

Sistem Administrasi Jadwal Pemeriksaan Rumah Sakit

Adam Yogisyah Putra
Dep. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada

AHMAD ALI MASYKUR
Dep. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada

Muhammad Arsyah Putra
Dep. Ilmu Komputer dan Elektronika
Universitas Gadjah Mada

I. INTRODUCTION

A. Latar Belakang

Rumah Sakit sebagai salah satu sub sistem pelayanan kesehatan menyelenggarakan dua jenis pelayanan, yaitu pelayanan kesehatan dan pelayanan administrasi. Pelayanan kesehatan meliputi pelayanan medik, penunjang medik, rehabilitasi medik, dan layanan keperawatan. Keempat jenis pelayanan tersebut dilaksanakan Unit Pelayanan Teknis seperti Unit Gawat Darurat, Unit Rawat Jalan, Unit Rawat Inap, Unit Transfusi Darah, unit Farmasi, dan sebagainya. Pelayanan administrasi mencakup semua jenis pelayanan yang bersifat administratif, termasuk administrasi keuangan yang fungsi utamanya adalah membantu kelancaran pelaksanaan pelayanan kesehatan. [1]

Pada perkembangannya saat ini, istilah sistem informasi lebih mengarah pada sistem informasi berbasis komputer yaitu sistem informasi yang memiliki perangkat keras dan lunak komputer, jaringan telekomunikasi (semacam internet ataupun lainnya), teknik manajemen sumber data berbasiskan komputer dan teknologi informasi yang mentransformasi sumber data tersebut menjadi berbagai variasi produk informasi yang siap digunakan. [2]

Sistem Informasi Rumah Sakit adalah suatu sistem teknologi informasi komunikasi yang memproses dan mengintegrasikan seluruh alur proses pelayanan Rumah Sakit dalam bentuk jaringan koordinasi, pelaporan dan prosedur administrasi untuk memperoleh informasi secara tepat dan akurat, dan merupakan bagian dari Sistem Informasi Kesehatan. Sistem ini memiliki 2 fungsi utama, yaitu untuk keperluan manajemen dan pengolahan data pasien.

Salah satu sub-sistem dari sistem informasi rumah sakit adalah sistem administrasi jadwal pemeriksaan antara pasien dengan dokter. Sistem ini diperlukan untuk mempermudah kerja administrasi rumah sakit dalam penjadwalan pemeriksaan dan manajemen data dokter dan pasien.

B. Related Work

Tersedia beberapa aplikasi yang memberikan layanan administrasi sistem informasi rumah sakit seperti rekmed.com dan open-emr. Kedua aplikasi tersebut menyediakan fitur seperti manajemen penjadwalan pemeriksaan, input dan penyimpanan data rekam medis pasien, dan pendaftaran pasien.

C. Proposed approach

Perlu adanya sebuah sistem yang menghubungkan pihak dokter dan pasien dalam membuat jadwal pemeriksaan. Hal ini agar proses pembuatan jadwal pemeriksaan lebih mudah dan sederhana serta mengantisipasi terjadinya jadwal pemeriksaan yang bertabrakan. Sistem ini dapat mempermudah staff administrasi dalam melakukan pencatatan rekam medis pasien. Sistem informasi kesehatan dapat membantu efisiensi dan mempermudah pekerjaan staff administrasi dalam mengelola data administrasi rumah sakit.

Terdapat 3 user class yang akan menggunakan sistem ini, yaitu user dokter, pasien, dan admin. Sistem perlu menyimpan data diri dokter seperti spesialis dan jam praktik dokter. Sistem perlu menyediakan fungsionalitas untuk user dokter seperti melihat jadwal pemeriksaan, menyetujui permintaan pemeriksaan, dan merubah data diri dokter.

User class pasien memerlukan fungsionalitas seperti melihat list dokter tersedia, membuat permintaan pemeriksaan, dan melihat status permintaan pemeriksaan. Sistem juga perlu menyediakan layanan pembuatan akun untuk setiap pengguna.

Administrator sistem dapat melihat daftar akun pengguna sistem dan memblokir akun pengguna yang melanggar ketentuan sistem rumah sakit. Administrator juga dapat menyetujui permintaan pembuatan akun dokter agar semua akun dokter yang terdapat dalam sistem telah terverifikasi kebenarannya.

Penyimpanan data dari sistem ini perlu diintegrasikan ke cloud system sehingga semua perubahan data dapat diperbarui secara real-time.

II. METHODS

A. Merancang Dokumen System Requirement Specification

Dokumen System Requirement Specification adalah deskripsi komprehensif tentang tujuan dan lingkungan yang dimaksudkan untuk perangkat lunak yang sedang dikembangkan. SRS sepenuhnya menjelaskan apa yang akan dilakukan perangkat lunak dan bagaimana kerjanya. [3]

SRS mendefinisikan bagaimana aplikasi akan berinteraksi dengan perangkat keras sistem, program lain, dan pengguna manusia dalam berbagai situasi dunia nyata. System Requirement Specification dari sistem administrasi jadwal pemeriksaan berisi tujuan, fungsionalitas, deskripsi, dan user class dari sistem.

B. Merancang Dokumen Software Analysis

Dokumen desain perangkat lunak adalah deskripsi perangkat lunak yang dibuat untuk memfasilitasi analisis, perencanaan, implementasi, dan pengambilan keputusan. Dokumen desain perangkat lunak (SDD) menjelaskan bagaimana produk perangkat lunak atau fitur akan dibangun untuk memenuhi serangkaian persyaratan teknis. Jika SRS menjelaskan "apa", dokumen desain berfokus pada "bagaimana". [4]

Dokumen software desain berisi memuat gambaran umum dari rancangan perangkat lunak secara keseluruhan. Dokumen ini memuat rancangan mulai dari desain user Interface dari perangkat lunak, user stories, desain data flow, dan use case dari sistem administrasi ini.

C. Melakukan Implementasi Software Coding

Implementasi software coding adalah tahap pengimplementasian sistem menjadi suatu produk yang dapat ditinjau Most Valuable Productnya. Tahap ini dilakukan dengan membangun perangkat lunak yang berjalan di web. Sistem ini dibangun menggunakan framework antarmuka pengguna bahasa pemrograman javascript yaitu ReactJS, service database MongoDB, dan framework javascript untuk membangun server, yaitu ExpressJS.

1) *Merancang Basis Data:* Basis Data adalah kumpulan terorganisir dari informasi terstruktur, atau data, biasanya disimpan secara elektronik dalam sistem komputer. Sebuah database biasanya dikendalikan oleh sistem manajemen database (DBMS). Bersama-sama, data dan DBMS, bersama dengan aplikasi yang terkait dengannya, disebut sebagai sistem basis data, sering disingkat menjadi basis data saja. [6]

Basis Data yang digunakan pada sistem ini merupakan basis data non-relasional. Terdapat dua model utama yang dikembangkan pada basis data ini, yaitu user dan dokter. User diidentifikasi melalui data id user dan dokter diidentifikasi dengan menggunakan data dokter. Database non-relasional menggunakan service MongoDB Atlas, service ini dipilih karena sudah stabil, mudah, dan sederhana dalam penggunaannya. Sebelum penggunaan, perlu dispesifikasikan skema dari model data yang akan digunakan. Database ini berjalan di cloud dan dapat di akses kapan saja.

2) *Merancang Server:* Server adalah komponen perangkat lunak tempat logika bisnis dari suatu sistem diolah. Server pada sistem perangkat lunak ini dibangun menggunakan bahasa pemrograman javascript dengan framework ExpressJS. Server dibangun dengan arsitektur RESTful. Arsitektur RESTful memungkinkan transfer, akses, dan manipulasi representasi data tekstual dengan metode stateless. Ketika digunakan dengan benar, arsitektur ini dapat memberikan interoperabilitas yang seragam antara berbagai aplikasi di internet. stateless sangat memungkinkan aplikasi untuk berkomunikasi secara agnostik. Layanan RESTful API diakses melalui Uniform Resource Locator (URL). [6]

Routing server pada sistem ini terdiri atas tiga bagian, yaitu route untuk dokter, route untuk user, dan route untuk admin. Terdapat beberapa route pada route dokter yaitu routing

untuk mendapatkan informasi dokter, routing untuk mengubah informasi dokter, route untuk melihat daftar janji temu dokter, dan routing untuk mengubah status janji temu. Routing admin memiliki beberapa endpoint seperti untuk mengubah status akun user atau dokter, mendapatkan list user yang ada, dan mendapatkan list dokter. Kemudian terdapat routing user yang memiliki beberapa endpoint dengan fungsi yang berbeda-beda, seperti route untuk manajemen user seperti login dan register, kemudian route untuk mendapatkan detail informasi user, route untuk melakukan pendaftaran dokter, route untuk membuat janji temu dengan dokter, lalu route untuk cek ketersediaan jadwal dokter, dan route untuk menampilkan list dokter yang aktif dan jadwal konsultasi yang aktif.

3) *Merancang Antarmuka Pengguna:* Antarmuka pengguna adalah sebuah bentuk tampilan yang berhubungan langsung dengan pengguna, dan fungsi user interface yaitu untuk menghubungkan antara pengguna dengan sistem sehingga sistem dapat dioperasikan. [7] Antarmuka Pengguna pada sistem ini mengutamakan fungsionalitas dan kesederhanaan sehingga komponen dan pemilihan warnanya dibuat sederhana.

Antarmuka pengguna dibuat dengan menggunakan bahasa pemrograman javascript dengan framework ReactJS. Terdapat beberapa halaman yang dapat dikelompokkan menjadi tiga, yaitu halaman user, halaman dokter, dan halaman admin. Pada halaman admin terdapat halaman untuk menampilkan list dokter dan list pengguna dimana admin dapat menerima atau menolak pendaftaran akun sebagai dokter dan admin juga dapat melakukan blokir terhadap user/dokter. Pada bagian dokter terdapat halaman untuk menerima atau menolak pengajuan janji konsultasi dan mengubah profil.

User dapat melakukan pengajuan update akun menjadi akun dokter pada halaman apply doctor. Dokter yang tersedia dapat dipilih oleh user di halaman beranda, kemudian user dapat melakukan cek ketersediaan jadwal kemudian membuat janji temu konsultasi dengan dokter yang dipilih.

III. RESULTS

A. User Interface Sistem

Tampilan dari sistem administrasi jadwal pemeriksaan rumah sakit adalah sebagai berikut:

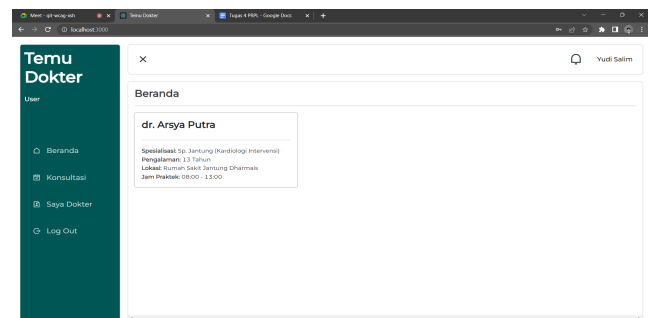


Fig. 1. Tampilan Beranda Pasien

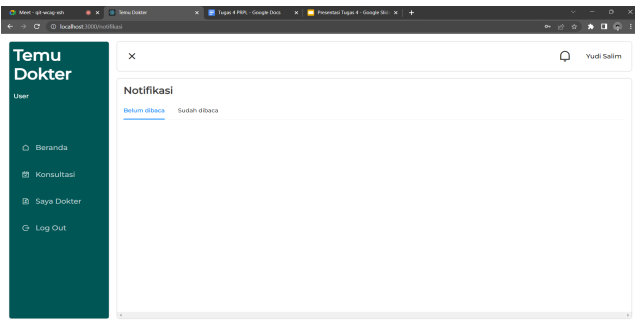


Fig. 2. Tampilan Notifikasi Pasien

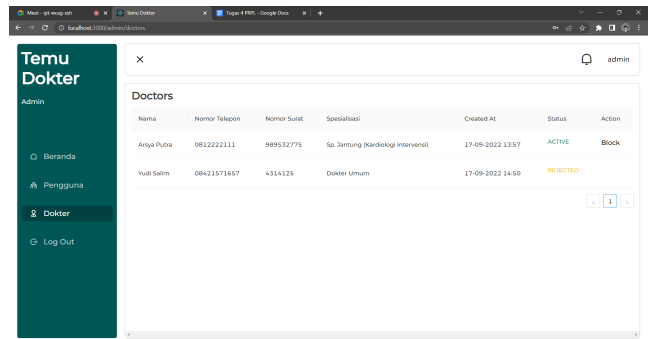


Fig. 6. Tampilan Halaman List Dokter

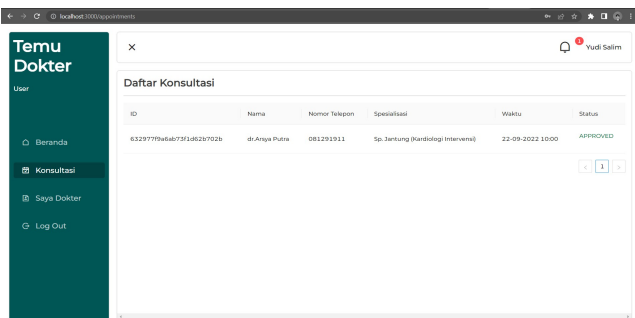


Fig. 3. Tampilan Halaman Daftar Konsultasi Pasien

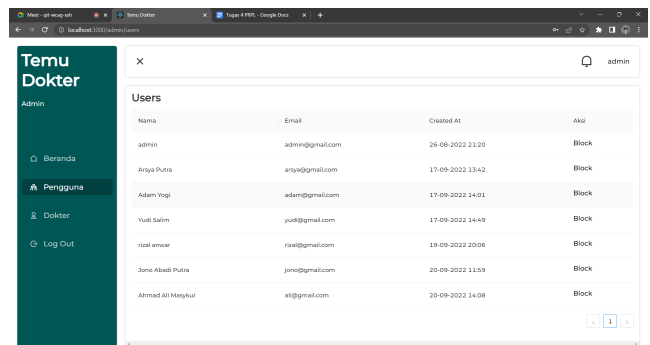


Fig. 7. Tampilan Halaman List Pengguna

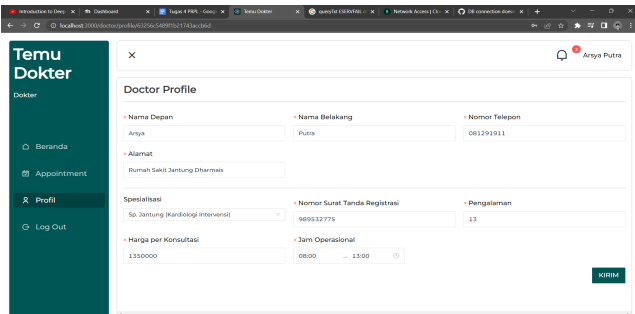


Fig. 4. Tampilan Halaman Profil Dokter

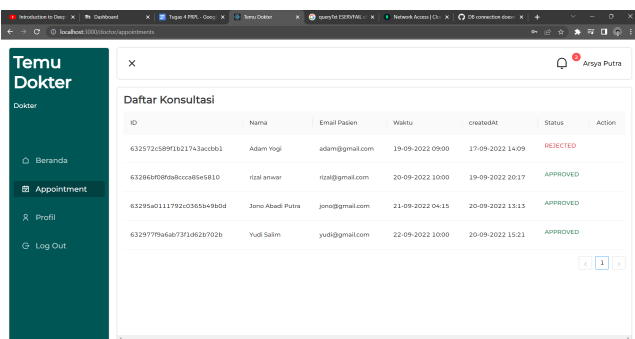


Fig. 5. Tampilan Halaman Daftar Konsultasi Dokter

B. Evaluasi Usability

Evaluasi usability sistem ini dilakukan dalam 2 tahapan, yaitu membuat parameter fungsionalitas sistem dan pengujian fitur-fitur sistem. parameter fungsionalitas dari sistem adalah sebagai berikut

- 1) User dapat login ke website dengan role user
- 2) User dapat membuat akun baru pada sistem
- 3) User dapat melihat daftar dokter yang tersedia pada halaman beranda
- 4) User dapat melihat profil ketika membuka halaman profil
- 5) User dapat mengisi form untuk mendaftarkan diri sebagai dokter
- 6) User dapat mengecek ketersediaan dokter
- 7) User dapat memilih jadwal bertemu dokter
- 8) Dokter dapat approve/reject perjanjian konsultasi user
- 9) Dokter dapat melihat daftar janji konsultasi
- 10) Admin dapat melakukan approve/reject pendaftaran dokter
- 11) Admin dapat melihat daftar dokter dan daftar user

Kita dapat menguji fitur-fitur dalam sistem untuk melihat apakah hasil pengujian setiap parameter sudah berhasil atau belum.

Parameter	Hasil	Keterangan
1	Berhasil	Kadang terjadi error tidak dapat terhubung ke database akibat network error.
2	Berhasil	Form sudah bekerja dengan seharusnya dan kemungkinan error muncul ketika terjadi network error pada client.
3	Berhasil	User sudah dapat melihat daftar dokter yang tersedia.
4	Gagal	Error ketika user mengklik logo user di pojok kanan.
5	Berhasil	Form untuk mendaftarkan diri sebagai dokter sudah berjalan sebagaimana mestinya.
6	Berhasil	User berhasil mendapatkan notifikasi feedback ketersediaan dokter.
7	Berhasil	Terdapat bug dimana user dokter dapat membuat jadwal pemeriksaan dengan dirinya sendiri.
8	Berhasil	Berjalan normal.
9	Berhasil	Berjalan normal.
10	Berhasil	Berjalan normal.
11	Berhasil	Berjalan normal.

IV. DISCUSSION

Perangkat ini dikembangkan menggunakan framework javascript pada bagian client dan server nya, framework yang digunakan untuk pengembangan user interface pada client adalah ReactJS yang memungkinkan pengembangan lebih cepat dibandingkan menggunakan javascript biasa, kami memilih ReactJS dengan mempertimbangkan batas waktu implementasi software, kemudian ReactJS juga memiliki komunitas yang sudah cukup besar sehingga referensinya mudah dicari dan dipahami. ExpressJS kami gunakan sebagai framework untuk server, mirip seperti ReactS framework ini memungkinkan proses pengembangan yang cepat dan modular, dan juga memiliki komunitas yang cukup besar sehingga resource nya mudah dicari. Kemudian basis data yang kami pilih adalah basis data non relasional yaitu MongoDB, kami memilih basis data ini karena memberikan kemudahan dalam menulis query dan menggunakan keamanan yang cukup bagus. Basis data non relasional dipilih berdasarkan pengalaman pengembang menggunakan basis data relasional yang membutuhkan cukup banyak konfigurasi untuk melakukan hal yang cukup simpel.

Secara umum, software bekerja dengan semestinya dan sudah dapat menjalankan tugas pokoknya yaitu menghubungkan pihak dokter dan pasien dalam membuat jadwal pemeriksaan. Namun, masih ada beberapa bug yang muncul tetapi sebagian besar hanyalah bug yang minor. Tampilan software ini juga masih sederhana dan masih terdapat banyak ruang untuk pengembangan. Software ini bisa dijadikan sebagai referensi atau basis untuk pengembangan sistem informasi rumah sakit, khususnya untuk fitur antrian pemeriksaan. Software ini masih cukup sederhana sehingga sangat memungkinkan untuk dikembangkan untuk fitur lainnya. Jika dibandingkan dengan software serupa seperti rekmed.com, halodoc, dan openEMR.com software ini masih terdapat limitasi fungsional seperti ruang chat dengan dokter, penyimpanan rekam medis, pemberian resep, dan keamanan yang masih belum memenuhi standar. Namun, sistem ini

memberikan gambaran alur pembuatan janji konsultasi antara pasien dan dokter yang cukup berbeda.

REFERENCES

- [1] A. A. G. Muninjaya, "Manajemen Mutu Pelayanan Kesehatan," Indonesia Onesearch. [Online]. Available: <https://onesearch.id/Record/IOS3420.slims-57048>. [Accessed: 30-Sep-2022].
- [2] B. S. SABARGUNA, "Sistem Informasi Manajemen Rumah Sakit," Indonesia Onesearch, 2003. [Online]. Available: <https://onesearch.id/Record/IOS3107.28552holdings>. [Accessed: 30-Sep-2022].
- [3] L. Rosencrance, "Software requirements specification (SRS)," SearchSoftwareQuality, 10-Sep-2019. [Online]. Available: <https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification>. [Accessed: 30-Sep-2022].
- [4] A. Zhang, "How to write a good software design doc," freeCodeCamp.org, 08-Jun-2018. [Online]. Available: <https://www.freecodecamp.org/news/how-to-write-a-good-software-design-document-66fcf019569c/>. [Accessed: 30-Sep-2022].
- [5] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database system concepts. New York, NY: McGraw-Hill, 2020. A. Silberschatz, H. F. Korth, and S. Sudarshan, Database system concepts. New York, NY: McGraw-Hill, 2020.
- [6] B. Sletten, "Foundations of RESTful Architecture - DZone Refcardz," dzone.com. [Online]. Available: <https://dzone.com/refcardz/rest-foundations-restful>. [Accessed: 30-Sep-2022].
- [7] N. Nimas, "Pengertian Dan Fungsi user interface (Antar Muka Pengguna) pada komputer Lengkap," Pro.Co.Id, 25-Feb-2022. [Online]. Available: <https://www.pro.co.id/pengertian-dan-fungsi-user-interface-antar-muka-pengguna-pada-komputer/>. [Accessed: 30-Sep-2022].