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* MENGGUNAKAN 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.	**Wahyono, S.Kom., Ph.D.**
Promotor	**Penguji**
	Aina Musdholifah, S.Kom., M.Kom. Ph.D
Ko-promotor	**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

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

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

Oleh

Damar Arba Pramuditya
22/501365/PPA/06386

Studi ini menggunakan pendekatan berbasis risiko untuk mengidentifikasi dan menilai potensi risiko yang terkait dengan otentikasi M2M. Ini melibatkan identifikasi pelaku ancaman potensial, kerentanan, dan dampak dari serangan yang berhasil. Studi ini juga mengevaluasi metode otentikasi M2M saat ini dan keefektifannya dalam mengurangi risiko yang teridentifikasi. Terakhir, penelitian ini merekomendasikan strategi untuk meningkatkan otentikasi M2M untuk mengurangi risiko serangan yang berhasil. Studi ini diharapkan dapat mengidentifikasi beberapa risiko yang terkait dengan otentikasi M2M, antara lain:

Akses tidak sah ke perangkat: Peretas dapat mengeksploitasi kerentanan dalam autentikasi M2M untuk mendapatkan akses tidak sah ke perangkat dan mencuri informasi sensitif. Serangan penolakan layanan: Penyerang dapat meluncurkan serangan penolakan layanan untuk mengganggu komunikasi M2M dan menyebabkan downtime sistem.

Studi ini memberikan wawasan berharga tentang risiko yang terkait dengan otentikasi M2M dan strategi untuk memitigasi risiko tersebut. Temuan penelitian ini berguna untuk organisasi yang menerapkan perangkat IoT khususnya pada sektor teknologi kesehatan.

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 M2M authentication. It involves identifying potential threat actors, vulnerabilities, and the impacts of successful attacks. The study also evaluates current M2M authentication methods and their effectiveness in reducing identified risks. Lastly, this research recommends strategies to enhance M2M authentication to mitigate successful attack risks. The study is expected to identify several risks associated with M2M authentication, including:

Unauthorized access to devices: Hackers can exploit vulnerabilities in M2M authentication to gain unauthorized access to devices and steal sensitive information. Denial of service attacks: Attackers can launch denial of service attacks to disrupt M2M communication and cause system downtime.

This study provides valuable insights into the risks associated with M2M authentication and strategies to mitigate these risks. The research findings are useful for organizations implementing IoT devices, particularly in the healthcare technology sector.

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.

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 Wiefling 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.
3. Masih menggunakan *singe factor authentication* yang rentan terhadap serangan *token replay*.

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 akan memfokuskan pada risiko yang terkait dengan otentikasi M2M pada FHIR.
2. Datasek yang digunakan dalam penelitian ini adalah data sekunder yang 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 sistem autentikasi mesin ke mesin berbasis risiko yang dapat meningkatkan keamanan sistem autentikasi 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 phising menjadi faktor utama penyebab terjadinya kebocoran kredensial dan disusul oleh keyloggers.

(Stephan Wiefling et al., 2022) mengemukakan Risk-Based Authentication (RBA) dapat memperkirakan apakah login itu sah atau merupakan upaya pengambilalihan akun. Ini dilakukan dengan memantau dan merekam sekumpulan fitur yang tersedia dalam konteks login. Fitur potensial berkisar dari jaringan (mis., alamat IP), perangkat atau klien (mis., string agen pengguna), hingga informasi biometrik perilaku (mis., waktu masuk).

Selain itu kelebihan RBA juga telah disurvey oleh (Cabarcos et al., 2019) menganalisis literatur tentang autentikasi adaptif berdasarkan prinsip-prinsip desain yang terkenal dalam disiplin sistem berbasis resiko dan tantangan nya adalah tidak ada satu ukuran yang cocok untuk semua dalam keamanan, tidak ada mekanisme baru yang akan menggantikan semua mekanisme lainnya dan diterima sebagai solusi universal. (Doerfler et al., 2019) menggambarkan bahwa tantangan login bertindak sebagai penghalang penting untuk pembajakan, tetapi gesekan dalam proses menyebabkan pengguna yang sah gagal masuk, meskipun pada akhirnya dapat

mengakses akun mereka lagi.

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

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

Penggunaan RBA masih terbatas pada major digital service, hal ini sebagian disebabkan oleh kurangnya pengetahuan dan implementasi terbuka yang memungkinkan penyedia layanan mana pun untuk meluncurkan perlindungan RBA kepada penggunanya. Untuk menutup kesenjangan ini, (Stephan Wiefling et al., 2021) memberikan analisis tentang karakteristik RBA dalam penerapan praktis sekaligus memberikan dataset yang dapat digunakan secara umum.

Penelitian lain (Misbahuddin et al., 2017) mengusulkan sistem RBA berbasis perangkat yang menggunakan jenis perangkat dan status perangkat untuk menentukan tingkat risiko suatu sesi. Penelitian tersebut menemukan bahwa sistem yang diusulkan secara efektif meningkatkan keamanan sistem dengan tetap mempertahankan kegunaan menggunakan machine learning.

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

(Roy & 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 o 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.

Tabel 2.1: Tinjauan Pustaka

Nama	**Penelitian**	**Metode**	**Hasil**
Thomas dkk (2017)	Pencurian kredensial dan menilai risiko yang ditimbulkannya bagi jutaan pengguna	Framework otomatis yang menggabungkan data Google Search dan Gmail untuk mengidentifikasi lebih dari satu miliar korban kebocoran kredensial, kit phising, dan keylogger.	Mengidentifikasi 788.000 calon korban keylogger siap pakai; 12,4 juta calon korban kit phishing; 1,9 miliar nama pengguna dan kata sandi yang terungkap melalui pelanggaran data dan diperdagangkan di forum pasar gelap.

Berlanjut di halaman selanjutnya

Table 2.1: Lanjutan Tinjauan Pustaka

Nama	**Penelitian**	**Metode**	**Hasil**
Stephan Wiefling dkk (2022)	Analisis RBA pada layanan online skala besar dunia nyata	Simple model, extended model, login dataset	RBA memblokir 99,5% penyerang naif. Simple model: targeted attackers dropped dari 0.9552 menjadi 0.5295.
Cabarcos dkk (2019)	Survey studi mengenai cara dinamis memilih mekanisme terbaik untuk mengautentikasi pengguna tergantung pada beberapa faktor	CARS-AD (Vector Space Model (VSM)), ASSO (SVM), Reinforced AuthN (Logistic Regresion)	Pengurangan overhead kata sandi (masing-masing 42% dan 47% lebih sedikit permintaan kata sandi).
Doerfler dkk (2019)	Manfaat fitur login keamanan untuk mencegah pengambilalihan akun	MFA	Memblokir lebih dari 94% upaya pembajakan.
Prasad dkk (2017)	Meningkatkan Layanan Mobile Banking menggunakan Otentikasi Berbasis Lokasi	GPS dan GPRS	GPS digunakan untuk menyediakan autentikasi lokasi, banyak informasi terkait satelit yang tidak mudah diimplementasikan.
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.

Berlanjut di halaman selanjutnya

Table 2.1: Lanjutan Tinjauan Pustaka

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

Berlanjut di halaman selanjutnya

Table 2.1: Lanjutan Tinjauan Pustaka

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

BAB III
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 moduler, 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.

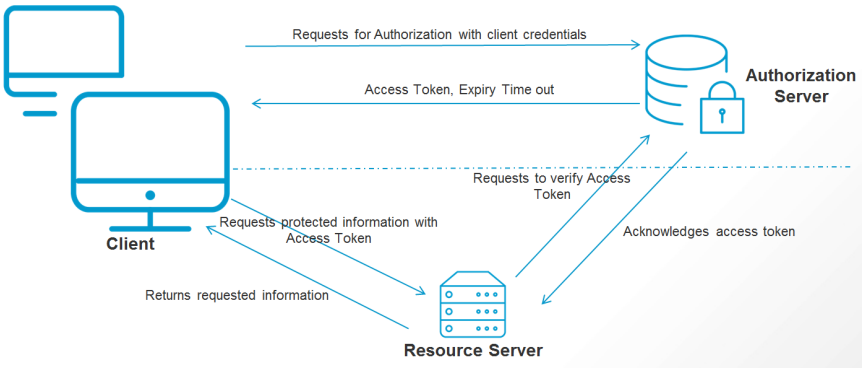
3.3.3 Autentikasi Mesin ke Mesin

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. Dalam konteks Machine-to-Machine (M2M) authentication, risk-based authentication digunakan untuk mengukur tingkat risiko dari suatu permintaan akses dan mengambil tindakan yang sesuai berdasarkan tingkat risiko tersebut. Prosesnya dapat dilakukan dengan cara menganalisis faktor-faktor

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

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

3.5 *Classification and Regression Tree (CART)*

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

Node Kiri:

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

Node Kanan:

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

Node t:

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

Keterangan:

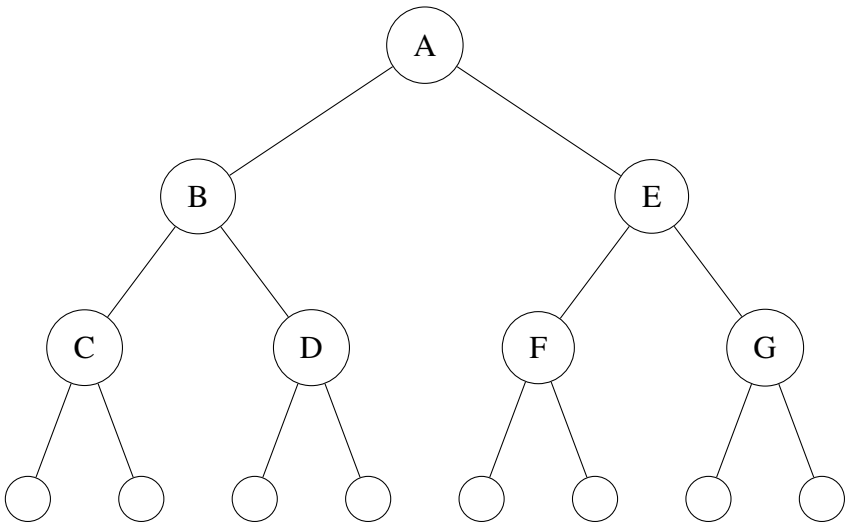
$$p_t(k) = \frac{n_t(k)}{n_t} \quad \text{dan} \quad p_t(l) = \frac{n_t(l)}{n_t} \tag{3.4}$$

$p_t(k), p_t(l)$: Proporsi objek kelas klasifikasi ke-k atau ke-l pada node t (3.5)

$n_t(k), n_t(l)$: Jumlah observasi kelas klasifikasi ke-k atau ke-l pada node t (3.6)

n_t : Jumlah seluruh observasi pada node t (3.7)

3.5.1 *Random Forest*



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

splitting tree atribut diambil berdasarkan banyaknya variabel yang terpilih melalui mtry. Selanjutnya, melakukan kembali pembentukan sampel bootstrap dan metode recursive partitioning untuk dalam membangun satu forest untuk melihat vote klasifikasi dari seluruh tree yang terbentuk.

3.5.2 Laju Galat klasifikasi

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

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

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

3.5.3 *Variable Importance Measure(VIM)*

Penggunaan analisis dalam RF secara umum sangat sulit untuk melakukan interpretasi dalam memperoleh informasi. Salah satu solusi untuk mempermudah memperoleh informasi dalam RF ialah dengan mengidentifikasi Variable Importance Measure (VIM) untuk variabel prediktor. Apabila variabel importance dapat diidentifikasi, maka hasil RF akan diperoleh metode penyeleksian variabel yang berpengaruh penting terhadap pembentukan tree dalam RF. Estimasi pemilihan variabel importance dalam random forest dapat dilakukan dengan melihat berapa banyak kenaikan prediksi error (OOB) data untuk variabel terpilih sementara yang lainnya tidak berubah Liaw & Wiener (2002). Metode representatif dari perhitungan

pengukuran variabel importance adalah Mean Decrease Impurity (MDI) atau disebut juga dengan Mean Decrease Gini (MDG) yang diusulkan oleh Breiman pada tahun 2001. Suatu p peubah penjelas dengan h=(1,2,...,p) maka rumus mengukur tingkat kepentingan peubah penjelas Xh dengan cara berikut (Xiao Li. dkk, 20 19).

$$\text{MDG}(\mathbf{x}_h) = \frac{1}{k}\sum_{t=1}^{k}\text{MDG}(\mathbf{X}_h, \mathbf{x}_t) \tag{3.9}$$

Keterangan:

$$\text{MDG}(\mathbf{X}_h, \mathbf{x}_t) = \sum_{t\in(T),v(t)=h}\frac{N_n(t)}{n}\Delta\mathbf{x}(t) \tag{3.10}$$

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

$$\text{MDA}(\mathbf{x}_h) = \frac{1}{k}\sum_{t=1}^{k}\sum_{i\in\text{OOB}(t)}\frac{I(y_i = \hat{y}_i(t)) - I(y_i = \hat{y}_i, h(t))}{|\text{OOB}(t)|} \tag{3.11}$$

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

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.

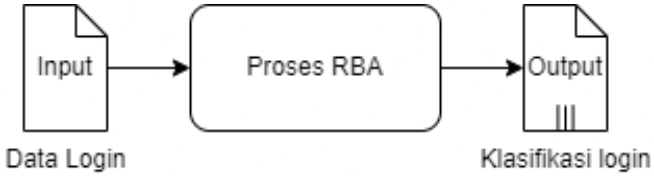
Skor F1 sedikit kurang intuitif karena menggabungkan presisi dan recall ke dalam satu metrik. Jika presisi dan recall keduanya tinggi, F1 juga akan tinggi. Jika keduanya rendah, F1 akan rendah. Jika salah satunya tinggi dan yang lainnya rendah, F1 akan rendah. F1 adalah cara cepat untuk mengetahui apakah 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. Prosessor 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

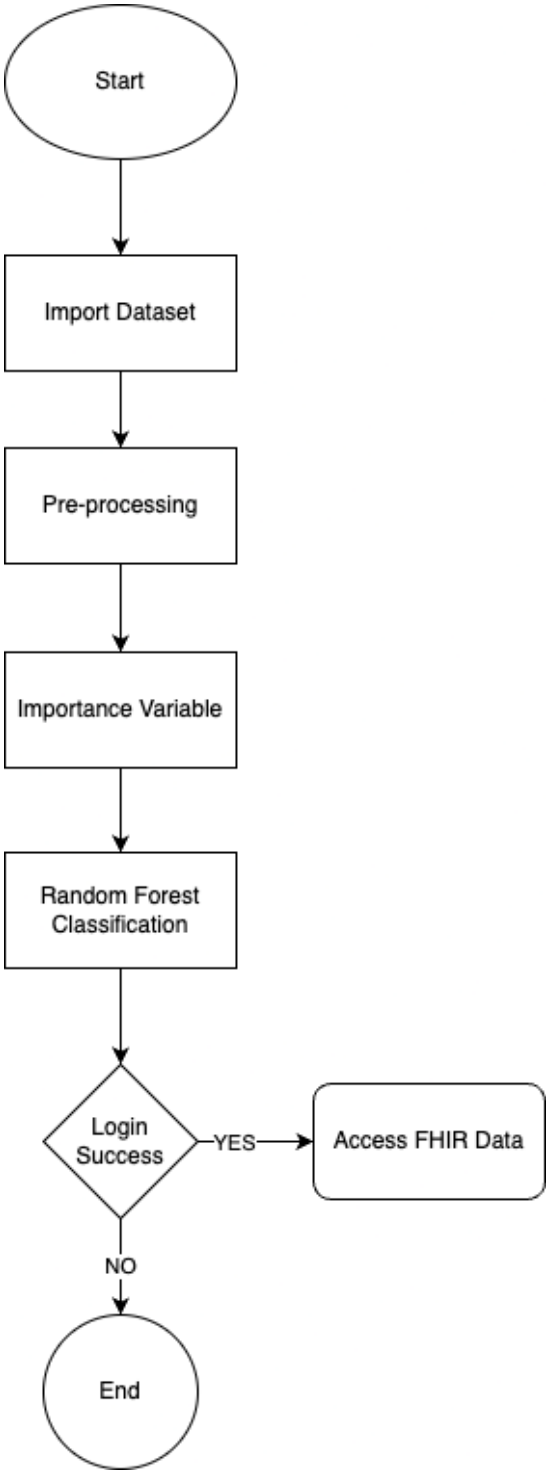
1. Bahasa pemrograman python versi 3.9 dengan *framework* anaconda 3
2. *Framework* flask untuk membuat endpoint
3. Sistem operasi dengan base unix untuk menjalankan sistem klasifikasi

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

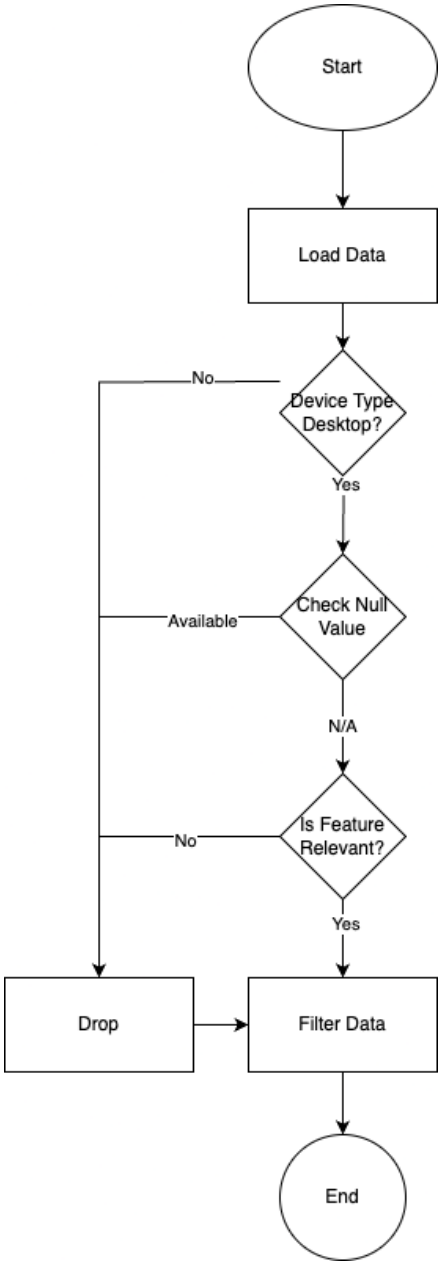
Pada Gambar 4.2, sistem ini akan mendapatkan data login dari datasest. Data

login ini kemudian akan digunakan sebagai input untuk melakukan analisis risiko autentikasi. Dalam menentukan risiko autentikasi, sistem ini akan menggunakan metode Random Forest. Metode Random Forest akan menghasilkan variabel kepentingan yang dapat digunakan untuk melakukan analisis risiko autentikasi.

Setelah itu, sistem ini akan terintegrasi dengan sistem FHIR. Sistem ini akan menggunakan FHIR API untuk mengakses data dari sistem FHIR. FHIR API akan mengakses data dari sistem FHIR dengan menggunakan *request* dan *response*.

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 dilakukkan 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 dilakukkan 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. 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

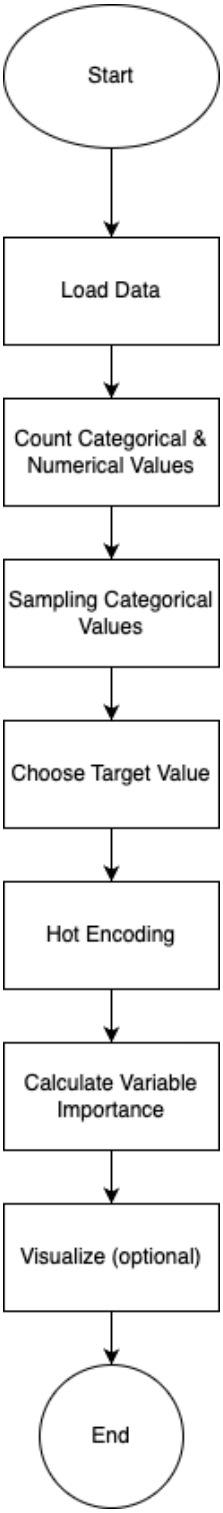
login_timestamp_1	login_timestamp_2	ip_address_1	ip_address_2	country_1	country_2	region_1	region_2	city_1
0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1

Cara kerja one-hot encoding adalah seperti ditunjukan 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.

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.



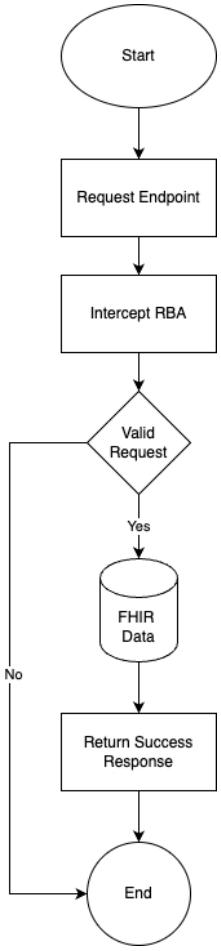
Gambar 4.4: Rancangan Variabel Kepentingan

Gambar 4.4 menjelaskan bahwa variabel kepentingan akan digunakan untuk

melakukan analisis risiko autentikasi. Variabel kepentingan ini akan digunakan sebagai input untuk melakukan analisis risiko autentikasi. Setelah data berhasil di-import akan diakukan penghitungan kategorikal dan numerikal data.

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

Untuk melakukan integrasi dengan sistem FHIR, sistem ini akan menggunakan FHIR API. FHIR API adalah sebuah API yang digunakan untuk

mengakses data dari sistem FHIR. FHIR API akan mengakses data dari sistem FHIR dengan menggunakan *request* dan *response*.

4.4 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.

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:

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

Browser Name and Version	String	Browser name and version derived from the user agent string	Chrome 70.0.3538
Device Type	String	Device type derived from the user agent string	(‘mobile‘, ‘desktop‘, ‘tablet‘, ‘bot‘, ‘unknown‘)
User ID	Integer	Identification number related to the affected user account	Random pseudonym
Login Timestamp	Integer	Timestamp related to the login attempt	64 Bit timestamp
Round-Trip Time (RTT) [ms]	Integer	Server-side measured latency between client and server	1 - 8600000
Login Successful	Boolean	‘True‘: Login was successful, ‘False‘: Login failed	(‘true‘, ‘false‘)
Is Attack IP	Boolean	IP address was found in known attacker data set	(‘true‘, ‘false‘)
Is Account Takeover	Boolean	Login attempt was identified as account takeover by incident response team of the online service	(‘true‘, ‘false‘)

Tabel 5.1: Deskripsi tabel fitur login

5.2 *Preprocessing Data*

Penggunaan dataset dalam penelitian ini membutuhkan beberapa tahapan persi- apan 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.

```
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.

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

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

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

value_counts_3 = df['is_account_takeover'].value_counts(normalize=True)
is_account_takeover_true = value_counts_3[True] * 100
is_account_takeover_false = value_counts_3[False] * 100
print("is_account_takeover")
print(f"Percentage of True values: {is_account_takeover_true:.2f}%")
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.

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

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

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

Berikut adalah contoh token yang digenerate.

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.

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

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.

```
# check lenght in column user_agent_string
features['length'] = features['user_agent_string'].apply(
    lambda row: min(len(row), len(row)) if isinstance(row, str) else None
)
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.

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

Setelah itu dilakukan penyaringan data dengan cara menghapus data yang

memiliki device type selain ’desktop’.

5.3.3 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.

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

Hasil keluaran dari tahap ini adalah sebagai berikut.

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. Pemilih- an 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.

```
categorical = [var for var in df.columns if df[var].dtype=='O']
print('There are {} categorical variables\n'.format(len(categorical)))
print('The categorical variables are :\n\n', categorical)

There are 8 categorical variables

The categorical variables are :
['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.

```
import category_encoders as ce

# One-hot encode the categorical features
# encode categorical variables with ordinal encoding
# see def preprocess_data(df) above
encoder = ce.OneHotEncoder(cols= ['login_timestamp', 'ip_address', 'country',
    'region', 'city', 'user_agent_string', 'browser', 'os_detail'])
X_train = encoder.fit_transform(X_train)

X_test = encoder.transform(X_test)
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.

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

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

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

# Top 10 important features
feature_scores.head(10)
```

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

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

5.5 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.

```
    # Make predictions on the test set
    y_pred = clf.predict(X_test)

    # Evaluate the accuracy of the model
    accuracy = accuracy_score(y_test, y_pred)
    print('Accuracy:', accuracy)

    # Calculate precision, recall, and F1 score
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    print('Precision:', precision)
    print('Recall:', recall)
    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.

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.

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, dilakukkan pengujian untuk beberapa ukuran dataset. Ukuran dataset yang digunakan adalah 10000, 20000, 30000, 40000 dan 50000. Berikut adalah hasil pengujian waktu data training.

Tabel 6.1: Hasil Pengujian Waktu Data Training

Ukuran Dataset	**Waktu Data Training (detik)**
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 x 5 x 5 x 5 = 625 percobaan.

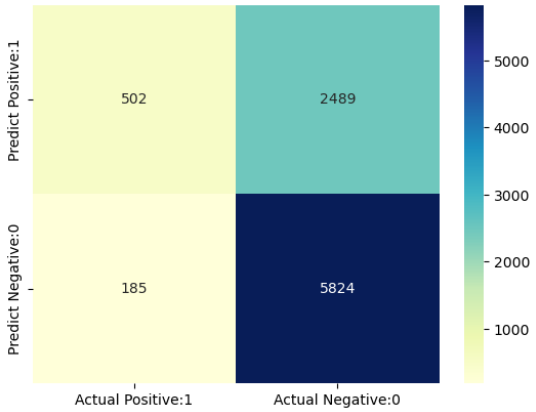
Tabel 6.3: Parameter Grid

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: Heuristic Authentication memiliki beberapa keunggulan dalam konteks autentikasi mesin ke mesin. Pertama, pendekatan ini sangat sederhana dan mudah diimplementasikan karena mengandalkan aturan-aturan sederhana yang telah ditentukan sebelumnya. Selain itu, Heuristic Authentication

tidak memerlukan sumber daya komputasi yang besar, sehingga cocok untuk lingkungan dengan keterbatasan sumber daya.

Analisis Kualitatif: Namun, ada beberapa batasan yang perlu dipertimbangkan dalam menggunakan Heuristic Authentication. Pertama, pendekatan ini mungkin kurang cocok untuk kasus-kasus autentikasi yang kompleks atau data yang tidak terstruktur dengan baik. Selain itu, aturan-aturan heuristik mungkin tidak cukup fleksibel untuk menangani variasi perilaku pengguna yang kompleks.

Feth dkk (2019) Secara kualitatif, para pengembang merespons positif terhadap pendekatan yang digunakan dalam penelitian ini. Mereka menyatakan bahwa mereka menemukan masalah baru atau dorongan pemikiran yang dapat membantu dalam pengembangan lebih lanjut. Namun, terdapat perbedaan pendapat mengenai tingkat detail dari model yang digunakan. Beberapa partisipan merasa bahwa model terlalu rinci untuk sebagian besar heuristik.

Kelebihan: Salah satu kelebihan utama Heuristic Authentication adalah kesederhanaan implementasinya. Pendekatan ini mudah diimplementasikan tanpa memerlukan analisis yang rumit atau data pelatihan yang besar. Selain itu, Heuristic Authentication tidak memerlukan sumber daya komputasi yang besar. Serta mayoritas partisipan pada penelitian yang dilakukkan Feth dkk (2019) menyatakan bahwa mereka menemukan model yang digunakan membantu dalam evaluasi usability langkah-langkah keamanan dalam perangkat lunak mereka

Kelemahan: Namun, kekurangan utama Heuristic Authentication adalah kurangnya fleksibilitas. Pendekatan ini tidak dapat menangani kasus-kasus autentikasi yang kompleks dengan baik dan mungkin kurang akurat dalam mengidentifikasi aktivitas yang mencurigakan atau autentikasi yang tidak sah dibandingkan dengan metode yang lebih canggih.

Selain itu, Feth dkk (2019) mengemukakan terdapat perbedaan pendapat di antara partisipan mengenai tingkat detail dari model yang digunakan, menunjukkan bahwa model tersebut mungkin perlu disesuaikan lebih lanjut untuk mencapai kesesuaian yang ideal.

Dengan mempertimbangkan kedua metode ini, organisasi harus memilih berdasarkan pada kebutuhan spesifik, sumber daya yang tersedia, dan tingkat keamanan yang diinginkan. Dalam beberapa kasus, kombinasi dari kedua metode ini juga dapat digunakan untuk meningkatkan keamanan autentikasi.

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 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 mengklasikasi risiko autentikasi. Dengan akurasi 70.8%, presisi 70.1%, *recall* 96.8%, dan *F1-score* 71.3%. Ketimpangan akurasi dan *recall* disebabkan oleh ketidakseimbangan jumlah data pada kelas yang berbeda.
2. Pembatasan fitur kepentingan dapat berpengaruh pada akurasi sistem.

7.2 Saran

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

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

DAFTAR PUSTAKA

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*

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*

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*