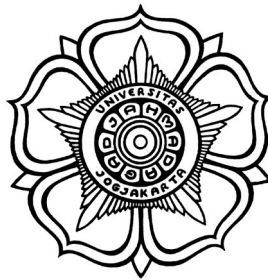


**TESIS**

**AUTENTIKASI MESIN KE MESIN BERBASIS RISIKO PADA KASUS *FAST HEALTH INTEROPERABILITY RESOURCES* MENGGUNAKAN RANDOM FOREST**

***RISK BASED MACHINE TO MACHINE AUTHENTICATION IN FAST HEALTH INTEROPERABILITY RESOURCES CASE USING RANDOM FOREST***



**DAMAR ARBA PRAMUDITYA**  
22/501365/PPA/06386

**PROGRAM STUDI MAGISTER MAGISTER ILMU KOMPUTER  
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2024**

**TESIS**

**AUTENTIKASI MESIN KE MESIN BERBASIS RISIKO PADA KASUS *FAST HEALTH INTEROPERABILITY RESOURCES* MENGGUNAKAN RANDOM FOREST**

***RISK BASED MACHINE TO MACHINE AUTHENTICATION IN FAST HEALTH INTEROPERABILITY RESOURCES CASE USING RANDOM FOREST***

Diajukan untuk memenuhi salah satu syarat memperoleh derajat  
Master of Computer Science



DAMAR ARBA PRAMUDITYA  
22/501365/PPA/06386

**PROGRAM STUDI MAGISTER MAGISTER ILMU KOMPUTER  
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2024**

**HALAMAN PENGESAHAN**

**TESIS**

**AUTENTIKASI MESIN KE MESIN BERBASIS RISIKO PADA  
KASUS *FAST HEALTH INTEROPERABILITY RESOURCES*  
MENGUNAKAN RANDOM FOREST**

**Telah dipersiapkan dan disusun oleh**

**DAMAR ARBA PRAMUDITYA  
22/501365/PPA/06386**

**Telah dipertahankan di depan Tim Penguji  
pada tanggal 17 Januari 2024**

**Susunan Tim Penguji**

**Dr. Lukman Heryawan, S.T., M.T.  
Promotor**

**Wahyono, S.Kom., Ph.D.  
Penguji**

**Ko-promotor**

**Aina Musdholifah, S.Kom.,  
M.Kom. Ph.D  
Penguji**

**Dr. Agus Sihabuddin, S.Si.,  
M.Kom.  
Penguji**

**Tesis ini telah diterima sebagai salah satu persyaratan  
Untuk memperoleh gelar Master of Science Fisika**

**Tanggal 17 Januari 2024**

**Aina Musdholifah, S.Kom., M.Kom. Ph.D**  
**Pengelola Program Studi Magister Magister Ilmu Komputer**

## **PERNYATAAN**

Dengan ini saya menyatakan bahwa dalam Tesis ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Master di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 17 Januari 2024

Damar Arba Pramuditya

Karya ini ku persembahkan kepada  
Ibu, Bapak, Kakak-kakakku, dan keponakanku tercinta  
serta semua teman-teman seperjuangan di Ilmu Komputer  
Universitas Gadjah Mada

## DAFTAR ISI

<b>Halaman Judul</b>	<b>ii</b>
<b>Halaman Pengesahan</b>	<b>iii</b>
<b>Halaman Pengesahan</b>	<b>iv</b>
<b>Halaman Pernyataan</b>	<b>v</b>
<b>Halaman Pernyataan</b>	<b>v</b>
<b>Halaman Persembahan</b>	<b>vi</b>
<b>DAFTAR ISI</b>	<b>vii</b>
<b>DAFTAR TABEL</b>	<b>x</b>
<b>DAFTAR GAMBAR</b>	<b>xi</b>
<b>INTISARI</b>	<b>xii</b>
<b>ABSTRACT</b>	<b>xiii</b>
<b>I PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
<b>II TINJAUAN PUSTAKA</b>	<b>4</b>
<b>III DASAR TEORI</b>	<b>10</b>
3.1 FHIR ( <i>Fast Healthcare Interoperability Resources</i> )	10
3.2 Autorisasi	10
3.3 Autentikasi	11
3.3.1 Standar Data Pada FHIR	11

3.3.2	Standar Autentikasi Pada FHIR	12
3.3.3	Autentikasi Mesin ke Mesin	13
3.3.4	Metode Autentikasi Mesin ke Mesin	13
3.4	<i>Risk-Based Authentication</i>	15
3.5	<i>Classification and Regression Tree (CART)</i>	16
3.5.1	<i>Random Forest</i>	18
3.5.2	Laju Galat klasifikasi	18
3.5.3	<i>Variable Importance Measure(VIM)</i>	19
3.5.4	Metriks dan <i>Scoring</i>	20
<b>IV</b>	<b>ANALISIS DAN PERANCANGAN SISTEM</b>	<b>22</b>
4.1	Deskripsi Umum Sistem	22
4.2	Analisis Kebutuhan Sistem	22
4.2.1	Analisis Kebutuhan Fungsional	22
4.2.2	Analisis Kebutuhan Perangkat Keras	22
4.2.3	Analisis Kebutuhan Perangkat Lunak	23
4.3	Rancangan Sistem	23
4.3.1	Rancangan Arsitektur Sistem	23
4.3.2	Rancangan Pembersihan Data	25
4.3.3	Rancangan <i>Encoding</i>	27
4.3.4	Rancangan <i>Variable Importance Measure</i>	28
4.3.5	Rancangan Integrasi Sistem FHIR	30
4.4	Rancangan Pengujian	33
<b>V</b>	<b>IMPLEMENTASI SISTEM</b>	<b>34</b>
5.1	Pengumpulan Data	34
5.2	<i>Preprocessing Data</i>	36
5.2.1	Eksplorasi Data	36
5.2.2	Pengecekan <i>Missing Value</i>	36
5.2.3	Pemilihan Target	37
5.2.4	Penambahan Kolom Token	38
5.3	Pembersihan Data	39
5.3.1	Penamaan Kolom	39
5.3.2	Penyaringan Data	40
5.3.3	Penghapusan Kolom	41
5.4	Pemilihan Fitur Menggunakan <i>Variable Importance Measure</i>	41



5.4.1	<i>Encoding Data</i>	42
5.4.2	<i>Gini Importance</i>	43
5.5	Implementasi <i>Random Forest</i>	44
5.5.1	Pembagian Data	44
5.5.2	Pembuatan Model	45
5.5.3	Visualisasi Model	46
5.5.4	Evaluasi Model	46
5.6	Implementasi Sistem FHIR	48
5.6.1	Implementasi API	48
5.6.2	Implementasi API	48
5.6.3	API Code	48
5.6.4	FHIR Data	51
<b>VI</b>	<b>HASIL DAN PEMBAHASAN</b>	<b>52</b>
6.1	Hasil Pengujian	52
6.1.1	Waktu Data <i>Training</i>	52
6.1.2	Penggunaan Memori dan CPU	52
6.1.3	Hasil Random Forest	53
6.2	Analisis Hasil Pengujian	53
6.2.1	<i>Random Forest Risk Based Authentication</i>	55
6.2.2	<i>Heuristic Authentication</i>	55
6.3	Pembahasan Hasil Pengujian	56
<b>VII</b>	<b>KESIMPULAN DAN SARAN</b>	<b>57</b>
7.1	Kesimpulan	57
7.2	Saran	57
	<b>DAFTAR PUSTAKA</b>	<b>58</b>

## DAFTAR TABEL

2.1	Tinjauan Pustaka	7
3.1	Permintaan HTTP	15
3.2	Respon HTTP	15
4.1	Sample Encoding Data	27
5.1	Deskripsi tabel fitur login	34
5.2	Missing Values in Each Feature	37
5.3	Hasil Sampling Data	38
5.4	Contoh Token	39
5.5	Column Renaming in DataFrame	40
5.6	Revised Initial Exploratory Data Analysis	41
5.7	Data Type of Each Column	42
5.8	Gini Importance of Each Feature	44
6.1	Hasil Pengujian Waktu Data Training Random Forest	52
6.2	Hasil Pengujian Penggunaan CPU dan Memory Data Training	53
6.3	Parameter Grid	53
6.4	Confusion Matrix	54
6.5	Hasil Pengujian Random Forest	54

## **DAFTAR GAMBAR**

3.1	Skema M2M Authentication	14
3.2	Skema M2M Authentication	16
3.3	Random Forest	19
4.1	Gambaran Umum Sistem	22
4.2	Rancangan Arsitektur Sistem	24
4.3	Rancangan Pembersihan Data	26
4.4	Rancangan Variabel Kepentingan	29
4.5	Rancangan Integrasi Dengan Sistem FHIR	31
6.1	Confusion Matrix	54

## INTISARI

### **AUTENTIKASI MESIN KE MESIN BERBASIS RISIKO PADA KASUS *FAST HEALTH INTEROPERABILITY RESOURCES* MENGUNAKAN RANDOM FOREST**

Oleh

Damar Arba Pramuditya

22/501365/PPA/06386

Studi ini menggunakan pendekatan berbasis risiko untuk mengidentifikasi dan menilai potensi risiko terkait otentikasi Machine-to-Machine (M2M). Analisis pelaku ancaman, kerentanan, dan dampak serangan dilakukan, serta evaluasi metode otentikasi M2M saat ini. Penelitian ini mengembangkan strategi peningkatan otentikasi guna mengurangi risiko serangan. Peningkatan perangkat IoT dalam teknologi perawatan kesehatan menekankan pentingnya otentikasi yang aman antar perangkat. Berbagai pendekatan seperti model enkripsi dan protokol otentikasi telah diusulkan, namun kurang dalam pendekatan berbasis risiko yang mengintegrasikan analisis ancaman dengan evaluasi efektivitas metode otentikasi.

Penelitian ini memanfaatkan algoritma Random Forest untuk mengklasifikasikan akses dan menilai efektivitas metode otentikasi saat ini. Temuan penting termasuk identifikasi risiko akses perangkat tidak sah seperti *replay attack*. Dalam konteks Fast Healthcare Interoperability Resources (FHIR), studi ini mencapai akurasi 0,708, presisi 0,701, recall 0,968, dan skor F1 0,813 dengan algoritma Random Forest.

Sebagai perbandingan, studi menggunakan Local Outlier Factor (LOF) untuk autentikasi berbasis data swipe pengguna smartphone menunjukkan bahwa meskipun model belum dioptimalkan mampu mencapai tingkat keberhasilan lebih dari 90% bahkan dengan FAR hingga 40%. Ini menunjukkan bahwa LOF dapat memberikan metrik autentikasi kompetitif dengan model kompleks seperti Random Forest.

Penelitian ini memberikan wawasan berharga bagi organisasi yang menerapkan perangkat IoT, khususnya di sektor teknologi perawatan kesehatan, untuk mengurangi risiko terkait otentikasi M2M.

Kata kunci : Otentikasi Machine-to-Machine (M2M), Fast Healthcare Interoperability Resources (FHIR), Random Forest, Local Outlier Factor (LOF), Perbaikan Autentikasi

## **ABSTRACT**

### **RISK BASED MACHINE TO MACHINE AUTHENTICATION IN *FAST HEALTH INTEROPERABILITY RESOURCES* CASE USING RANDOM FOREST**

By

Damar Arba Pramuditya

22/501365/PPA/06386

This study employs a risk-based approach to identify and assess potential risks associated with Machine-to-Machine (M2M) authentication. Threat actor analysis, vulnerability assessment, and evaluation of the impact of attacks are conducted, along with an evaluation of current M2M authentication methods. The research develops strategies to enhance authentication to mitigate attack risks. The increase in IoT devices in healthcare technology emphasizes the importance of secure inter-device authentication. Various approaches such as encryption models and authentication protocols have been proposed, but they fall short of a risk-based approach that integrates threat analysis with the evaluation of authentication effectiveness.

This study utilizes the Random Forest algorithm to classify access and assess the effectiveness of current authentication methods. Key findings include the identification of risks such as unauthorized device access like replay attacks. In the context of Fast Healthcare Interoperability Resources (FHIR), this study achieves an accuracy of 0.708, a precision of 0.701, a recall of 0.968, and an F1 score of 0.813 using the Random Forest algorithm.

For comparison, a study using the Local Outlier Factor (LOF) for swipe-based user authentication on smartphones showed that even without optimization, the model could achieve success rates of over 90% with a FAR of up to 40%. This indicates that LOF can provide competitive authentication metrics with complex models like Random Forest.

This research offers valuable insights for organizations implementing IoT devices, particularly in the healthcare technology sector, to reduce risks associated with M2M authentication.

**Keywords:** Machine-to-Machine (M2M) Authentication, Fast Healthcare Interoperability Resources (FHIR), Random Forest, Local Outlier Factor (LOF), Authentication Improvement

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

*Risk-based M2M (machine-to-machine) authentication* merupakan metode otentikasi yang mengukur tingkat risiko yang terkait dengan suatu perangkat atau sistem dan menyesuaikan tingkat otentikasi yang diperlukan sesuai dengan tingkat risiko tersebut. Dalam sistem kesehatan, FHIR (Fast Healthcare Interoperability Resources) menjadi standar yang digunakan untuk pertukaran informasi kesehatan secara elektronik. Kerangka kerja OAuth menyediakan autentikasi dan otorisasi menggunakan profil dan kredensial pengguna di penyedia identitas yang ada. Hal ini membuat memungkinkan penyerang untuk mengeksploitasi kerentanan apa pun yang timbul dari pertukaran data dengan penyedia. Kerentanan dalam OAuth Alur otorisasi OAuth memungkinkan penyerang untuk mengubah urutan alur normal protokol OAuth (Rahat, Tamjid Al et al., 2021). Sehingga, sistem otentikasi FHIR saat ini hanya didasarkan pada OAuth2 dan OpenID Connect, sehingga risiko dari perangkat yang terhubung tidak diperhitungkan dalam otentikasi.

*Fast Healthcare Interoperability Resources* atau yang selanjutnya kita singkat sebagai FHIR adalah sebuah acuan / standar yang digunakan dalam pertukaran informasi tentang kesehatan secara elektronik atau online. FHIR dikembangkan dan diawasi oleh sebuah organisasi yang bernama HL7 (Health Level Seven International) (Mark L. Braunstein, 2022). HL7 adalah sebuah non-profit organisasi yang menyediakan sebuah framework dan acuan-acuan dalam pertukaran, integrasi, pembagian dan penerimaan informasi tentang kesehatan yang dapat membantu praktik dalam kesehatan, manajemen serta evaluasi pelayanan kesehatan. Dalam konteks FHIR, ini penting karena FHIR digunakan untuk pertukaran data kesehatan elektronik antar sistem dan perangkat medis (Solapurkar, 2016). Karena FHIR digunakan untuk mengakses data kesehatan yang sensitif, penting untuk memastikan bahwa hanya perangkat dan sistem yang sah yang diizinkan untuk mengakses data. Namun, tidak semua perangkat atau sistem memiliki tingkat risiko yang sama (Dutson et al., 2019). Misalnya, perangkat medis yang digunakan untuk mengadministrasikan obat kepada pasien memiliki risiko yang lebih tinggi dibandingkan dengan sensor suhu di ruangan.

Salah satu serangan yang umum terjadi pada kasus autentikasi token ini adalah replay attack, bentuk serangan jaringan di mana transmisi data yang valid diulang atau ditunda secara jahat atau curang. Dengan mengimplementasikan metode autentikasi berbasis risiko (Stephan Wiefeling et al., 2021), sistem dapat menyesuaikan tingkat keamanan yang dibutuhkan sesuai dengan tingkat risiko dari perangkat atau sistem yang berkomunikasi, sehingga dapat meningkatkan keamanan dalam pertukaran data kesehatan melalui FHIR.

## **1.2 Rumusan Masalah**

1. FHIR masih bergantung pada external identity management sistem untuk otentikasi dan otorisasi.
2. FHIR tidak memiliki mekanisme otentikasi yang mempertimbangkan risiko dari perangkat yang terhubung.

## **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 hanya memfokuskan pada risiko yang terkait dengan otentikasi M2M pada FHIR.
2. Datasek yang digunakan dalam penelitian ini diperoleh dari literatur yang relevan.
3. Pemilihan fitur yang digunakan dalam penelitian ini adalah fitur yang relevan dengan risiko otentikasi M2M pada FHIR.

## **1.4 Tujuan Penelitian**

Tujuan penelitian ini adalah mengimplementasikan, mengevaluasi, dan membandingkan sistem autentikasi mesin ke mesin yang dapat meningkatkan keamanan sistem autentikasi, serta membandingkannya dengan heuristik autentikasi yang lain, yang nantinya dapat digunakan dalam sistem penyedia layanan kesehatan.

### **1.5 Manfaat Penelitian**

Manfaat penelitian yang didapat sebagai berikut:

1. Dapat memodelkan masalah otentikasi mesin ke mesin berbasis risiko pada FHIR.
2. Meminimalisir risiko yang terkait dengan otentikasi mesin ke mesin pada FHIR.
3. Menganalisa apakah otentikasi mesin ke mesin berbasis risiko dengan Random Forest dapat meningkatkan keamanan sistem otentikasi pada FHIR.



## **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 phishing menjadi faktor utama penyebab terjadinya kebocoran kredensial dan disusul oleh keyloggers.

(Stephan Wiefeling 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 disurvei oleh (Cabarcos et al., 2019) 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 kelebihanannya, 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 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 penggunaannya. Untuk menutup kesenjangan ini, (Stephan Wiefeling 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 & 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 & Vuong, 2013; Speiser 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 dan banyak digunakan untuk otentikasi M2M berbasis risiko.

Dalam studi ini ditawarkan pendekatan autentikasi berbasis risiko dengan menggunakan dalam kasus machine to machine device yang dikaitkan dalam FHIR service. Selain itu, penelitian ini juga melakukan komparasi antara metode autentikasi yang diimplementasikan dengan metode autentikasi berbasis heuristik lainnya. Dengan melakukan evaluasi terhadap berbagai metode autentikasi, diharapkan dapat diperoleh wawasan yang lebih mendalam mengenai keefektifan dan efisiensi masing-masing metode, sehingga dapat menentukan solusi terbaik untuk meningkatkan keamanan dalam sistem penyedia layanan kesehatan.

Tabel 2.1: Tinjauan Pustaka

<b>Nama</b>	<b>Penelitian</b>	<b>Metode</b>	<b>Hasil</b>
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 phishing, 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.

Berlanjut di halaman selanjutnya

**Table 2.1:** Lanjutan Tinjauan Pustaka

<b>Nama</b>	<b>Penelitian</b>	<b>Metode</b>	<b>Hasil</b>
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.
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 Wiefeling 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.

Berlanjut di halaman selanjutnya

**Table 2.1:** Lanjutan Tinjauan Pustaka

<b>Nama</b>	<b>Penelitian</b>	<b>Metode</b>	<b>Hasil</b>
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.
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**

### **DASAR TEORI**

#### **3.1 FHIR (*Fast Healthcare Interoperability Resources*)**

FHIR, singkatan dari Fast Healthcare Interoperability Resources, merupakan standar internasional yang diperkenalkan oleh Health Level Seven International (HL7) untuk memfasilitasi pertukaran data kesehatan elektronik. Standar ini dirancang untuk mengatasi tantangan interoperabilitas antara sistem-sistem informasi kesehatan yang beragam, dengan tujuan memungkinkan pertukaran data yang cepat, fleksibel, dan terstandarisasi di seluruh industri kesehatan.

FHIR menggunakan format data yang ringan seperti JSON atau XML, dan protokol komunikasi web standar seperti HTTP atau HTTPS, yang memfasilitasi integrasi dengan sistem-sistem modern dengan lebih mudah. Dengan pendekatan modular, FHIR memungkinkan akses granular terhadap informasi kesehatan, sesuai kebutuhan aplikasi atau pengguna.

Adopsi FHIR diharapkan dapat meningkatkan interoperabilitas di seluruh rantai perawatan kesehatan, memungkinkan pertukaran informasi yang lebih efisien dan akurat, serta mendukung pengembangan aplikasi kesehatan yang inovatif dan terintegrasi. Sebagai hasilnya, FHIR juga membuka pintu bagi pengembangan solusi-solusi teknologi kesehatan yang lebih canggih, seperti analisis big data dan kecerdasan buatan, serta integrasi dengan perangkat medis wearable.

#### **3.2 Autorisasi**

Otorisasi merujuk pada proses yang menentukan hak akses yang diberikan kepada entitas setelah autentikasi identitasnya berhasil dilakukan. Otorisasi memainkan peran penting dalam mengatur akses ke sumber daya dan layanan di dalam suatu sistem. Ini melibatkan penentuan apakah subjek atau entitas memiliki izin yang sesuai untuk melakukan tindakan tertentu dalam lingkungan yang diberikan. Proses otorisasi sering kali dilakukan setelah proses autentikasi yang sukses, di mana autentikasi memverifikasi identitas entitas. Dengan adanya otorisasi, sistem dapat memastikan bahwa hanya entitas yang memiliki hak yang sesuai yang diberikan akses ke sumber daya atau layanan tertentu, yang pada gilirannya membantu menjaga keamanan sistem secara keseluruhan. Misalnya, dalam sebuah aplikasi

perbankan, setelah seorang pengguna berhasil mengautentikasi identitasnya, proses otorisasi akan menentukan hak akses pengguna tersebut terhadap fungsi-fungsi seperti pengecekan saldo, transfer dana, atau pembayaran tagihan. Oleh karena itu, pemahaman yang mendalam tentang konsep otorisasi penting untuk merancang dan mengimplementasikan sistem informasi yang aman dan efektif.

### 3.3 Autentikasi

Autentikasi adalah konsep fundamental yang diperlukan untuk memvalidasi keaslian identitas entitas tertentu dalam suatu sistem. Identitas, sebagai inti dari autentikasi, merujuk pada informasi yang digunakan untuk mengidentifikasi subjek. Kredensial, sebagai elemen kunci dalam proses autentikasi, terdiri dari informasi otentikasi yang diperlukan untuk membuktikan identitas subjek, seperti kata sandi, token, atau biometrik.

Metode autentikasi beragam dan dapat mencakup kata sandi, token, biometrik, sertifikat digital, serta otorisasi multi-faktor (MFA). Protokol autentikasi, sebagai serangkaian langkah atau aturan, memberikan panduan bagi pelaksanaan autentikasi dalam suatu sistem, contohnya OAuth, OpenID, SAML, dan Kerberos.

Keamanan merupakan aspek krusial dalam autentikasi, yang mencakup kerahasiaan kredensial, integritas data autentikasi, dan non-repudiasi. Pemahaman akan kelemahan dan ancaman terhadap sistem autentikasi, seperti serangan phishing, brute force, dan man-in-the-middle, penting untuk meningkatkan ketahanan sistem.

Selain itu, autentikasi harus dapat diandalkan, sehingga sistem dapat memberikan verifikasi identitas yang konsisten dan akurat

#### 3.3.1 Standar Data Pada FHIR

Standar data pada FHIR (Fast Healthcare Interoperability Resources) didasarkan pada model data yang kuat dan fleksibel yang dirancang untuk memfasilitasi pertukaran informasi kesehatan elektronik secara efisien. Berikut adalah beberapa karakteristik utama dari standar data pada FHIR seperti yang dikutip dari (Sujudi, Heryawan et al., 2022):

- **Model Data Berbasis Sumber Daya (Resource-Based):** FHIR menggunakan pendekatan berbasis sumber daya di mana setiap entitas informasi kesehatan diwakili sebagai "sumber daya" yang dapat diakses dan dimanipulasi melalui



API RESTful. Contoh sumber daya dalam FHIR termasuk Pasien, Organisasi, dan Pemberi Layanan Kesehatan.

- **Representational State Transfer (RESTful) API:** FHIR menggunakan arsitektur RESTful API yang memungkinkan sistem untuk berkomunikasi dan bertukar data dengan cara yang sederhana dan efisien. API RESTful memungkinkan sistem untuk mengakses sumber daya kesehatan dan melakukan operasi CRUD (Create, Read, Update, Delete) pada sumber daya tersebut.
- **Format Data yang Fleksibel:** FHIR mendukung format data yang fleksibel, termasuk JSON (JavaScript Object Notation) dan XML (eXtensible Markup Language). Hal ini memungkinkan sistem kesehatan untuk berinteraksi dan berbagi data menggunakan format yang sesuai dengan kebutuhan mereka.
- **Terminologi Standar:** FHIR menggunakan terminologi standar seperti LOINC (Logical Observation Identifiers Names and Codes) dan SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) untuk memastikan konsistensi dan interoperabilitas data kesehatan di seluruh sistem.
- **Interoperabilitas yang Tinggi:** Dengan menggunakan standar data yang terdefinisi dengan jelas dan terstruktur, FHIR memungkinkan berbagai sistem kesehatan untuk berkomunikasi dan bertukar informasi tanpa hambatan. Hal ini meningkatkan interoperabilitas antara organisasi kesehatan dan memungkinkan pertukaran informasi yang lebih efisien.

Dengan karakteristik ini, standar data pada FHIR memberikan kerangka kerja yang kokoh dan terstandarisasi untuk pertukaran informasi kesehatan elektronik, memfasilitasi kolaborasi antara berbagai entitas kesehatan dan meningkatkan kualitas layanan pasien.

### 3.3.2 Standar Autentikasi Pada FHIR

Hubungan antara autentikasi dan FHIR berkaitan dengan keamanan dan akses kontrol dalam pertukaran data kesehatan elektronik. Autentikasi digunakan untuk memverifikasi identitas entitas yang terlibat dalam pertukaran data menggunakan standar FHIR. Setelah identitas tersebut diverifikasi, otorisasi diterapkan untuk menentukan hak akses entitas tersebut terhadap data yang disediakan oleh layanan FHIR.

Dalam konteks FHIR, autentikasi digunakan untuk memastikan bahwa entitas yang mencoba mengakses atau menyediakan data kesehatan melalui API FHIR adalah entitas yang sah. Ini bisa berarti memverifikasi identitas pengguna, aplikasi, atau sistem yang berusaha berinteraksi dengan layanan FHIR. Autentikasi bisa dilakukan menggunakan berbagai metode, seperti kata sandi, token, atau mekanisme autentikasi yang lebih kuat seperti sertifikat digital atau biometrik, tergantung pada kebutuhan dan kebijakan keamanan sistem.

Setelah autentikasi berhasil dilakukan, otorisasi diterapkan untuk menentukan apa yang diizinkan entitas tersebut lakukan dengan data yang tersedia melalui layanan FHIR. Misalnya, seorang dokter mungkin memiliki akses penuh untuk melihat dan mengubah catatan medis pasien tertentu, sementara seorang petugas administrasi hanya diizinkan untuk melihat informasi dasar pasien tanpa memiliki kemampuan untuk mengubahnya. Otorisasi dalam konteks FHIR memastikan bahwa akses ke data kesehatan dikontrol sesuai dengan kebutuhan dan kebijakan privasi yang berlaku.

Dengan demikian, autentikasi dan otorisasi berperan penting dalam menjaga keamanan dan kerahasiaan data kesehatan yang ditangani oleh layanan FHIR, memastikan bahwa hanya entitas yang berwenang yang dapat mengakses informasi yang sensitif dan penting tersebut.

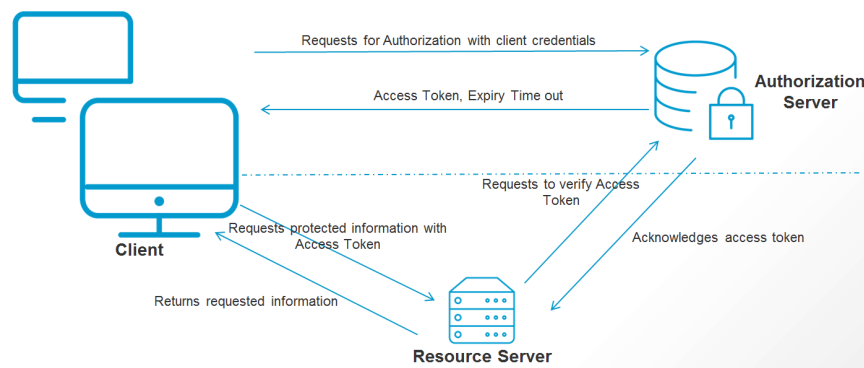
### **3.3.3 Autentikasi Mesin ke Mesin**

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 3.1.

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.

### **3.3.4 Metode Autentikasi Mesin ke Mesin**

Salah satu metode autentikasi Machine-to-Machine (M2M) menggunakan token merujuk pada proses verifikasi identitas antara dua atau lebih perangkat atau



Gambar 3.1: Skema M2M Authentication

sistem tanpa intervensi manusia. Dalam skenario ini, token digunakan sebagai kredensial atau kunci otentikasi yang diberikan kepada perangkat atau sistem untuk membuktikan identitasnya kepada sistem yang lain.

### ***Basic Access Authentication***

**Token** 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:

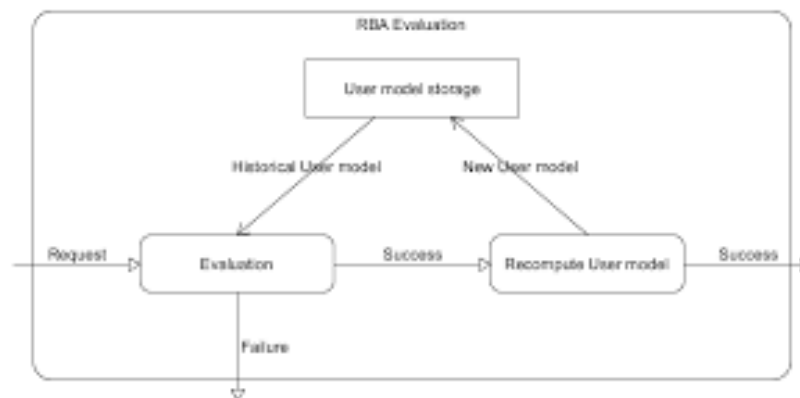
Tabel 3.2: Respon HTTP

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 (Smith & Jones, 2020).

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 (Chen et al.,



Gambar 3.2: Skema M2M Authentication

2018). Prosesnya dapat dilakukan dengan cara menganalisis faktor-faktor yang dapat meningkatkan risiko, seperti lokasi geografis, waktu akses, dan jenis perangkat yang digunakan.

Dalam penelitian ini, risk-based authentication hanya melibatkan dua output risiko, yaitu risiko tinggi dan risiko rendah, untuk menyederhanakan proses keputusan. 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 (Johnson & Walker, 2019).

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

### 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:

$$\text{imp}(\mathbf{t}_L) = \sum_{l=1}^2 p_{tL}(l)(1 - p_{tL}(l)) \quad (3.1)$$

Node Kanan:

$$\text{imp}(\mathbf{t}_R) = \sum_{l=1}^2 p_{tR}(l)(1 - p_{tR}(l)) \quad (3.2)$$

Node  $t$ :

$$\text{imp}(\mathbf{t}) = \sum_{k=1}^2 p_t(k)(1 - p_t(k)) \quad (3.3)$$

Keterangan:

$$p_t(k) = \frac{n_t(k)}{n_t} \quad \text{dan} \quad p_t(l) = \frac{n_t(l)}{n_t} \quad (3.4)$$

$$p_t(k), p_t(l) : \text{Proporsi objek kelas klasifikasi ke-}k \text{ atau ke-}l \text{ pada node } t \quad (3.5)$$

$$n_t(k), n_t(l) : \text{Jumlah observasi kelas klasifikasi ke-}k \text{ atau ke-}l \text{ pada node } t \quad (3.6)$$

$$n_t : \text{Jumlah seluruh observasi pada node } t \quad (3.7)$$

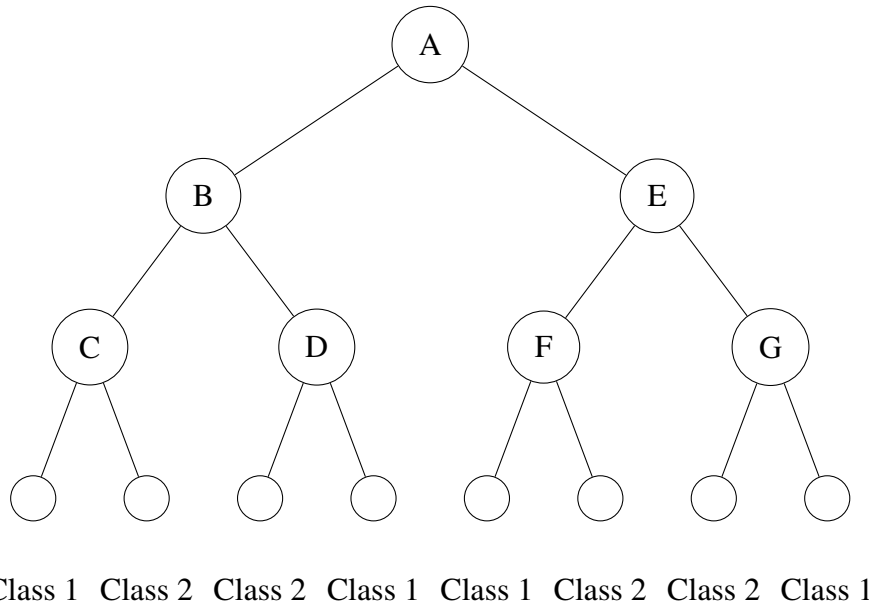
### 3.5.1 *Random Forest*

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).

Seperti dicontohkan pada Gambar 3.3 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



Gambar 3.3: Random Forest

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.

$$\text{Laju Galat OOB}_i = \frac{1}{n} \sum_{i=1}^n \mathbb{I}(Y_i \neq P_i) \quad (3.8)$$

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

### 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  $h=(1,2,\dots,p)$  maka rumus mengukur tingkat kepentingan peubah penjelas  $X_h$  dengan cara berikut (Xiao Li. dkk, 2019).

$$MDG(x_h) = \frac{1}{k} \sum_{t=1}^k MDG(X_h, x_t) \quad (3.9)$$

Keterangan:

$$MDG(X_h, x_t) = \sum_{t \in (T), v(t)=h} \frac{N_n(t)}{n} \Delta x(t) \quad (3.10)$$

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  $X_h$  dan menghitung perbedaannya, dengan rumus sebagai berikut Strobl dkk. (2008)

$$MDA(x_h) = \frac{1}{k} \sum_{t=1}^k \sum_{i \in OOB(t)} \frac{I(y_i = \hat{y}_i(t)) - I(y_i = \hat{y}_i, h(t))}{|OOB(t)|} \quad (3.11)$$

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

### 3.5.4 Metriks dan Scoring

Ada beberapa cara untuk mengukur kinerja pengklasifikasi, tetapi yang paling umum adalah menggunakan matriks kebingungan, presisi, recall, dan skor F1.

*Confusion matrix* atau dikenal juga dengan matriks kebingungan adalah cara untuk mengekspresikan berapa banyak prediksi pengklasifikasi yang benar,

dan ketika salah, di mana pengklasifikasi mengalami kebingungan (sesuai dengan namanya). Pada matriks kebingungan di bawah ini, baris mewakili label yang benar dan kolom mewakili label yang diprediksi. Nilai pada diagonal mewakili jumlah (atau persen, dalam matriks kebingungan yang dinormalisasi) dari waktu di mana label yang diprediksi cocok dengan label yang sebenarnya. Nilai di sel lainnya mewakili contoh di mana pengklasifikasi salah memberi label pada pengamatan; kolom memberi tahu kita apa yang diprediksi oleh pengklasifikasi, dan baris memberi tahu kita apa label yang benar.

Presisi adalah jumlah anggota kelas yang diidentifikasi dengan benar dibagi dengan semua kali model memprediksi kelas tersebut. Dalam kasus Aspens, skor presisi adalah jumlah Aspens yang diidentifikasi dengan benar dibagi dengan jumlah total kali pengklasifikasi memprediksi Aspen, baik benar maupun salah.

Recall adalah jumlah anggota kelas yang diidentifikasi dengan benar oleh pengklasifikasi dibagi dengan jumlah total anggota dalam kelas tersebut. Untuk Aspen, ini adalah jumlah Aspen aktual yang diidentifikasi dengan benar oleh pengklasifikasi.

Skor F1 sedikit kurang intuitif karena menggabungkan presisi dan recall ke dalam satu metrik. Jika presisi dan recall keduanya tinggi, F1 juga akan tinggi. Jika keduanya rendah, F1 akan rendah. Jika salah satunya tinggi dan yang lainnya rendah, F1 akan rendah. F1 adalah cara cepat untuk mengetahui apakah pengklasifikasi benar-benar baik dalam mengidentifikasi anggota kelas, atau apakah pengklasifikasi menemukan jalan pintas (misalnya, hanya mengidentifikasi segala sesuatu sebagai anggota kelas yang besar).

## **BAB IV**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **4.1 Deskripsi Umum Sistem**

Analisis sistem terdiri dari gambaran umum sistem yang dapat dilihat pada Gambar 4.1, deskripsi sistem, dan diagram alir sistem. Gambaran umum sistem menjelaskan secara singkat tentang sistem yang akan dibangun. Deskripsi sistem menjelaskan tentang sistem yang akan dibangun secara rinci. Diagram alir sistem menjelaskan tentang alur kerja sistem yang akan dibangun.



Gambar 4.1: Gambaran Umum Sistem

#### **4.2 Analisis Kebutuhan Sistem**

Dalam membangun sistem ini, diperlukan analisa kebutuhan fungsional. Kebutuhan fungsional adalah kebutuhan yang berkaitan dengan fungsi-fungsi yang harus ada dalam sistem. Serta akan dijelaskan kebutuhan perangkat keras dan perangkat lunak yang dibutuhkan dalam membangun sistem ini.

##### **4.2.1 Analisis 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.

##### **4.2.2 Analisis Kebutuhan Perangkat Keras**

1. Laptop atau PC dengan RAM minimal 8gb.

2. Processor dengan minimum 5 CPU Core.
3. *Storage* dengan minimum 50gb.
4. Sistem operasi dengan base unix untuk menjalankan sistem klasifikasi.

#### **4.2.3 Analisis Kebutuhan Perangkat Lunak**

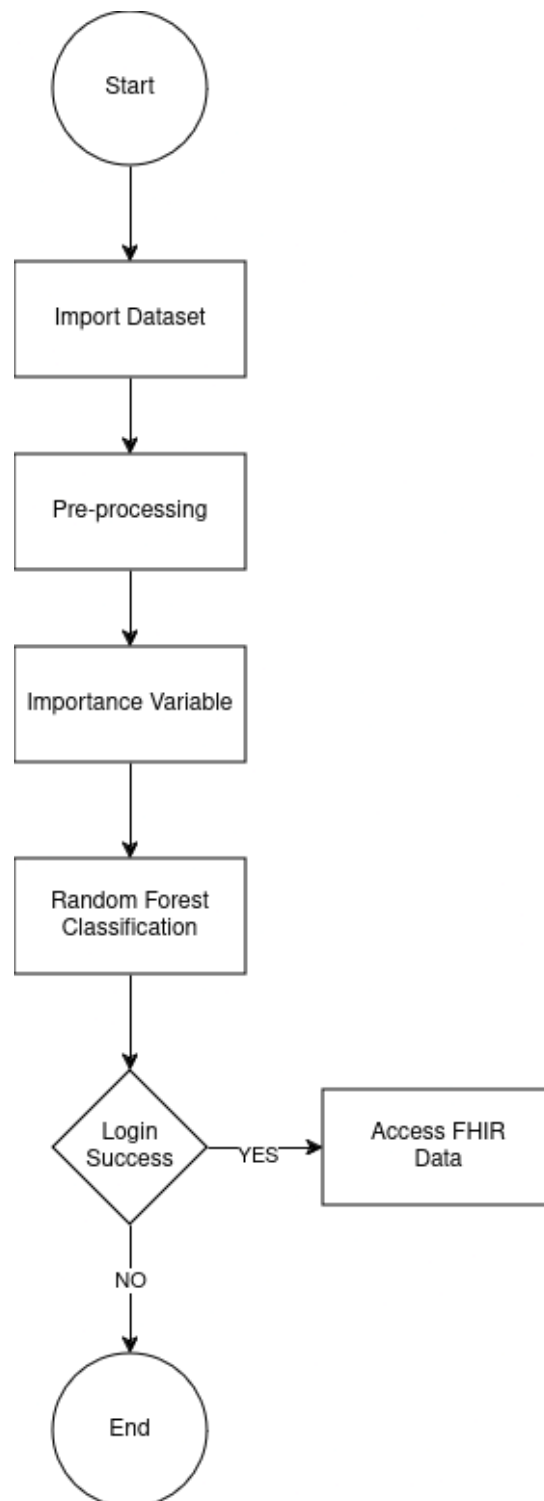
1. Bahasa pemrograman python versi 3.9 dengan *framework* anaconda 3
2. *Framework* flask untuk membuat endpoint
3. Sistem operasi dengan base unix untuk menjalankan sistem klasifikasi

### **4.3 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.

#### **4.3.1 Rancangan Arsitektur Sistem**

Rancangan arsitektur sistem dapat dilihat pada Gambar 4.2. 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

Pada Gambar 4.2, mengilustrasikan proses sistem yang menggabungkan

analisis dataset menggunakan klasifikasi Random Forest dan mekanisme pengendalian akses terhadap data FHIR. Proses dimulai dengan mengimpor dataset yang akan digunakan untuk analisis selanjutnya. Setelah dataset diimpor, langkah berikutnya adalah melakukan pemrosesan awal (pre-processing) guna membersihkan dan mempersiapkan data. Tahapan pre-processing ini meliputi penanganan data yang hilang, normalisasi, dan transformasi data agar siap untuk diterapkan dalam algoritma pembelajaran mesin.

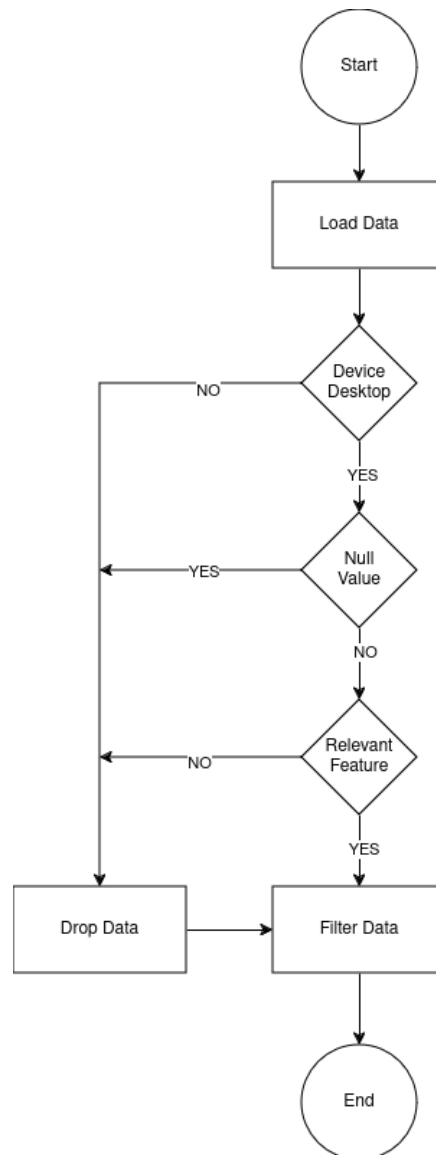
Berikutnya, tahap identifikasi variabel penting dilakukan untuk menentukan fitur yang signifikan dalam dataset. Proses ini mungkin melibatkan analisis statistik atau teknik machine learning untuk memastikan variabel-variabel yang digunakan adalah yang paling relevan untuk klasifikasi. Setelah variabel penting diidentifikasi, data tersebut kemudian diklasifikasikan menggunakan algoritma Random Forest, sebuah metode pembelajaran ensemble yang memanfaatkan beberapa pohon keputusan untuk mencapai hasil yang lebih akurat dan dapat diandalkan.

Langkah selanjutnya adalah verifikasi login pengguna. Sistem akan memeriksa apakah login pengguna berhasil atau tidak. Jika login berhasil, pengguna akan diberikan akses ke data FHIR (Fast Healthcare Interoperability Resources), sebuah standar untuk pertukaran informasi kesehatan elektronik. Jika login gagal, proses akan berakhir tanpa memberikan akses lebih lanjut.

Secara keseluruhan, diagram ini mengintegrasikan metode klasifikasi data dengan langkah verifikasi pengguna untuk mengendalikan akses terhadap data sensitif. Proses ini sangat berguna dalam konteks di mana prediksi berbasis data dan akses berbasis autentikasi sangat penting, seperti dalam analisis data kesehatan atau medis. Implementasi ini memastikan keamanan dan integritas data, sekaligus memberikan hasil analisis yang akurat dan dapat diandalkan.

#### **4.3.2 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

Dalam melakukan pembersihan data, sistem ini akan dua metode yaitu :

1. *Missing Value* : Menghapus data yang memiliki nilai kosong.
2. *Duplicate Elimination* : Menghapus duplikasi data sehingga hanya satu dari data duplikat yang disimpan.

Pembersihan tahap satu dapat dilakukan menggunakan fitur pandas yaitu `isnull`. Setelah itu didapatkan jumlahnya dengan `sum`. Dengan cara ini didapatkan jumlah data kosong untuk setiap fitur. Untuk fitur yang terdapat nilai kosong akan

dibuang Pembersihan tahap dua dilakukan dengan cara menyaring fitur *User Agent and Device Type* Setelah data dibersihkan, data akan digunakan sebagai input untuk melakukan analisis risiko autentikasi. Data ini kemudian akan digunakan sebagai input untuk melakukan analisis risiko autentikasi.

#### 4.3.3 Rancangan *Encoding*

One-hot encoding adalah teknik pengkodean kategoris yang umum digunakan dalam pemrosesan data. Mengutip dari James, G, et al. (2013) teknik ini cocok untuk variabel kategori di mana kategori tidak memiliki urutan atau hubungan yang melekat dengan nilai numeriknya. Ide di balik pengkodean one-hot adalah untuk mewakili setiap kategori sebagai vektor biner yang panjangnya sama dengan jumlah kategori unik dalam variabel tersebut.

Tabel 4.1: Sample Encoding Data

login_timestamp_1	login_timestamp_2	ip_address_1	ip_address_2	country_1	country_2	region_1	region_2	city_1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1

Cara kerja one-hot encoding adalah seperti ditunjukkan pada Tabel 4.1 dengan membuat kolom biner baru untuk setiap kategori dalam kolom kategori. Jika terdapat N kategori unik, maka akan dibuat N kolom biner baru. Setiap kolom biner akan memiliki nilai 1 jika kategori tersebut hadir dalam observasi, dan nilai 0 jika tidak hadir. Dengan demikian, setiap observasi direpresentasikan sebagai vektor biner dengan panjang N, di mana setiap elemen vektor menunjukkan keberadaan atau ketiadaan kategori yang sesuai dalam observasi. Teknik ini memungkinkan model machine learning untuk memahami dan memproses variabel kategori dalam bentuk yang sesuai dengan algoritma yang digunakan, seperti algoritma regresi logistik atau jaringan saraf.

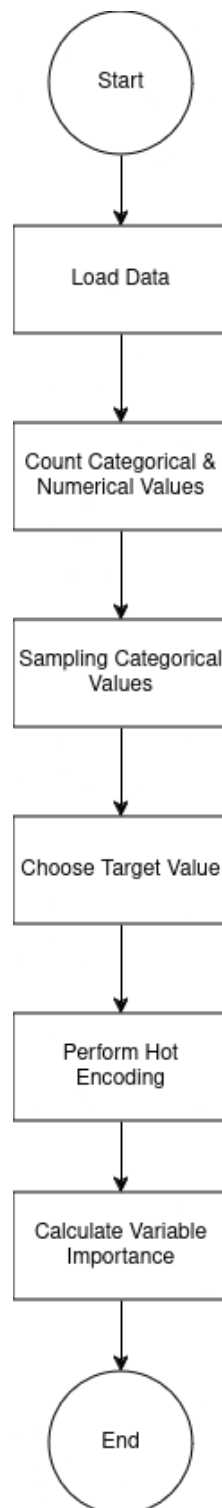
Meskipun memiliki kelebihan, One-Hot Encoding juga memiliki beberapa kekurangan signifikan. Salah satu kekhawatiran utama adalah potensi untuk menghasilkan data berdimensi tinggi. Untuk fitur kategorikal dengan banyak nilai unik, One-Hot Encoding dapat menghasilkan sejumlah besar fitur biner, yang dapat meningkatkan biaya komputasi dan risiko overfitting. Masalah ini terutama terlihat pada dataset dengan kardinalitas tinggi, di mana jumlah fitur biner yang baru dapat menjadi sangat banyak. Selain itu, peningkatan dimensi dapat menyebabkan penggunaan memori yang lebih tinggi, terutama dalam dataset besar dengan



banyak variabel kategorikal. Kebutuhan memori tambahan ini dapat mempengaruhi kecepatan dan efisiensi proses pelatihan.

#### **4.3.4 Rancangan *Variable Importance Measure***

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

Gambar 4.4 menggambarkan proses analisis data secara struktural yang

digunakan dalam pengolahan data untuk tujuan tertentu. Proses dimulai dengan memuat data (*Load Data*) yang akan digunakan dalam analisis. Setelah data dimuat, langkah berikutnya adalah menghitung nilai kategorikal dan numerik dalam dataset (*Count Categorical & Numerical Values*) guna memahami distribusi serta tipe data yang ada.

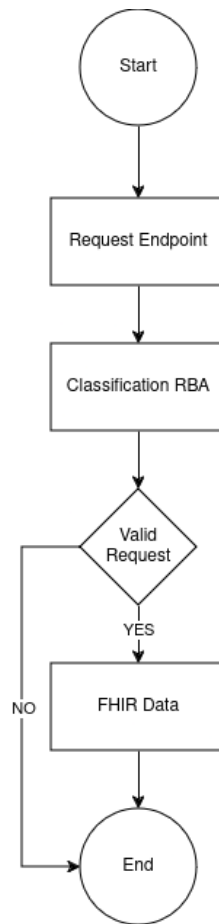
Setelah nilai-nilai tersebut dihitung, proses berlanjut dengan pengambilan sampel nilai kategorikal (*Sampling Categorical Values*). Pengambilan sampel ini bertujuan untuk memperoleh representasi yang lebih seimbang dan bermakna dari kategori yang ada dalam dataset. Kemudian, target value yang akan dianalisis ditentukan (*Choose Target Value*), yang menjadi fokus utama dalam proses analisis data lanjutan.

Langkah selanjutnya adalah melakukan penyandian hot encoding (*Perform Hot Encoding*) pada nilai kategorikal. Hot encoding adalah teknik yang mengubah berbagai nilai kategorikal menjadi format numerik biner, sehingga bisa digunakan dalam algoritma pembelajaran mesin. Setelah proses hot encoding selesai, tahapan berikutnya adalah menghitung tingkat pentingnya variabel (*Calculate Variable Importance*). Ini merupakan analisis untuk mengidentifikasi variabel mana yang paling berpengaruh terhadap target value yang telah dipilih sebelumnya.

Akhirnya, setelah seluruh proses perhitungan dan penyandian selesai, proses berakhir (*End*), menandakan bahwa data telah siap untuk tahap analisis atau pemodelan lebih lanjut.

#### **4.3.5 Rancangan Integrasi 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

Gambar 4.5 dijelaskan proses penggunaan model prediksi Random Forest dengan API, dan bagaimana permintaan data FHIR (Fast Healthcare Interoperability Resources) diproses. Proses dimulai dengan menerima permintaan (Request Endpoint) dari pengguna untuk memperoleh prediksi. Setelah menerima permintaan, sistem kemudian melakukan klasifikasi RBA (Risk-Based Authentication), sebuah langkah untuk mengidentifikasi dan mengevaluasi risiko terkait permintaan tersebut.

Pada tahap berikutnya, sistem mengevaluasi validitas permintaan melalui titik keputusan (Valid Request). Jika permintaan tersebut valid, maka data FHIR yang diminta akan dikirimkan (FHIR Data) dan proses berakhir. Namun, jika permintaannya tidak valid, proses dihentikan tanpa mengakses data FHIR.

Dalam konteks ini, API schema berikut menjelaskan struktur yaml yang digunakan untuk melakukan permintaan prediksi menggunakan model Random Forest:

---

```

1  tags:
2    - Prediction
3  parameters:
4    - name: body
5      in: body
6      required: true
7      schema:
8        type: object
9        required:
10         - features
11         - patient_id
12      properties:
13        features:
14          type: array
15          items:
16            type: number
17          example: [5.1, 3.5, 1.4, 0.2]
18        patient_id:
19          type: string
20          example: "example_patient_id"
21  responses:
22    200:
23      description: Prediction result
24      schema:
25        type: object
26        properties:
27          prediction:
28            type: integer
29            example: 1
30          observation:
31            type: object
32    400:
33      description: Bad Request
34      schema:
35        type: object
36        properties:
37          error:
38            type: string
39            example: Missing features key in the JSON data
40    500:
41      description: Internal Server Error
42      schema:
43        type: object
44        properties:
45          error:
46            type: string
47            example: Internal error message

```

---

Schema yaml tersebut digunakan dalam permintaan untuk prediksi dengan Random Forest Model. Permintaan ini harus menyertakan dua parameter penting: `features` dan `patient_id`. Parameter `features` adalah array berisi nilai numerik yang akan digunakan model untuk prediksi, sementara `patient_id` adalah string pengidentifikasi pasien yang bersangkutan. Respon yang diberikan oleh API memiliki format JSON yang mencakup hasil prediksi (`prediction`) dan observasi tambahan (`observation`), atau deskripsi kesalahan jika terjadi kesalahan saat pemrosesan.

Secara keseluruhan, diagram dan schema ini menjelaskan alur proses yang harus diikuti untuk melakukan prediksi menggunakan Random Forest Model dan mengakses data FHIR, yang diawali dari penerimaan permintaan hingga penyediaan hasil atau tanggapan kesalahan. Pendekatan ini memastikan adanya langkah-langkah yang terstruktur dan aman dalam pengelolaan serta pemanfaatan data.

#### **4.4 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. Menentukan Evaluasi : Akurasi, Presisi, *Recall*, *F1 Score*, dan *Confusion Matrix* akan digunakan untuk menentukan evaluasi dari sistem.
3. Membandingkan dengan autentikasi heuristik lainnya

## BAB V

### IMPLEMENTASI SISTEM

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:

Tabel 5.1: Deskripsi tabel fitur login

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; ...
Lanjutan pada halaman berikutnya			

<b>Feature</b>	<b>Data Type</b>	<b>Description</b>	<b>Range or Example</b>
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’)



## 5.2 *Preprocessing Data*

Penggunaan dataset dalam penelitian ini membutuhkan beberapa tahapan persiapan data, yaitu pengumpulan data, pembersihan data, dan pemilihan fitur.

### 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.

### 5.2.2 *Pengecekan Missing Value*

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

---

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

---

hasilnya adalah sebagai berikut:

Tabel 5.2: Missing Values in Each Feature

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

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.

### 5.2.3 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.

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

---

```

1  # calculate the percentage of True and False values in boolean char'
2  value_counts_1 = df['is_login_success'].value_counts(normalize=True)
3  is_login_success_true = value_counts_1[True] * 100

```

```

4     is_login_success_false = value_counts_1[False] * 100
5     print("is_login_success")
6     print(f"Percentage of True values: {is_login_success_true:.2f}%")
7     print(f"Percentage of False values: {is_login_success_false:.2f}%")
8
9     value_counts_2 = df['is_attack_ip'].value_counts(normalize=True)
10    is_attack_ip_true = value_counts_2[True] * 100
11    is_attack_ip_false = value_counts_2[False] * 100
12    print("is_attack_ip")
13    print(f"Percentage of True values: {is_attack_ip_true:.2f}%")
14    print(f"Percentage of False values: {is_attack_ip_false:.2f}%")
15
16    value_counts_3 = df['is_account_takeover'].value_counts(normalize=True)
17    is_account_takeover_true = value_counts_3[True] * 100
18    is_account_takeover_false = value_counts_3[False] * 100
19    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}%")

```

Berikut adalah hasil sampling data.

Tabel 5.3: 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%

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.

#### 5.2.4 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.

```

1     # generate SHA512 Hash from user_id as m2m token
2     import hashlib
3
4     def generate_sha512_hash(user_id):
5         sha512_hash = hashlib.sha512()
6         sha512_hash.update(str(user_id).encode('utf-8'))
7         return sha512_hash.hexdigest()
8
9     features['token'] = features['user_id'].apply(generate_sha512_hash)

```

Berikut adalah contoh token yang digenerate.

Tabel 5.4: Contoh Token

User ID	Token
-3065936140549856249	4ffe29f1960c24624ec2c36909f3b39cb8d59fa18515f4
5932501938287412564	ecee6cc95d3b047c8f796b8e772a468124b7ddb599a7a3

### 5.3 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.

#### 5.3.1 Penamaan Kolom

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

---

```

1  # rename above columns to snake case
2  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'})

```

---

Tabel 5.5: Column Renaming in DataFrame

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

### 5.3.2 Penyaringan Data

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())

```

---

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

```

---

Setelah itu dilakukan penyaringan data dengan cara menghapus data yang

memiliki device type selain 'desktop'.

### 5.3.3 Penghapusan Kolom

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.

---

```

1  # drop unsued columns
2  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.

Tabel 5.6: Revised Initial Exploratory Data Analysis

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

Berdasarkan tabel 5.6, 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'.

## 5.4 Pemilihan Fitur Menggunakan *Variable Importance Measure*

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.

Sebelum itu dilakukan eksplorasi data untuk mengetahui jumlah baris dan kolom, tipe data, dan statistik deskriptif dari 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']

```

---

Berikut adalah hasil keluaran dari tahap ini.

Tabel 5.7: Data Type of Each Column

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

Berdasarkan tabel 5.7, terlihat bahwa kolom 'ASN' memiliki tipe data numerik, sedangkan kolom lainnya memiliki tipe data string.

#### 5.4.1 Encoding Data

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',
7  'region', 'city', 'user_agent_string', 'browser', 'os_detail'])
8  X_train = encoder.fit_transform(X_train)
9
10 X_test = encoder.transform(X_test)
11 X_train.head()

```

---

Berikut adalah hasil keluaran dari tahap ini.

### 5.4.2 Gini Importance

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

---

```

1  ### Gini importance
2  # create the classifier with n_estimators = default
3  clf = RandomForestClassifier(random_state=0)
4
5  # fit the model to the training set
6  clf.fit(X_train, y_train)
7
8  # view the feature scores
9  feature_scores = pd.Series(clf.feature_importances_,
10 index=X_train.columns).sort_values(ascending=False)
11
12 # Top 10 important features
13 feature_scores.head(10)

```

---

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



Tabel 5.8: Gini Importance of Each Feature

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

Dalam tabel 5.8, jika di lakukan 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.

## 5.5 Implementasi *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.

### 5.5.1 Pembagian Data

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

---

```

1  # Separate the features (X) and the target (y)
2  X = df_encoded.drop(columns=['is_login_success'])
3  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.

**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.

---

```

1  # Split the data into training and test sets
2  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.

### 5.5.2 Pembuatan 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.

### 5.5.3 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)

```

---

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,

### 5.5.4 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.

## 5.6 Implementasi Sistem FHIR

Pada bagian ini dijelaskan proses integrasi antara model Random Forest dengan data FHIR.

### 5.6.1 Implementasi API

### 5.6.2 Implementasi API

Di bawah ini adalah implementasi Flask API yang menggunakan model Random Forest untuk prediksinya. API terintegrasi dengan klien tiruan FHIR untuk pengambilan data pasien.

### 5.6.3 API Code

---

```

1  # api.py
2  from flask import Flask, request, jsonify
3  import joblib
4  import numpy as np
5  from flasgger import Swagger
6  from mock_fhir import MockFHIRClient # Import the mock FHIR client
7
8  app = Flask(__name__)
9  swagger = Swagger(app)
10
11 # Load the trained model from the pickle file
12 model = joblib.load('./rfc_model.pkl')
13
14 # Initialize the mock FHIR client
15 fhir_client = MockFHIRClient()
16
17 @app.route('/')
18 def index():
19     return "Random Forest API is up and running!"
20
21 @app.route('/predict', methods=['POST'])
22 def predict():
23     # Function documentation
24     """
25     Predict using Random Forest Model
26     ---
27     tags:
28     - Prediction
29     parameters:
30     - name: body
31       in: body
32       required: true
33     schema:
34       type: object
35       required:
36       - features
37       - patient_id
38     properties:
39       features:

```

```

40         type: array
41         items:
42             type: number
43         example: [5.1, 3.5, 1.4, 0.2]
44     patient_id:
45         type: string
46         example: "example_patient_id"
47 responses:
48     200:
49         description: Prediction result
50         schema:
51             type: object
52             properties:
53                 prediction:
54                     type: integer
55                     example: 1
56                 observation:
57                     type: object
58     400:
59         description: Bad Request
60         schema:
61             type: object
62             properties:
63                 error:
64                     type: string
65                 example: Missing features key in the JSON data
66     500:
67         description: Internal Server Error
68         schema:
69             type: object
70             properties:
71                 error:
72                     type: string
73                 example: Internal error message
74 """
75 if request.method == 'POST':
76     try:
77         # Get the JSON from the request
78         data = request.get_json()
79
80         # Ensure the 'features' key is in the incoming JSON data
81         if 'features' not in data:
82             return jsonify({'error': 'Missing features key in the JSON data'}), 400
83
84         # Ensure the 'patient_id' key is in the incoming JSON data
85         if 'patient_id' not in data:
86             return jsonify({'error': 'Missing patient_id key in the JSON data'}),
87                 400
88
89         # Extract patient ID
90         patient_id = data['patient_id']
91
92         # Convert data into a numpy array
93         input_features = np.array(data['features']).reshape(1, -1)
94
95         # Make a prediction
96         prediction = model.predict(input_features)
97         if prediction[0] == 0:
98             return jsonify({'error': 'User not authorized'}), 401
99
100        # Store prediction result as an Observation
101        observation_result = fhir_client.get_patient(patient_id)

```

```

102         # Send the response back to the client
103         return jsonify({
104             'prediction': int(prediction[0]),
105             'observation': observation_result
106         })
107     except Exception as e:
108         return jsonify({'error': str(e)}), 500
109
110 if __name__ == '__main__':
111     app.run(debug=True)

```

---

**Memulai Aplikasi dan Inisialisasi Swagger** Aplikasi Flask diinisialisasi, dan Swagger digunakan untuk mendokumentasikan API.

#### Listing V.1: Inisialisasi Flask dan Swagger

---

```

1 # api.py
2 from flask import Flask, request, jsonify
3 import joblib
4 import numpy as np
5 from flasgger import Swagger
6 from mock_fhir import MockFHIRClient # Import the mock FHIR client
7
8 app = Flask(__name__)
9 swagger = Swagger(app)

```

---

**Memuat Model Terlatih** Model Random Forest yang sudah dilatih dimuat dari file .pkl.

#### Listing V.2: Memuat Model

---

```

1 # Load the trained model from the pickle file
2 model = joblib.load('./rfc_model.pkl')

```

---

**Inisialisasi Mock FHIR Client** Klien mock FHIR diinisialisasi untuk mensimulasikan interaksi dengan sistem FHIR.

#### Listing V.3: Inisialisasi Mock FHIR Client

---

```

1 # Initialize the mock FHIR client
2 fhir_client = MockFHIRClient()

```

---

**Endpoint Awal** Endpoint utama / hanya mengembalikan pesan bahwa API berjalan.

#### Listing V.4: Endpoint Utama

---

```

1 @app.route('/')
2 def index():
3     return "Random Forest API is up and running!"

```

---

**Endpoint Prediksi** Endpoint /predict menerima permintaan POST dengan JSON yang berisi fitur dan patient\_id, lalu mengembalikan prediksi dan observasi.

#### 5.6.4 FHIR Data

Kode berikut mendefinisikan klien tiruan FHIR yang digunakan untuk mensimulasikan interaksi dengan server FHIR.

Listing V.5: Mock FHIR Client Implementation

---

```

1 # mock_fhir.py
2 class MockFHIRClient:
3     def __init__(self):
4         self.dummy_patient_data = {
5             "resourceType": "Patient",
6             "id": "example_patient_id",
7             "name": [{ "use": "official", "family": "Doe", "given": ["John"] }],
8             "gender": "male",
9             "birthDate": "1990-01-01"
10        }
11
12    def get_patient(self, patient_id):
13        if patient_id == self.dummy_patient_data["id"]:
14            return self.dummy_patient_data
15        else:
16            return {"error": f"Patient with ID {patient_id} not found."}
17
18    def create_observation(self, patient_id, prediction):
19        dummy_observation = {
20            "resourceType": "Observation",
21            "status": "final",
22            "category": [{
23                "coding": [{
24                    "system":
25                        "http://terminology.hl7.org/CodeSystem/observation-category",
26                    "code": "laboratory",
27                    "display": "Laboratory"
28                }]
29            }],
30            "code": {
31                "coding": [{
32                    "system": "http://loinc.org",
33                    "code": "1975-2",
34                    "display": "Prediction Result"
35                }]
36            },
37            "subject": {
38                "reference": f"Patient/{patient_id}"
39            },
40            "valueQuantity": {
41                "value": prediction,
42                "unit": "class",
43                "system": "http://unitsofmeasure.org",
44                "code": "class"
45            }
46        }
47        return dummy_observation

```

---



## **BAB VI**

### **HASIL DAN PEMBAHASAN**

#### **6.1 Hasil Pengujian**

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. Dilakukan perbandingan dengan penelitian lain dengan tema serupa namun dengan metode heuristik, seperti pada penelitian Kalantzis et al (2022)

##### **6.1.1 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, dilakukan pengujian untuk beberapa ukuran dataset. Ukuran dataset yang digunakan adalah 10000, 20000, 30000, 40000 dan 50000. Berikut adalah hasil pengujian waktu data training dapat dilihat pada 6.1.

Tabel 6.1: Hasil Pengujian Waktu Data Training Random Forest

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

##### **6.1.2 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.

Tabel 6.2: Hasil Pengujian Penggunaan CPU dan Memory Data Training

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

### 6.1.3 Hasil 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  $5 \times 5 \times 5 = 625$  percobaan.

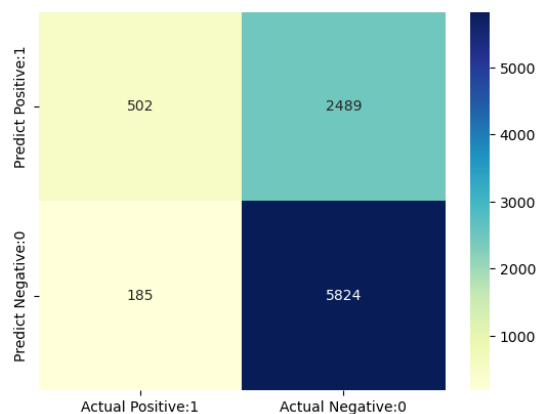
Tabel 6.3: Parameter Grid

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

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

## 6.2 Analisis Hasil Pengujian

Analisis hasil pengujian berisi analisis terhadap hasil pengujian yang telah dilakukan. Analisis dilakukan dengan membandingkan hasil pengujian



Gambar 6.1: Confusion Matrix

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

Tabel 6.4: Confusion Matrix

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

Hasil tertinggi yang diperoleh dari Tabel ?? adalah sebagai berikut:

Tabel 6.5: Hasil Pengujian Random Forest

Parameter	Values
n_estimators	500
max_depth	None
min_samples_split	2
min_samples_leaf	2

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

### 6.2.1 *Random Forest Risk Based Authentication*

**Analisis Kuantitatif:** Random Forest memiliki beberapa keunggulan dalam konteks autentikasi mesin ke mesin. Pertama, Random Forest dapat memberikan akurasi yang rendah dalam memprediksi perilaku pengguna dan mengidentifikasi aktivitas yang mencurigakan. Algoritme ini efisien dalam pengolahan data kompleks dan tidak linear, cocok untuk situasi di mana terdapat banyak variabel dan interaksi antara variabel tersebut.

**Analisis Kualitatif:** Di sisi lain, Random Forest mampu menangani pengambilan keputusan yang kompleks dengan memodelkan hubungan yang kompleks antara variabel-variabel dalam data, sehingga cocok untuk kasus-kasus autentikasi yang kompleks. Namun, untuk membangun model yang akurat, Random Forest membutuhkan data yang memadai untuk melatihnya, yang mungkin sulit diperoleh terutama dalam konteks keamanan informasi yang sensitif.

**Kelebihan:** Salah satu kelebihan utama Random Forest adalah kemampuannya dalam menangani data yang kompleks dan mengurangi risiko overfitting. Cocok untuk situasi di mana terdapat banyak variabel dan interaksi antara variabel tersebut.

**Kelemahan:** Namun, penggunaan Random Forest juga memiliki beberapa kelemahan. Pertama, memproses ensambel pohon keputusan dapat membutuhkan sumber daya komputasi yang besar. Selain itu, kinerja Random Forest dapat bervariasi tergantung pada pengaturan hyperparameter yang dipilih.

### 6.2.2 *Heuristic Authentication*

**Analisis Kuantitatif:** Penelitian Kalantzis (2022) mencakup sejumlah besar eksperimen yang terperinci, termasuk eksperimen untuk pengguna dan penyerang. Sebagai contoh, satu bagian mencakup 30 eksperimen untuk pengguna dan 1190 untuk penyerang.

**Analisis Kualitatif:** dengan penggunaan metrik yang terperinci seperti False Acceptance Rate (FAR) dan False Rejection Rate (FRR), yang dilaporkan pada berbagai konfigurasi dan pengguna. Paper ini juga membandingkan kinerja metodologi yang diusulkan dengan model lain melalui analisis komparatif dan menggunakan grafis untuk menggambarkan peningkatan dalam metrik FAR dan FRR secara visual. Seperti yang digunakan pada penelitian Feth (2019)

**Kelebihan:** Keuntungan (pros) dari pendekatan ini termasuk pengujian yang sangat komprehensif, yang memberikan keandalan tinggi pada temuan yang dihasilkan, serta metodologi yang menunjukkan peningkatan yang jelas dalam metrik FAR sebesar 29% dan FRR sebesar 11%. Metode yang rinci dan penggunaan tabel serta grafik untuk menyajikan data eksperimen membantu pemahaman tentang peningkatan yang dicapai

**Kelemahan:** Namun, ada beberapa kelemahan (cons) yang perlu diperhatikan. Salah satunya adalah bias model autoencoder terhadap prediksi yang positif, yang dapat mempengaruhi kemampuan generalisasi model. Selain itu, meskipun ada peningkatan signifikan dalam FAR, peningkatan FRR kurang mengejutkan, yang menunjukkan variabilitas dalam kinerja model. Nilai awal beberapa metrik sudah rendah, membatasi potensi peningkatan lebih lanjut, terutama pada FRR.

### 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.

Random Forest cenderung tidak lebih akurat dan dapat menangani kasus yang lebih kompleks, namun memerlukan lebih banyak sumber daya komputasi dan data. Sementara itu, Heuristic Authentication lebih sederhana namun mungkin kurang akurat dalam kasus-kasus yang kompleks. Pemilihan antara kedua pendekatan ini harus mempertimbangkan kebutuhan spesifik organisasi dan lingkungan operasionalnya.

## **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 mengklasifikasi 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. Optimasi parameter Random Forest masih dapat dilakukan lebih lanjut.
4. Dapat dilakukan perbandingan dengan memilih target parameter yang berbeda.

## DAFTAR PUSTAKA

- Agarwal, L., Khan, H., & Hengartner, U. (2016). Ask Me Again But Don't Annoy Me: Evaluating Re-authentication Strategies for Smartphones. 221–236. <https://www.usenix.org/conference/soups2016/technical-sessions/presentation/agarwal>
- Alam, M. S., & Vuong, S. T. (2013). 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, 663–669. <https://doi.org/10.1109/greencom-ithings-cpscom.2013.122>
- Braunstein, Mark L. (2022). FHIR. Computers in Health Care, 233–291. [https://doi.org/10.1007/978-3-030-91563-6\\_9](https://doi.org/10.1007/978-3-030-91563-6_9)
- Cabarcos, P. A., Arias-Cabarcos, P., Krupitzer, C., & Becker, C. (2019). A Survey on Adaptive Authentication. ACM Computing Surveys, 52(4), 80. <https://doi.org/10.1145/3336117>
- Doerfler, P., Thomas, K., Marincenko, M., Ranieri, J., Jiang, Y., Moscicki, A., & McCoy, D. (2019). Evaluating Login Challenges as a Defense Against Account Takeover. 372–382. <https://doi.org/10.1145/3308558.3313481>
- Dutson, J., Allen, D., Eggett, D. L., & Seamons, K. E. (2019). Don't Punish all of us: Measuring User Attitudes about Two-Factor Authentication. 119–128. <https://doi.org/10.1109/eurospw.2019.00020>
- Feth, Denis, dan Svenja Polst. "Heuristics and Models for Evaluating the Usability of Security Measures." Dalam Proceedings of Mensch und Computer 2019, 275–85. MuC '19. New York, NY, USA: Association for Computing Machinery, 2019. doi:10.1145/3340764.3340789.
- Misbahuddin, M., B. S. Bindhumadhava, B. S. Bindhumadhava, Bindhumadhava, B. S., & Dheeptha, B. (2017). Design of a risk-based authentication system using machine learning techniques. 1–6. <https://doi.org/10.1109/uic-atc.2017.8397628>
- Prasad, K. K., K, K. P., & Aithal, S. (2017). A Study on Enhancing Mobile Banking Services Using Location Based Authentication. <https://doi.org/10.47992/ijmts.2581.6012.0006>

- Rahat, Tamjid Al, Feng, Yu, & Tian, Yuan. (2021). Cerberus. Cornell University - ArXiv. <https://doi.org/10.1145/3548606.3559381>
- Roy, A., & Dasgupta, D. (2018). A fuzzy decision support system for multifactor authentication. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 22(12), 3959–3981. <https://doi.org/10.1007/s00500-017-2607-6>
- Solapurkar, P. (2016). Building secure healthcare services using OAuth 2.0 and JSON web token in IOT cloud scenario. *International Conferences on Contemporary Computing and Informatics*, 99–104. <https://doi.org/10.1109/ic3i.2016.7917942>
- Speiser, J. L., Miller, M., Miller, M. E., Tooze, J. A., & Ip, E. H. (2019). A Comparison of Random Forest Variable Selection Methods for Classification Prediction Modeling. *Expert Systems With Applications*, 134, 93–101. <https://doi.org/10.1016/j.eswa.2019.05.028>
- Sujudi, Hammam Mahfuzh, dan Lukman Heryawan. “An Automatic Data Mapping for Interoperability of OpenEMR Medical Practice Management Software Using the Fast Healthcare Interoperability Resources.” *Advanced Biomedical Engineering* 11 (2022): 186–93. doi:10.14326/abe.11.186.
- Taneja, M. (2013). An analytics framework to detect compromised IoT devices using mobility behavior. *Information and Communication Technology Convergence*, 38–43. <https://doi.org/10.1109/ictc.2013.6675302>
- Thomas, K., Li, F., Zand, A., Barrett, J., Ranieri, J., Invernizzi, L., Markov, Y., Comanescu, O., Eranti, V., Moscicki, A., Margolis, D., Paxson, V., & Bursztein, E. (2017). Data Breaches, Phishing, or Malware?: Understanding the Risks of Stolen Credentials. 1421–1434. <https://doi.org/10.1145/3133956.3134067>
- Wiefling, Stephan, Markus Dürmuth, & Luigi Lo Iacono. (2021). What’s in Score for Website Users: A Data-driven Long-term Study on Risk-based Authentication Characteristics. *Financial Cryptography*. [https://doi.org/10.1007/978-3-662-64331-0\\_19](https://doi.org/10.1007/978-3-662-64331-0_19)
- Wiefling, Stephan, Paul René Jørgensen, Sigurd Thunem, & Luigi Lo Iacono. (2022). Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service. *ACM Transactions on Privacy and Security*. <https://doi.org/10.1145/3546069>



Zhang, F., Kondoro, A., & Muftic, S. (2012). Location-Based Authentication and Authorization Using Smart Phones. 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 1285–1292. <https://doi.org/10.1109/trustcom.2012.198>

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer Texts in Statistics.