakukan sinkronisasi dan menerima semua masukan dari pengguna.

2. Use Case Definition

Pada bagian ini akan menjelaskan tentang use case yang ada pada diagram tersebut.

Table 3. Tabel deskripsi Use Case Definition

No	Use Case	Deskripsi
1	Menambah Data	Menambah data pada basis data lokal maupun cloud.
2	Mengubah Data	Mengubah data pada basis data lokal maupun cloud.
3	Menghapus Data	Menghapus data pada basis data lokal maupun cloud.
4	Membaca Data	Membaca data pada basis data lokal maupun cloud.
5	Menjalankan Sinkronisasi	Menjalankan perintah sinkronisasi yang disediakan oleh sistem
6	Melakukan sinkronisasi pada basis data lokal dan cloud	Menjalankan sinkronisasi untuk menjaga konsistensi data antara basis data lokal dan cloud.
7	Melakukan segala perubahan yang terjadi pada basis data	Menerima semua masukan pengguna, kemudian akan dilakukan perubahan berdasarkan masukan dari pengguna.

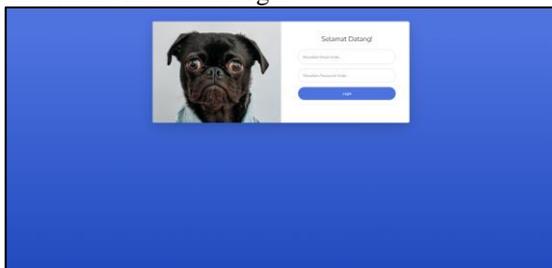
IV. ANALISIS DAN PEMBAHASAN

A. Implementasi Antarmuka

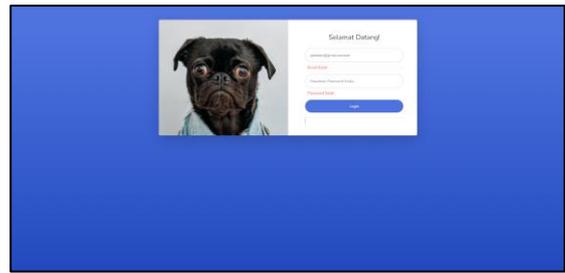
1. Implementasi Antarmuka Situs Web

Pada bagian ini akan dijelaskan tentang implementasi antarmuka yang telah dibuat berdasarkan dengan desain antarmuka yang telah dirancang sebelumnya untuk antarmuka pada situs web.

a. Antarmuka Halaman Login



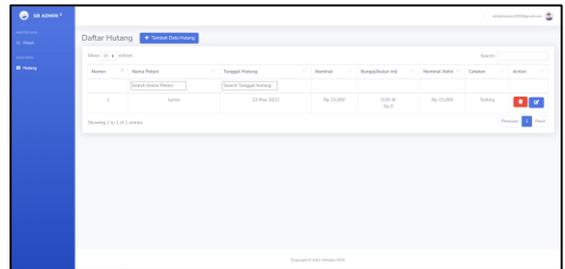
Gambar 4. Tampilan halaman Login untuk situs web



Gambar 5. Tampilan halaman Login ketika salah input password dan email untuk situs

Untuk tampilan halaman login pada situs web dibuat sesuai dengan desain yang ada. Pengguna perlu memasukkan email dan password yang ada dapat dilihat pada gambar 4. Juga ditambahkan suatu tulisan yang berguna untuk pemberitahuan ketika email dan password yang dimasukkan salah maka akan muncul tulisan merah dibawah field email dan password berada dapat dilihat pada gambar 5.

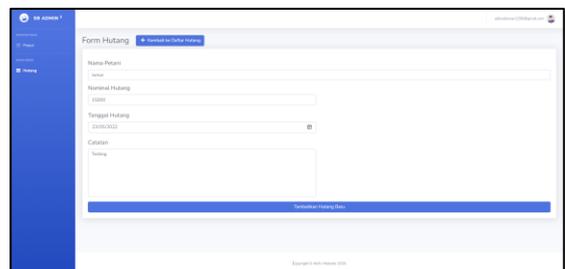
b. Antarmuka Halaman Hutang



Gambar 6. Tampilan halaman hutang untuk situs web

Untuk tampilan halaman hutang pada situs web dibuat sesuai dengan desain yang ada dapat dilihat pada gambar 6. Pada halaman ini terdapat beragam data yang ditampilkan seperti nama petani yang meminjam hutang, tanggal peminjaman hutang, nominal dari hutang yang dipinjam, dan bunga yang dimiliki oleh hutang tersebut. Serta ditambahkan juga catatan dan nominal akhir yang merupakan hasil dari penjumlahan antara nominal dengan nominal dikalikan dengan bunga. Di bagian paling kanan juga terdapat tombol aksi seperti hapus dan update yang masing-masing berguna untuk menghapus data dan melakukan perubahan pada data yang dipilih.

c. Antarmuka Halaman Insert Data – Hutang

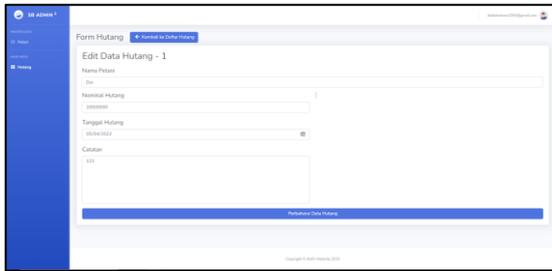


Gambar 7. Tampilan halaman hutang untuk insert data untuk situs web

Untuk halaman insert pada hutang dapat dilihat sebagai berikut pada gambar 7 dimana pengguna dapat mengisi nilai pada nama petani, tanggal hutang, nominal hutang, dan catatan hutang. Lalu ketika sudah dimasukkan semua

datanya maka pengguna hanya tinggal menyentuh tombol yang bertuliskan “Tambahkan Data Hutang”.

d. Antarmuka Halaman Update Data – Hutang



Gambar 8. Tampilan halaman hutang untuk update data untuk situs web

Untuk halaman update pada hutang dapat dilihat pada gambar 8. Pada gambar tersebut pengguna dapat mengganti nilai pada nama petani, tanggal hutang, nominal hutang, dan catatan hutang. Lalu ketika sudah dimasukkan semua datanya maka pengguna hanya tinggal menyentuh tombol yang bertuliskan “Perbaharui Data Hutang”.

2. Implementasi Antarmuka Aplikasi Perangkat Bergerak

Pada bagian ini akan dijelaskan tentang implementasi antarmuka yang telah dibuat berdasarkan dengan desain antarmuka yang telah dirancang sebelumnya untuk antarmuka pada aplikasi perangkat bergerak.

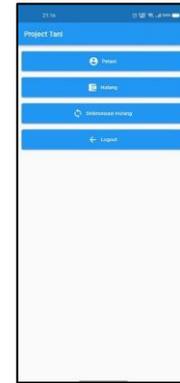
a. Antarmuka Halaman Login



Gambar 9. Tampilan halaman Login untuk perangkat bergerak

Untuk tampilan halaman login pada aplikasi perangkat bergerak dibuat sesuai dengan desain yang ada. Pengguna perlu memasukkan email dan password yang ada dapat dilihat pada gambar 9.

b. Antarmuka Halaman Menu



Gambar 10. Tampilan halaman Menu untuk perangkat bergerak

Untuk halaman menu terdapat empat menu yang disediakan yaitu Petani, Hutang, Sinkronisasi Hutang, dan Log Out. Masing-masing tombol berfungsi untuk berpindah halaman sesuai dengan halaman masing-masing. Ketika disentuh maka akan langsung diarahkan kepada halaman yang bersangkutan.

c. Antarmuka Halaman Hutang



Gambar 11. Tampilan halaman hutang untuk perangkat bergerak

Untuk halaman hutang dapat dilihat pada gambar 11. Pada gambar tersebut dapat dilihat bahwa terdapat nomor data yang ditampilkan nama petani yang melakukan hutang, tanggal dimana transaksi hutang terjadi, nominal hutang, bunga hutang yang ada untuk hutang tersebut, dan nominal total hutang. Serta ditambahkan juga dua tombol yaitu tombol delete dan update yang masing-masing digunakan untuk menghapus data hutang dan melakukan update pada data hutang yang ada.

d. Antarmuka Halaman Insert Hutang



Gambar 12. Tampilan halaman hutang untuk insert data perangkat bergerak

Untuk halaman insert pada hutang dapat dilihat pada gambar 12. Pada gambar tersebut dimana pengguna diharuskan untuk melakukan masukkan pada nama petani, tanggal hutang, nominal hutang, dan catatan hutang. Lalu ketika sudah dimasukkan semua datanya maka pengguna hanya perlu menyentuh tombol yang bertuliskan “Tambah Data Hutang”.

e. Antarmuka Halaman Update Hutang



Gambar 13. Tampilan halaman hutang untuk update data perangkat bergerak

Untuk halaman update pada hutang dapat dilihat pada gambar 13. Pada gambar tersebut pengguna dapat mengganti nilai pada nama petani, tanggal hutang, nominal hutang, dan catatan hutang. Lalu ketika sudah dimasukkan semua datanya maka pengguna hanya tinggal menyentuh tombol yang bertuliskan “Update Data Hutang”.

B. Pembuatan Algoritma

Untuk algoritma yang akan dibuat, sebelumnya akan ada dua kondisi dimana perangkat akan bekerja yaitu:

1. Kondisi terhubung dengan jaringan internet

Untuk kondisi ini, perangkat bergerak akan bekerja dengan cara mengakses basis data cloud secara langsung.

2. Kondisi tidak terhubung dengan jaringan internet

Untuk kondisi ini, perangkat bergerak akan menggunakan basis data lokal pada perangkat bergerak dan membutuhkan sinkronisasi ketika kembali terhubung dengan jaringan internet. Dengan alasan bahwa ketika data telah diubah dalam keadaan tidak terhubung dengan jaringan internet tetapi pada aplikasi situs web telah dilakukan perubahan maka akan terjadi tumpang tindih data pada basis data cloud. Tetapi dengan menggunakan sinkronisasi maka data akan menjadi lebih konsisten dan teratur sehingga mengurangi data yang salah atau kacau pada perangkat bergerak dari keadaan tidak terhubung jaringan internet lalu terhubung jaringan internet.

Untuk rumus pembuatan id pada data yang akan dilakukan sinkronisasi adalah dengan menggunakan rumus sebagai berikut:

$$\text{Tahun} + \text{Bulan} + \text{Tanggal} + \text{Jam} + \text{Menit} + \text{Detik} + \text{MiliDetik}$$

Semuanya akan diubah tipe datanya menjadi tipe data String, sehingga dapat digabungkan. Berikut merupakan contoh implementasi rumus tersebut:

Waktu pembuatan data = 2022-05-24 20:36:54.555556
 Sehingga didapatkan data sebagai berikut:

- a. Tahun : 2022
- b. Bulan : 05
- c. Tanggal : 24
- d. Jam : 20
- e. Menit : 36
- f. Detik : 54
- g. MiliDetik: 555

Hasilnya adalah 20220524203654555 yang digunakan sebagai acuan id untuk satu data yang akan dilakukan sinkronisasi.

C. Implementasi Algoritma

Algoritma yang telah dibuat akan dibuat dengan menggunakan bahasa pemrograman Javascript yang berada pada sisi client dari aplikasi situs web. Algoritma yang digunakan sebagai acuan untuk penelitian penulis adalah SAMD. Pada algoritma ini, pada basis data akan dibuatkan satu tabel tambahan yang berfungsi untuk pengacu antara data yang diubah. Tetapi pada topik peneliti algoritma SAMD yang menggunakan Message Digest Value sebagai pembanding antara data yang diubah dengan yang tidak diubah. Karena sinkronisasi akan berlangsung ketika tidak ada koneksi internet, maka dibutuhkan suatu waktu dimana data tersebut ditambahkan, diubah, atau dihapus. Dari keempat kebutuhan tersebut, dengan mengacu pada algoritma SAMD maka akan ditambahkan dua kolom baru pada tabel yang akan digunakan untuk kebutuhan sinkronisasi yaitu kolom timeStamp dan flag, dimana flag akan berisi nilai “add”, “update”, dan “delete” sedangkan timeStamp akan berisi waktu saat data tersebut dilakukan suatu operasi.

D. Pengujian Aktual

1. Sinkronisasi Normal

a. Sinkronisasi ketika Create Data (TC101)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

b. Sinkronisasi ketika Update Data (TC102)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

c. Sinkronisasi ketika Delete Data (TC103)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

d. Membaca Data ketika Perangkat Tidak Terhubung Internet (TC104)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

e. Mematikan Koneksi Internet ketika Proses Sinkronisasi (TC105)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

2. Sinkronisasi Dengan Pengkondisian Tertentu

a. Sinkronisasi ketika Create Data – Tidak Normal 1 (TC201)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

b. Sinkronisasi ketika Create Data – Tidak Normal 2 (TC202)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

c. Sinkronisasi ketika Update Data – Tidak Normal 1 (TC203)

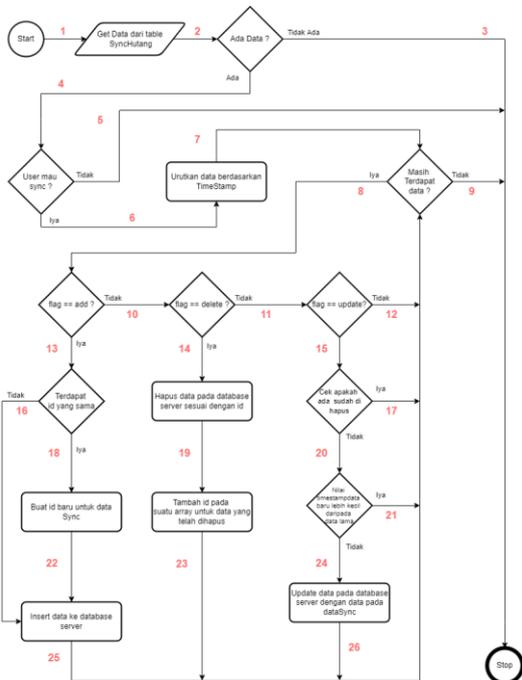
Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

d. Sinkronisasi ketika Update Data – Tidak Normal 2 (TC204)

Dapat berjalan dengan baik hasilnya sesuai dengan apa yang diharapkan oleh penulis.

E. Pengujian Kotak Putih

Pada bagian ini penulis akan melakukan pengujian berdasarkan pada kode program yang telah dibuat oleh penulis. Untuk pengujian pada bagian ini sebelumnya akan dilakukan penentuan jalur yang ada pada diagram alur yang telah dibuat.



Gambar 14. Diagram alur dengan penomoran jalur untuk pengujian kotak putih

Pada gambar 14 dapat dilihat bahwa terdapat 26 jalur atau jalan untuk menuju ke proses selesai. Dengan menggunakan rumus yang sudah disebutkan sebelumnya pada rancangan pengujian maka didapat nilai dari Cyclomatic Complexity adalah sebagai berikut:

$$\begin{aligned}
 M &= E - N + 2P \\
 M &= 26 - 18 + 2.1 \\
 M &= 8 + 2 \\
 M &= 10
 \end{aligned}$$

Sehingga didapatkan bahwa nilai dari Cyclomatic Complexity berdasarkan pada diagram alur adalah 10.

Table 4 Tabel skala pengukuran tingkat resiko untuk nilai CC

Nilai CC	Tipe Prosedur	Tingkat Resiko
1-4	Prosedur sederhana	Rendah
5-10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11-20	Prosedur yang lebih kompleks	Menengah
21-50	Prosedur yang kompleks dan kritis	Tinggi
> 50	Rentan terhadap kesalahan	Sangat Tinggi

Table 5. Tabel skala probabilitas perbaikan untuk nilai CC

Nilai CC	Probabilitas Perbaikan
1-10	5%
20-30	20%
>50	40%
Mendekati 100	60%

Jika dilihat pada tabel 4.2 nilai CC yang didapat pada pengujian yang dilakukan memiliki resiko rendah terhadap jalannya program. Lalu jika dilihat pada tabel 4.3 memiliki probabilitas 5% untuk dilakukan perbaikan pada program yang telah dibuat. Maka akan mendapatkan nilai Path Coverage 100%. Karena nilai CC yang didapatkan pada rumus sebelumnya adalah 10, maka akan terdapat 10 pengujian yang akan dilakukan beserta apa yang menjadi ekspektasi dan masukkan harapan pada tiap pengujian yang dilakukan.

Table 6. Tabel hasil pengujian untuk jalur yang didapatkan berdasarkan nilai CC

No	Urutan Jalur	Data Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
1	1-2-3	Tidak ada data untuk proses sinkronisasi	Tidak terjadi proses sinkronisasi	Sesuai harapan	Valid
2	1-2-4-5	Terdapat data untuk proses sinkronisasi	Tidak terjadi proses sinkronisasi	Sesuai harapan	Valid
3	1-2-4-6-7-8-9	Terdapat data untuk proses sinkronisasi	Kondisi tidak akan terpenuhi karena ketika terdapat data sinkronisasi maka jalur nomor empat akan terpenuhi	Sesuai harapan	Valid

			tetapi jalur nomor 9 tidak akan dapat terpenuhi.		
4	1-2-4-6-7-8-13-16-25-9	Terdapat data untuk proses sinkronisasi dengan flag bernilai "add"	Terjadi proses sinkronisasi	Sesuai harapan	Valid
5	1-2-4-6-7-8-13-18-22-25-9	Terdapat data untuk proses sinkronisasi dengan flag bernilai "add" dan id yang ditambahkan sama dengan id pada basis data <i>cloud</i>	Terjadi proses sinkronisasi	Sesuai harapan	Valid
6	1-2-4-6-7-8-10-14-19-23-9	Terdapat data untuk proses sinkronisasi dengan flag bernilai "delete"	Terjadi proses sinkronisasi	Sesuai harapan	Valid
7	1-2-4-6-7-8-10-11-12	Terdapat data untuk proses sinkronisasi.	Tidak terjadi proses sinkronisasi, kondisi ini tidak akan terpenuhi karena sebelum masuk ke tabel <i>syncHutang</i> , nilai pada kolom flag pasti akan terisi.	Sesuai harapan	Valid
8	1-2-4-6-7-8-10-11-15-17	Terdapat data untuk proses sinkronisasi dengan flag bernilai "update"	Terjadi proses sinkronisasi	Sesuai harapan	Valid

			dengan kondisi data yang ingin dilakukan proses <i>update</i> sudah dihapus.		
9	1-2-4-6-7-8-10-11-15-20-21	Terdapat data untuk proses sinkronisasi dengan flag bernilai "update" dengan kondisi data yang ingin dilakukan proses <i>update</i> tidak dihapus, dengan kondisi timestamp data baru lebih kecil dari data yang sudah ada.	Terjadi proses sinkronisasi	Sesuai harapan	Valid
10	1-2-4-6-7-8-10-11-15-20-24	Terdapat data untuk proses sinkronisasi dengan flag bernilai "update" dengan kondisi data yang ingin dilakukan proses <i>update</i> tidak dihapus, dengan kondisi timestamp data baru lebih besar dari data	Terjadi proses sinkronisasi	Sesuai harapan	Valid

		yang sudah ada.			
--	--	-----------------	--	--	--

***Semua pengujian dilakukan dengan asumsi data sinkronisasi hanya berjumlah 1 yang dapat memenuhi masing-masing jalur pengujian yang ada.**

Berdasarkan pada pengujian yang telah dilakukan pada tabel 4.4 dapat disimpulkan bahwa semua jalur yang ada pada algoritma telah teruji dan memiliki hasil yang valid pada tiap jalur yang dilalui beserta tiap data yang memenuhi tiap jalur yang ada.

F. Pengujian Konsistensi Data

Table 7. Tabel hasil pengujian konsistensi data

Testcase ID	Jumlah Data					Keterangan
	10	50	100	150	200	
TC101	100%	100%	100%	100%	100%	Valid
TC102	100%	100%	100%	100%	100%	Valid
TC103	100%	100%	100%	100%	100%	Valid
TC104	100%	100%	100%	100%	100%	Valid
TC105	100%	100%	100%	100%	100%	Valid
TC201	100%	100%	100%	100%	100%	Valid
TC202	100%	100%	100%	100%	100%	Valid
TC203	100%	100%	100%	100%	100%	Valid
TC204	100%	100%	100%	100%	100%	Valid

Berikut merupakan penjelasan terkait dengan pengujian:

1. TC101

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian dilakukan dengan menggunakan data dengan id yang berbeda antara satu dengan yang lain.

2. TC102

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian dilakukan dengan memasukkan data pada tabel ps_hutang terlebih dahulu, kemudian dilakukan sinkronisasi untuk proses update.

3. TC103

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian dilakukan dengan memasukkan data pada tabel ps_hutang terlebih dahulu, kemudian dilakukan sinkronisasi untuk proses delete.

4. TC104

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian untuk masalah pembacaan agak sedikit berbeda dengan pengujian lainnya karena yang diuji hanya hasil setelah data didapatkan dari basis data cloud lalu disimpan pada basis data lokal pada perangkat bergerak.

5. TC105

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pada saat pengujian koneksi internet yang terhubung sewaktu proses sinkronisasi berjalan dimatikan secara interval 2 detik setelah sinkronisasi berjalan.

6. TC201

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian dilakukan dengan menempatkan data dengan id yang sama pada basis data lokal untuk tabel syncHutang dan basis data cloud untuk tabel hutang.

7. TC202

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%.

8. TC203

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian dilakukan dengan penghapusan data pada tabel ps_hutang pada basis data cloud lalu kemudian dilakukan sinkronisasi untuk melakukan proses update.

9. TC204

Untuk pengujian pada scenario ini, didapatkan hasil akhir bahwa semua data yang disinkronisasikan dari basis data lokal pada perangkat bergerak ke basis data cloud memiliki persentase konsistensi data sebesar 100%. Pengujian dilakukan dengan menggunakan 2 perangkat bergerak yang berbeda dimana masing-masing melakukan perubahan pada data dengan id yang sama. Didapatkan hasil bahwa data dengan nilai timestamp lebih besar yang akan dipilih untuk dilakukan perubahan data pada id tersebut.

G. Analisis dan Pembahasan

Dari berbagai hasil pengujian yang telah dilakukan didapatkan bahwa proses sinkronisasi pada basis data lokal dengan basis data cloud dapat dibangun dan dikerjakan dengan baik. Hal ini terbukti dengan berbagai pengujian yang dilakukan untuk menemukan hasil konsistensi dari algoritma yang dibuat.

Pengujian yang dilakukan untuk pengujian konsistensi data hanya menggunakan sampel data dari interval 10 hingga 200 dengan pembagian 10,50,100,150 dan 200. Dari pengujian tersebut didapatkan 100% tingkat konsistensi data yang dilakukan proses sinkronisasi dengan catatan bahwa pengujian hanya dilakukan berdasarkan pada rancangan pengujian yang telah dirancang sebelumnya.

Hal ini juga didukung dengan pengujian kotak putih yang memiliki tingkat resiko yang rendah dan tingkat probabilitas perbaikan sebesar 5%. Juga berdasarkan pada pengujian yang dilakukan pada semua jalur pada algoritma juga menunjukkan hasil yang valid untuk semua jalur yang dilalui.

Sehingga dapat disimpulkan kelebihan dari sistem dan algoritma yang telah dibangun sebagai berikut:

1. Algoritma dapat melakukan sinkronisasi dari perangkat bergerak ketika kondisi perangkat bergerak tidak terhubung dengan jaringan internet lalu dihubungkan kembali dengan jaringan internet.

2. Berdasarkan dengan pengujian yang dilakukan sistem dapat melakukan sinkronisasi dengan data sampai 200 data dengan konsistensi data sebesar 100%.

Adapun kelemahan sistem dan algoritma yang telah dibangun sebagai berikut:

1. Algoritma terlalu bergantung kepada aplikasi situs web sehingga tidak dapat melakukan sinkronisasi ketika menggunakan perangkat bergerak.

2. Diperlukan fitur backup ketika sudah selesai melakukan sinkronisasi sehingga ketika pengguna menginginkan kembali ke versi sebelumnya, pengguna dapat kembali tanpa perlu memeriksa data secara manual.

V. KESIMPULAN

Berdasarkan dengan penelitian dan pengujian yang telah dilakukan oleh penulis, maka didapatkan kesimpulan sebagai berikut:

1. Algoritma sinkronisasi hanya dapat dijalankan pada aplikasi situs web.

2. Algoritma sinkronisasi memiliki acuan dengan algoritma SAMD dengan menambahkan kolom baru pada tabel sinkronisasi yaitu kolom timestamp dan flag dimana masing-masing berfungsi untuk menjalankannya

3. Algoritma sinkronisasi yang telah dibuat dapat menangani kasus sederhana untuk operasi CRUD pada basis data lokal untuk perangkat bergerak dengan basis data cloud dengan konsistensi data sebesar 100%.

4. Algoritma sinkronisasi yang telah dibuat dapat menangani kasus yang memiliki kasus perluasan yang dapat terjadi dalam kehidupan sehari-hari untuk operasi CRUD pada basis data lokal untuk perangkat bergerak dengan basis data cloud dengan konsistensi data sebesar 100%.

5. Untuk kasus TC204 pada pengujian yang telah dilakukan merupakan kasus yang dapat menyebabkan inkonsistensi pada basis data setelah melakukan proses sinkronisasi, sehingga dibuatlah suatu notifikasi dan suatu riwayat yang berguna untuk melihat proses pengubahan data tersebut. Sehingga ketika ditemukan kesalahan pada riwayat tersebut maka pengguna dapat mengubah data tersebut. Untuk membuat pengguna lebih sadar akan data yang tertimpa juga dibuat suatu notifikasi pada situs web dan pada aplikasi perangkat bergerak.

VI. SARAN

Berdasarkan dengan penelitian dan pengujian yang telah dilakukan oleh penulis, maka didapatkan beberapa kekurangan didalamnya. Adapun saran untuk pengembangan algoritma ini kedepannya yakni sebagai berikut:

1. Menambah fitur backup, yang berguna ketika selesai proses sinkronisasi dan terdapat data yang salah maka dapat dikembalikan ke versi sebelum dilakukannya proses sinkronisasi.

2. Berdasarkan pada kasus TC204 dapat dibuatkan suatu proses pemilihan data diantara data yang diubah ke dalam

suatu data, sehingga pengguna dapat memilih data yang akan digunakan untuk data tersebut.

DAFTAR PUSTAKA

- [1] S. Kemp, "DIGITAL 2021: INDONESIA," 11 February 2021. [Online]. Available: <https://datareportal.com/reports/digital-2021-indonesia>.
- [2] Ş. Kocakoyun, "Developing of Android Mobile Application Using Java and Eclipse.," *International Journal Of Electronics, Mechanical And Mechatronics Engineering*, vol. 7, no. 1, p. 18, 2017.
- [3] N. Hansson dan T. Vidhall, "Effects On Performance And Usability For Cross-Platform Application Development Using React Native," p. 93, 2016.
- [4] R. Payne, *Beginning App Development with Flutter: Create Cross-Platform*, Dallas: Apress, 2019.
- [5] G. Wells, "The Future of iOS Development: Evaluating the," p. 32, 2015.
- [6] W. Wu, "React Native vs Flutter, cross-platform mobile application frameworks," p. 34, 2018.
- [7] Y. D. Setiyawati, R. R. Isnanto dan K. T. Martono, "Pembuatan Aplikasi Antar-Jemput Laundry Berbasis Web Service pada Platform Android," *Jurnal Teknologi dan Sistem Komputer*, pp. 150-162, 2016.
- [8] I. G. N. A. WICAKSANA, "Sinkronisasi Basis Data Sql Dengan Basis Data Nosql Menggunakan Data Adapter Dengan Pendekatan Query Direct Access.," p. 130, 2017.
- [9] D. . K. Pamuji, M. Yunus dan D. W. Widarti, "Implementasi Sinkronisasi Database Berbasis RESTful Web Services pada Aplikasi Presensi," *Journal of Information Technology and Computer Science*, p. 8, 2019.
- [10] M. B. Kekgathetse dan K. Letsholo, "A Survey On Database Synchronization Algorithms For Mobile Device," *Journal of Theoretical and Applied Information Technology*, vol. 86, no. 1, p. 9, 2016.
- [11] S. K. Jha dan S. Sebastian, "An Algorithm To Synchronize The Local Database with Cloud," *International Journal of Innovative Research in Advanced Engineering*, vol. 4, no. 4, pp. 22-27, 2017.
- [12] D. Ratnasari, D. B. Qur'ani dan Apriani, "Sistem Informasi Pencarian Tempat Kos Berbasis Android," *Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi*, p. 14, 2018.
- [13] A. Setiyadi dan T. Harihayati, "PENERAPAN SQLITE PADA APLIKASI PENGATURAN WAKTU UJIAN," p. 6, 2014.
- [14] N. Chatterjee, S. Chakraborty, A. Decosta dan D. A. Nath, "Real-time Communication Application Based on Android Using Google Firebase," *International Journal of Advance Research in*, p. 6, 2018.
- [15] M. E. Khan dan F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *International Journal of Advanced Computer Science and Applications*, p. 4, 2012.
- [16] Jindal, R, "Review Paper On Database Synchronization Between Local And Server," *International Journal of Engineering Sciences and Research Technology*, 2016.
- [17] A. Kaur, A, "Synchronized Algorithm for Database and Image Processing Between Client and Server," *International Journal of Computer Science and Information Technologies*, 2014.