

Pemanfaatan Metode Sinkron-Asinkron Komunikasi Data Dengan *Flutter* Sebagai Solusi Layanan Aplikasi Untuk Daerah Dengan Keterbatasan Sinyal Internet

Agus Hermanto^{a1}, Aditya Nanda Utama^{b2}, Yusrida Muflihah^{a3}, Ardy Januantoro^{a4}, Geri Kusnanto^{a5}

^aDepartemen Sistem Informasi dan Teknologi, Universitas 17 Agustus 1945 Surabaya
Surabaya, Indonesia

¹hermanto_if@untag-sby.ac.id

^b Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia

²6026231036@student.its.ac.id

Abstrak

Dalam era digital yang semakin maju, tantangan pengumpulan data di daerah dengan keterbatasan sinyal internet menjadi semakin mendesak. Infrastruktur yang minim di daerah-daerah terpencil atau terpinggirkan membatasi akses yang konsisten ke internet, mengakibatkan masalah serius dalam efisiensi pengumpulan dan pengiriman data. Ketika sinyal lemah atau tidak ada sama sekali, pengguna rentan kehilangan data, menimbulkan risiko pengulangan pekerjaan, atau mengalami penundaan informasi yang krusial. Untuk mengatasi tantangan ini, aplikasi yang memanfaatkan metode Sinkron-Asinkron telah dikembangkan. Ketika pengguna beroperasi dalam kondisi sinyal terbatas, data yang diinput disimpan secara lokal di perangkat. Hal ini memungkinkan kelancaran pekerjaan tanpa risiko kehilangan data. Namun, penggunaan metode sinkronisasi saja memiliki kelemahan yang perlu diperhatikan. Gangguan pada proses sinkronisasi, baik karena putusnya koneksi internet atau kegagalan server, dapat menyebabkan kehilangan data yang telah disimpan secara lokal. Selain itu, proses sinkronisasi yang lambat juga bisa mengakibatkan penundaan dalam mengakses atau memperbarui informasi terbaru, mengganggu produktivitas dan efisiensi kerja secara keseluruhan. Oleh karena itu, penggunaan kombinasi metode sinkron-asinkron menjadi solusi yang lebih holistik. Dengan memanfaatkan kedua metode secara bersamaan, aplikasi dapat memberikan pengguna pengalaman yang lebih baik dalam mengumpulkan dan mengelola data, terutama di daerah dengan keterbatasan sinyal internet.

Kata kunci: Sinkron-Asinkron, Keterbatasan Sinyal, Data Lokal, *Flutter*

Utilization of Synchronous-Asynchronous Method of Data Communication with *Flutter* as an Application Service Solution for Areas with Limited Internet Signal

Abstract

In an increasingly digital age, there are challenges to be faced when collecting data in areas that are signal-constrained or have limited internet coverage. These areas are constrained by a lack of infrastructure that limits consistent access to the internet. Limited internet signal can cause serious problems in efficiently collecting data and sending it to the server. When the signal is weak or nonexistent, users may not be able to access the server or store the collected data securely. This can result in lost data, repetitive work, or delays in collecting important information. To solve this problem, an application was created that utilizes the Synchronous-Asynchronous method that allows users to continue working and collecting data even in areas with limited internet signal. When users fill out forms or entries in the app with limited signal, the data will be stored locally on the user's device. This allows users to do their work without worrying about data loss due to signal issues. When the internet connection improves, users can start the synchronization process. The data that has been stored offline will be sent to the server and integrated with the existing data. This also allows other users or related systems to access the data in real-time, even if they are in areas with limited internet signal.

Keywords: Synchronous-Asynchronous, Signal Limitations, Local Data, *Flutter*

I. PENDAHULUAN

Dalam era digital yang semakin maju, pengumpulan data menjadi aspek penting dalam berbagai bidang seperti penelitian, survei, pemantauan, dan manajemen informasi. Namun, terdapat tantangan yang harus dihadapi ketika mengumpulkan data di daerah yang terkendala sinyal atau memiliki keterbatasan jaringan internet. Daerah-daerah terpencil atau terpencil sering menghadapi masalah infrastruktur yang membatasi akses yang konsisten ke internet [1].

Keterbatasan sinyal internet dapat menyebabkan masalah serius dalam mengumpulkan data secara efisien dan mengirimkannya ke server [2]. Ketika sinyal lemah atau tidak ada sama sekali, pengguna mungkin tidak dapat mengakses server atau menyimpan data yang dikumpulkan dengan aman. Ini dapat mengakibatkan data yang hilang, pengulangan kerja, atau penundaan dalam pengumpulan informasi yang penting.

Untuk mengatasi masalah tersebut perlu dikembangkan sebuah aplikasi berbasis *flutter* yang memanfaatkan metode Sinkron-Asinkron yang memungkinkan pengguna untuk terus bekerja dan mengumpulkan data walaupun berada di daerah dengan sinyal internet yang terbatas. Selain itu, aplikasi tersebut memiliki kemampuan untuk menyinkronkan data saat koneksi internet tersedia dengan menekan tombol sinkron sekarang, sehingga memastikan bahwa data yang dikumpulkan terintegrasi dengan server dan tersedia untuk akses *real-time* [3].

Sinkron adalah langkah-langkah untuk menyesuaikan atau menyelaraskan data di antara beberapa sumber data yang berbeda agar mereka sejalan atau memiliki kesamaan. Tujuannya adalah memastikan bahwa data yang tersimpan di berbagai tempat atau platform tetap konsisten dan diperbarui secara bersamaan [4].

Asinkron adalah sebuah situasi di mana entitas yang terlibat dalam pertukaran data tidak perlu melakukan interaksi secara langsung atau secara *real-time* [5]. Data dapat disimpan atau dikirimkan secara asinkron, yang berarti mereka dapat diproses pada waktu yang berbeda atau dalam urutan yang berbeda tanpa mempengaruhi ketersediaan atau integritas data. Asinkron data juga digunakan dalam sistem yang memperbarui data secara berkala atau di latar belakang, seperti sinkronisasi data otomatis antara perangkat dan server [6].

Flutter merupakan kerangka kerja sumber terbuka yang dikembangkan oleh Google untuk membangun aplikasi *cross-platform* menggunakan bahasa pemrograman Dart. *Flutter* memungkinkan pengembang untuk membuat antarmuka pengguna yang kaya dan responsif dengan cepat untuk Android, iOS, web, dan platform lainnya dari satu kode sumber [7]. *Flutter* menggunakan pendekatan yang berbeda dibandingkan dengan teknologi *cross-platform* lainnya. Daripada menggunakan komponen antarmuka asli atau *wrapper*, *Flutter* menggunakan *rendering* sendiri, sehingga dapat menciptakan antarmuka yang sepenuhnya konsisten dan responsif di seluruh platform [8]. Ini juga memungkinkan pengembang untuk memiliki kontrol penuh atas tampilan dan perilaku aplikasi mereka.

Berdasarkan latar belakang tersebut. Aplikasi berbasis *Flutter* yang memungkinkan pengguna untuk menginput dan menyimpan data secara *offline*, sambil tetap menyinkronkannya dengan server saat ada koneksi internet, menjadi solusi yang efisien dan efektif. Aplikasi ini membantu mengatasi tantangan pengumpulan data di daerah dengan keterbatasan sinyal atau terkendala jaringan internet, memastikan data yang terkini dan tersedia dengan mudah bagi pengguna dan pihak terkait..

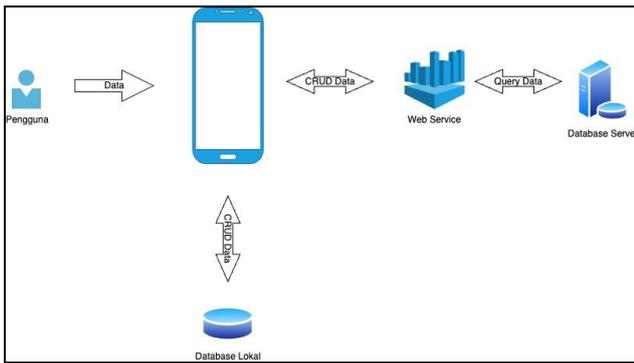
II. METODOLOGI

Dalam pengembangan aplikasi, penggunaan metode sinkron-asinkron dimanfaatkan untuk mengatasi daerah yang memiliki keterbatasan sinyal internet [9]. Metode sinkron digunakan ketika sinyal internet tersedia dan stabil, sehingga aplikasi dapat berkomunikasi secara langsung dengan server atau layanan eksternal untuk memperoleh data atau melakukan tindakan tertentu. Dalam konteks ini, sinkronisasi waktu dan urutan antara permintaan dan respons menjadi penting untuk memastikan data yang akurat dan konsisten.

Namun, di daerah yang memiliki keterbatasan sinyal internet, metode asinkron menjadi solusi yang lebih efektif. Dalam metode ini, aplikasi dapat melakukan tindakan atau mengumpulkan data secara mandiri tanpa harus terus terhubung dengan server atau layanan eksternal. Data yang terkumpul akan disimpan secara lokal dalam aplikasi dan dapat disinkronkan dengan server ketika koneksi internet tersedia.

Penggunaan metode sinkron saja memiliki beberapa kelemahan. Pertama, metode ini sangat tergantung pada koneksi internet yang stabil untuk mengirim dan menerima data secara *real-time*. Jika koneksi internet tidak tersedia atau tidak stabil, proses sinkronisasi akan terganggu, mengakibatkan data yang dikumpulkan tidak dapat dikirim ke server dan berpotensi menyebabkan kehilangan data. Kedua, jika koneksi internet terputus selama proses sinkronisasi, data yang sedang dikirim dapat hilang atau rusak. Ketiga, proses sinkronisasi yang lambat dapat menyebabkan penundaan dalam mengakses atau memperbarui informasi terbaru, mengganggu produktivitas dan efisiensi kerja. Terakhir, selama proses sinkronisasi, beban pada server bisa meningkat secara signifikan jika banyak pengguna melakukan sinkronisasi secara bersamaan, menyebabkan penurunan kinerja server dan memperpanjang waktu sinkronisasi [1].

Dengan memanfaatkan metode sinkron-asinkron dalam pengembangan aplikasi, pengembang dapat menghadapi tantangan daerah dengan keterbatasan sinyal internet. Metode asinkron memungkinkan aplikasi tetap berfungsi dan memberikan pengalaman pengguna yang baik, bahkan ketika koneksi internet tidak stabil atau terputus. Data yang terkumpul secara asinkron dapat disinkronkan dengan menekan tombol sinkron sekarang ketika koneksi internet pulih, sehingga meminimalkan hilangnya data atau gangguan dalam penggunaan aplikasi seperti pada Gambar 1.

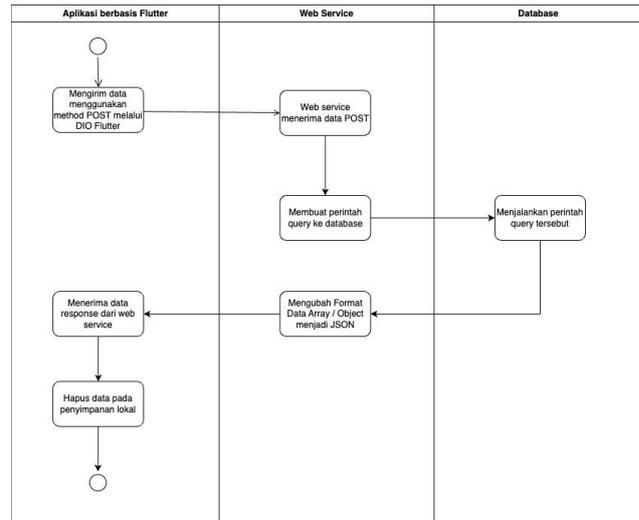


Gambar 1. Diagram Sinkron-Asinkron

A. Sinkronisasi Data Antara Aplikasi Berbasis Flutter Dengan Server

Sistem yang menghubungkan serta dapat melakukan sinkronisasi data antara server merupakan sebuah *RESTful* Web API yang terhubung dengan aplikasi berbasis *flutter*. Hal tersebut diperlukan untuk membuat sistem terpadu [10]. REST bekerja dengan menavigasi melalui tautan HTTP untuk melakukan aktivitas tertentu. Perintah HTTP yang biasanya digunakan dalam REST adalah fungsi GET, POST, PUT atau DELETE [11]. *RESTful* Web API dikembangkan dengan kerangka kerja berbasis Laravel. Laravel adalah *framework* web PHP sumber terbuka yang dikembangkan oleh Taylor Otwell [12]. *Framework* ini dirancang untuk membuat aplikasi web yang menggunakan pola arsitektur *model-view-controller* (MVC) dan didasarkan pada *Symfony* [13].

Pemanfaatan arsitektur *Model View Controller* (MVC) dalam mengimplementasikan *E-Test* melibatkan pembuatan model yang mewakili data dari *database*, tampilan untuk menampilkan data, dan pengontrol untuk menggabungkan keduanya dan mengelola tugas-tugas lain [14]. MVC adalah arsitektur pengembangan yang menggarisbawahi tiga komponen penting, dengan masing-masing komponen memiliki fokus khusus pada perhatian, tanggung jawab, dan logika untuk meningkatkan kinerjanya. Struktur MVC memungkinkan aplikasi untuk mencapai peningkatan keamanan dan penggunaan kembali [15].

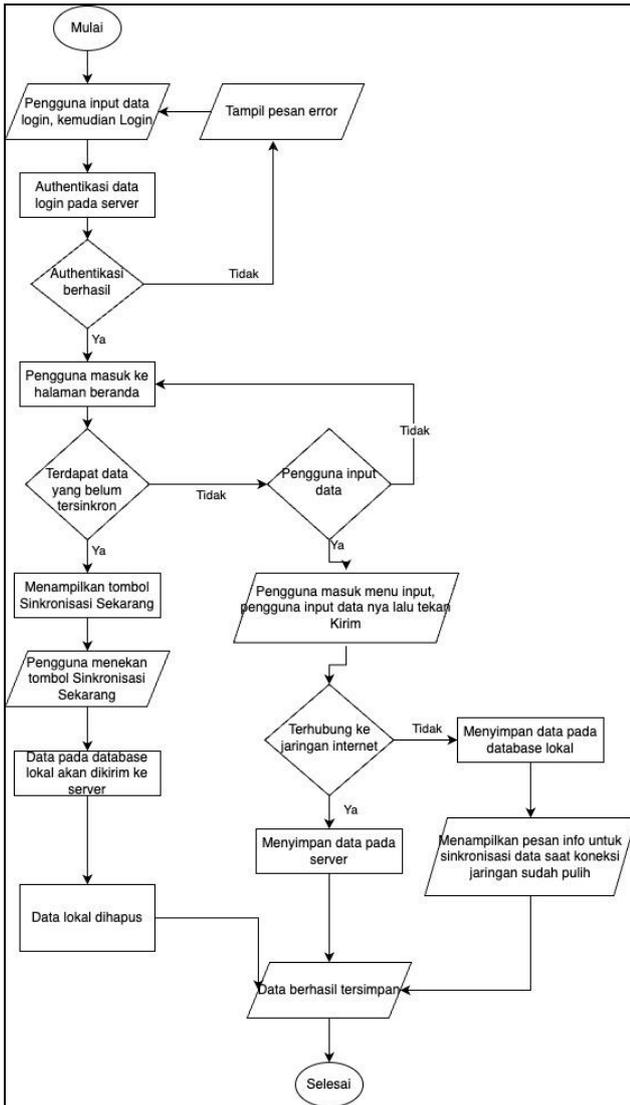


Gambar 2. Sinkronisasi data antara aplikasi berbasis *flutter* dengan server

Aplikasi berinteraksi dengan *web service* dan *database*, seperti yang ditunjukkan pada Gambar 2. Saat aplikasi mengumpulkan dan mengirimkan data, metode POST harus digunakan. Setelah itu, *web service* akan menerima data dari aplikasi dan jika diperlukan data dari *database*, akan membuat perintah *query* sesuai dengan permintaan aplikasi dan bagian *database* akan menjalankan perintah *query* tersebut. Kemudian, *web service* mengubah data dari *database* menjadi format JSON dan kemudian dikirim dalam bentuk *string* melalui protokol HTTP menuju aplikasi. Jika *respon* yang didapat adalah sukses maka aplikasi akan menghapus data lokal yang telah berhasil tersinkronisasi

B. Sinkron-Asinkron Data Antara Aplikasi Dengan Lokal Storage dan Server

Metode Asinkron memanfaatkan penyimpanan data lokal yang berada pada aplikasi berbasis Flutter [16]. Penyimpanan data lokal yang digunakan memanfaatkan RDBMS SQLite. Pada *flutter*, SQLite dapat diimplementasikan menggunakan plugin pihak ketiga yaitu SQFlite. SQFlite adalah paket dalam bahasa pemrograman Dart yang berguna untuk mengakses dan mengelola basis data SQLite di aplikasi Flutter [17]. SQLite adalah sistem manajemen basis data relasional (RDBMS) ringan yang terkenal [18], dan SQFlite menyediakan antarmuka untuk berinteraksi dengan *database* SQLite menggunakan *Flutter*.



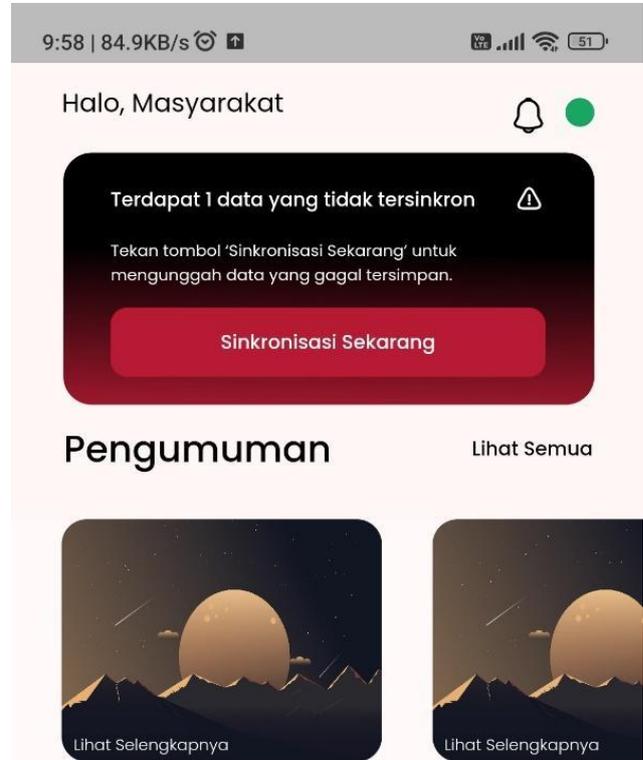
Gambar 3. Sinkron-Asinkron data antara Aplikasi dengan lokal storage dan server

Aplikasi berinteraksi dengan penyimpanan data lokal dan server diilustrasikan pada Gambar 3. Saat pengguna berhasil login, Informasi data yang belum tersinkron akan muncul. Pengguna dapat melakukan sinkronisasi data jika sinyal internet sudah terhubung dengan baik dengan menekan tombol Sinkronisasi Sekarang. Pada proses sinkronisasi data akan dikirim menuju server. Server menerima data kemudian memberikan respon kembali menuju aplikasi. Jika respon sukses maka data pada penyimpanan lokal dihapus. Kemudian ketika pengguna input data, jika sinyal internet saat itu mengalami keterbatasan maka data akan disimpan kedalam penyimpanan lokal sehingga nanti ketika mengunjungi beranda akan ada informasi data yang belum tersinkron. Namun jika kondisi sinyal tersebut adalah baik, maka data akan dikirim langsung menuju server.

III. HASIL DAN PEMBAHASAN

Penerapan aplikasi dengan metode sinkron-asinkron dalam pengembangan aplikasi, pengembang dapat menghadapi tantangan daerah dengan keterbatasan sinyal

internet telah berhasil diimplementasikan. Gambar 4. adalah tampilan beranda dari aplikasi apabila terdapat data yang belum tersinkron (biasanya disebabkan oleh gangguan koneksi saat mengirim data), pengguna dapat melakukan sinkronisasi dengan menekan tombol “Sinkronisasi Sekarang” pada halaman beranda.



Gambar 4. Tampilan beranda dari aplikasi apabila terdapat data yang belum tersinkron

Berikut pada Gambar 5 adalah pseudocode pada saat proses sinkronisasi dari penyimpanan data lokal aplikasi terhadap server.

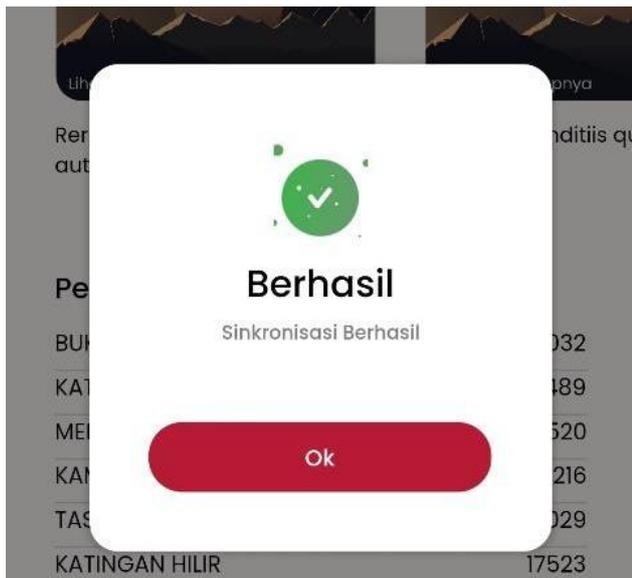
```

1 // validasi model
2 List<KasusSpes> kasusSpes = [];
3
4 // ambil data dari SQLite, lalu tempung data tersebut ke dalam variabel model
5 await MySQLite.instance.getKasusSPM().then((value) async {
6   if(value != null){
7     kasusSpes = value;
8   }
9 }
10
11 // melakukan insert data pada server
12 await Future.forEach(kasusSpes, (element) async {
13   await KasusSPMModel.insertKasusSPM(element).then((value_online) async {
14     if(value_online != value_online) {
15       if(element.status == 1) {
16         // jika status respon yang dikirim server success, maka data pada SQLite
17         await MySQLite.instance.deleteKasusSPM(element.id.toString());
18       } else {
19         customDialogWarningMessage("Tidak dapat sinkron data, silahkan cek kembali datanya", context);
20       }
21     }
22   });
23 });

```

Gambar 5. Pseudocode saat proses sinkronisasi dari penyimpanan data lokal aplikasi terhadap server

Ketika sinkronisasi sukses dilakukan seperti pada Gambar 6, maka informasi sinkronisasi pada beranda akan hilang seperti pada Gambar 7.



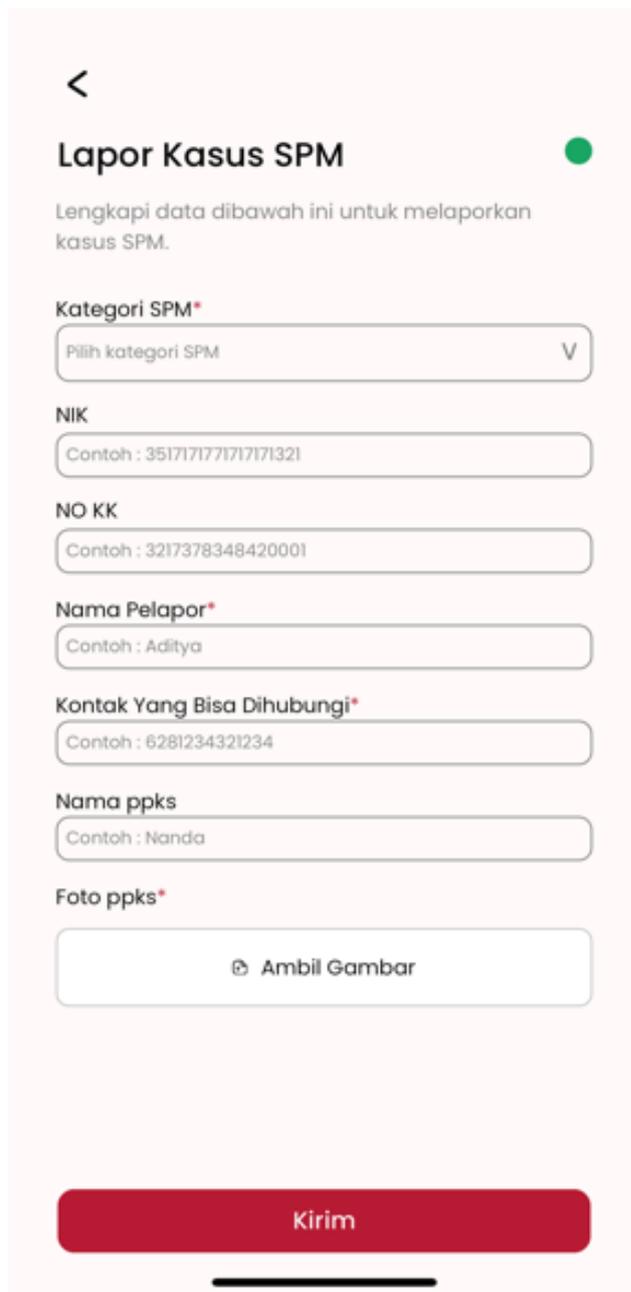
Gambar 6. Sinkronisasi Sukses



Gambar 7. Tampilan beranda dari aplikasi apabila data telah tersinkron

Tampilan pada Gambar 8 adalah *form* untuk menginputkan data pada aplikasi berbasis Flutter. Terdapat indikator ikon warna pada pojok kanan atas, jika berwarna hijau maka sinyal stabil pada Gambar 8, namun jika berwarna merah maka sinyal mengalami kendala pada Gambar 9.

Pengguna dapat mengisi semua form yang telah disediakan, jika semua form data telah diisi, pengguna dapat menekan tombol kirim. Saat tombol kirim ditekan maka data akan diproses untuk dikirim menuju server. Namun saat server tidak merespon karena terkendala atau keterbatasan sinyal internet untuk berkomunikasi, maka data akan disimpan pada penyimpanan lokal serta aplikasi akan menampilkan peringatan seperti pada Gambar 10. Nantinya saat pengguna mengunjungi halaman beranda, maka akan muncul Informasi Sinkronisasi data.



Gambar 8. Indikator Hijau ketika tersedia Sinyal

Menguji kelayakan aplikasi adalah langkah penting dalam pengembangan aplikasi. Dari dua puluh peserta, didapatkan pada Tabel 1 hasil yang sangat memuaskan dengan rata-rata 95%.

TABEL I
UKURAN HURUF UNTUK TABEL

No	Topik Pertanyaan	Jumlah Jawaban Responden					Jumlah Skor	PI
		1	2	3	4	5		
1	Kemudahan dalam menggunakan aplikasi				5	15	95	95%
2	UI yang menarik dan ramah pengguna			1	5	14	93	93%
3	Informasi yang disajikan sederhana untuk dipahami.			2	4	14	92	92%
4	Semua fungsi berfungsi dengan baik.				3	17	97	97%
5	Aplikasi dapat digunakan dan diakses oleh masyarakat umum.				3	17	97	97%
6	Aplikasi yang sesuai dengan perangkat pengguna			1	4	15	94	94%
7	Aplikasi dapat diakses secara offline atau dengan keterbatasan sinyal internet.				1	19	99	99%
Rata-rata								95%

IV. KESIMPULAN

Metode Sinkron-Asinkron berhasil membuat aplikasi berbasis Flutter berfungsi secara *offline* dan mengizinkan sinkronisasi dengan *database* server ketika koneksi internet tersedia. Dengan demikian, pengguna dapat terus melakukan *input* data tanpa memerlukan akses internet. Pengujian aplikasi menunjukkan bahwa selama pengujian, proses *offline* dan sinkronisasi data dinyatakan berhasil dengan hasil yang mencapai 95%.

Untuk pengembangan selanjutnya, dapat ditingkatkan lagi melakukan proses sinkronisasi otomatis saat pengguna terkoneksi ke jaringan internet setelah berhasil *login* pada aplikasi. Dengan demikian, pengguna tidak perlu lagi secara manual menekan tombol untuk memulai sinkronisasi data.

DAFTAR PUSTAKA

[1] D. I. Suhada, D. Delviga, L. Agustina, R. S. Siregar, and M. Mahidin, "Analisis Keterbatasan Akses Jaringan Internet Terkait Pembelajaran Daring selama Pandemi Covid-19 (Studi Kasus

Desan Talun Kondot, Kec. Panombeian Panei, Kab. Simalungun," *J. Pendidik. Tambusai*, vol. 6, no. 1, pp. 256–262, Jan. 2022, doi: 10.31004/jptam.v6i1.2861.

[2] F. Arofiati, "Radio Edukasi Pendidikan UMY: Solusi Pasti Keterbatasan Sinyal Internet," in *Prosiding Seminar Nasional Program Pengabdian Masyarakat*, 2021. Accessed: Jan. 02, 2024. [Online]. Available: <http://download.garuda.kemdikbud.go.id/article.php?article=2586907&val=24333&title=RADIO%20EDUKASI%20PENDIDIKAN%20UMY%20SOLUSI%20PASTI%20KETERBATASAN%20SINYAL%20INTERNET>

[3] D. K. Pamuji, M. Yunus, and D. W. Widarti, "Implementasi Sinkronisasi Database Berbasis RESTful Web Services pada Aplikasi Presensi," *J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, pp. 1–8, 2020, doi: 10.31328/jointecs.v5i1.1190.

[4] A. Phelps and D. Vlachopoulos, "Successful transition to synchronous learning environments in distance education: A research on entry-level synchronous facilitator competencies," *Educ. Inf. Technol.*, vol. 25, no. 3, pp. 1511–1527, May 2020, doi: 10.1007/s10639-019-09989-x.

[5] O. Morin, P. Kelly, and J. Winters, "Writing, Graphic Codes, and Asynchronous Communication," *Top. Cogn. Sci.*, vol. 12, no. 2, pp. 727–743, 2020, doi: 10.1111/tops.12386.

[6] M. Jhala and R. Menon, "Examining the impact of an asynchronous communication platform versus existing communication methods: an observational study," *BMJ Innov.*, p. bmjinnov-2019, 2020.

[7] M. Olsson, *A Comparison of Performance and Looks Between Flutter and Native Applications : When to prefer Flutter over native in mobile application development*. 2020. Accessed: Jan. 02, 2024. [Online]. Available: <https://um.kb.se/resolve?urn=urn:nbn:se:bth-19712>

[8] D. K. Hiuredhy, "Aplikasi Reservasi Ibadah Gereja Mawar Sharon Salatiga Menggunakan Flutter," Thesis, 2022. Accessed: Dec. 20, 2023. [Online]. Available: <https://repository.uksw.edu/handle/123456789/24435>

[9] G. Hendranto *et al.*, "Implementasi Workshop Penyegehan Bagi Dosen dan Sarjana Bidang Telekomunikasi di Luar Jawa dan Indonesia Timur dengan Kombinasi Metode Daring Sinkron dan Asinkron," *Sewagati*, vol. 4, no. 3, pp. 220–226, 2020.

[10] I. A. K. P. Paramitha, D. M. Wiharta, and I. M. A. Suyadnya, "PERANCANGAN DAN IMPLEMENTASI RESTFUL API PADA SISTEM INFORMASI MANAJEMEN DOSEN UNIVERSITAS UDAYANA," *J. SPEKTRUM*, vol. 9, no. 3, pp. 15–23, Sep. 2022, doi: 10.24843/SPEKTRUM.2022.v09.i03.p3.

[11] I. Sukarsa, I. Piarsa, and I. Putra, "Application of MVP Architecture in Developing Android-Based Seminar Ticket Booking Applications," *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 4, pp. 513–520, Jun. 2020, doi: 10.29207/resti.v4i3.1396.

[12] I. Ahmad, E. Suwarni, R. I. Borman, F. Rossi, and Y. Jusman, "Implementation of RESTful API Web Services Architecture in Takeaway Application Development," in *2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, IEEE, 2021, pp. 132–137. Accessed: Jan. 02, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9649679/>

[13] A. Hermanto, R. Koesdijarto, and G. Kusnanto, "Pengabdian Masyarakat di Ponpes At-Tahririyah Pembuatan Web Profil Menggunakan Framework Laravel," *Pros. SENIATI*, vol. 5, no. 1, Art. no. 1, Feb. 2019, doi: 10.36040/seniati.v5i1.461.

[14] A. Ramirez-Noriega, Y. Martinez-Ramirez, J. C. Lizarraga, K. V. Niebla, and J. Soto, "A software tool to generate a Model-View-Controller architecture based on the Entity-Relationship Model," in *2020 8th International Conference in Software Engineering Research and Innovation (CONISOFT)*, IEEE, 2020, pp. 57–63. Accessed: Jan. 02, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9307798/>

[15] S. Ningsih, E. Erdisna, F. Suryana, Y. Desnelita, and S. Andini, "The Application of E-Test On English Subject Using The Model View Controller," *J. RESTI Rekayasa Sist. Dan Teknol. Inf.*, vol. 7, pp. 1040–1048, Sep. 2023, doi: 10.29207/resti.v7i5.5090.

[16] E. Haryanto and M. A. M. Setiawan, "Perancangan Sistem Informasi Pembelajaran Praktikum Online Multiplatform Berbasis Framework Flutter," *Pros. Disem. Has. Penelit.*, 2021, Accessed: Jan. 02, 2024. [Online]. Available: <http://e-journal.janabadra.ac.id/index.php/PSN/article/view/1568>

- [17] H. Huynh, "Vaasa student guide mobile application in Flutter," 2023, Accessed: Jan. 02, 2024. [Online]. Available: <https://www.theseus.fi/bitstream/handle/10024/800119/Hong%20Huynh%20-%201900318%20thesis.pdf?sequence=2>
- [18] S. S. Shimmi, G. Dorai, U. Karabiyik, and S. Aggarwal, "Analysis of iOS SQLite schema evolution for updating forensic data extraction tools," in *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, IEEE, 2020, pp. 1–7. Accessed: Jan. 02, 2024. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9116208/>