Aplikasi *Mobile* Terintegrasi OpenSID dan *Back-End* Laravel untuk Pelayanan Publik di Desa Sukapura, Kec. Dayeuhkolot, Kab. Bandung

1st Mushlih Nur Islam
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
mushlihnur@student.telkomuniversity.ac.id

2nd Burhanuddin Dirgantoro
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
burhanuddin@telkomuniversity.ac.id

3rd Reza Rendian Septiawan
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
zaseptiawan@telkomuniversity.ac.id

***Abstrak* — Penelitian ini berhasil mengembangkan solusi digitalisasi pelayanan publik di Desa Sukapura melalui pembuatan aplikasi *mobile* terintegrasi dengan sistem OpenSID. Aplikasi ini menghadirkan dua fitur utama, yakni pengkinian dan integrasi data penduduk desa, serta digitalisasi seluruh proses pelayanan publik. Pengembangan aplikasi menggunakan *framework* Flutter dan bahasa pemrograman Dart. Pengujian *alpha testing* dan *beta testing* berhasil memverifikasi aplikasi sesuai standar dan spesifikasi yang ditetapkan. Hasil pengujian menunjukkan performa aplikasi yang memuaskan dalam mengolah data dan memberikan pelayanan publik persuratan. Dalam pengembangan *back-end*, digunakan bahasa pemrograman PHP dengan *framework* Laravel, serta API untuk menghubungkan aplikasi dengan *back-end*. Pengujian unit dilakukan dengan Postman untuk memastikan data dari API diolah dengan akurat. Integrasi aplikasi dengan *database* OpenSID memastikan data yang diperlukan tersedia dalam sistem. Secara keseluruhan, aplikasi ini diharapkan memberikan pelayanan publik yang efisien dan terintegrasi di Desa Sukapura, sehingga masyarakat dan perangkat desa dapat dengan mudah mengakses dan melaksanakan pelayanan publik terkhusus pengajuan surat di Desa Sukapura.**

Kata Kunci: Integrasi Data, Sistem OpenSID, Pengujian *Alpha* dan *Beta*, *Back-End*, API, Desa Sukapura.

I. PENDAHULUAN

Penelitian ini menjelaskan mengenai *Capstone Design* yang bertujuan untuk mengembangkan solusi digitalisasi pelayanan publik di Desa Sukapura, Kecamatan Dayeuhkolot, Kabupaten Bandung. Solusi ini berfokus pada pembuatan aplikasi *mobile* yang terintegrasi dengan sistem OpenSID, yang merupakan Sistem Informasi Desa terbuka dan dapat dikembangkan oleh komunitas peduli SID [1]. Aplikasi ini akan memudahkan masyarakat dan perangkat desa dalam mengakses dan melaksanakan pelayanan publik, khususnya terkait pembuatan surat-surat.

Pada ini menjelaskan spesifikasi produk yang akan diusulkan sesuai dengan permasalahan yang telah diidentifikasi. Spesifikasi produk mencakup arsitektur aplikasi, fitur aplikasi, dan spesifikasi teknis untuk pengembangan aplikasi *mobile*. Tujuan dari penulisan dokumen ini adalah untuk menguraikan solusi produk yang akan diusulkan dan menunjukkan verifikasi pengujian aplikasi berdasarkan spesifikasi yang telah ditentukan.

Aplikasi *mobile* ini akan dikembangkan menggunakan *framework* Flutter. Flutter adalah *toolkit* UI portabel Google yang memungkinkan pembuatan aplikasi yang dapat dijalankan di perangkat Android, iOS, web, dan *desktop* dengan menggunakan satu basis kode [2]. Flutter menggunakan bahasa pemrograman Dart untuk pengembangan aplikasi dan menyediakan struktur dan templat awal untuk membuat aplikasi yang efisien [3].

Aplikasi ini memiliki dua fitur utama yang sangat penting. Pertama, aplikasi dapat melakukan pengkinian dan integrasi data penduduk desa, sehingga memudahkan proses pelayanan. Dengan kemampuan ini, data penduduk dapat diperbarui secara *real-time* dan diintegrasikan dalam satu platform, memberikan akses yang lebih efisien dan akurat bagi masyarakat dan perangkat desa. Kedua, aplikasi akan mendigitalisasi seluruh proses pelayanan publik di Desa Sukapura, memungkinkan masyarakat dan perangkat desa untuk mengakses dan melaksanakan semua layanan publik dalam satu aplikasi yang terintegrasi. Digitalisasi ini akan meningkatkan efisiensi dan kualitas pelayanan, serta memberikan kemudahan bagi semua pihak yang terlibat dalam proses pelayanan publik di desa tersebut.

Aplikasi ini memiliki sejumlah spesifikasi yang mendetail untuk memastikan pengembangan yang sesuai dengan kebutuhan. Spesifikasi pertama, yaitu Spesifikasi 1, akan menampilkan data pengguna dan jenis pelayanan publik desa menggunakan sistem *Front-End*. Untuk bagian *Front-End*, aplikasi akan dikembangkan menggunakan framework Flutter, dan sebelumnya akan dibuat *prototype* UI/UX menggunakan Figma untuk mempermudah perancangan antarmuka aplikasi [4]. Selanjutnya, Spesifikasi 2 akan mengolah data penduduk desa melalui sistem *Back-End* yang menggunakan bahasa pemrograman PHP dan *framework* Laravel, serta dilakukan *unit testing* menggunakan Postman untuk memastikan data yang dimasukkan sesuai dengan yang diharapkan [5].

Spesifikasi 3 akan mengintegrasikan aplikasi dengan *database* OpenSID yang menggunakan MySQL, sehingga data yang diperlukan dalam aplikasi sudah tersedia dalam sistem OpenSID dan masyarakat tidak perlu memasukkan data yang sama saat mengajukan surat [6]. Terakhir, Spesifikasi 4 akan memungkinkan masyarakat untuk mengakses berbagai jenis pelayanan publik persuratan melalui *Mobile Apps*, dan pengajuan surat akan dipermudah dengan mencari data pengguna menggunakan NIK yang sudah terdaftar di sistem OpenSID. Semua spesifikasi ini dirancang untuk menghadirkan aplikasi yang efisien, terintegrasi, dan memudahkan proses pelayanan publik di Desa Sukapura.

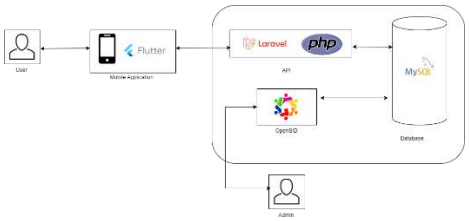
Untuk memastikan aplikasi memenuhi standar dan spesifikasi yang telah ditentukan sebelumnya, kami akan melakukan serangkaian verifikasi dan pengujian. Pertama, pada Verifikasi Spesifikasi 1, kami akan melakukan UI Testing untuk memastikan aplikasi memiliki tampilan yang baik, mudah digunakan, dan sesuai dengan fungsionalitas yang telah direncanakan. Pengujian akan dilakukan secara manual untuk memverifikasi setiap aspek antarmuka aplikasi. Selanjutnya, pada Verifikasi Spesifikasi 2, kami akan melakukan *Unit Testing* menggunakan Postman untuk menguji sistem *Back-End*. Hal ini bertujuan untuk memastikan bahwa aplikasi dapat mengolah data yang dimasukkan oleh pengguna melalui API dengan akurat dan sesuai dengan harapan [5].

Selanjutnya, Verifikasi Spesifikasi 3 akan memfokuskan pada pengintegrasian database desa menggunakan OpenSID. Pengujian ini akan memastikan bahwa aplikasi dapat menampilkan data pengguna dan jenis pelayanan publik persuratan yang sudah terintegrasi dengan sistem OpenSID. Dengan adanya integrasi, masyarakat tidak perlu memasukkan data yang sama saat mengajukan surat, karena data kependudukan mereka sudah tersedia dalam sistem. Terakhir, Verifikasi Spesifikasi 4 akan menguji pengajuan pelayanan publik persuratan melalui aplikasi *Mobile Apps*. Pengujian ini akan memastikan bahwa masyarakat dapat mengakses berbagai jenis pelayanan publik dan memproses pengajuan surat dengan lancar dan sesuai dengan format yang tersedia. Semua verifikasi dan pengujian ini dijalankan untuk memastikan bahwa aplikasi yang dikembangkan dapat memberikan pelayanan publik yang optimal, efisien, dan terintegrasi di Desa Sukapura.

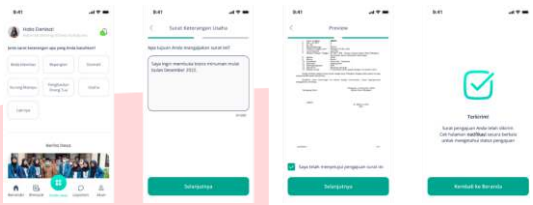
II. KAJIAN TEORI

A. Arsitektur Aplikasi

Dalam dokumen, dijelaskan spesifikasi produk dari usulan solusi digitalisasi pelayanan desa menggunakan aplikasi mobile yang terintegrasi dengan OpenSID. OpenSID adalah Sistem Informasi Desa (SID) yang dirancang agar terbuka dan dapat dikembangkan bersama oleh komunitas peduli SID. Aplikasi *mobile* yang akan dikembangkan terdiri dari *Front-end*, *Back-end*, *Database*, dan Pelayanan Publik [7]. Fitur utama dari aplikasi ini adalah pengkinian dan integrasi data serta digitalisasi pelayanan desa. Aplikasi ini diharapkan dapat mempermudah masyarakat dan perangkat desa dalam mengakses pelayanan desa, serta dapat terintegrasi dengan pemerintah pusat sehingga dapat digunakan secara berkelanjutan [8]. Arsitektur aplikasi ini akan mencakup bagian *Front-end*, *Back-end*, *Database*, dan Pelayanan Publik, seperti ditunjukkan pada Gambar 1.



GAMBAR 1
Arsitektur Aplikasi



GAMBAR 2
Mockup Aplikasi

Berdasarkan gambar *mockup* dan arsitektur rancangan aplikasi, aplikasi ini akan memiliki fitur-fitur seperti pengajuan surat keterangan, pengajuan izin usaha, pengajuan surat keterangan domisili, dan pengajuan surat keterangan kelahiran. Solusi digitalisasi desa juga telah diimplementasikan di beberapa daerah di Indonesia, seperti di Desa Pao-Pao, Kabupaten Sidenreng Rappang, Sulawesi Selatan, yang menerapkan digitalisasi desa melalui SIAP (Sistem Informasi Administrasi Pemerintahan) [9]. Selain itu, penelitian juga menunjukkan bahwa digitalisasi desa dapat meningkatkan percepatan pelayanan publik dan mewujudkan *good governance* [10].

B. Spesifikasi Produk

Dalam mengembangkan bagian *front-end* dari sistem aplikasi digitalisasi pelayanan desa, akan menggunakan *framework* Flutter yang dapat dijalankan pada perangkat Android dan iOS [3]. Flutter adalah *toolkit* UI portabel Google untuk membuat aplikasi yang dikompilasi secara native untuk seluler, web, dan desktop dari satu basis kode. Flutter menggunakan bahasa pemrograman Dart dan gratis serta open source [2]. Sedangkan untuk mengembangkan sistem *back-end*, akan menggunakan bahasa pemrograman PHP dan *framework* Laravel yang bersifat *open source* dan berisi banyak modul dasar yang dapat mengoptimalkan dalam pengembangan web [3][4]. *Framework* Laravel menggunakan struktur MVC (*Model View Controller*) yang memisahkan antara data dan tampilan berdasarkan komponen aplikasi[4]. Untuk pengintegrasian *database* desa, akan menggunakan MySQL yang merupakan *database* server gratis dengan lisensi GNU *General Public License* sehingga dapat dipakai untuk keperluan pribadi atau komersil [11]. Data yang akan digunakan sebagai acuan adalah data yang berasal dari sistem OpenSID karena data yang terdapat dalam sistem tersebut sudah cukup lengkap [1]. Aplikasi OpenSID memiliki fitur terintegrasinya data baik data kependudukan hingga data administrasi desa. Berdasarkan Peraturan Presiden Nomor 39 Tahun 2019 tentang Satu Data Indonesia, pemerintah pusat menimbang untuk memperoleh data yang akurat, mutakhir, terpadu, dapat dipertanggungjawabkan,

mudah diakses, dan dibagipakaikan, diperlukan perbaikan tata kelola data yang dihasilkan oleh pemerintah melalui penyelenggaran Satu Data Indonesia [12]. Dalam aplikasi pelayanan publik yang akan dibuat, terdapat beberapa jenis pelayanan publik seperti persuratan, yang bertujuan untuk memenuhi kebutuhan pelayanan sesuai dengan peraturan perundang undangan bagi setiap warga negara dan penduduk atas barang, jasa, dan/atau pelayanan administratif yang disediakan penyelenggara pelayanan publik.

TABEL 1
Spesifikasi Produk

No.	Hal	Rincian
1.	Menampilkan data pengguna dan jenis pelayanan publik desa menggunakan sistem *front-end.*	Aplikasi dapat menampilkan pelayanan publik yang dapat diajukan oleh masyarakat dan sudah terdapat data masyarakat yang mengajukan pelayanan tersebut sesuai dengan akun yang terdapat di aplikasi.
2.	Mengolah data penduduk desa menggunakan sistem *back-end.*	Aplikasi dapat mengolah data masyarakat yang mengajukan pelayanan publik agar data yang diajukan oleh masyarakat dan data masyarakat dapat digunakan untuk membuat surat yang dibutuhkan oleh masyarakat.
3.	Pengintegrasian *database* desa menggunakan OpenSID.	Aplikasi dapat menyimpan seluruh data yang dibutuhkan oleh aplikasi pada *database* dan dapat terintegrasi oleh layanan OpenSID Desa Sukapura.
4.	Pengajuan Pelayanan Publik Persuratan.	Sistem akan menampilkan beberapa pelayanan publik dengan format surat yang tersedia.

C. Verifikasi

Dalam verifikasi ini, akan membahas rincian dari spesifikasi produk yang dirancang akan diverifikasi dan diukur kinerjanya. Berikut merupakan rincian dari masing-masing spesifikasi yang ada dalam produk.

TABEL 2
Menampilkan Data Pengguna dan Jenis Pelayanan Publik Desa Menggunakan Sistem *Front-End*

Hal	
Rincian	Aplikasi dirancang untuk memiliki tampilan yang baik, mudah digunakan, dan sesuai dengan fungsionalitas aplikasi.
Metode Pengujian	Melakukan *UI Testing* terhadap aplikasi yang dirancang.
Prosedur Pengujian	Melakukan pengujian *UI Testing* secara manual untuk memastikan aplikasi dapat menampilkan informasi dengan baik dan benar.

TABEL 3
Mengolah Data Penduduk Desa menggunakan Sistem *Back-End*

Hal	
Rincian	Aplikasi dapat mengolah Data yang dimasukkan oleh pengguna melalui API yang telah dibuat.
Metode Pengujian	Melakukan *Unit Testing* menggunakan Postman.
Prosedur Pengujian	Pengguna melakukan pengajuan surat lalu nanti akan dicekdi Postman apakah Data yang di input kan sudah sesuai atau tidak.

TABEL 4
Pengintegrasian *Database* Desa menggunakan *OpenSID*

Hal	
Rincian	Aplikasi dapat menampilkan data pengguna dan jenis pelayanan publik persuratan yang sudah terintegrasidengan sistem OpenSID.
Metode Pengujian	Masyarakat dapat membuat surat keterangan tanpa menginput data karena sudah terintegrasi pada sistem.
Prosedur Pengujian	Masyarakat melakukan pengajuan surat, menggunakan *mobile apps,* secara otomatis sistem akan mencari data pengguna menggunakan NIK yang sudah terdaftar. Setelahmasyarakat mengajukan, perangkat desa akan menerima pengajuan melalui sistem OpenSID dan kemudian akan dikirimkan kembali ke *mobile apps* masyarakat.

TABEL 5
Pengajuan Pelayanan Publik Persuratan

Hal	
Rincian	Masyarakat dapat mengakses pelayanan publik melalui *Mobile Apps*.
Metode Pengujian	Masyarakat dapat membuat surat keterangan tanpa harus ke kantor desa karena dapat diurus melalui aplikasi.
Prosedur Pengujian	Masyarakat memilih pelayanan yang ingin di akses melalui *Mobile Apps,* dan melakukan proses pengajuan surathingga menerima surat yang sudah di tanda tangan.

D. Alpha Testing dan *Beta Testing*

Alpha testing adalah jenis pengujian perangkat lunak yang dilakukan pada tahap akhir pengembangan produk sebelum dirilis ke pelanggan eksternal. Pengujian ini dilakukan oleh pengguna atau *end user* pada lingkungan pengembangan yang terkendali. Tujuan dari *alpha testing* adalah untuk memastikan bahwa aplikasi berfungsi dengan baik dan tidak memiliki cacat atau kegagalan aplikasi yang signifikan [13].

Beta testing adalah jenis pengujian perangkat lunak yang dilakukan setelah alpha testing dan melibatkan pengguna nyata atau *end user* untuk menguji produk atau aplikasi dalam lingkungan nyata sebelum dirilis secara resmi ke masyarakat luas [14]. Tujuan dari *beta testing* adalah untuk mengumpulkan umpan balik dari pengguna mengenai kinerja, keandalan, keamanan, dan kesesuaian produk dengan kebutuhan pengguna.

E. Back End

Back-end developer adalah seorang pengembang yang bertanggung jawab untuk mengembangkan sisi server dari sebuah aplikasi atau situs web [15]. Mereka menggunakan bahasa pemrograman untuk mengelola server, *database*, dan logika aplikasi. Tugas utama *back-end developer* adalah mengembangkan, menyimpan data, dan mengatur komunikasi antara *database* situs web atau aplikasi. *Back-end developer* juga harus memastikan bahwa seluruh program di balik *software* bisa berjalan dengan sempurna. Beberapa bahasa pemrograman yang umum digunakan dalam back-end development antara lain PHP, Java, Python, Ruby, dan Node.js [16]. *Back-end developer* bekerja sama dengan front-end developer untuk memberikan pengalaman terbaik bagi pengguna. *Front-end developer* bertanggung jawab untuk mengembangkan tampilan situs web atau aplikasi dan berorientasi dengan kenyamanan pengunjung. Sedangkan *back-end developer* lebih banyak bekerja dengan hal-hal yang terkait server dari sebuah aplikasi atau situs web. Keduanya saling berkaitan dan memegang peran penting dalam pengembangan sebuah web developer maupun aplikasi [17].

Back-end dalam pengembangan aplikasi adalah bagian dari aplikasi yang berada di sisi server atau *server-side* yang bertanggung jawab untuk mengelola logika bisnis, pemrosesan data, dan interaksi dengan database. Dalam proyek "Digitalisasi Pelayanan Publik Berbasis Aplikasi Mobile di Desa Sukapura", *back-end* akan mengolah data penduduk desa yang dimasukkan melalui API dan melakukan integrasi dengan sistem OpenSID untuk mengakses dan menyimpan data yang diperlukan [6].

Penggunaan bahasa pemrograman PHP memungkinkan pengembang menyusun logika aplikasi dan memproses data dengan efisien serta memiliki dukungan dari komunitas pengembang. *Framework* Laravel digunakan untuk mempercepat pembangunan aplikasi dan memberikan struktur yang terorganisir untuk sistem *back-end* dengan keamanan yang baik dan penanganan data yang efisien [5]. Pengujian back-end dilakukan dengan metode Unit Testing menggunakan Postman untuk memastikan data yang dimasukkan oleh pengguna melalui API dapat diolah dengan benar dan sesuai harapan.

F. API

API (*Application Programming Interface*) atau antarmuka pemrograman aplikasi adalah sekumpulan aturan dan protokol yang memungkinkan berbagai aplikasi atau sistem berinteraksi dan saling berkomunikasi [18]. API memungkinkan dua aplikasi yang berbeda untuk saling berbicara satu sama lain dan bertukar data tanpa harus mengetahui detail implementasi internal masing-masing aplikasi. Dalam konteks pengembangan aplikasi, API digunakan sebagai jembatan yang menghubungkan bagian *front-end* (antarmuka pengguna) dengan bagian *back-end* (logika bisnis dan pemrosesan data) [15]. API dapat berfungsi sebagai penghubung antara aplikasi *mobile* dan server, memungkinkan data yang diminta oleh aplikasi mobile dikirimkan dari server dan data yang dimasukkan oleh pengguna melalui aplikasi *mobile* dapat diproses dan disimpan di server.

Dalam proyek "Digitalisasi Pelayanan Publik Berbasis Aplikasi Mobile di Desa Sukapura", API digunakan untuk menghubungkan antara aplikasi mobile yang akan digunakan oleh masyarakat dan perangkat desa dengan sistem *back-end* yang berada di server. API ini akan memungkinkan aplikasi mobile untuk mengirimkan permintaan data ke server dan menerima respon dari server dengan data yang diinginkan. Selain itu, API juga digunakan untuk mengolah data yang dimasukkan oleh pengguna melalui aplikasi *mobile*, seperti permohonan surat-surat pelayanan publik, dan menyimpannya di *database* server untuk diproses lebih lanjut.

III. METODE

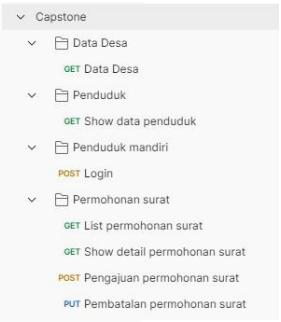
Pengujian yang dilakukan terdiri dari dua metode, yaitu *Alpha Testing* dan *Beta Testing*.

A. *Alpha Testing*

Alpha testing dilakukan untuk menguji *load testing* dan *performance testing* pada API yang telah dibuat. Berikut adalah penjelasan dari setiap langkah yang dilakukan dalam alpha testing:

1. Membuat Postman *Collection*

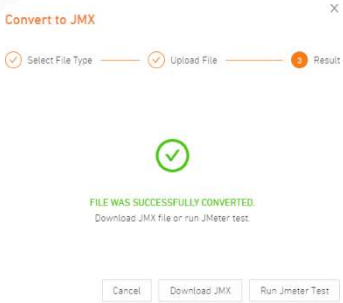
Dalam langkah ini, pengembang membuat sebuah koleksi request yang disimpan dengan baik dalam aplikasi Postman. Koleksi ini berisi berbagai permintaan API yang akan diuji, termasuk POST *Login*, GET Data Penduduk, POST Pengajuan Permohonan Surat, dan PUT Pembatalan Permohonan Surat.



GAMBAR 1
Postman collection

2. Mengekpor Postman *Collection* menjadi JMX

Loadium digunakan untuk mengonversi Postman *Collection* menjadi JMX (Java JMeter XML), yang nantinya akan digunakan dalam proses load testing. Loadium adalah alat yang memungkinkan berbagai macam jenis pengujian dilakukan.

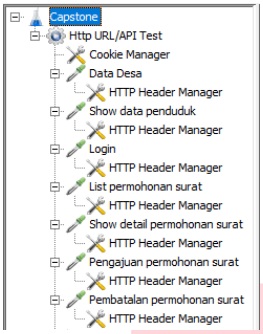


GAMBAR 2
Proses konversi *Postman collection* ke JMX

3. Mengimpor *file* JMX ke Apache JMeter

Hasil konversi Postman *Collection* menjadi JMX kemudian diimpor ke Apache JMeter. Dalam Apache JMeter,

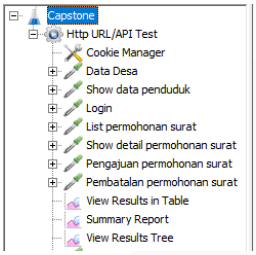
Postman *Collection* diubah menjadi *Test Plan* dan *Threads*. *Threads* ini berisi beberapa sampler yang akan menguji *endpoint* yang telah dikembangkan. Setiap *sampler* harus memiliki *HTTP Request* dan *HTTP Header Manager* untuk mengatur permintaan dan *header* yang sesuai.



GAMBAR 3
Hasil *JMX di Apache JMeter*

4. Menambahkan *listener* dalam *threads* yang sama

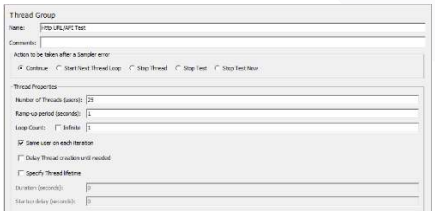
Listener ditambahkan dalam *threads* yang ada untuk menampilkan hasil pengujian. Beberapa jenis *listener* yang digunakan adalah *View Result in Table*, *Summary Report*, dan *View Result Tree*. *Listener* membantu pengembang untuk memahami hasil pengujian dengan lebih baik.



GAMBAR 4
Listener dalam *Thread*

5. Mengatur variabel pengujian dalam *Thread Properties*

Dalam langkah ini, pengembang mengatur variabel pengujian seperti *Number of Threads (Users)*, *Ramp-up period (second)*, dan *Loop count*. *Number of Threads* mengatur jumlah pengguna virtual yang digunakan dalam pengujian, *Ramp-up period* mengatur kecepatan penambahan pengguna virtual ke dalam pengujian, dan *Loop count* mengatur jumlah perulangan skenario pengujian oleh setiap pengguna.



GAMBAR 5
Konfigurasi *thread group* untuk pengujian

6. Menjalankan salah satu *HTTP request* yang ingin diuji

Pengembang menjalankan *HTTP Request* yang ingin diuji untuk endpoint tertentu, misalnya, *endpoint* untuk melakukan *login*. Hasil dari permintaan ini akan menjadi dasar analisis performa API.

GAMBAR 6
HTTP request POST *Login*

7. Melihat hasil pengujian di listener yang telah dibuat

Hasil dari pengujian ditampilkan dalam *listener* yang telah ditambahkan. Dari *listener*, pengembang dapat menganalisis performa API, termasuk melihat jumlah *request* yang berhasil dan gagal, waktu respon, dan performa keseluruhan dari API yang diuji.

GAMBAR 7
Hasil *View Result Tree* pengujian *GET* Data Desa

B. *Beta Testing*

Penelitian ini bertujuan untuk melakukan beta testing terhadap Aplikasi *Mobile* Desa Sukapura dengan fokus pada *user acceptance testing*. Pengujian dilakukan menggunakan *Google Form* untuk mengumpulkan tanggapan dari partisipan yang merupakan masyarakat Desa Sukapura. Berikut adalah langkah-langkah yang diikuti dalam penelitian ini:

1. Membuat Pertanyaan di *Google Form*.

Pertama-tama, kuisoner untuk beta testing dibuat menggunakan Google Form. Kuisoner ini berisi pertanyaan-pertanyaan yang akan diajukan kepada partisipan untuk mengumpulkan tanggapan mereka terkait Aplikasi *Mobile* Desa Sukapura.

2. Mencari Partisipan untuk Pengujian

Partisipan untuk *beta testing* diambil dari masyarakat Desa Sukapura. Target jumlah partisipan yang dibutuhkan adalah 30 orang, sesuai dengan rekomendasi uji validitas dari Sugiono (2009). Uji validitas dilakukan terhadap 30 responden karena hasil uji mendekati kurva normal.

3. Partisipan Mencoba Aplikasi *Mobile* Desa Sukapura

Setelah berhasil mendapatkan partisipan, mereka diminta untuk mengunjungi Aplikasi *Mobile* Desa Sukapura dan menjelajahi semua fitur yang tersedia. Partisipan diarahkan untuk melakukan berbagai aktivitas, mulai dari *login* hingga mencoba mengajukan surat melalui aplikasi. Tujuannya adalah agar partisipan dapat merasakan pengalaman penggunaan aplikasi secara menyeluruh.

4. Mengisi *Google Form*

Setelah mengalami pengalaman menggunakan aplikasi, partisipan diminta untuk mengisi kuisoner yang telah disediakan dalam *Google Form*. Kuisoner ini berisi 27 pertanyaan terkait Aplikasi *Mobile* Desa Sukapura dan

meminta masukan dari partisipan terkait pengalaman dan persepsi mereka terhadap aplikasi.

5. Melihat Hasil Melalui *Google Form* atau *Spreadsheet*

Hasil dari pengisian kuisoner oleh partisipan otomatis terinput pada *Google Form* atau *spreadsheet* yang terhubung. Dari sana, data dapat dilihat dan dianalisis untuk mengevaluasi tanggapan partisipan terhadap aplikasi.

6. Menentukan Reliabilitas Pertanyaan Menggunakan Metode *Cronbach Alpha*

Untuk mengukur reliabilitas pertanyaan dalam kuisoner, digunakan metode *Cronbach Alpha*. Metode ini akan menghitung sejauh mana pertanyaan-pertanyaan tersebut dapat diandalkan dalam mengumpulkan data yang konsisten. Koefisien reliabilitas (r11) dihitung dengan menggunakan rumus *Cronbach Alpha*. Rumusnya sebagai berikut:

$$r_{11} = \left[\frac{k}{k-1}\right]\left[1 - \frac{\sum \sigma_b^2}{\sigma_t^2}\right] \quad (1)$$

Keterangan:

r_{11} = Koefisien reliabilitas instrumen (total tes).

k = Jumlah butir pertanyaan

$\sum \sigma_b^2$= Jumlah varian butir

σ_t^2 = Varian skor total

Hasil dari beta testing akan memberikan wawasan yang berharga tentang bagaimana aplikasi dapat diterima oleh pengguna potensial.

IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian

Hasil Analisis Pengujian *Alpha Testing* dan *Beta Testing*:

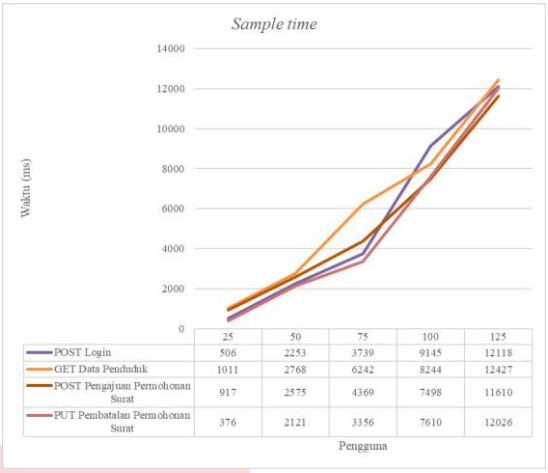
1. *Alpha Testing*

Alpha testing dilakukan untuk menguji *load testing* dan *performance testing* pada API yang telah dibuat. Berikut adalah hasil analisis dari setiap parameter yang diukur dalam *alpha testing*. Dalam pengujian API yang telah dibuat, dari masing-masing *endpoint* memiliki *send bytes* yang berbeda dan ini memberikan beban yang berbeda kepada server untuk masing-masing *endpoint*. Send bytes menunjukkan jumlah total bytes yang dikirimkan ke server melalui *request*. *Endpoint* POST Pengajuan Permohonan Surat mengirimkan *bytes* terbanyak, yang berarti beban yang diterima oleh server lebih besar.

TABEL 1
Tabel *Send Bytes endpoint* Pengujian

No.	*endpoint*	*Send Bytes*
1	POST *Login*	245
2	GET Data Penduduk	198
3	POST Pengajuan Permohonan Surat	789
4	PUT Pembatalan Permohonan Surat	293

a. Sample Time



	25	50	75	100	125
POST Login	506	2253	3739	9145	12118
GET Data Penduduk	1011	2768	6242	8244	12427
POST Pengajuan Permohonan Surat	917	2575	4369	7498	11610
PUT Pembatalan Permohonan Surat	376	2121	3356	7610	12026

GAMBAR 8
Grafik *sample time* Pengujian

Dari grafik *sample time*, dapat dilihat bahwa semakin banyak pengguna virtual yang mengirimkan permintaan, maka semakin lama waktu yang dibutuhkan untuk menyelesaikan permintaan dan menerima tanggapan dari server. *Endpoint* GET Data Penduduk memiliki waktu sample time yang lebih besar dibandingkan dengan *endpoint* lainnya.

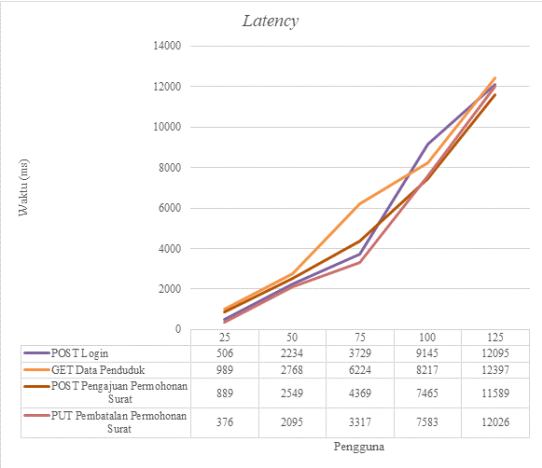
b. Error

	25	50	75	100	125
POST Login	0,00%	0,00%	0,00%	14,00%	28,80%
GET Data Penduduk	0,00%	0,00%	4,00%	25,00%	34,40%
POST Pengajuan Permohonan Surat	0,00%	0,00%	18,67%	22,00%	28,00%
PUT Pembatalan Permohonan Surat	0,00%	0,00%	5,33%	15,00%	35,20%

GAMBAR 9
Grafik *error* Pengujian

Grafik *error* menunjukkan bahwa semakin banyak pengguna virtual yang mengirimkan permintaan, maka semakin tinggi tingkat eror yang terjadi selama pengujian. Eror yang terjadi bisa berkaitan dengan kesalahan koneksi, kesalahan server, atau kesalahan dalam mengirimkan request atau *response*. *Endpoint* POST *Login* memiliki tingkat eror yang paling kecil, sedangkan endpoint GET Data Penduduk, POST Pengajuan Permohonan Surat, dan PUT Pembatalan Permohonan Surat mengalami eror pada pengujian dengan jumlah pengguna 75.

c. Latency

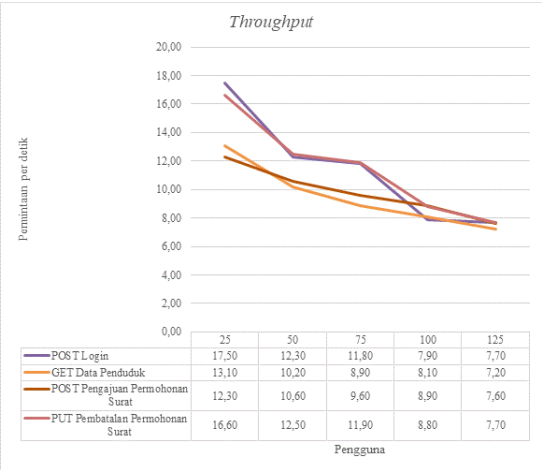


	25	50	75	100	125
POST Login	506	2234	3729	9145	12095
GET Data Penduduk	989	2768	6224	8217	12397
POST Pengajuan Permohonan Surat	889	2549	4369	7465	11589
PUT Pembatalan Permohonan Surat	376	2095	3317	7583	12026

GAMBAR 10
Grafik *letancy* Pengujian

Grafik *latency* menunjukkan waktu yang diperlukan untuk mengirim paket data dari pengirim ke penerima dan sebaliknya. Semakin banyak pengguna yang mengirimkan *request*, maka semakin lama waktu yang dibutuhkan oleh data untuk sampai ke server. *Endpoint* GET Data Penduduk memiliki *latency* yang lebih besar dibandingkan dengan *endpoint* lainnya.

d. Throughput



	25	50	75	100	125
POST Login	17,50	12,30	11,80	7,90	7,70
GET Data Penduduk	13,10	10,20	8,90	8,10	7,20
POST Pengajuan Permohonan Surat	12,30	10,60	9,60	8,90	7,60
PUT Pembatalan Permohonan Surat	16,60	12,50	11,90	8,80	7,70

GAMBAR 11
Grafik *Throughput* Pengujian

Grafik *throughput* menunjukkan jumlah total tugas yang dapat dilakukan dalam satu unit waktu tertentu. *Endpoint* PUT Pembatalan Permohonan Surat merupakan endpoint yang paling banyak dikerjakan dalam satu detik, sedangkan *endpoint* GET Data Penduduk adalah endpoint yang paling sedikit dikerjakan.

2. Beta Testing

Beta testing dilakukan untuk menguji *user acceptance testing* menggunakan *Google Form*. Berikut adalah hasil analisis dari *beta testing*:

a. Validitas Kuisioner

$$r_{11} = \left[\frac{k}{k-1}\right]\left[1 - \frac{\sum \sigma_b^2}{\sigma_t^2}\right] \quad (2)$$

$$r_{11} = \left[\frac{5}{5-1}\right]\left[1 - \frac{2{,}17}{5{,}74}\right]$$

$$r_{11} = [1{,}25][0{,}62]$$

$$r_{11} = [0{,}77]$$

Koefisien reliabilitas (r11) dari kuesioner *beta testing* adalah 0,77, yang lebih dari 0,7 yang merupakan nilai minimal suatu kuesioner dianggap reliabel. Ini menunjukkan bahwa kuesioner yang digunakan dalam beta testing memiliki tingkat validitas yang baik untuk mengumpulkan masukan dan umpan balik dari para partisipan.

Hasil Pengujian dari metode *alpha testing* dan *beta testing* memberikan gambaran tentang performa sistem dan penerimaan pengguna terhadap Aplikasi *Mobile* Desa Sukapura. Hasil pengujian terdiri dari dua metode, yaitu *alpha testing* dan *beta testing*. Dalam *alpha testing*, dilakukan pengujian terhadap empat endpoint pada API dengan parameter *sample time*, *error*, *latency*, dan *throughput*. *Endpoint* POST *Login* memiliki *send bytes* sebesar 245 *bytes*, dengan hasil pengujian baik dan tingkat eror kurang dari 30% saat 100 pengguna.

Endpoint GET Data Penduduk memiliki *send bytes* 198 *bytes*, dengan hasil pengujian kurang baik, tingkat eror 25% saat 100 pengguna, dan 34,4% saat 125 pengguna. *Endpoint* POST Pengajuan Permohonan Surat memiliki *send bytes* 789 *bytes*, hasil pengujian cukup baik, dan tingkat eror 18,67% saat 75 pengguna. *Endpoint* PUT Pembatalan Permohonan Surat memiliki *send bytes* 293 *bytes*, dengan hasil pengujian terbaik dan tingkat eror 35% saat 125 pengguna. Selanjutnya, pada *beta testing* menggunakan kuisoner *user acceptance test* dengan koefisien reliabilitas 0,77, hasilnya memuaskan. Secara keseluruhan, hasil pengujian *alpha* dan *beta* menunjukkan performa aplikasi yang cukup baik.

V. KESIMPULAN

Penelitian ini berhasil mengembangkan solusi digitalisasi pelayanan publik di Desa Sukapura melalui pembuatan aplikasi *mobile* yang terintegrasi dengan sistem OpenSID. Pengembangan *backend* menggunakan bahasa pemrograman PHP dengan *framework* Laravel dan API digunakan untuk menghubungkan aplikasi dengan *backend*. Pengujian unit dilakukan menggunakan Postman untuk memastikan data yang dimasukkan oleh pengguna melalui API diolah dengan akurat. Integrasi aplikasi dengan database OpenSID memastikan data yang diperlukan tersedia dalam sistem. Secara keseluruhan, hasil pengujian *alpha* dan *beta* menunjukkan aplikasi berhasil mencapai performa yang baik dalam mengolah data dan mengintegrasikan dengan database desa. Aplikasi ini diharapkan memberikan pelayanan publik yang efisien dan terintegrasi di Desa Sukapura, memberikan kemudahan bagi masyarakat dan perangkat desa dalam mengakses dan melaksanakan pelayanan publik.

Hasil Pengujian dari *alpha testing* menunjukkan performa aplikasi yang cukup baik dalam mengolah data dan memberikan pelayanan publik persuratan. Pengujian ini menguji empat *endpoint* pada API dan menunjukkan hasil yang baik untuk dua *endpoint*, sedangkan dua lainnya memiliki tingkat eror yang masih dapat diterima.

Dengan hasil analisis pengujian yang telah dilakukan terhadap kedua metode yaitu *alpha testing* dan *beta testing*

menghasilkan hasil yang baik. Dari keempat *endpoint* yang telah dibuat, seluruh *endpoint* memiliki nilai eror kurang dari 40% pada pengujian 125 pengguna. Sedangkan, pada metode *beta testing*, hasil dari perhitungan menggunakan metode *Cronbach alpha* bernilai 0,77 yang ini telah melebihi minimal suatu kuesioner dianggap reliabel.

REFERENSI

[1] Komunitas Peduli SID, "Tanya Jawab OpenSID vs SID CRI," *Komunitas Peduli SID*, Dec. 14, 2021. [Online]. Available: https://github.com/OpenSID/OpenSID/wiki/Tanya-Jawab-OpenSID-vs-SID- CRI. [Diakses 14 Desember 2022].

[2] Flutter Developer, "FAQ Flutter," *Flutter*, Dec. 04, 2018. [Online]. Available: https://docs.flutter.dev/resources/faq#what-is-flutter. [Diakses 18 Desember 2022].

[3] R. R. D. S. Ryan Taruna, Umar Ali Ahmad, "Perancangan FrontendUntuk Aplikasi Peer-To-PeerLending''Salur'' Berbasis Android Menggunakan Flutter," *e-Proceeding Eng.*, vol. 9, No. 3, p. 1195, 2022.

[4] A. Josi, "Penerapan Metode Prototiping Dalam Pembangunan Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang)," Jun. 2017. [Online]. Available: http://download.garuda.kemdikbud.go.id/article.php?article=1121983&val=13036&title=PENERAPAN%2520METODE%2520PROTOTIPING%2520DALAM%2520PEMBANGUNAN%2520WEBSITE%2520DESA%2520STUDI%2520KASUS%2520DESA%2520SUGIHAN%2520KECAMATAN%2520RAMBANG. [Diakses%0918 Desember 2022].

[5] Laravel Developer, "Installation Laravel," *Laravel*, Jun. 11, 2011. [Online]. Available: https://laravel.com/docs/9.x#why-laravel. [Diakses 15 Desember 2022].

[6] M. A. Kasyfurrahman, "Implementasi Framework Laravel 9 dalam Pengembangan Sistem Manajemen Pengunjung di PT. Cilegon Fabricators Menggunakan Metode MVC," 2022. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/183417/slug/implementasi-framework-laravel-9-dalam-pengembangan-sistem-manajemen-pengunjung-di-pt-cilegon-fabricators-menggunakan-metode-mvc.html. [Diakses 18 Desember 2022].

[7] H. B. Irland Fardani, Gina Puspitasari Rochman, Lely Syiddatul Akliyah, "DIGITALISASI DESA DI DESA CIKOLE LEMBANG," *J. STIEM*, 2021, [Online]. Available: https://journal.stiem.ac.id/index.php/resona/article/download/806/496

[8] Adminpanda, "Pelayanan Desa Online: Kemudahan dalam Mengakses Layanan Desa," May 26, 2023. [Online]. Available: https://www.panda.id/pelayanan-desa-online/. [Diakses 26 Mei 2023]

[9] Leski Rizkinaswara, "Desa Pao-Pao Terapkan Digitalisasi Desa Melalui SIAP," *APTIKA KOMINFO*, Sep. 18, 2020. [Online]. Available: https://aptika.kominfo.go.id/2020/09/desa-pao-pao-terapkan-digitalisasi-desa-melalui-siap/. [Diakses 26 Juli 2023].

[10] Kusroh Lailiyah, "DIGITALISASI DESA SEBAGAI UPAYA PERCEPATAN PELAYANAN PUBLIK DALAM MEWUJUDKAN GOOD GOVERNANCE," *RISTEK J. Riset, Inov. dan Teknol.*, vol. 6 No. 2, pp. 26–34, 2022, [Online]. Available: https://ojs.batangkab.go.id/index.php/ristek/article/download/112/214/210

[11] A. D. Febriyanto, "Sistem Keamanan Data Pada IoT Berbasis MQTT Dan Database MySQL Menggunakan Metode RSA," 2022. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/181896/slug/sistem%0A-keamanan-data-pada-iot-berbasis-mqtt-dan-database-mysql-menggunakan- metode-rsa.html. [Diakses 18 Desember 2022].%0A

[12] Pemerintah Pusat, "Peraturan Presiden (Perpres) tentang Satu Data Indonesia," Jun. 12, 2019. [Online]. Available: https://peraturan.bpk.go.id/Home/Details/108813/perpres-no-39-tahun-2019. [Diakses 20 November 2022].

[13] M. P. M. W. Andre Antolis, Andhika Ramadhan Putra, Farabela Putri D. H., Hany Alexandra Philemon, "Alpha dan Beta Testing," *Binus University*, Jun. 30, 2020. [Online]. Available: https://socs.binus.ac.id/2020/06/30/alpha-dan-beta-testing/. [Diakses 25 Juli 2023]

[14] Anggi, "Beta Testing Adalah: Pengertian dan Bedanya dengan Alpha Testing," *Accurate*, Mar. 08, 2021. [Online]. Available: https://accurate.id/marketing-manajemen/beta-testing-adalah/. [Diakses 25 Juli 2023]

[15] Gulman Azkiya, "Mengenal Front End dan Back End Developer: Definisi, Tugas, Skill yang Dibutuhkan, Hingga Perbedaannya," *Skill Academy*, Dec. 01, 2021. [Online]. Available: https://blog.skillacademy.com/perbedaan-front-end-dan-back-end. [Diakses 26 Juli 2023].

[16] AdminDicoding, "Apa itu Back-End dan Back-End Developer?," *Dicoding Indonesia*, May 08, 2021. [Online]. Available: https://www.dicoding.com/blog/apa-itu-back-end/. [Diakses 25 Juli 2023]

[17] Abrar Firdiansyah, "Back End Developer: Lingkup Kerja, Skill, dan Tanggung Jawabnya," *Glints*, Jul. 29, 2022. [Online]. Available: https://glints.com/id/lowongan/pekerjaan-back-end-developer/. [Diakses 25 Juli 2023]

[18] F. D. Muqorobin, M. M., Anggraini, A., Rahmawati, A. D., Yohanes, D., & Ifkarina, "Pengaruh Open Banking berbasis Open API terhadap Eksistensi Perbankan," *MAKSIMUM Media Akunt. Univ. Muhammadiyah Semarang*, vol. 11 No.2, pp. 75–84,

2021.

[1] Komunitas Peduli SID, "Tanya Jawab OpenSID vs SID CRI," *Komunitas Peduli SID*, Dec. 14, 2021. [Online]. Available: https://github.com/OpenSID/OpenSID/wiki/Tanya-Jawab-OpenSID-vs-SID- CRI. [Diakses 14 Desember 2022].

[2] Flutter Developer, "FAQ Flutter," *Flutter*, Dec. 04, 2018. [Online]. Available: https://docs.flutter.dev/resources/faq#what-is-flutter. [Diakses 18 Desember 2022].

[3] R. R. D. S. Ryan Taruna, Umar Ali Ahmad, "Perancangan FrontendUntuk Aplikasi Peer-To-PeerLending''Salur'' Berbasis Android Menggunakan Flutter," *e-Proceeding Eng.*, vol. 9, No. 3, p. 1195, 2022.

[4] A. Josi, "Penerapan Metode Prototiping Dalam Pembangunan Website Desa (Studi Kasus Desa Sugihan Kecamatan Rambang)," Jun. 2017. [Online]. Available: http://download.garuda.kemdikbud.go.id/article.php?article=1121983&val=13036&title=PENERAPAN%2520METODE%2520PROTOTIPING%2520DALAM%2520PEMBANGUNAN%2520WEBSITE%2520DESA%2520STUDI%2520KASUS%2520DESA%2520SUGIHAN%2520KECAMATAN%2520RAMBANG. [Diakses%0918 Desember 2022].

[5] Laravel Developer, "Installation Laravel," *Laravel*, Jun. 11, 2011. [Online]. Available: https://laravel.com/docs/9.x#why-laravel. [Diakses 15 Desember 2022].

[6] M. A. Kasyfurrahman, "Implementasi Framework Laravel 9 dalam Pengembangan Sistem Manajemen Pengunjung di PT. Cilegon Fabricators Menggunakan Metode MVC," 2022. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/183417/slug/imple mentasi-framework-laravel-9-dalam-pengembangan-sistem-manajemen-pengunjung-di-pt-cilegon-fabricators-menggunakan-metode-mvc.html. [Diakses 18 Desember 2022].

[7] H. B. Irland Fardani, Gina Puspitasari Rochman, Lely Syiddatul Akliyah, "DIGITALISASI DESA DI DESA CIKOLE LEMBANG," *J. STIEM*, 2021, [Online]. Available: https://journal.stiem.ac.id/index.php/resona/article/download/806/496

[8] Adminpanda, "Pelayanan Desa Online: Kemudahan dalam Mengakses Layanan Desa," May 26, 2023. [Online]. Available: https://www.panda.id/pelayanan-desa-online/. [Diakses 26 Mei 2023]

[9] Leski Rizkinaswara, "Desa Pao-Pao Terapkan Digitalisasi Desa Melalui SIAP," *APTIKA KOMINFO*, Sep. 18, 2020. [Online]. Available: https://aptika.kominfo.go.id/2020/09/desa-pao-pao-terapkan-digitalisasi-desa-melalui-siap/. [Diakses 26 Juli 2023].

[10] Kusroh Lailiyah, "DIGITALISASI DESA SEBAGAI UPAYA PERCEPATAN PELAYANAN PUBLIK DALAM MEWUJUDKAN GOOD GOVERNANCE," *RISTEK J. Riset, Inov. dan Teknol.*, vol. 6 No. 2, pp. 26–34, 2022, [Online]. Available: https://ojs.batangkab.go.id/index.php/ristek/article/download/112/214/210

[11] A. D. Febriyanto, "Sistem Keamanan Data Pada IoT Berbasis MQTT Dan Database MySQL Menggunakan Metode RSA," 2022. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/181896/slug/sistem%0A-keamanan-data-pada-iot-berbasis-mqtt-dan-database-mysql-menggunakan- metode-rsa.html. [Diakses 18 Desember 2022].%0A

[12] Pemerintah Pusat, "Peraturan Presiden (Perpres) tentang Satu Data Indonesia," Jun. 12, 2019. [Online]. Available: https://peraturan.bpk.go.id/Home/Details/108813/perpres-no-39-tahun-2019. [Diakses 20 November 2022].

[13] M. P. M. W. Andre Antolis, Andhika Ramadhan Putra, Farabela Putri D. H., Hany Alexandra Philemon, "Alpha dan Beta Testing," *Binus University*, Jun. 30, 2020. [Online]. Available: https://socs.binus.ac.id/2020/06/30/alpha-dan-beta-testing/. [Diakses 25 Juli 2023]

[14] Anggi, "Beta Testing Adalah: Pengertian dan Bedanya dengan Alpha Testing," *Accurate*, Mar. 08, 2021. [Online]. Available: https://accurate.id/marketing-manajemen/beta-testing-adalah/. [Diakses 25 Juli 2023]

[15] Gulman Azkiya, "Mengenal Front End dan Back End Developer: Definisi, Tugas, Skill yang Dibutuhkan, Hingga Perbedaannya," *Skill Academy*, Dec. 01, 2021. [Online]. Available: https://blog.skillacademy.com/perbedaan-front-end-dan-back-end. [Diakses 26 Juli 2023].

[16] AdminDicoding, "Apa itu Back-End dan Back-End Developer?," *Dicoding Indonesia*, May 08, 2021. [Online]. Available: https://www.dicoding.com/blog/apa-itu-back-end/. [Diakses 25 Juli 2023]

[17] Abrar Firdiansyah, "Back End Developer: Lingkup Kerja, Skill, dan Tanggung Jawabnya," *Glints*, Jul. 29, 2022. [Online]. Available: https://glints.com/id/lowongan/pekerjaan-back-end-developer/. [Diakses 25 Juli 2023]

[18] F. D. Muqorobin, M. M., Anggraini, A., Rahmawati, A. D., Yohanes, D., & Ifkarina, "Pengaruh Open Banking berbasis Open API terhadap Eksistensi Perbankan," *MAKSIMUM Media Akunt. Univ. Muhammadiyah Semarang*, vol. 11 No.2, pp. 75–84, 2021.