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

SKRIPSI

Disusun oleh:

«AUTHOR»
«NIM»

PROGRAM STUDI «NAMA PRODI»
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA
«TAHUN PENDADARAN»

HALAMAN PENGESAHAN

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

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Departemen Teknik Elektro dan Teknologi Informasi
Fakultas Teknik
Universitas Gadjah Mada

Disusun oleh:

«AUTHOR»
«NIM»

Telah disetujui dan disahkan

Pada tanggal . . . . . .

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

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

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

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

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

Yogyakarta, tanggal-bulan-tahun

Materai Rp10.000

(Tanda tangan)

Nama Mahasiswa
NIM

HALAMAN PERSEMBAHAN

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

[contoh]

KATA PENGANTAR

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

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

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

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SINGKATAN

[SAMPLE]

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

INTISARI

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

Kata kunci : RBA, Autentikasi, M2M, Random Forest

ABSTRACT

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

Keywords : RBA, Authentication, M2M, Random Forest

BAB I
PENDAHULUAN

1.1 Latar Belakang

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

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

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

1.2 Rumusan Masalah

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

1.3 Batasan Masalah

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

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

negger dkk. (2016).

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

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

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

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah diharapkan sebagai berikut:

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

BAB II
TINJAUAN PUSTAKA

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

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

Studi yang dilakukan oleh Thomas dkk. (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 dkk. (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 dkk. (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 dkk. (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 dkk. (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 dkk. (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 dkk. (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 dkk. (2021) memberikan analisis tentang karakteristik RBA dalam penerapan praktis sekaligus memberikan dataset yang dapat digunakan secara umum. Penelitian lain Misbahuddin dkk. (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 dan 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, F dkk. (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 dan Vuong (2013), Speiser dkk. (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.

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

BAB III
LANDASAN TEORI

3.1 FHIR (Fast Healthcare Interoperability Resources)

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

FHIR menyediakan kumpulan resource yang dapat digunakan untuk pertukaran data kesehatan, seperti informasi pasien, informasi medis, dan informasi billing. Resource ini dapat ditransmisikan dalam format yang berbeda, seperti JSON atau XML. FHIR juga menyediakan API (Application Programming Interface) yang dapat digunakan untuk mengakses data dan layanan yang tersedia melalui jaringan. FHIR dapat digunakan untuk meningkatkan interoperabilitas sistem kesehatan dan mengijinkan data kesehatan untuk ditransmisikan dengan cepat dan aman antara sistem yang berbeda. Standar ini juga memudahkan pengembangan aplikasi yang dapat mengakses data kesehatan dari berbagai sumber dan digunakan dalam berbagai konteks, seperti telemedicine, pengelolaan kesehatan, dan analisis data kesehatan.

3.2 Machine-to-Machine (M2M) Authentication

Machine-to-Machine (M2M) authentication adalah proses verifikasi yang digunakan untuk mengautentikasi perangkat atau mesin yang terhubung ke jaringan, seperti komputer, perangkat IoT, atau perangkat mobile. Proses ini memastikan bahwa hanya perangkat yang sah yang dapat terhubung ke jaringan dan mengakses data atau layanan yang tersedia seperti skema pada Gambar 3.1.

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.

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

3.3 Klien Kredensial

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

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.

BAB IV
ANALISIS DAN RANCANGAN SISTEM

4.1 Analisis Sistem

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

4.1.1 Gambaran Umum Sistem

Gambar 4.2 menjelaskan secara umum system bekerja dengan menggunakan metadata login sebagai input untuk mengidentifikasi risiko dari suatu transaksi. Metadata login ini kemudian diolah dan dianalisis menggunakan metode Random Forest untuk menghasilkan prediksi risiko autentikasi dengan output nya adalah klasifikasi.

4.1.2 Analisis Kebutuhan Sistem

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

4.1.2.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem ini adalah sebagai berikut:

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

4.1.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sistem ini adalah sebagai berikut:

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

4.2 Rancangan Sistem

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

4.2.1 Rancangan Arsitektur Sistem

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

4.2.2 Rancangan Pembersihan Data

Rancangan pembersihan data dapat dilihat pada Gambar 4.4. 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 :

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

4.2.3 Rancangan Variabel Kepentingan

Rancangan variabel kepentingan akan dilakukan dengan menggunakan metode Random Forest. Metode Random Forest akan menghasilkan variabel kepentingan yang dapat dilihat pada Gambar 4.5. Variabel kepentingan ini akan digunakan untuk melakukan analisis risiko autentikasi.

4.2.4 Rancangan Integrasi Dengan Sistem FHIR

Rancangan integrasi dengan sistem FHIR dapat dilihat pada Gambar 4.6. Sistem ini akan terintegrasi dengan sistem FHIR untuk mendapatkan data login dari pasien. Data login ini kemudian akan digunakan sebagai input untuk melakukan analisis risiko autentikasi. Untuk melakukan integrasi dengan sistem FHIR, sistem ini akan menggunakan FHIR API. FHIR API adalah sebuah API yang digunakan untuk mengakses data dari sistem FHIR. FHIR API akan mengakses data dari sistem FHIR dengan menggunakan *request* dan *response*.

4.3 Rancangan Pengujian

Pengujian sistem ini akan dilakukan dengan menggunakan data login dari dataset. Data login ini akan digunakan sebagai input untuk melakukan analisis risiko autentikasi. Pengujian sistem ini akan dilakukan dengan menggunakan metode *10-fold cross validation*. Metode *10-fold cross validation* akan membagi data menjadi 10 bagian. 9 bagian

data akan digunakan sebagai data latih dan 1 bagian data akan digunakan sebagai data uji. Pengujian ini akan dilakukan sebanyak 10 kali dengan menggunakan data yang berbeda-beda. Pengujian ini akan menghasilkan nilai akurasi, presisi, *recall*, dan *f-measure*.

BAB V
IMPLEMENTASI

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

5.1 Pengumpulan Data

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

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

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

Tabel 5.1. Deskripsi tabel fitur login

5.2 Persiapan Data

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

5.2.1 Eksplorasi Data

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

5.2.1.1 Sampling Data

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

```
import pandas as pd

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

```

5.2.2 Pemilihan Target

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

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

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

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

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

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

Berikut adalah hasil sampling data.

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

Tabel 5.2. Hasil Sampling Data

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

5.2.3 Pengecekan *Missing Value*

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

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

hasilnya adalah sebagai berikut:

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

Tabel 5.3. Missing Values in Each Feature

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

5.2.4 Penambahan Kolom Token

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

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

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

```
        return sha512_hash.hexdigest()

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

```

Berikut adalah contoh token yang digenerate.

User ID	Token
-3065936140549856249	4ffe29f1960c24624ec2c36909f3b39cb8d59fa18515f4
5932501938287412564	ecee6cc95d3b047c8f796b8e772a468124b7ddb599a7a3

Tabel 5.4. Contoh Token

5.2.5 Pembersihan Data

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

5.2.5.1 Penamaan Kolom

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

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

```

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

Tabel 5.5. Column Renaming in DataFrame

5.2.5.2 Penyaringan User Agent dan Device Type

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

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

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

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

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

```

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

5.2.6 Menghapus Kolom yang Tidak Diperlukan

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

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

```

Hasil keluaran dari tahap ini adalah sebagai berikut.

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

Tabel 5.6. Revised Initial Exploratory Data Analysis

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

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

5.3 Implementasi Pemilihan Fitur

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

5.3.0.1 Eksplorasi Tipe Data

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

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

Berikut adalah hasil keluaran dari tahap ini.

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

Tabel 5.7. Data Type of Each Column

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

5.3.1 Encoding

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

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

Berikut adalah hasil keluaran dari tahap ini.

5.3.2 Gini Importance

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

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

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

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

# Top 10 important features
feature_scores.head(10)

```

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

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

Tabel 5.8. Gini Importance of Each Feature

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

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

5.4 Pembuatan Random Forest

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

5.4.1 Pembagian Data

Pada tahap ini dilakukan pembagian data meliputi

5.4.1.1 Pembagian Data Fitur dan Target

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

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

```

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

5.4.1.2 Pembagian Data Training dan Data Testing

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

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

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

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

5.4.2 Pembuatan Model dan Pelatihan Model

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

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

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

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

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

5.4.3 Evaluasi Model

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

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

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

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

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

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

5.4.4 Visualisasi Model

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

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

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

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

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

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

5.5 Pembagunan Sistem

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

5.5.1 Pembangunan API

BAB VI
HASIL DAN PEMBAHASAN

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

6.1 Hasil Pengujian

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

6.2 Analisis Hasil Pengujian

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

6.3 Pembahasan Hasil Pengujian

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

BAB VII
KESIMPULAN DAN SARAN

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

7.1 Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

7.2 Saran

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

DAFTAR PUSTAKA

[1] L. E. Nugroho, "E-book as a platform for exploratory learning interactions," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 11, no. 01, pp. 62–65, 2016. [Online]. Available: http://www.online-journals.org/index.php/i-jet/article/view/5011

[2] P. I. Santosa, "User?s preference of web page length," *International Journal of Research and Reviews in Computer Science*, pp. 180–185, 2011.

[3] N. A. Setiawan, "Fuzzy decision support system for coronary artery disease diagnosis based on rough set theory," *International Journal of Rough Sets and Data Analysis (IJRSDA)*, vol. 1, no. 1, pp. 65–80, 2014.

[4] C. P. Wibowo, P. Thumwarin, and T. Matsuura, "On-line signature verification based on forward and backward variances of signature," in *Information and Communication Technology, Electronic and Electrical Engineering (JICTEE), 2014 4th Joint International Conference on*. IEEE, 2014, pp. 1–5.

[5] D. A. Marenda, A. Nasikun, and C. P. Wibowo, "Digitory, a smart way of learning islamic history in digital era," *arXiv preprint arXiv:1607.07790*, 2016.

[6] S. Wibirama, S. Tungjitkusolmun, and C. Pintavirooj, "Dual-camera acquisition for accurate measurement of three-dimensional eye movements," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 8, no. 3, pp. 238–246, 2013.

[7] C. P. Wibowo, "Clustering seasonal performances of soccer teams based on situational score line," *Communications in Science and Technology*, vol. 1, no. 1, 2016.

Catatan: Daftar pustaka adalah apa yang dirujuk atau disitasi, bukan apa yang telah dibaca, jika tidak ada dalam sitasi maka tidak perlu dituliskan dalam daftar pustaka.

LAMPIRAN

L.1 Isi Lampiran

Lampiran bersifat opsional bergantung hasil kesepakatan dengan pembimbing dapat berupa:

1. Bukti pelaksanaan Kuesioner seperti pertanyaan kuesioner, resume jawaban responden, dan dokumentasi kuesioner.
2. Spesifikasi Aplikasi atau Sistem yang dikembangkan meliputi spesifikasi teknis aplikasi, tautan unduh aplikasi, manual penggunaan aplikasi, hingga screenshot aplikasi.
3. Cuplikan kode yang sekiranya penting dan ditambahkan.
4. Tabel yang terlalu panjang yang masih diperlukan tetapi tidak memungkinkan untuk ditayangkan di bagian utama skripsi.
5. Gambar-gambar pendukung yang tidak terlalu penting untuk ditampilkan di bagian utama. Akan tetapi, mendukung argumentasi/pengamatan/analisis.
6. Penurunan rumus-rumus atau pembuktian suatu teorema yang terlalu panjang dan terlalu teknis sehingga Anda berasumsi bahwa pembaca biasa tidak akan menelaah lebih lanjut. Hal ini digunakan untuk memberikan kesempatan bagi pembaca tingkat lanjut untuk melihat proses penurunan rumus-rumus ini.

LAMPIRAN

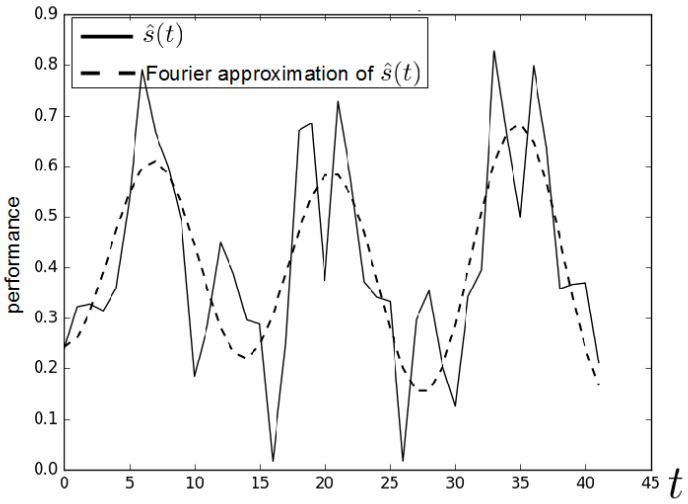
L.2 Panduan Latex

L.2.1 Syntax Dasar

L.2.1.1 Penggunaan Sitasi

Contoh penggunaan sitasi [1, 2] [3] [4] [5] [6, 7]

L.2.1.2 Penulisan Gambar



Gambar 1. Contoh gambar.

Contoh gambar terlihat pada Gambar 1. Gambar diambil dari [7].

L.2.1.3 Penulisan Tabel

Tabel 1. Tabel ini

ID	Tinggi Badan (cm)	Berat Badan (kg)
A23	173	62
A25	185	78
A10	162	70

Contoh penulisan tabel bisa dilihat pada Tabel 1.

L.2.1.4 Penulisan formula

Contoh penulisan formula

$$L_{\psi_z} = \{t_i \mid v_z(t_i) \leq \psi_z\} \tag{1}$$

Contoh penulisan secara *inline*: $PV = nRT$. Untuk kasus-kasus tertentu, kita membutuhkan perintah "mathit" dalam penulisan formula untuk menghindari adanya jeda saat penulisan formula.

Contoh formula **tanpa** menggunakan "mathit": $PVA = RTD$

Contoh formula **dengan** menggunakan "mathit": $PVA = RTD$

L.2.1.5 Contoh list

Berikut contoh penggunaan list

1. First item
2. Second item
3. Third item

L.2.2 Blok Beda Halaman

L.2.2.1 Membuat algoritma terpisah

Untuk membuat algoritma terpisah seperti pada contoh berikut, kita dapat memanfaatkan perintah *algstore* dan *algrestore* yang terdapat pada paket *algcompatible*. Pada dasarnya, kita membuat dua blok algoritma dimana blok pertama kita simpan menggunakan *algstore* dan kemudian di-restore menggunakan *algrestore* pada algoritma kedua. Perintah tersebut dimaksudkan agar terdapat kesinamungan antara kedua blok yang sejatinya adalah satu blok.

Algorithm 1 Contoh algorima

```
1: procedure CREATESET(v)
2:     Create new set containing v
3: end procedure
```

Pada blok algoritma kedua, tidak perlu ditambahkan caption dan label, karena sudah menjadi satu bagian dalam blok pertama. Pembagian algoritma menjadi dua bagian ini berguna jika kita ingin menjelaskan bagian-bagian dari sebuah algoritma, maupun untuk memisah algoritma panjang dalam beberapa halaman.

```
4: procedure CONCATSET(v)
5:     Create new set containing v
6: end procedure
```

L.2.2.2 Membuat tabel terpisah

Untuk membuat tabel panjang yang melebihi satu halaman, kita dapat mengganti kombinasi *table* + *tabular* menjadi *longtable* dengan contoh sebagai berikut.

Tabel 2. Contoh tabel panjang

header 1	header 2
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar

L.2.2.3 Menulis formula terpisah halaman

Terkadang kita butuh untuk menuliskan rangkaian formula dalam jumlah besar sehingga melewati batas satu halaman. Solusi yang digunakan bisa saja dengan memindahkan satu blok formula tersebut pada halaman yang baru atau memisah rangkaian formula menjadi dua bagian untuk masing-masing halaman. Cara yang pertama mungkin akan menghasilkan alur yang berbeda karena ruang kosong pada halaman pertama akan diisi oleh teks selanjutnya. Sehingga di sini kita dapat memanfaatkan *align* yang sudah diatur dengan mode *allowdisplaybreaks*. Penggunakan *align* ini memungkinkan satu rangkaian formula terpisah berbeda halaman.

Contoh sederhana dapat digambarkan sebagai berikut.

$$\begin{aligned} x &= y^2 \\ x &= y^3 \\ a + b &= c \\ x &= y - 2 \\ a + b &= d + e \\ x^2 + 3 &= y \\ a(x) &= 2x \end{aligned} \tag{2}$$

$$
\begin{aligned}
b_i &= 5x \\
10x^2 &= 9x \\
2x^2 + 3x + 2 &= 0 \\
5x - 2 &= 0 \\
d &= \log x \\
y &= \sin x
\end{aligned}
$$

LAMPIRAN

L.3 Format Penulisan Referensi

Penulisan referensi mengikuti aturan standar yang sudah ditentukan. Untuk internasionalisasi DTETI, maka penulisan referensi akan mengikuti standar yang ditetapkan oleh IEEE (*International Electronics and Electrical Engineers*). Aturan penulisan ini bisa diunduh di http://www.ieee.org/documents/ieeecitationref.pdf. Gunakan Mendeley sebagai *reference manager* dan *export* data ke format Bibtex untuk digunakan di Latex.

Berikut ini adalah sampel penulisan dalam format IEEE:

L.3.1 Book

Basic Format:

[1] J. K. Author, "Title of chapter in the book," in Title of His Published Book, xth ed. City of Publisher, Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.

Examples:

[1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.

[2] L. Stein, "Random patterns," in Computers and You, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.

[3] R. L. Myer, "Parametric oscillators and nonlinear materials," in Nonlinear Optics, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.

[4] M. Abramowitz and I. A. Stegun, Eds., Handbook of Mathematical Functions (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.

[5] E. F. Moore, "Gedanken-experiments on sequential machines," in Automata Studies (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.

[6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), Integrated Electronic Systems. Englewood Cliffs, NJ: Prentice-Hall, 1970.

[7] M. Gorkii, "Optimal design," Dokl. Akad. Nauk SSSR, vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., The Mathematical Theory of Optimal Processes. New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).

[8] G. O. Young, "Synthetic structure of industrial plastics," in Plastics, vol. 3,

Polymers of Hexadromicon, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64.

L.3.2 Handbook

Basic Format:

[1] Name of Manual/Handbook, x ed., Abbrev. Name of Co., City of Co., Abbrev. State, year, pp. xx-xx.

Examples:

[1] Transmission Systems for Communications, 3rd ed., Western Electric Co., Winston Salem, NC, 1985, pp. 44-60.

[2] Motorola Semiconductor Data Manual, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

[3] RCA Receiving Tube Manual, Radio Corp. of America, Electronic Components and Devices, Harrison, NJ, Tech. Ser. RC-23, 1992.

Conference/Prosiding

Basic Format:

[1] J. K. Author, "Title of paper," in Unabbreviated Name of Conf., City of Conf., Abbrev. State (if given), year, pp.xxx-xxx.

Examples:

[1] J. K. Author [two authors: J. K. Author and A. N. Writer] [three or more authors: J. K. Author et al.], "Title of Article," in [Title of Conf. Record as], [copyright year] © [IEEE or applicable copyright holder of the Conference Record]. doi: [DOI number]

Sumber Online/Internet

Basic Format:

[1] J. K. Author. (year, month day). Title (edition) [Type of medium]. Available: http://www.(URL)

Examples:

[1] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com

Skripsi, Tesis dan Disertasi

Basic Format:

[1] J. K. Author, "Title of thesis," M.S. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

[2] J. K. Author, “Title of dissertation,” Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

Examples:

[1] J. O. Williams, “Narrow-band analyzer,” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993. [2] N. Kawasaki, “Parametric study of thermal and chemical nonequilibrium nozzle flow,” M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993

LAMPIRAN

L.4 Contoh Source Code

L.4.1 Sample algorithm

Algorithm 2 Kruskal's Algorithm

```
1:  procedure MAKESET(v)
2:      Create new set containing v
3:  end procedure
4:
5:  function FINDSET(v)
6:      return a set containing v
7:  end function
8:
9:  procedure UNION(u,v)
10:     Unites the set that contain u and v into a new set
11: end procedure
12:
13: function KRUSKAL(V, E, w)
14:     A ← {}
15:     for each vertex v in V do
16:         MakeSet(v)
17:     end for
18:     Arrange E in increasing costs, ordered by w
19:     for each (u,v) taken from the sorted list do
20:         if FindSet(u) ≠ FindSet(v) then
21:             A ← A ∪ {(u, v)}
22:             Union(u, v)
23:         end if
24:     end for
25:     return A
26: end function
```

L.4.2 Sample Python code

```
import numpy as np

def incmatrix(genl1,genl2):
  m = len(genl1)
  n = len(genl2)
  M = None #to become the incidence matrix
  VT = np.zeros((n*m,1), int)  #dummy variable

  #compute the bitwise xor matrix
  M1 = bitxormatrix(genl1)
  M2 = np.triu(bitxormatrix(genl2),1)

  for i in range(m-1):
    for j in range(i+1, m):
      [r,c] = np.where(M2 == M1[i,j])
      for k in range(len(r)):
        VT[(i)*n + r[k]] = 1;
        VT[(i)*n + c[k]] = 1;
        VT[(j)*n + r[k]] = 1;
        VT[(j)*n + c[k]] = 1;

  if M is None:
    M = np.copy(VT)
  else:
    M = np.concatenate((M, VT), 1)

  VT = np.zeros((n*m,1), int)

  return M
```

L.4.3 Sample Matlab code

```
function X = BitXorMatrix(A,B)
%function to compute the sum without charge of two vectors

  %convert elements into usigned integers
  A = uint8(A);
  B = uint8(B);

  m1 = length(A);
  m2 = length(B);
  X = uint8(zeros(m1, m2));
  for n1=1:m1
    for n2=1:m2
      X(n1, n2) = bitxor(A(n1), B(n2));
    end
  end
```