

Pembuatan Aplikasi Sistem Manajemen File Sederhana Berbasis Client Server pada Lingkungan Jaringan Puskesmas Melur

Satria Julhendra¹ Muhammad Haikal Fikri² Bayu Samudra³ Anandaku Aksana⁴ Linna Oktaviana Sari⁵

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Riau, Kota Pekanbaru, Provinsi Riau, Indonesia^{1,2,3,4,5}

Email: satria.julhendra2730@student.unri.ac.id¹ muhhammad.haikal5643@student.unri.ac.id² bayu.samudra3601@student.unri.ac.id³ anandaku.aksana5273@student.unri.ac.id⁴ linnaoasari@lecturer.unri.ac.id⁵

Abstrak

Pengelolaan informasi di institusi pelayanan kesehatan, seperti Puskesmas Melur, menjadi fokus penting untuk mendukung pelayanan yang efektif dan efisien. Tantangan utama adalah mengatur sistem manajemen berkas yang memfasilitasi pengumpulan, penyimpanan, dan pertukaran informasi antar unit. Penelitian ini mengadopsi pendekatan *client-server* dengan menggunakan Java sebagai bahasa pemrograman. Melalui metode pengembangan waterfall, hasil penelitian mencakup analisis kebutuhan, desain arsitektur sistem, dan implementasi aplikasi manajemen *file*. Dalam pengujian, aplikasi memenuhi kebutuhan pengguna dan beroperasi secara baik dalam lingkungan jaringan Puskesmas. Aplikasi ini dapat meningkatkan efektivitas manajemen *file* dengan mempermudah pengiriman dan pengelolaan *file*, termasuk dokumen medis dan administratif. Kesimpulan penelitian menunjukkan bahwa aplikasi ini memberikan kontribusi positif dalam meningkatkan produktivitas dan kualitas layanan kesehatan di Puskesmas Melur.

Kata Kunci: Sistem Manajemen *File*, *Client-Server*, Pemrograman Java, Pemrograman Jaringan



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

PENDAHULUAN

Pengelolaan informasi di institusi pelayanan kesehatan, seperti Puskesmas Melur, merupakan aspek penting dalam mendukung pelayanan yang efektif dan efisien. Salah satu tantangan dalam konteks ini adalah bagaimana mengatur sistem manajemen berkas yang memfasilitasi pengumpulan, penyimpanan, dan pertukaran informasi antar unit dalam jaringan Puskesmas. Sistem manajemen *file* berbasis *client-server* merupakan solusi potensial untuk mengatasi kendala tersebut. Dengan mengadopsi teknologi ini, Puskesmas Melur dapat mengatur, menyimpan, dan mengakses data pasien, laporan medis, dan informasi kesehatan lainnya secara lebih terstruktur. Penelitian ini bertujuan untuk mengembangkan aplikasi sistem pengelolaan *file* sederhana berbasis *client-server* yang dapat diimplementasikan di Puskesmas Melur.

Pendekatan ini menggunakan model arsitektur *client-server* dimana klien bertindak sebagai stasiun peminta atau pengakses, sedangkan server berfungsi sebagai penyedia layanan atau sumber daya. Kedua entitas ini akan berkomunikasi melalui jaringan lokal Puskesmas menggunakan protokol TCP/IP, sehingga menjamin keamanan dan keandalan dalam pertukaran data. Pemilihan bahasa pemrograman Java sebagai kerangka aplikasi ini didasarkan pada keserbagunaannya tidak bergantung pada *platform*, memberdayakan aplikasi untuk berjalan secara konsisten di berbagai sistem operasi yang digunakan di Puskesmas Melur. Implementasi *socket* Java akan memperlancar pertukaran data antara klien dan *server*. Diharapkan bahwa aplikasi yang dihasilkan dari penelitian ini dapat memberikan kontribusi positif dalam meningkatkan efektivitas manajemen *file* di Puskesmas Melur. Kemudahan dalam

pengiriman dan pengelolaan *file* teks, termasuk dokumen medis dan administratif, diharapkan dapat meningkatkan produktivitas dan kualitas layanan kesehatan di lingkungan ini.

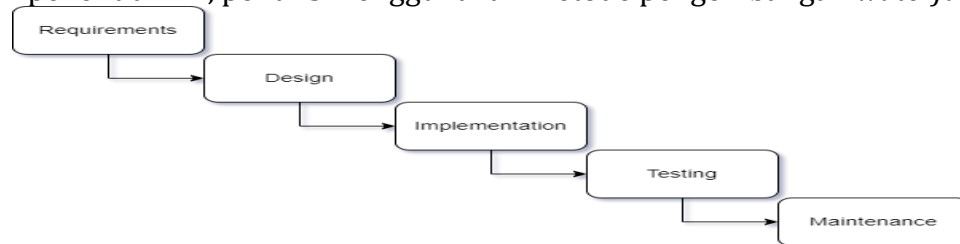
Tinjauan Pustaka

1. Konsep Aplikasi. Konsep aplikasi adalah ide atau gagasan dasar yang mendasari pengembangan suatu aplikasi. Konsep aplikasi dapat mencakup berbagai aspek, seperti tujuan aplikasi, fitur dan fungsionalitas aplikasi, target pengguna aplikasi, dan lingkungan aplikasi (Suhardi, 2022). Konsep aplikasi adalah hal yang penting dalam pengembangan aplikasi. Konsep aplikasi yang jelas akan membantu pengembang dalam menentukan tujuan, fitur, dan fungsionalitas aplikasi. Konsep aplikasi yang jelas juga akan membantu pengembang dalam berkomunikasi dengan pengguna dan pemangku kepentingan lainnya.
2. Sistem. Sistem adalah kumpulan elemen yang saling berhubungan dan saling berinteraksi untuk mencapai tujuan tertentu. Elemen-elemen dalam sistem dapat berupa komponen fisik, seperti perangkat keras dan perangkat lunak, atau komponen *non*-fisik, seperti data, informasi, dan manusia. Sistem adalah kumpulan elemen-elemen yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu (Azhar, 2023).
3. Management *File*. Manajemen *file* adalah proses mengatur dan mengelola penyimpanan *file* dalam sistem komputer. Manajemen *file* mencakup berbagai aspek, seperti penyimpanan *file*, akses *file*, penggunaan *file*, dan keamanan *file* (Kurniawan, 2022).
4. *Socket*. *Socket* adalah titik antarmuka antara proses dan jaringan yang digunakan untuk komunikasi antara proses-proses yang berjalan di komputer yang berbeda atau di komputer yang sama (Syahputra, 2021).
5. *Client-Server*. *Client-server* adalah sebuah arsitektur sistem yang membagi sistem menjadi dua komponen utama, yaitu *client* dan *server*. *Client* adalah komponen yang meminta layanan, sedangkan *server* adalah komponen yang menyediakan layanan (Azhar, 2023). *Client-server* adalah arsitektur perangkat lunak yang terdiri dari dua komponen utama, yaitu *client* dan *server*. *Client* adalah komponen yang meminta layanan dari *server*, sedangkan *server* adalah komponen yang menyediakan layanan kepada *client* (Dwi, 2023).
6. Java. Java adalah bahasa pemrograman berorientasi objek yang dirancang untuk dapat berjalan di berbagai platform sistem operasi (Ayu, 2023). Java adalah bahasa pemrograman yang dirancang untuk dapat dijalankan di berbagai platform perangkat keras dan sistem operasi. Java dikembangkan oleh Sun Microsystems pada tahun 1995 dan dirilis ke publik pada tahun 1996 (Budi, 2023).
7. Java Swing. Java Swing adalah kerangka kerja GUI yang digunakan untuk mengembangkan aplikasi desktop Java. Swing menyediakan komponen GUI yang dapat digunakan untuk membuat antarmuka pengguna yang responsif dan dapat disesuaikan (Putra, 2022).
8. Netbeans. NetBeans adalah IDE yang gratis dan sumber terbuka yang digunakan untuk mengembangkan aplikasi Java, *web*, dan *mobile*. NetBeans menyediakan berbagai fitur yang memudahkan pengembangan aplikasi, seperti dukungan untuk berbagai bahasa pemrograman, *framework*, dan teknologi (Budiarta, 2021).
9. Metode Pengembangan *Waterfall*. Metode pengembangan *waterfall* adalah sebuah model pengembangan perangkat lunak yang membagi proses pengembangan menjadi beberapa fase yang berurutan, sehingga setiap fase harus diselesaikan sebelum fase berikutnya dimulai (Budiarta, 2021).
10. Arsitektur *Client Server*. Arsitektur *client-server* adalah model arsitektur perangkat lunak yang membagi tugas antara komputer klien dan komputer *server*. Klien adalah komputer yang meminta layanan dari *server*, sedangkan *server* adalah komputer yang menyediakan layanan kepada klien (O'Brien, 2022).

11. *Use Case Diagram*. *Use case diagram* adalah sebuah diagram yang menggambarkan interaksi antara actor dengan *use case*. *Actor* adalah pengguna sistem, sedangkan *use case* adalah fungsionalitas yang disediakan sistem (Ayu, 2023).
12. *Flowchart*. *Flowchart* adalah sebuah diagram yang menggambarkan aliran data dan proses dalam suatu sistem. *Flowchart* dapat digunakan untuk menggambarkan berbagai jenis proses, mulai dari proses yang sederhana hingga proses yang kompleks (Ayu, 2023).

METODE PENELITIAN

Dalam penelitian ini, penulis menggunakan metode pengembangan *waterfall*.



Gambar 1. Metode *Waterfall*

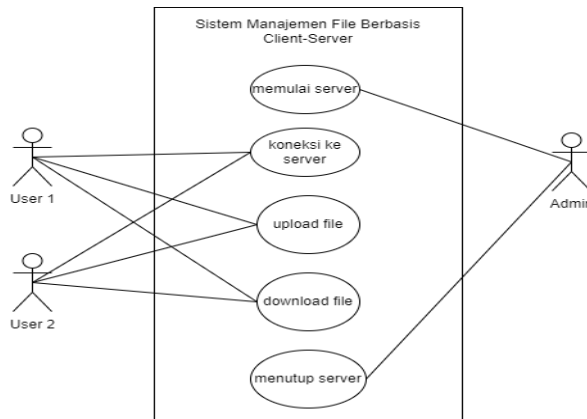
Tahapan metode *waterfall*:

1. *Analisa Kebutuhan*. Untuk mendapatkan gambaran jelas mengenai apa yang menjadi dasar pembuatan aplikasi ini maka dilakukan wawancara terhadap pihak puskesmas. Pihak yang diwawancarai adalah penanggung jawab di bagian administrasi puskesmas dan beberapa staf yang sering bertukar data dengan pihak administrasi. Selanjutnya dilakukan Observasi untuk mendapatkan gambaran sistem seperti apa yang akan dirancang dan Apa saja keperluan yang diharapkan serta bagaimana sistem yang telah ada saat ini bekerja.
2. *Design*. Pada tahapan desain peneliti akan menggambarkan semua kebutuhan sebuah rancangan sistem yang akan dibuat, seperti desain alur sistem. Dari desain tersebut akan dilakukan peneliti untuk melakukan aktivitas pembuatan sistem.
3. *Implementation*. Dalam tahap implementasi ini, peneliti menerjemahkan kedalam bahasa komputer sesuai dengan perancangan sistem dari desain yang telah dibuat, yaitu dengan mengimplementasikan ke dalam coding menggunakan bahasa pemrograman Java.
4. *Integration dan Testing*. Pengujian dilakukan dengan menggunakan metode *black box testing*. Pengujian ini dilakukan oleh tim penguji yang terdiri dari staf Puskesmas Melur dan mahasiswa. Kriteria keberhasilan pengujian adalah aplikasi dapat memenuhi semua kebutuhan pengguna dan dapat bekerja dengan baik pada lingkungan jaringan Puskesmas.
5. *Maintenance*. Tahap akhir dalam metode *waterfall*. Peneliti melakukan pemeliharaan sistem yang sudah dibuat yang sudah melewati tahap *testing*. Dan pemeliharaan ini termasuk dalam memperbaiki kesalahan yang tidak ditemukan dalam tahap sebelumnya.

HASIL PENELITIAN DAN PEMBAHASAN

1. *Analisis Kebutuhan*. Berdasarkan hasil wawancara dan observasi, ditemukan beberapa kebutuhan pengguna terhadap aplikasi sistem manajemen *file*. Secara fungsional, pengguna membutuhkan kemampuan untuk mengunggah dan mengunduh *file* dengan pengguna lain di dalam jaringan Puskesmas. Secara *non-fungsional*, pengguna menginginkan aplikasi yang mudah digunakan tanpa pelatihan khusus, dapat beroperasi dalam lingkungan jaringan komputer, dan dapat diintegrasikan dengan sistem informasi yang sudah ada di Puskesmas, seperti sistem informasi manajemen pasien. Kebutuhan-kebutuhan tersebut akan menjadi acuan dalam perancangan dan implementasi aplikasi sistem manajemen file sederhana berbasis *client-server* pada lingkungan jaringan Puskesmas Melur.

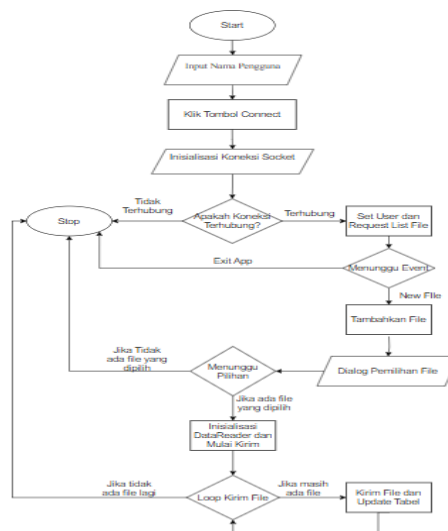
2. **Arsitektur Sistem.** Arsitektur sistem yang digunakan dalam aplikasi sistem manajemen *file* sederhana berbasis *client-server* pada lingkungan jaringan Puskesmas Melur adalah arsitektur *client-server*. Arsitektur ini membagi sistem menjadi dua komponen utama, yaitu klien dan server.
3. **Use Case Diagram.** Aplikasi *Management file* pada Puskesmas melur yang dirancang ini aktor yang berperan aktif dalam sistem ini adalah karyawan sebagai aktor yang berperan menjadi *client*.



Gambar 2. Use Case Diagram

Proses dimulai dengan admin yang memulai *server*. Ini adalah langkah penting pertama dalam manajemen *file*. *Server* diinisialisasi dan siap menerima koneksi dari pengguna. Setelah *server* dimulai, mencoba untuk terhubung ke *server*. Setelah berhasil terhubung ke *server*, pengguna dapat memilih *file* dari sistem lokal mereka dan mengunggahnya ke *server*. *File* ini kemudian disimpan di *server* dan tersedia untuk diakses oleh pengguna lain. Setelah *file* berhasil diunggah, *file* tersebut menjadi tersedia di *server*. Pengguna dapat mengunduh *file* pengguna lain. Mereka dapat memilih *file* yang ingin diunduh. Setelah semua operasi selesai, admin memiliki opsi untuk menutup *server*. Ini adalah langkah akhir dalam proses manajemen *file*. Menutup *server* akan mengakhiri semua koneksi aktif dan menghentikan semua layanan *server*. Dari pemodelan dengan use case diagram diatas dapat diketahui bahwa *client* berperan aktif dalam program ini. dimana *client* dapat mengirim dan mengunduh *file* dengan pengguna lain di dalam jaringan Puskesmas.

4. **Flowchart**



Gambar 3. Flowchart Diagram

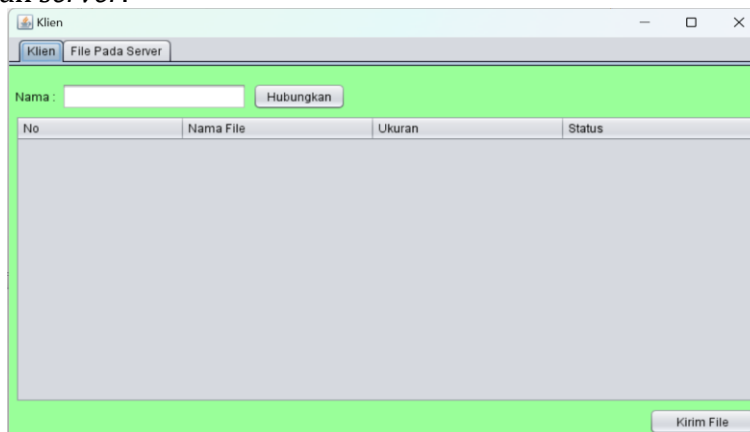
Aplikasi dimulai dengan inialisasi *server socket* untuk mendengarkan koneksi. Setelah inialisasi, *server* berada dalam mode menunggu, memantau koneksi yang masuk, dan saat klien terhubung, *server* menerima koneksi menggunakan metode *accept()*. Selanjutnya, server dan klien membuka input dan output streams untuk memungkinkan pertukaran data selama sesi koneksi. *Server* kemudian menerima permintaan dari klien, seperti permintaan untuk mengirim atau menerima *file*, dan berdasarkan jenis permintaan yang diterima, *server* mengidentifikasi dan menjalankan operasi yang sesuai. Proses pengiriman atau penerimaan *file* antara *server* dan klien dilakukan melalui input/output streams. Setelah operasi selesai, langkah konfirmasi dan penanganan kesalahan dilakukan. Jika terjadi kesalahan, langkah ini menangani penanganan kesalahan. Selanjutnya, koneksi antara server dan klien ditutup setelah semua operasi selesai. Proses penutupan koneksi memastikan pembebasan sumber daya yang digunakan selama sesi koneksi. Akhirnya, langkah terakhir menunjukkan bahwa aplikasi telah menyelesaikan prosesnya dan siap untuk mengakhiri eksekusi. Implementasi dari langkah-langkah ini akan memerlukan pemahaman yang baik tentang konsep Socket Programming dan detail-detail teknis dalam bahasa pemrograman Java.

5. Implementasi. Berdasarkan analisis kebutuhan yang telah dilakukan, didapatkan kesimpulan akhir bahwa pengguna ingin sebuah manajemen *file* sederhana untuk mengirim dan mengunduh *file*. Perancangan aplikasi dijalankan sesederhana mungkin agar mudah dipahami oleh pengguna namun tetap memenuhi keinginan pengguna. Pada awal aplikasi berjalan, server harus terlebih dahulu dinyalakan agar dapat menerima seluruh *file* yang akan diunggah oleh pengguna.



Gambar 4. Tampilan Awal Server

Setelah *server* berjalan, pengguna diminta untuk mengisi nama yang nantinya akan ditampilkan pada *server*, kemudian pengguna mengklik tombol "Hubungkan" untuk dapat terhubung dengan *server*.



Gambar 5. Tampilan Awal Klien



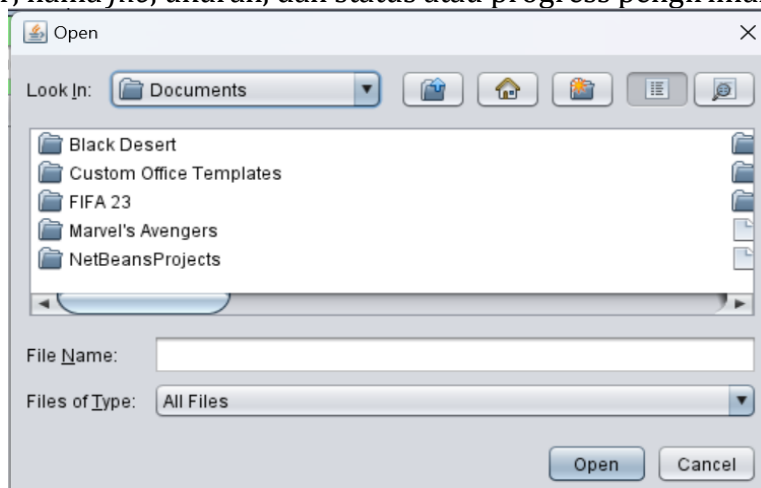
Gambar 6. Tampilan Klien Saat Menghubungi Server

Pada server, terlihat nomor urutan pengguna yang terhubung dengan server beserta dengan nama pengguna. Server hanya dapat menampilkan pengguna yang terhubung, sedangkan yang sudah tidak terhubung tidak dapat ditampilkan atau tidak menyisakan riwayat selain file yang telah diunggah saat pengguna terhubung sebelumnya.

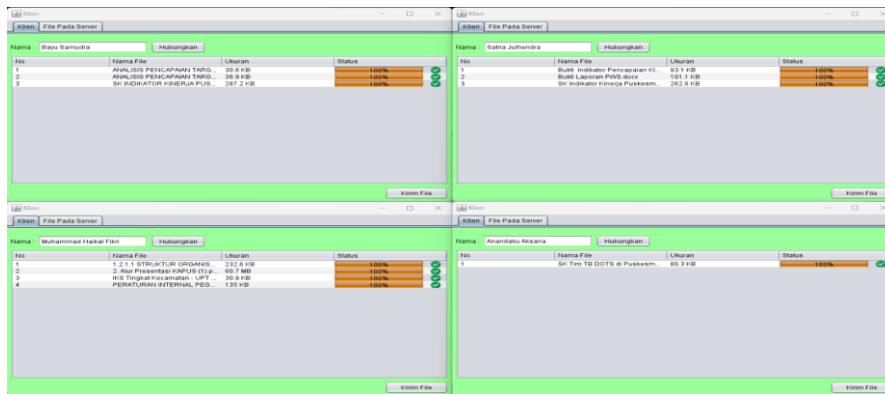


Gambar 7. Tampilan Server Setelah Terhubung Dengan Klien

Pengguna selanjutnya dapat mengirim file ke server dimana akan muncul notifikasi untuk memilih file pada komputer pengguna. Selanjutnya pada antarmuka pengguna akan ditampilkan nomor, nama file, ukuran, dan status atau progress pengiriman file.



Gambar 8. Notifikasi Pemilihan File Pada Komputer Pengguna

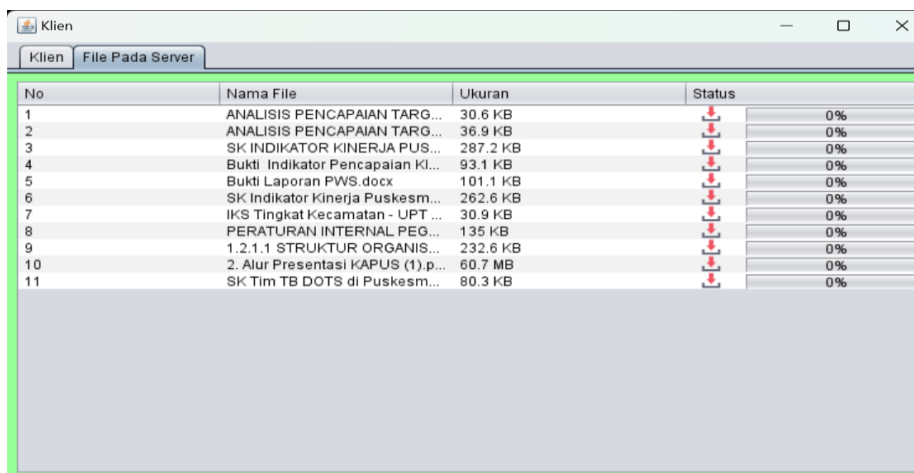


Gambar 9. Tampilan Klien Saat Mengirim File ke Server

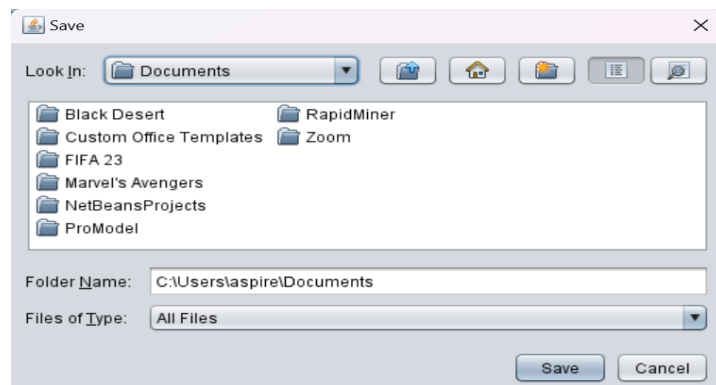
File yang diunggah pengguna ke *server*, dapat terlihat di dalam *server*. Pada *server* ditampilkan nama *file*, ukuran *file*, dan status pengiriman *file* oleh pengguna. Selain itu *file* yang telah terunggah ke *server* juga dapat diakses oleh pengguna lain di menu “*File pada Server*”. Pada menu tersebut pengguna dapat mengunduh *file* yang ada pada *server*.



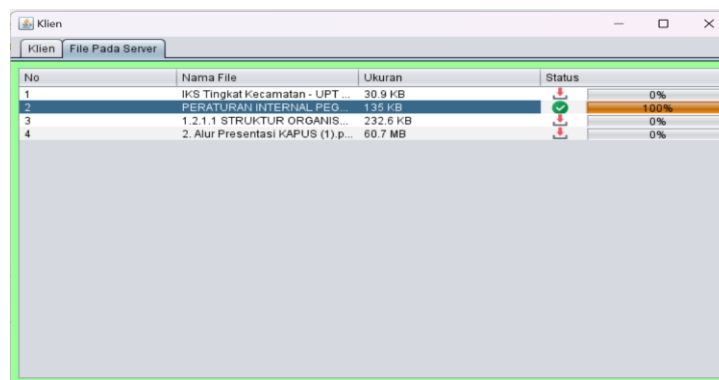
Gambar 10. Tampilan File Yang Telah Terupload di Dalam Server



Gambar 11. Tampilan Klien Saat Mengakses File Yang Ada Pada Server



Gambar 12. Notifikasi Simpan *File* Saat Ingin Mengunduh *File* Pada *Server*



Gambar 13. Tampilan Saat Berhasil Mengunduh *File* Dari *Server*

KESIMPULAN

Aplikasi manajemen *file* berbasis *client-server* yang dikembangkan dalam penelitian ini merupakan solusi yang efektif untuk mengatasi tantangan dalam pengelolaan *file* di institusi pelayanan kesehatan terkhusus pada Puskesmas Melur. Aplikasi ini dapat membantu petugas administrasi dan staf untuk mengunggah dan mengunduh *file* dengan lebih efisien, akurat, dan aman. Selain itu aplikasi ini juga mudah untuk digunakan dengan antarmuka yang mudah dipahami oleh segala macam pengguna.

DAFTAR PUSTAKA

- Ayu Lestari, I. G. A. N., Wijaya, I. G. M. A., & Sutarta, I. M. (2023). Pengembangan Aplikasi *Chatting* Menggunakan Java. *Jurnal Teknologi Informasi dan Komunikasi*, 15(1), 1-12.
- Azhar Susanto. (2023). *Sistem Informasi Manajemen* (Edisi 10). Jakarta: Salemba Empat.
- Budi Sutedjo. (2023). *Pemrograman Java: Pemrograman Berorientasi Objek* (Edisi 10). Jakarta: Penerbit Erlangga.
- Budiarta, I. M., Wijaya, I. G. M. A., & Sutarta, I. M. (2021). Pengembangan Aplikasi Pembayaran Listrik Online Menggunakan Java. *Jurnal Teknologi Informasi dan Komunikasi*, 14(2), 1-12.
- Dwi Astuti, T., Kurniawan, A., & Sutarta, I. M. (2023). Pengembangan Aplikasi *Client-Server* Menggunakan Metode *Waterfall*. *Jurnal Teknologi Informasi dan Komunikasi*, 16(1), 1-12.
- Kurniawan, A., Wijaya, I. G. M. A., & Sutarta, I. M. (2022). Pengembangan Aplikasi Manajemen *File* Menggunakan Metode *Waterfall*. *Jurnal Teknologi Informasi dan Komunikasi*, 15(1), 13-24.
- O'Brien, J. A., & Marakas, G. M. (2022). *Management Information Systems: Managing the Digital Firm* (16th ed.). Boston: McGraw-Hill Education.

- Suhardi, I. G. A. P., Wijaya, I. G. M. A., & Sutarta, I. M. (2022). Pengembangan Aplikasi *Mobile* untuk Meningkatkan Literasi Keuangan. *Jurnal Teknologi Informasi dan Komunikasi*, 15(1), 1-12.
- Syahputra, M., Kurniawan, A., & Sutarta, I. M. (2021). Pengembangan Aplikasi *Chatting* Menggunakan *Socket Programming*. *Jurnal Teknologi Informasi dan Komunikasi*, 14(1), 1-12.