Penerapan Multi-Tier Architecture Pada Pembuatan Sistem Surat Menyurat Berbasis Web Di Kantor Desa Pakning Asal

Hastita Sari[1], M. Asep Subandri[2]

[1,2] *Program Studi Rekayasa Perangkat Lunak- Politeknik Negeri Bengkalis*

Jalan Bathin Alam Sei Alam – Bengkalis - Indonesia

hastita14@gmail.com, subandri@polbeng.ac.id

***Abstrak*—Pemanfaatan teknologi informasi dalam pelayanan administrasi desa menjadi kebutuhan untuk meningkatkan efektivitas dan efisiensi kerja aparatur desa. Proses surat menyurat di Kantor Desa Pakning Asal yang masih dilakukan secara manual menyebabkan keterlambatan pelayanan, kesalahan pencatatan, serta kesulitan dalam pengelolaan arsip surat. Penelitian ini bertujuan untuk menerapkan arsitektur multi-tier pada pembuatan sistem surat menyurat berbasis web di Kantor Desa Pakning Asal guna menghasilkan sistem yang terstruktur dan mudah dipelihara. Penerapan arsitektur multi-tier dilakukan dengan memisahkan lapisan presentasi, lapisan logika aplikasi, dan lapisan akses data sehingga setiap lapisan memiliki tanggung jawab yang jelas. Pengujian sistem dilakukan untuk memastikan fungsionalitas dan integrasi antar lapisan berjalan dengan baik. Hasil penelitian menunjukkan bahwa penerapan arsitektur multi-tier mampu meningkatkan keteraturan struktur aplikasi, mempermudah proses pengembangan dan pemeliharaan sistem, serta meningkatkan kualitas pelayanan administrasi surat menyurat di Kantor Desa Pakning Asal secara signifikan. Berdasarkan hasil tersebut, sistem ini dapat menjadi fondasi yang kuat bagi pengembangan layanan administrasi desa di masa mendatang.**

***Kata Kunci*— Multi-Tier Architecture, Sistem Surat Menyurat.**

Abstract— The utilization of information technology in village administrative services has become an absolute necessity to improve the effectiveness and efficiency of village officials' work. The correspondence process at the Pakning Asal Village Office, which is still carried out manually, causes service delays, recording errors, and difficulties in managing document archives. This study aims to implement a multi-tier architecture in the development of a web-based correspondence system at the Pakning Asal Village Office in order to produce a well-structured and easily maintainable system. The implementation of the multi-tier architecture is carried out by separating the presentation layer, application logic layer, and data access layer so that each layer has clear and distinct responsibilities. System testing was conducted to ensure that the functionality and integration between layers operate properly. The results of this study indicate that the implementation of a multi-tier architecture significantly improves the structural organization of the application, facilitates system development and maintenance processes, and enhances the quality of administrative correspondence services at the Pakning Asal Village Office. Based on these outcomes, this system can serve as a solid foundation for the future development of village administrative services.

Keywords: Multi-Tier Architecture, Correspondence System.

I. PENDAHULUAN

Perkembangan teknologi informasi yang begitu pesat telah memberikan dampak signifikan terhadap berbagai aspek kehidupan, termasuk sektor pemerintahan desa. Sistem berbasis web tidak lagi hanya berfungsi sebagai alat informasi, tetapi juga telah menjadi platform layanan digital yang mampu menunjang kegiatan administratif secara real-time dan efisien. Namun pada kenyataannya, proses pembuatan website masih sering dilakukan tanpa memperhatikan struktur arsitektur yang baik. Banyak pengembang mencampuradukkan tampilan antarmuka, proses logika, dan pengelolaan data dalam satu bagian, tanpa pemisahan yang jelas. Akibatnya, sistem menjadi tidak teratur, sulit dikembangkan, dan rawan terjadi kesalahan di berbagai tempat. Kondisi ini juga menyulitkan dalam proses perbaikan maupun penambahan fitur di kemudian hari karena struktur sistem yang tidak tertata dengan baik sejak awal. Solusi yang dapat diterapkan untuk mengatasi permasalahan tersebut adalah multi-tier architecture, yang secara umum membagi sistem menjadi beberapa lapisan logis dan fisik untuk meningkatkan skalabilitas keamanan, dan kemudahan pemeliharaan. Dalam penelitian oleh singh, dijelaskan bahwa arsitekur multi-tier mampu memisahkan komponen sistem seperti presentasi, logika aplikasi, dan penyimpanan data, sehingga mempermudahkan proses pengembangan dan memungkinkan pengelolaan beban kerja secara efisien [1] Selain itu, penelitian oleh Nisar et al. menekankan pentingnya auto-scaling pada arsitekture multi-tier, khususnya dalam konteks aplikasi berbasis cloud. Mereka mengembangkan pendekatan kombinasi horizontal scaling pada lapisan web dan vertical scaling pada lapisan database untuk meningkatkan kinerja dan menjaga konsisten waktu respon sistem. Pendekatan ini relevan bagi pengembangan sistem surat desa, yang umumnya memiliki beban kerja yang bervarian tergantung pada pelayanan masyarakat [2] Lebih mendalam, penelitian dilakukan oleh varga dalam studinya mengenai arsitektur web multi-lapisan menyatakan arsitektur multi-layer atau multi-tier tidak hanya mendukung pengembangan sistem secara modular, tetapi juga mempermudah proses pengujian dan pemeliharaan sistem. Ia membagi arsitektur web ke lima lapisan, yaitu: presentasi, controller, repository, model, dan auxiliary, yang masing-masing memiliki tanggung jawab

spesifikasi dalam pengelolaan alur informasi [3] Penerapan pendekatan arsitektur ini juga telah dibuktikan dalam studi oleh pasha et al. Pengujian otomatis pada aplikasi berbasis multi-tier juga dinilai penting untuk menjaga integrasi antar lapisan sistem serta meningkatkan kualitas perangkat lunak secara keseluruhan [4]. Selain itu, penerapan arsitektur multi-tier mampu meningkatkan skalabilitas, keamanan, dan kemudahan pemeliharaan aplikasi berbasis web karena adanya pemisahan tanggung jawab pada setiap lapisan system [5] Penerapan keamanan pada setiap lapisan arsitektur multi-tier juga penting untuk menjaga stabilitas dan integritas aplikasi web agar sistem dapat berjalan lebih aman dan terstruktur [6]. Kondisi ini tentu sangat relevan diterapkan dalam pengembangan sistem administrasi surat menyurat desa yang memiliki struktur data dan logika bisnis yang jelas, serta memungkinkan pengembangan berkelanjutan [7]. pada sistem akademik di lingkungan pendidikan. Dalam studi tersebut, multi-tier architecture diterapkan untuk memisahkan logika bisnis dari antarmuka pengguna dan penyimpanan data. Hasilnya menunjukkan bahwa sistem menjadi lebih mudah dikembangkan kembali untuk kebutuhan lain tanpa harus membangun dari awal. Arsitektur multi-tier juga dapat meningkatkan efisiensi pengembangan sistem melalui pemisahan komponen yang lebih terorganisir sehingga proses pengembangan dan pemeliharaan menjadi lebih mudah [8]. Selain dari sisi teknis, kebutuhan di lapangan juga mendesak. Sistem pelayanan administrasi desa berbasis web dinilai mampu meningkatkan efisiensi pelayanan serta mempermudah masyarakat dalam proses pengajuan surat secara digital [9]. Adapun hasil wawancara dengan staf desa menunjukkan bahwa proses administrasi surat menyurat masih dilakukan secara manual, mengakibatkan keterlambatan layanan, kesalahan pencatatan, dan rentan kehilangan data. Dengan menerapkan multi-tier architecture, sistem dapat dibangun secara modular dan terstruktur, dimana setiap surat baik itu surat keterangan tidak mampu, surat keterangan domisili, dan surat keterangan usaha dikelola melalui modul yang saling terintegrasi namun terpisah secara logis. Dengan demikian, pendekatan multi-tier bukan hanya sekedar solusi teknis, tetapi juga merupan strategi arsitektual yang mampu meningkatkan kualitas layanan publik di tingkat desa. Penerapannya dalam sistem surat menyurat desa pakning asal dapat memberikan efisiensi tinggi untuk pengembangan sistem di masa mendatang. Sistem ini tidak hanya menjawab kebutuhan digitalisasi saat ini tetapi juga membentuk fondasi kuat untuk pelayanan publik yang adaptif dan berkelanjutan.

II. METODE PENELITIAN

Penelitian ini dilaksanakan dengan menggunakan pendekatan yang sistematis agar sistem yang dikembangkan dapat mendukung pengelolaan administrasi surat menyurat secara efektif dan efisien. Metodologi penelitian terdiri dari dua tahapan utama, yaitu melalui metode pengumpulan data serta metode pengembangan sistem.

A. Metode Pengumpulan Data

Metode pengumpulan data digunakan untuk mendapatkan informasi yang tepat mengenai kebutuhan sistem surat menyurat di Kantor Desa Pakning Asal. Teknik yang diterapkan dalam penelitian ini mencakup:

a. Observasi

Penulis melakukan pengamatan secara langsung terhadap proses pengelolaan surat menyurat yang masih dilakukan secara manual di Kantor Desa Pakning Asal. Kegiatan observasi ini bertujuan untuk menemukan berbagai permasalahan yang muncul pada sistem yang masih berjalan secara manual, seperti keterlambatan pelayanan, kesalahan pencatatan, serta kesulitan dalam pengelolaan arsip surat. Observasi juga dilakukan untuk memahami alur kerja pembuatan surat keterangan yang meliputi Surat Keterangan Tidak Mampu (SKTM), Surat Keterangan Domisili, dan Surat Keterangan Usaha (SKU).

b. Wawancara

Wawancara dilakukan dengan aparatur desa dan staf Kantor Desa Pakning Asal untuk mengumpulkan informasi terkait kebutuhan sistem, kendala yang dihadapi dalam proses pelayanan surat menyurat, serta harapan terhadap sistem yang akan dikembangkan. Keterlibatan pihak terkait sangat penting untuk memastikan sistem yang dibangun sesuai dengan kebutuhan pengguna. Wawancara juga dilakukan dengan masyarakat untuk memahami kesulitan yang mereka alami dalam mengurus surat secara manual.

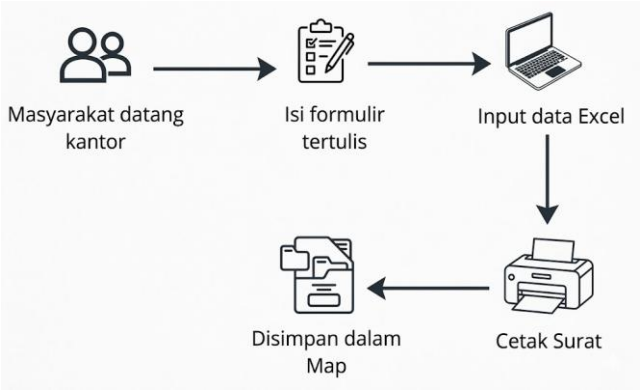
c. Studi Pustaka

Penulis melakukan studi literatur terhadap berbagai sumber seperti buku, jurnal, artikel ilmiah, dan dokumen lainnya yang berkaitan dengan arsitektur multi-tier, sistem informasi surat menyurat, pengembangan aplikasi berbasis web, serta teknologi yang digunakan seperti PHP, MySQL, dan XAMPP. Studi pustaka ini bertujuan untuk memperkuat landasan teori dan memahami penelitian-penelitian terdahulu yang relevan.

B. Perancangan

a. Analisis Sistem Berjalan

Berdasarkan hasil observasi dan wawancara di Kantor Desa Pakning Asal, proses surat menyurat masih dilakukan secara manual. Pengajuan dilakukan dengan datang langsung, mengisi formulir tertulis, dan pencatatan dilakukan lewat *Microsoft Excel*. Arsip surat disimpan dalam bentuk cetak. Proses ini rawan keterlambatan, kehilangan dokumen, dan sulit dalam pencarian arsip karena belum adanya sistem digital yang terintegrasi.

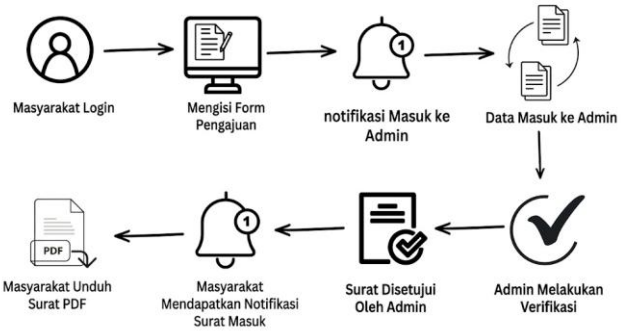


Gbr, 1 Gambaran Sistem Berjalan

b. Analisis Sistem Usulan

Sistem yang diusulkan bertujuan mendigitalisasi proses surat menyurat dengan pendekatan arsitektur *Multi-Tier*. Sistem akan dibagi ke dalam tiga lapisan utama:

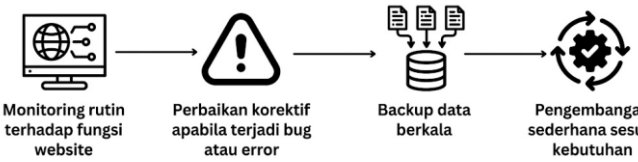
1. *Presentation Tier* (Lapisan Tampilan): tempat interaksi pengguna seperti login, pengisian formulir surat, dan notifikasi status.
2. *Logic Tier* (Lapisan Logika Aplikasi): tempat pemrosesan logika, seperti validasi form, otorisasi admin, dan pemrosesan status surat.
3. *Data Tier* (Lapisan Basis Data): menyimpan semua data pengguna, surat, riwayat pengajuan, dan log sistem.



Gbr, 2 Sistem Usulan

c. Rencana Pemeliharaan Sistem

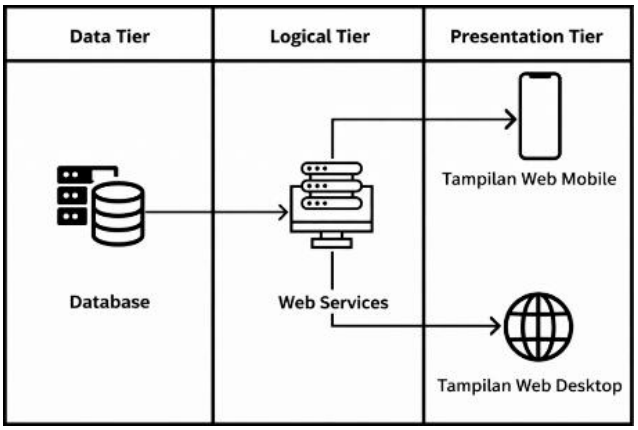
Pemeliharaan website administrasi masyarakat juga menjadi aspek yang sangat penting untuk menjaga keberlanjutan layanan, namun seringkali terkendala oleh keterbatasan pengetahuan pengurus dalam melakukan perawatan [10]. Hal ini menunjukkan bahwa sistem berbasis web tidak hanya membutuhkan implementasi, tetapi juga strategi pemeliharaan agar dapat terus digunakan. Oleh karena itu, penelitian ini merencanakan pemeliharaan sistem surat menyurat berbasis web melalui monitoring rutin terhadap fungsi sistem, perbaikan bug atau error secara korektif, backup data secara berkala, serta pengembangan sederhana apabila diperlukan. Dengan langkah-langkah tersebut, sistem dapat digunakan secara berkelanjutan dan mendukung pelayanan administrasi desa secara efektif.



Gbr, 3 Pemeliharaan Sistem

d. Penerapan Multi-Tier pada Website Surat Menyurat

Pengembangan sistem surat menyurat berbasis web ini menerapkan *Multi-Tier Architecture* yang membagi aplikasi ke dalam tiga lapisan utama yaitu, *Presentation Tier, Logic Tier*, dan *Data Tier*. Pemisahan ini dilakukan untuk memastikan setiap lapisan memiliki tanggung jawab yang jelas sehingga sistem menjadi terstruktur, mudah dikembangkan, dirawat, dan aman dari kesalahan berantai:



Gbr, 4 *Multi-tier*

C. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini mengacu pada pendekatan terstruktur dengan tahapan sebagai berikut:

a. Analisis Kebutuhan

Tahap ini dilakukan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional mencakup fitur-fitur yang harus tersedia seperti login, pengajuan surat, verifikasi, notifikasi, dan pengunduhan surat. Sedangkan kebutuhan non-fungsional meliputi aspek usability, compatibility, performance, security, dan maintainability.

b. Perancangan Sistem

Pada tahap ini dilakukan perancangan arsitektur sistem menggunakan pendekatan multi-tier architecture yang membagi sistem menjadi tiga lapisan utama: Presentation Tier: Lapisan antarmuka pengguna untuk interaksi dengan sistem Logic Tier: Lapisan logika bisnis yang menangani proses validasi dan pengolahan data Data Tier: Lapisan basis data untuk penyimpanan dan pengelolaan data Perancangan juga mencakup pembuatan Use Case Diagram dan Activity Diagram untuk memodelkan alur sistem.

c. Implementasi

Tahap implementasi dilakukan dengan membangun sistem berdasarkan rancangan yang telah dibuat menggunakan teknologi PHP Native, MySQL, dan XAMPP sebagai server lokal. Implementasi multi-tier architecture diterapkan dengan memisahkan kode program ke dalam folder-folder yang sesuai dengan tier-nya masing-masing.

d. *Pengujian dan Evaluasi*

Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan semua fungsi sistem berjalan sesuai dengan spesifikasi. Pengujian juga dilakukan untuk mengukur keberhasilan penerapan multi-tier architecture melalui analisis struktur folder dan pemisahan tanggung jawab setiap lapisan.

e. *Dokumentasi dan Kesimpulan*

Tahap akhir berupa dokumentasi seluruh proses penelitian dan penarikan kesimpulan berdasarkan hasil implementasi dan pengujian sistem.

III. HASIL DAN PEMBAHASAN

Hasil penelitian ini berfokus pada penerapan arsitektur Multi-Tier dalam sistem informasi surat menyurat berbasis web. Penerapan arsitektur tersebut diwujudkan melalui pemisahan sistem ke dalam tiga lapisan utama, yaitu Presentation Tier, Logic Tier, dan Data Tier, yang masing-masing dirancang untuk menangani fungsi tampilan, logika bisnis, dan pengelolaan data secara terpisah. Sebagai hasilnya, sistem memiliki struktur arsitektur yang jelas dan berlapis, di mana perubahan pada satu lapisan tidak berdampak langsung pada lapisan lainnya.

A. Penerapan Multi Tier Architecture

Penerapan arsitektur *multi-tier* pada sistem informasi surat menyurat berbasis web ini dilakukan dengan membagi aplikasi ke dalam tiga lapisan utama, yaitu *Presentation Tier, Logic Tier, dan Data Tier*. Pembagian ini bertujuan untuk menerapkan prinsip separation of concerns, sehingga setiap lapisan memiliki fungsi dan tanggung jawab yang jelas serta tidak saling tumpang tindih.

Berbeda dengan pendekatan pengembangan konvensional yang menempatkan seluruh komponen dalam satu struktur tanpa batasan lapisan yang tegas, arsitektur multi-tier menekankan pemisahan peran antara tampilan, pengolahan logika bisnis, dan pengelolaan data. Dengan demikian, sistem menjadi lebih terstruktur, mudah dipelihara, serta lebih fleksibel dalam proses pengembangan lanjutan.

Hasil implementasi menunjukkan bahwa perubahan pada satu lapisan, seperti penyesuaian tampilan antarmuka, tidak memengaruhi proses logika maupun pengelolaan data. Kondisi ini membuktikan bahwa penerapan arsitektur *multi-tier* pada sistem surat menyurat yang dikembangkan telah memenuhi karakteristik arsitektur berlapis, yaitu modularitas, independensi antar lapisan, dan kemudahan pengujian.

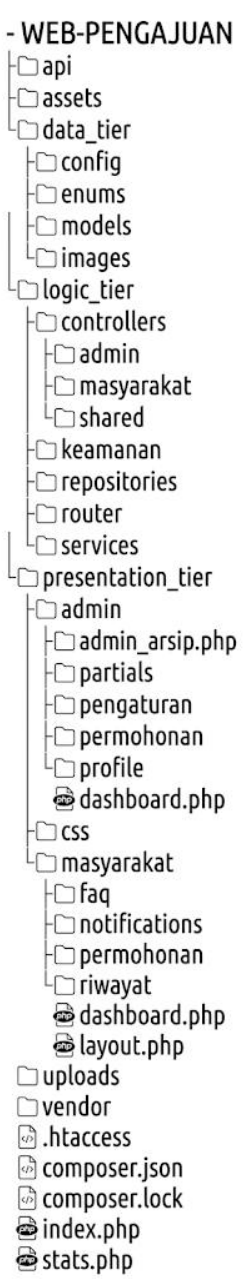
a. *Struktur Sistem Sebelum Menerapkan Multi Tier.*

Gbr, 5 Sebelum Menerapkan Multi Tier

Berdasarkan struktur tersebut dapat dilihat bahwa seluruh file ditempatkan dalam satu folder utama tanpa pembagian lapisan yang jelas. File tampilan seperti halaman dashboard berada dalam lokasi yang sama dengan file proses seperti proses_login serta file koneksi basis data. Dalam satu direktori yang sama terdapat kode HTML, proses validasi, hingga query database. Kondisi ini sering disebut sebagai pendekatan PHP spageti karena kode program bercampur dan tidak memiliki batas pemisah yang tegas antar fungsinya.

Struktur seperti ini menyebabkan tingkat ketergantungan antar file menjadi tinggi. Apabila terjadi perubahan pada satu bagian, misalnya perubahan struktur tabel pada database, maka beberapa file lain harus ikut disesuaikan karena saling berkaitan secara langsung. Selain menyulitkan proses pemeliharaan, kondisi tersebut juga berpotensi menimbulkan kesalahan saat dilakukan pengembangan fitur baru. Hal ini menunjukkan bahwa sebelum penerapan arsitektur berlapis, sistem masih memiliki modularitas yang rendah dan belum terorganisir secara optimal.

Struktur Sistem Sesudah Menerapkan Multi Tier



Gbr, 6 Setelah Menerapkan Multi Tier

Setelah menerapkan *multi tier architecture*, struktur sistem dibagi secara tegas ke dalam tiga lapisan utama. Presentation Tier difokuskan untuk menangani seluruh komponen antarmuka pengguna. Seluruh file tampilan ditempatkan pada folder presentation_tier dan dipisahkan berdasarkan peran pengguna yaitu admin dan masyarakat. Pada lapisan ini tidak ditemukan kode koneksi database maupun query secara langsung karena fungsinya hanya menampilkan data yang telah diproses oleh lapisan di bawahnya.

Logic Tier berperan sebagai pengatur alur sistem. Pada lapisan ini terdapat controller yang menerima permintaan dari pengguna serta service yang menangani proses bisnis utama. Repository juga ditempatkan sebagai penghubung antara logika dan data sehingga akses ke basis data tidak dilakukan secara langsung dari tampilan. Sementara itu, Data Tier berfokus pada representasi dan pengelolaan data melalui model, konfigurasi, dan komponen pendukung lainnya. Dengan pembagian ini, setiap lapisan memiliki tanggung jawab yang jelas dan bekerja sesuai perannya masing masing.

b. Analisis Perbandingan

Berdasarkan pengamatan terhadap struktur folder sebelum dan sesudah penerapan multi tier architecture, terlihat adanya perbedaan yang cukup jelas dalam pengelolaan dan penempatan komponen sistem. Pada struktur awal, file tampilan, proses, dan koneksi basis data berada dalam satu direktori yang sama tanpa adanya pembagian lapisan. Kondisi tersebut menunjukkan bahwa sistem belum memiliki pemisahan tanggung jawab yang tegas antar bagian. Hal ini dapat dilihat langsung dari keberadaan file tampilan dan file proses dalam satu lokasi yang sama.

Setelah dilakukan penerapan multi tier architecture, struktur sistem dibagi ke dalam tiga lapisan utama, yaitu presentation tier, logic tier, dan data tier. Pembagian ini terlihat secara eksplisit pada susunan folder yang terpisah sesuai fungsinya masing masing. File tampilan ditempatkan pada lapisan presentasi, proses bisnis berada pada lapisan logika, dan pengelolaan data difokuskan pada lapisan data. Berdasarkan perbedaan struktur tersebut, dapat disimpulkan bahwa sistem setelah penerapan arsitektur berlapis memiliki susunan yang lebih terorganisir dibandingkan sebelumnya.

Dengan adanya pemisahan ini, setiap perubahan dapat difokuskan pada lapisan yang bersangkutan tanpa harus memodifikasi seluruh bagian sistem. Kesimpulan ini diperoleh dari hasil implementasi dan pengamatan langsung terhadap struktur proyek yang dikembangkan dalam penelitian ini. Oleh karena itu, penerapan multi tier architecture pada sistem yang dibangun menunjukkan peningkatan dalam hal keteraturan struktur dan kejelasan pembagian tanggung jawab antar komponen.

B. Pengujian Sistem

Pengujian sistem dilakukan untuk memvalidasi bahwa seluruh fungsi dalam Sistem Administrasi Surat-Menyurat Desa Pakning Asal berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Pendekatan pengujian diterapkan pada setiap lapisan arsitektur, mulai dari presentation tier untuk validasi antarmuka, logic tier untuk pemrosesan aturan bisnis melalui *service*, hingga data *tier* untuk memastikan integritas penyimpanan data pada basis data.

Metode yang digunakan dalam proses ini adalah pengujian *black-box*, di mana pengujian difokuskan sepenuhnya pada fungsionalitas sistem melalui masukan (input) dan keluaran (output) tanpa meninjau struktur kode internal. Melalui pengujian ini, setiap fitur dipastikan dapat merespons interaksi pengguna secara akurat, menjaga alur data antar lapisan tetap sinkron, dan menjamin bahwa sistem siap dioperasikan untuk mendukung pelayanan administrasi desa secara transparan.

a. Tolak Ukur Pemisahan Tanggung Jawab pada Arsitektur Multi-Tier

Pada tahap ini dilakukan evaluasi teknis terhadap struktur perangkat lunak untuk memastikan bahwa penerapan arsitektur *multi-tier* tidak hanya bersifat konseptual, tetapi juga dapat dibuktikan secara terukur. Evaluasi difokuskan pada penilaian tingkat pemisahan tanggung jawab antar lapisan (*separation of concerns*), modularitas, serta keterpisahan fungsi antara *presentation layer, business logic layer,* dan *data access layer*. Arsitektur berlapis dirancang untuk membagi sistem ke dalam komponen yang memiliki tanggung jawab spesifik dan terdefinisi dengan jelas, sehingga mampu mengurangi ketergantungan langsung antar bagian sistem serta meningkatkan kualitas struktural perangkat lunak. Dengan adanya batasan tanggung jawab yang tegas, setiap lapisan dapat dikembangkan dan dipelihara secara independen tanpa memengaruhi lapisan lainnya secara signifikan [1].

Selain itu, kualitas arsitektur berlapis juga dapat dianalisis melalui indikator kuantitatif yang mencerminkan tingkat kompleksitas dan distribusi logika dalam sistem. Pengukuran terhadap ukuran kode, kompleksitas metode, serta pola ketergantungan antar komponen menjadi dasar dalam menilai apakah tanggung jawab sistem telah terdistribusi secara proporsional. Pendekatan ini selaras dengan prinsip evaluasi arsitektur modern yang menekankan pentingnya modularitas, maintainability, dan reusabilitas komponen [1]. Oleh karena itu, penelitian ini menerapkan metode static code analysis dengan memanfaatkan metrik seperti *Logical Lines of Code* (LLoC) dan analisis struktur komponen untuk memastikan bahwa implementasi arsitektur multi-tier benar-benar memenuhi prinsip pemisahan tanggung jawab secara kuantitatif.

Deskripsi dan Pemilihan Tools

Pada penelitian ini digunakan sebuah tools sederhana berupa script PHP bernama *stats.php* yang dibuat khusus untuk menganalisis struktur arsitektur sistem. Tools ini dijalankan melalui terminal dan berfungsi untuk memindai struktur folder berdasarkan pembagian *multi tier architecture*, yaitu *presentation tier, logic tier, dan data tier.*

Script ini menghitung jumlah file, jumlah class, jumlah method, jumlah baris kode, serta rata rata method per file pada setiap kategori. Selain itu, script juga memberikan analisis otomatis terkait beban controller, pemanfaatan *service*, serta pemisahan akses data melalui *repository*.

Tools ini dipilih karena sistem dikembangkan menggunakan PHP native tanpa framework, sehingga tidak tersedia fitur analisis arsitektur bawaan. Oleh karena itu, diperlukan alat bantu yang dapat memberikan data kuantitatif secara langsung dari struktur proyek.

1. Dasar Pemilihan Tools

Dasar pemilihan tools ini adalah untuk memperoleh bukti terukur mengenai penerapan arsitektur berlapis. Indikator yang digunakan dalam pengukuran meliputi:

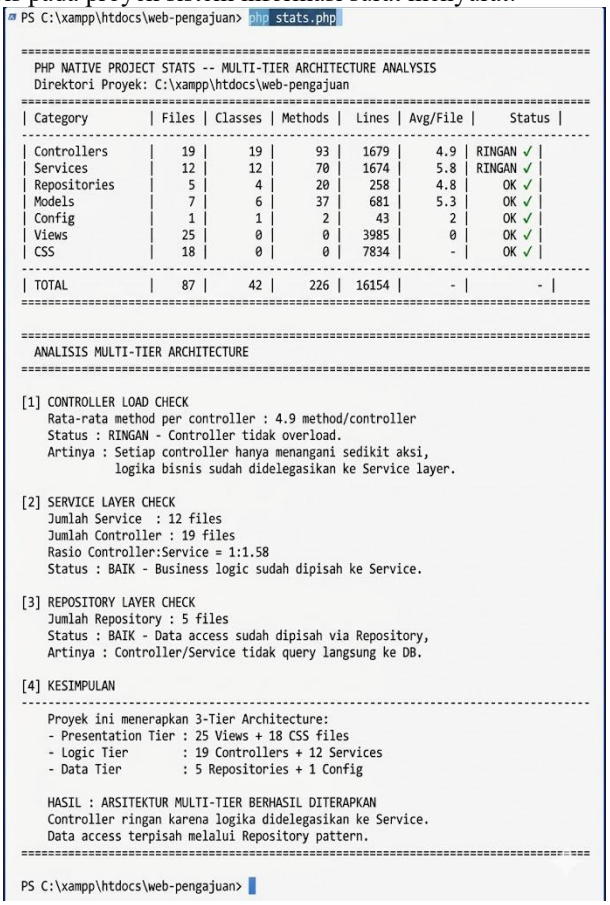
1). Rata rata method pada controller
2). Jumlah service dibandingkan jumlah controller
3). Jumlah repository sebagai pemisah akses data

Jika controller memiliki terlalu banyak method, maka dapat diasumsikan terjadi penumpukan logika. Sebaliknya, jika jumlah service dan repository memadai, maka menunjukkan bahwa logika bisnis dan akses data telah dipisahkan sesuai prinsip multi tier architecture.

Dengan demikian, tools ini dipilih karena mampu memberikan gambaran objektif mengenai struktur dan modularitas sistem.

2. Tahap Pengukuran

Pengukuran dilakukan dengan menjalankan perintah berikut pada terminal di dalam folder proyek: php stats.php Setelah perintah dijalankan, sistem secara otomatis memindai seluruh direktori dan menampilkan hasil analisis dalam bentuk tabel serta kesimpulan otomatis. Berikut adalah hasil eksekusi tools pada proyek sistem informasi surat menyurat:



```
PS C:\xampp\htdocs\web-pengajuan> php stats.php

================================================================================
  PHP NATIVE PROJECT STATS -- MULTI-TIER ARCHITECTURE ANALYSIS
  Direktori Proyek: C:\xampp\htdocs\web-pengajuan
================================================================================
| Category        | Files | Classes | Methods |  Lines | Avg/File |     Status |
--------------------------------------------------------------------------------
| Controllers     |    19 |      19 |      93 |   1679 |      4.9 | RINGAN ✓ |
| Services        |    12 |      12 |      70 |   1674 |      5.8 | RINGAN ✓ |
| Repositories    |     5 |       4 |      20 |    258 |      4.8 |      OK ✓ |
| Models          |     7 |       6 |      37 |    681 |      5.3 |      OK ✓ |
| Config          |     1 |       1 |       2 |     43 |        2 |      OK ✓ |
| Views           |    25 |       0 |       0 |   3985 |        0 |      OK ✓ |
| CSS             |    18 |       0 |       0 |   7834 |        - |      OK ✓ |
--------------------------------------------------------------------------------
| TOTAL           |    87 |      42 |     226 |  16154 |        - |          - |
================================================================================

================================================================================
  ANALISIS MULTI-TIER ARCHITECTURE
================================================================================

[1] CONTROLLER LOAD CHECK
    Rata-rata method per controller : 4.9 method/controller
    Status : RINGAN - Controller tidak overload.
    Artinya : Setiap controller hanya menangani sedikit aksi,
              logika bisnis sudah didelegasikan ke Service layer.

[2] SERVICE LAYER CHECK
    Jumlah Service  : 12 files
    Jumlah Controller : 19 files
    Rasio Controller:Service = 1:1.58
    Status : BAIK - Business logic sudah dipisah ke Service.

[3] REPOSITORY LAYER CHECK
    Jumlah Repository : 5 files
    Status : BAIK - Data access sudah dipisah via Repository,
    Artinya : Controller/Service tidak query langsung ke DB.

[4] KESIMPULAN
--------------------------------------------------------------------------------
    Proyek ini menerapkan 3-Tier Architecture:
    - Presentation Tier : 25 Views + 18 CSS files
    - Logic Tier        : 19 Controllers + 12 Services
    - Data Tier         : 5 Repositories + 1 Config

    HASIL : ARSITEKTUR MULTI-TIER BERHASIL DITERAPKAN
    Controller ringan karena logika didelegasikan ke Service.
    Data access terpisah melalui Repository pattern.
================================================================================

PS C:\xampp\htdocs\web-pengajuan>
```

Gbr, 7 mengukur keberhasilan

b. Hasil Pengukuran dan Bukti Keberhasilan Arsitektur

Berdasarkan hasil yang ditampilkan pada terminal, diperoleh data sebagai berikut:

Tabel Hasil Pengukuran

Kategori	File	Classes	Methods	Lines	Avg/File	Status
Controllers	19	19	93	1048	4.9	Ringan
Services	12	12	-	-	5.8	Ringan
Repositori	5	4	24	258	4.8	OK
Models	7	6	37	681	5.3	OK
Config	1	1	2	43	2	OK
Views	25	0	0	3985	0	OK
CSS	18	0	0	7834	0	OK
TOTAL	87	42	226	16154	-	-

Pada bagian analisis otomatis diperoleh hasil:

1. *Controller Load Check*
   Rata rata method per controller adalah 4.9 method dan dikategorikan ringan. Hal ini menunjukkan bahwa controller tidak mengalami overload dan hanya menangani sedikit aksi.
2. *Service Layer Check*
   Jumlah service sebanyak 12 file dan controller sebanyak 19 file dengan rasio 1:1.58. Status dinyatakan baik karena logika bisnis telah dipisahkan ke service layer.
3. *Repository Layer Check*
   Jumlah repository sebanyak 5 file dan dinyatakan baik karena akses data telah dipisahkan melalui repository. Hal ini menunjukkan bahwa controller maupun service tidak melakukan query langsung ke database.

Pada bagian kesimpulan otomatis, tools menyatakan bahwa proyek ini telah menerapkan 3 tier architecture dengan pembagian:

1. *Presentation Tier*: 25 Views dan 18 CSS
2. *Logic Tier*: 19 Controllers dan 12 Services
3. *Data Tier*: 5 Repositories dan 1 Config

Hasil akhir menunjukkan bahwa arsitektur multi tier berhasil diterapkan.

c. Analisis Hasil dan Kesimpulan

Berdasarkan hasil pengukuran yang diperoleh, dapat dianalisis bahwa sistem telah menerapkan prinsip pemisahan tanggung jawab secara jelas. Rata rata 4.9 method pada controller menunjukkan bahwa controller hanya berperan sebagai pengatur alur permintaan dan tidak memuat logika bisnis secara berlebihan.

Keberadaan 12 service menunjukkan bahwa logika utama aplikasi telah dipindahkan dari controller ke lapisan khusus. Selain itu, 5 repository yang tersedia menunjukkan bahwa akses data telah dipisahkan dari logika aplikasi, sehingga tidak terjadi query langsung pada controller.

Dengan demikian, penerapan multi tier architecture pada sistem informasi surat menyurat ini dapat dikatakan berhasil, karena didukung oleh struktur folder yang terpisah dan hasil pengukuran kuantitatif yang menunjukkan rendahnya beban controller serta adanya pemisahan logika dan akses data..

IV. KESIMPULAN DAN SARAN

A. Kesimpulan.

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa penerapan arsitektur multi-tier pada sistem surat menyurat di Kantor Desa Pakning Asal berhasil diterapkan dengan baik. Pemisahan sistem ke dalam lapisan presentasi, lapisan logika aplikasi, dan lapisan akses data mampu menghasilkan struktur aplikasi yang lebih terorganisir dan memiliki pembagian tanggung jawab yang jelas pada setiap tier. Pengujian yang dilakukan pada masing-masing tier menunjukkan bahwa fungsi antarmuka, proses bisnis, dan pengelolaan data berjalan sesuai dengan kebutuhan pengguna. Penerapan arsitektur multi-tier juga mempermudah proses pemeliharaan dan pengembangan sistem di masa mendatang serta meningkatkan efisiensi pelayanan administrasi surat menyurat di lingkungan Kantor Desa Pakning Asal

B. Saran.

Berdasarkan hasil penelitian yang telah dilakukan, penerapan arsitektur multi-tier pada sistem surat menyurat di Kantor Desa Pakning Asal masih dapat dikembangkan lebih lanjut. Pengembangan disarankan untuk memperkuat aspek keamanan sistem pada setiap tier, terutama dalam pengelolaan hak akses pengguna dan perlindungan data. Selain itu, diperlukan pengujian lanjutan terhadap kinerja sistem untuk mengetahui kemampuan sistem dalam menangani peningkatan jumlah pengguna. Sistem juga dapat dikembangkan dengan menambahkan fitur manajemen admin guna memudahkan pengelolaan pengguna dan pengaturan hak akses secara terpusat. Dengan pengembangan tersebut, penerapan arsitektur multi-tier diharapkan dapat meningkatkan keandalan dan kualitas layanan sistem.

DAFTAR PUSTAKA

[1] A. Singh, “Evolutionary Architectures in Web Applications: A Comprehensive Study of Client-Server, Multi-Tier, and Service-Oriented Approaches,” *International Journal for Multidisciplinary Research (IJFMR)*, vol. 6, pp. 1–12, 2024, doi: 10.13140/RG.2.2.25457.49767.

[2] A. Nisar, W. Iqbal, F. Bokhari, F. Bukhari, and K. Almustafa, “Dynamic horizontal and vertical scaling for multi-tier web applications,” *Intelligent Automation and Soft Computing*, vol. 26, no. 2, pp. 353–365, 2020, doi: 10.31209/2019.100000159.

[3] M. Varga, “Web Programming and Multi-Tier Architecture of Web Applications,” *TEM Journal*, vol. 13, no. 4, pp. 3286–3294, Nov. 2024, doi: 10.18421/TEM134-63.

[4] S. U. Optum and Y. S. Mikhailovich, “Automated Testing Of Web Applications With Multi-Tier Architecture,” 2024.

[5] A. Rida and A. Ait Lahcen, “Towards DevSecOps Model for Multi-tier Web Applications,” *ITM Web of Conferences*, vol. 69, p. 04018, 2024, doi: 10.1051/itmconf/20246904018.

[6] A. Ramsingh, “Improving Interoperability in Distributed Multi-Tier Software Stacks,” 2022.

[7] M. Pasha, A. H. Mirza, and R. Andryani, “Penerapan Arsitektur Multi-Tier Pada Sistem Informasi Akademik (SIA) Sma Negeri 7 Prabumulih,” 2020. [Online]. Available: https://journal-computing.org/index.php/journal-sea/index

[8] D. Sereda, “Creating a Multi-tier Architecture for Web Applications: Design and Implementation,” 2025, [Online]. Available: https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

[9] A. Nanda, M. Jamil, A. Khoirunnita, and R. Alex, “Digitalisasi Pelayanan Administrasi Desa Menggunakan Website E-surat Pada Desa Handil

Terusan," *Ilmu Komputer Untuk Masyarakat*, vol. 5, no. 2, pp. 60–69, 2024.

[10] N. Nurmalasari, S. Masturoh, R. Lusiana Pratiwi, and F. Aziz, "Pengabdian Masyarakat Mengembangkan Keberlanjutan dalam Pemeliharaan Situs Web Sistem Administrasi RW013 Cipinang Melayu," *Jurnal Pengabdian Masyarakat Indonesia*, vol. 4, no. 3, pp. 491–495, Jun. 2024, doi: 10.52436/1.jpmi.2348.