

KAJIAN ARSITEKTUR BERLAPIS UNTUK PENGEMBANGAN APLIKASI DALAM Mendukung PEMBANGUNAN APLIKASI DI PEMERINTAHAN PROVINSI JAWA BARAT

STUDY OF LAYERED ARCHITECTURE DESIGN FOR APPLICATION DEVELOPMENT TECHNOLOGY TO SUPPORT THE GOVERNMENT APPLICATIONS IN WEST JAVA PROVINCE

Taufik Iqbal Ramdhani

Pusat Riset Kecerdasan Artifisial dan Keamanan Siber, Badan Riset dan Inovasi Nasional
Kawasan Sains dan Teknologi Samaun Samadikun, Jl. Sangkuriang, Dago, Kecamatan Coblong, Kota
Bandung 40135
tauf022@brin.go.id

(naskah masuk: 24 April 2024, naskah direvisi 27 Mei 2024, naskah diterima 01 Juni 2024)

ABSTRACT

The Electronic-Based Government System (EBGS) includes regulations for application development in the context of digital transformation to enhance public services. However, the increasing innovation in the government sector faces challenges in governing and standardizing application development technology, especially for specific applications in the West Java Provincial Government. The gap between technology selection and human resource planning can lead to an excessive number of applications without optimizing maintenance and development. Therefore, this study emphasizes the importance of technology standardization in application development to support the government's digital transformation. The study presents a four-layer technology architecture design: front-end, middleware, back-end, and persistent. The classification of technology standards is divided into three levels: small, medium, and large, based on the number of users, transactions, and data. By standardizing, it is expected to increase the efficiency and effectiveness of application development, reduce development costs, increase team productivity, and optimize application quality. This study also highlights the relevance of implementing this architecture in the West Java Provincial Government environment, with reference to existing local regulations and policies. The results of the study indicate that this four-layer technology standardization can be an effective solution to overcome the challenges of government application development and support sustainable digital transformation.

Keywords: *EBGS, standardization, application development*

ABSTRAK

Sistem Pemerintahan Berbasis Elektronik (SPBE) memuat peraturan mengenai pembangunan aplikasi dalam rangka transformasi digital untuk meningkatkan pelayanan publik. Namun, peningkatan inovasi di sektor pemerintahan menghadapi tantangan dalam tata kelola dan standarisasi teknologi pengembangan aplikasi, terutama aplikasi khusus di lingkungan Pemerintahan Provinsi Jawa Barat. Kesenjangan antara pemilihan teknologi dan perencanaan sumber daya manusia dapat menghasilkan aplikasi yang berlebihan tanpa optimalisasi pemeliharaan dan pengembangan. Oleh karena itu, penelitian ini menekankan pentingnya standarisasi teknologi dalam pengembangan aplikasi untuk mendukung transformasi digital pemerintah. Penelitian ini menyajikan perancangan arsitektur teknologi berbasis empat lapisan: *front-end, middleware, back-end, dan persistent*. Klasifikasi standar teknologi dibedakan menjadi tiga tingkatan: kecil, menengah, dan besar, berdasarkan jumlah pengguna, transaksi, dan data. Dengan standarisasi ini, diharapkan dapat meningkatkan efisiensi dan efektivitas pengembangan aplikasi, mengurangi biaya pengembangan, meningkatkan produktivitas tim, dan mengoptimalkan kualitas aplikasi. Penelitian ini juga menyoroti relevansi penerapan arsitektur ini di lingkungan Pemerintahan Provinsi Jawa Barat, dengan mengacu pada regulasi dan kebijakan lokal yang ada. Hasil penelitian menunjukkan bahwa standarisasi teknologi berbasis empat lapisan ini dapat menjadi solusi efektif untuk mengatasi tantangan pengembangan aplikasi pemerintah dan mendukung transformasi digital yang berkelanjutan.

Kata kunci: SPBE, arsitektur, pengembangan aplikasi

PENDAHULUAN

Konsep *Society 5.0* yang muncul sebagai respons terhadap keterbatasan pengembangan Industri 4.0, menunjukkan pergeseran fokus ke arah pembangunan yang lebih berpusat pada kebutuhan masyarakat. *Society 5.0* bertujuan untuk mengintegrasikan ruang fisik dan virtual melalui sistem fisik siber (SFS), menciptakan budaya yang sejalan dengan kebutuhan warganya (Nair dkk., 2021). Dalam konteks ini, transformasi digital di sektor pemerintahan Indonesia telah diakselerasi dengan penerbitan berbagai regulasi, seperti Peraturan Menteri Komunikasi dan Informatika No. 2 Tahun 2021 dan Peraturan Presiden No. 17 Tahun 2023 (Rahman & Megah Sari, 2023).

Meskipun dorongan inovasi semakin meningkat, sektor pemerintahan menghadapi tantangan dalam tata kelola dan standarisasi teknologi pengembangan aplikasi (Sudanta dkk., 2021). Kesenjangan antara pemilihan teknologi dan perencanaan pengembangan sumber daya manusia dapat menyebabkan jumlah aplikasi yang berlebihan, dengan pemeliharaan dan pengembangan yang tidak optimal (N. Y. Puspitasari dkk., 2019). Oleh karena itu, perlu diterapkan standarisasi teknologi dalam pengembangan aplikasi untuk mendukung transformasi digital pemerintah.

Pada perancangan transformasi digital pemerintahan yang tertuang dalam Sistem Pemerintahan Berbasis Elektronik (SPBE) menyebutkan bahwa pengembangan aplikasi dibagi menjadi dua yaitu aplikasi umum yang dapat digunakan oleh seluruh organisasi pemerintahan dan aplikasi khusus yang menyelesaikan persoalan khusus di masing-masing organisasi dikarenakan ada perlakuan khusus yang tidak dapat digeneralisasikan. Pemerintahan Provinsi Jawa Barat membuat aturan turunan mengenai hal tersebut yang tertuang pada Peraturan Gubernur Nomor 161 Tahun 2022 agar nilai-nilai daerah tetap dapat menjadi warna dalam menjalankan transformasi digital kepada masyarakat (Maulana, 2022). Selain itu, standar teknis dan prosedur mengenai pembangunan dan pengembangan aplikasi khusus di lingkungan pemerintahan Jawa Barat yang tertuang dalam Keputusan Gubernur Jawa Barat Nomor 48 Kep 437-Diskominfo Tahun 2023 yang menyatakan bahwa pentingnya suatu tatanan dalam pengembangan aplikasi khusus di lingkungan pemerintahan Jawa Barat.

Pada penelitian ini, dilakukan perancangan standarisasi teknologi yang melibatkan penggunaan pedoman dan desain untuk

memastikan konsistensi, kompatibilitas, dan interoperabilitas pada pengembangan aplikasi. Perancangan standarisasi ini dibagi menjadi empat lapisan yaitu: *front-end*, *middleware*, *back-end*, dan *persistent*. Hal ini memastikan pengembangan yang lebih terstruktur, efisien, dan konsisten.

Pentingnya standarisasi juga tercermin dalam pemisahan tingkatan skala kecil, menengah, dan besar, masing-masing dengan persyaratan teknologi yang berbeda (Renaldo Prasena & Sama, 2020). Penggolongan ini didasarkan pada jumlah pengguna, transaksi, dan data, dengan setiap tingkatan memiliki kebutuhan teknologi yang khusus untuk mencapai efisiensi dan efektivitas sumber daya.

Standarisasi teknologi dalam pengembangan aplikasi menawarkan sejumlah manfaat, termasuk konsistensi, efisiensi, kompatibilitas, keamanan, dan skalabilitas (Sigit Wahono & Alkadhim Alhabshy, 2023). Dengan menerapkan standar yang diakui secara luas, pengembang dapat mengurangi kompleksitas, menghemat waktu, dan memastikan hasil pengembangan aplikasi yang optimal (Irman Hermadi Yani Nurhadryani, 2023). Oleh karena itu, standarisasi teknologi memainkan peran krusial dalam mencapai konsistensi, kompatibilitas, ketertelusuran, dan kemajuan dalam pengembangan aplikasi untuk mendukung transformasi digital di lingkungan pemerintahan.

Penelitian ini disusun dengan sistematika penulisan sebagai berikut. Pada bagian 1 menjelaskan mengenai latar belakang penelitian. Pada bagian 2 mencakup penelitian terkait mengenai penerapan desain pengembangan aplikasi. Bagian 3 menyajikan pemilihan tinjauan teknologi untuk penelitian ini. Pada bagian 4 terdiri dari desain arsitektur kebutuhan teknologi. Terakhir pada bagian 5 meliputi kesimpulan dan pembahasan penelitian.

PENELITIAN TERKAIT

E-Government adalah hasil dari perubahan digital dalam ranah pelayanan publik dan evolusi layanan tersebut yang memanfaatkan kemajuan teknologi informasi (Hediyanto, 2018). Transformasi ini mendorong pelayanan publik untuk meningkatkan mutu layanan dengan mengintegrasikan inovasi teknologi. Pentingnya pengembangan aplikasi dalam konteks ini sangat terkait dengan penciptaan *e-government* yang efektif, di mana aplikasi-aplikasi tersebut memainkan peran krusial dalam memperbaiki

dan menyederhanakan berbagai aspek layanan publik secara menyeluruh.

Penelitian yang dilakukan oleh (Rohiatna, 2020) melalui pembangunan pelayanan perizinan pelayanan terpadu memudahkan masyarakat dalam melakukan permintaan pelayanan perizinan secara *online*. Pengembangan aplikasi tersebut dilakukan dengan pemanfaatan perangkat web dengan strategi *fullstack* dan pemanfaatan basis data. Namun pengembangan aplikasi tersebut masih belum mempertimbangkan skalabilitas terhadap lonjakan pengguna hingga tingkat provinsi.

Kompatibilitas merupakan kebutuhan krusial untuk menjamin suatu sistem dapat bekerja secara efisien dan optimal pada berbagai perangkat keras, sistem operasi, dan perangkat lunak pendukung lainnya. (Setianingrum dkk., 2020) melakukan penelitian mengenai penerapan aplikasi *e-government* untuk meningkatkan kualitas layanan publik dengan menerapkan prinsip kompatibilitas. Namun, hal ini tidak menjamin kemudahan jika terjadi diskrepansi terhadap suatu transaksi.

Ketertelusuran adalah komponen keamanan penting dalam audit dan penelusuran transaksi, maupun untuk keperluan kemudahan penelusuran jika terjadi suatu permasalahan. Penelitian yang dilakukan oleh Widiastuti & Krisnadwipayana (2022) mengadopsi pendekatan akuntabilitas dengan menyimpan transaksi dalam bentuk log transaksi. Selain itu, (Suprayogi & Rahmansa, 2019) melakukan penelitian dengan menerapkan pemantauan terhadap kinerja sistem dengan mencatat segala aktivitas untuk memudahkan penelusuran jika terjadi kendala pada sistem.

Penerapan suatu standar yang konsisten diperlukan untuk efisiensi serta kemudahan pemeliharaan sistem (Karniawati & Rahmadani, 2011). Diperlukan implementasi suatu prosedur dalam mendukung konsistensi terhadap suatu pengembangan aplikasi. Implementasi kebijakan terhadap pelayanan publik dilakukan oleh (Asri, 2020) untuk meningkatkan efisiensi terhadap layanan publik di masa pandemi. Selain itu, diperlukan faktor lain untuk menjamin kemudahan dalam pertukaran data antar layanan.

Interoperabilitas data merujuk pada kemampuan untuk saling berbagi, memanfaatkan, dan mengartikan data di dalam sistem yang memiliki karakteristik yang beragam atau heterogeny (N. Puspitasari dkk., 2021). (Fathan Ali dkk., 2023) melakukan pembangunan sistem dengan model terintegrasi menerapkan *government service bus* untuk meningkatkan interoperabilitas data antar layanan.

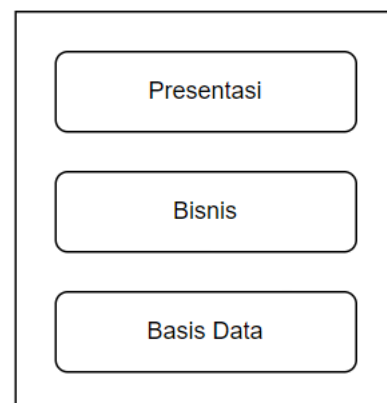
TINJAUAN TEKNOLOGI

Pada bagian ini menjelaskan mengenai teknologi yang digunakan pada penelitian ini. Terdapat tiga arsitektur utama yang digunakan pada penelitian ini yaitu arsitektur berlapis, arsitektur berorientasi layanan, dan arsitektur *microservice*. Tinjauan diawali mengenai penjelasan singkat mengenai arsitektur berlapis. Selanjutnya penjelasan mengenai arsitektur berorientasi layanan berbasis orientasi. Arsitektur *microservice* menjadi arsitektur pendukung terakhir yang digunakan pada penelitian ini sebagai komplemen dari arsitektur lainnya.

A. Arsitektur Berlapis

Arsitektur berlapis (AL), yang juga dikenal sebagai gaya arsitektur n-tier menjadi pilihan umum dalam pengembangan perangkat lunak (Mrabet dkk., 2020). Keberlanjutan, keakraban, dan keterjangkauan menjadi alasan utama di balik penerimaan luas gaya arsitektur ini, terutama dalam konteks aplikasi bisnis di berbagai organisasi (Liu dkk., 2019).

Dalam arsitektur berlapis, komponen sistem diatur ke dalam lapisan-lapisan logis horizontal (Din dkk., 2019). Setiap lapisan memiliki fungsi khususnya dalam aplikasi, seperti logika presentasi, bisnis, perantara, dan basis data. Meskipun tidak ada batasan baku terkait jumlah dan jenis lapisan, kebanyakan arsitektur berlapis terdiri dari tiga lapisan standar - presentasi, bisnis, dan basis data. Ada juga kemungkinan untuk menggabungkan lapisan bisnis dan perantara, terutama ketika logika persistensi tertanam dalam lapisan bisnis. Dengan demikian, aplikasi dapat memiliki tiga hingga lima lapisan, tergantung pada kompleksitas dan kebutuhan bisnis.



Gambar 1. Contoh Rancangan Arsitektur Berlapis

Setiap lapisan memiliki tanggung jawab dan tugas spesifik pada arsitektur ini. Lapisan presentasi menangani antarmuka pengguna dan komunikasi browser, sementara lapisan bisnis menjalankan aturan bisnis yang spesifik untuk permintaan tertentu. Pemisahan perhatian ini memfasilitasi penciptaan model peran dan tanggung jawab yang efisien dalam arsitektur. Misalnya, lapisan presentasi hanya bertanggung jawab untuk logika presentasi, sementara lapisan bisnis fokus pada logika bisnis, dan seterusnya.

Gaya arsitektur berlapis memungkinkan pemisahan konsep yang jelas, memudahkan pengembangan model peran dan tanggung jawab yang lebih efisien. Setiap lapisan hanya berurusan dengan logika yang terkait dengannya, memungkinkan pengembang untuk fokus pada aspek teknis domain mereka. Namun, perlu dicatat bahwa keuntungan ini dapat mengakibatkan kurangnya kelincahan, membuat respons terhadap perubahan menjadi lebih sulit.

Arsitektur berlapis menawarkan kelebihan terhadap kemudahan implementasi yang sederhana dan hemat biaya. Kelebihan ini terutama terlihat dalam kemudahan pemahaman dan pemeliharaan. Namun, seiring dengan pertumbuhan arsitektur, kompleksitasnya meningkat, dan keuntungan yang diperoleh menjadi berkurang. Keterbatasan pengujian dan risiko penyebaran manual juga dapat meningkatkan risiko kesalahan. Selain itu, arsitektur berlapis kurang mendukung toleransi kesalahan, yang dapat berdampak pada ketersediaan dan waktu pemulihan.

Evaluasi kinerja arsitektur berlapis menunjukkan bahwa implementasi yang sederhana dan hemat biaya, namun kurang cocok untuk sistem yang membutuhkan kinerja tinggi. Kompleksitas yang meningkat memerlukan upaya ekstra untuk meningkatkan skalabilitas dan kinerja dalam mengembangkan aplikasi menggunakan arsitektur ini.

B. Arsitektur Berorientasi Layanan

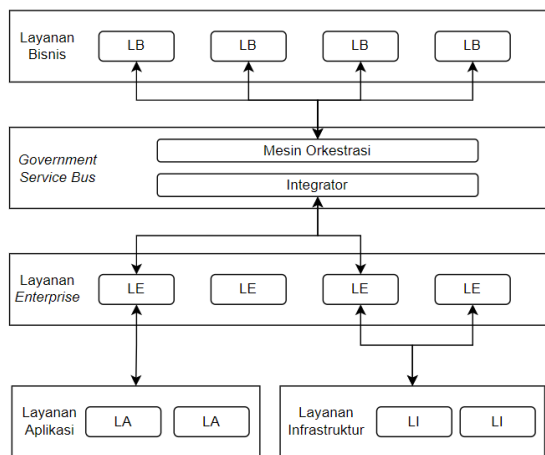
Arsitektur Berorientasi Layanan (AOL) memiliki filosofi yang menitikberatkan pada efisiensi penggunaan kembali terhadap fungsi yang digunakan. Fokus utama dari AOL yaitu memungkinkan pengembangan perilaku bisnis yang dapat digunakan kembali secara progresif. Para arsitek yang bekerja dengan AOL diarahkan untuk secara aktif mengidentifikasi dan memanfaatkan peluang

penggunaan kembali sebanyak mungkin, menjadikan AOL sebagai pendekatan yang sangat strategis dalam pengembangan perangkat lunak.

Komponen utama dalam AOL terdiri dari beberapa jenis layanan, antara lain:

- a. Layanan bisnis: merupakan tingkat arsitektur tertinggi yang berfungsi sebagai pintu gerbang utama. Layanan bisnis merepresentasikan perilaku domain bisnis. Definisi layanan ini melibatkan input, output, dan terkadang skema informasi, tanpa menyertakan implementasi kode aktual. Pengguna bisnis secara umum yang mendefinisikan Layanan Bisnis, memberikan karakteristik khusus pada tingkat ini.
- b. Layanan umum: didesain untuk memfasilitasi implementasi yang mulus di berbagai domain bisnis oleh tim pengembangan. Layanan ini bertanggung jawab untuk menciptakan perilaku atomik di sekitar domain bisnis tertentu, seperti pembuatan pelanggan atau perhitungan penawaran. Layanan perusahaan berfungsi sebagai blok bangunan bagi layanan bisnis yang lebih besar, yang terkoneksi melalui mesin orkestrasi. Pendekatan ini memisahkan tanggung jawab untuk memungkinkan penggunaan kembali komponen, meminimalkan upaya ulang dalam pengembangan aplikasi.
- c. Layanan Aplikasi: beberapa layanan mungkin tidak memerlukan granularitas tingkat perusahaan atau dapat digunakan kembali seperti layanan perusahaan. Layanan aplikasi umumnya bersifat satu kali implementasi dan dikelola oleh tim aplikasi tertentu. Misalnya, jika aplikasi memerlukan lokasi geografis, pembuatan layanan yang dapat digunakan kembali untuk itu mungkin tidak efisien bagi organisasi, sehingga layanan aplikasi menjadi solusi yang berguna dalam kasus ini.
- d. Layanan Infrastruktur: menyediakan dukungan operasional, termasuk pemantauan, pencatatan, autentikasi, dan otorisasi. Layanan ini biasanya merupakan implementasi konkret yang dikelola oleh tim infrastruktur yang bekerja bersama-sama dengan tim operasi.

- e. Mesin Orkestrasi: Merupakan inti dari arsitektur AOL, berfungsi menyatukan implementasi layanan bisnis. Mesin orkestrasi memungkinkan fitur seperti koordinasi transaksional dan transformasi pesan. Berbeda dengan arsitektur layanan mikro yang mungkin menggunakan database per layanan, AOL biasanya menggunakan satu atau beberapa database relasional. Mesin ini berperan dalam mendefinisikan hubungan antara layanan bisnis dan perusahaan, serta batas-batas transaksi.



Gambar 2. Contoh Rancangan Arsitektur Berorientasi Layanan

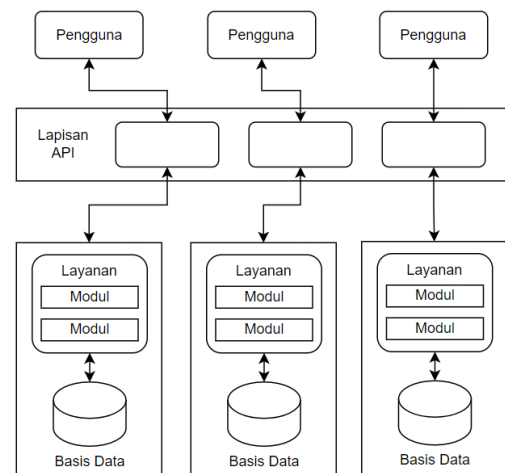
Meskipun AOL memberikan kontribusi signifikan dalam penggunaan kembali komponen bisnis, tantangan yang perlu dicermati meliputi granularitas transaksi, di mana sulit menentukan tingkat granularitas yang tepat dalam arsitektur AOL, dan memindahkan perilaku transaksi ke dalam alat orkestrasi dapat menyulitkan menentukan batas transaksi yang tepat antara layanan.

Tantangan lain yaitu, *deployability* dan *testability* dikarenakan dukungan yang kurang dan prioritas rendah terhadap aspek ini. Selain itu, factor kinerja bukan menjadi fokus utama dalam AOL, terutama karena setiap permintaan bisnis tersegmentasi, yang dapat mempengaruhi kinerja arsitektur.

Dengan memahami secara mendalam komponen dan tantangan yang terkait dengan AOL, penelitian ini memberikan pandangan yang dapat membimbing pengembang dan arsitek perangkat lunak dalam membuat keputusan desain yang informasional dan efektif.

C. Arsitektur Layanan Mikro

Arsitektur Layanan Mikro (ALM) merupakan suatu paradigm arsitektural yang telah mencapai prominensi sejak diperkenalkan oleh Martin Fowler dan James Lewis melalui tulisan blog pada Maret 2014. Gaya arsitektural ini memanfaatkan prinsip-prinsip krusial dari Desain Berbasis Domain (DBD), khususnya konsep "konteks terbatas." Dalam ALM, prinsip ini tercermin dengan jelas, menciptakan pemisahan yang kuat antar bagian-bagian sistem yang beroperasi secara independen sesuai dengan konteksnya masing-masing.



Gambar 3. Contoh Rancangan Arsitektur Layanan Mikro

Filosofi inti dari ALM adalah untuk mencapai pemisahan tinggi yang sejalan dengan pemisahan logis dalam konteks terbatas. Setiap komponen sistem hanya terhubung dengan elemen-elemen lain dalam konteks terbatasnya, menciptakan fleksibilitas yang optimal dalam pengembangan dan pemeliharaan perangkat lunak.

Keuntungan yang diperoleh ALM meliputi skalabilitas dan elastisitas melalui otomatisasi dan integrasi yang cerdas, ALM mendukung skalabilitas, memungkinkan pengembang untuk membangun sistem yang dapat menyesuaikan diri secara dinamis tergantung pada beban kerja.

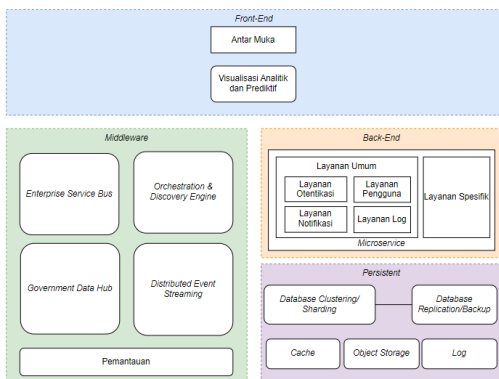
Selain itu, pemisahan tinggi pada arsitektur ini memungkinkan pemisahan yang tinggi pada tingkat inkremental, memfasilitasi evolusi perubahan dan adaptasi sistem terhadap kebutuhan bisnis yang berubah.

Walaupun memberikan keunggulan tersebut, MS juga dihadapkan pada sejumlah tantangan, seperti tantangan kinerja yang memiliki sifat terdistribusi sering mengakibatkan panggilan jaringan yang

berlebihan, berpotensi merugikan kinerja sistem. Tantangan ini diatasi dengan menerapkan pola seperti *caching* dan replikasi data serta pendekatan koreografi untuk mengoptimalkan komunikasi antar layanan. Selain itu, isu keamanan pada penggunaan beberapa layanan dapat menimbulkan tantangan terkait pemeriksaan keamanan pada setiap titik akhir, yang dapat menambah *overhead*.

HASIL DAN PEMBAHASAN

Berdasarkan tiga gaya arsitektur sebelumnya, divisi lapisan diciptakan untuk solusi standarisasi teknologi pengembangan perangkat lunak. Untuk pembagian layer, penggabungan dan pembentukan layer baru adalah *front-end*, *middleware*, *back-end*, dan *persistent*. Dari penelitian yang dilakukan oleh (Gong dkk., 2020) seiring dengan perkembangan zaman, terutama di era *big data*, perlu dilakukan pemisahan arsitektur berupa *front-end* dan *back-end* untuk memastikan bahwa sistem yang dibuat dapat memenuhi kebutuhan pelanggan. Sementara itu, (Champaneria dkk., 2023) menyatakan bahwa *middleware* digunakan sebagai perantara untuk komunikasi antara berbagai layanan bisnis. menyebutkan bahwa lapisan *middleware* merupakan solusi untuk meningkatkan interoperabilitas dan menghilangkan kompleksitas dalam komunikasi antar *Web-of-Things* (WoT). Dalam penelitian oleh (Larian dkk., 2022) *middleware* memiliki kemampuan untuk reaktif dalam perubahan data *real-time* dengan ketersediaan penataan data otomatis. (Greco dkk., 2019) menerapkan lapisan persisten yang bertanggung jawab untuk menyimpan data. Oleh karena itu, standarisasi ini menambahkan lapisan *middleware* sebagai gerbang komunikasi antara *front-end* dan *back-end* yang disertai penyimpanan data hasil transaksi pada lapisan *persistent*.



Gambar 4. Solusi Rancangan Arsitektur Teknologi Pengembangan Aplikasi

Berikut adalah deskripsi singkat dari masing-masing komponen:

- a. *Front-end* adalah bagian dari aplikasi yang berhubungan langsung dengan pengguna akhir. Ini termasuk antarmuka pengguna, desain grafis, elemen interaktif, dan fitur visual lainnya yang digunakan oleh pengguna untuk berinteraksi dengan aplikasi. *Front-end* biasanya menggabungkan bahasa pemrograman seperti HTML, CSS, dan JavaScript untuk membangun tampilan aplikasi yang menarik dan responsif. Tujuan utama dari *front-end* adalah untuk memberikan pengalaman pengguna yang baik dan intuitif.
- b. *Middleware* adalah lapisan perangkat lunak yang berada di antara *front-end* dan *back-end* (IBM). *Middleware* bertindak sebagai jembatan komunikasi antara kedua komponen ini. *Middleware* memproses permintaan dari *front-end* dan menyampaikannya ke *back-end* untuk diproses lebih lanjut. Ini juga bertanggung jawab untuk mengelola logika bisnis, alur kerja, dan integrasi dengan sistem lain.
- c. *Back-end* adalah bagian dari aplikasi yang bertanggung jawab untuk pemrosesan data, logika bisnis, dan interaksi dengan sistem atau database. Ini melibatkan manajemen server, pemrosesan data, pengaturan keamanan, dan penyimpanan informasi. *Back-end* menggunakan bahasa pemrograman seperti *Java*, *Python*, *C #*, atau *PHP* dan sering menggunakan *database* untuk menyimpan dan mengelola data aplikasi. Tujuan utama dari *back-end* adalah untuk menyediakan dan memanipulasi data yang dibutuhkan oleh *front-end* serta menjalankan logika bisnis yang sesuai.
- d. *Persistent* mengacu pada bagian situs web yang bertanggung jawab untuk menyimpan, mengelola, dan mengakses data yang diperlukan oleh situs web. *Persistent* berfungsi sebagai tempat penyimpanan informasi dan dapat diakses oleh pengguna melalui aplikasi atau *application programming interface* (API). Secara arsitektur, *persistent* dibagi menjadi 3 bagian, yaitu:
 - e. Basis data berfungsi sebagai tempat penyimpanan data yang digunakan oleh

aplikasi. Ketika pengguna menggunakan aplikasi, data yang diperlukan seperti informasi pengguna, pesanan, konten, dan lainnya akan disimpan ke dalam database.

- f. *Storage server* atau *server* penyimpanan adalah server yang berfungsi untuk menyimpan dan mengelola data dalam jumlah besar. *Server* penyimpanan digunakan untuk menyimpan data dalam jumlah besar yang biasanya tidak dapat disimpan pada hard disk atau memori internal pada komputer biasa. *Server* penyimpanan menyediakan ruang penyimpanan terpusat dan dapat diakses dari jaringan.
- g. *Cache* berfungsi sebagai penyimpanan data sementara yang digunakan berulang kali oleh suatu sistem. *Caching* digunakan untuk mempercepat proses akses data dengan menyimpan salinan data yang sering digunakan secara lokal, sehingga meminimalkan waktu akses ke sumber data yang lebih lambat seperti disk atau jaringan.

Pembagian ini memungkinkan pengembang untuk memisahkan peran dan tanggung jawab dalam pengembangan aplikasi. *Front-end* berfokus pada penampilan dan interaksi dengan pengguna, *middleware* mengatur komunikasi antara *front-end* dan *back-end*, sedangkan *back-end* menangani pemrosesan data dan logika bisnis. Kolaborasi yang baik antara ketiga komponen ini memungkinkan pengembangan aplikasi yang efektif dan berfungsi dengan baik.

Berdasarkan ketiga referensi gaya arsitektur tersebut, standarisasi teknologi pengembangan aplikasi dilakukan menjadi tiga klasifikasi tingkat kecil, menengah, dan *enterprise* berdasarkan tiga kriteria, yaitu: jumlah pengguna, jumlah transaksi, dan jumlah data. Dalam pengembangan aplikasi, standarisasi teknologi berdasarkan tiga klasifikasi (kecil, menengah, dan *enterprise*) dan tiga kriteria (jumlah pengguna, jumlah transaksi, dan jumlah data) adalah pendekatan yang umum untuk memastikan bahwa aplikasi yang dikembangkan dapat memenuhi kebutuhan operasional dengan efisien dan efektif.

Ketiga kriteria ini memegang peranan penting dalam menentukan klasifikasi standarisasi ini. Jumlah pengguna digunakan sebagai acuan penting untuk menentukan apakah suatu aplikasi memerlukan seperangkat teknologi tertentu atau hanya satu atau dua teknologi. Begitu juga dengan jumlah transaksi dan jumlah data. Hal ini untuk melakukan efisiensi dan

efektivitas sumber daya, baik infrastruktur, teknologi, maupun manusia.

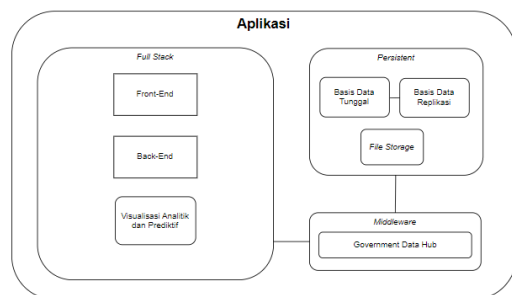
Jumlah pengguna menunjukkan skala aplikasi dan menentukan beban pada sistem. Lebih banyak pengguna berarti lebih banyak permintaan simultan, yang membutuhkan kapasitas server dan bandwidth yang memadai. Liu dkk. (2019) menunjukkan bahwa arsitektur sistem harus mempertimbangkan jumlah pengguna untuk memastikan responsivitas dan ketersediaan layanan. Untuk jumlah transaksi menunjukkan intensitas penggunaan aplikasi. Semakin tinggi jumlah transaksi, semakin besar beban pemrosesan pada sistem back-end dan middleware. Gong dkk. (2020) menekankan pentingnya pemisahan front-end dan back-end untuk menangani jumlah transaksi yang tinggi secara efisien.

Pertumbuhan data menunjukkan kebutuhan penyimpanan dan manajemen data. Semakin besar pertumbuhan data, semakin kompleks kebutuhan infrastruktur penyimpanan dan akses data. Greco dkk. (2019) membahas pentingnya arsitektur penyimpanan yang *scalable* untuk mendukung pertumbuhan data yang besar dan memastikan performa aplikasi tetap optimal. Dengan demikian, mengklasifikasikan arsitektur berdasarkan tiga kriteria ini membantu dalam merancang sistem yang *scalable*, efisien, dan siap untuk pertumbuhan dan peningkatan kebutuhan di masa depan.

Berikut pembagian klasifikasi level standarisasi teknologi pengembangan aplikasi yang diusulkan:

a. Kecil

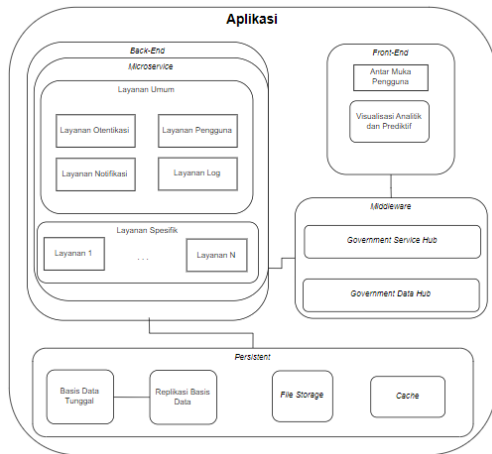
Pada tingkat skala kecil, fokus utamanya adalah pada pengembangan aplikasi yang relatif sederhana dan memiliki ruang lingkup terbatas. Biasanya, proyek pada tingkat ini melibatkan tim kecil pengembang dan waktu pengembangan yang singkat. Pada skala ini adalah untuk sistem dengan kurang dari 500 pengguna, dan/atau kurang dari 1.000 transaksi harian, dan/atau pertumbuhan data bulanan kurang dari 1 GB.



Gambar 5. Solusi Rancangan Arsitektur Pengembangan Aplikasi Skala Kecil

b. Medium

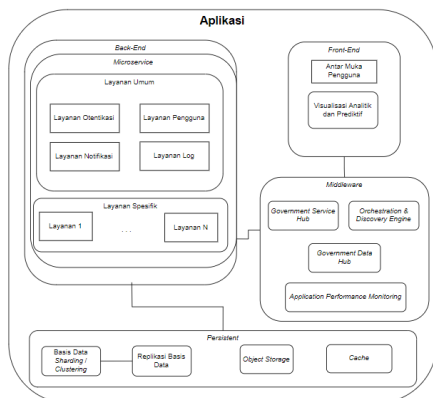
Level medium mengacu pada proyek pengembangan aplikasi yang lebih kompleks dengan kebutuhan yang lebih besar. Pada level ini, mungkin terdapat tim pengembang yang lebih besar, dan proyek memerlukan waktu pengembangan yang lebih lama. Pada skala ini diperuntukkan untuk sistem dengan jumlah pengguna kurang dari 5.000, dan/atau transaksi harian kurang dari 100.000, dan/atau pertumbuhan data per bulan kurang dari 10 GB.



Gambar 6. Solusi Rancangan Arsitektur Pengembangan Aplikasi Skala Menengah

c. Besar

Level besar adalah level tertinggi dalam standarisasi teknologi pengembangan aplikasi. Proyek pada level ini memiliki skala dan kompleksitas yang sangat besar, melibatkan tim pengembang yang besar dan memerlukan waktu pengembangan yang signifikan. Pada skala ini diperuntukkan untuk sistem dengan jumlah pengguna lebih dari 5.000, dan/atau transaksi harian lebih dari 100.000, dan/atau pertumbuhan data per bulan lebih dari 10 GB.



Gambar 7. Solusi Rancangan Arsitektur Pengembangan Aplikasi Skala Besar

Tabel 1. Perbandingan Arsitektur dan Teknologi Aplikasi Pemerintahan di Jawa Barat

Judul Penelitian	Arsitektur	Teknologi
Implementasi Pelayanan Perizinan Secara Online Melalui SIMPATEN (Rohiatna, 2020)	Arsitektur Berlapis (2-tier)	Front-end: HTML, CSS, JavaScript Back-end: PHP, MySQL
Penerapan E-Government dalam Meningkatkan Kualitas Pelayanan Publik (Setianingrum dkk., 2020)	Arsitektur Berorientasi Layanan (SOA)	Front-end: HTML, CSS, AngularJS Back-end: Java, Spring Middleware: Apache Camel
Sistem Informasi Pelayanan Desa Berbasis Web di Desa Wanajaya (Widiastuti & Krisnadwipayana, 2022)	Arsitektur Berlapis (2-tier)	Front-end: HTML, CSS, JavaScript Back-end: PHP, Laravel, MySQL
Collaborative Digital Transformation Pemerintah Provinsi Jawa Barat (Maulana, 2022)	Arsitektur Microservices	Front-end: ReactJS Back-end: Node.js, Express, MongoDB Middleware: Nginx
Model Arsitektur Sistem Pemerintahan Berbasis Elektronik (SPBE) (Sudanta dkk., 2021)	Arsitektur Berorientasi Layanan (SOA)	Front-end: HTML, CSS, JavaScript, AngularJS Back-end: Java, Spring Boot Middleware: Apache Kafka

Arsitektur yang digunakan dalam lima referensi aplikasi pemerintahan di Jawa Barat menunjukkan beberapa kelemahan jika dibandingkan dengan arsitektur yang diusung dalam penelitian ini. Arsitektur Berlapis (2-tier) yang digunakan dalam SIMPATEN dan Sistem Informasi Pelayanan Desa cenderung memiliki kompleksitas dalam skalabilitas dan pemeliharaan. Pemisahan tugas yang tidak selalu jelas dan komunikasi antar lapisan yang terbatas sering menjadi bottleneck, sehingga menghambat efisiensi dan fleksibilitas dalam pengembangan dan pengoperasian aplikasi.

Arsitektur Berorientasi Layanan (SOA) yang diterapkan di Dinas Penanaman Modal dan PTSP Kota Bandung serta Sistem Pemerintahan Berbasis Elektronik (SPBE) juga memiliki kelemahan, terutama dalam hal manajemen layanan yang rumit dan overhead kinerja yang tinggi. Kompleksitas orkestrasi layanan sering menambah beban pengelolaan dan dapat menyebabkan penurunan kinerja aplikasi. Selain itu, integrasi antar layanan dari berbagai vendor dapat menjadi tantangan besar yang mempengaruhi interoperabilitas sistem.

Sementara itu, Arsitektur *Microservices* yang digunakan dalam transformasi digital kolaboratif oleh Pemerintah Provinsi Jawa Barat menghadapi tantangan dalam manajemen dan isolasi data. Pengelolaan banyak layanan kecil secara terpisah menambah kompleksitas pengelolaan secara keseluruhan. Selain itu, komunikasi antar layanan *microservices* dapat menambah latensi jaringan dan mempengaruhi kinerja aplikasi secara keseluruhan. Kelemahan-kelemahan ini menunjukkan bahwa arsitektur yang diusung dalam penelitian ini, dengan penggunaan *middleware* yang terpusat dan struktur lapisan yang lebih jelas, menawarkan solusi yang lebih efisien dan scalable untuk pengembangan aplikasi pemerintah.

Tabel 2. Klasifikasi Skala Aplikasi

Klasifikasi	Jumlah Pengguna	Jumlah Transaksi Harian	Pertumbuhan Data Bulanan
Kecil	< 500	< 1.000	< 1 GB
Menengah	500 - 5.000	1.000 - 100.000	1 - 10 GB
Besar	> 5.000	> 100.000	> 10 GB

Tabel 3. Saran implementasi

Aplikasi	Klasifikasi
Aplikasi Pelayanan Perizinan (SIMPATEN)	Jumlah Pengguna: 300 Jumlah Transaksi Harian: 500 Pertumbuhan Data Bulanan: 0.5 GB Klasifikasi: Kecil
Aplikasi E-Government DPMPSTP Kota Bandung	Jumlah Pengguna: 2.000 Jumlah Transaksi Harian: 50.000 Pertumbuhan Data Bulanan: 5 GB Klasifikasi: Menengah
Sistem Pemerintahan Berbasis Elektronik (SPBE)	Jumlah Pengguna: 10.000 Jumlah Transaksi Harian: 200.000 Pertumbuhan Data Bulanan: 20 GB Klasifikasi: Besar

KESIMPULAN DAN SARAN

Penelitian ini mengembangkan standar arsitektur teknologi berbasis empat lapisan (*front-end, middleware, back-end, dan persistent*) untuk mendukung pengembangan aplikasi khusus di lingkungan Pemerintahan Provinsi Jawa Barat. Standarisasi ini dirancang untuk meningkatkan efisiensi dan efektivitas pengembangan aplikasi melalui struktur yang lebih terorganisir dan interoperabilitas yang lebih baik. Dibandingkan dengan arsitektur yang digunakan dalam referensi, arsitektur ini menawarkan keuntungan dalam hal skalabilitas, pemeliharaan, dan komunikasi antar layanan.

Penggunaan lapisan *middleware* yang terpusat memungkinkan pengurangan kompleksitas dan

meningkatkan kinerja aplikasi secara keseluruhan. Selain itu, standarisasi teknologi ini mendukung pemisahan tanggung jawab yang jelas antar lapisan, memudahkan pengembangan dan pemeliharaan berkelanjutan. Dengan mengimplementasikan arsitektur ini, diharapkan aplikasi yang dikembangkan dapat lebih responsif terhadap kebutuhan pengguna, lebih mudah diintegrasikan dengan sistem lain, dan lebih efisien dalam pemanfaatan sumber daya.

DAFTAR PUSTAKA

Asri, O. (2020). IMPLEMENTASI KEBIJAKAN PELAYANAN PUBLIK BERBASIS APLIKASI PADA ERA COVID19 DI KOTA CIMAH I PROVINSI JAWA BARAT. *Jurnal Pemerintahan Daerah di Indonesia*, 12(4), 695–712. <https://doi.org/https://doi.org/10.54783/jv.v12i4.330>

Champaneria, T., Jardosh, S., & Makwana, A. (2023). *Microservices in IoT Middleware Architectures: Architecture, Trends, and Challenges*. Dalam J. Choudrie, P. Mahalle, T. Perumal, & A. Joshi (Ed.), *IOT with Smart Systems* (hlm. 381–395). Springer Nature Singapore.

Din, S., Paul, A., & Rehman, A. (2019). 5G-enabled Hierarchical architecture for software-defined intelligent transportation system. *Computer Networks*, 150, 81–89. <https://doi.org/10.1016/j.comnet.2018.11.035>

Fathan Ali, D., Siti Fatonah, N., Firmansyah, G., & Akbar, H. (2023). Data Interoperability Model in Integrated Public Service Applications Based on Government Service Bus (Case Study: Tangerang Regency Communication and Information Office). *Jurnal Indonesia Sosial Sains*, 4(09), 778. <http://jiss.publikasiindonesia.id/>

Gong, Y., Gu, F., Chen, K., & Wang, F. (2020). The Architecture of Micro-services and the Separation of Frond-end and Back-end Applied in a Campus Information System. *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 321–324. <https://doi.org/10.1109/AEECA49918.2020.9213662>

Greco, L., Ritrovato, P., & Xhafa, F. (2019). An edge-stream computing infrastructure for real-time analysis of wearable sensors data. *Future Generation Computer Systems*, 93, 515–528. <https://doi.org/https://doi.org/10.1016/j.future.2018.10.058>

Hediyanto, U. Y. K. S. (2018). Penelitian E-Government di Propinsi Jawa Barat: Kajian Pustaka Sistematis. *Jurnal Metris*, 19(02), 95–104. <https://scholar.googleusercontent.com/scholar>

- ?q=cache:D5usxYnQ-NQJ:scholar.google.com/+sistem+informasi+java+barat&hl=en&as_sdt=0,5
- Irman Hermadi Yani Nurhadryani, R. (2023). Analisis Uji Performa Aplikasi Dari Hasil Implementasi Refactoring Arsitektur Monolitik Ke Mikroservis dengan Decomposition dan Strangler Pattern. *Jurnal Sistem Cerdas*, 6(3), 189–203.
- Karniawati, N., & Rahmadani, R. (2011). ANALISIS KEBIJAKAN PENERAPAN E-GOVERNMENT MELALUI SISTEM INFORMASI MANAJEMEN KEPEGAWAIAN (SIMPEG) (SUATU STUDI PADA BIRO KEPEGAWAIAN SEKRETARIAT DAERAH PROVINSI JAWA BARAT). *Majalah Ilmiah UNIKOM*.
- Larian, H., Larian, A., Sharifi, M., & Movahednejad, H. (2022). *Towards Web of Things Middleware: A Systematic Review*. <http://arxiv.org/abs/2201.08456>
- Liu, K., Xu, X., Chen, M., Liu, B., Wu, L., & Lee, V. C. S. (2019). A Hierarchical architecture for the future internet of vehicles. *IEEE Communications Magazine*, 57(7), 41–47. <https://doi.org/10.1109/MCOM.2019.1800772>
- Maulana, R. Y. (2022). Collaborative Digital Transformation Pemerintah Provinsi Jawa Barat. *Jurnal Noken: Ilmu-Ilmu Sosial*, 7(2), 263–277. <https://doi.org/https://doi.org/10.33506/jn.v7i2.1784>
- Mrabet, H., Belguith, S., Alhomoud, A., & Jemai, A. (2020). A survey of IoT security based on a layered architecture of sensing and data analysis. Dalam *Sensors (Switzerland)* (Vol. 20, Nomor 13, hlm. 1–20). MDPI AG. <https://doi.org/10.3390/s20133625>
- Nair, M. M., Tyagi, A. K., & Sreenath, N. (2021, Januari 27). The Future with Industry 4.0 at the Core of Society 5.0: Open Issues, Future Opportunities and Challenges. 2021 *International Conference on Computer Communication and Informatics, ICCCI 2021*. <https://doi.org/10.1109/ICCCI50826.2021.9402498>
- Puspitasari, N., Budiman, E., Sulaiman, Y. N., & Firdaus, M. B. (2021). Microservice API Implementation for E-Government Service Interoperability. *Journal of Physics: Conference Series*, 1807(1). <https://doi.org/10.1088/1742-6596/1807/1/012005>
- Puspitasari, N. Y., Ams, A., Badan, P., Nasional, S., Bppt, G. I., Mh, J., No, T., & Pusat, J. (2019). STRATEGI TATA KELOLA PERENCANAAN DAN PEMANFAATAN TIK GUNA Mendukung Peningkatan Kualitas SPBE DI BSN. *Pertemuan dan Presentasi Ilmiah Standardisasi 2019*. <https://doi.org/10.31153/ppis.2019.30>
- Rahman, A., & Megah Sari, D. (2023). Analisis Transformasi Digital Pelayanan Publik Menggunakan Metode Gartner Analytic Asceedency (Studi Kasus : Pelayanan Perizinan Pemerintah Daerah Kabupaten Majene. *Jurnal Komputer Terapan*, 9(2), 111–121. <https://doi.org/10.35143/jkt.v9i2.5916>
- Renaldo Prasena, R., & Sama, H. (2020). STUDI KOMPARASI PENGEMBANGAN WEBSITE DENGAN FRAMEWORK CODEIGNITER DAN LARAVEL. *Conference on Business, Social Sciences and Innovation Technology*, 1. <http://journal.uib.ac.id/index.php/cbssit>
- Rohiatna, N. (2020). IMPLEMENTASI PELAYANAN PERIZINAN SECARA ONLINE MELALUI SISTEM INFORMASI MANAJEMEN PELAYANAN ADMINISTRASI TERPADU KECAMATAN (SIMPATEN/OSS) DI KECAMATAN BANDUNG KIDUL KOTA BANDUNG PROVINSI JAWA BARAT. *Jurnal Pemerintahan Daerah di Indonesia*, 12(3), 515–538. <https://doi.org/https://doi.org/10.54783/jv.v12i3.296>
- Setianingrum, O. K., Sumaryadi, N., Wargadinata, E., & Kasini. (2020). PENERAPAN EGOVERNMENT DALAM MENINGKATKAN KUALITAS PELAYANAN PUBLIK DI DINAS PENANAMAN MODAL DAN PELAYANAN TERPADU SATU PINTU KOTA BANDUNG PROVINSI JAWA BARAT. *Jurnal Pemerintahan Daerah di Indonesia*, 12(4), 843–854. <https://doi.org/https://doi.org/10.54783/jv.v12i4.344>
- Sigit Wahono, P., & Alkadhim Alhabshy, M. (2023). PENGGUNAAN APLIKASI E-AUDIT DALAM SISTEM INFORMASI MANAJEMEN INSPEKTORAT POLRI. *Jurnal Ilmiah Global Education*, 4(2).
- Sudanta, G., Kuru, N., Fajar, S., Gumilang, S., & Nugraha, R. A. (2021). MODEL ARSITEKTUR SISTEM PEMERINTAHAN BERBASIS ELEKTRONIK (SPBE) DOMAIN PROSES BISNIS PADA PEMERINTAH KABUPATEN KUNINGAN. *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 6(2).
- Suprayogi, B., & Rahmanesa, A. (2019). Penerapan Framework Bootstrap dalam Sistem Informasi Pendidikan SMA Negeri 1 Pacet Cianjur Jawa Barat. *Jurnal Teknologi Informasi Komunikasi*, 6(2). <https://doi.org/https://doi.org/10.38204/tematik.v6i2.244>
- Widiastuti, I., & Krisnadwipayana, U. (2022). Sistem Informasi Pelayanan Desa Berbasis Web di Desa Wanajaya Jawa Barat. *Jurnal Pengabdian Masyarakat: DIKMAS*, 877(3). <https://doi.org/10.37905/dikmas.2.3.877-886.2022>