

Otomatisasi dan Digitalisasi Administrasi Pelayanan Rumah Sakit (Studi Kasus: Pelayanan Administrasi BPJS Rumah Sakit Oetomo Bandung)

1st Galang Aditya Darmawan

Fakultas Teknik Elektro
Universitas Telkom
Bandun, Indonesia

galangad@student.telkomuniversity.ac.id

2nd Rita Purnamasari

Fakultas Teknik Elektro
Universitas Telkom
Bandun, Indonesia

ritapurnamasari@telkomuniversity.ac.id

3rd Rustam

Fakultas Teknik Elektro
Universitas Telkom
Bandun, Indonesia

rustamtelu@telkomuniversity.ac.id

Abstrak — Studi ini mengkaji tantangan implementasi program BPJS di Rumah Sakit Oetomo Bandung, yang mencakup manajemen data luas, kesalahan dalam memasukkan kode ICD-10, serta masalah diagnosis penyakit yang mempengaruhi klaim BPJS. Fokus utama penelitian adalah kesulitan verifikasi klaim INA-CBG secara manual oleh tim rekam medis sebelum diinputkan ke sistem BPJS. Penelitian ini menawarkan solusi melalui pengembangan sistem otomatisasi yang menghubungkan dokter dengan tim rekam medis, termasuk otomatisasi manajemen data, pencarian kode ICD-10, dan verifikasi klaim INA-CBG. Sistem ini memungkinkan pencarian kode ICD-10 dengan opsi yang lebih luas dan mempermudah proses pendaftaran pasien, serta pencatatan data administrasi. Hasil penelitian menunjukkan bahwa meskipun pencarian kode ICD-10 manual dan sistem memberikan hasil yang sama, sistem otomatisasi menawarkan lebih banyak opsi kode dan efisiensi dalam verifikasi klaim INA-CBG.

Kata kunci— ICD-10, Verifikasi Klaim INA-CBG, BPJS, Otomasi, Layanan Kesehatan

I. PENDAHULUAN

Implementasi program BPJS di Rumah Sakit Oetomo Bandung menghadapi berbagai tantangan, terutama dalam hal manajemen data dan proses verifikasi klaim. Kesalahan dalam memasukkan kode ICD-10 dan kendala dalam diagnosis penyakit dapat mempengaruhi validitas klaim BPJS, sementara proses manual yang dilakukan oleh tim rekam medis sering kali menyebabkan keterlambatan dan potensi kesalahan. Untuk mengatasi masalah ini, penelitian ini mengusulkan pengembangan sebuah sistem website yang dirancang untuk mengotomatiskan proses pencarian dan pemilihan kode ICD-10[1].

Website ini juga akan memfasilitasi verifikasi klaim INA-CBG dengan menyediakan kolom pencarian yang memungkinkan hasil tampilan secara simultan, sehingga mempercepat dan meningkatkan akurasi proses tersebut. Hasil penelitian menunjukkan bahwa penggunaan sistem

website ini tidak hanya menyederhanakan pencarian kode ICD-10 tetapi juga menawarkan lebih banyak opsi yang relevan, memungkinkan dokter untuk memilih kode yang paling tepat. Dengan adanya website ini, diharapkan proses pengelolaan klaim BPJS di Rumah Sakit Oetomo Bandung menjadi lebih efisien dan akurat, mengurangi beban kerja manual dan meningkatkan kualitas layanan Kesehatan [2].

II. KAJIAN TEORI

A. Support Vector Machine (SVM)

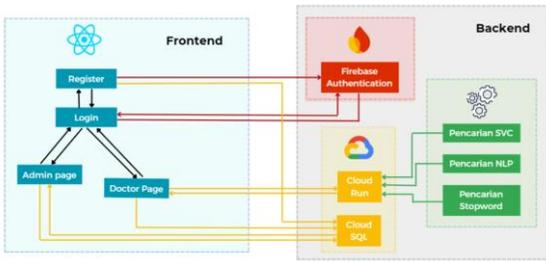
SVM menonjol sebagai salah satu algoritma pembelajaran yang sangat fleksibel dan efisien, baik untuk tugas klasifikasi maupun regresi. Kemampuannya dalam menangani baik data terstruktur maupun non-terstruktur menjadikannya solusi yang kuat dan serbaguna untuk berbagai jenis permasalahan. SVM dapat digunakan untuk masalah klasifikasi, baik dengan satu fitur (*univariate*) maupun dengan lebih dari satu fitur (*multivariate*). Dalam kasus *univariate*, SVM akan mencoba menemukan batas keputusan (*decision boundary*) yang memisahkan dua kelas berdasarkan satu fitur saja. Namun, dalam praktiknya, SVM lebih umum digunakan untuk masalah dengan lebih dari satu fitur, di mana SVM mencoba menemukan hiperplane yang memisahkan dua kelas dalam ruang fitur yang lebih tinggi[3].

B. React

React adalah *library JavaScript* yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna berbasis komponen. React memanfaatkan *Virtual DOM* untuk meningkatkan efisiensi pembaruan UI (*User Interface*), membuatnya sangat cocok untuk aplikasi yang membutuhkan interaktivitas tinggi.

III. METODE

A. Flowchart Website



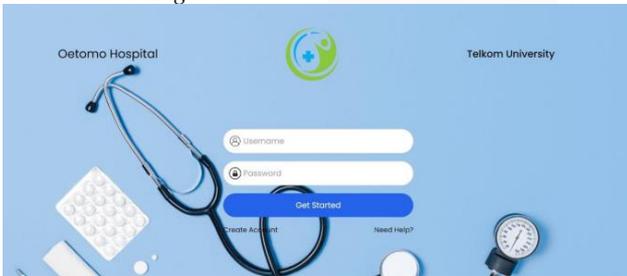
GAMBAR 1 (Arsitektur Website)

Gambar 1 menunjukkan *flowchart website* yang terdiri dari dua komponen utama, yaitu *frontend* dan *backend*. Bagian *frontend* menggunakan React sebagai kerangka kerja dan mencakup fitur seperti halaman *registrasi*, *login*, *admin*, dan *dokter*.

Bagian *backend*, *Firebase Authentication* digunakan untuk otentikasi pengguna, sementara untuk layanan *Cloud Run* serta *Cloud SQL* dari GCP menangani operasi *server-side* dan penyimpanan data. *Backend* juga mencakup beberapa modul pencarian yang spesifik, yaitu pencarian dengan menggunakan model *SVC*, pencarian dengan metode *NLP*, dan pencarian menggunakan metode *Stopwords* dari pustaka *NLTK*, yang diakses melalui *Cloud Run*. Aliran data ditunjukkan dengan garis yang menghubungkan berbagai komponen, menggambarkan interaksi antara *frontend* dan *backend* untuk menyediakan layanan yang dibutuhkan pengguna.

IV. HASIL DAN PEMBAHASAN

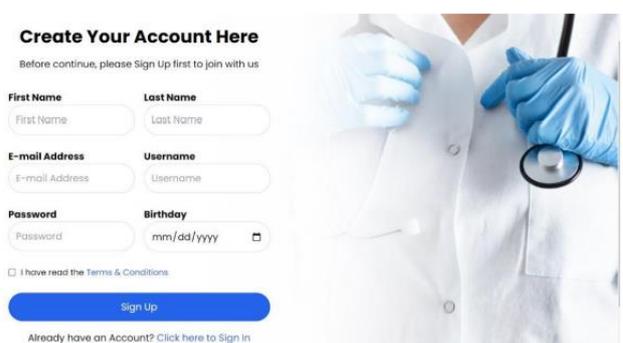
A. Halaman Login



GAMBAR 1 (Halaman Login)

Gambar 1 merupakan halaman default yang akan terbuka saat dokter membuka *website*, dokter akan diminta untuk mengisi *username* dan *password* mereka untuk bisa masuk menggunakan akun mereka. Pada halaman ini juga terdapat teks *create account* yang ditujukan kepada dokter yang belum memiliki akun, teks ini akan memindahkan dokter ke halaman *sign up*.

B. Halaman Register



GAMBAR 2 (Halaman Register)

Gambar 2 merupakan halaman *sign up* yang diperuntukkan untuk pembuatan akun dengan cara mengisi formulir dan membaca syarat dan ketentuan yang dapat dibaca pada *terms & condition*. Formulir akan meminta beberapa data seperti nama depan dan belakang, alamat email, *username* dan *password* yang ingin dibuat, dan tanggal ulang tahun.

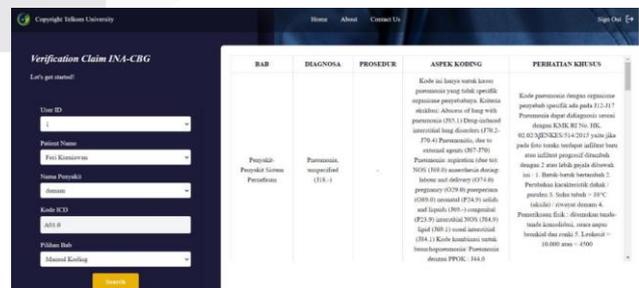
C. Halaman Dokter



GAMBAR 3 (Halaman Untuk Dokter)

Gambar 3 merupakan halaman pemilihan *ICD-10*, pertama-tama dokter diharuskan untuk memasukkan *ID dokter* pada kolom *Doctor ID*. Pada kolom *Disease Name*, dokter memasukkan hasil diagnosa penyakit pasien. Dan yang terakhir dokter harus memasukkan nama pasien pada kolom *Patient's Name*. Setelah memastikan data sudah terisi dengan benar, dokter harus menekan tombol *search* untuk menjalankan sistem.

D. Halaman Administrasi



GAMBAR 4 (Halaman Untuk Administrasi)

Gambar 4 halaman administrasi, sistem akan menampilkan data yang sudah di-input oleh dokter, staf administrasi harus mengisi seluruh kolom untuk

mendapatkan hasil panduan verifikasi klaim BPJS. Pada kolom *User ID* terdapat pilihan *ID* dokter mana yang ingin di pilih 18 oleh staf administrasi. Kolom Nama Pasien akan menampilkan nama pasien yang telah di diagnosa oleh dokter dengan *ID* yang telah dipilih. Kolom Nama Penyakit akan menampilkan nama penyakit pasien terpilih yang telah di diagnosa oleh dokter. Sedangkan kolom kode ICD -10 akan menampilkan kode ICD-10 penyakit yang telah dipilih.

E. Pengujian Website

Pengujian dilakukan terhadap beberapa fungsi utama backend dalam sistem untuk memastikan waktu respons sudah optimal. Pengujian ini, yang berfokus pada *respons time*, dilakukan menggunakan *Postman* dengan melalui beberapa tahapan. Pertama, lingkungan di *Postman* diatur dengan mengonfigurasi variabel *URL endpoint*, isi *body*, dan parameter *request*. Setelah itu, dibuat permintaan HTTP yang sesuai untuk setiap fungsi yang akan diuji, kemudian permintaan tersebut dikirim ke server *backend*. Waktu respons yang tercatat untuk setiap fungsi dianalisis, dan hasil pengujian tersebut didokumentasikan. Tabel 1 berisi dokumentasi hasil pengujian waktu respons dari fungsi *backend* yang ada pada website menggunakan *Postman*.

TABEL 1
(TABEL HASIL PENGUJIAN WEBSITE)

Fungsi	Respons Time
Pencarian Berdasarkan Prediksi Chapter	8.12 detik
Pencarian Menggunakan NLP	4.97 detik
Pencarian Menggunakan Stopwords Dari Pustaka NLTK	15.83 detik
Pilih Kode ICD-10	2.99 detik
Pengiriman Dokter ID	1.77 detik
Pengiriman Nama Pasien Berdasarkan Dokter ID	1.75 detik
Pengiriman Nama Penyakit Berdasarkan Dokter ID Dan Nama Pasien	1.76 detik
Pengiriman Kode ICD-10 Berdasarkan Dokter ID, Nama Pasien, Dan Nama Penyakit	1.75 detik
Pencarian Kata Pada Bab Manual Koding	2.03 detik
Pencarian Kata Pada Bab Medis	1.69 detik
Pencarian Kata Pada Bab Administrasi	1.41 detik

Tabel 1 hasil pengujian menunjukkan hasil dengan rata-rata kecepatan respons time seluruh fungsi di angka 4 detik, hasil ini terbilang cukup baik. namun terdapat fungsi yang memiliki respons time diatas 5 detik, yaitu fungsi pencarian menggunakan stopwords dari NLTK dengan catatan respons time selama '15.83' detik dan fungsi pencarian menggunakan machine learning yang memiliki catatan respons time selama '8.12' detik. Selain kedua fungsi tersebut, fungsi yang lain mencatatkan respons time yang sangat baik dengan rata-rata '2.23' detik.

V. KESIMPULAN

Kesimpulan dari studi ini menyoroti tantangan signifikan yang dihadapi dalam implementasi program BPJS di Rumah Sakit Oetomo Bandung, terutama terkait dengan kesalahan dalam pengelolaan data dan verifikasi klaim INA-CBG. Untuk mengatasi masalah ini, penelitian ini berhasil

mengembangkan sebuah sistem website yang mengotomatiskan proses pencarian dan pemilihan kode ICD-10 serta memfasilitasi verifikasi klaim.

Hasil pengujian menunjukkan bahwa sistem ini tidak hanya meningkatkan efisiensi proses verifikasi dengan menawarkan lebih banyak opsi kode ICD-10 yang relevan, tetapi juga secara signifikan mengurangi waktu respons fungsi backend, dengan sebagian besar fungsi mencatatkan waktu respons yang optimal di bawah 5 detik. Meskipun beberapa fungsi masih menunjukkan waktu respons yang lebih tinggi, secara keseluruhan, sistem ini diharapkan mampu meningkatkan akurasi dan efisiensi dalam pengelolaan klaim BPJS, serta mengurangi beban kerja manual di Rumah Sakit Oetomo Bandung.

REFERENSI

- [1] R. Amran, "Prosedur BPJS dan Klaim BPJS oleh Rumah Sakit," *HEME*, vol. V, no. 2, Mei 2023, Diakses: 25 Juli 2024. [Daring]. Tersedia pada: <https://jurnal.unbrah.ac.id/index.php/heme/issue/view/49> World Wide Web
- [2] A. Priyam, R. Gupta, A. Rathee, dan S. Srivastava, "Comparative Analysis of Decision Tree Classification Algorithms," 2013. [Daring]. Tersedia pada: <http://inpressco.com/category/ijcet>
- [3] F. Colas[^] dan P. Brazdil[^], "Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks."
- [4] Amity University, Institute of Electrical and Electronics Engineers. United Kingdom and Republic of Ireland Section, dan Institute of Electrical and Electronics Engineers, *Abstract proceedings of International Conference on Automation, Computational and Technology Management (ICACTM-2019)* : 24th - 25th April 2019.
- [5] A. Natekin dan A. Knoll, "Gradient boosting machines, a tutorial," *Front Neurorobot*, vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.
- [6] A. Paul, D. P. Mukherjee, P. Das, A. Gangopadhyay, A. R. Chintha, dan S. Kundu, "Improved Random Forest for Classification," *IEEE Transactions on Image Processing*, vol. 27, no. 8, hlm. 4012–4024, Agu 2018, doi: 10.1109/TIP.2018.2834830.
- [7] S. Boukhary dan E. Colmenares, "A clean approach to flutter development through the flutter clean architecture package," dalam *Proceedings - 6th Annual Conference on Computational Science and Computational Intelligence, CSCI 2019*, Institute of Electrical and Electronics Engineers Inc., Des 2019, hlm. 1115–1120. doi: 10.1109/CSCI49370.2019.00211.
- [8] <https://www.geeksforgeeks.org/>, "Pandas Read CSV in Python." Diakses: 25 Juli 2024. [Daring]. Tersedia pada: https://www.geeksforgeeks.org/python-read-csv-usingpandas-read_csv/.
- [9] <https://www.geeksforgeeks.org/>, "Understanding TF-IDF (Term Frequency-Inverse Document Frequency)." Diakses: 25 Juli 2024. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inversedocument-frequency/>.

- [10] <https://www.geeksforgeeks.org/>, "Label Encoding in Python." Diakses: 25 Juli 2024. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/ml-label-encoding-ofdatasets-in-python/>.
- [11] Susan Li, "Model Comparison and Selection," Towards Data Science. Diakses: 25 Juli 2024. [Daring]. Tersedia pada: <https://towardsdatascience.com/multi-class-textclassification-model-comparison-and-selection-5eb066197568>.
- [12] V. Apostolidis-Afentoulis, "SVM CLASSIFICATION WITH LINEAR AND RBF KERNELS".

