AUTENTIKASI MESIN KE MESIN BERBASIS RISIKO PADA KASUS FHIR MENGGUNAKAN RANDOM FOREST

Disusun oleh:

DAMAR ARBA PRAMUDITTA
22/501365/PPA/06386

PROGRAM STUDI MAGISTER ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA
2024

HALAMAN PENGESAHAN

AUTENTIKASI MESIN KE MESIN BERBASIS RISIKO PADA KASUS FHIR MENGGUNAKAN RANDOM FOREST

THESIS

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
Fakultas Teknik
Universitas Gadjah Mada

Disusun oleh:

DAMAR ARBA PRAMUDITYA
22/501365/PPA/06386

Telah disetujui dan disahkan

Pada tanggal . . . . . .

Dosen Pembimbing I	Dosen Pembimbing II
Dosen Pembimbing 1, S.T., M.Eng., PhD.	**Dosen Pembimbing 2, S.T., M.Eng., PhD.**
«NIP xxxxxx»	**«NIP xxxxxx»**

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama	:
NIM	:
Tahun terdaftar	:
Program Studi	:
Fakultas	: Teknik Universitas Gadjah Mada

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, tanggal-bulan-tahun

Materai Rp10.000

(Tanda tangan)

Nama Mahasiswa
NIM

HALAMAN PERSEMBAHAN

Tugas akhir ini kupersembahkan kepada kedua orang tuaku. Kupersembahkan pula kepada keluarga dan teman-teman semua, serta untuk bangsa, negara, dan agamaku.

[contoh]

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik. Dalam hal penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. <isi dengan nama Kadep>
2. <isi dengan nama Sekdep>
3. <isi dengan nama Dosen Pembimbing>
4. Kedua Orang Tua, kakak, dan adik yang selalu memberikan arahan selama belajar dan menyelesaikan tugas akhir ini.
5. <isi dengan nama orang lainnya>

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua, aamiin. [Contoh]

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SINGKATAN

[SAMPLE]

b	=	bias
$K(x_i, x_j)$	=	fungsi kernel
y	=	kelas keluaran
C	=	parameter untuk mengendalaikan besarnya pertukaran antara penalti variabel slack dengan ukuran margin
L_D	=	persamaan Lagrange dual
L_P	=	persamaan Lagrange primal
w	=	vektor bobot
x	=	vektor masukan
ANFIS	=	Adaptive Network Fuzzy Inference System
ANSI	=	American National Standards Institute
DAG	=	Directed Acyclic Graph
DDAG	=	Decision Directed Acyclic Graph
HIS	=	Hue Saturation Intensity
QP	=	Quadratic Programming
RBF	=	Radial Basis Function
RGB	=	Red Green Blue
SV	=	Support Vector
SVM	=	Support Vector Machines

INTISARI

Otentikasi M2M adalah komponen penting untuk mengamankan komunikasi FHIR *(Fast Healthcare Interoperability Resources)*, namun kredensial yang bocor adalah faktor paling umum yang menyebabkan pelanggaran data. Memastikan bahwa hanya perangkat resmi yang dapat mengakses dan bertukar data satu sama lain. Studi ini bertujuan untuk menilai risiko yang terkait dengan otentikasi M2M dan mengidentifikasi risiko tersebut. Selain itu studi ini akan menggunakan pendekatan berbasis risiko untuk mengidentifikasi dan menilai potensi risiko yang terkait dengan otentikasi M2M. Ini akan melibatkan identifikasi pelaku ancaman potensial, kerentanan, dan dampak dari serangan yang berhasil. Studi ini juga akan mengevaluasi metode otentikasi M2M saat ini dan keefektifannya dalam mengurangi risiko yang teridentifikasi.

Kata kunci : RBA, Autentikasi, M2M, Random Forest

ABSTRACT

M2M authentication is an important component for securing FHIR (Fast Healthcare Interoperability Resources) communication, but leaked credentials are the most common factor that leads to data breaches. It ensures that only authorized devices can access and exchange data with each other. This study aims to assess the risks associated with M2M authentication and identify those risks. In addition, this study will use a risk-based approach to identify and assess potential risks associated with M2M authentication. This will involve identifying potential threat actors, vulnerabilities, and impact of successful attacks. The study will also evaluate current M2M authentication methods and their effectiveness in reducing identified risks.

Keywords : RBA, Authentication, M2M, Random Forest

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam sistem kesehatan digital, FHIR *(Fast Healthcare Interoperability Resources)* telah menjadi standar yang umum digunakan untuk berbagi data medis antar sistem. Autentikasi mesin ke mesin (M2M) digunakan untuk mengamankan akses ke data FHIR oleh aplikasi kesehatan dan sistem lainnya. Namun, metode autentikasi M2M saat ini cenderung kurang adaptif terhadap risiko keamanan yang berbeda-beda pada setiap transaksi. Hal ini dapat menyebabkan celah keamanan dan penyalahgunaan data medis oleh pihak yang tidak berwenang.

Untuk mengatasi masalah ini, penelitian sebelumnya telah mengusulkan penggunaan autentikasi M2M berbasis risiko pada aplikasi online. Namun, kebanyakan penelitian hanya menggunakan model statistik sederhana atau aturan heuristik untuk membangun sistem autentikasi M2M berbasis risiko, seperti yang dilakukan Steinegger et al. [2016]. Hal ini dapat membatasi kemampuan sistem untuk mengenali ancaman keamanan yang kompleks.

Oleh karena itu, dalam penelitian ini, kami mengusulkan penggunaan metode machine learning, khususnya Random Forest, untuk membangun sistem autentikasi M2M berbasis risiko pada sistem FHIR. Dalam penelitian ini, kami akan membandingkan kinerja sistem autentikasi M2M berbasis risiko menggunakan Random Forest dengan kondisi sekarang. Kami juga akan mengevaluasi efektivitas dan efisiensi dari sistem autentikasi M2M berbasis risiko yang diusulkan. Dengan demikian, penelitian ini diharapkan dapat meningkatkan keamanan dan keandalan sistem autentikasi M2M pada sistem kesehatan digital berbasis FHIR.

1.2 Rumusan Masalah

Aturan heuristik untuk membangun sistem autentikasi dinilai membatasi kemampuan sistem untuk mengenali ancaman keamanan yang kompleks seperti *token reply.*

1.3 Batasan Masalah

Agar penelitian ini dapat dilakukan dengan baik, maka perlu dibuat batasan masalah. Batasan masalah pada penelitian ini adalah:

1. Penelitian ini fokus pada mekanisme autentikasi M2M *machine to machine* pada sistem FHIR.
2. Dataset yang digunakan adalah dataset sintetis login M2M yang dibuat oleh Stei-

negger et al. [2016]

3. Pemilihan fitur dan dataset akan dibatasi dimaksudkan untuk mengurangi kompleksitas model dan keterbatasan sumber daya komputasi.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Membangun sistem autentikasi M2M berbasis risiko menggunakan Random Forest.
2. Mengevaluasi kinerja sistem autentikasi M2M berbasis risiko menggunakan Random Forest.
3. Mengevaluasi efektivitas dan efisiensi sistem autentikasi M2M berbasis risiko menggunakan Random Forest.
4. Meningkatkan keamanan dan keandalan sistem autentikasi M2M pada sistem kesehatan digital berbasis FHIR.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan sebagai berikut:

1. Meningkatkan keamanan dan keandalan sistem autentikasi M2M pada sistem kesehatan digital berbasis FHIR.
2. Menambah pengetahuan dan wawasan mengenai autentikasi M2M berbasis risiko menggunakan Random Forest.
3. Meminimalisir risiko keamanan pada sistem kesehatan digital berbasis FHIR.
4. Dapat memodelkan masalah keamanan dengan menggunakan metode machine learning.

BAB II
TINJAUAN PUSTAKA

Autentikasi berbasis risiko (RBA) adalah metode untuk memverifikasi identitas pengguna dengan menyesuaikan tingkat autentikasi secara dinamis berdasarkan tingkat risiko sesi saat ini. Pendekatan ini bertujuan untuk menyeimbangkan keamanan dan kenyamanan dengan menyediakan langkah-langkah autentikasi yang lebih kuat ketika tingkat risiko tinggi, dan langkah-langkah yang lebih longgar ketika tingkat risiko rendah.

Sebuah tinjauan literatur mengenai Autentikasi Berbasis Risiko menemukan bahwa banyak penelitian telah dilakukan pada topik ini dan berbagai teknik telah diusulkan. Salah satu teknik yang paling umum adalah menggunakan algoritma penilaian risiko untuk secara dinamis menyesuaikan tingkat otentikasi berdasarkan tingkat risiko.

Studi yang dilakukan oleh Thomas et al. [2017] membahas resiko dari password yang dicuri dan bagaimana kebocoran kredensial dapat terjadi. Tidak hanya itu namun studi tersebut juga menampilkan situs situs yang banyak mengalami kebocoran data. Resiko yang paling besar dapat terjadi adalah data-data kita disalahgunakan hingga mengalami kerugian material. Sedangkan phising menjadi faktor utama penyebab terjadinya kebocoran kredensial dan disusul oleh keyloggers.

Stephan Wiefling et al. [2022] mengemukakan Risk-Based Authentication (RBA) dapat memperkirakan apakah login itu sah atau merupakan upaya pengambilalihan akun. Ini dilakukan dengan memantau dan merekam sekumpulan fitur yang tersedia dalam konteks login. Fitur potensial berkisar dari jaringan (mis., alamat IP), perangkat atau klien (mis., string agen pengguna), hingga informasi biometrik perilaku (mis., waktu masuk).Selain itu kelebihan RBA juga telah disurvey oleh Cabarcos et al. [2019] yang menganalisis literatur tentang autentikasi adaptif berdasarkan prinsip-prinsip desain yang terkenal dalam disiplin sistem berbasis resiko dan tantangan nya adalah tidak ada satu ukuran yang cocok untuk semua dalam keamanan, tidak ada mekanisme baru yang akan menggantikan semua mekanisme lainnya dan diterima sebagai solusi universal. Doerfler et al. [2019] menggambarkan bahwa tantangan login bertindak sebagai penghalang penting untuk pembajakan, tetapi gesekan dalam proses menyebabkan pengguna yang sah gagal masuk, meskipun pada akhirnya dapat mengakses akun mereka lagi.

Banyak sistem yang sudah mengimplementasikan RBA karena kelebihannya, studi yang dilakukan oleh Prasad et al. [2017] menjadi awal mula bagaimana sistem perbankan mulai menerapkan autentikasi berdasarkan risiko dengan kombinasi lokasi. Sedangkan dalam sektor kesehatan sendiri autentikasi standar seperti user dan password masih banyak digunakan, karena sistem IT kesehatan masih fokus dalam mengembangkan The Fast Health Interoperability Resources (FHIR) Ayaz et al. [2021]

Selanjutnya, beberapa studi dalam literatur mengusulkan metode otentikasi berbasis risiko yang menggunakan berbagai faktor seperti lokasi, waktu, dan jenis perangkat untuk menentukan tingkat risiko suatu sesi. Sebagai contoh, sebuah penelitian oleh Agarwal et al. [2016] mengusulkan sistem RBA berbasis lokasi yang menggunakan lokasi perangkat pengguna untuk menentukan tingkat risiko suatu sesi. Studi ini menemukan bahwa sistem yang diusulkan secara efektif meningkatkan keamanan sistem dengan tetap mempertahankan kegunaan.

Penggunaan RBA masih terbatas pada major digital service, hal ini sebagian disebabkan oleh kurangnya pengetahuan dan implementasi terbuka yang memungkinkan penyedia layanan mana pun untuk meluncurkan perlindungan RBA kepada penggunanya. Untuk menutup kesenjangan ini, Stephan Wiefling et al. [2021] memberikan analisis tentang karakteristik RBA dalam penerapan praktis sekaligus memberikan dataset yang dapat digunakan secara umum. Penelitian lain Misbahuddin et al. [2017] mengusulkan sistem RBA berbasis perangkat yang menggunakan jenis perangkat dan status perangkat untuk menentukan tingkat risiko suatu sesi. Penelitian tersebut menemukan bahwa sistem yang diusulkan secara efektif meningkatkan keamanan sistem dengan tetap mempertahankan kegunaan menggunakan machine learning.

Penggunaan analisis berbasis risiko dalam konteks machine to machine dibahas dalam studi yang dilakukan oleh Taneja [2013]. Mekanisme keamanan tertentu mengasumsikan bahwa akhir perangkat sudah diamankan. Dalam jaringan IoT, perangkat IoT itu sendiri dapat dikompromikan. Seorang penyerang dapat mencuri perangkat, mendapatkan akses mengaksesnya dan menggunakannya untuk serangan yang lebih merusak.

Roy and Dasgupta [2018] sudah meneliti bahwa fuzzy dapat menjadi terobosan dalam menentukan multifaktor autentikasi. Selain itu, banyak penelitian juga telah mengusulkan penggunaan algoritma pembelajaran mesin seperti pohon keputusan, Random Forest, dan jaringan syaraf untuk meningkatkan kinerja RBA. Sebagai contoh, sebuah penelitian oleh Zhang et al. [2012] mengusulkan sistem RBA yang menggunakan algoritma Random Forest untuk menentukan tingkat risiko dari sebuah sesi. Penelitian ini menemukan bahwa sistem yang diusulkan mencapai tingkat akurasi yang tinggi dan meningkatkan keamanan sistem. Dalam studi lain Alam and Vuong [2013], Cabarcos et al. [2019] menunjukkan bahwa Random Forest adalah pilihan yang baik karena dapat secara efektif mengklasifikasikan transaksi berdasarkan tingkat resikonya menggunakan serangkaian fitur yang berasal dari data transaksi. Random Forest adalah algoritma pembelajaran mesin yang kuat yang dapat menangani kumpulan data besar dan mampu menangani kebisingan dan nilai yang hilang dengan baik. Selain itu, dapat memberikan skor kepentingan fitur, yang dapat digunakan untuk mengidentifikasi fitur yang paling penting untuk klasifikasi risiko. Secara keseluruhan, Random Forest adalah algoritma pembelajaran mesin yang efektif.

Rangkuman penelitian sebelumnya dapat dilihat pada Tabel 2.1. Dalam studi ini ditawarkan pendekatan autentikasi berbasis risiko dengan menggunakan dalam kasus machine to machine device yang dikaitkan dalam FHIR service.

Tabel 2.1. Tinjauan Pustaka

Nama	**Penelitian**	**Metode**	**Hasil**
Thomas dkk (2017)	Pencurian kredensial dan menilai risiko yang ditimbulkannya bagi jutaan pengguna	Framework otomatis yang menggabungkan data Google Search dan Gmail untuk mengidentifikasi lebih dari satu miliar korban kebocoran kredensial, kit phising, dan keylogger.	Mengidentifikasi 788.000 calon korban keylogger siap pakai; 12,4 juta calon korban kit phishing; 1,9 miliar nama pengguna dan kata sandi yang terungkap melalui pelanggaran data dan diperdagangkan di forum pasar gelap.
Stephan Wiefling dkk (2022)	Analisis RBA pada layanan online skala besar dunia nyata	Simple model, extended model, login dataset	RBA memblokir 99,5% penyerang naif. Simple model: targeted attackers dropped dari 0.9552 menjadi 0.5295.
Cabarcos dkk (2019)	Survey studi mengenai cara dinamis memilih mekanisme terbaik untuk mengautentikasi pengguna tergantung pada beberapa faktor	CARS-AD (Vector Space Model (VSM)), ASSO (SVM), Reinforced AuthN (Logistic Regresion)	Pengurangan overhead kata sandi (masing-masing 42% dan 47% lebih sedikit permintaan kata sandi).
Doerfler dkk (2019)	Manfaat fitur login keamanan untuk mencegah pengambilalihan akun	MFA	Memblokir lebih dari 94% upaya pembajakan.
Prasad dkk (2017)	Meningkatkan Layanan Mobile Banking menggunakan Otentikasi Berbasis Lokasi	GPS dan GPRS	GPS digunakan untuk menyediakan autentikasi lokasi, banyak informasi terkait satelit yang tidak mudah diimplementasikan.

Berlanjut di halaman selanjutnya

Table 2.1: Lanjutan Tinjauan Pustaka

Nama	**Penelitian**	**Metode**	**Hasil**
Agarwal dkk (2016)	Mengevaluasi strategi autentikasi ulang untuk ponsel	Implicit authentication, Context-aware authentication, App-specific authentication	Dalam hal kinerja tugas, konfigurasi yang diusulkan bekerja sebaik konfigurasi default, namun konfigurasi yang diusulkan dianggap lebih nyaman dan tidak terlalu mengganggu oleh pengguna.
Stephan Wiefling dkk (2021)	Memperkuat otentikasi berbasis kata sandi menggunakan Otentikasi berbasis risiko (RBA)	simple model (SIMPLE), extended model (EXTEND), Data e-learning website untuk mahasiswa kedokteran	RBA dapat mencapai tingkat autentikasi ulang yang rendah untuk pengguna yang sah saat memblokir lebih dari 99,45% serangan yang ditargetkan dengan model EXTEND.
Misbahuddin dkk (2017)	Desain sistem otentikasi berbasis risiko menggunakan machine learning	Profile analysis block, Risk Engine, Adaptive Authentication Block, SVM	Teknik yang diajukan menawarkan tiga pilihan untuk risk engine, sehingga dapat beroperasi dalam situasi yang berbeda.
Taneja dkk (2013)	Mendeteksi perangkat IoT (M2M) yang disusupi menggunakan perilaku mobilitas	Wireless gateway checking	Metode ini mendeteksi perangkat yang disusupi untuk skenario dimana perilaku device telah berubah.
Dasgupta dkk (2018)	Multifactor authentication menggunakan fuzzy decision support system	fuzzy, genetic algorithm	Perbandingan akurasi dengan metode lain: FIDO 89%, Microsoft Azure 92%, Adaptive MFA 95%.
Zhang dkk (2012)	Authentikasi dan otorisasi berdasarkan lokasi	Spoofing on the hardware level (GPS), Spoofing on the OS level, Spoofing on the application level (IP, MAC)	Mekanisme autentikasi dan otorisasi berbasis lokasi menjadi lebih aman dan valid.

Berlanjut di halaman selanjutnya

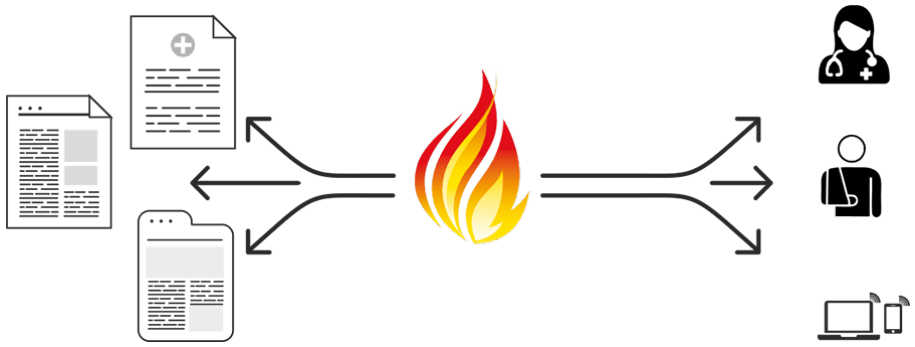
Table 2.1: Lanjutan Tinjauan Pustaka

Nama	**Penelitian**	**Metode**	**Hasil**
Alam dkk (2013)	Mendeteksi malware pada Android dengan random forest	Random forest, dataset antimalware	99,9 persen sampel benar.
Speicher dkk (2019)	Perbandingan metode pemilihan variabel random forest untuk pemodelan prediksi klasifikasi	Random forest, kondisional random forest	Standar random forest memiliki waktu komputasi dan error rate yang lebih baik dibandingkan dengan kondisional random forest.

BAB III
LANDASAN TEORI

3.1 FHIR (Fast Healthcare Interoperability Resources)

FHIR *(Fast Healthcare Interoperability Resources)* menurut Mark L. Braunstein [2022] adalah standar pertukaran data kesehatan yang dikembangkan oleh HL7 (Health Level Seven International). FHIR dapat digunakan untuk mengintegrasikan sistem kesehatan yang berbeda dan mengijinkan pertukaran data yang cepat dan aman antara sistem yang berbeda.



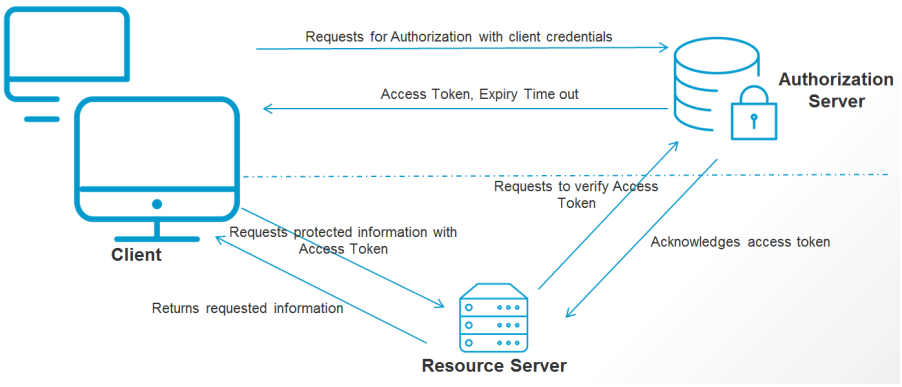
Gambar 3.1. Gambaran FHIR

FHIR menyediakan kumpulan resource yang dapat digunakan untuk pertukaran data kesehatan, seperti informasi pasien, informasi medis, dan informasi billing. Resource ini dapat ditransmisikan dalam format yang berbeda, seperti JSON atau XML. FHIR juga menyediakan API (Application Programming Interface) yang dapat digunakan untuk mengakses data dan layanan yang tersedia melalui jaringan.

FHIR dapat digunakan untuk meningkatkan interoperabilitas sistem kesehatan dan mengijinkan data kesehatan untuk ditransmisikan dengan cepat dan aman antara sistem yang berbeda. Standar ini juga memudahkan pengembangan aplikasi yang dapat mengakses data kesehatan dari berbagai sumber dan digunakan dalam berbagai konteks, seperti telemedicine, pengelolaan kesehatan, dan analisis data kesehatan.

3.2 Machine-to-Machine (M2M) Authentication

Machine-to-Machine (M2M) authentication adalah proses verifikasi yang digunakan untuk mengautentikasi perangkat atau mesin yang terhubung ke jaringan, seperti komputer, perangkat IoT, atau perangkat mobile. Proses ini memastikan bahwa hanya perangkat yang sah yang dapat terhubung ke jaringan dan mengakses data atau layanan yang tersedia seperti skema pada Gambar 3.2.



Gambar 3.2. Skema M2M Authentication

M2M authentication dapat menggunakan berbagai metode, seperti pengenalan suara, pengenalan wajah, pengenalan sidik jari, atau kombinasi dari metode tersebut. Dalam beberapa kasus, M2M authentication juga dapat menggunakan teknologi kriptografi, seperti enkripsi atau sertifikat digital, untuk memastikan keamanan komunikasi antar perangkat.

M2M authentication juga dapat digabungkan dengan metode risk-based authentication untuk meningkatkan keamanan sistem. Dengan menganalisis faktor- faktor yang dapat meningkatkan risiko, seperti lokasi geografis, waktu akses, dan jenis perangkat yang digunakan, sistem dapat mengambil tindakan yang sesuai untuk menangani ancaman potensial.

3.3 Klien Kredensial

Klien membuat permintaan ke server otorisasi dengan mengirimkan ID klien, rahasia klien, bersama dengan audiens dan klaim-klaim lainnya. Server otorisasi memvalidasi permintaan tersebut, dan, jika berhasil, mengirimkan respons dengan token akses. Klien sekarang dapat menggunakan token akses untuk meminta sumber daya yang dilindungi dari server sumber daya. Karena klien harus selalu menjaga rahasia klien, pemberian ini hanya dimaksudkan untuk digunakan pada klien terpercaya. Dengan kata lain, klien yang menyimpan rahasia klien harus selalu digunakan di tempat di mana tidak ada risiko rahasia tersebut disalahgunakan. Sebagai contoh, meskipun mungkin ide yang baik untuk menggunakan hibah kredensial klien di sistem internal yang mengirimkan laporan di seluruh web ke bagian lain dari sistem Anda, namun tidak dapat digunakan untuk alat publik yang dapat diakses oleh pengguna eksternal mana pun. Berikut ini adalah permintaan HTTP yang relevan pada Tabel 3.1 berikut:

Tabel 3.1. Permintaan HTTP

Permintaan	Deskripsi
POST	Metode HTTP
/token	Endpoint
grant_type=client_credentials	Jenis hibah
	ID klien
	Rahasia klien
	Audiens

Sedangkan berikut contoh respon HTTP yang relevan pada Tabel 3.2 berikut:

Tabel 3.2. Respon HTTP

Respon	Deskripsi
200 OK	Kode status HTTP
Content-Type: application/json	Header HTTP
Cache-Control: no-store	Header HTTP
Pragma: no-cache	Header HTTP
{	Body
"access_token": "2YotnFZFE	
"token_type": "example",	
"expires_in": 3600,	
"example_parameter": "example_value"	
}	

3.4 *Risk Based Authentication*

Risk-based adalah suatu metode yang digunakan untuk mengukur dan mengelola risiko. Dalam konteks keamanan, risk-based authentication adalah metode autentikasi yang mengukur tingkat risiko dari suatu permintaan akses, dan mengambil tindakan yang sesuai berdasarkan tingkat risiko tersebut. Metode ini bertujuan untuk mengenali dan menangani ancaman potensial tanpa mengekang fleksibilitas dan kenyamanan pengguna. Dalam konteks Machine-to-Machine (M2M) authentication, risk-based authentication digunakan untuk mengukur tingkat risiko dari suatu permintaan akses dan mengambil tindakan yang sesuai berdasarkan tingkat risiko tersebut. Prosesnya dapat dilakukan dengan cara menganalisis faktor-faktor yang dapat meningkatkan risiko, seperti lokasi geografis, waktu akses, dan jenis perangkat yang digunakan. Setelah tingkat risiko diukur, sistem

dapat mengambil tindakan yang sesuai. Jika tingkat risiko dianggap rendah, maka autentikasi dapat dilakukan secara otomatis tanpa intervensi manusia. Namun, jika tingkat risiko dianggap tinggi, maka autentikasi dapat dilakukan dengan cara yang lebih ketat, seperti mengharuskan verifikasi melalui kode SMS atau panggilan telepon, atau pembatasan akses sesuai dengan level risiko. Risk-based authentication juga dapat digabungkan dengan metode analisis risiko dinamis, yaitu mengukur risiko secara real-time dan mengambil tindakan sesuai dengan situasi yang ada. Ini dapat membantu sistem untuk mengenali dan menangani ancaman potensial secara efektif tanpa mengekang fleksibilitas dan kenyamanan pengguna seperti ilustrasi pada Gambar 3.2.

Bagian ini membahas pertimbangan etis penelitian dan [potensi] masalah serta keterbatasannya. Jika menyangkut penelitian dengan makhluk hidup, maka dibutuhkan adanya *ethical clearance*, di bagian ini hal itu akan dibahas. Demikian juga tentang keterbatasan ataupun masalah yang akan timbul.

3.5 *Classification and Regression Tree (CART)*

Metode CART merupakan suatu metode pohon keputusan (decision tree) yang bersifat recursive partitioning. Satu tree terdiri atas tiga komponen utama yaitu root node, internal node dan terminal node. Pada metode CART simpul akar (root node) dipartisi menjadi dua simpul anak (internal node), masing-masing simpul anak kemudian dipartisi menjadi dua simpul anak yang baru hingga menjadi terminal node yang bersifat homogen sebagai interpretasi dari tree Zhang, H & Singer (2010). CART membentuk tree dengan dua langkah yaitu, pembentukan maksimal dari decision tree berdasarkan proses splitting (pemilahan) dan pemangkasan (pruning) dengan mempertimbangkan tree dan cabang pohon yang terbentuk. Proses splitting variabel pada percabangan node pada tree dilihat dari variabel yang memiliki nilai goodness of split maksimal. Nilai ini dilihat berdasarkan perubahan gini impurity/gini index pada node t dan percabangan nodenya menurut Gordon dkk. (1984) dengan rumus sebagai berikut.

Node Kiri:

$$\text{imp}(\mathbf{t}_{\text{L}}) = \sum_{l=1}^{2} p_{\text{tL}}(l)(1 - p_{\text{tL}}(l)) \tag{3-1}$$

Node Kanan:

$$\text{imp}(\mathbf{t}_{\text{R}}) = \sum_{l=1}^{2} p_{\text{tR}}(l)(1 - p_{\text{tR}}(l)) \tag{3-2}$$

Node t:

$$\text{imp}(\mathbf{t}) = \sum_{k=1}^{2} p_{\text{t}}(k)(1 - p_{\text{t}}(k)) \tag{3-3}$$

Keterangan:

$$p_t(k) = \frac{n_t(k)}{n_t} \quad \text{dan} \quad p_t(l) = \frac{n_t(l)}{n_t} \tag{3-4}$$

$p_t(k), p_t(l)$: Proporsi objek kelas klasifikasi ke-k atau ke-l pada node t (3-5)

$n_t(k), n_t(l)$: Jumlah observasi kelas klasifikasi ke-k atau ke-l pada node t (3-6)

n_t : Jumlah seluruh observasi pada node t (3-7)

Gini Impurity berfungsi untuk menentukan seberapa banyak pemisah yang akan dibentuk decision tree. Sementara dalam pemilihan variabel s yang digunakan untuk memilah ditentukan oleh nilai Goodness of Split sebagai berikut.

$$\Delta\text{imp}(\mathbf{s}, \mathbf{t}) = \text{imp}(\mathbf{t}) - \text{p}_{t\text{L}}\text{imp}(\mathbf{t}_\text{L}) - \text{p}_{t\text{R}}\text{imp}(\mathbf{t}_\text{R}) \tag{3-8}$$

Keterangan:

$$p_{t\text{L}} = \frac{n_{t\text{L}}}{n_t} \quad \text{and} \quad p_{t\text{R}} = \frac{n_{t\text{R}}}{n_t} \tag{3-9}$$

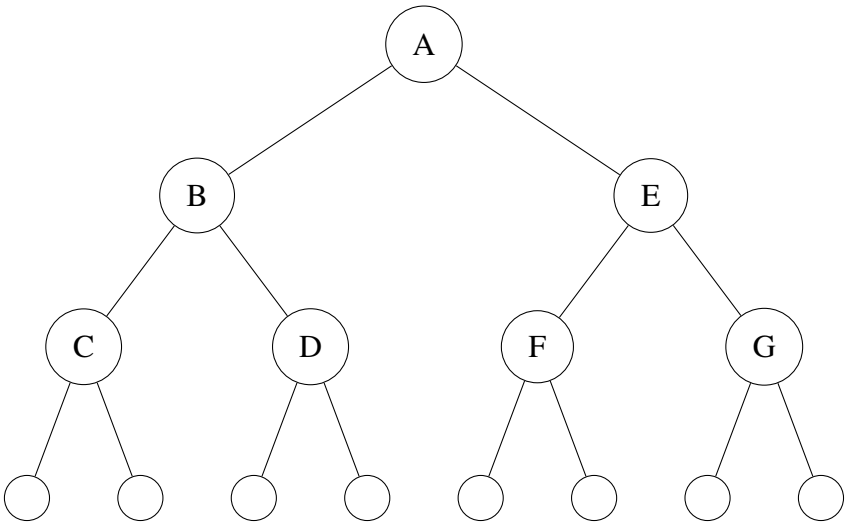
$p_t(\text{L atau R})$: Proporsi objek pada node t yang memilah pada node t_L atau node t_R (3-10)

$n_t(\text{L atau R})$: Jumlah observasi pada node t yang memilah pada node t_L atau node t_R (3-11)

n_t : Jumlah seluruh observasi pada node t (3-12)

Variabel pemilah s yang memiliki goodness of split maksimal merupakan variabel yang lebih baik digunakan untuk melakukan proses splitting. Serta apabila terminal node yang terbentuk dari internal node memiliki nilai gini index lebih besar maka sebaiknya proses splitting dihentikan pada internal node sehingga menjadi terminal node.

3.5.1 Random Forest



Membentuk tree lainnya sehingga terbentuk beberapa tree berdasarkan ntree Random Forest (RF) merupakan pengembangan metode CART. RF merupakan kumpulan banyak decision tree untuk membangun satu forest dan melihat vote klasifikasi dari tree yang menghasilkan prediktif lebih akurat Genuer dkk. (2008). Tree di RF dibentuk tidak menggunakan seluruh sampel melainkan menggunakan sampel bootstrap dan tidak melakukan pruning. Bootstrap merupakan metode berbasis resampling data dengan syarat pengembalian dalam menyelesaikan suatu permasalahan James dkk. (2021). Pada RF sampel bootstrap yang digunakan adalah 2/3 data original dengan pengembalian sehingga membentuk sampel bootstrap yang memiliki jumlah sama dengan data original sedangkan 1/3 data original lainnya disebut sampel out of bag (OOB) yang digunakan untuk pengujian prediksi tree yang sudah terbentuk dari sampel bootstrap Breiman (2001). Terdapat tiga tuning parameter yang digunakan metode RF yaitu mtry (banyak input variabel secara acak terpilih dalam satu node pemilahan) yang secara default mtry = $\sqrt{p}$ untuk kasus klasifikasi, ntree (jumlah banyaknya tree dalam forest) yang secara default ntree = 500, penelitian ini menggunakan ntree berjumlah 100, 250, 500, dan 1000, serta node size (minimum nomor observasi dalam sebuah node) yang secara default 1 untuk klasifikasi Probst dkk. (2019). Pembentukan tree pada RF dilakukan dengan cara membentuk sampel bootstrap, lalu melakukan teknik recursive partitioning pada sampel bootstrap sehingga menghasilkan sebuah tree, dimana dalam proses splitting tree atribut diambil berdasarkan banyaknya variabel yang terpilih melalui mtry. Selanjutnya, melakukan kembali pembentukan sampel bootstrap dan metode recursive partitioning untuk dalam membangun satu forest untuk melihat vote klasifikasi dari seluruh tree yang terbentuk.

3.5.2 Laju Galat Klasifikasi

OOB sampel berfungsi sebagai percobaan prediksi tree yang terbentuk dikarenakan setiap tree memiliki sampel bootstrap yang berbeda, sehingga setiap amatan dapat menjadi sampel OOB dan perlu diprediksi menggunakan beberapa tree yang dibangun tidak menggunakan sampel tersebut. Estimasi error pada hasil prediksi RF dapat diduga dengan menggunakan laju galat OOB (OOB error rate) yang dihitung dari hasil proporsi kesalahan prediksi klasifikasi setiap amatan dari hasil RF Janitza & Hornung (2018). Penggunaan mtry untuk melihat hasil dari OOB error diharapkan tidak terlalu rendah, dikarenakan apabila terlalu rendah, maka hasil OOB error akan semakin tinggi yang menghasilkan RF memiliki kinerja yang buruk. OOB error rate diharapkan memiliki nilai terkecil (minimum). Berikut perhitungan laju galat OOB dalam klasifikasi.

$$\text{Laju Galat OOB}_i = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(Y_i \neq P_i) \tag{3-13}$$

OOB error rate digunakan untuk memprediksi observasi ke- i dari Xi dimana prediksi hanya berlaku untuk suatu tree yang sampel bootstrapnya tidak mengandung (Xi, Yi)

3.5.3 *Variable Importance Measure (VIM)*

Penggunaan analisis dalam RF secara umum sangat sulit untuk melakukan interpretasi dalam memperoleh informasi. Salah satu solusi untuk mempermudah memperoleh informasi dalam RF ialah dengan mengidentifikasi Variable Importance Measure (VIM) untuk variabel prediktor. Apabila variabel importance dapat diidentifikasi, maka hasil RF akan diperoleh metode penyeleksian variabel yang berpengaruh penting terhadap pembentukan tree dalam RF. Estimasi pemilihan variabel importance dalam random forest dapat dilakukan dengan melihat berapa banyak kenaikan prediksi error (OOB) data untuk variabel terpilih sementara yang lainnya tidak berubah Liaw & Wiener (2002). Metode representatif dari perhitungan pengukuran variabel importance adalah Mean Decrease Impurity (MDI) atau disebut juga dengan Mean Decrease Gini (MDG) yang diusulkan oleh Breiman pada tahun 2001. Suatu p peubah penjelas dengan h=(1,2,. . . ,p) maka rumus mengukur tingkat kepentingan peubah penjelas Xh dengan cara berikut (Xiao Li. dkk, 20 19).

$$\text{MDG}(\mathbf{x}_h) = \frac{1}{k}\sum_{t=1}^{k} \text{MDG}(\mathbf{X}_h, \mathbf{x}_t) \tag{3-14}$$

Keterangan:

$$\text{MDG}(\mathbf{X}_h, \mathbf{x}_t) = \sum_{t\in(T), v(t)=h} \frac{N_n(t)}{n} \Delta\mathbf{x}(t) \tag{3-15}$$

Selain itu, perhitungan VIM dapat juga dengan menggunakan perhitungan Mean Decrease Accuracy (MDA) atau Permutation Importance yang menggunakan OOB untuk membagi data sampelnya, dimana OOB memperkirakan nilai prediksi dengan menghitung nilai akurasi OOB sebelum dan sesudah permutasi Xh dan menghitung perbedaannya, dengan rumus sebagai berikut Strobl dkk. (2008)

$$\text{MDA}(\mathbf{x}_h) = \frac{1}{k}\sum_{t=1}^{k}\sum_{i\in\text{OOB}(t)}\frac{I(y_i = \hat{y}_i(t)) - I(y_i = \hat{y}_i, h(t))}{|\text{OOB}(t)|} \tag{3-16}$$

dimana $OOB(t)$ adalah sampel OOB untuk satu tree ke- t , dengan t elemen dari 1,2,3, $\dots$, k, tingkat kepentingan variabel Xh dalam tree ke $-t$ adalah nilai rata- rata dari perbedaan antara kelas prediksi sebelum permutasi Xh yaitu $\hat{y}_i(t) = f(t)(x_i)$ dan kelas prediksi setelah permutasi Xh, yaitu $\hat{y}_{i,h}(t) = f(t)(x_{i,h})$ dalam i observasi tertentu.

BAB IV
ANALISIS DAN RANCANGAN SISTEM

4.1 Analisis Sistem

Analisis sistem terdiri dari gambaran umum sistem yang dapat dilihat pada baguan 4.1.1 dan analisis kebutuhan sistem yang dapat dilihat pada bagian 4.1.2.

4.1.1 Gambaran Umum Sistem

Gambar 4.1. Gambaran Umum Sistem

Gambar 4.1 menjelaskan secara umum system bekerja dengan menggunakan metadata login sebagai input untuk mengidentifikasi risiko dari suatu transaksi. Metadata login ini kemudian diolah dan dianalisis menggunakan metode Random Forest untuk menghasilkan prediksi risiko autentikasi dengan output nya adalah klasifikasi.

4.1.2 Analisis Kebutuhan Sistem

Dalam membangun sistem ini, diperlukan analisa kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional adalah kebutuhan yang berkaitan dengan fungsi-fungsi yang harus ada dalam sistem. Kebutuhan non-fungsional adalah kebutuhan yang berkaitan dengan kualitas sistem yang dibangun. Kebutuhan fungsional dan non-fungsional dapat dilihat pada bagian 4.1 dan Tabel 4.2.

4.1.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem ini adalah sebagai berikut:

1. Sistem dapat melakukan analisis risiko autentikasi dengan menggunakan metode Random Forest.
2. Sistem dapat men-genrate token autentikasi dari input user id.
3. Sistem risiko autentikasi dapat terintegrasi dengan sistem FHIR.

4.1.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sistem ini adalah sebagai berikut:

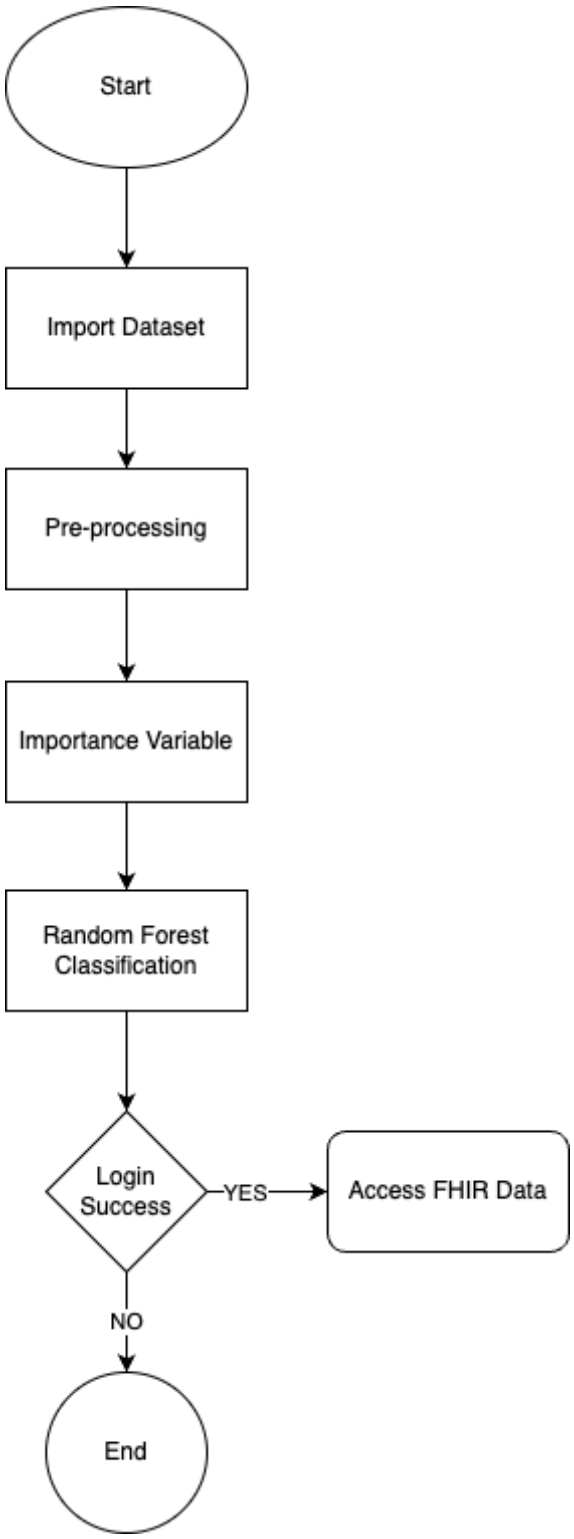
1. Keamanan data : Sistem dapat melindungi data dari akses yang tidak sah.

4.2 Rancangan Sistem

Berikut adalah rancangan sistem yang akan dibangun. Rancangan sistem terdiri dari rancangan arsitektur sistem, rancangan pembersihan data, rancangan variabel kepentingan, dan rancangan integrasi dengan sistem FHIR.

4.2.1 Rancangan Arsitektur Sistem

Rancangan arsitektur sistem dapat dilihat pada Gambar 4.3. Sistem ini terdiri dari 3 komponen utama yaitu komponen *data preprocessing*, komponen *data mining*, dan komponen *data integration*. Komponen *data preprocessing* berfungsi untuk membersihkan data dari *noise* dan *outlier*. Komponen *data mining* berfungsi untuk melakukan analisis risiko autentikasi dengan menggunakan metode Random Forest. Komponen *data integration* berfungsi untuk mengintegrasikan sistem dengan sistem FHIR.

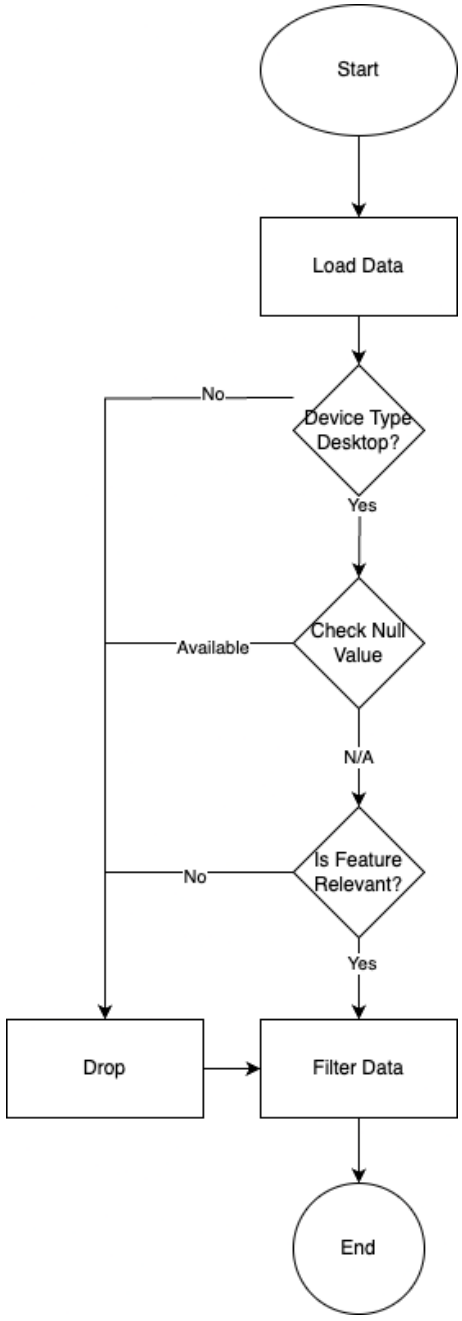


Gambar 4.2. Rancangan Arsitektur Sistem

4.2.2 Rancangan Pembersihan Data

Rancangan pembersihan data dapat dilihat pada Gambar 4.3 Pada tahap ini, data akan dibersihkan dari *noise* dan *outlier*. *Noise* adalah data yang tidak memiliki nilai yang berarti. *Outlier* adalah data yang memiliki nilai yang ekstrim. Pada tahap ini, data akan

dibersihkan dari *noise* dan *outlier* dengan menggunakan beberapa metode yaitu :



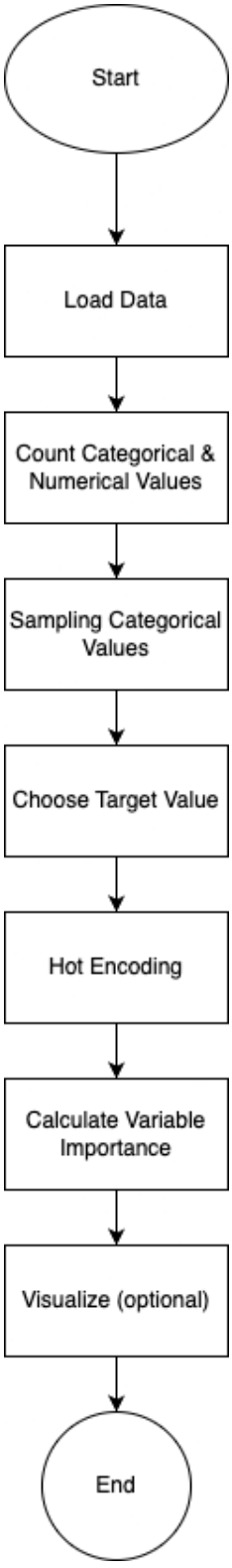
Gambar 4.3. Rancangan Pembersihan Data

1. *Outlier* : Menghapus data yang memiliki nilai yang ekstrim.
2. *Noise* : Menghapus data yang tidak memiliki nilai yang berarti.
3. *Missing Value* : Menghapus data yang memiliki nilai kosong.

4.2.3 Rancangan Variabel Kepentingan

Rancangan variabel kepentingan akan dilakukan dengan menggunakan metode Random Forest. Metode Random Forest akan menghasilkan variabel kepentingan yang

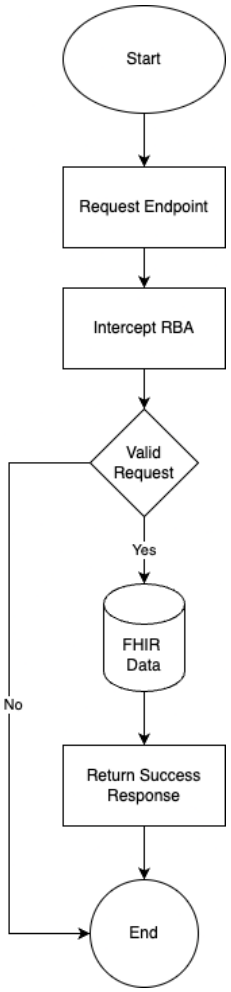
dapat dilihat pada Gambar 4.4. Variabel kepentingan ini akan digunakan untuk melakukan analisis risiko autentikasi. Berikut adalah rancangan variabel kepentingan yang akan digunakan untuk melakukan analisis risiko autentikasi.



Gambar 4.4. Rancangan Variabel Kepentingan

4.2.4 Rancangan Integrasi Dengan Sistem FHIR

Rancangan integrasi dengan sistem FHIR dapat dilihat pada Gambar 4.6. Sistem ini akan terintegrasi dengan sistem FHIR untuk mendapatkan data login dari pasien. Data login ini kemudian akan digunakan sebagai input untuk melakukan analisis risiko autentikasi.



Gambar 4.5. Rancangan Integrasi Dengan Sistem FHIR

Untuk melakukan integrasi dengan sistem FHIR, sistem ini akan menggunakan FHIR API. FHIR API adalah sebuah API yang digunakan untuk mengakses data dari sistem FHIR. FHIR API akan mengakses data dari sistem FHIR dengan menggunakan *request* dan *response*.

4.3 Rancangan Pengujian

Pengujian sistem ini akan dilakukan dengan menggunakan beberapa metode yaitu:

1. Pengujian Fungsional : Pengujian fungsional dilakukan untuk menguji apakah sistem dapat berjalan dengan baik sesuai dengan kebutuhan fungsional yang telah

ditentukan.

2. Pengujian Non-Fungsional : Pengujian non-fungsional dilakukan untuk menguji apakah sistem dapat berjalan dengan baik sesuai dengan kebutuhan non-fungsional yang telah ditentukan.
3. Pengujian Eksperimental : Pengujian eksperimental dilakukan untuk menguji apakah sistem dapat berjalan dengan baik sesuai dengan kebutuhan eksperimental yang telah ditentukan.
4. Menentukan Evaluasi : Akurasi, Presisi, *Recall*, *F1 Score*, dan *Confusion Matrix* akan digunakan untuk menentukan evaluasi dari sistem.

BAB V
IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai implementasi dari sistem yang telah dibangun. Implementasi sistem ini terdiri dari pengumpulan data, persiapan data, pemilihan fitur, dan pembangunan sistem.

5.1 Pengumpulan Data

Dalam penelitian ini data yang digunakan adalah data fitur login dari lebih dari 33 juta upaya login dan lebih dari 3,3 juta pengguna pada layanan online berskala besar di Norwegia. Data asli dikumpulkan antara Februari 2020 dan Februari 2021 dari Kaggle. Data ini berisi 284807 baris data dengan 31 kolom. Kolom-kolom tersebut adalah sebagai berikut:

Feature	**Data Type**	**Description**	**Range or Example**
IP Address	String	IP address belonging to the login attempt	0.0.0.0 - 255.255.255.255
Country	String	Country derived from the IP address	US
Region	String	Region derived from the IP address	New York
City	String	City derived from the IP address	Rochester
ASN	Integer	Autonomous system number derived from the IP address	0 - 600000
User Agent String	String	User agent string submitted by the client	Mozilla/5.0 (Windows NT 10.0; Win64; ...
OS Name and Version	String	Operating system name and version derived from the user agent string	Windows 10
Browser Name and Version	String	Browser name and version derived from the user agent string	Chrome 70.0.3538

Device Type	String	Device type derived from the user agent string	(‘mobile‘, ‘desktop‘, ‘tablet‘, ‘bot‘, ‘unknown‘)
User ID	Integer	Identification number related to the affected user account	Random pseudonym
Login Timestamp	Integer	Timestamp related to the login attempt	64 Bit timestamp
Round-Trip Time (RTT) [ms]	Integer	Server-side measured latency between client and server	1 - 8600000
Login Successful	Boolean	‘True‘: Login was successful, ‘False‘: Login failed	(‘true‘, ‘false‘)
Is Attack IP	Boolean	IP address was found in known attacker data set	(‘true‘, ‘false‘)
Is Account Takeover	Boolean	Login attempt was identified as account takeover by incident response team of the online service	(‘true‘, ‘false‘)

Tabel 5.1. Deskripsi tabel fitur login

5.2 Persiapan Data

Penggunaan dataset dalam penelitian ini membutuhkan beberapa tahapan persiapan data, yaitu:

5.2.1 Eksplorasi Data

Tahap ini diperlukan untuk mendapat gambaran umum mengenai data yang digunakan. Pada tahap ini dilakukan eksplorasi data untuk mengetahui jumlah baris dan kolom, tipe data, dan statistik deskriptif dari data. Hasil eksplorasi data dapat dilihat pada Tabel 5.1.

5.2.1.1 Sampling Data

Berikut sampling data menggunakan metode random sampling dengan jumlah data 5 baris.

```
import pandas as pd

features = pd.read_csv('data.csv')
features.head()

```

5.2.2 Pemilihan Target

Pada tahap ini dilakukan pemilihan target yang akan diprediksi. Sebagaimana Random Forest merupakan algoritma klasifikasi, maka penelitian ini memerlukan fitur apa yang menjadi target.

Melakukkan sampling terhadap tiga kolom yang dapat menjadi target, yaitu 'Login Successful', 'Is Attack IP', dan 'Is Account Takeover'. Berikut adalah kode untuk sampling data.

```
# calculate the percentage of True and False values in bolean char'
value_counts_1 = df['is_login_success'].value_counts(normalize=True)
is_login_success_true = value_counts_1[True] * 100
is_login_success_false = value_counts_1[False] * 100
print("is_login_success")
print(f"Percentage of True values: {is_login_success_true:.2f}%")
print(f"Percentage of False values: {is_login_success_false:.2f}%")

value_counts_2 = df['is_attack_ip'].value_counts(normalize=True)
is_attack_ip_true  = value_counts_2[True] * 100
is_attack_ip_false = value_counts_2[False] * 100
print("is_attack_ip")
print(f"Percentage of True values: {is_attack_ip_true:.2f}%")
print(f"Percentage of False values: {is_attack_ip_false:.2f}%")

value_counts_3 = df['is_account_takeover'].value_counts(normalize=True)
is_account_takeover_true  = value_counts_3[True] * 100
is_account_takeover_false = value_counts_3[False] * 100
print("is_account_takeover")
```

```
20  print(f"Percentage of True values: {is_account_takeover_true
    :.2f}%")
21  print(f"Percentage of False values: {
    is_account_takeover_false:.2f}%")
22
```

Berikut adalah hasil sampling data.

Target	**True**	**False**
Login Successful	67,35%	32,65%
Is Attack IP	3,09%	96,91%
Is Account Takeover	0,01%	99,99%

Tabel 5.2. Hasil Sampling Data

Dari hasil sampling data di atas, terlihat bahwa kolom 'Login Successful' memiliki persentase True yang lebih besar dibandingkan dengan False, sehingga kolom ini dipilih sebagai target.

5.2.3 Pengecekan *Missing Value*

Menggunakan kode berikut untuk mengecek apakah ada nilai yang hilang pada setiap kolom.

```
1  features.isnull().sum()
2
```

hasilnya adalah sebagai berikut:

Feature	Missing Values
Index	0
Login Timestamp	0
User ID	0
Round-Trip Time [ms]	29993329
IP Address	0
Country	0
Region	47409
City	8590
ASN	0
User Agent String	0
Browser Name and Version	0
OS Name and Version	0
Device Type	1526
Login Successful	0
Is Attack IP	0
Is Account Takeover	0

Tabel 5.3. Missing Values in Each Feature

Dari tabel, terlihat bahwa sebagian besar kolom tidak memiliki nilai yang hilang, namun ada juga yang memilikinya. Misalnya, kolom 'Waktu Pulang Pergi [ms]' memiliki 29993329 nilai yang hilang, kolom 'Wilayah' memiliki 47409 nilai yang hilang, kolom 'Kota' memiliki 8590 nilai yang hilang, dan kolom 'Jenis Perangkat' memiliki 1526 nilai yang hilang.

5.2.4 Penambahan Kolom Token

Kolom token dibuat untuk menyimpan token yang digunakan untuk mengakses API. Kolom ini dibuat dengan cara mengenerate token secara acak menggunakan SHA512. Berikut adalah contoh kode untuk membuat kolom token.

```
# generate SHA512 Hash from user_id as m2m token
import hashlib

def generate_sha512_hash(user_id):
    sha512_hash = hashlib.sha512()
    sha512_hash.update(str(user_id).encode('utf-8'))
```

```
        return sha512_hash.hexdigest()

features['token'] = features['user_id'].apply(
generate_sha512_hash)

```

Berikut adalah contoh token yang digenerate.

User ID	Token
-3065936140549856249	4ffe29f1960c24624ec2c36909f3b39cb8d59fa18515f4
5932501938287412564	ecee6cc95d3b047c8f796b8e772a468124b7ddb599a7a3

Tabel 5.4. Contoh Token

5.2.5 Pembersihan Data

Pada proses pembersihan data, dilakukan penamaan kolom, pembersihan data yang tidak diperlukan, seperti kolom 'Index' dan lainnya. Berikut adalah contoh kode untuk melakukan pembersihan data.

5.2.5.1 Penamaan Kolom

Penamaan kolom dilakukan untuk mempermudah pemanggilan kolom. Berikut adalah contoh kode untuk melakukan penamaan kolom.

```
# rename above columns to snake case
features = features.rename(columns={'Login Timestamp': '
login_timestamp', 'User ID': 'user_id', 'Round-Trip Time [ms
]':'round_trip','Region':'region', 'City':'city', 'ASN':'asn
', 'IP Address': 'ip_address', 'Country': 'country', 'User
Agent String': 'user_agent_string','Device Type': '
device_type', 'Browser Name and Version': 'browser', 'Is
Account Takeover':'is_account_takeover', 'OS Name and Version
':'os_detail','Login Successful':'is_login_success','Is
Attack IP':'is_attack_ip'})

```

Original Column Name	New Column Name
Login Timestamp	login_timestamp
User ID	user_id
Round-Trip Time [ms]	round_trip
Region	region
City	city
ASN	asn
IP Address	ip_address
Country	country
User Agent String	user_agent_string
Device Type	device_type
Browser Name and Version	browser
Is Account Takeover	is_account_takeover
OS Name and Version	os_detail
Login Successful	is_login_success
Is Attack IP	is_attack_ip

Tabel 5.5. Column Renaming in DataFrame

5.2.5.2 Penyaringan User Agent dan Device Type

Hal ini dilakukan untuk membatasi jumlah dataset dan device type yang bertujuan mengurangi waktu komputasi dalam pembuatan model. Berikut adalah contoh kode untuk melakukan penyaringan user agent dan device type.

```
1     # check lenght in column user_agent_string
2     features['length'] = features['user_agent_string'].apply(
3         lambda row: min(len(row), len(row)) if isinstance(row,
    str) else None
4     )
5     print(features['length'].mean())
6
```

Kode di atas digunakan untuk mengetahui panjang rata-rata string pada kolom 'User Agent String'. Hasilnya adalah 136.652141700553. Setelah itu dilakukan penyaringan data dengan cara menghapus data yang memiliki panjang string lebih dari 136. Berikut adalah contoh kode untuk melakukan penyaringan data.

```
1     # only keep rows with device type desktop
```

```
    features = features[features.device_type == 'desktop']
    # filter the DataFrame based on the length of column 'user_agent_string'
    features = features[features['user_agent_string'].str.len() < 136]

```

Setelah itu dilakukan penyaringan data dengan cara menghapus data yang memiliki device type selain 'desktop'.

5.2.6 Menghapus Kolom yang Tidak Diperlukan

Pada tahap ini dilakukan penghapusan kolom yang tidak diperlukan. Kolom yang dihapus adalah kolom 'Round-Trip Time [ms]', 'Index', 'Is Attack IP', 'Is Account Takeover', 'User ID', 'Token', 'Device Type', dan 'Length'. Berikut adalah contoh kode untuk menghapus kolom yang tidak diperlukan.

```
    # drop unsued columns
    features = features.drop(['round_trip', 'index', 'is_attack_ip', 'is_account_takeover', 'user_id', 'token', 'device_type', 'length'], axis=1, inplace=True)

```

Hasil keluaran dari tahap ini adalah sebagai berikut.

Column Name	Data Type	#Distinct	NA Values
login_timestamp	object	30000	0
ip_address	object	17387	0
country	object	75	0
region	object	273	14
city	object	1414	7
asn	int64	792	0
user_agent_string	object	637	0
browser	object	167	0
os_detail	object	61	0
is_login_success	bool	2	0

Tabel 5.6. Revised Initial Exploratory Data Analysis

Berdasarkan tabel di atas, diperoleh 30000 data, dengan 10 kolom, dan ada 14

data yang memiliki nilai kosong pada kolom 'Region' dan 7 data yang memiliki nilai kosong pada kolom 'City'.

5.3 Implementasi Pemilihan Fitur

Pada bagian ini akan dijelaskan mengenai implementasi pemilihan fitur. Pemilihan fitur dilakukan dengan cara memilih fitur yang memiliki korelasi tinggi dengan target. Berikut adalah tahapan pemilihan fitur.

5.3.0.1 Eksplorasi Tipe Data

Tahap ini diperlukan untuk mengetahui tipe data dari setiap kolom. Asumsi yang digunakan adalah kolom yang memiliki tipe data numerik memiliki korelasi yang lebih tinggi dibandingkan dengan kolom yang memiliki tipe data string. Berikut adalah contoh kode untuk mengetahui tipe data dari setiap kolom.

```
1    categorical = [var for var in df.columns if df[var].dtype=='
     O']
2    print('There are {} categorical variables\n'.format(len(
     categorical)))
3    print('The categorical variables are :\n\n', categorical)
4
5    There are 8 categorical variables
6
7    The categorical variables are :
8    ['login_timestamp', 'ip_address', 'country', 'region', 'city
     ', 'user_agent_string', 'browser', 'os_detail']
9
```

Berikut adalah hasil keluaran dari tahap ini.

Column Name	Data Type
login_timestamp	object
ip_address	object
country	object
region	object
city	object
asn	int64
user_agent_string	object
browser	object
os_detail	object
is_login_success	bool

Tabel 5.7. Data Type of Each Column

Berdasarkan tabel di atas, terlihat bahwa kolom 'ASN' memiliki tipe data numerik, sedangkan kolom lainnya memiliki tipe data string.

5.3.1 Encoding

Berdasarkan tabel di atas, terlihat bahwa kolom 'ASN' memiliki tipe data numerik, sedangkan kolom lainnya memiliki tipe data string. Oleh karena itu, perlu dilakukan encoding terhadap kolom-kolom yang memiliki tipe data string. Berikut adalah contoh kode untuk melakukan encoding.

```
1     import category_encoders as ce
2
3     # One-hot encode the categorical features
4     # encode categorical variables with ordinal encoding
5     # see def preprocess_data(df) above
6     encoder = ce.OneHotEncoder(cols= ['login_timestamp', '
    ip_address', 'country', 'region', 'city', 'user_agent_string
    ', 'browser', 'os_detail'])
7     X_train = encoder.fit_transform(X_train)
8
9     X_test = encoder.transform(X_test)
10    X_train.head()
11
```

Berikut adalah hasil keluaran dari tahap ini.

5.3.2 Gini Importance

Setelah dilakukan encoding, maka seluruh kolom memiliki tipe data numerik. Berikut adalah contoh kode untuk melakukan pemilihan fitur menggunakan Gini Importance.

```
### Gini importance
# create the classifier with n_estimators = default
clf = RandomForestClassifier(random_state=0)

# fit the model to the training set
clf.fit(X_train, y_train)

# view the feature scores
feature_scores = pd.Series(clf.feature_importances_, index=
X_train.columns).sort_values(ascending=False)

# Top 10 important features
feature_scores.head(10)

```

Pada kode di atas dilakukan pemilihan 10 fitur teratas. Dikarenakan jumlah fitur yang banyak, setelah dilakukkan encoding maka akan sulit untuk memvisualisasikan seluruh fitur. Berikut adalah hasil keluaran dari tahap ini.

Feature	**Gini Importance**
asn	0.017551
country_2	0.009943
country_4	0.004708
country_6	0.003670
ip_address_23	0.003618
os_detail_1	0.003317
browser_1	0.002975
os_detail_16	0.002832
user_agent_string_49	0.002508
browser_2	0.002213

Tabel 5.8. Gini Importance of Each Feature

Dalam tabel di atas, jika di lakukkan pengelompokkan maka akan terlihat bahwa

fitur 'asn', 'country', 'ip_address', 'os_detail', 'browser', dan 'user_agent_string' memiliki nilai Gini Importance yang tinggi. Namun, hanya 4 group teratas yang memiliki nilai Gini Importance yang tinggi, yaitu 'asn', 'country', 'ip_address', dan 'os_detail' yang akan digunakan sebagai fitur dalam pembuatan model.

5.4 Pembuatan Random Forest

Pada bagian ini akan dijelaskan mengenai implementasi pembuatan Random Forest. Pembuatan Random Forest dapat dilakukan setelah memilih fitur yang memiliki korelasi tinggi dengan target. Berikut adalah tahapan pembuatan Random Forest. Dari proses eksplorasi tipe data tabel 5.7 dan 5.2.2 pemilihan target, maka diperoleh bahwa kolom 'Login Successful' memiliki korelasi yang tinggi dengan target. Oleh karena itu, kolom ini dipilih sebagai target.

5.4.1 Pembagian Data

Pada tahap ini dilakukan pembagian data meliputi

5.4.1.1 Pembagian Data Fitur dan Target

Pada tahap ini dilakukan pembagian data fitur dan target. Berikut adalah contoh kode untuk melakukan pembagian data fitur dan target.

```
# Separate the features (X) and the target (y)
X = df_encoded.drop(columns=['is_login_success'])
y = df_encoded['is_login_success']

```

Kode di atas digunakan untuk memisahkan fitur dan target. Fitur disimpan pada variabel X, sedangkan target disimpan pada variabel y.

5.4.1.2 Pembagian Data Training dan Data Testing

Pada tahap ini dilakukan pembagian data training dan data testing. Berikut adalah contoh kode untuk melakukan pembagian data training dan data testing.

```
# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.3, random_state=42)
```

Kode di atas digunakan untuk membagi data menjadi data training dan data testing. Data training disimpan pada variabel X_train dan y_train, sedangkan data testing disimpan pada variabel X_test dan y_test. Set pelatihan digunakan untuk melatih model, dan set pengujian digunakan untuk mengevaluasi performa model pada data yang

tidak terlihat. Fungsi train_test_split dari modul sklearn.model_selection digunakan untuk melakukan ini. Parameter test_size disetel ke 0,3, artinya 30% data akan digunakan untuk set pengujian, dan sisanya 70% akan digunakan untuk set pelatihan. Parameter random_state disetel ke 42 untuk memastikan bahwa pemisahan yang dihasilkan dapat direproduksi.

5.4.2 Pembuatan Model dan Pelatihan Model

Pada tahap ini dilakukan pembuatan model. Berikut adalah contoh kode untuk melakukan pembuatan model.

```
1    # Create the classifier with n_estimators = 0
2    clf = RandomForestClassifier(random_state=0)
3
4    # Fit the model to the data
5    clf.fit(X_train, y_train)
```

Kode Python yang dipilih ini menginisialisasi dan melatih klasifikasi Random Forest. Berikut adalah penjelasannya:

1. **Menginisialisasi klasifikasi Random Forest:** Baris 2 membuat instance baru dari klasifikasi Random Forest. Parameter `random_state` diatur ke 0 untuk reproduktibilitas. Ini berarti bahwa pemisahan yang dihasilkan dapat direproduksi, yang penting untuk hasil yang konsisten di berbagai penjalanan.
2. **Melatih klasifikasi Random Forest:** Baris ke 5 melatih klasifikasi Random Forest pada data latihan. Metode `fit` menerima dua argumen: fitur (`X_train`) dan target (`y_train`). Fitur adalah input untuk model, dan target adalah apa yang ingin kita prediksi dari model.

Kelas `RandomForestClassifier` memiliki banyak parameter yang dapat disesuaikan untuk mengoptimalkan kinerja model. Dalam kasus ini, hanya parameter `random_state` yang diatur, dan semua parameter lain dibiarkan sebagai nilai default.

5.4.3 Evaluasi Model

Pada tahap ini dilakukan evaluasi model. Berikut adalah contoh kode untuk melakukan evaluasi model.

```
1    # Make predictions on the test set
2    y_pred = clf.predict(X_test)
3
4    # Evaluate the accuracy of the model
5    accuracy = accuracy_score(y_test, y_pred)
6    print('Accuracy:', accuracy)
```

```
7
8     # Calculate precision, recall, and F1 score
9     precision = precision_score(y_test, y_pred)
10    recall = recall_score(y_test, y_pred)
11    f1 = f1_score(y_test, y_pred)
12
13    print('Precision:', precision)
14    print('Recall:', recall)
15    print('F1 Score:', f1)
```

Kode di atas digunakan untuk melakukan evaluasi model. Berikut adalah penjelasannya: Tahap ini mengevaluasi kinerja model machine learning menggunakan beberapa metrik: akurasi, presisi, recall, dan skor F1. Berikut adalah penjelasannya:

1. **Evaluasi Akurasi:** Beberapa baris pertama menghitung akurasi prediksi model. Akurasi adalah proporsi prediksi yang benar dari semua prediksi. Ini adalah metrik umum untuk masalah klasifikasi. Fungsi `accuracy_score` dari `sklearn.metrics` digunakan untuk menghitung akurasi. Hasilnya dicetak ke konsol.
2. **Menghitung Presisi, Recall, dan Skor F1:** Sisa kode menghitung presisi, recall, dan skor F1 dari prediksi model. Ini adalah metrik umum lainnya untuk masalah klasifikasi.
    - Presisi adalah proporsi prediksi positif benar dari semua prediksi positif. Ini adalah ukuran berapa banyak prediksi positif yang sebenarnya benar.
    - Recall (juga dikenal sebagai sensitivitas) adalah proporsi prediksi positif benar dari semua positif aktual. Ini adalah ukuran berapa banyak instansi positif aktual yang dapat diidentifikasi model.
    - Skor F1 adalah rata-rata harmonik dari presisi dan recall. Ini memberikan skor tunggal yang menyeimbangkan kedua kekhawatiran presisi dan recall dalam satu angka.

Metrik ini dihitung menggunakan fungsi `precision_score`, `recall_score`, dan `f1_score` dari `sklearn.metrics`, masing-masing. Hasilnya kemudian dicetak ke konsol.

5.4.4 Visualisasi Model

Pada tahap ini dilakukan visualisasi model. Berikut adalah contoh kode untuk melakukan visualisasi model.

```
1     # Visualize a single decision tree
```

```
2     plt.figure(figsize=(12,12))
3     tree = plot_tree(clf.estimators_[0], feature_names=X.columns
    , filled=True, rounded=True, fontsize=10)
4
```

Kode di atas digunakan untuk melakukan visualisasi model. Berikut adalah penjelasannya: Tahap ini memvisualisasikan satu pohon keputusan dari model Random Forest. Ini memberikan gambaran tentang bagaimana model membuat prediksi. Berikut adalah penjelasannya:

1. **Menginisialisasi plot:** Baris 2 menginisialisasi plot dengan ukuran 12 x 12 inci. Ini memastikan bahwa plot cukup besar untuk ditampilkan dengan jelas.
2. **Membuat plot:** Baris 3 membuat plot menggunakan fungsi `plot_tree` dari `sklearn.tree`. Ini mengambil tiga argumen: model (`clf.estimators_[0]`), nama fitur (`X.columns`), dan beberapa parameter untuk mengontrol penampilan plot. Hasilnya adalah plot pohon keputusan.

Gambar 5.1 menunjukkan plot pohon keputusan. Setiap node dalam pohon mewakili satu aturan yang digunakan untuk membuat prediksi. Pada node akar, model memeriksa apakah nilai fitur 'asn' lebih kecil dari 0,5. Jika iya, maka model akan memprediksi bahwa pengguna tidak berhasil login. Jika tidak, maka model akan memeriksa apakah nilai fitur 'asn' lebih kecil dari 1,5. Jika iya, maka model akan memprediksi bahwa pengguna berhasil login. Jika tidak, maka model akan memeriksa apakah nilai fitur 'asn' lebih kecil dari 2,5. Jika iya,

5.5 Pembagunan Sistem

Sistem dibangun berbasis API dengan menggunakan bahasa pemrograman Python 3.9.13 . Sistem ini menggunakan beberapa library, yaitu: 1. Anaconda 3 versi 2022.10 untuk mengatur lingkungan kerja Python. 2. Flask versi 2.0.2 untuk membuat API. 3. Pandas versi 1.3.4 untuk memanipulasi data. 4. Scikit-learn versi 1.0 untuk membangun model. 5. Pickle versi 4.0 untuk menyimpan model. 6. Algoritma SHA512 untuk membuat token. 7. Pytest versi 6.2.5 untuk melakukan testing.

5.5.1 Pembangunan API

BAB VI
HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan mengenai hasil dari penelitian yang telah dilakukan. Penjelasan dibagi menjadi beberapa bagian, yaitu hasil pengujian, analisis hasil pengujian, dan pembahasan hasil pengujian.

6.1 Hasil Pengujian

Hasil pengujian berupa hasil pengujian fungsional, hasil pengujian non-fungsional, hasil pengujian eksperimental, dan hasil pengujian lainnya. Hasil pengujian fungsional berisi hasil pengujian terhadap fitur-fitur yang ada pada sistem. Hasil pengujian non-fungsional berisi hasil pengujian terhadap aspek non-fungsional yang ada pada sistem. Hasil pengujian eksperimental berisi hasil pengujian terhadap sistem yang dibandingkan dengan sistem lainnya. Hasil pengujian lainnya berisi hasil pengujian yang tidak termasuk dalam hasil pengujian fungsional, non-fungsional, dan eksperimental.

6.1.1 Kinerja Sistem

Hasil pengujian kinerja sistem berisi hasil pengujian terhadap kinerja sistem. Pengujian kinerja sistem dilakukan dengan cara mengukur waktu yang dibutuhkan oleh sistem untuk menyelesaikan suatu proses. Pengujian kinerja sistem dilakukan dengan cara membandingkan waktu yang dibutuhkan oleh sistem untuk menyelesaikan suatu proses dengan waktu yang dibutuhkan oleh sistem lainnya untuk menyelesaikan proses yang sama.

6.1.1.1 Waktu Data Training

Hasil pengujian waktu data training berisi hasil pengujian terhadap waktu yang dibutuhkan oleh sistem untuk melakukan proses data training. Pengujian ini dilakukan dengan cara membandingkan waktu yang dibutuhkan oleh sistem untuk melakukan proses data training. Dalam pengujian ini, dilakukkan pengujian untuk beberapa ukuran dataset. Ukuran dataset yang digunakan adalah 10000, 20000, 30000, 40000 dan 50000. Berikut adalah hasil pengujian waktu data training.

Ukuran Dataset	Waktu Data Training (detik)
10000	453
20000	901
30000	1937
40000	2237
50000	error

Tabel 6.1. Hasil Pengujian Waktu Data Training

6.1.1.2 Penggunaan Memori dan CPU

Hasil pengujian penggunaan memori dan CPU berisi hasil pengujian terhadap penggunaan memori dan CPU oleh sistem. Pengujian ini dilakukan dengan cara membandingkan penggunaan memori dan CPU.

Ukuran Dataset	Penggunaan CPU (%)	Memory Usage (MB)
10000	5	600
20000	9,2	701
30000	14	722
40000	18,5	729

Tabel 6.2. Hasil Pengujian Penggunaan CPU dan Memory Data Training

6.1.2 Performa Sistem

Hasil pengujian performa sistem berisi hasil pengujian terhadap performa sistem. Pengujian performa sistem dilakukan dengan cara membandingkan performa sistem dengan performa sistem lainnya.

6.1.2.1 Pengujian Random Forest

Hasil pengujian random forest berisi hasil pengujian terhadap random forest. Pada pengujian pertama ingin dilihat bagaimana performa random forest dengan menggunakan beberapa parameter yang berbeda. Pada pengujian kedua ingin dilihat bagaimana performa random forest dengan menggunakan parameter yang telah dioptimasi. Dalam percobaan ini dipilih 4 parameter yang akan dioptimasi, yaitu: max_depth, min_samples_leaf, min_samples_split, dan n_estimators. Untuk setiap parameter, akan dicoba beberapa nilai yang berbeda. Untuk setiap kombinasi parameter, akan dilakukan 5 kali percobaan. Untuk setiap percobaan, akan dilakukan 5 kali validasi silang. Dengan demikian, total percobaan yang dilakukan adalah 5 x 5 x 5 x 5 = 625 percobaan.

Parameter	Values
n_estimators	100, 200, 500
max_depth	None, 10, 20
min_samples_split	2, 5, 10
min_samples_leaf	1, 2, 4

Tabel 6.3. Parameter Grid

Pada Tabel 6.3 ditunjukkan hasil pengujian random forest dengan menggunakan beberapa parameter yang berbeda. Pada tabel 6.4 ditunjukkan sampel hasil pengujian random forest dengan menggunakan parameter yang telah dioptimasi.

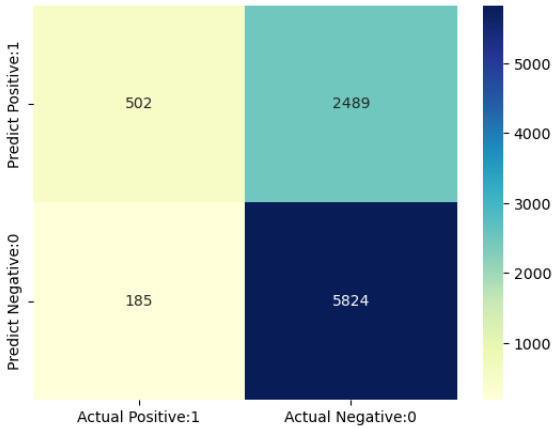
Parameters	Accuracy	Precision	Recall	F1 Score
{'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	0.668	0.737	0.788	0.762
{'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}	0.669	0.739	0.790	0.763
{'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 500}	0.708	0.701	0.968	0.813

Tabel 6.4. Model Parameters and Performance Metrics

Tabel 6.4 adalah hasil pengujian random forest dengan menggunakan parameter yang telah dioptimasi. Ditemukan bahwa parameter yang menghasilkan performa terbaik adalah {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200} dengan akurasi 0.708, presisi 0.701, recall 0.968, dan F1 score 0.813.

6.2 Analisis Hasil Pengujian

Analisis hasil pengujian berisi analisis terhadap hasil pengujian yang telah dilakukan. Analisis dilakukan dengan membandingkan hasil pengujian dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya. Apabila hasil pengujian sesuai dengan spesifikasi kebutuhan, maka sistem dapat dikatakan berhasil. Sebaliknya, apabila hasil pengujian tidak sesuai dengan spesifikasi kebutuhan, maka sistem dapat dikatakan gagal. Dalam melakukan analisis hasil pengujian, dapat digunakan beberapa metode, yaitu: Confusion Matrix: Implementasi confusion matrix membantu memahami sejauh mana model dapat mengidentifikasi True Positives (mesin yang diakui dengan benar), True



Gambar 6.1. Confusion Matrix

Negatives (mesin yang ditolak dengan benar), False Positives (mesin yang salah diakui), dan False Negatives (mesin yang salah ditolak).

	Predicted: 0	**Predicted: 1**
Actual: 0	True Negative	False Positive
Actual: 1	False Negative	True Positive

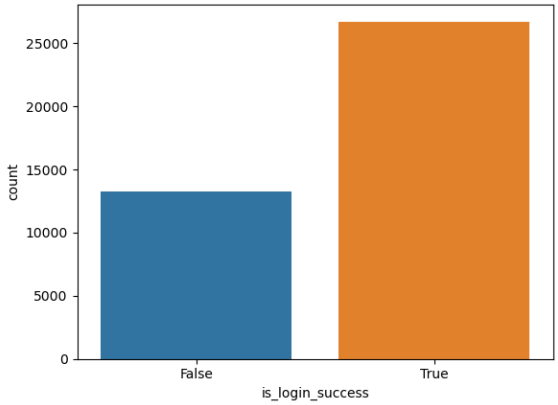
Tabel 6.5. Confusion Matrix

Hasil tertinggi yang diperoleh dari Tabel 6.4 adalah sebagai berikut:

Parameter	**Values**
n_estimators	500
max_depth	None
min_samples_split	2
min_samples_leaf	2

Tabel 6.6. Hasil Pengujian Random Forest

Dengan hasil sebagai akurasi 0.708, presisi 0.701, recall 0.968, dan F1 score 0.813. Dengan hasil ini diperoleh akurasi yang lebih rendah dari yang diharapkan. Hal ini disebabkan oleh dataset yang digunakan tidak seimbang. Dengan demikian, model yang dihasilkan cenderung memprediksi kelas mayoritas. Untuk membuktikan hal ini, dapat dilakukan pengecekan dengan melihat presentase target pada dataset. Berikut adalah presentase target pada dataset



Gambar 6.2. Presentase Target pada Dataset

Berdasarkan Gambar 6.2, dapat dilihat bahwa presentase target pada dataset adalah 0.5. Dengan demikian, dapat disimpulkan bahwa dataset yang digunakan tidak seimbang. Menurut (Sun dkk., 2009) dataset yang digunakan tidak seimbang dapat menyebabkan model yang dihasilkan cenderung memprediksi kelas mayoritas. Hal ini menyebabkan akurasi yang dihasilkan lebih rendah dari yang diharapkan.

6.3 Pembahasan Hasil Pengujian

Pembahasan hasil pengujian berisi pembahasan terhadap hasil pengujian yang telah dilakukan. Pembahasan dilakukan dengan membandingkan hasil pengujian dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya. Apabila hasil pengujian sesuai dengan spesifikasi kebutuhan, maka sistem dapat dikatakan berhasil. Sebaliknya, apabila hasil pengujian tidak sesuai dengan spesifikasi kebutuhan, maka sistem dapat dikatakan gagal.

BAB VII
KESIMPULAN DAN SARAN

Pada bagian ini dijelaskan mengenai kesimpulan dari penelitian yang telah dilakukan. Penjelasan dibagi menjadi beberapa bagian, yaitu kesimpulan, dan saran.

7.1 Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Model yang dihasilkan belum dapat mengklasifikasi risiko autentikasi dengan baik. Sistem autentikasi M2M berbasis risiko menggunakan Random Forest dapat mengklasikasi risiko autentikasi. Dengan akurasi 70.8%, presisi 70.1%, *recall* 96.8%, dan *F1-score* 71.3%. Ketimpangan akurasi dan *recall* disebabkan oleh ketidakseimbangan jumlah data pada kelas yang berbeda.
2. Pembatasan fitur kepentingan dapat berpengaruh pada akurasi sistem.

7.2 Saran

Penelitian ini masih memiliki beberapa kekurangan yang dapat diperbaiki pada penelitian selanjutnya, yaitu:

1. Penelitian ini masih menggunakan dataset hybrid. Sehingga perlu dilakukan penelitian lebih lanjut dengan menggunakan dataset asli.
2. Akurasi sistem masih dapat ditingkatkan, serta perlu dilakukan penelitian lebih lanjut untuk meningkatkan keamanan sistem.
3. Opitimasi parameter Random Forest masih dapat dilakukan lebih lanjut.
4. Dapat dilakukan perbandingan dengan memilih target parameter yang berbeda.

DAFTAR PUSTAKA

Lalit Agarwal, Hassan Khan, and Urs Hengartner. Ask Me Again But Don't Annoy Me: Evaluating Re-authentication Strategies for Smartphones. pages 221–236, 2016. ISBN 978-1-931971-31-7. URL https://www.usenix.org/conference/soups2016/technical-sessions/presentation/agarwal.

Mohammed S. Alam and Son T. Vuong. Random Forest Classification for Detecting Android Malware. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pages 663–669, August 2013. doi: 10.1109/greencom-ithings-cpscom.2013.122. MAG ID: 2031254140 S2ID: 3b274b5e931c46819bedbe430eee99a3330c857d.

Muhammad Ayaz, Muhammad Fermi Pasha, Mohammed Alzahrani, Mohammed Y. Alzahrani, Rahmat Budiarto, Deris Stiawan, and Deris Stiawan. Correction: The Fast Health Interoperability Resources (FHIR) Standard: Systematic Literature Review of Implementations, Applications, Challenges and Opportunities. *JMIR medical informatics*, 9(8), August 2021. doi: 10.2196/32869. MAG ID: 3195573207.

Patricia Arias Cabarcos, Patricia Arias-Cabarcos, Christian Krupitzer, and Christian Becker. A Survey on Adaptive Authentication. *ACM Computing Surveys*, 52(4):80, September 2019. doi: 10.1145/3336117. MAG ID: 2953371218.

Periwinkle Doerfler, Kurt Thomas, Maija Marincenko, Juri Ranieri, Yu Jiang, Angelika Moscicki, and Damon McCoy. Evaluating Login Challenges as aDefense Against Account Takeover. pages 372–382, May 2019. doi: 10.1145/3308558.3313481. MAG ID: 2914845368.

Mark L. Braunstein. FHIR. *Computers in health care*, pages 233–291, January 2022. doi: 10.1007/978-3-030-91563-6_9. MAG ID: 4226475611 S2ID: 30019acc30c7d4e00d22b4dacece471c70bd18e8.

Mohammed Misbahuddin, B. S. Bindhumadhava, B. S. Bindhumadhava, B. S. Bindhumadhava, and B. Dheeptha. Design of a risk based authentication system using machine learning techniques. pages 1–6, August 2017. doi: 10.1109/uic-atc.2017.8397628. MAG ID: 2810458423.

K. Krishna Prasad, Krishna Prasad K, and Sreeramana Aithal. A Study on Enhancing Mobile Banking Services Using Location Based Authentication. May 2017. doi: 10.47992/ijmts.2581.6012.0006. MAG ID: 2619119693 S2ID: fd10b17cfe4ead0d2551438a1dadcf8b1f06fd68.

Arunava Roy and Dipankar Dasgupta. A fuzzy decision support system for multifactor authentication. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 22(12):3959–3981, June 2018. doi: 10.1007/s00500-017-2607-6. MAG ID: 2613668921 S2ID: bf2fe16da902e41a3530ababe5eb1a619dd3b501.

Roland H. Steinegger, Daniel Deckers, Pascal Giessler, and Sebastian Abeck. Risk-based authenticator for web applications. page 16, July 2016. doi: 10.1145/3011784.3011800. MAG ID: 2589105923.

Stephan Wiefling, Markus Dürmuth, and Luigi Lo Iacono. What's in Score for Website Users: A Data-driven Long-term Study on Risk-based Authentication Characteristics. *Financial Cryptography*, 2021. doi: 10.1007/978-3-662-64331-0_19. ARXIV_ID: 2101.10681 S2ID: 2ea6241f8379f950bb2d91b1deab89123ef5680f.

Stephan Wiefling, Paul René Jørgensen, Sigurd Thunem, and Luigi Lo Iacono. Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service. *ACM transactions on privacy and security*, June 2022. doi: 10.1145/3546069. ARXIV_ID: 2206.15139 MAG ID: 4283724392 S2ID: 9166596f4e27d39728a8f80f0c2aa35b20095c10.

Mukesh Taneja. An analytics framework to detect compromised IoT devices using mobility behavior. *Information and Communication Technology Convergence*, pages 38–43, December 2013. doi: 10.1109/ictc.2013.6675302. MAG ID: 1992207658 S2ID: 738eb53e5da58684c1854f75406c45d22690df63.

Kurt Thomas, Frank Li, Ali Zand, Jacob Barrett, Juri Ranieri, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Daniel Margolis, Vern Paxson, and Elie Bursztein. Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials. pages 1421–1434, October 2017. doi: 10.1145/3133956.3134067. MAG ID: 2765227388.

Feng Zhang, Aron Kondoro, and Sead Muftic. Location-Based Authentication and Authorization Using Smart Phones. *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1285–1292, June 2012. doi: 10.1109/trustcom.2012.198. MAG ID: 2134348919 S2ID: b854f83ab609af8ac44c08e92a98a61fd84090ce.