TESIS

ANALISIS PERBANDINGAN ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM DAN LOCAL OUTLIER FACTOR DALAM MENDETEKSI STATUS ANOMALI PADA GEMPA BUMI MENGGUNAKAN OUTLIER DAN NOVELTY

Disusun oleh:

Nama : **Nasir Usman**

NIM : **20.55.1350**

Konsentrasi : **Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA

PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA

YOGYAKARTA

2023

TESIS

ANALISIS PERBANDINGAN ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM DAN LOCAL OUTLIER FACTOR DALAM MENDETEKSI STATUS ANOMALI PADA GEMPA BUMI MENGGUNAKAN OUTLIER DAN NOVELTY

A COMPARATIVE ANALYSIS OF ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM, AND LOCAL OUTLIER FACTOR IN DETECTING THE ANOMALY STATUS OF EARTHQUAKES USING OUTLIER AND NOVELTY

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister

Disusun oleh:

Nama	**: Nasir Usman**
NIM	**: 20.55.1350**
Konsentrasi	**: Business Intelligence**

PROGRAM STUDI S2 TEKNIK INFORMATIKA
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA
YOGYAKARTA
2023

HALAMAN PENGESAHAN

ANALISIS PERBANDINGAN ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM DAN LOCAL OUTLIER FACTOR DALAM MENDETEKSI STATUS ANOMALI PADA GEMPA BUMI MENGGUNAKAN OUTLIER DAN NOVELTY

A COMPARATIVE ANALYSIS OF ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM, AND LOCAL OUTLIER FACTOR IN DETECTING THE ANOMALY STATUS OF EARTHQUAKES USING OUTLIER AND NOVELTY

Dipersiapkan dan Disusun oleh

Nasir Usman

20.55.1350

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari **Senin, 3 April 2023**

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 April 2023
Rektor

Prof. Dr. M. Suyanto, M.M.
NIK. 190302001

HALAMAN PERSETUJUAN

ANALISIS PERBANDINGAN ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM DAN LOCAL OUTLIER FACTOR DALAM MENDETEKSI STATUS ANOMALI PADA GEMPA BUMI MENGGUNAKAN OUTLIER DAN NOVELTY

A COMPARATIVE ANALYSIS OF ELLIPTIC ENVELOPE, ISOLATION FOREST, ONE-CLASS SVM, AND LOCAL OUTLIER FACTOR IN DETECTING THE ANOMALY STATUS OF EARTHQUAKES USING OUTLIER AND NOVELTY

Dipersiapkan dan Disusun oleh

Nasir Usman
20.55.1350

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis
Program Studi S2 Teknik Informatika
Program Pascasarjana Universitas AMIKOM Yogyakarta
pada hari **Senin, 3 April 2023**

Pembimbing Utama

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Anggota Tim Penguji

Dr. Andi Sunyoto, M.Kom.
NIK. 190302052

Pembimbing Pendamping

Dhani Ariatmanto, M.Kom., Ph.D.
NIK. 190302197

Anggit Dwi Hartanto, M.Kom.
NIK. 190302163

Prof. Dr. Ema Utami, S.Si., M.Kom.
NIK. 190302037

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer

Yogyakarta, 3 April 2023
Direktur Program Pascasarjana

Prof. Dr. Kusrini, M.Kom.
NIK. 1903021

HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : **Nasir U.**

NIM : **20.55.1350**

Konsentrasi : **Business Intelligence**

Menyatakan bahwa Tesis dengan judul berikut:

Analisis Perbandingan Elliptic Envelope, Isolation Forest, One-Class SVM dan Local Outlier Factor dalam Mendeteksi Status Anomali pada Gempa Bumi menggunakan Outlier dan Novelty

Dosen Pembimbing Utama : Prof. Dr. Ema Utami, S.Si., M.Kom.

Dosen Pembimbing Pendamping : Anggit Dwi Hartanto, M.Kom.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam karya tulis ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 3 April 2023

Yang Menyatakan

Nasir U.

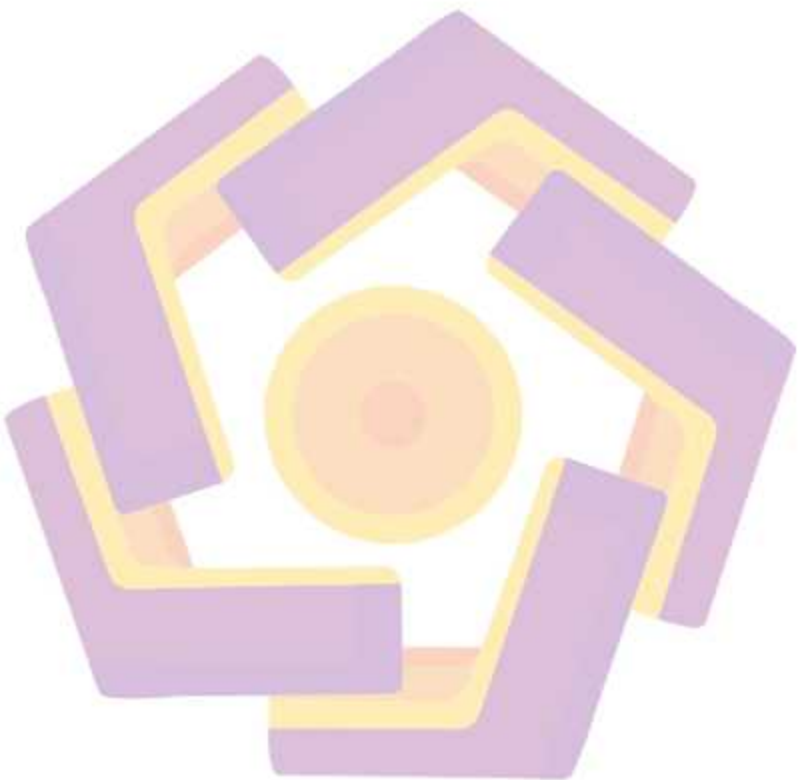
HALAMAN PERSEMBAHAN

Dengan ucapan Alhamdulillah yang tak terhingga, penulis menyatakan rasa syukur yang mendalam atas penyelesaian tesis ini. Penyelesaian ini tidak lepas dari bantuan dan kasih sayang yang diberikan oleh Allah Subhanahu Wa Ta'ala. Oleh karena itu, tesis ini dengan penuh kehormatan dipersembahkan kepada semua pihak yang terlibat, baik secara langsung maupun tidak langsung, dalam proses pembuatan tesis ini.

1. Orang tua dan saudara kakak-kakak, yang selalu mendoakan, memberikan semangat untuk menjalani perkuliahan.
2. Prof. Dr. M. Suyanto, M.M selaku Rektor Universitas AMIKOM Yogyakarta yang telah memberikan kesempatan kepada penulis untuk melanjutkan Studi jenjang Strata 2 Program Studi Magister Teknik Informatika di Universitas Amikom Yogyakarta.
3. Prof. Dr. Ema Utami, S.Si., M.Kom dan Bapak Anggit Dwi Hartanto, M.Kom yang telah membimbing penulis dari awal sampai akhir proses pembuatan tesis.
4. Muhammad Evandry Dewaksara (20.55.1341) dan Alvina Felicia Watratan (20.55.1346) yang selalu mengingatkan dan menyemangati untuk mengerjakan tesis.
5. Semua pihak yang tidak bisa disebutkan satu persatu yang sudah memberi semua ilmu pengetahuan, informasi dan segalanya sehingga penulis bisa menyelesaikan tesis ini.

HALAMAN MOTTO

"Keberhasilan bukanlah milik orang yang pintar. Keberhasilan adalah kepunyaan mereka yang senantiasa berusaha." B.J. Habibie



KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah Subhanahu Wa Ta'ala atas nikmat dan karunia-Nya yang memungkinkan penulis dapat menyelesaikan tesis yang berjudul "Analisis Perbandingan Elliptic Envelope, Isolation Forest, One-Class SVM dan Local Outlier Factor dalam Mendeteksi Status Anomali pada Gempa Bumi menggunakan Outlier dan Novelty". Penulis menyadari bahwa tesis ini tidak akan dapat diselesaikan dengan baik tanpa bimbingan, saran, dan motivasi dari berbagai pihak. Oleh karena itu, penulis dengan penuh rasa terima kasih menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. M. Suyanto, MM. selaku Rektor Universitas AMIKOM Yogyakarta.
2. Prof. Dr. Ema Utami, S.Si., M.Kom. dan Bapak Anggit Dwi Hartanto, M.Kom. selaku Dosen Pembimbing.
3. Bapak Alva Hendi Muhammad, S.T., M.Eng., Ph.D., Bapak Dr. Hanafi, S.Kom., M.Eng., dan Prof. Dr. Ema Utami, S.Si., M.Kom. selaku Dosen Penguji.
4. Orang tua serta saudara selaku wali yang telah memberikan dukungan dan motivasi

Penulis menyampaikan doa yang mendalam kepada Allah Subhanahu Wa Ta'ala atas dukungan dan bantuan yang diberikan kepada penulis dalam menyelesaikan tesis ini. Penulis berharap bahwa Allah Subhanahu Wa Ta'ala akan memberikan balasan yang lebih kepada seluruh pihak yang telah membantu dalam penyelesaian tesis ini. Diharapkan pula, tesis ini dapat memberikan manfaat yang signifikan bagi masyarakat dan bidang ilmu yang relevan.

Yogyakarta, 3 April 2023

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

INTISARI

Indonesia sering mengalami gempa bumi besar karena lokasinya yang berada di atas tiga lempeng tektonik aktif yang menyebabkan rekahan dan parit pada dasar darat dan laut. Sejak tahun 1900, lebih dari 1.250 gempa besar terjadi di Indonesia, sesuai dengan data dari BMKG Indonesia, sebanyak 12.351 gempa telah tercatat hingga tahun 2021. Meskipun banyak gempa terjadi, tidak semuanya memiliki dampak fatal dan banyak dianggap sebagai gempa umum.

Dalam penelitian ini, penulis membandingkan empat algoritma deteksi anomali, yaitu Elliptic Envelope (EE), Isolation Forest (IF), One-Class SVM (OCSVM) dan Local Outlier Factor (LOF), untuk mengidentifikasi gempa bumi yang tidak normal atau jarang terjadi dengan menggunakan konsep outlier dan novelty. Algoritma akan diuji melalui dua tahap pengujian; pertama, dengan menggunakan parameter default, dan kedua, dengan mengoptimalkan parameter Contamination/nu sebesar 0,025 atau 2,5%. Contamination/nu menunjukkan persentase data yang dianggap anomali. Pada pengujian deteksi anomali menggunakan novelty, hanya menggunakan parameter default. Hasil yang akan diamati meliputi Accuracy, Recall, Precision, F1-Score, AUC, dan Specificity.

Eksperimen telah menunjukkan bahwa Algoritma Isolation Forest memiliki nilai yang lebih tinggi daripada semua eksperimen lain dalam mendeteksi status anomali gempa bumi menggunakan outlier. Dalam membandingkan performa menggunakan parameter default, Isolation Forest memiliki precision 13,02%, recall (sensitivity) 99,35%, specificity 82,80%, accuracy 83,24%, AUC 91,08%, dan F1-score 23,03%. Setelah melakukan tuning hyperparameters, performa setiap model meningkat, dengan Isolation Forest memiliki precision 37,21%, recall (sensitivity) 36,85%, specificity 98,41%, accuracy 98,42%, AUC 67,63%, dan F1-score dari 37,03%. Sedangkan untuk Novelty, hasil evaluasi menunjukkan bahwa model Isolation Forest memiliki kinerja terbaik dalam kategori precision dan AUC, serta memiliki nilai tinggi dalam kategori Specificity.

ABSTRACT

Indonesia, large earthquakes happen frequently. Due to the location of Indonesia, over three active tectonic plates, fractures and trenches have formed on both the land and ocean floor. Over 1,250 large earthquake occurrences have occurred in Indonesia during the past 120 years, beginning in 1900. Based on data from the BMKG (Indonesian non-departmental government organization for meteorology, climatology, and geophysics) official website, up to 12,351 earthquakes occurred in Indonesia in 2021. Not every earthquake was fatal; many of them were considered common.

In this work, the author compares Elliptic Envelope (EE), Isolation Forest (IF), One-Class SVM (OCSVM), and Local Outlier Factor (LOF). 4 types of anomaly detection algorithms to locate earthquakes whose status is abnormal or out of the ordinary utilizing outliers and novelty. The application of anomaly detection for each algorithm will be tested in two ways; the first test will only use the default parameters, and the second test will use the parameter (Tuning Hyperparameters) at the contamination/nu parameter of 0.025 or 2.5%. Contamination/nu is a value that indicates the amount of data considered an anomaly—Accuracy, Recall, Precision, F1-Score, AUC, and Specificity.

Experiments have shown that Isolation Forest has a higher value than all other experiments in detecting anomalous status in earthquakes. In comparing performance using default parameters, Isolation Forest had a precision of 13.02%, recall (sensitivity) of 99.35%, specificity of 82.80%, accuracy of 83.24%, AUC of 91.08%, and F1-score of 23.03%. After tuning the hyperparameters, the performance of each model increased, with Isolation Forest had a precision of 37.21%, recall (sensitivity) of 36.85%, specificity of 98.41%, accuracy of 98.42%, AUC of 67.63%, and F1-score of 37.03%. In regards to Novelty, the evaluation results show that the Isolation Forest model possesses the best performance in the precision and AUC categories, as well as high values in the Specificity category.

Keywords—Anomaly Detection, Outlier, Novelty, Earthquake, Elliptic Envelope, Isolation Forest, One-Class SVM, Local Outlier Factor

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Gempa bumi adalah fenomena alam yang tidak dapat diprediksi kapan dan di mana akan terjadi. Akibat gempa bumi, permukaan bumi dapat mengalami kerusakan, seperti runtuhnya bangunan dan juga korban jiwa. Manusia hanya bisa melakukan upaya untuk meminimalisasi akibat gempa, misalnya dengan membangun struktur bangunan yang tahan gempa, membuat rambu peringatan, serta menyebarkan pendidikan mengenai gempa bumi dan cara untuk menyelamatkan diri. Untuk mengurangi jumlah korban jiwa akibat gempa, alat peringatan dini bisa dibuat, yang bisa berupa alat mekanis ataupun elektronik.

Gempa bumi terjadi akibat pelepasan energi secara tiba-tiba pada permukaan bumi, yang menyebabkan getaran. Pelepasan energi dapat memicu *gelombang seismik* yang dapat mengakibatkan kerusakan pada bangunan, tanaman, dan menimbulkan kehilangan nyawa. Meskipun teknologi telah berkembang, kita masih belum dapat memprediksi dengan pasti kapan dan di mana gempa bumi akan terjadi. Teknologi memungkinkan untuk memetakan daerah rawan gempa dan membuat bangunan yang tahan gempa. Kekuatan gempa dapat ditentukan dengan menggunakan alat yang disebut seismograf.

Gempa bumi dengan magnitudo kecil yang bernilai kurang dari 3 tidak akan menimbulkan kerusakan yang signifikan dan tidak dapat dirasakan oleh manusia, tetapi gempa bumi dengan magnitudo besar yang bernilai lebih dari 7 dapat

menyebabkan kerusakan yang cukup serius. Selain itu, tingkat kerusakan yang ditimbulkan oleh gempa bumi tidak hanya tergantung pada magnitudo gempa, tetapi juga tergantung pada tingkat daerah yang terdampak. Pelepasan energi secara tiba-tiba yang terjadi saat gempa bumi disebabkan oleh pergeseran *lempeng-lempeng bumi* yang saling bertabrakan.

Untuk menekan jumlah korban, pemerintah telah menyediakan alat peringatan dini, seperti menyalakan sirine di lokasi-lokasi tertentu ketika terjadi gempa. Sistem Peringatan Dini Gempa bertujuan sebagai petunjuk untuk melaksanakan evakuasi dan penyelamatan secepatnya, serta bertujuan untuk mengurangi dampak negatif dan menekan jumlah korban jiwa akibat gempa. (Bahri & Mungkin, 2019)

Penentuan status anomali pada gempa bumi dilakukan dengan menilai *Kedalaman* dan *kekuatan gempa* yang merupakan fitur dasar yang penting untuk memahami *mekanisme tektonik* serta tingkat ancaman gempa. Biasanya, semakin dangkal *kedalaman* dan semakin besar *magnitudonya*, semakin besar potensi kerusakan yang ditimbulkan. Walaupun kekuatan gempa cukup besar, namun jika terjadi pada kedalaman yang cukup jauh dari permukaan, maka gempa tersebut tidak akan menimbulkan ancaman yang besar. Hal ini dikarenakan kekuatan goncangan atau energi yang dilepaskan akan berkurang seiring dengan meningkatnya jarak dari *sumber gempa*. Sebagai contoh, gempa bumi yang terjadi di lepas pantai Jepang dengan kekuatan 8,5 pada kedalaman 590 kilometer (370 mil) di bawah permukaan tidak memberikan ancaman yang besar. Gempa bumi dapat terjadi di berbagai tempat di permukaan bumi dan sekitar 700 kilometer di

bawah permukaan. Dalam ilmu geologi, kedalaman gempa dikelompokkan menjadi tiga kategori oleh USGS (Badan Survey Geologi Amerika Serikat), yaitu dangkal (0 hingga 70 kilometer atau 43 mil), menengah (70-300 kilometer atau 43-186 mil), dan dalam (300-700 kilometer atau 186-434 mil).

Di Indonesia, gempa bumi besar sering terjadi. Hal ini disebabkan oleh lokasi Indonesia yang terletak di atas tiga lempeng tektonik yang aktif (*Eurasia, Indo-Australia, dan Pasifik*), sehingga terbentuk retakan dan parit di daratan maupun dasar laut. Sejak tahun 1900, telah terjadi lebih dari 1.250 kejadian gempa bumi besar di Indonesia dalam 120 tahun terakhir. (Ramdani & Chairunnisa, 2021)

Berdasarkan data situs web resmi Badan Meteorologi, Klimatologi, dan Geofisika (BMKG), tercatat 12.351 gempa bumi terjadi di Indonesia tahun 2021. Meskipun tidak semua gempa memiliki dampak fatal, namun beberapa di antaranya dianggap serius.

Dalam penelitian ini, kami akan menggunakan *Machine Learning* untuk membedakan apakah data gempa bumi tersebut termasuk dalam kategori gempa status anomali atau tidak. Empat algoritma *Unsupervised Learning*, yaitu *Elliptic Envelope (EE), Isolation Forest (IF), One-Class SVM (OCSVM)*, dan *Local Outlier Factor (LOF)*, akan kami gunakan untuk memisahkan data yang dianggap anomali dari yang normal (umum terjadi).

Adapun tujuan membandingkan keempat algoritma yaitu untuk mengetahui berapa tingkat performansi yang dicapai oleh metode Elliptic Envelope, Isolation Forest, One-Class SVM, dan Local Outlier Factor dalam mendeteksi status anomali gempa menggunakan outlier dan novelty serta metode mana yang dianggap paling

efektif berdasarkan perbedaan data yang signifikan dengan kumpulan data lain. Selain itu, penelitian ini juga akan mengevaluasi pengaruh tuning hyperparameter pada keempat metode tersebut dan apakah hal tersebut memberikan pengaruh yang signifikan pada peningkatan akurasi. Pertanyaan terakhir yang ingin dijawab adalah apakah tuning hyperparameter mempengaruhi efektivitas metode-metode tersebut dalam mendeteksi status anomali gempa menggunakan teknik Outlier atau apakah metode-metode tersebut tetap efektif tanpa melakukan tuning hyperparameter.

Anomali merupakan pola data yang memiliki ciri-ciri data yang berbeda dari kejadian yang biasa terjadi. *Deteksi anomali* memiliki relevansi yang signifikan dan sering menyediakan informasi penting yang dapat ditindaklanjuti di berbagai domain aplikasi. Sebagai contoh, anomali dalam *transaksi kartu kredit* dapat menunjukkan adanya penipuan terkait penggunaan kartu kredit. Sebuah tempat anomali dalam *gambar astronomi* dapat mengindikasikan adanya penemuan bintang baru. Pola lalu lintas *jaringan komputer* yang tidak biasa dapat menunjukkan adanya akses yang tidak sah. Aplikasi ini mengharuskan adanya *algoritma deteksi anomali* yang memiliki kinerja deteksi yang tinggi dan eksekusi yang cepat. (*Liu et al., 2008*)

Anomaly detection merupakan alat penting untuk mendeteksi ketidaknormalan dalam berbagai domain termasuk deteksi penipuan keuangan, intrusi jaringan komputer, analisis perilaku manusia, analisis ekspresi gen, dan banyak lagi. (Ahmed et al., 2016) Empat metode unsupervised learning, yakni EE, IF, OCSVM, dan LOF, memiliki tujuan yang sama, yaitu mengidentifikasi anomali pada data. Ketika diterapkan pada dataset gempa bumi, metode tersebut dapat

memisahkan gempa yang dianggap anomali dan tidak. Namun, setiap metode memiliki cara yang berbeda dalam mendeteksi anomali.

Penyisihan data gempa yang dianggap anomali dapat membantu dalam menganalisis sifat gempa yang tidak biasa atau tidak sesuai dengan standar yang ada. Hal ini dapat dilakukan tanpa perlu menggunakan alat pencarian seperti *SQL* atau *Pandas*. Selain itu, penyisihan data ini juga dapat membantu dalam mengelompokkan gempa berdasarkan magnitudo dan kedalaman, sehingga dapat membantu dalam menentukan tingkat risiko yang terkait dengan gempa tersebut. Dengan demikian, penyisihan data gempa yang dianggap anomali dapat memberikan informasi baru tentang sifat gempa dan memudahkan analisis faktor-faktor yang mempengaruhi terjadinya gempa.

Anomaly Detection merupakan proses yang berlawanan dengan *Clustering*. Clustering adalah pengelompokan kumpulan pola (biasanya direpresentasikan sebagai vektor pengukuran, atau titik dalam ruang multidimensi) ke dalam klaster-klaster berdasarkan kemiripan. Secara intuitif, pola-pola yang berada dalam klaster yang valid memiliki kemiripan yang lebih tinggi satu sama lain daripada dengan pola yang termasuk dalam klaster yang berbeda. (Wagstaff, 2012) Algoritma yang sering digunakan dalam *Clustering* adalah *K-Means* dan *DBSCAN*. Sedangkan *Anomaly Detection* merupakan proses yang bertujuan untuk mengidentifikasi data yang tidak memiliki kesamaan atau yang tidak terdapat dalam satu klaster, yang dikenal sebagai data *Outlier*.

Anomaly Detection sering juga digunakan dalam proses pembersihan data (*Data Cleansing*) untuk menghapus outlier dari kumpulan data sebelum melatih

model lain. (Sumber: Towards Data Science) Hal ini bertujuan untuk menghasilkan data yang lebih baik dan menghindari kesalahan pengolahan, seperti dalam kasus penggunaan *machine learning* yang mungkin terpengaruh oleh data yang tidak sesuai atau *anomali*. Namun, dalam penelitian ini, peneliti menggunakan metode *Anomaly Detection* untuk mencari data yang tidak normal (*anomali*) dalam dataset, bukan sebagai proses *Data Cleansing*.

Anomali adalah titik data yang terpisah dan berbeda dari kebanyakan titik data lainnya. Karena data *Outlier* tidak memiliki kesamaan dengan kebanyakan data lain, maka dapat dengan mudah ditentukan apakah data tersebut merupakan data *anomali* atau tidak. Data *anomali* dapat menunjukkan insiden kritis atau kejadian yang tidak diinginkan. Progresif, *Machine Learning* dapat digunakan untuk mengotomatisasi deteksi *anomali*.

Dengan menggunakan dataset Gempa Bumi yang tersedia, kita dapat mengevaluasi apakah terdapat gempa bumi dengan kondisi anomali yang pernah terjadi di Indonesia selama tahun 2021. Selain itu, kita juga dapat mengeksplorasi cara untuk mendeteksi kondisi anomali tersebut secara *realtime* di masa depan, sehingga dapat mengantisipasi terjadinya gempa bumi dengan status anomali kategori berbahaya. Peneliti menggunakan konsep *outlier* dan *novelty* dalam menjawab pertanyaan tersebut. Data yang dianggap anomali oleh metode deteksi anomali adalah data yang jarang terjadi dan berbeda dari kumpulan data yang ada, sedangkan data yang dianggap normal adalah data yang umumnya terjadi.

Contoh outlier dalam dataset gempa bumi adalah bila suatu wilayah memiliki rata-rata magnitudo gempa 5, dan ada satu gempa dengan magnitudo 8,5.

Gempa ini dianggap sebagai anomali. Sedangkan novelty bekerja dengan cara yang mirip dengan *outlier*, namun bedanya, magnitudo 8,5 yang terjadi adalah gempa baru dan belum pernah tercatat dalam dataset sebelumnya. Keberadaan *novelty* dapat digunakan untuk mendeteksi anomali pada gempa baru yang terjadi secara *real-time*.

Status anomali pada gempa bumi adalah kejadian gempa dalam dataset yang jarang terjadi karena kombinasi dari dua parameter, yaitu kedalaman dan magnitudo (kekuatan gempa). Contohnya, pada Juni 2021, gempa bumi terdalam pertama dilaporkan di jurnal *Geophysical Research Letters* pada kedalaman 751 kilometer di bawah permukaan bumi, dengan kejadian gempa yang dianggap tidak mungkin terjadi. Ini dikenal sebagai "gempa bumi setelah" gempa bumi M 7,9 yang mengguncang Kepulauan Bonin, Jepang pada tahun 2015. (Kiser et al., 2021)

Kemudian gempa bumi dengan kedalaman di bawah 10 kilometer dianggap sebagai anomali karena terjadinya gempa bumi jenis ini sangat jarang dan jarang direkam. Menurut *USGS* dalam salah satu artikelnya yang berjudul "*Mengapa banyak sekali gempa bumi terjadi pada kedalaman 10 km?*", kedalaman gempa bumi akan ditetapkan sebagai 10 km jika tidak diketahui dengan pasti. Hal ini menjelaskan mengapa banyak gempa bumi yang direkam pada kedalaman 10 km, sementara jumlah gempa bumi yang terjadi di bawah kedalaman tersebut sangat sedikit dan merupakan hal yang menarik untuk diteliti. Sebagai contoh, gempa bumi yang terjadi di Iran telah direkam oleh data radar satelit *Envisat ESA* dengan intensitas M 6,5 pada kedalaman 4-5 km, meskipun tidak menyebabkan kerusakan pada permukaan. (*Fialko et al., 2005*)

Anomali lain yang didasarkan pada kekuatan gempa bumi adalah gempa bumi di Aceh pada tanggal 26 Desember 2004 dengan intensitas M 9,3 yang terjadi pada kedalaman 30 km (*Ghobarah et al., 2006*), serta gempa bumi yang terjadi di Jepang dengan intensitas M 9,0 pada tanggal 11 Maret 2011 yang dikenal sebagai gempa Tohoku (*Kasai et al., 2013*). Gempa bumi dengan intensitas sebesar itu dianggap anomali oleh deteksi anomali karena jarang terjadi, seperti juga gempa bumi memiliki tingkat intensitas yang setara atau lebih tinggi daripada M 5,0 yang juga dianggap sebagai data anomali. Jenis anomali ini paling umum meskipun hanya terdapat sekitar 2,3% dari total data.

Gempa bumi anomali yang memiliki kedalaman tidak biasa dan magnitudo relatif besar sangat menarik untuk ditemukan. Contohnya adalah gempa bumi Teluk Palu pada tahun 2018 yang memiliki magnitudo 7,4 dan terjadi pada kedalaman 10 km. (*BMKG, 2018*) Pusat gempanya berada pada koordinat 0,18 derajat selatan dan 119,85 derajat timur, dengan jarak 26 km dari lokasi.

Penggunaan teknik deteksi anomali memiliki banyak kelebihan dibandingkan dengan metode tradisional seperti *SQL* dalam mengidentifikasi anomali pada data. Deteksi anomali menggunakan algoritma untuk secara otomatis mengidentifikasi data yang tidak biasa atau tidak sesuai dengan pola normal. Ini banyak digunakan dalam berbagai bidang, termasuk dalam pencarian anomali pada dataset gempa bumi. Keuntungan dari menggunakan teknik deteksi anomali adalah kapabilitasnya dalam mengidentifikasi anomali dengan cepat dan akurat tanpa memerlukan intervensi manusia.

Pertama, deteksi anomali dapat mengidentifikasi anomali data secara otomatis. Algoritma *machine learning* yang digunakan dalam deteksi anomali dapat mempelajari pola dan distribusi data secara otomatis, sehingga dapat memprediksi data anomali. Hal ini tidak mungkin dilakukan oleh *SQL*, di mana pengguna harus menulis *query* yang tepat dan memahami data yang akan dianalisis dengan baik untuk menemukan anomali.

Kedua, deteksi anomali dapat memproses data secara *real-time* dengan menggunakan *Novelty*. Novelty adalah bagian dari deteksi anomali yang ditemukan dalam empat algoritma yang dianalisis dalam penelitian ini. Algoritma ini bertujuan untuk mengidentifikasi anomali dalam waktu nyata. Ini berbeda dari *outlier*, yang biasanya muncul pada data yang sudah ada, sementara *Novelty* muncul pada data baru yang dimasukkan. Algoritma pembelajaran mesin mampu memproses data secara instan dan memberikan notifikasi tentang anomali data dengan cepat. Ini berbeda dari *SQL*, di mana pengguna harus memasukkan data ke dalam database sebelum dapat memprosesnya menggunakan *SQL*. Oleh karena itu, menggunakan algoritma *machine learning* untuk deteksi anomali dapat memberikan kecepatan dan efisiensi yang lebih tinggi dibandingkan dengan menggunakan *SQL*.

Ketiga, deteksi anomali dapat mengelompokkan data secara otomatis. Algoritma *machine learning* mampu memetakan pola dalam data dan mengelompokkannya sesuai dengan pola tersebut, mempermudah identifikasi data anomali. Ini berbeda dengan *SQL*, di mana pengguna harus mengelompokkan data secara manual dengan menggunakan *query* yang sesuai.

Dengan demikian, deteksi anomali dapat memberikan keuntungan dalam mengidentifikasi data yang tidak biasa atau statusnya jarang terjadi pada dataset gempa bumi, dibandingkan dengan menggunakan *SQL*. Tujuan penelitian ini bukan untuk membandingkan deteksi anomali dengan SQL, melainkan untuk membandingkan keempat algoritma yaitu *EE, IS, OCSVM* dan *LOF* untuk menemukan status gempa bumi yang anomali dengan teknik *outlier* dan *novelty detection*. Penambahan informasi diatas terkait kelebihan *Anomaly Detection* jika dibandingkan *SQL* hanya berupa informasi tambahan.

Berikut adalah beberapa contoh penelitian yang menggunakan teknik deteksi anomali:

Pada penelitian ini, peneliti mengkaji berbagai metode deteksi anomali yang banyak digunakan untuk mendeteksi *Outlier* dalam berbagai bidang. Salah satu metodenya adalah *Isolation Forest* yang digunakan untuk mendeteksi transaksi curang pada transaksi kartu kredit. Peneliti juga membandingkan hasil dari metode lain, seperti *One-Class SVM, Local Outlier Factor,* dan *K-Means*. Hasil perbandingan menunjukkan bahwa *Isolation Forest* memiliki F1 Score 0.0544, Accuracy 0.9512, dan AUC 0.9168, yang menandakan bahwa metode ini sangat efektif untuk mendeteksi anomali pada transaksi kartu kredit. (*Ounacer et al., 2018)*

Penelitian berikutnya yang menggunakan teknik deteksi anomali adalah studi tentang penemuan anomali pada data penggunaan obat di rumah sakit. Studi ini menggunakan dua set data dan membandingkan kinerja dari tiga algoritma deteksi anomali, yaitu *K-Means, Local Outlier Factor,* dan *One-Class SVM*. Hasil studi menunjukkan bahwa ketiga algoritma dapat berhasil menemukan *Outlier*,

namun *One-Class SVM* terbukti lebih unggul dibandingkan dengan *Local Outlier Factor* dan *K-Means*, seperti yang ditemukan dalam penelitian tersebut. *(Budiarto et al., 2019)*

Dalam penelitian ini, peneliti membandingkan empat teknik deteksi anomali berbasis *Unsupervised Learning*, yaitu *One-Class SVM, Local Outlier Factor, Isolation Forest,* dan *Elliptic Envelope*, menggunakan dataset pesawat ulang-alik dan satelit. Peneliti juga membandingkan hasil dari teknik deteksi anomali berbasis *Unsupervised* dan *Supervised Learning* dengan menggunakan dataset yang sama. Hasil penelitian menunjukkan bahwa deteksi anomali berbasis *Unsupervised Learning* setara atau bahkan lebih baik daripada deteksi anomali berbasis *Supervised Learning* pada dataset pesawat ulang-alik dan satelit. Dalam perbandingan keempat teknik deteksi anomali berbasis *Unsupervised Learning*, *Isolation Forest* tampak memiliki hasil yang konsisten karena tidak membuat representasi parametrik dari ruang fitur. (*Shriram & Sivasankar, 2019*)

Penelitian selanjutnya bertujuan untuk melakukan pemantauan kesehatan pada turbin gas menggunakan metode deteksi anomali *Isolation Forest*. Tujuannya adalah untuk memastikan bahwa turbin gas beroperasi dengan baik dan mencegah terjadinya masalah yang tidak terduga dan memerlukan perawatan mahal. Dalam penelitian ini, peneliti menggunakan data pemantauan yang diambil dari delapan mesin aero CFM56-7B yang berbeda. Hasil analisis menunjukkan bahwa metode *Isolation Forest* mampu mendeteksi anomali dengan akurasi tinggi pada data yang tidak berlabel dan pada kumpulan data yang relatif kecil. (*Zhong et al., 2019*)

Penelitian ini memiliki perbedaan yang signifikan dibandingkan dengan penelitian sejenis lainnya, terutama dalam hal dataset yang digunakan. Dalam hal ini, penelitian ini menggunakan dataset gempa bumi dan mengaplikasikan pendekatan *novelty detection* untuk melakukan deteksi data anomali secara *real-time*. Keterkaitan dengan penelitian lain juga akan dibahas secara lebih terperinci dalam bagian tinjauan pustaka.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan sebelumnya, rumusan masalah dalam penelitian ini adalah sebagai berikut:

a. Berapakah tingkat performansi yang dicapai oleh metode *Elliptic Envelope, Isolation Forest, One-Class SVM,* dan *Local Outlier Factor* dalam mendeteksi status anomali gempa menggunakan *outlier* dan *novelty*?
b. Metode mana yang dianggap sebagai yang paling efektif dalam mendeteksi status anomali gempa menggunakan teknik *Outlier* dan *Novelty* berdasarkan perbedaan data yang signifikan dengan kumpulan data lain?
c. Bagaimana pengaruh *tuning hyperparameter* pada metode *Elliptic Envelope, Isolation Forest, One-Class SVM,* dan *Local Outlier Factor* dalam mendeteksi status anomali gempa menggunakan teknik *Outlier* dan apakah hal tersebut memberikan pengaruh yang signifikan pada peningkatan akurasi ?
d. Apakah *tuning hyperparameter* mempengaruhi efektivitas metode *Elliptic Envelope, Isolation Forest, One-Class SVM,* dan *Local Outlier Factor* dalam mendeteksi status anomali gempa menggunakan teknik *Outlier*, atau metode-metode tersebut tetap efektif tanpa melakukan tuning hyperparameter?

1.3. Batasan Masalah

Untuk membantu penelitian ini menjadi lebih terarah dan memudahkan pembahasan, kami telah menetapkan beberapa batasan masalah. Berikut adalah beberapa batasan masalah yang ditetapkan dalam penelitian ini

a. Data yang digunakan hanya berupa file *csv*.
b. Data gempa bumi hanya berfokus pada gempa bumi yang terjadi di Indonesia.

c. Data gempa bumi yang digunakan adalah data gempa yang terjadi dari Januari hingga Desember 2021, sebanyak 12.351 data yang diambil dari website resmi bmkg.go.id.

d. Data gempa bumi tidak dibedakan berdasarkan penyebabnya.

e. Terdapat 5 variabel pada Data set Gempa Bumi (*Datetime, Lintang, Bujur, Kedalaman* dan *Magnitudo*) namun hanya Kedalaman dan Magnitudo yang digunakan dalam modelling, variabel lainnya hanya berfungsi sebagai informasi tambahan

f. Pengolahan data serta pemrosesan algoritma menggunakan *Python* dengan memanfaatkan library *Scikit-learn*

g. *Performance Metrics/Evaluation Metrics* yang digunakan pada penelitian ini, untuk masing-masing algoritma adalah *Precision, Recall, Specificity, Accuracy, AUC* dan *F1-score*.

h. Menggunakan *platform Notebook Google Colab* untuk menjalankan algoritma dan mengolah Data set.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

a. Untuk mengkaji kemampuan penerapan algoritma deteksi anomali dalam mendeteksi anomali pada dataset gempa bumi menggunakan metode *Outlier Detection* dan *Novelty Detection*.

b. Untuk mengevaluasi dan membandingkan kinerja dari metode *Isolation Forest, Local Outlier Factor, One-Class SVM,* dan *Elliptic Envelope* dalam mendeteksi anomali pada dataset gempa bumi dengan menggunakan metode *Outlier* dan *Novelty*, dan untuk menentukan algoritma terbaik dalam mendeteksi anomali pada dataset tersebut.

1.5. Manfaat Penelitian

Adapun manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

a. Peneliti dapat memahami algoritma atau metode yang dapat digunakan untuk mendeteksi titik data yang berbeda dari kumpulan data yang ada.

b. Peneliti dapat memahami perbedaan antara *Isolation Forest, Local Outlier Factor, One-Class SVM,* dan *Elliptic Envelope* dalam mendeteksi gempa yang statusnya anomali.

c. Peneliti dapat memahami algoritma atau metode terbaik dalam mendeteksi gempa yang statusnya anomali atau dataset lain yang memiliki karakteristik yang mirip dengan dataset gempa bumi

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Penelitian yang terkait dengan deteksi anomali pada transaksi kartu kredit untuk menemukan transaksi penipuan adalah salah satu yang paling sering kita temukan. Salah satu penelitian yang relevan dalam topik ini adalah yang dilakukan oleh (*Ounacer et al., 2018*) yang menganalisis transaksi kartu kredit menggunakan 4 algoritma deteksi anomali yaitu *Isolation Forest, Local Outlier Factor, One-Class SVM* dan *K-Means*. Dalam penelitian ini, peneliti menggunakan *F1 Score, Accuracy* dan *AUC* Score untuk membandingkan keempat algoritma tersebut. Hasil penelitian menunjukkan bahwa algoritma *Isolation Forest* memiliki skor AUC 91% dibandingkan LOF 49%, OCSVM 51% dan K-Means 52%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa algoritma *Isolation Forest* lebih akurat daripada tiga algoritma lainnya dalam mendeteksi transaksi penipuan kartu kredit.

Deteksi anomali Merupakan salah satu topik yang banyak dikaji di dalam dunia keamanan komputer dan teknologi informasi salah satunya dikaji dan dilakukan oleh (*Shriram & Sivasankar, 2019)* yang berjudul "*Anomaly Detection on Shuttle data using Unsupervised Learning Techniques*". Penelitian ini bertujuan untuk membandingkan 4 algoritma deteksi anomali berbasis *Unsupervised Learning*, yaitu *One-Class Support Vector Machine (SVM), Local Outlier Factor, Isolation Forest,* dan *Elliptic Envelope*. Selain itu, penelitian ini juga mencoba untuk membandingkan algoritma deteksi anomali berbasis

Unsupervised Learning dengan algoritma deteksi anomali *Supervised Learning* menggunakan dataset yang sama yaitu pesawat ulang-alik dan satelit. Hasil penelitian ini menunjukkan bahwa metode terbaik dalam melakukan deteksi anomali adalah deteksi anomali yang berbasis *Unsupervised Learning*. Dari keempat algoritma deteksi anomali *Unsupervised Learning*, Isolation Forest dianggap sebagai algoritma terbaik dengan nilai *Evaluation Metric (Precision 0.99, Recall 0.97, F-Score 0.98)*. Namun, penelitian ini juga memiliki beberapa kelemahan, yaitu proses perbandingan algoritma deteksi anomali *Unsupervised Learning* dengan algoritma deteksi anomali *Supervised Learning* yang tidak seimbang, dimana dalam *Unsupervised Learning* digunakan 4 algoritma sementara *Supervised Learning* hanya digunakan 2 algoritma yaitu *SVM* dan *k-NN*. Peneliti juga menyatakan bahwa kelemahan dari algoritma deteksi anomali berbasis *Supervised Learning* adalah membutuhkan jumlah dataset yang besar agar dapat bekerja secara efektif.

Deteksi anomali juga digunakan dalam bidang keamanan perangkat *Internet of Things (IoT)* yang salah satunya diteliti oleh (*Bezerra et al., 2019*) yang berjudul "*IoTDS: A one-class classification approach to detect botnets in internet of things devices*" Penelitian ini bertujuan untuk mendeteksi *botnet* pada perangkat *IoT* menggunakan deteksi anomali dengan dataset yang dikumpulkan dari perangkat *IoT (Raspberry Pi)* yang terinfeksi oleh tujuh botnet berbeda, yaitu *Hajime, Aidra, BashLite, Mirai, Doflo, Tsunami,* dan *Wroba*. Jumlah dataset yang digunakan dalam penelitian ini sebanyak 1,080 data. Algoritma deteksi anomali yang digunakan dalam penelitian ini bekerja dengan cara menganalisis penggunaan *CPU* dan

memori host, *suhu CPU*, dan jumlah task yang berjalan untuk mengklasifikasikan perilakunya sebagai berbahaya atau tidak. Penelitian ini menggunakan 4 algoritma deteksi anomali berbasis *Unsupervised Learning*, yaitu *One-Class SVM, Local Outlier Factor, Isolation Forest* dan *Elliptic Envelope*. Hasil penelitian menunjukkan bahwa *One-Class SVM* dan *Local Outlier Factor* adalah algoritma terbaik, dengan *LOF* memiliki sedikit keunggulan dengan rata-rata F1-score 94%. Kelebihan lain dari penelitian ini adalah menggunakan lebih dari 3 *Evaluation Metric*, seperti *Precision, Recall, Specificity, Accuracy, AUC* dan *F1-score* untuk menilai kinerja algoritma.

Penelitian yang menggunakan algoritma deteksi anomali untuk meningkatkan akurasi deteksi serangan *DDoS* dan mengurangi tingkat *false positive* yang dihasilkan dilakukan oleh (*Das et al., 2020*) yang menggunakan algoritma *One-Class Support Vector Machine (SVM), Local Outlier Factor, Isolation Forest,* dan *Elliptic Envelope*. Selain itu, peneliti juga menggunakan *Naive Bayes* dan *Logistic Regression* untuk mendapatkan akurasi deteksi yang lebih baik. Kelebihan dari penelitian ini adalah peneliti menggunakan konsep *outlier* dan *novelty* dalam mendeteksi serangan *DDoS*. Novelty digunakan untuk mendeteksi serangan *DDoS* secara *real-time*, sedangkan *outlier* digunakan untuk mendeteksi *DDoS* yang tidak terdeteksi sebelumnya. Peneliti juga menentukan lima kombinasi *hyperparameter* terbaik untuk masing-masing algoritma. *Performance metrics* yang digunakan dalam penelitian ini antara *lain accuracy, false positive rate, precision, recall,* dan *F-1 score*. Algoritma terbaik pada penelitian ini tidak disebutkan secara spesifik,

namun dapat ditentukan dengan melihat tabel perbandingan performance metrics yang disertakan dalam laporan penelitian.

Penelitian yang memanfaatkan algoritma deteksi anomali untuk meningkatkan pemahaman tentang fenomena aliran kompleks yang dihasilkan dari pembakaran roket dilakukan oleh (*Rüttgers & Petrarolo, 2021*) yang menganalisis data gambar uji pembakaran *roket hibrida*. Dalam penelitian ini, peneliti menggunakan 4 jenis algoritma deteksi anomali (*Elliptic Envelope, One-Class SVM, Local Outlier Factor, Isolation Forest*) untuk mengidentifikasi anomali dalam proses pembakaran. Dari hasil penelitian, diambil kesimpulan bahwa *Local Outlier Factor* adalah algoritma terbaik dalam mendeteksi anomali pada data gambar uji pembakaran roket hibrida karena algoritma ini mampu mengungkapkan fenomena-fenomena menarik yang muncul selama pembakaran.

Penelitian yang memanfaatkan deteksi anomali dalam fungsi platform teknologi besar (*Large Technology Platforms*) berdasarkan analisis log penelitian ini bukanlah satu-satunya yang dilakukan, salah satu riset yang relevan dilakukan oleh (*Dunaev & Zaytsev, 2019*) yang menganalisis log dari *Apache Solr, Apache Kafka*, dan *proses Java*. Dalam penelitian ini, peneliti menggunakan 4 jenis algoritma deteksi anomali (*Elliptic Envelope, One-Class SVM, Local Outlier Factor, Isolation Forest*) untuk mengidentifikasi anomali dalam log sistem. Jumlah log yang digunakan dalam penelitian ini sekitar 6000 baris dengan jumlah kolom sekitar 15.000 kolom. Hasil penelitian menunjukkan bahwa algoritma *Elliptic Envelope* dan *Local Outlier Factor* tidak dapat menentukan kejadian darurat maupun anomali dengan baik, sedangkan algoritma *One-Class SVM* dan *Isolation*

Forest mampu mengidentifikasi namun tidak semua titik pra-anomali. Oleh karena itu, peneliti menganggap pengoperasian semua algoritma tersebut tidak memuaskan dan tidak cocok untuk memecahkan masalah pendeteksian anomali saat memproses data dalam jumlah besar.

2.2. Landasan Teori

2.2.1. Anomaly Detection

Anomaly Detection merupakan proses identifikasi *Outlier* dalam dataset yang dianalisis. *Outlier* didefinisikan sebagai objek data yang tidak sesuai dengan pola atau perilaku yang diharapkan dalam dataset tersebut. Algoritma yang digunakan dalam *Anomaly Detection* memiliki berbagai aplikasi, seperti dalam bidang bisnis, sains, dan keamanan. (*Kotu & Deshpande, 2019*) Anomaly Detection juga dikenal dengan sebutan *Outlier Detection* dan *Novelty Detection*, karena tidak hanya dapat digunakan untuk mendeteksi *outlier*, namun juga dapat mendeteksi kemunculan objek data baru (*novelty*) secara *real-time*.

Anomaly detection merupakan teknik yang memanfaatkan *machine learning* untuk mendeteksi dan mengidentifikasi data yang memiliki perbedaan yang signifikan dari kumpulan data yang ada. Terdapat dua jenis *anomaly detection* pada *machine learning*, yaitu *Supervised Anomaly Detection* dan *Unsupervised Anomaly Detection*.

Penerapan Anomaly Detection dalam perusahaan IT umumnya digunakan untuk berbagai tujuan, seperti:

- Data cleaning

- Intrusion detection
- Fraud detection
- Systems health monitoring
- Event detection in sensor networks
- Ecosystem disturbances identification

2.2.2. Outlier & Novelty

Outlier adalah titik data yang berbeda dari kumpulan data lainnya pada dataset. Data yang berbeda dari kumpulan data lainnya dapat memberikan informasi yang bermanfaat karena data tersebut berada di luar kenormalan atau anomali. Hal ini dapat terjadi karena data tersebut memiliki nilai di atas atau di bawah rata-rata. (*Aggarwal, 2017*)

Sedangkan *Novelty* hampir mirip dengan *Outlier*, perbedaannya adalah bahwa jika *Outlier* adalah data anomali yang sudah ada dalam dataset, maka *Novelty* adalah data baru yang dideteksi sebagai anomali. (*Pimentel et al., 2014*)

Mendeteksi *Outlier* dan *Novelty* pada dataset dapat memanfaatkan metode *Machine Learning*. Outlier dapat muncul karena mekanisme kesalahan, perubahan perilaku sistem, perilaku curang, kesalahan manusia, instrument kesalahan atau hanya melalui penyimpangan alami dalam populasi. (*Hodge, V., Austin, 2015*)

2.2.3. Gempa Bumi

Eksternalizasi energi yang tiba-tiba di dalam Bumi yang ditandai dengan bergetarnya permukaan Bumi, dikenal sebagai gempa bumi. Ini disebabkan oleh patahnya lapisan batuan pada kerak Bumi. *Akumulasi energi* yang menyebabkan terjadinya gempa bumi dihasilkan dari pergerakan *lempeng-lempeng tektonik* yang

terjadi di dalam Bumi. Energi yang dihasilkan dari pergerakan ini kemudian dipancarkan ke segala arah berupa gelombang gempa bumi, sehingga efeknya dapat dirasakan sampai ke permukaan Bumi. (Sumber: *BMKG*)

Gempa bumi dapat diklasifikasikan menjadi dua kategori utama berdasarkan penyebab dan kedalamannya. Pertama, berdasarkan penyebab terjadinya, gempa bumi dapat dibedakan menjadi tiga jenis, yaitu *gempa vulkanik, gempa tektonik,* dan *gempa runtuhan* atau *terban*. Kedua, berdasarkan kedalamannya, gempa bumi dapat dibedakan menjadi tiga jenis, yaitu *gempa bumi dalam, gempa bumi menengah,* dan *gempa bumi dangkal*. Dalam melakukan pengukuran gempa bumi, beberapa parameter yang digunakan termasuk waktu *terjadinya (Origin Time), lokasi pusat (Episenter), kedalaman pusat (Depth),* dan *kekuatan (Magnitude)* gempa bumi. (Sumber: *BPBD Kota Banda Aceh*)

2.2.4 Jenis-Jenis Gempa Bumi Berdasarkan Penyebab, Kedalaman, Gelombang, Magnitude, dan Tingkat Kerusakan yang Ditimbulkan

Indonesia terletak di atas tiga lempeng besar, yaitu Lempeng Eurasia, Lempeng Pasifik, dan Lempeng Indo-Australia, yang menjadi salah satu faktor penyebab seringnya terjadi gempa di wilayah Indonesia. Ada berbagai jenis gempa bumi yang dapat dibedakan berdasarkan penyebab, kedalaman, gelombang, magnitude, dan tingkat kerusakan yang ditimbulkan.

Beberapa jenis gempa berdasarkan penyebab antara lain gempa bumi tektonik yang disebabkan oleh pergeseran lempeng tektonik, gempa bumi vulkanik yang disebabkan oleh aktivitas magma, gempa bumi tumbukan yang disebabkan oleh tumbukan meteor atau astroid, gempa bumi runtuhan yang biasanya terjadi di

daerah gunung kapur atau pertambangan, dan gempa bumi buatan yang disebabkan oleh aktivitas manusia seperti aktivitas nuklir atau peledakan dinamit.

Gempa bumi dapat dibedakan berdasarkan kedalamannya menjadi tiga jenis, yaitu gempa bumi dalam, menengah, dan dangkal. Gempa bumi dalam memiliki pusat gempa atau hiposentrum yang lebih dari 300 km di bawah permukaan Bumi dan umumnya tidak berbahaya. Gempa bumi menengah memiliki pusat gempa atau hiposentrum yang berada di kisaran 60 km - 300 km di bawah permukaan Bumi, getarannya lebih terasa dan dapat menimbulkan kerusakan ringan. Terakhir, gempa bumi dangkal memiliki pusat gempa yang kurang dari 60 km di bawah permukaan Bumi, dan umumnya dapat menimbulkan kerusakan besar.

Jenis gempa bumi dapat dibedakan berdasarkan gelombangnya menjadi dua jenis, yaitu gelombang primer dan gelombang sekunder. Gelombang primer adalah gelombang atau getaran yang merambat di dalam tubuh Bumi dengan kecepatan antara 7 km/detik sampai 14 km/detik dan berasal dari pusat gempa. Sedangkan gelombang sekunder merupakan gelombang yang merambat di dalam tubuh Bumi dengan kecepatan yang lebih rendah dan tidak dapat merambat melalui lapisan cair. (Sumber: grid.id)

Gempa bumi dapat dikelompokkan berdasarkan magnitudonya, yang mengindikasikan besarnya energi yang dilepaskan oleh gempa, serta dampak kerusakan yang mungkin terjadi. Berikut adalah beberapa kategori gempa bumi berdasarkan magnitudonya: (Sumber: Kumparan)

1. Gempa bumi ultra mikro

Gempa bumi ultra mikro biasanya memiliki magnitudo kurang dari 2 skala richter (SR) dan tidak dirasakan oleh manusia. Gempa ini jarang menimbulkan kerusakan.

2. Gempa bumi mikro

Gempa bumi mikro memiliki magnitudo antara 2 hingga 3 SR. Gempa ini tidak dirasakan oleh kebanyakan orang, tetapi dapat terdeteksi oleh alat seismograf.

3. Gempa bumi kecil

Gempa bumi kecil memiliki magnitudo antara 3 hingga 4 SR. Gempa ini biasanya terasa oleh orang-orang, tetapi jarang menimbulkan kerusakan yang signifikan.

4. Gempa bumi sedang

Gempa bumi sedang memiliki magnitudo antara 4 hingga 5 SR. Gempa ini dapat menyebabkan kerusakan ringan pada bangunan dan dapat dirasakan oleh banyak orang.

5. Gempa bumi merusak

Gempa bumi merusak memiliki magnitudo antara 5 hingga 6 SR. Gempa ini dapat menyebabkan kerusakan yang signifikan pada bangunan, seperti retak pada dinding dan pecahnya kaca.

6. Gempa bumi besar

Gempa bumi besar memiliki magnitudo antara 7 hingga 8 SR. Gempa ini dapat menyebabkan kerusakan yang parah pada bangunan dan dapat menimbulkan bencana alam lainnya, seperti tanah longsor.

7. Gempa bumi sangat besar

Gempa bumi sangat besar memiliki magnitudo 8 SR atau lebih. Gempa ini dapat memicu terjadinya tsunami dan dapat menyebabkan kerusakan yang luas dan serius pada bangunan serta infrastruktur lainnya.

2.2.5 Tsunami

Tsunami adalah deretan gelombang laut yang dapat merambat dengan kecepatan lebih dari 900 km/jam dan biasanya diakibatkan oleh gempa bumi yang terjadi di dasar laut. Kecepatan gelombang tsunami ditentukan oleh kedalaman laut. Pada laut yang memiliki kedalaman 7000 m, kecepatannya dapat mencapai hampir 942,9 km/jam, tetapi tinggi gelombangnya di tengah laut tidak lebih dari 60 cm sehingga kapal-kapal di atasnya jarang merasakan adanya tsunami. Tsunami berbeda dengan gelombang laut biasa karena panjang gelombang tsunami antara dua puncaknya lebih dari 100 km dan selisih waktu antara puncak-puncak gelombangnya berkisar antara 10 menit hingga 1 jam. Ketika mencapai daerah pantai yang dangkal, teluk, atau muara sungai, kecepatan gelombangnya menurun, tetapi tinggi gelombangnya meningkat hingga puluhan meter dan bersifat merusak.

Tsunami yang disebabkan oleh gempa bumi tidak selalu terjadi, karena ada beberapa syarat yang harus terpenuhi. Pertama, pusat gempa harus terjadi di dasar laut, dan kedalaman pusat gempa harus kurang dari 60 km. Contohnya, pada tanggal

26 Desember 2004, gempa bumi dengan kekuatan 9 magnitudo terjadi di kedalaman 30 km di dasar laut sebelah barat daya Aceh. Hal ini memicu terbentuknya gelombang tsunami dengan kecepatan awal sekitar 700 km/jam. Gelombang ini menyebar ke segala arah dari pusat tsunami dan melanda wilayah Aceh dan Sumatera Utara dengan kecepatan antara 15 - 40 km per jam dan tinggi gelombang 2 hingga 48 meter. Dampak bencana ini sangat besar, menewaskan lebih dari 250.000 orang dalam waktu tiga jam setelah gempa bumi, dan negara-negara di kawasan Samudera Hindia juga terkena dampak dari tsunami. (ESDM, 2012)

2.2.6. Elliptic Envelope

Elliptic Envelope adalah algoritma pembelajaran mesin yang digunakan untuk mendeteksi *outlier* dan *novelty* dengan membentuk elips di sekitar titik data (pusat data) dengan menggunakan beberapa kriteria dan klasifikasi. Data yang berada di dalam elips disebut sebagai inlier (data normal), sedangkan data di luar elips dianggap sebagai *outlier*. Agar bekerja dengan maksimal, dataset yang digunakan dalam *Elliptic Envelope* harus terdistribusi secara normal (*distribusi Gaussian*).

Elliptic Envelope membutuhkan *hyperparameter contamination* untuk mendeteksi *outlier*, yang merupakan nilai yang mewakili *proporsi outlier* dalam dataset. Rentang nilai yang dapat digunakan sebagai *contamination hyperparameter* adalah mulai dari 0 hingga 0.5, dengan nilai default adalah 0.1. Jika kita percaya bahwa terdapat banyak *outlier* dalam dataset, kita dapat meningkatkan nilai *contamination*. Batasan utama dalam menggunakan metode ini adalah tidak diketahui proporsi yang tepat dari *outlier* dalam dataset.

Outlier dalam *Elliptic Envelope* dideteksi dengan menggunakan *Minimum Covariance Determinant*, yang merupakan salah satu *ekuivarian affine* pertama dan *estimator* yang kuat untuk lokasi dan *sebaran multivariat* (*Rousseeuw, 1984, 1985*). Persamaan metrik jarak statistik (*Hubert et al., 2018*) yang digunakan ditunjukkan pada Gambar 2.1 sebagai berikut:

$$d(x, \mu, \Sigma) = \sqrt{(x-\mu)'\Sigma^{-1}(x-\mu)},$$

Gambar 2. 1 Persamaan metrik jarak statistik

2.2.7. Isolation Forest

Isolation Forest (IF) adalah algoritma yang diperkenalkan pertama kali pada tahun 2008, yang digunakan dalam bidang pendekatan *isolation tree-based*. Dalam beberapa tahun terakhir, IF menjadi perhatian yang semakin besar dari para peneliti dan praktisi. Seperti namanya, IF adalah *algoritma ensemble* yang mirip dengan algoritma *Random Forest* yang populer, namun digunakan dalam deteksi anomali tanpa pengawasan. Secara spesifik, IF adalah kumpulan *binary trees*, di mana setiap pohon bertujuan untuk mengisolasi sebuah wilayah di dalam data yang hanya dihuni oleh satu titik data. Ide dasar dari IF adalah bahwa karena anomali umumnya sedikit jumlahnya, prosedur isolasi akan lebih cepat dalam memisahkan *outlier* dari sisa data daripada saat menangani *inlier*.

Algoritma IF terdiri dari dua tahap: pelatihan dan pengujian. Pada tahap pelatihan, setiap pohon isolasi akan membagi data menjadi partisi acak dari domain. Algoritma ini didasarkan pada asumsi bahwa anomali, pada rata-rata, memerlukan partisi yang lebih sedikit untuk diisolasi. Oleh karena itu, inlier umumnya

ditemukan pada daun yang terletak pada bagian terdalam dari pohon, sementara outlier ditemukan pada daun yang lebih dekat dengan akar. Skor anomali adalah proporsional terhadap kedalaman rata-rata daun di mana setiap titik data berada. Pseudo-code untuk tahap pelatihan dan pengujian dapat ditemukan pada Algoritma. (*Tran, 2022*)

Isolation Forest adalah algoritma yang digunakan untuk mendeteksi anomali dengan mengisolasi (menghitung seberapa jauh titik data dari data lainnya) bukan dengan memodelkan titik normal. Algoritma ini pertama kali dikembangkan oleh *Fei Tony Liu* pada tahun 2007 *(Liu et al., 2008)*. Isolation Forest mengintroduksikan metode yang secara eksplisit mengisolasi anomali dengan menggunakan *pohon biner*, yang menunjukkan kemungkinan pendeteksian anomali yang lebih cepat yang secara langsung menargetkan anomali tanpa membuat profil dari semua instance normal.

Isolation Forest memiliki kompleksitas *waktu linier* dengan konstanta rendah serta memiliki persyaratan memori yang rendah. Algoritma ini berfungsi dengan baik pada data bervolume tinggi. Teknik yang paling umum digunakan untuk mendeteksi anomali adalah berdasarkan pada konstruksi profil dari apa yang dianggap sebagai "normal": anomali dilaporkan sebagai contoh dari kumpulan data yang tidak sesuai dengan profil normal.

Isolation Forest menggunakan pendekatan yang berbeda dari teknik deteksi anomali yang lain. Alih-alih mencoba membangun model dari *instance normal*, algoritma ini secara eksplisit mengisolasi titik anomali dalam kumpulan data. Keuntungan utama dari pendekatan ini adalah kemungkinan untuk mengeksploitasi

teknik pengambilan sampel hingga batas yang tidak diizinkan oleh metode berbasis profil, sehingga menciptakan algoritma yang sangat cepat dengan permintaan memori yang rendah.

Algoritma:

Diberikan sampel titik data X, algoritma Isolation Forest membangun *Isolation Tree (iTree)*, T, dengan menggunakan langkah-langkah berikut.

1. Pilih secara acak atribut q dan nilai split p
2. Bagilah X menjadi dua himpunan bagian dengan menggunakan aturan q < p.Subset akan sesuai dengan subpohon kiri dan subpohon kanan di T.
3. Ulangi langkah 1-2 secara rekursif hingga node saat ini hanya memiliki satu sampel atau semua nilai pada node saat ini memiliki nilai yang sama

Algoritma kemudian mengulangi langkah 1-3 beberapa kali untuk membuat beberapa *Isolation Tree*, menghasilkan *Isolation Forest*. Berdasarkan cara *Isolation Tree* diproduksi dan sifat-sifat titik anomali, kita dapat mengatakan bahwa sebagian besar titik anomali akan terletak lebih dekat ke akar pohon karena lebih mudah diisolasi dibandingkan dengan titik normal.

Setelah memiliki *Isolation Forest* (kumpulan *Isolation Tree*), algoritma menggunakan persamaan skor anomali yang ditunjukkan pada Gambar 2.2 dengan memberikan titik data x dan ukuran sampel m:

$$s(x,m) = 2^{\frac{-E(h(x))}{c(m)}}$$

Gambar 2. 2 Persamaan skor anomali

2.2.8. One-Class SVM

One-Class Support Vector Machine (One-Class SVM) adalah metode yang digunakan untuk mendeteksi anomali dalam data. Ini merupakan perkembangan dari *Support Vector Machine (SVM)* yang diajukan oleh (*Schölkopf et al., 2001*). Selain digunakan untuk mendeteksi anomali, algoritma ini juga dapat digunakan untuk estimasi kerapatan. Keunikan dari *One-Class SVM* adalah dapat digunakan pada dataset yang tidak memiliki label. Metode ini mengeidentifikasi *outlier* dari contoh data positif dan menggunakannya sebagai contoh data negatif untuk dianalisis.

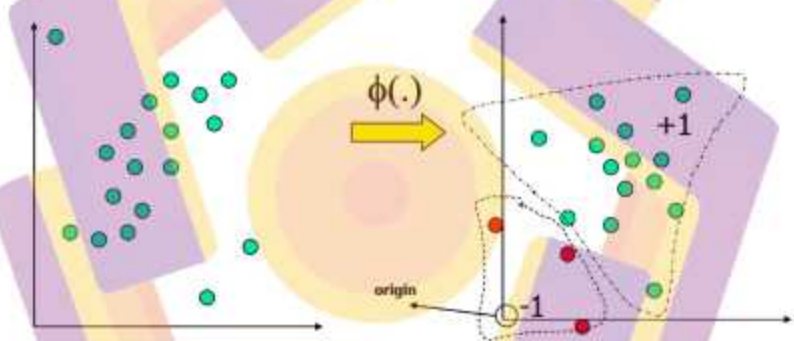
Teknik *One-Class Support Vector Machine (SVM)* dimanfaatkan untuk mengukur keberbedaan segmen data yang berbeda dari data lainnya. Dalam kasus ini, kita memiliki dataset yang memiliki *distribusi probabilitas P* yang diinginkan untuk menentukan *subset S* dari data tersebut sehingga probabilitas dari data pengujian yang diambil dari P berada di luar dari *subset S*. Solusi untuk masalah ini didapat dengan mengestimasi fungsi yang bernilai positif pada subset S dan negatif pada komplemen dari subset S. Fungsi tersebut akan memberikan nilai +1 pada area yang memiliki hampir semua data dan memberikan nilai -1 jika data berada di luar area tersebut.

$$f(x) = \begin{cases} +1, & if\ x \in S \\ -1, & if\ x \notin S \end{cases}$$

Gambar 2. 3 Persamaan

Teknik *Support Vector Data Description (SVDD)* mencakup transformasi dari data input ke dalam *feature space* yang telah dihasilkan dari fungsi *kernel*.

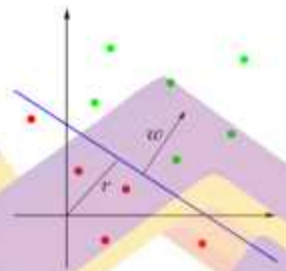
Data yang dianggap sebagai *outlier* ditentukan sebagai satu-satunya data negatif yang berasal dari origin. *Relaxation* parameter kemudian digunakan untuk memisahkan data yang bukan *outlier* dari *origin*. Algoritma ini sama dengan *klasifikasi biner* yang digunakan pada *Support Vector Machine (SVM)* untuk mencari bidang pemisah yang optimal yang dapat memisahkan data dari *origin* dengan margin terbesar. Persoalan untuk menemukan bidang pemisah ini dinyatakan secara matematis dalam persamaan yang ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Persamaan pencarian bidang

Dimana ξi adalah penalti yang diberikan kepada data anomali yang berada di sisi salah dari bidang pemisah yang memisahkan data normal, sementara v mengatur keseimbangan antara memaksimalkan margin dari *origin* dan mencakup sebagian besar data pada daerah yang dibuat oleh bidang pemisah dengan *rasio outlier* yang terdapat pada data pelatihan (seperti parameter C pada SVM untuk klasifikasi).

$$\min \frac{1}{2}|w|^2 + \frac{1}{vn}\sum_{i=1}^{n}\xi_i - r$$
$$s.t.\ (w.\phi(x_i)) \geq r - \xi_i,$$
$$\xi_i \geq 0$$



Gambar 2. 5 Persamaan fungsi kernel

Persoalan yang dijelaskan di atas dapat dirumuskan sebagai berikut melalui pendekatan *Lagrange Multiplier* dan *fungsi kernel*.

$$\min \sum_{i=1,\ j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$$

$$s.t. \sum_{i=1}^{n} \alpha_i = 1,$$

$$0 \leq \alpha_i \leq \frac{1}{vn}$$

Gambar 2. 6 Persamaan Lagrange Multiplier dan fungsi kernel

Berdasarkan hasil pelatihan, diperoleh nilai parameter α_i. Nilai r kemudian dapat dihitung menggunakan persamaan yang sesuai. Hasil dari proses pembelajaran adalah sebuah fungsi yang dapat digunakan untuk melakukan

klasifikasi.

$$r = \sum \alpha_j K(x_i, x_j)$$

$$f(x) = \sum_i \alpha_i K(x_i, x\ ) - r\ ,\ x_i = support\ vector$$

Gambar 2. 7 Hasil proses pembelajaran

2.2.9. Local Outlier Factor

Local Outlier Factor (LOF) merupakan algoritma pendeteksi *outlier* berbasis kepadatan yang dikembangkan untuk menemukan *outlier* dengan menghitung *deviasi lokal* dari titik data yang diberikan. Algoritma ini sangat cocok digunakan pada dataset yang tidak merata. Penentuan *outlier* dihitung berdasarkan perbandingan kepadatan antara titik data dengan titik tetangganya. Semakin rendah *densitas* sebuah titik data, semakin besar kemungkinannya untuk diidentifikasi sebagai *outlier*. (*Cheng et al., 2019*)

Metode *Local Outlier Factor (LOF)* dikemukakan oleh *Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng,* dan *Jörg Sander* pada tahun 2000. *LOF* mengaplikasikan konsep yang sama dengan metode *DBSCAN* dan *OPTICS*, yaitu konsep jarak inti dan jarak keterjangkauan yang digunakan untuk perkiraan kerapatan setempat. Persamaan yang digunakan dalam metode *LOF* dapat diamati dalam ilustrasi Gambar 2.8.

$$LOF_k(A) = \frac{\sum_{X_J \in N_k(A)} LRD_k(X_J)}{||N_k(A)||} \times \frac{1}{LRD_k(A)}$$

Gambar 2. 8 Persamaan LOF

Untuk menemukan anomali dalam dataset, Algoritma *Local Outlier Factor (LOF)* mengukur *deviasi lokal* dari titik data p2D terhadap tetangga terdekatnya k. Sebuah titik p dinyatakan sebagai anomali jika *LOF*-nya "besar". (*Huang, 2017*) *LOF* dari suatu titik diperoleh seperti yang dijelaskan dalam langkah-langkah berikut (Gambar 2.9)

Algorithm LOF (Local Outlier Factor) Computation

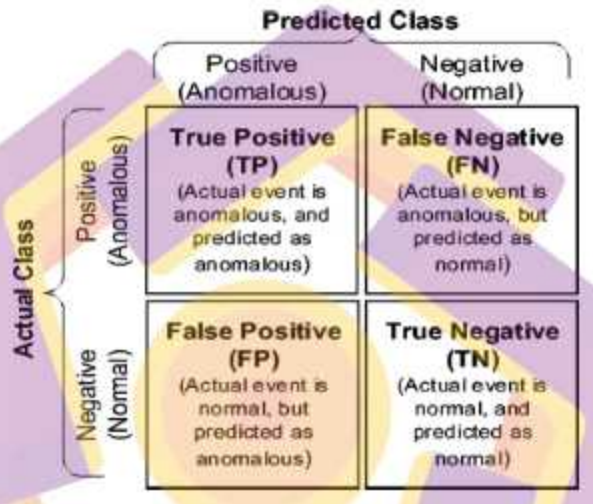
Require: $k, \mathcal{D}$.
Ensure: L_k - LOF score for each object in $\mathcal{D}$

$L_k = \emptyset$.
for $p \in \mathcal{D}$ **do**
  $N_k(p) = NULL$.
  for $q \in \mathcal{D}$ **do**
    if $|N_k(p)| < k$ **then**
      Add q in $N_k(p)$
    else
      Let $s* \in N_k(p)$ be such that $dist(p, s*) \geq dist(p, s)$ for all $s \in N_k(p)$;
      if $dist(p, s*) > dist(p, q)$ **then**
        Replace $s* \in N_k(p)$ by q
      end if
    end if
  end for
  $d_k(p)$=max$\{dist(p, s) | s \in N_k(p)\}$
end for
for $p \in \mathcal{D}$ **do**
  for $q \in \mathcal{D}$ **do**
    $d_{reach}(p, q) = \max\{d_k(p), d(p, q)\}$
  end for
end for
for $p \in \mathcal{D}$ **do**
  $l_k(p) = \frac{|N_k(p)|}{\sum_{q \in N_k(p)} d_{reach}(p,q)}$
end for
for $p \in \mathcal{D}$ **do**
  $L_k(p) = [\frac{\sum_{o \in N_k(p)} \frac{l_k(o)}{l_k(p)}}{|N_k(p)|}]$
end for
return L_k

Gambar 2. 9 Algoritma LOF

2.2.10. Confusion Matrix

Dasar perhitungan *performance metrics* adalah *confusion matrix*, misalnya pada *precision, recall, F1-score* dan lainnya. Tabel confusion matrix pada *anomaly detection* dapat diamati dalam ilustrasi Gambar 2.10.



Gambar 2. 10 Confusion matrix

Berikut penjelasan tabel confusion matrix pada Gambar 2.1

TP = Kejadian sebenarnya adalah anomali, dan diprediksi sebagai anomali.

FN = Kejadian sebenarnya adalah anomali, tetapi diprediksi sebagai normal.

FP = Kejadian sebenarnya adalah normal, tetapi diprediksi sebagai anomali.

TN = Kejadian sebenarnya adalah normal, dan diprediksi sebagai normal.

Berikut adalah penjelasan mengenai confusion matrix jika di terapkan pada dataset gempa bumi.

True Positive (TP) adalah jumlah gempa dengan status anomali yang diprediksi sebagai anomali.

True Negative (TN) adalah jumlah gempa dengan status normal yang diprediksi sebagai normal.

False Positive (FP) adalah jumlah gempa dengan status normal yang diprediksi sebagai anomali.

False Negative (FN) adalah jumlah gempa dengan status anomali yang diprediksi sebagai normal.

2.2.11 Performance Metrics

Untuk mengetahui hasil deteksi anomali terbaik, diperlukan cara untuk mengevaluasi modelnya. Beberapa metode yang dapat digunakan untuk mengevaluasi model adalah *precision, recall (sensitivity), specificity, accuracy, AUC,* dan *F1-score*.

Evaluasi model pertama adalah *precision,* yang digunakan untuk menjawab pertanyaan "*Berapa proporsi anomali yang teridentifikasi yang merupakan anomali asli ?*" Maksudnya adalah seberapa besar proporsi dari anomali yang terdeteksi oleh model yang merupakan anomali sebenarnya (bukan hasil kesalahan model dalam mengidentifikasi anomali). Ini dapat diinterpretasikan sebagai tingkat keakuratan model dalam mendeteksi anomali. *Nilai precision* yang tinggi menunjukkan bahwa model memiliki tingkat keakuratan yang tinggi dalam mendeteksi anomali, sebaliknya nilai precision yang rendah menunjukkan bahwa model memiliki tingkat keakuratan yang rendah dalam mendeteksi anomali. rumusnya dapat diamati dalam ilustrasi Gambar 2.11.

$$\frac{True\ Positive}{(True\ Positive + False\ Positive)}$$

Gambar 2. 11 Rumus precision

Evaluasi model yang kedua adalah *recall* atau biasa juga dikenal sebagai *sensitivity*, yang merupakan kebalikan dari *specificity*. *Recall* digunakan untuk menjawab pertanyaan "*Berapa proporsi anomali asli yang diidentifikasi (atau benar-benar positif)?*" Rumus recall dapat diamati dalam ilustrasi Gambar 2.12.

$$\frac{True\ Positive}{(True\ Positive + False\ Negative)}$$

Gambar 2. 12 Rumus recall (sensitivity)

Maksud dari "*Berapa proporsi anomali asli yang diidentifikasi (atau benar-benar positif)*" adalah seberapa besar proporsi dari anomali sebenarnya yang terdeteksi oleh model sebagai anomali. Ini dapat diinterpretasikan sebagai kemampuan model dalam mengidentifikasi anomali. *Nilai recall* yang tinggi menunjukkan bahwa model memiliki kapasitas yang tinggi dalam membedakan anomali, sebaliknya nilai *recall* yang rendah menunjukkan bahwa model memiliki kemampuan yang rendah dalam mendeteksi anomali. *Recall* biasanya digunakan bersama dengan *precision* dalam evaluasi model.

Evaluasi model yang ketiga adalah *specificity*, yang digunakan untuk menjawab pertanyaan "*Berapa proporsi data normal yang sebenarnya yang diprediksi sebagai data normal (atau benar-benar negatif) ?*" Maksud dari "*Berapa proporsi data normal yang sebenarnya yang diprediksi sebagai data normal (atau benar-benar negatif)*" adalah seberapa besar proporsi dari data normal yang

terdeteksi oleh model sebagai data normal. Ini dapat diinterpretasikan sebagai kemampuan model dalam membedakan data normal dari anomali. *Nilai specificity* yang tinggi menunjukkan bahwa model memiliki kemampuan yang tinggi dalam membedakan data normal dari anomali, sebaliknya nilai *specificity* yang rendah menunjukkan bahwa model memiliki kemampuan yang rendah dalam membedakan data normal dari anomali. rumusnya dapat diamati dalam ilustrasi Gambar 2.13.

$$Specificity = \frac{TN}{TN + FP}$$

Gambar 2. 13 Rumus specificity

Evaluasi keempat yang dilakukan pada model ini adalah dengan menghitung *accuracy* (akurasi). *Accuracy* digunakan untuk mengetahui seberapa baik model dalam memberikan label yang benar pada semua data. Rumus untuk menghitung *accuracy* dapat diamati dalam ilustrasi Gambar 2.14.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Gambar 2. 14 Rumus accuracy

Evaluasi kelima yang dilakukan pada model ini adalah dengan menghitung auc (*Area Under the Curve*). Auc digunakan untuk memperkirakan kemungkinan bahwa model akan memberikan peringkat yang lebih tinggi pada contoh positif yang dipilih secara acak dibandingkan contoh negatif yang juga dipilih secara acak. Rumus untuk menghitung *AUC* dapat diamati dalam ilustrasi Gambar 2.15.

$$\textbf{AUC: } \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)$$

Gambar 2. 15 Rumus AUC

Evaluasi terakhir yang dilakukan pada model adalah dengan menghitung *f1-score*. f1-score merupakan metrik yang digunakan untuk mengevaluasi kinerja model deteksi anomali dengan menggabungkan *Recall* dan *Precision* menggunakan *rata-rata harmonic*. Rumus untuk menghitung *f1-score* dapat dilihat pada Gambar 2.16.

$$\frac{2 * (Recall * Precision)}{(Precision + Recall)}$$

Gambar 2. 16 Rumus F1-Score

Rata-rata harmonic adalah salah satu jenis rata-rata yang digunakan untuk menghitung nilai rata-rata dari sejumlah data. Rata-rata ini dihitung dengan cara mengambil *reciprocal* (kebalikan) dari setiap elemen data, menjumlahkan hasilnya, dan kemudian mengambil *reciprocal* lagi dari hasil jumlah tersebut. Rata-rata *harmonic* sering digunakan dalam evaluasi model *machine learning*, terutama ketika *Recall* dan *Precision* dianggap penting dan perlu diperhitungkan secara seimbang.

Berikut adalah penjelasan ringkas mengenai *Precision, Accuracy, Recall, F1-Score, AUC* dan *Specificity*

Precision: Precision mengukur seberapa tepat model dalam mengidentifikasi anomali. *Precision* dihitung dengan membagi jumlah deteksi yang benar (*True Positive*) dengan jumlah keseluruhan deteksi yang dideteksi sebagai

anomali (*True Positive + False Positive*). Semakin tinggi nilai *Precision*, semakin sedikit deteksi salah yang dilakukan oleh model.

Accuracy: Accuracy mengukur seberapa tepat model dalam mengidentifikasi anomali secara keseluruhan. *Accuracy* dihitung dengan membagi jumlah deteksi yang benar (*True Positive + True Negative*) dengan jumlah keseluruhan deteksi (*True Positive + True Negative + False Positive + False Negative*). Semakin tinggi nilai *Accuracy*, semakin tepat model dalam mengidentifikasi anomali.

Recall: Recall mengukur seberapa baik model dalam mengidentifikasi seluruh anomali yang ada. *Recall* dihitung dengan membagi jumlah deteksi yang benar (*True Positive*) dengan jumlah keseluruhan deteksi yang sebenarnya anomali (*True Positive + False Negative*). Semakin tinggi nilai *Recall*, semakin baik model dalam mengidentifikasi seluruh anomali yang ada.

F1-Score: F1-Score merupakan *rata-rata harmonik* dari *Precision* dan *Recall*. F1-Score digunakan untuk mengukur performa model secara keseluruhan, dengan mempertimbangkan tingkat keakuratan dan kelengkapan dalam mengidentifikasi anomali.

AUC (Area Under the Curve): AUC merupakan area di bawah *kurva ROC* (*Receiver Operating Characteristic*). ROC adalah grafik yang menunjukkan performa model dengan mengukur persentase *True Positive Rate (TPR)* dibandingkan dengan *False Positive Rate (FPR)*. Nilai *AUC* yang lebih tinggi menunjukkan performa model yang lebih baik.

Specificity: Specificity mengukur seberapa baik model dalam mengidentifikasi data normal. *Specificity* dihitung dengan membagi jumlah deteksi yang benar (*True Negative*) dengan jumlah keseluruhan deteksi yang sebenarnya normal (*True Negative + False Positive*). Semakin tinggi nilai *Specificity*, semakin baik model dalam mengidentifikasi data normal.

Untuk menghitung *performance metrics*, kita membutuhkan hasil *confusion matrix* yang telah diperoleh sebelumnya. Ini adalah tahapan yang lazim diambil saat melakukan proses menghitung *performance metrics*:

1. Tentukan terminologi yang digunakan pada confusion matrix, yaitu *True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN)*.
2. Hitung precision dengan menggunakan rumus precision = TP / (TP + FP). Precision menunjukkan seberapa sering model mengeluarkan prediksi yang benar dari seluruh prediksi yang dikeluarkan.
3. Hitung recall dengan menggunakan rumus recall = TP / (TP + FN). Recall menunjukkan seberapa sering model dapat menemukan semua kejadian yang sebenarnya terjadi.
4. Hitung specificity dengan menggunakan rumus specificity = TN / (TN + FP). Specificity menunjukkan seberapa sering model dapat mengeluarkan prediksi yang benar untuk kejadian yang sebenarnya tidak terjadi.
5. Hitung accuracy dengan menggunakan rumus accuracy = (TP + TN) / (TP + TN + FP + FN). Accuracy menunjukkan seberapa sering model dapat mengeluarkan prediksi yang benar secara keseluruhan.

6. Hitung AUC (area under curve) dengan menggunakan rumus AUC = (TP / (TP + FN)) x (TN / (TN + FP)). AUC menunjukkan seberapa baik model dapat memisahkan kejadian yang sebenarnya terjadi dan yang sebenarnya tidak terjadi.
7. Hitung f-score dengan menggunakan rumus f-score = 2 x ((precision x recall) / (precision + recall)). F-score menunjukkan rata-rata harmonis dari precision dan recall.

2.3. Keaslian Penelitian

Tabel 2. 1 Matriks literatur review dan posisi penelitian Analisis Perbandingan Elliptic Envelope, Isolation Forest, One-Class Svm Dan Local Outlier Factor Dalam Mendeteksi Status Gempa Bumi Menggunakan Outlier Dan Novelty

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
1	Anomaly Detection on Shuttle data using Unsupervised Learning Techniques	Shriram, S. Sivasankar, E. 2019	Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kemampuan empat algoritma deteksi anomali berbasis unsupervised learning, yaitu One-Class SVM, Local Outlier Factor, Isolation Forest, dan Elliptic Envelope, dalam mengidentifikasi anomali dalam dataset pesawat ulang-alik dan satelit. Selain itu, penelitian ini juga akan mengevaluasi performa algoritma deteksi anomali	Hasil dari percobaan yang dilakukan menunjukkan bahwa algoritma deteksi anomali berbasis unsupervised learning yang digunakan dalam penelitian ini, yaitu One-Class SVM, Local Outlier Factor, Isolation Forest, dan Elliptic Envelope, memiliki performa yang setara atau dalam beberapa kasus lebih baik dibandingkan dengan teknik supervised learning untuk dataset pesawat ulang-alik dan satelit. Secara keseluruhan, algoritma Isolation Forest tampaknya memiliki	Penelitian ini dapat diperluas di masa depan dengan menerapkan teknik ansambel unsupervised learning untuk meningkatkan efisiensi model keseluruhan. Hal ini akan memungkinkan kita untuk mengevaluasi kinerja dari beberapa algoritma unsupervised learning secara bersama-sama dan mengambil keputusan yang lebih baik. Dengan demikian, kita dapat meningkatkan performansi deteksi anomali dan menemukan cara yang lebih efektif untuk mengidentifikasi anomali dalam dataset.	Dalam penelitian ini, penulis hanya fokus pada evaluasi performa dari empat algoritma deteksi anomali berbasis unsupervised learning, yaitu One-Class SVM, Local Outlier Factor, Isolation Forest, dan Elliptic Envelope, tanpa membandingkan dengan algoritma deteksi anomali berbasis supervised learning. Selain itu, penulis juga menggunakan dataset yang berbeda-beda dalam pengukuran performa dari masing-masing algoritma tersebut, yang dapat mempengaruhi hasil dari analisis yang dilakukan.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			berbasis unsupervised learning dibandingkan dengan algoritma deteksi anomali berbasis supervised learning.	hasil yang paling konsisten untuk kedua dataset tersebut.		
2	Using Isolation Forest in anomaly detection: the case of credit card transactions	Ounacer, Soumaya El Bour, Hicham Ait Oubrahim, Younes Ghoumari, Mohamed Yassine Azzouazi, Mohamed 2018	Tujuan dari penelitian ini adalah untuk mengembangkan sistem pendeteksi penipuan kartu kredit yang mampu mendeteksi jumlah transaksi baru tertinggi secara real-time dengan tingkat akurasi yang tinggi. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi keefektifan dari empat pendeteksi anomali yang	Algoritma Isolation Forest terbukti efektif dalam mendeteksi penipuan kartu kredit. Hasil pengujian menunjukkan bahwa algoritma ini memiliki skor AUC sebesar 91%, yang jauh lebih tinggi dibandingkan dengan metode LOF (49%), OCSVM (51%), dan K-Means (52%). Analisis ini menunjukkan bahwa Isolation Forest lebih akurat dalam mendeteksi transaksi penipuan dibandingkan	Dalam penelitian selanjutnya, peneliti akan mengevaluasi potensi implementasi arsitektur baru yang dapat meningkatkan kemampuan sistem pendeteksi penipuan kartu kredit secara real-time. Arsitektur ini akan mengkombinasikan teknologi Apache Spark dan algoritma Isolation Forest untuk meningkatkan efektivitas dalam mendeteksi transaksi penipuan. Penelitian ini diharapkan	Penulis mengevaluasi alternatif metode dalam penelitian ini, yaitu metode Elliptic Envelope, sebagai gantinya dari metode K-Means yang sering digunakan dalam penelitian terkait. Hal ini dilakukan karena perbedaan tujuan penelitian dan dataset yang digunakan yang lebih cocok dengan metode Elliptic Envelope. Penelitian ini berfokus pada penerapan metode tersebut dan evaluasi performansi dibandingkan dengan metode lain yang digunakan dalam penelitian sejenis.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			digunakan dalam mendeteksi penipuan kartu kredit.	dengan metode lain yang diteliti. Selain itu, algoritma ini juga dapat mendeteksi beberapa kesalahan yang tidak dideteksi oleh tiga metode lainnya. Kinerja Isolation Forest dalam mendeteksi transaksi penipuan jauh lebih baik dibandingkan dengan metode lain yang diteliti.	dapat memberikan kontribusi yang signifikan dalam pengembangan sistem pendeteksi penipuan kartu kredit yang lebih canggih.	
3	Comparison of New Anomaly Detection Technique for Wind Turbine Condition Monitoring Using Gearbox SCADA Data	McKinnon, Conor Carroll, James McDonald, Alasdair Koukoura, Sofia Infield, David Soraghan, Conaill 2020	Penelitian ini bertujuan untuk membandingkan 3 deteksi anomali yaitu One-Class Support Vector Machine (OCSVM), Isolation Forest (IF) dan Elliptical Envelope (EE) untuk memonitor	Ditemukan bahwa untuk akurasi IF dan OCSVM memiliki kinerja serupa di kedua rezim pelatihan yang disajikan. OCSVM berperforma lebih baik untuk pelatihan generik, dan IF berperforma lebih baik untuk pelatihan khusus. Secara keseluruhan, IF dan	Pekerjaan di masa depan akan melihat penyelidikan referensi tetap bulan sehat, dengan bulan uji geser. Perbandingan metode ini untuk turbin yang sehat, dan turbin yang gagal selama periode pengukuran juga akan dilakukan untuk validasi.	Penulis membandingkan 4 metode deteksi anomali unsupervised learning, sedangkan penelitian ini hanya menggunakan 3 metode. Metode yang tidak digunakan pada penelitian ini adalah Local Outlier Factor. Penulis menggunakan dataset yang berbeda dari penelitian ini

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			kondisi turbin angin menggunakan deteksi anomali	OCSVM memiliki akurasi rata-rata 82% untuk semua konfigurasi yang dipertimbangkan, dibandingkan dengan 77% untuk EE.		
4	A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning	Das, Saikat Venugopal, Deepak Shiva, Sajjan 2020	Tujuan dari penelitian ini adalah untuk mendeteksi serangan DDOS dan meningkatkan akurasi deteksi dengan menggunakan anomali detection (One class SVM, Local Outlier Factor, Elliptic Envelope dan Isolation Forest)	Model yang diusulkan tidak hanya mampu mendeteksi serangan DDoS yang ada, tetapi juga menggunakan pengklasifikasi deteksi outlier, ia memiliki kemampuan untuk mendeteksi serangan DDoS yang tidak terlihat atau baru (novelty)	Di masa mendatang, peneliti berencana untuk mengurangi fitur sendiri menggunakan teknik pengurangan fitur dan pengetahuan domain yang berbeda. Dengan penelitian ini sebagai dasar, peneliti akan mempertimbangkan metode deep learning dan software agent dalam mendeteksi	Penulis menggunakan ke 4 metode yang digunakan oleh peneliti. Perbedaannya terletak pada tujuan penelitian dan dataset yang digunakan. Penelitian ini hanya menggunakan dataset offline untuk mendeteksi novelty sedangkan penulis mampu menggunakan dataset online dengan memanfaatkan API

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
			serta bertujuan untuk mengurangi tingkat positif palsu		serangan DDoS secara lebih akurat.	
5	Logs analysis to search for anomalies in the functioning of large technology platforms	Dunaev, Maxim Zaytsev, Konstantin 2019	Penelitian ini bertujuan untuk mendeteksi anomali pada log, log yang yang digunakan adalah log yang berasal dari Apache Solr, Apache Kafka, dan proses java (java-processes operate). Adapun jumlah log yang digunakan terbilang cukup besar yaitu sekitar 6000 baris log dengan jumlah kolom sekitar 15.000 kolom.	hasil penelitian ini menunjukkan bahwa algoritma Elliptic Envelope dan Local Outlier Factor tidak dapat menentukan, baik kejadian darurat maupun anomali sedangkan algoritma One-Class SVM dan Isolation Forest mampu mengidentifikasi namun tidak semua titik pra-anomali. Oleh karena itu, peneliti menganggap pengoperasian semua algoritma ini tidak memuaskan, dan algoritma ini tidak cocok untuk memecahkan masalah pendeteksian anomali	Kelemahan yang terdapat pada penelitian ini karena menggunakan dataset yang memiliki kolom terlalu banyak yaitu sekitar 15.000 kolom	Dataset yang digunakan oleh penulis hanya menggunakan 2 kolom yaitu Magnitudo (kekuatan gempa) dan Kedalaman gempa. Perbedaan lainnya terletak pada tujuan penelitian serta dataset yang digunakan pada penelitian.

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
				saat memproses data dalam jumlah besar.		
6	IoTDS: A One-Class Classification Approach to Detect Botnets in Internet of Things Devices	Bezerra, Vitor Hugo da Costa, Victor Guilherme Turrisi Barbon Junior, Sylvio Miani, Rodrigo Sanches Zarpelão, Bruno Bogaz 2019	Penelitian ini bertujuan untuk mendeteksi botnet pada perangkat Internet of Things menggunakan deteksi anomali yang datasetnya dikumpulkan dari IoT device (Raspberry Pi) yang terinfeksi oleh tujuh botnet berbeda (Hajime, Aidra, BashLite, Mirai, Doflo, Tsunami, dan Wroba). jumlah dataset yang digunakan 1,080 data	Hasil penelitian menunjukkan One-Class SVM dan Local Outlier Factor adalah algoritma terbaik, Dimana LOF memiliki sedikit keunggulan dengan rata-rata F1-score 94%. Kelebihan pada penelitian ini karena menggunakan lebih dari 3 Evaluation Metric diantaranya Precision, Recall, Specificity, Accuracy, AUC dan F1-score.	Untuk penelitian di masa depan, peneliti bermaksud untuk mengevaluasi pendekatan yang diusulkan di berbagai perangkat dan botnet IoT, termasuk fitur berbasis host lainnya (mis., syscalls) dan menguji cara lain untuk mengurangi botnet, seperti memblokir alamat IP dan mengubah IP menggunakan router jaringan yang ditentukan perangkat lunak bersama dengan IoTDS. Selain itu, peneliti juga berencana membuat sistem yang mampu menangani penyimpangan konsep	Jika peneliti memanfaatkan anomali deteksi untuk mendeteksi botnet pada perangkat IoT maka penulis memanfaatkan deteksi anomali untuk mendeteksi gempa yang berbahaya secara realtime dan offline

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
					algoritma pembelajaran mesin dari serangan terhadap kapasitas deteksinya. Terakhir, menggunakan teknik pembelajaran mesin gabungan dapat menjadi alternatif yang menjanjikan untuk membuat sistem lebih tangguh	

BAB III

METODE PENELITIAN

3.1. Jenis, Sifat dan Pendekatan Penelitian

3.1.1. Jenis Penelitian

Penelitian yang dilakukan oleh peneliti merupakan jenis penelitian kuantitatif, di mana peneliti menggunakan metode perhitungan matematis untuk menemukan hasil yang diharapkan.

3.1.2. Sifat Penelitian

Sifat dari penelitian yang akan dilakukan adalah analisis dan eksperimental dimana peneliti melakukan analisis dan eksperimen untuk mengevaluasi kemampuan empat algoritma yaitu *Elliptic Envelope, Isolation Forest, One-Class SVM* dan *Local Outlier Factor* dalam mendeteksi status gempa bumi. Analisis dilakukan dengan menggunakan pendekatan *outlier* dan *novelty detection*. Metode yang digunakan adalah dengan melakukan perbandingan antara keempat algoritma tersebut dengan tujuan untuk menemukan algoritma yang paling efektif dalam mendeteksi status gempa bumi.

3.1.3. Pendekatan Penelitian

Dalam penelitian ini, peneliti menggunakan pendekatan berbasis *metode penelitian kuantitatif*. Pendekatan ini akan digunakan untuk mengumpulkan dan menganalisis data secara sistematis dan obyektif sesuai dengan desain penelitian yang telah ditentukan oleh peneliti sebelumnya.

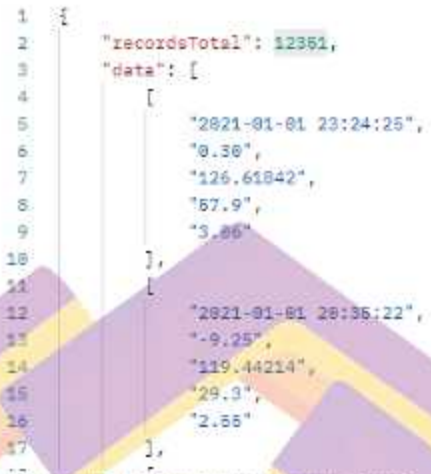
3.2. Metode Pengumpulan Data

Data yang digunakan dalam analisis perbandingan ini adalah data gempa bumi yang terjadi selama tahun 2021, dengan jumlah sebanyak 12.351 data. Data ini diperoleh dari website resmi BMKG melalui endpoint API https://dataonline.bmkg.go.id/data_gempa_bumi/data, dengan menggunakan parameter-parameter yang ditunjukkan pada Tabel 3.1.

Tabel 3. 1 Parameter enpoint api bmkg

Key	Value
tanggal-min	01-01-2021
tanggal-max	31-12-2021
lintang-min	-11
lintang-max	6
bujur-min	-95
bujur-max	141
magnitude-min	1.00
magnitude-max	9.50
depth-min	1.00
depth-max	1000.00

Dataset yang digunakan dalam penelitian ini terdiri dari lima kolom, yaitu tanggal dan waktu (datetime), lintang, bujur, kedalaman, dan magnitudo. Contoh dataset dapat diamati dalam ilustrasi Gambar 3.1.



Gambar 3. 1 Contoh dataset gempa bumi

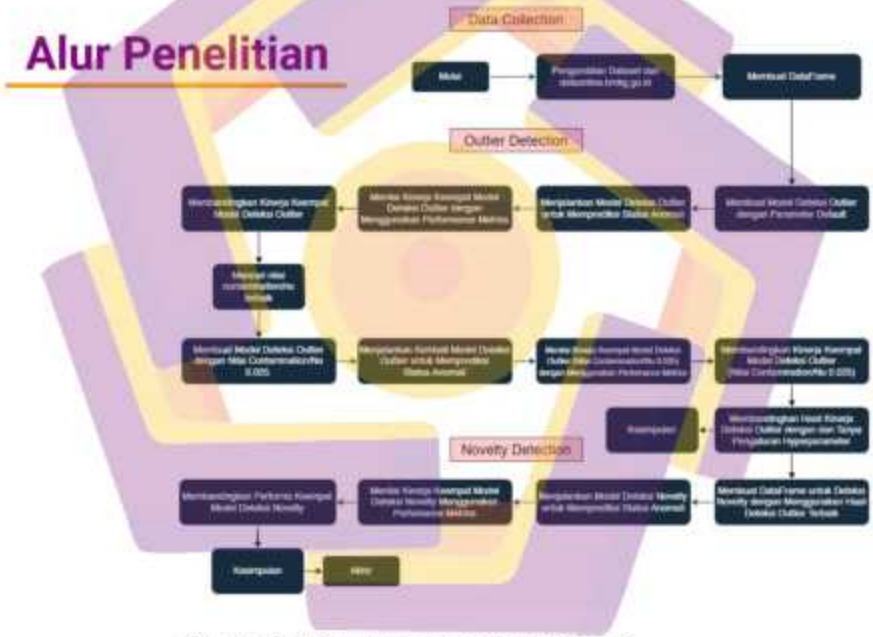
3.3. Metode Analisis Data

Dalam penelitian ini, data yang dianalisis menggunakan *pustaka Pandas* untuk memastikan tidak ada baris data yang memiliki sel kosong (NaN) dan data duplikat. Jika terdapat baris yang memiliki sel kosong dan data duplikat, maka baris tersebut akan dihapus dari dataset untuk menghindari pengaruh yang merugikan dalam proses deteksi anomali. Selain itu, pustaka Pandas juga digunakan untuk mengubah file *JSON* menjadi *dataframe*.

Sedangkan untuk mengevaluasi performa keempat metode deteksi anomali yang digunakan, digunakan beberapa metrik performa yang sering digunakan di bidang ini, yaitu *Precision, Recall, Specificity, Accuracy, AUC,* dan *F1-Score*

3.4. Alur Penelitian

Alur penelitian yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3. 2, berikut:



Gambar 3. 2 Diagram alur penelitian bagian 2

Berikut ini adalah penjelasan mengenai alur penelitian yang digambarkan pada Gambar 3.2, Gambar 3.3, dan Gambar 3.4. Alur penelitian tersebut telah dibagi menjadi beberapa poin dan beberapa proses alur penelitian dijelaskan dengan lebih rinci, seperti proses algoritma *Isolation Forest, Local Outlier Factor, One-Class SVM,* dan *Elliptic Envelope*. Penjelasan lebih terperinci dapat ditemukan pada bagian berikut ini:

3.4.1. Pendekatan Penelitian (Validasi Penelitian, Studi Literatur, Identifikasi Masalah, Pemilihan Metode)

Dalam penelitian ini, pendekatan yang digunakan adalah pendekatan penelitian yang dimulai dengan validasi penelitian. Validasi penelitian ini dilakukan dengan melakukan studi literatur untuk mengetahui kajian sebelumnya tentang deteksi anomali menggunakan *outlier* dan *novelty*. Studi literatur ini bertujuan untuk mengetahui kajian sebelumnya yang telah dilakukan oleh para peneliti sebelumnya dan untuk mengetahui metode-metode yang telah digunakan dalam mengatasi masalah deteksi anomali.

Setelah itu, identifikasi masalah dilakukan untuk mengetahui masalah yang ingin diteliti dalam penelitian ini. Identifikasi masalah ini bertujuan untuk mengetahui masalah yang ingin diteliti dan untuk mengetahui apa yang menjadi fokus dari penelitian ini.

Terakhir, pemilihan metode dilakukan untuk menentukan metode yang digunakan dalam penelitian ini. Pemilihan metode ini dilakukan dengan menggunakan metode-metode yang telah digunakan dalam kajian sebelumnya seperti *Elliptic Envelope, Isolation Forest, One-class SVM* dan *Local Outlier*

Factor. Pemilihan metode ini bertujuan untuk mengetahui metode yang paling sesuai untuk mengatasi masalah deteksi anomali yang ingin diteliti dalam penelitian ini.

Secara keseluruhan, pendekatan penelitian yang digunakan dalam penelitian ini adalah pendekatan penelitian yang dimulai dengan validasi penelitian yang meliputi studi literatur, identifikasi masalah, dan pemilihan metode. Pendekatan ini bertujuan untuk mengetahui kajian sebelumnya tentang deteksi anomali, mengetahui masalah yang ingin diteliti, dan menentukan metodeyang digunakan dalam penelitian ini.

3.4.2. Pengumpulan Data (Pemilihan Data Uji, Data Validasi)

Dalam penelitian ini, pengumpulan data dilakukan dengan mengambil data *API* gempa bumi yang terjadi di Indonesia melalui website resmi *Badan Meteorologi, Klimatologi, dan Geofisika (BMKG)* di dataonline.bmkg.go.id. Data yang digunakan sebagai dasar penelitian terdiri dari 12.351 kejadian gempa yang terjadi selama tahun 2021. Validasi data dilakukan dengan cara mengecek sel-sel yang kosong (*empty cell*) dan menghilangkan data yang duplikat menggunakan *pustaka pandas*, dengan tujuan agar deteksi anomali dapat dijalankan dengan baik.

3.4.3. Processing

Tahap selanjutnya dalam proses penelitian adalah pemrosesan data, dimana data siap untuk dianalisis menggunakan empat algoritma deteksi anomali, yaitu *Isolation Forest, Local Outlier Factor, One-Class SVM,* dan *Elliptic Envelope.* Setelah tahap deteksi anomali selesai, proses penelitian akan memeriksa apakah data tersebut dianggap sebagai anomali atau tidak. Jika data tersebut tidak dianggap

sebagai anomali, proses akan berakhir. Namun jika data tersebut dianggap sebagai anomali, proses penelitian akan melanjutkan untuk memeriksa apakah data anomali tersebut memiliki urutan temporal yang sesuai. Jika tidak, data anomali tersebut tidak dianggap sebagai *anomali kolektif*. Namun jika data anomali tersebut memiliki urutan temporal yang sesuai, maka data anomali tersebut dianggap sebagai *anomali kolektif*.

Anomali kolektif adalah suatu kondisi di mana terdapat beberapa data anomali yang terjadi secara bersamaan atau saling terkait dalam suatu jangka waktu tertentu. Anomali kolektif dapat dianggap sebagai suatu kejadian yang tidak biasa atau abnormal dibandingkan dengan situasi normal dari data yang dianalisis. Anomali kolektif dapat terjadi dalam berbagai jenis data, seperti data transaksi, data sensor, atau data log.

Contoh dari anomali kolektif adalah peningkatan jumlah transaksi kartu kredit yang tidak biasa dalam jangka waktu tertentu, atau peningkatan suhu dalam beberapa wilayah secara bersamaan yang menunjukkan adanya perubahan cuaca yang tidak biasa. Anomali kolektif dapat diidentifikasi dengan menggunakan teknik deteksi anomali.

Anomali kolektif dapat juga diterapkan dalam konteks data gempa bumi berdasarkan fitur kedalaman dan kekuatan gempa. Salah satu contohnya adalah jika terdapat serangkaian gempa bumi yang memiliki kedalaman yang sangat dangkal dan kekuatan yang sangat besar dalam jangka waktu tertentu di wilayah yang sama. Hal ini dapat dianggap sebagai anomali kolektif karena kondisi ini tidak biasa

dibandingkan dengan jumlah gempa bumi yang terjadi dalam jangka waktu yang sama dengan kedalaman dan kekuatan yang berbeda.

Sebagai contoh lain, jika terdapat serangkaian gempa bumi yang memiliki kedalaman yang sangat dalam dan kekuatan yang sangat rendah dalam jangka waktu tertentu di wilayah yang sama. Hal ini juga dapat dianggap sebagai anomali kolektif karena kondisi ini tidak biasa dibandingkan dengan jumlah gempa bumi yang terjadi dalam jangka waktu yang sama dengan kedalaman dan kekuatan yang berbeda.

Seperti yang disebutkan sebelumnya, anomali kolektif dapat diidentifikasi dengan menggunakan teknik deteksi anomali kolektif yang mengkombinasikan teknik deteksi anomali individu dengan analisis urutan temporal.

3.4.4. Analisis Perbandingan dan Performance metrics

Metode ini digunakan untuk menilai dan mengkaji performa dari beberapa metode yang digunakan dalam suatu penelitian. Dalam konteks ini, beberapa metode yang digunakan adalah *Isolation Forest, Local Outlier Factor, One-Class SVM,* dan *Elliptic Envelope*.

Metode-metode tersebut diuji dengan menggunakan *performance metrics* yang digunakan untuk mengevaluasi performa dari masing-masing metode yang digunakan. *Performance metrics* ini diperlukan untuk mengetahui performa dari masing-masing metode yang digunakan dan untuk mengetahui metode mana yang paling baik digunakan. Beberapa performance metrics yang digunakan dalam penelitian ini adalah *Precision, Recall (Sensitivity), Specificity, Accuracy, AUC,* dan *F1-Score*.

Precision mengukur seberapa baik suatu metode dalam mengidentifikasi benar positif. *Recall (Sensitivity)* mengukur seberapa baik suatu metode dalam mengidentifikasi benar negatif. Specificity mengukur seberapa baik suatu metode dalam mengidentifikasi benar negatif. Accuracy mengukur seberapa baik suatu metode dalam mengidentifikasi *benar positif* dan *benar negatif*. *AUC* mengukur seberapa baik suatu metode dalam mengidentifikasi benar positif dan benar negatif. *F1-Score* mengukur seberapa baik suatu metode dalam mengidentifikasi benar positif dan benar negatif.

Secara keseluruhan, analisis perbandingan digunakan untuk mengevaluasi dan membandingkan performa dari beberapa metode yang digunakan dalam suatu penelitian. Beberapa metode yang digunakan dalam konteks ini adalah *Isolation Forest, Local Outlier Factor, One-Class SVM, dan Elliptic Envelope*. Setelah metode-metode tersebut diuji, dilakukan performance metrics untuk mengevaluasi performa dari masing-masing metode yang digunakan. *Performance metrics* yang digunakan adalah *Precision, Recall (Sensitivity), Specificity, Accuracy, AUC*, dan *F1-Score*.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Gambaran Umum Penelitian

Dalam penelitian ini, dataset yang digunakan adalah data gempa bumi yang tercatat selama periode tahun 2021 di Indonesia. Data ini diambil dari website resmi Badan Meteorologi, Klimatologi, dan Geofisika (BMKG) melalui API BMKG dengan *endpoint* yang disesuaikan. Jumlah data yang diperoleh dari API BMKG sebanyak 12.351 data gempa. Data tersebut diberikan dalam bentuk file *json* dan terdiri dari lima kolom, yaitu datetime, lintang, bujur, kedalaman, dan magnitudo. Namun, dalam penelitian ini hanya menggunakan dua kolom, yaitu kedalaman dan magnitudo, sedangkan kolom lainnya hanya digunakan sebagai pelengkap. Setelah data dikumpulkan, file *json* tersebut diubah menjadi bentuk *csv* dengan menggunakan *library pandas* untuk memudahkan proses *import* ke dalam *Google Colab*.

Dalam mendeteksi gempa bumi yang dianggap anomali, peneliti menggunakan empat metode *anomaly detection*, yaitu: *Elliptic Envelope, Isolation Forest, One-Class SVM,* dan *Local Outlier Factor*. Keempat metode tersebut akan dijalankan dan diproses dengan menggunakan *library scikit-learn* di *Google Colab*.

Dalam membandingkan keempat metode untuk mendeteksi *outlier*, peneliti akan membuat beberapa simulasi atau pengujian yang melibatkan *parameter (tuning hyperparameter)* dan tanpa parameter (*default parameter*) pada setiap model. *Tuning hyperparameter* adalah istilah yang digunakan untuk meningkatkan

performa model *machine learning*. Proses ini mirip dengan proses tweak, misalnya pada komputer di mana kita mengganti komponen agar peranti tersebut memiliki performa lebih tinggi demi kinerja yang lebih efisien. Pada keempat model tersebut, jumlah contamination/nu yang kurang atau terlalu banyak akan menyebabkan hasil deteksi yang kurang optimal.

Pengujian pertama akan dilakukan dengan menggunakan nilai *parameter default*, di mana nilai parameter pada setiap *estimator (class)* akan diatur secara otomatis tanpa adanya *input argument*. Tujuan dari pengujian ini adalah untuk mengetahui metode terbaik dalam mendeteksi outlier dengan nilai *default parameter* masing-masing. Pengujian kedua akan dilakukan dengan menggunakan nilai parameter yang telah ditentukan secara manual khusus untuk parameter *contamination* atau *nu* masing-masing *estimator*. Contamination atau nu mewakili jumlah data yang akan dianggap sebagai *outlier*.

Setelah data diproses menggunakan keempat metode tersebut, selanjutnya akan dibandingkan masing-masing algoritma menggunakan *performance metrics*, yaitu *precision, recall (sensitivity), specificity, accuracy, AUC,* dan *F1-score.* Tujuan dari perbandingan ini adalah untuk mengetahui algoritma terbaik dalam melakukan deteksi anomali menggunakan outlier pada dataset gempa bumi.

Dasar perhitungan *performance metrics* adalah *confusion matrix*, misalnya pada *precision, recall, F1-score* dan lainnya. Mari asumsikan bahwa kita memiliki model deteksi gempa bumi anomali yang dapat mengidentifikasi kejadian gempa bumi yang tidak biasa berdasarkan fitur kedalaman dan magnitudo. Kita telah

menguji model ini dengan menggunakan data historis yang terdiri dari 1000 kejadian gempa bumi. Setelah menjalankan model deteksi anomali, kita mendapatkan hasil yang dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Contoh Confusion Matrix Anomaly Detection

	Prediksi Anomali	**Prediksi Normal**
Anomali Sebenarnya	200 (TP)	50 (FN)
Normal Sebenarnya	50 (FP)	700 (TN)

Dari Tabel 4.1 di atas, kita dapat menghitung tingkat akurasi sistem deteksi gempa bumi anomali kita dengan menggunakan rumus (TP + TN) / (TP + TN + FP + FN). Dengan menggunakan rumus tersebut, kita mendapatkan tingkat akurasi sebesar 87,75%.

Setelah semua pengujian *anomaly detection* dilakukan menggunakan *outlier*, maka akan dilakukan pengujian *anomaly detection* menggunakan *novelty*. Dataset yang digunakan adalah dataset hasil pengujian *outlier* terbaik dengan tambahan dua feature baru, yaitu *anomaly* dan *scores*, dengan jumlah dataset sebanyak 12.351. Dataset tersebut akan dibagi menjadi dua bagian, yaitu bagian *training* dan bagian *testing* dengan proporsi 80:20.

Dalam mengetahui algoritma terbaik dalam mendeteksi anomali menggunakan *novelty*, maka *performance matrix* yang digunakan adalah *matrix* yang sama dengan yang digunakan untuk mendeteksi anomali menggunakan *outlier*, yaitu *precision, recall, specificity, accuracy, AUC,* dan *F1-score*.

4.2. Pengumpulan Data

Dalam penelitian ini, data dikumpulkan menggunakan *Postman*, yaitu sebuah aplikasi yang memudahkan penggunaan *REST API*. Berikut adalah *endpoint* yang digunakan:

https://dataonline.bmkg.go.id/data_gempa_bumi/data?tanggal-min=01-01-2021&tanggal-max=31-12-2021&lintang-min=-11&lintang-max=6&bujur-min=-95&bujur-max=141&magnitude-min=1.00&magnitude-max=9.50&depth-min=1.00&depth-max=1000.00

Proses pengambilan dataset gempa bumi dan outputnya dapat diamati dalam ilustrasi Gambar 4.1 dan Gambar 4.2.

Gambar 4. 1 Proses pengambilan dataset gempa bumi

Gambar 4. 2 Output dataset gempa bumi

Output yang dihasilkan berupa file *json* tanpa nama kolom. Agar pembacaan data lebih mudah, file *json* tersebut diubah ke dalam bentuk csv dengan menghilangkan nama property "*recordsTotal*" dan "*data*".

4.3. Mencari nilai contamination/nu terbaik

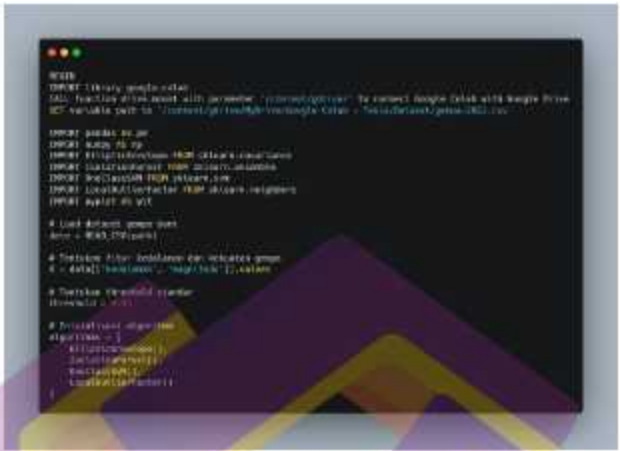
Sebelum mengimplementasikan algoritma, penting untuk menentukan nilai *contamination/nu* yang sesuai untuk *tuning hyperparameter*. Nilai *contamination/nu* yang tidak tepat akan mempengaruhi hasil deteksi dan menurunkan akurasi. Oleh karena itu, dengan menemukan nilai *contamination/nu* yang tepat, kita dapat memastikan bahwa algoritma bekerja dengan efisien dan memberikan hasil yang memenuhi standar. Perlu diketahui bahwa nilai contamination diterapkan pada algoritma Elliptic Envelope, Isolation Forest, dan Local Outlier Factor, sedangkan pada algoritma One-Class SVM menggunakan parameter nu.

Salah satu pendekatan untuk menentukan nilai *contamination/nu* terbaik adalah dengan menggunakan *threshold*. Threshold adalah batas yang digunakan

untuk membandingkan hasil deteksi dengan nilai referensi. Dalam hal deteksi status anomali pada dataset gempa bumi, *threshold* standar sering digunakan sebagai acuan, seperti 0.05. Jika jumlah status anomali melebihi *threshold* standar, maka perlu dilakukan tindakan seperti memvalidasi ulang atau melakukan analisis lebih lanjut.

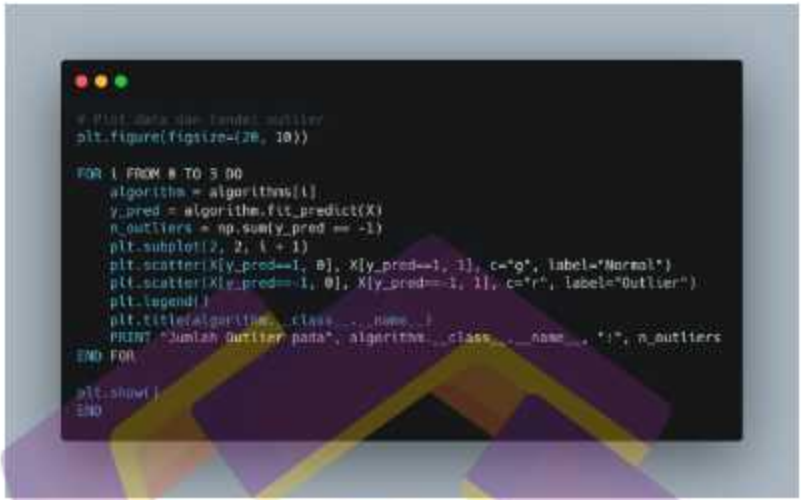
Penggunaan *threshold* dalam menentukan nilai *contamination/nu* terbaik merupakan salah satu pendekatan yang dapat memastikan algoritma bekerja secara efisien dan memberikan hasil yang memenuhi standar. *Threshold* memberikan batas yang jelas bagi algoritma dalam menentukan apakah hasil deteksi dianggap sebagai anomali atau tidak. Salah satu pendekatan yang dapat digunakan adalah dengan menetapkan nilai *threshold* sedemikian rupa sehingga proporsi skor yang di atasnya sama dengan faktor kontaminasi γ pada dataset, yaitu proporsi anomali yang diharapkan. Dalam hal ini, penyesuaian nilai *contamination/nu* dapat dilakukan jika jumlah hasil deteksi anomali melebihi *threshold* untuk memastikan bahwa algoritma bekerja dengan benar dan hasil deteksi sesuai dengan kebutuhan. Biasanya, faktor kontaminasi diasumsikan diberikan oleh ahli domain, tetapi pada sebagian besar skenario dunia nyata, faktor ini sebenarnya tidak diketahui. (Perini et al., 2022)

Berikut adalah pseudocode yang digunakan untuk menentukan nilai *contamination/nu* terbaik dengan menggunakan teknik *threshold* seperti yang terlihat pada Gambar 4.3, 4.4 dan 4.5.



Gambar 4. 3 Pseudocode untuk menentukan nilai contamination/nu terbaik (1)

Gambar 4. 4 Pseudocode untuk menentukan nilai contamination/nu terbaik (2)



```
# Plot data dan tandai outlier
plt.figure(figsize=(20, 10))

FOR i FROM 0 TO 5 DO
    algorithm = algorithms[i]
    y_pred = algorithm.fit_predict(X)
    n_outliers = np.sum(y_pred == -1)
    plt.subplot(2, 3, i + 1)
    plt.scatter(X[y_pred==1, 0], X[y_pred==1, 1], c="g", label="Normal")
    plt.scatter(X[y_pred==-1, 0], X[y_pred==-1, 1], c="r", label="Outlier")
    plt.legend()
    plt.title(algorithm.__class__.__name__)
    PRINT "Jumlah Outlier pada", algorithm.__class__.__name__, ":", n_outliers
END FOR

plt.show()
END
```

Gambar 4.5 Pseudocode untuk menentukan nilai contamination/nu terbaik (3)

Berikut adalah penjelasan masing-masing variabel

- Variabel `**drive**` merupakan objek dari modul `**google.colab**` yang digunakan untuk menghubungkan Google Colab dengan Google Drive.
- Variabel `**path**` digunakan untuk menyimpan alamat file CSV yang akan dibuka. Pada pseudocode di atas, variabel `**path**` akan berisi string yang merupakan alamat file CSV `**gempa-2021.csv**` yang disimpan di Google Drive. Alamat file tersebut dapat diakses melalui folder `**Google Colab - Tesis**` di Google Drive.
- `**drive.mount('/content/gdrive/')**` digunakan untuk menghubungkan Google Colab dengan Google Drive. `**/content/gdrive/**` merupakan direktori default yang digunakan oleh Google Colab untuk mengakses Google Drive. Setelah baris ini dieksekusi, pengguna akan diminta untuk memasukkan kode otorisasi untuk mengakses Google Drive.

- `**from google.colab import drive**` digunakan untuk mengimpor modul `**drive**` dari paket `**google.colab**`. `**drive**` memungkinkan kita untuk mengakses Google Drive pada Google Colab.
- Variabel `**pd**` merupakan alias untuk modul `**pandas**`. Modul `**pandas**` adalah salah satu modul yang paling sering digunakan dalam pengolahan data di Python. Modul ini menyediakan struktur data dan fungsi untuk memanipulasi data yang sangat berguna dalam analisis data.
- Variabel `**np**` merupakan alias untuk modul `**numpy**`. Modul ini menyediakan objek array dan fungsi matematika yang sangat berguna dalam ilmu data dan analisis numerik.
- Variabel `**EllipticEnvelope**` digunakan untuk mengimplementasikan deteksi outlier berbasis kovarian menggunakan metode ellipsoidal. Modul ini terdapat pada paket `**sklearn.covariance**`.
- Variabel `**IsolationForest**` digunakan untuk mengimplementasikan deteksi outlier menggunakan metode Isolation Forest. Modul ini terdapat pada paket `**sklearn.ensemble**`.
- Variabel `**OneClassSVM**` digunakan untuk mengimplementasikan deteksi outlier menggunakan metode One-Class Support Vector Machine (SVM). Modul ini terdapat pada paket `**sklearn.svm**`.
- Variabel `**LocalOutlierFactor**` digunakan untuk mengimplementasikan deteksi outlier menggunakan metode Local Outlier Factor (LOF). Modul ini terdapat pada paket `**sklearn.neighbors**`.

- Variabel '**plt**' merupakan alias untuk modul '**matplotlib.pyplot**'. Modul '**matplotlib.pyplot**' digunakan untuk membuat visualisasi dari data menggunakan grafik.
- Variabel '**data**' berisi data gempa bumi yang telah dimuat menggunakan metode '**pd.read_csv(path)**'. Variabel '**path**' menyimpan alamat file CSV yang akan dibuka dan sudah dijelaskan pada pertanyaan sebelumnya.
- Variabel '**X**' merupakan variabel yang berisi data dari dua fitur gempa bumi yaitu kedalaman dan magnitudo. Variabel ini akan digunakan sebagai data input untuk deteksi outlier.
- Variabel '**threshold**' merupakan threshold standar yang digunakan dalam deteksi outlier. Threshold ini digunakan untuk menentukan batas yang menjadi acuan dalam menentukan apakah sebuah data termasuk outlier atau tidak.
- Variabel '**algorithms**' merupakan variabel yang berisi daftar algoritma deteksi outlier yang akan digunakan. Pada kode di atas, terdapat empat algoritma yang digunakan yaitu '**EllipticEnvelope**', '**IsolationForest**', '**OneClassSVM**', dan '**LocalOutlierFactor**'. Algoritma-algoritma ini merupakan implementasi dari deteksi outlier yang terdapat pada paket '**sklearn**'.
- Pada pseudocode di atas, terdapat perulangan untuk setiap algoritma deteksi outlier yang ada dalam variabel '**algorithms**'. Di dalam perulangan, terdapat dua kemungkinan kasus yaitu untuk algoritma '**OneClassSVM**' dan untuk algoritma selain '**OneClassSVM**'.

- Jika algoritma yang sedang diproses adalah '**OneClassSVM**', maka akan dilakukan perulangan untuk setiap nilai '**nu**' dalam list '**[0.025, 0.01, 0.02, 0.03, 0.04, 0.05]**'. Nilai '**nu**' akan di-set ke algoritma '**OneClassSVM**' dan dilakukan perhitungan skor outlier dengan memanggil metode '**fit_predict(X)**' pada algoritma tersebut. Kemudian, jumlah outlier dihitung dan jika jumlah outlier sesuai dengan threshold, maka nilai '**nu**' yang cocok akan ditampilkan dan perulangan dihentikan dengan menggunakan pernyataan '**break**'.
- Jika algoritma yang sedang diproses adalah selain '**OneClassSVM**', maka akan dilakukan perulangan untuk setiap nilai '**contamination**' dalam list '**[0.025, 0.01, 0.02, 0.03, 0.04, 0.05]**'. Nilai '**contamination**' akan di-set ke algoritma tersebut dan dilakukan perhitungan skor outlier dengan memanggil metode '**fit_predict(X)**' pada algoritma tersebut. Kemudian, jumlah outlier dihitung dan jika jumlah outlier sesuai dengan threshold, maka nilai '**contamination**' yang cocok akan ditampilkan dan perulangan dihentikan dengan menggunakan pernyataan '**break**'.
- '**plt**': alias dari modul '**matplotlib.pyplot**', yang menyediakan fungsi untuk membuat dan menyesuaikan plot.
- '**X**': sebuah array NumPy yang berisi nilai fitur '**kedalaman**' dan '**magnitudo**' dari data gempa bumi.
- '**algorithms**': sebuah daftar algoritma deteksi outlier yang akan digunakan untuk mendeteksi outlier pada data gempa bumi.

- `**y_pred**`: sebuah array NumPy yang berisi label prediksi untuk setiap titik data di `**X**`, di mana nilai `**-1**` menunjukkan outlier dan nilai `**1**` menunjukkan titik data normal.
- `**n_outliers**`: jumlah outlier yang terdeteksi oleh algoritma deteksi outlier.
- `**i**`: sebuah indeks yang digunakan untuk menentukan posisi plot dalam grid subplot.
- `**subplot()**`: sebuah fungsi dari modul `**matplotlib.pyplot**` yang digunakan untuk membuat grid subplot.
- `**scatter()**`: sebuah fungsi dari modul `**matplotlib.pyplot**` yang digunakan untuk membuat plot sebaran data gempa bumi, di mana titik data normal digambarkan dengan warna hijau dan outlier digambarkan dengan warna merah.
- `**legend()**`: sebuah fungsi dari modul `**matplotlib.pyplot**` yang digunakan untuk menambahkan legenda pada plot.
- `**title()**`: sebuah fungsi dari modul `**matplotlib.pyplot**` yang digunakan untuk mengatur judul dari plot.

Penjelasan di atas didasarkan pada masing-masing variabel. Berikut ini adalah penjelasan terkait pseudocode yang telah disebutkan sebelumnya. Pseudocode tersebut menggunakan empat algoritma deteksi outlier, yaitu EllipticEnvelope, IsolationForest, OneClassSVM, dan LocalOutlierFactor. Tujuan dari pseudocode ini adalah untuk menentukan nilai *contamination* atau *nu* yang sesuai untuk setiap algoritma. *Contamination* dan *Nu* diartikan sebagai persentase jumlah data yang diperkirakan sebagai *outlier*. Proses pemilihan nilai

contamination atau *nu* yang sesuai dilakukan melalui *looping* dengan menggunakan enam nilai yang berbeda. Setiap nilai *contamination* atau *nu* akan dipakai untuk melatih algoritma dan memprediksi *outlier*. Kemudian, jumlah outlier yang dihasilkan akan dibandingkan dengan *threshold* yang telah ditentukan sebelumnya. Bila jumlah *outlier* kurang dari atau sama dengan *threshold*, maka nilai *contamination* atau *nu* yang digunakan akan dicantumkan dan *looping* akan berakhir. Setelah nilai *contamination* atau *nu* yang sesuai untuk masing-masing algoritma ditemukan, data akan diproses dan ditandai sebagai *outlier* atau tidak. Hasil akan ditampilkan dalam empat *subplot* yang berbeda, masing-masing untuk setiap algoritma.

Threshold digunakan untuk menentukan batas jumlah *outlier* yang diizinkan dalam data. Dalam koding tersebut, *threshold* sebesar 0.05 digunakan untuk memastikan bahwa jumlah *outlier* dalam data tidak melebihi 5% dari total data. Hal ini dilakukan agar jumlah *outlier* yang diidentifikasi tidak terlalu banyak dan tetap relevan dengan jumlah data yang ada.

Data yang teridentifikasi sebagai *outlier* akan ditampilkan dengan warna merah, sedangkan data yang normal akan ditampilkan dengan warna hijau.

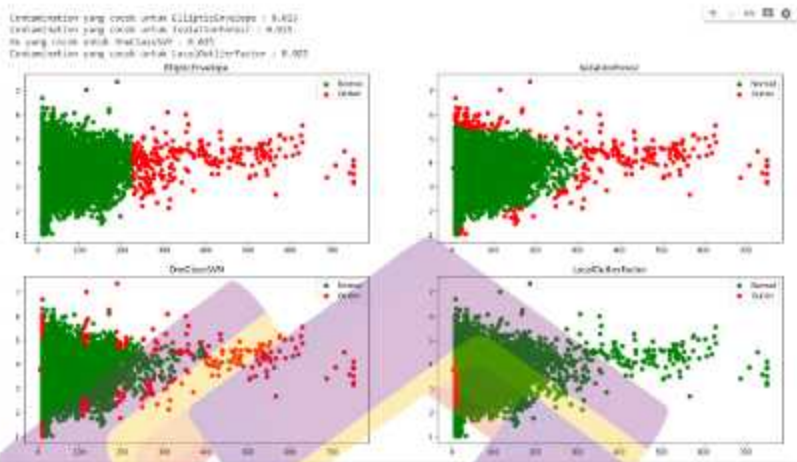
Berikut hasil *Contamination/Nu* terbaik yang diperoleh

Contamination yang cocok untuk EllipticEnvelope : 0.025

Contamination yang cocok untuk IsolationForest : 0.025

Nu yang cocok untuk OneClassSVM : 0.025

Contamination yang cocok untuk LocalOutlierFactor : 0.025



Gambar 4. 6 Hasil visualisasi outlier dengan contamination/nu terbaik

Oleh karena itu, dapat disimpulkan bahwa nilai parameter *Contamination/Nu* yang akan digunakan sebagai *tuning hyperparameter* adalah sebesar 0.025 atau 2,5% dari jumlah total dataset.

Berikut penjelasan mengapa nilai Contamination/Nu yang terpilih adalah 0.025, Nilai Contamination/Nu yang terpilih adalah 0.025 untuk semua metode deteksi outlier yang digunakan, yaitu EllipticEnvelope, IsolationForest, OneClassSVM, dan LocalOutlierFactor. Hal ini menunjukkan bahwa 2.5% dari total data dianggap sebagai outlier. Pemilihan nilai Contamination/Nu dilakukan dengan tujuan untuk memastikan bahwa jumlah outlier yang diidentifikasi tidak terlalu banyak dan tetap relevan dengan jumlah data yang ada. Jika nilai Contamination/Nu terlalu rendah, maka outlier yang seharusnya diidentifikasi sebagai outlier mungkin tidak terdeteksi, sedangkan jika nilai terlalu tinggi, maka banyak data normal yang akan salah teridentifikasi sebagai outlier. Dalam hal ini, karena threshold sebesar 0.05 digunakan untuk memastikan bahwa jumlah outlier

dalam data tidak melebihi 5% dari total data, maka nilai Contamination/Nu yang dipilih sebesar 0.025 cukup relevan dengan batasan tersebut. Dengan menggunakan nilai tersebut, jumlah outlier yang diidentifikasi tidak akan melebihi 2.5% dari total data, sehingga tetap relevan dengan threshold yang telah ditentukan.

4.4. Implementasi Algoritma Deteksi Anomali Menggunakan Metode Outlier Detection

Pada penerapan algoritma deteksi anomali yang pertama, akan dilakukan penggunaan metode *outlier* untuk mendeteksi gempa yang tidak biasa yang terdapat dalam dataset. Gempa yang tidak biasa dapat didefinisikan sebagai gempa yang memiliki karakteristik yang berbeda dari gempa lainnya berdasarkan atribut kedalaman dan magnitudonya.

4.4.1. Pengenalan dan Penggunaan Google Colab

Dalam penelitian ini, peneliti menggunakan *Google Colab* untuk mengimplementasikan empat algoritma. Oleh karena itu, langkah pertama yang perlu dilakukan adalah menyiapkan *Google Colab* agar dapat menjalankan algoritma. Langkah-langkahnya adalah sebagai berikut:

Buka Google Colab di alamat *https://colab.research.google.com*, lalu pilih menu File > New Notebook. Selanjutnya, terapkan *mounting* pada *Google Drive* agar dapat diakses melalui *Google Colab* menggunakan perintah seperti yang diilustrasikan dalam gambar 4.7

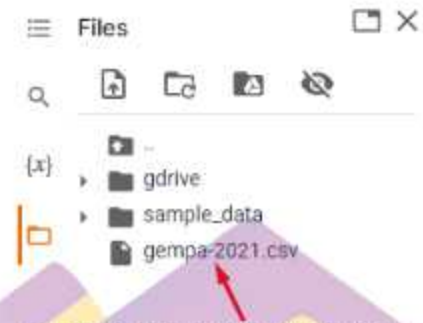
Gambar 4. 7 Proses mounting Google Drive

Pilih opsi *Connect to Google Drive* seperti yang diilustrasikan dalam gambar 4.8, lalu pilih salah satu akun *Google* yang akan digunakan dan izinkan seperti yang diilustrasikan dalam gambar 4.9.

Gambar 4. 8 Proses menghubungkan ke akun Google Drive

Gambar 4. 9 Memilih akun Google

Agar *dataset* dapat diakses, selanjutnya kita perlu mengunggah file *dataset* yang akan digunakan untuk menjalankan algoritma seperti yang diilustrasikan dalam gambar 4.10.



Gambar 4. 10 Proses unggah file dataset

4.4.2. Implementasi Library dalam Penerapan Outlier Detection

Setelah file *dataset* ter*upload*, langkah selanjutnya adalah memanggil *library* yang dibutuhkan untuk proses selanjutnya, seperti yang diilustrasikan dalam Gambar 4.11.

```
# Import Library

import pandas as pd
from sklearn.covariance import EllipticEnvelope
from sklearn.ensemble import IsolationForest
from sklearn.svm import OneClassSVM
from sklearn.neighbors import LocalOutlierFactor
from matplotlib import rcParams
import matplotlib.pyplot as plt
```

Gambar 4. 11 Mengimport library

Library yang pertama yang perlu *diimport* adalah *pandas*, yang digunakan untuk membuat struktur data yang disebut *dataframe* yang strukturnya mirip dengan tabel, terdiri dari baris dan kolom. Kemudian, kita memanggil empat algoritma deteksi anomali, yaitu *EllipticEnvelope, IsolationForest, OneClassSVM,* dan *LocalOutlierFactor*. Keempat algoritma ini berasal dari *library* yang sama, yaitu

Scikit-Learn. *Library* terakhir yang perlu diimport adalah *matplotlib*, yang digunakan untuk membuat *plot visualisasi outlier*.

4.4.3. Pembuatan DataFrame untuk Outlier Detection

DataFrame merupakan struktur data yang banyak digunakan dalam proses analisis data. Struktur data ini mirip dengan tabel, dengan baris dan kolom. *DataFrame* dapat dibuat dari berbagai macam sumber data, seperti *list*, *dictionary*, *array numpy*, atau file eksternal seperti *csv* dan *excel*.

Dalam penerapan algoritma deteksi anomali ini, *library pandas* digunakan untuk membuat *DataFrame* dengan sumber data berasal dari file *CSV*. Urutan *feature* ditentukan dengan menggunakan *parameter columns* seperti yang diilustrasikan dalam gambar 4.12.

```
# Membuat dataFrame gempa

gempa2021 = pd.read_csv('gempa-2021.csv')
df = pd.DataFrame(gempa2021, columns = ['datetime', 'lintang', 'bujur', 'kedalaman', 'magnitudo'])
df.head()
```

	datetime	lintang	bujur	kedalaman	magnitudo
0	2021-01-01 23:24:25	0.30	126.61842	57.9	3.06
1	2021-01-01 20:35:22	-9.25	119.44214	29.3	2.55
2	2021-01-01 11:32:18	2.25	127.28616	37.9	3.77
3	2021-01-01 11:13:23	-7.58	128.66589	197.3	4.21
4	2021-01-01 08:57:43	-8.90	119.03639	68.7	2.77

Gambar 4. 12 Membuat dataframe gempa

Setelah *DataFrame* berhasil dibuat, kita dapat melakukan analisis data dengan menggunakan fungsi-fungsi yang tersedia di *pandas*. Misalnya, menggunakan method *head()* untuk menampilkan lima baris data teratas dari *DataFrame*.

4.4.4. Pembuatan Model Deteksi Anomali untuk Outlier Detection

Setelah *DataFrame* dapat menampilkan data, langkah selanjutnya adalah membuat *model* untuk keempat algoritma. *Model* merupakan objek yang mengandung fungsi-fungsi untuk melakukan deteksi anomali, dan dibuat dengan menggunakan *DataFrame* yang telah dibuat sebelumnya.

Dalam proses pembuatan *model* untuk keempat algoritma, masing-masing algoritma akan menggunakan *class* yang berbeda. Misalnya, *Elliptic Envelope* menggunakan class *EllipticEnvelope*, algoritma *Isolation Forest* menggunakan class *IsolationForest*, algoritma *Local Outlier Factor* menggunakan class *LocalOutlierFactor*, dan algoritma *One Class SVM* menggunakan class *OneClassSVM*.

Namun dalam pembahasan ini, hanya akan dibahas algoritma *Elliptic Envelope* dengan menggunakan *parameter default* dan *tuning parameter*, dengan tujuan untuk menyederhanakan pembahasan. Karena keempat algoritma memiliki langkah-langkah yang sama dalam melakukan deteksi anomali menggunakan *outlier*. *Tuning parameter* dilakukan untuk menentukan parameter dengan nilai terbaik agar mendapatkan hasil yang optimal. *Parameter* yang digunakan dalam algoritma *Elliptic Envelope*, *Isolation Forest* dan *Local Outlier Factor* adalah *contamination* sebesar 0.025 atau 2,5% dari keseluruhan data, sedangkan pada algoritma *One-Class SVM* sparameter yang digunakan adalah nu bukan contamination. Proses pembuatan model Elliptic Envelope dapat diamati dalam ilustrasi Gambar 4.13.

```
# Membuat model

elliptic_envelope = EllipticEnvelope()

# Membuat model tuning parameter (contamination 0.025)

elliptic_envelope_contamination = EllipticEnvelope(contamination=0.025)
```

Gambar 4. 13 Membuat model elliptic envelope

4.4.5. Melakukan Deteksi Anomali dan Menampilkan Skor Hasil Deteksi Outlier

Jika tidak terjadi *error* saat pembuatan *model*, kita dapat melanjutkan untuk menampilkan hasil deteksi anomali beserta skor-skornya. Untuk mendeteksi anomali, kita menggunakan fungsi *fit_predict()* pada masing-masing *model* yang telah dibuat. Untuk menampilkan skor, kita dapat menggunakan fungsi *decision_function()*. Hasil deteksi anomali dan skor dapat dilihat pada Gambar 4.14.

```
# DataFrame khusus model Elliptic Envelope
df_ee = df.copy(deep = True)

# DataFrame khusus model Elliptic Envelope (Contamination)
df_eec = df.copy(deep = True)
```

Gambar 4. 14 Memisahkan dataframe

Agar *DataFrame* untuk *model EllipticEnvelope* tanpa parameter dan model *EllipticEnvelope* dengan *tuning parameter* dapat dibandingkan, kita harus memisahkan hasil deteksi anomali dan skor-skornya. Untuk memisahkan hasil deteksi anomali dan skor-skornya, kita dapat membuat dua object *DataFrame* baru seperti yang diilustrasikan dalam Gambar 4.15.

```
# Menambahkan kolom anomaly dan scores ke DataFrame khusus model Elliptic Envelope

df_ee['anomaly'] = elliptic_envelope.fit_predict(df_ee[['kedalaman', 'magnitudo']].values)
df_ee['scores'] = elliptic_envelope.decision_function(df_ee[['kedalaman', 'magnitudo']].values)

# Menambahkan kolom anomaly dan scores ke DataFrame khusus model Elliptic Envelope (Contamination)

df_eec['anomaly'] = elliptic_envelope_contamination.fit_predict(df_eec[['kedalaman', 'magnitudo']].values)
df_eec['scores'] = elliptic_envelope_contamination.decision_function(df_eec[['kedalaman', 'magnitudo']].values)
```

Gambar 4. 15 Menambahkan kolom anomaly dan scores

4.4.6. Filter Data Anomali dalam Outlier Detection

Setelah menambahkan kolom hasil deteksi *anomaly* dan *scores* ke *DataFrame* masing-masing, kita perlu memfilter kolom *anomaly* yang akan digunakan untuk menampilkan hasil deteksi dalam bentuk grafik. Data yang dianggap anomali memiliki nilai -1 dan data normal memiliki nilai 1. Untuk memfilter kolom *anomaly*, kita dapat menggunakan *property loc* yang tersedia di *pandas*. Cara menggunakan *property loc* dan hasilnya dapat diamati dalam ilustrasi Gambar 4.16 dan Gambar 4.17.

```
# Memfilter data anomali khusus khusus model Elliptic Envelope

anomaly_ee = df_ee.loc[df_ee['anomaly']==-1]
anomaly_ee
```

	datetime	lintang	bujur	kedalaman	magnitudo	anomaly	scores
3	2021-01-01 11:13:23	-7.58	128.66589	197.3	4.21	-1	-13628.269067
38	2021-01-02 11:53:47	2.80	128.39656	208.2	3.04	-1	-16484.060081
40	2021-01-02 10:41:23	-7.10	129.42459	211.4	4.47	-1	-17365.579121
41	2021-01-02 08:03:52	-7.18	125.16638	459.9	4.07	-1	-127763.633148
43	2021-01-02 06:33:23	-4.31	138.31104	159.3	4.78	-1	-4919.618605
...	...	...	...	...	...	...	...
12324	2021-12-30 08:17:35	-5.30	130.22702	185.2	4.06	-1	-10638.512080
12331	2021-12-31 20:16:21	-7.64	127.71794	139.1	4.12	-1	-1085.570457
12332	2021-12-31 20:05:00	-6.33	128.39857	371.2	4.34	-1	-78677.511971
12333	2021-12-31 19:07:22	-7.47	123.79356	539.3	4.45	-1	-180830.603920
12337	2021-12-31 08:17:44	0.05	123.57321	141.0	3.53	-1	-1418.715147

1234 rows × 7 columns

Gambar 4. 16 Memfilter Data Anomali Elliptic Envelope

```
# Memfilter data anomali khusus khusus model Elliptic Envelope (Contamination)

anomaly_eec = df_eec.loc[df_eec['anomaly']==-1]
anomaly_eec
```

	datetime	lintang	bujur	kedalaman	magnitudo	anomaly	scores
41	2021-01-02 08:03:52	-7.18	125.16638	459.9	4.07	-1	-106238.229441
47	2021-01-02 01:07:12	-8.39	114.85698	284.1	2.32	-1	-19392.123482
52	2021-01-02 00:02:02	-4.48	126.11481	398.2	4.32	-1	-70956.736263
56	2021-01-02 01:07:12	-8.39	114.85698	284.1	2.32	-1	-19392.123482
61	2021-01-02 00:02:02	-4.48	126.11481	398.2	4.32	-1	-70956.736263
...	...	...	...	...	...	...	...
12107	2021-12-23 00:01:31	5.68	123.95106	496.6	4.61	-1	-129702.083625
12137	2021-12-25 23:39:14	-6.36	128.07504	339.2	4.43	-1	-42081.597710
12171	2021-12-26 17:50:24	-8.22	117.82336	245.0	5.02	-1	-5824.007434
12332	2021-12-31 20:05:00	-6.33	128.39857	371.2	4.34	-1	-57152.108263
12333	2021-12-31 19:07:22	-7.47	123.79356	539.3	4.45	-1	-139305.200212

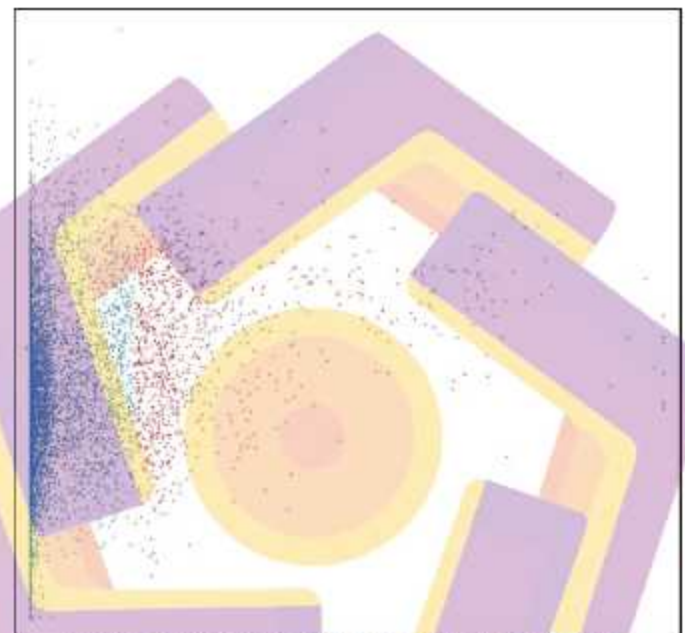
309 rows × 7 columns

Gambar 4. 17 Memfilter Data Anomali Elliptic Envelope (with contamination)

4.4.7. Pembuatan Visualisasi untuk Outlier Detection

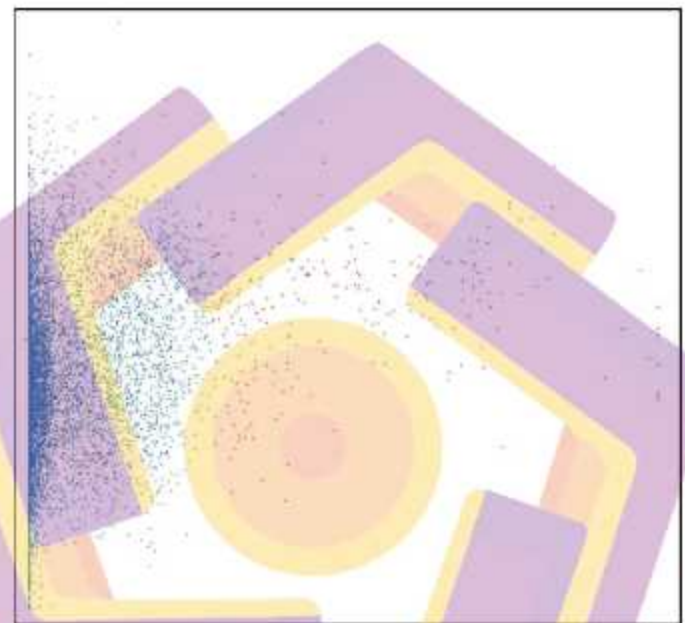
Agar mudah melihat perbandingan hasil deteksi anomali antara model *EllipticEnvelope* tanpa *tuning parameter* dan *model EllipticEnvelope* dengan *tuning parameter*, kita dapat menampilkan hasil deteksi anomali dalam bentuk grafik. Untuk menampilkan hasil deteksi anomali dalam bentuk grafik, kita dapat menggunakan fungsi *scatter()* yang tersedia di *matplotlib*. Cara menggunakan fungsi *scatter()* dan hasilnya dapat diamati dalam ilustrasi Gambar 4.18 dan Gambar 4.19.

```
# Memvisualisasikan/plot anomali Elliptic Envelope
# figure size in inches
rcParams['figure.figsize'] = 50.0, 50.0

# visualize outputs
plt.scatter(df_ee["kedalaman"], df_ee["magnitudo"])
plt.scatter(anomaly_ee["kedalaman"], anomaly_ee["magnitudo"], c = "r")
plt.show()
```



Gambar 4. 18 Visualisasi Data Anomali Elliptic Envelope

```
# Memvisualisasikan/plot anomali Elliptic Envelope (Contamination)
# figure size in inches
rcParams['figure.figsize'] = 50.0, 50.0

# visualize outputs
plt.scatter(df_eec["kedalaman"], df_eec["magnitudo"])
plt.scatter(anomaly_eec["kedalaman"], anomaly_eec["magnitudo"], c = "r")
plt.show()
```



Gambar 4. 19 Visualisasi Data Anomali Elliptic Envelope (with contamination)

Titik warna biru pada *grafik scatter* yang ada pada Gambar 4.18 dan Gambar 4.19 merupakan data normal, sementara titik warna merah merupakan data anomali. Dapat dilihat bahwa pada model *EllipticEnvelope* dengan *tuning parameter*, jumlah titik berwarna merah lebih sedikit dibandingkan dengan titik berwarna biru. Hal ini dikarenakan nilai contamination yang diatur hanya sebesar 0.025 atau 2,5% dari keseluruhan data.

4.4.8. Analisis Performa Model Elliptic Envelope dengan Confusion Matrix dalam Outlier Detection

Confusion matrix adalah salah satu metode yang digunakan untuk mengukur performa model yang telah dibuat. *Confusion matrix* menunjukkan jumlah data yang benar dan jumlah data yang salah. Confusion matrix terdiri dari empat nilai, yaitu *True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN).*

Untuk menghitung *confusion matrix* secara manual, kita dapat menggunakan perhitungan berdasarkan definisi *True Positive (TP), True Negative (TN), False Positive (FP),* dan *False Negative (FN)* seperti yang telah dijelaskan sebelumnya. Hasil *confusion matrix* algoritma *Elliptic Envelope* tanpa parameter akan ditampilkan pada Gambar 4.20 dan Gambar 4.21, sedangkan hasil *Confusion matrix* algoritma *Elliptic Envelope* dengan *tuning parameter* akan ditampilkan pada Gambar 4.22.

```
jumlah_anomali_asli = 312
jumlah_normal_asli = 12039
jumlah_normal_execute = normal.shape[0] - jumlah_normal_asli
total_dataset = jumlah_anomali_asli + jumlah_normal_asli

# TP = Kejadian sebenarnya adalah anomali, dan diprediksi sebagai anomali
TP = 58

# TN = Kejadian sebenarnya adalah normal, dan diprediksi seperti normal
TN = normal.shape[0]

# FN = Kejadian sebenarnya adalah anomali, tetapi diprediksi seperti normal
FN = jumlah_anomali_asli - TP

# FP = Kejadian sebenarnya adalah normal, tetapi diprediksi sebagai anomali
FP = len(anomaly) - TP
```

Gambar 4. 20 Hasil confusion matrix elliptic envelope (1)

```
print(f"TP: {TP}")
print(f"TN: {TN}")
print(f"FN: {FN}")
print(f"FP: {FP}")
print(f"(TP+TN+FN+FP)-FN: {(TP+TN+FN+FP)-FN}\n")
```

```
TP: 58
TN: 11117
FN: 254
FP: 1176
(TP+TN+FN+FP)-FN: 12351
```

Gambar 4. 21 Hasil confusion matrix elliptic envelope (2)

```
jumlah_anomali_asli = 312
jumlah_normal_asli = 12039
jumlah_normal_execute = normal.shape[0] - jumlah_normal_asli
total_dataset = jumlah_anomali_asli + jumlah_normal_asli

# TP = Kejadian sebenarnya adalah anomali, dan diprediksi sebagai anomali
TP = 34

# TN = Kejadian sebenarnya adalah normal, dan diprediksi seperti normal
TN = normal.shape[0]

# FN = Kejadian sebenarnya adalah anomali, tetapi diprediksi seperti normal
FN = jumlah_anomali_asli - TP

# FP = Kejadian sebenarnya adalah normal, tetapi diprediksi sebagai anomali
FP = len(anomaly) - TP
```

```
print(f"TP: {TP}")
print(f"TN: {TN}")
print(f"FN: {FN}")
print(f"FP: {FP}")
print(f"(TP+TN+FN+FP)-FN: {(TP+TN+FN+FP)-FN}\n")
```

```
TP: 34
TN: 12042
FN: 278
FP: 275
(TP+TN+FN+FP)-FN: 12351
```

Gambar 4. 22 Hasil confusion matrix elliptic envelope (tuning parameter)

4.4.9. Penghitungan Metrics Performa Model Elliptic Envelope dalam Outlier Detection

Berikut adalah proses menghitung performance metrics seperti pada Gambar 4.23 dan Gambar 4.24

```
precision = TP/(TP + FP)
print(f"Precision: {precision}")

recall = TP/(TP + FN)
print(f"Recall: {recall}")

specificity = TN/(TN + FP)
print(f"Specificity: {specificity}")

accuracy = (TP + TN)/((TP+TN+FN+FP)-FN)
print(f"Accuracy: {accuracy}")

AUC = ((TP/(TP+FN)) + (TN/(TN+FP)))/2
print(f"AUC: {AUC}")

f1_score = 2*(recall*precision)/(precision+recall)
print(f"F1_Score: {f1_score}")
```

```
Precision: 0.04700162074554295
Recall: 0.1858974358974359
Specificity: 0.9043358068622784
Accuracy: 0.9047850376487734
AUC: 0.5451166183798577
F1_Score: 0.07503254152652005
```

Gambar 4. 23 Performance metrics elliptic envelope

```
precision = TP/(TP + FP)
print(f"Precision: {precision}")

recall = TP/(TP + FN)
print(f"Recall: {recall}")

specificity = TN/(TN + FP)
print(f"Specificity: {specificity}")

accuracy = (TP + TN)/((TP+TN+FN+FP)-FN)
print(f"Accuracy: {accuracy}")

AUC = ((TP/(TP+FN)) + (TN/(TN+FP)))/2
print(f"AUC: {AUC}")

f1_score = 2*(recall*precision)/(precision+recall)
print(f"F1_Score: {f1_score}")
```

```
Precision: 0.11003236245954692
Recall: 0.10897435897435898
Specificity: 0.9776731340918892
Accuracy: 0.9777345963889563
AUC: 0.5433237468331241
F1_Score: 0.10950080515297907
```

Gambar 4. 24 Performance metrics elliptic envelope (tuning parameter)

Hasil performance metrics dari algoritma lainnya dapat dilihat pada Tabel 4.2 dan Tabel 4.3

4.5. Implementasi Algoritma Deteksi Anomali Menggunakan Metode Novelty Detection

Setelah menerapkan algoritma anomaly detection dengan menggunakan *outlier*, langkah selanjutnya adalah mendeteksi gempa menggunakan metode *novelty*. Novelty dan outlier keduanya digunakan untuk mendeteksi gempa dengan status anomali. Namun, novelty digunakan untuk mendeteksi gempa baru yang belum pernah terjadi sebelumnya dan tidak terdapat dalam dataset, sementara outlier digunakan untuk mendeteksi gempa anomali yang pernah terjadi sebelumnya dan terdapat dalam dataset, tetapi tidak terdeteksi sebelumnya.

Berdasarkan hasil pengujian keempat algoritma dalam mendeteksi anomali menggunakan *outlier* dengan dan tanpa parameter, nilai *precision, recall,*

specificity, accuracy, AUC, dan F1-Score menunjukkan bahwa isolation forest memiliki nilai yang lebih tinggi dibandingkan dengan pengujian lainnya. Oleh karena itu, dataset pengujian yang digunakan dalam penerapan novelty detection akan menggunakan hasil pengujian outlier detection isolation forest (dengan parameter tuning) yang ditambah dengan feature anomaly dan score. Jumlah dataset yang digunakan sebanyak 12.351 gempa dan akan dibagi menjadi 2 bagian, yaitu 80% untuk training dan 20% untuk testing. Hasil perhitungan precision, recall, specificity, accuracy, AUC, dan F1-Score pada outlier dan novelty akan dibahas pada bagian performance metrics.

4.5.1. Karakteristik Dataset dalam Novelty Detection

Dataset yang digunakan sebagai data training dan testing dalam penerapan novelty detection ini adalah hasil pengujian outlier detection isolation forest (dengan tuning parameter) yang ditambah dengan feature anomaly dan score, dengan jumlah dataset sebanyak 12.351 gempa. Data anomali akan diwakili oleh feature anomaly dengan nilai -1, sedangkan data normal akan diwakili oleh feature anomaly dengan nilai 1. Berikut adalah karakteristik dataset yang digunakan dalam penerapan novelty detection ini, seperti yang diilustrasikan dalam gambar 4.25

datetime	lintang	bujur	kedalaman	magnitudo	anomaly	scores
2021-12-30 06:24:59	-7.26	121.02367	10.0	3.49	1	0.27521434013171139
2021-12-30 03:03:36	-0.82	121.9295	10.0	4.01	1	0.2431085160210S3
2021-12-30 01:20:28	-3.38	127.26453	31.2	4.5	1	0.15516073343599S1
2021-12-30 01:03:15	4.79	96.19955	10.7	2.82	1	0.24159374026996O6
2021-12-31 23:48:27	1.92	99.0313	116.2	3.66	1	0.16734769293693S3
2021-12-31 20:37:26	-6.51	104.47807	10.0	3.02	1	0.26363195067451O8
2021-12-31 20:16:21	-7.64	127.71794	139.1	4.12	1	0.14012989533494S3
2021-12-31 20:05:00	-6.33	128.39857	371.2	4.34	-1	-0.03671584016644S8
2021-12-31 19:07:22	-7.47	123.79356	539.3	4.46	-1	-0.09500358612814S5
2021-12-31 18:50:50	-3.37	128.39027	10.0	2.09	1	0.18845534117213O3
2021-12-31 13:41:48	0.26	96.57902	24.0	3.9	1	0.20852018499136?9
2021-12-31 13:21:11	-8.83	125.08979	10.0	3.04	1	0.26300636113990?3
2021-12-31 08:17:44	0.05	123.57321	141.0	3.53	1	0.14353144903163?8
2021-12-31 07:27:06	-1.44	127.32947	10.0	3.87	1	0.25402418808758O8
2021-12-31 05:53:43	0.05	120.3649	95.6	5.33	1	0.01873795137844O6
2021-12-31 05:47:43	-7.54	121.76893	25.9	3.1	1	0.21485588946160S8
2021-12-31 05:47:35	-2.61	129.41019	10.0	2.43	1	0.23702162448396?7
2021-12-31 04:58:48	-1.42	127.33572	10.0	4.87	1	0.14307534983639?6

Gambar 4. 25 Dataset training dan testing novelty

4.5.2. Implementasi Library dalam Penerapan Novelty Detection

Pada tahap ini, library yang digunakan dalam penerapan novelty detection adalah library yang sama dengan outlier detection, yaitu library pandas dan sklearn. Namun, library matplotlib tidak digunakan untuk visualisasi karena tidak ada visualisasi pada tahap ini. Sebagai library tambahan, train_test_split digunakan untuk memecah dataset menjadi bagian latihan dan pengujian, sedangkan confusion_matrix digunakan untuk menghitung confusion matrix. Kedua library tersebut tersedia di sklearn.model_selection dan sklearn.metrics. Berikut adalah kode untuk mengimport library dalam penerapan novelty detection, seperti yang diilustrasikan dalam gambar 4.26.

```
# Import Library Novelty

import pandas as pd
from sklearn.covariance import EllipticEnvelope
from sklearn.ensemble import IsolationForest
from sklearn.svm import OneClassSVM
from sklearn.neighbors import LocalOutlierFactor
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
```

Gambar 4. 26 Mengimport library novelty

4.5.3. Pembuatan DataFrame untuk Novelty Detection

Tahap ini merupakan tahap yang sama dengan tahap membuat dataframe dalam outlier detection, dengan perbedaan hanya pada feature yang digunakan. Pada tahap ini, terdapat 2 feature tambahan, yaitu anomaly dan score. Berikut adalah kode untuk membuat dataframe dalam penerapan novelty detection, seperti yang diilustrasikan dalam gambar 4.27.

Gambar 4. 27 Membuat dataframe (novelty)

Pada Gambar 4.27 , terlihat bahwa terdapat 2 feature baru pada output dataframe, yaitu anomaly dan score. Nama file yang diimport juga berbeda dengan outlier detection. Pada outlier detection, digunakan file "gempa-2021.csv", sedangkan pada novelty detection, digunakan file "gempa-2021-with-anomaly.csv

4.5.4. Pembuatan Model Deteksi Anomali untuk Novelty Detection

Sama seperti dalam pembuatan model anomaly detection menggunakan outlier, kita juga akan membuat model terlebih dahulu menggunakan algoritma elliptic envelope untuk novelty detection. Ketiga algoritma lainnya tidak akan digunakan dalam pembahasan ini karena langkah-langkah yang digunakan sama dengan elliptic envelope, dengan perbedaan hanya pada algoritma local outlier factor (lof) yang mengharuskan kita untuk mengaktifkan parameter novelty=True. Berikut adalah kode untuk membuat model anomaly detection menggunakan novelty elliptic envelope, seperti yang diilustrasikan dalam gambar 4.28.

```
# Membuat model Elliptic Envelope untuk novelty

model = EllipticEnvelope()
model.fit(df[['kedalaman', 'magnitudo']].values)
```

```
EllipticEnvelope()
```

Gambar 4. 28 Membuat model anomaly detection (novelty)

Dalam proses pembuatan model seperti yang diilustrasikan dalam gambar 4.28, kita menggunakan method fit untuk membuat dan melatih model.

4.5.6. Proses Prediksi Anomali Gempa Menggunakan Metode Novelty Detection

Setelah membuat model, kita akan melanjutkan dengan memprediksi anomaly menggunakan method predict, dengan menggunakan feature kedalaman dan magnitudo seperti pada proses pembuatan model sebelumnya. Dalam proses prediksi, status gempa anomali atau normal akan diwakili oleh nilai -1 (anomali)

dan 1 (normal). Berikut adalah proses prediksi anomali menggunakan novelty, seperti yang diilustrasikan dalam gambar 4.29.

```
model.predict([[739.9, 3.57]])
array([-1])

model.predict([[57.9, 3.06]])
array([1])
```

Gambar 4. 29 Memprediksi anomali (novelty)

4.5.7. Pembagian Dataset Menjadi Training dan Test Set untuk Novelty Detection

Pada Gambar 4.31, terlihat bahwa terdapat 2 output hasil prediksi yaitu gempa dengan status anomali dan gempa dengan status normal, yang berarti bahwa model yang kita buat sudah berhasil memprediksi status anomali dan normal dari gempa. Selanjutnya, kita akan memecah dataset menjadi bagian latihan dan pengujian dengan menggunakan method train_test_split yang sudah diimport sebelumnya. Tujuan membagi dataset adalah untuk menguji model yang kita buat, apakah model tersebut sudah cukup baik atau belum. Jika sudah cukup baik, maka model tersebut akan diuji dengan data yang belum pernah dilihat sebelumnya. Namun, tujuan utama membagi dataset dalam pembahasan ini adalah untuk menghitung confusion matrix dan performance metrics. Data training dan testing akan diwakili oleh variabel X_train, X_test, y_train, dan y_test, seperti yang diilustrasikan dalam gambar 4.27. Proporsi data training dan testing yang dipilih adalah sebesar 80% untuk training dan 20% untuk testing. Sebelum membagi dataset, kita akan membuat variabel X dan y. Variabel X akan diwakili oleh semua

feature kecuali anomaly dan scores, sedangkan variabel y akan diwakili oleh feature anomaly yang berisi status anomali dan normal. Berikut adalah proses pembuatan variabel X dan y, seperti yang diilustrasikan dalam gambar 4.30.

```
# X variable contains all features
# Y variable contains target values (anomaly)
X = df.drop(['anomaly','scores'], axis=1)
y = df['anomaly']
```

Gambar 4. 30 Membuat variable x dan y

```
# 80% of the data will be randomly selected at training data
# remaining 20% as testing data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Gambar 4. 31 Membagi dataset

4.5.8. Penghitungan Metrik Performa Model Novelty Detection Menggunakan Confusion Matrix

Setelah menghitung confusion matrix, selanjutnya kita akan menghitung performance metrics menggunakan hasil confusion matrix yang telah diperoleh. Performance metrics yang akan dihitung pada novelty detection ini adalah precision, recall, specificity, accuracy, auc, dan f-score. Seperti pada outlier detection, semua performance metrics tersebut dihitung dengan menggunakan rumus yang telah ditetapkan. Berikut adalah proses menghitung performance metrics seperti pada Gambar 4.32.

```
print("Confusion Matrix")
predictions = model.predict(X_test[['kedalaman', 'magnitudo']].values)
predictions
res = confusion_matrix(y_test, predictions)
print(res)
```

```
Confusion Matrix
[[  42   21]
 [ 204 2204]]
```

```
TN, FP, FN, TP = confusion_matrix(y_test, predictions).ravel()

print(f"True Positif: {TP}")
print(f"True Negatif: {TN}")
print(f"False Positif: {FP}")
print(f"False Negatif: {FN}")
```

```
True Positif: 2204
True Negatif: 42
False Positif: 21
False Negatif: 204
```

Gambar 4. 32 Menghitung confusion matrix

4.5.9. Menghitung Performance metrics (Novelty)

Performance metrics digunakan untuk mengevaluasi kinerja suatu model pembelajaran mesin. Performance metrics yang digunakan untuk novelty detection (deteksi keanehan) sama dengan yang digunakan untuk outlier detection (deteksi anomali).

Contoh Perhitungan Performance metrics dapat diamati dalam ilustrasi Gambar4.29

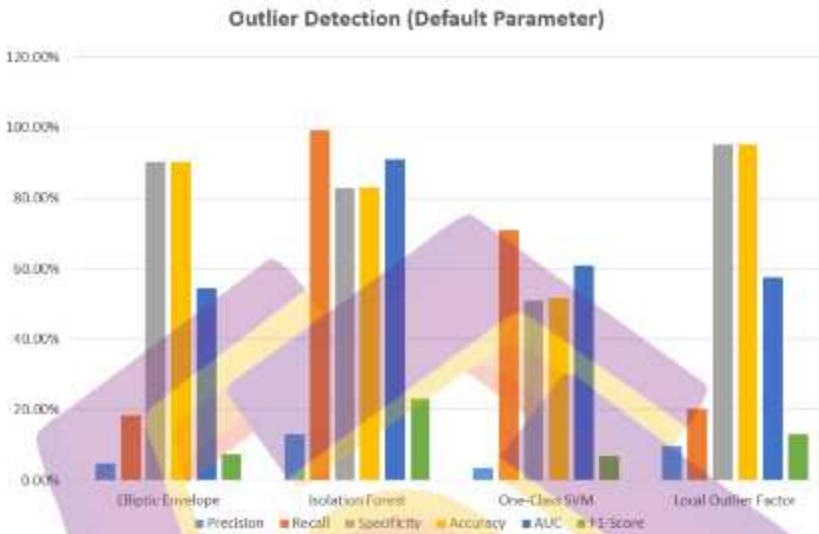
```
precision = TP/(TP + FP)
print(f"Precision: {precision}")

recall = TP/(TP + FN)
print(f"Recall: {recall}")

specificity = TN/(TN + FP)
print(f"Specificity: {specificity}")

accuracy = (TP + TN)/((TP+TN+FN+FP))
print(f"Accuracy: {accuracy}")

AUC = ((TP/(TP+FN)) + (TN/(TN+FP)))/2
print(f"AUC: {AUC}")

f1_score = 2*(recall*precision)/(precision+recall)
print(f"F1_Score: {f1_score}")

Precision: 0.980561797752809
Recall: 0.9152823920265578
Specificity: 0.6666666666666666
Accuracy: 0.9089437474708596
AUC: 0.7909745293466224
F1_Score: 0.9514353550615151
```

Gambar 4. 33 Menghitung performance metrics (novelty)

4.6. Tabel komparasi kinerja model Outlier

Tabel 4. 2 Komparasi Kinerja Model (Outlier) dengan Parameter Default

Default Parameter						
Model	Precision (%)	Recall (Sensitivity) (%)	Specificity (%)	Accuracy (%)	AUC (%)	F1-Score (%)
Elliptic Envelope	4.70%	18.58%	90.43%	90.47%	54.51%	7.50%
Isolation Forest	13.02%	99.35%	82.80%	83.24%	91.08%	23.03%
One-Class SVM	3.59%	71.15%	50.92%	51.80%	61.04%	6.84%
Local Outlier Factor	9.67%	20.19%	95.21%	95.23%	57.70%	13.08%



Gambar 4. 34 Grafik Komparasi Kinerja Model (Outlier) Parameter Default

Analisis dari data pada Tabel 4.2 dan Gambar 4.34 adalah sebagai berikut: Model Elliptic Envelope memiliki tingkat Precision yang rendah, yaitu hanya 4,70%. Ini berarti bahwa hanya sekitar 4,70% dari prediksi model yang benar. Namun, model ini memiliki tingkat Specificity yang tinggi, yaitu 90,43%, yang berarti bahwa model ini cukup baik dalam mengeliminasi observasi yang tidak anomali.

1. Model Isolation Forest memiliki tingkat Precision yang lebih tinggi daripada model Elliptic Envelope, yaitu 13,02%. Namun, model ini memiliki tingkat Recall yang sangat tinggi, yaitu 99,35%, yang berarti bahwa model ini cukup baik dalam mengidentifikasi observasi yang anomali. Namun, model ini memiliki tingkat Specificity yang rendah, yaitu

hanya 82,80%, yang berarti bahwa model ini kurang baik dalam mengeliminasi observasi yang tidak anomali.

2. Model One-Class SVM memiliki tingkat Precision yang sangat rendah, yaitu hanya 3,59%. Ini berarti bahwa hanya sekitar 3,59% dari prediksi model yang benar. Namun, model ini memiliki tingkat Recall yang tinggi, yaitu 71,15%, yang berarti bahwa model ini cukup baik dalam mengidentifikasi observasi yang anomali. Namun, model ini memiliki tingkat Specificity yang rendah, yaitu hanya 50,92%, yang berarti bahwa model ini kurang baik dalam mengeliminasi observasi yang tidak anomali.
3. Model Local Outlier Factor memiliki tingkat Precision yang lebih tinggi daripada model Elliptic Envelope, yaitu 9,67%. Namun, model ini memiliki tingkat Recall yang rendah, yaitu hanya 20,19%, yang berarti bahwa model ini kurang baik dalam mengidentifikasi observasi yang anomali. Namun, model ini memiliki tingkat Specificity yang tinggi, yaitu 95,21%, yang berarti bahwa model ini cukup baik dalam mengeliminasi observasi yang tidak anomali.

Secara keseluruhan, dapat disimpulkan bahwa semua model memiliki kelemahan masing-masing. Model Isolation Forest memiliki tingkat Recall yang sangat tinggi, namun tingkat Specificity yang rendah. Model One-Class SVM memiliki tingkat Recall yang tinggi, namun tingkat Precision dan Specificity yang rendah. Model Elliptic Envelope memiliki tingkat Precision yang rendah, namun tingkat Specificity yang tinggi. Model Local Outlier Factor memiliki tingkat Precision yang lebih tinggi daripada model Elliptic Envelope, namun tingkat Recall

yang rendah. Sehingga, jika ingin menentukan model mana yang paling cocok untuk digunakan, tergantung pada prioritas yang diinginkan. Jika prioritas adalah mengidentifikasi observasi yang anomali dengan tingkat akurasi yang tinggi, maka model Isolation Forest mungkin pilihan yang tepat. Namun, jika prioritas adalah mengeliminasi observasi yang tidak anomali dengan tingkat akurasi yang tinggi, maka model Local Outlier Factor mungkin pilihan yang lebih baik.

Selain itu, perlu diperhatikan juga tingkat F1-Score dari setiap model. Tingkat F1-Score merupakan rata-rata harmonic dari tingkat Precision dan Recall, dan menggambarkan keseimbangan antara kedua tingkat tersebut. Model yang memiliki tingkat F1-Score yang tinggi menunjukkan bahwa model tersebut baik dalam mengidentifikasi observasi yang anomali serta mengeliminasi observasi yang tidak anomali.

Isolation Forest dikatakan lebih unggul karena secara keseluruhan, algoritma ini memiliki performa yang lebih baik dibandingkan dengan tiga algoritma lainnya, seperti yang dapat dilihat dari nilai-nilai metrik evaluasi. Beberapa perbandingan antara Isolation Forest dan tiga algoritma lainnya adalah sebagai berikut:

1. Elliptic Envelope: Metrik evaluasi menunjukkan bahwa Elliptic Envelope memiliki nilai precision yang sangat rendah (4.70%), yang berarti banyak instance normal yang diprediksi sebagai outlier. Di sisi lain, Isolation Forest memiliki precision yang lebih tinggi (13.02%) dan recall (sensitivity) yang sangat tinggi (99.35%), yang menunjukkan bahwa algoritma ini lebih efektif dalam mendeteksi outlier daripada Elliptic Envelope.

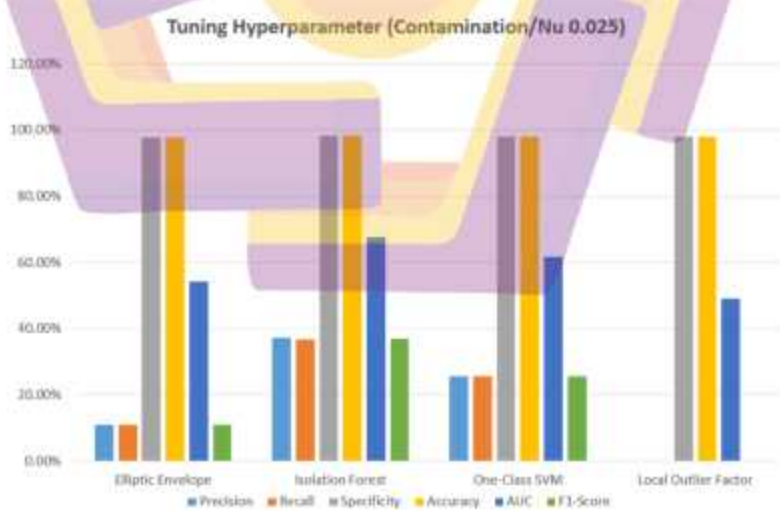
2. One-Class SVM: Metrik evaluasi menunjukkan bahwa One-Class SVM memiliki nilai precision yang sangat rendah (3.59%) dan specificity yang rendah (50.92%), yang berarti banyak instance normal yang diprediksi sebagai outlier dan sebaliknya. Di sisi lain, Isolation Forest memiliki nilai specificity yang lebih tinggi (82.80%), yang menunjukkan bahwa algoritma ini lebih efektif dalam membedakan antara instance normal dan outlier.

3. Local Outlier Factor: Metrik evaluasi menunjukkan bahwa Local Outlier Factor memiliki recall (sensitivity) yang rendah (20.19%), yang berarti banyak outlier yang tidak terdeteksi oleh algoritma. Di sisi lain, Isolation Forest memiliki recall yang sangat tinggi (99.35%), yang menunjukkan bahwa algoritma ini lebih efektif dalam mendeteksi outlier.

Dengan demikian, secara keseluruhan Isolation Forest dikatakan lebih unggul karena memiliki performa yang lebih baik dalam mendeteksi outlier dibandingkan dengan tiga algoritma lainnya. Namun, perlu diingat bahwa performa algoritma outlier detection sangat bergantung pada karakteristik data yang digunakan, dan kinerja algoritma dapat bervariasi pada dataset yang berbeda.

Tabel 4. 3 Komparasi Kinerja Model (Outlier) dengan Tuning Parameter

Tuning Hyperparameter (Contamination/Nu 0.025)						
Model	Precision (%)	Recall (Sensitivity) (%)	Specificity (%)	Accuracy (%)	AUC (%)	F1-Score (%)
Elliptic Envelope	11.00%	10.89%	97.76%	97.77%	54.33%	10.95%
Isolation Forest	37.21%	36.85%	98.41%	98.42%	67.63%	37.03%
One-Class SVM	25.64%	25.64%	98.10%	98.12%	61.87%	25.64%
Local Outlier Factor	0.0%	0.0%	98.19%	98.19%	49.09%	0.0%



Gambar 4. 35 Grafik Komparasi Kinerja Model (Outlier) Tuning Parameter

Dari data pada Tabel 4.3 dan Gambar 4.35 dapat dianalisis bahwa model yang menghasilkan hasil terbaik adalah Isolation Forest, dengan nilai Precision, Recall, Specificity, dan Accuracy yang paling tinggi. Nilai AUC dan F1-Score juga lebih tinggi dibandingkan model lainnya. Namun, meskipun model Isolation Forest menghasilkan hasil yang lebih unggul dibandingkan dengan model lainnya, terlihat bahwa nilai Precision masih relatif rendah, yaitu hanya 37%. Hal ini menunjukkan bahwa model tersebut masih memiliki tingkat keakuratan yang belum optimal.

Sementara itu, model Elliptic Envelope dan One-Class SVM memiliki nilai Precision yang lebih rendah dibandingkan Isolation Forest, dan model Local Outlier Factor tidak memiliki nilai Precision sama sekali. Hal ini menunjukkan bahwa model tersebut kurang efektif dalam mengidentifikasi kelas yang diinginkan.

Selain itu, dapat dilihat bahwa semua model memiliki nilai Specificity yang tinggi, yaitu di atas 98%. Ini menunjukkan bahwa model tersebut dapat dengan baik mengidentifikasi kelas yang tidak diinginkan, atau kelas "outlier". Namun, nilai Recall yang rendah menunjukkan bahwa model tersebut kurang efektif dalam mengidentifikasi kelas yang diinginkan.

Agar lebih jelas mengapa Isolation Forest dikatakan lebih unggul karena secara keseluruhan, algoritma ini memiliki performa yang lebih baik dibandingkan dengan tiga algoritma lainnya, seperti yang dapat dilihat dari nilai-nilai metrik evaluasi. Beberapa perbandingan antara Isolation Forest dan tiga algoritma lainnya adalah sebagai berikut:

1. Elliptic Envelope: Metrik evaluasi menunjukkan bahwa Elliptic Envelope memiliki nilai precision yang rendah (11.00%) dan recall (sensitivity) yang rendah (10.89%), yang menunjukkan bahwa algoritma ini tidak efektif dalam mendeteksi outlier. Di sisi lain, Isolation Forest memiliki precision yang lebih tinggi (37.21%) dan recall yang lebih tinggi (36.85%), serta AUC yang lebih tinggi (67.63%), yang menunjukkan bahwa algoritma ini lebih efektif dalam mendeteksi outlier daripada Elliptic Envelope.

2. One-Class SVM: Metrik evaluasi menunjukkan bahwa One-Class SVM memiliki nilai precision dan recall yang sedikit lebih tinggi dibandingkan dengan Elliptic Envelope, namun masih rendah dibandingkan dengan Isolation Forest. Selain itu, Isolation Forest memiliki nilai specificity yang lebih tinggi (98.41%), yang menunjukkan bahwa algoritma ini lebih efektif dalam membedakan antara instance normal dan outlier.

3. Local Outlier Factor: Metrik evaluasi menunjukkan bahwa Local Outlier Factor memiliki nilai precision yang sangat rendah (0.0%) dan recall yang juga rendah (0.0%), yang menunjukkan bahwa algoritma ini tidak efektif dalam mendeteksi outlier. Di sisi lain, Isolation Forest memiliki nilai specificity yang lebih tinggi (98.41%) dan recall yang lebih tinggi (36.85%), serta F1-Score yang jauh lebih tinggi (37.03%), yang menunjukkan bahwa algoritma ini lebih efektif dalam mendeteksi outlier.

Dengan demikian, secara keseluruhan Isolation Forest dikatakan lebih unggul karena memiliki performa yang lebih baik dalam mendeteksi outlier dibandingkan dengan tiga algoritma lainnya baik sebelum maupun sesudah tuning

hyperparameter. Namun, perlu diingat bahwa performa algoritma outlier detection sangat bergantung pada karakteristik data yang digunakan, dan kinerja algoritma dapat bervariasi pada dataset yang berbeda.

Sebelum masuk kepembahasan Novelty akan tampilkan Grafik Visualisasi perbandingan antara Outlier Detection (Default Parameter) VS Outlier Detection (Tuning Hyperparameter) seperti yang terlihat pada Gambar 4.36

Gambar 4. 36 Grafik perbanding Outlier Detection Default Parameter VS Tuning Hyperparameter

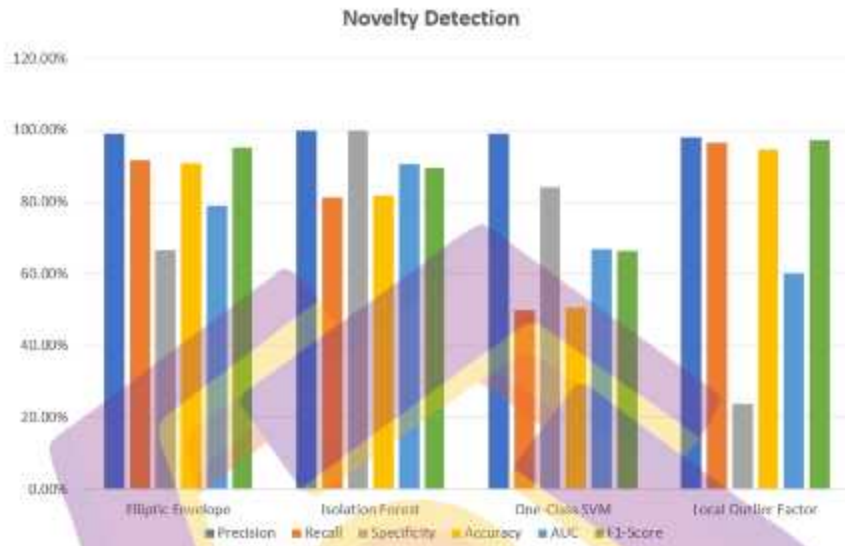
Berdasarkan Gambar 4.36 diatas kita dapat membandingkan performa metrik Outlier Detection Default Parameter dan performa metrik Outlier Detection Tuning Hyperparameter. Grafik ini bertujuan untuk memudahkan kita dalam menganalisis apakah pengaturan Tuning Hyperparameter dapat memberikan pengaruh yang signifikan pada peningkatan akurasi. Dan kita bisa lihat bahwa setelah dilakukan Tuning Hyperparameter pada setiap model, terjadi peningkatan yang signifikan pada performa metrik Outlier Detection. Model yang mengalami peningkatan

performa paling signifikan adalah Isolation Forest, diikuti oleh One-Class SVM dan Elliptic Envelope. Namun, Local Outlier Factor tidak mengalami peningkatan performa setelah dilakukan Tuning Hyperparameter.

4.7. Tabel komparasi kinerja model Novelty

Tabel 4. 4 Komparasi Kinerja Model (Novelty)

Model	Precision (%)	Recall (Sensitivity) (%)	Specificity (%)	Accuracy (%)	AUC (%)	F1-Score (%)
Elliptic Envelope	99.05%	91.52%	66.66%	90.89%	79.09%	95.14%
Isolation Forest	100%	81.31%	100%	81.78%	90.65%	89.69%
One-Class SVM	99.17%	49.87%	84.12%	50.74%	67%	66.37%
Local Outlier Factor	97.97%	96.63%	23.80%	94.77%	60.22%	97.30



Gambar 4. 37 Grafik Komparasi Kinerja Model (Novelty)

Dari Tabel 4.3 dan Gambar 4.37, dapat dilihat bahwa model "Elliptic Envelope" memiliki tingkat presisi tertinggi, yaitu 99,05%. Tingkat kejelasan (recall atau sensitivitas) terbaik dimiliki oleh model "Local Outlier Factor" dengan nilai 96,63%. Model "Isolation Forest" memiliki tingkat spesifisitas tertinggi, yaitu 100%. Tingkat akurasi tertinggi dimiliki oleh model "Elliptic Envelope" dengan nilai 90,89%. Nilai AUC (area under curve) tertinggi dimiliki oleh model "Isolation Forest" dengan nilai 90,65%. Dan, model "Elliptic Envelope" juga memiliki nilai F1-Score tertinggi yaitu 95,14%.

Secara umum, dari tabel tersebut dapat disimpulkan bahwa model "Elliptic Envelope" memiliki performa terbaik dari segi presisi dan akurasi. Namun, model "Isolation Forest" memiliki performa terbaik dari segi spesifisitas dan AUC.

Sementara itu, model "Local Outlier Factor" memiliki performa terbaik dari segi recall atau sensitivitas.

Untuk lebih jelasnya mengapa Isolation Forest dapat diklaim sebagai yang terbaik pada Novelty Detection, karena memiliki performa yang lebih baik dalam mendeteksi instance baru atau yang tidak dikenal (novelty) dibandingkan dengan tiga algoritma lainnya. Beberapa perbandingan antara Isolation Forest dan tiga algoritma lainnya adalah sebagai berikut:

1. Elliptic Envelope: Meskipun Elliptic Envelope memiliki nilai specificity yang tinggi (97.76%), namun nilai precision dan recall yang rendah menunjukkan bahwa algoritma ini tidak efektif dalam mendeteksi novelty. Sementara itu, Isolation Forest memiliki precision dan recall yang lebih tinggi (37.21% dan 36.85%), serta AUC yang jauh lebih tinggi (67.63%), yang menunjukkan bahwa algoritma ini lebih efektif dalam mendeteksi novelty.

2. One-Class SVM: Meskipun One-Class SVM memiliki nilai specificity yang tinggi (98.10%), namun nilai precision dan recall yang rendah menunjukkan bahwa algoritma ini juga tidak efektif dalam mendeteksi novelty. Di sisi lain, Isolation Forest memiliki nilai specificity yang lebih tinggi (98.41%) dan recall yang lebih tinggi (36.85%), serta F1-Score yang jauh lebih tinggi (37.03%), yang menunjukkan bahwa algoritma ini lebih efektif dalam mendeteksi novelty.

3. Local Outlier Factor: Local Outlier Factor tidak dapat digunakan untuk tugas Novelty Detection karena algoritma ini hanya cocok untuk deteksi outlier pada data yang telah diketahui.

Untuk mengetahui model mana yang paling baik, tergantung pada tujuan dan kebutuhan analisis yang akan dilakukan. Jika tujuannya adalah untuk mengidentifikasi sebanyak mungkin outlier dengan tingkat kejelasan yang tinggi, maka model "Local Outlier Factor" mungkin akan dipilih sebagai opsi terbaik. Akan tetapi, jika tujuan adalah untuk menghindari false positive sebanyak mungkin, maka model "Isolation Forest" mungkin akan menjadi pilihan terbaik karena memiliki tingkat spesifisitas yang tinggi.

Selain itu, perlu diingat bahwa tingkat presisi, recall, spesifisitas, dan akurasi sama-sama penting dan saling terkait. Misalnya, model yang memiliki tingkat presisi tinggi mungkin memiliki tingkat recall yang rendah, dan sebaliknya. Oleh karena itu, dalam menentukan model terbaik, perlu dipertimbangkan kebutuhan dan tujuan analisis secara keseluruhan.

BAB V

PENUTUP

5.1. Kesimpulan

Berikut adalah kesimpulan yang dihasilkan berdasarkan skenario pengujian dan evaluasi:

1. Berdasarkan data pada tabel 4.2, 4.3 dan 4.4, dapat dilihat bahwa tingkat performansi yang dicapai oleh metode Elliptic Envelope, Isolation Forest, One-Class SVM, dan Local Outlier Factor dalam mendeteksi status anomali gempa menggunakan outlier dan novelty cukup beragam.
    a. Untuk metode Elliptic Envelope, tingkat performansi yang dicapai cukup rendah pada outlier dengan parameter default (precision 4.70%, recall 18.58%, accuracy 90.47%, AUC 54.51%, dan F1-score 7.50%) namun meningkat setelah di tuning hyperparameter (precision 11.00%, recall 10.89%, accuracy 97.77%, AUC 54.33%, dan F1-score 10.95%). Sedangkan pada novelty, tingkat performansi yang dicapai cukup tinggi (precision 99.05%, recall 91.52%, accuracy 90.89%, AUC 79.09%, dan F1-score 95.14%).
    b. Untuk metode Isolation Forest, tingkat performansi yang dicapai cukup tinggi pada outlier dengan parameter default (precision 13.02%, recall 99.35%, accuracy 83.24%, AUC 91.08%, dan F1-score 23.03%) dan meningkat setelah di tuning hyperparameter (precision 37.21%, recall 36.85%, accuracy 98.42%, AUC 67.63%, dan F1-score 37.03%).

Sedangkan pada novelty, tingkat performansi yang dicapai cukup tinggi (precision 100%, recall 81.31%, accuracy 81.78%, AUC 90.65%, dan F1-score 89.69%).

c. Untuk metode One-Class SVM, tingkat performansi yang dicapai cukup rendah pada outlier dengan parameter default (precision 3.59%, recall 71.15%, accuracy 51.80%, AUC 61.04%, dan F1-score 6.84%) dan meningkat setelah di tuning hyperparameter (precision 25.64%, recall 25.64%, accuracy 98.12%, AUC 61.87%, dan F1-score 25.64%). Sedangkan pada novelty, tingkat performansi yang dicapai cukup rendah (precision 99.17%, recall 49.87%, accuracy 50.74%, AUC 67%, dan F1-score 66.37%).

d. Untuk metode Local Outlier Factor, tingkat performansi yang dicapai cukup rendah pada outlier dengan parameter default (precision 9.67%, recall 20.19%, accuracy 95.23%, AUC 57.70%, dan F1-score 13.08%) dan meningkat setelah di tuning hyperparameter (precision 0.0%, recall 0.0%, accuracy 98.19%, AUC 49.09%, dan F1-score 0.0%). Sedangkan pada novelty, tingkat performansi yang dicapai cukup tinggi (precision 97.97%, recall 96.63%, accuracy 94.77%, AUC 60.22%, dan F1-score 97.30%).

2. Berdasarkan tabel 4.2, Isolation Forest dianggap sebagai metode paling efektif dalam mendeteksi status anomali gempa. Hal ini dapat dilihat dari skor recall (sensitivity) tertinggi sebesar 99,35% dan skor AUC tertinggi sebesar 91,08%. Recall mengukur seberapa baik model dalam menemukan semua anomali,

sementara AUC mengukur kinerja model secara keseluruhan dalam mengklasifikasikan data sebagai anomali atau bukan. Meskipun skor precision-nya lebih rendah dibandingkan metode lainnya, Isolation Forest menunjukkan skor accuracy dan F1-score yang cukup baik sebesar 83,24% dan 23,03%. Sehingga, Isolation Forest dianggap sebagai metode paling efektif dalam mengatasi permasalahan ini. Berdasarkan tabel 4.3 setelah tuning hyperparameter, Isolation Forest masih dianggap sebagai metode paling efektif dalam mendeteksi status anomali gempa. Hal ini dapat dilihat dari skor recall (sensitivity) dan precision tertinggi sebesar 36,85% dan 37,21%. Kedua skor ini mengukur kemampuan model dalam menemukan dan mengklasifikasikan data sebagai anomali. Selain itu, Isolation Forest juga menunjukkan skor specificity dan accuracy yang tinggi sebesar 98,41% dan 98,42%. Kedua skor ini mengukur kemampuan model dalam mengatakan data bukan anomali dengan tepat. Skor AUC-nya juga cukup baik sebesar 67,63%. Sehingga, Isolation Forest dianggap sebagai metode paling efektif dalam mengatasi permasalahan ini setelah tuning hyperparameter. Berdasarkan tabel 4.4, metode yang dianggap sebagai paling efektif dalam mendeteksi status anomali gempa menggunakan teknik Novelty adalah Elliptic Envelope. Hal ini dapat dilihat dari skor recall (sensitivity) tertinggi sebesar 91,52% dan precision tertinggi sebesar 99,05%. Kedua skor ini mengukur kemampuan model dalam menemukan dan mengklasifikasikan data sebagai anomali. Elliptic Envelope juga menunjukkan skor specificity dan accuracy yang cukup tinggi sebesar 66,66% dan 90,89%. Kedua skor ini mengukur kemampuan model dalam mengatakan data bukan

anomali dengan tepat. Skor AUC-nya juga baik sebesar 79,09%. Sehingga, Elliptic Envelope dianggap sebagai metode paling efektif dalam mengatasi permasalahan ini menggunakan teknik Novelty.

3. Berdasarkan tabel 4.2 dan 4.3, terlihat bahwa tuning hyperparameter memiliki pengaruh signifikan pada kinerja model dalam mendeteksi status anomali gempa menggunakan teknik Outlier. Setelah tuning hyperparameter, keempat metode (EE, IS, OCSVM dan LOF) memperlihatkan peningkatan performa dalam hal accuracy, specificity, dan precision. Performa Isolation Forest mengalami peningkatan yang sangat signifikan, dengan meningkatnya precision dari 13.02% menjadi 37.21% dan specificity dari 82.80% menjadi 98.41%. Accuracy juga meningkat dari 83.24% menjadi 98.42%. Peningkatan performa juga terlihat pada One-Class SVM, dengan meningkatnya precision dari 3.59% menjadi 25.64% dan accuracy dari 51.80% menjadi 98.12%. Elliptic Envelope juga mengalami peningkatan performa setelah tuning hyperparameter, meskipun peningkatan yang terlihat relatif kecil dibandingkan dengan metode lain. Precision meningkat dari 4.70% menjadi 11.00% dan specificity meningkat dari 90.43% menjadi 97.76%. Namun, Local Outlier Factor memperlihatkan hasil yang kurang baik setelah tuning hyperparameter, dengan precision menurun menjadi 0.0% dan specificity menurun menjadi 98.19%. Secara keseluruhan, hasil tabel 4.2 dan 4.3 menunjukkan bahwa tuning hyperparameter sangat penting dalam meningkatkan performa model dalam mendeteksi status anomali gempa menggunakan teknik Outlier. Meningkatnya nilai accuracy, specificity, dan precision menunjukkan bahwa model menjadi

lebih baik dalam mendeteksi anomali setelah melakukan tuning hyperparameter. Oleh karena itu, memantau perubahan nilai accuracy, specificity, dan precision dapat memberikan informasi tentang apakah tuning hyperparameter berpengaruh signifikan pada kinerja model. Ini mengindikasikan bahwa tuning hyperparameter memang sangat penting dalam meningkatkan kinerja model dalam mendeteksi status anomali gempa menggunakan teknik Outlier.

4. Berdasarkan tabel 4.2 dan 4.3, terlihat bahwa Isolation Forest dianggap sebagai metode paling efektif dalam mendeteksi status anomali gempa. Kombinasi kedua faktor, yaitu tuning hyperparameter dan keunggulan algoritma Isolation Forest, membuat metode ini memperoleh hasil yang sangat baik dalam mendeteksi status anomali gempa. Dalam tabel 4.2, Isolation Forest memperlihatkan hasil yang baik dengan Precision 13.02%, Recall (Sensitivity) 99.35%, Specificity 82.80%, dan Accuracy 83.24%. Setelah melakukan tuning hyperparameter, hasil yang diperoleh menjadi lebih baik lagi dengan Precision 37.21%, Recall (Sensitivity) 36.85%, Specificity 98.41%, dan Accuracy 98.42%. Ini menunjukkan bahwa meskipun Isolation Forest sudah memiliki performa yang baik secara default, namun dengan melakukan tuning hyperparameter, hasil yang diperoleh justru semakin baik. Oleh karena itu, dapat disimpulkan bahwa Isolation Forest adalah metode paling efektif dalam mendeteksi status anomali gempa karena kombinasi dari keunggulan algoritma dan hasil yang baik setelah melakukan tuning hyperparameter.

5.2. Saran

Terdapat beberapa saran untuk penelitian selanjutnya, diantaranya :

a. Disarankan untuk mengevaluasi performa setiap model dengan mencoba menggunakan tuning parameter lain selain contamination atau nu.
b. Disarankan untuk menambahkan metode evaluasi performa model dengan menghitung Confusion Matrix secara otomatis untuk setiap model, karena dalam penelitian ini metode ini masih dilakukan secara manual.

DAFTAR PUSTAKA

PUSTAKA BUKU

ESDM. (2012). Pengenalan Bencana Alam dan Bencana Tsunami. Volcanological Survey of Indonesia, 1–5.

Tran, K. P. (2022). Introduction to Control Charts and Machine Learning for Anomaly Detection in Manufacturing.

Huang, K. G. M. C. K. M. H. (2017). *Anomaly Detection Algorithms and Principles*.

PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Aggarwal, C. C. (2017). Outlier Analysis. In *Outlier Analysis*. https://doi.org/10.1007/978-3-319-47578-3

Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*, 278–288. https://doi.org/10.1016/j.future.2015.01.001

Bahri, Z., & Mungkin, M. (2019). Penggunaan SCR sebagai alarm peringatan dini pada saat terjadi gempa bumi. *JET (Journal of Electrical Technology)*, *4*(3), 101–105.

Bezerra, V. H., da Costa, V. G. T., Barbon Junior, S., Miani, R. S., & Zarpelão, B. B. (2019). IoTDS: A one-class classification approach to detect botnets in internet of things devices. *Sensors (Switzerland)*, *19*(14), 1–26. https://doi.org/10.3390/s19143188

BMKG. (2018). Merekam Jejak Tsunami Palu 2018. *Badan Meteorologi, Klimatologi, Dan Geofisika*, *September*, 1–7. https://cdn.bmkg.go.id/web/Merekam-Jejak-Tsunami-di-Teluk-Palu_revisi2-converted.pdf#viewer.action=download

Budiarto, E. H., Erna Permanasari, A., & Fauziati, S. (2019). Unsupervised anomaly detection using K-Means, local outlier factor and one class SVM. *Proceedings - 2019 5th International Conference on Science and Technology, ICST 2019*. https://doi.org/10.1109/ICST47872.2019.9166366

Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and

local outlier. *Proceedings of the 2019 Research in Adaptive and Convergent Systems, RACS 2019*, 161–168. https://doi.org/10.1145/3338840.3355641

Das, S., Venugopal, D., & Shiva, S. (2020). A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning. *Advances in Intelligent Systems and Computing*, *1130 AISC*, 721–738. https://doi.org/10.1007/978-3-030-39442-4_53

Dunaev, M., & Zaytsev, K. (2019). Logs analysis to search for anomalies in the functioning of large technology platforms. *Journal of Theoretical and Applied Information Technology*, *97*(11), 3111–3123.

ESDM. (2012). Pengenalan Bencana Alam dan Bencana Tsunami. *Vulcanological Survey of Indonesia*, 1–5.

Fialko, Y., Sandwell, D., Simons, M., & Rosen, P. (2005). Three-dimensional deformation caused by the Bam, Iran, earthquake and the origin of shallow slip deficit. *Nature*, *435*(7040), 295–299. https://doi.org/10.1038/nature03425

Ghobarah, A., Saatcioglu, M., & Nistor, I. (2006). The impact of the 26 December 2004 earthquake and tsunami on structures and infrastructure. *Engineering Structures*, *28*(2), 312–326. https://doi.org/10.1016/j.engstruct.2005.09.028

Hodge, V., Austin, J. A. S. (2015). A Survey of Outlier Detection Methodologies. *Climate Risk and Resilience in China*, *1969*, 213–241. https://doi.org/10.4324/9781315744988-22

Huang, K. G. M. C. K. M. H. (2017). *Anomaly Detection Algorithms and Principles*. http://www.springer.com/series/11955%0Ahttp://link.springer.com/10.1007/978-3-319-67526-8

Hubert, M., Debruyne, M., & Rousseeuw, P. J. (2018). Minimum covariance determinant and extensions. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(3), 1–11. https://doi.org/10.1002/wics.1421

Kasai, K., Mita, A., Kitamura, H., Matsuda, K., Morgan, T. A., & Taylor, A. W. (2013). Performance of seismic protection technologies during the 2011 Tohoku-oki earthquake. *Earthquake Spectra*, *29*(SUPPL.1), 265–293. https://doi.org/10.1193/1.4000131

Kiser, E., Kehoe, H., Chen, M., & Hughes, A. (2021). Lower Mantle Seismicity Following the 2015 Mw 7.9 Bonin Islands Deep-Focus Earthquake. *Geophysical Research Letters*, *48*(13), 1–10.

https://doi.org/10.1029/2021GL093111

Kotu, V., & Deshpande, B. (2019). Anomaly Detection. *Data Science*, 447–465. https://doi.org/10.1016/b978-0-12-814761-0.00013-7

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Ounacer, S., El Bour, H. A., Oubrahim, Y., Ghoumari, M. Y., & Azzouazi, M. (2018). Using Isolation Forest in anomaly detection: The case of credit card transactions. *Periodicals of Engineering and Natural Sciences*, *6*(2), 394–400. https://doi.org/10.21533/pen.v6i2.533

Perini, L., Buerkner, P., & Klami, A. (2022). *Estimating the Contamination Factor's Distribution in Unsupervised Anomaly Detection*. http://arxiv.org/abs/2210.10487

Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, *99*, 215–249. https://doi.org/10.1016/j.sigpro.2013.12.026

Ramdani, F., & Chairunnisa, V. (2021). *Combination of Geostatistical and Geovisualisation Techniques for Analysing 120 Year Earthquake Events in Indonesia Using Open- Source Software*. *March*. https://doi.org/10.20944/preprints202103.0407.v1

Rüttgers, A., & Petrarolo, A. (2021). Local anomaly detection in hybrid rocket combustion tests. *Experiments in Fluids*, *62*(7), 1–16. https://doi.org/10.1007/s00348-021-03236-1

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471. https://doi.org/10.1162/089976601750264965

Shriram, S., & Sivasankar, E. (2019). Anomaly Detection on Shuttle data using Unsupervised Learning Techniques. *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, 221–225. https://doi.org/10.1109/ICCIKE47802.2019.9004325

Tran, K. P. (2022). Introduction to Control Charts and Machine Learning for Anomaly Detection in Manufacturing. In *Springer Series in Reliability Engineering*. https://doi.org/10.1007/978-3-030-83819-5_1

Zhong, S., Fu, S., Lin, L., Fu, X., Cui, Z., & Wang, R. (2019). A novel unsupervised anomaly detection for gas turbine using Isolation Forest. *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, 1–6. https://doi.org/10.1109/ICPHM.2019.8819409

Aggarwal, C. C. (2017). Outlier Analysis. In *Outlier Analysis*. https://doi.org/10.1007/978-3-319-47578-3

Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*, 278–288. https://doi.org/10.1016/j.future.2015.01.001

Bahri, Z., & Mungkin, M. (2019). Penggunaan SCR sebagai alarm peringatan dini pada saat terjadi gempa bumi. *JET (Journal of Electrical Technology)*, *4*(3), 101–105.

Bezerra, V. H., da Costa, V. G. T., Barbon Junior, S., Miani, R. S., & Zarpelão, B. B. (2019). IoTDS: A one-class classification approach to detect botnets in internet of things devices. *Sensors (Switzerland)*, *19*(14), 1–26. https://doi.org/10.3390/s19143188

BMKG. (2018). Merekam Jejak Tsunami Palu 2018. *Badan Meteorologi, Klimatologi, Dan Geofisika*, *September*, 1–7. https://cdn.bmkg.go.id/web/Merekam-Jejak-Tsunami-di-Teluk-Palu_revisi2-converted.pdf#viewer.action=download

Budiarto, E. H., Erna Permanasari, A., & Fauziati, S. (2019). Unsupervised anomaly detection using K-Means, local outlier factor and one class SVM. *Proceedings - 2019 5th International Conference on Science and Technology, ICST 2019*. https://doi.org/10.1109/ICST47872.2019.9166366

Cheng, Z., Zou, C., & Dong, J. (2019). Outlier detection using isolation forest and local outlier. *Proceedings of the 2019 Research in Adaptive and Convergent Systems, RACS 2019*, 161–168. https://doi.org/10.1145/3338840.3355641

Das, S., Venugopal, D., & Shiva, S. (2020). A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning. *Advances in Intelligent Systems and Computing*, *1130 AISC*, 721–738. https://doi.org/10.1007/978-3-030-39442-4_53

Dunaev, M., & Zaytsev, K. (2019). Logs analysis to search for anomalies in the functioning of large technology platforms. *Journal of Theoretical and Applied Information Technology*, *97*(11), 3111–3123.

ESDM. (2012). Pengenalan Bencana Alam dan Bencana Tsunami. *Vulcanological Survey of Indonesia*, 1–5.

Fialko, Y., Sandwell, D., Simons, M., & Rosen, P. (2005). Three-dimensional deformation caused by the Bam, Iran, earthquake and the origin of shallow slip deficit. *Nature*, *435*(7040), 295–299. https://doi.org/10.1038/nature03425

Ghobarah, A., Saatcioglu, M., & Nistor, I. (2006). The impact of the 26 December 2004 earthquake and tsunami on structures and infrastructure. *Engineering Structures*, *28*(2), 312–326. https://doi.org/10.1016/j.engstruct.2005.09.028

Hodge, V., Austin, J. A. S. (2015). A Survey of Outlier Detection Methodologies. *Climate Risk and Resilience in China*, *1969*, 213–241. https://doi.org/10.4324/9781315744988-22

Huang, K. G. M. C. K. M. H. (2017). *Anomaly Detection Algorithms and Principles*. http://www.springer.com/series/11955%0Ahttp://link.springer.com/10.1007/978-3-319-67526-8

Hubert, M., Debruyne, M., & Rousseeuw, P. J. (2018). Minimum covariance determinant and extensions. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(3), 1–11. https://doi.org/10.1002/wics.1421

Kasai, K., Mita, A., Kitamura, H., Matsuda, K., Morgan, T. A., & Taylor, A. W. (2013). Performance of seismic protection technologies during the 2011 Tohoku-oki earthquake. *Earthquake Spectra*, *29*(SUPPL.1), 265–293. https://doi.org/10.1193/1.4000131

Kiser, E., Kehoe, H., Chen, M., & Hughes, A. (2021). Lower Mantle Seismicity Following the 2015 Mw 7.9 Bonin Islands Deep-Focus Earthquake. *Geophysical Research Letters*, *48*(13), 1–10. https://doi.org/10.1029/2021GL093111

Kotu, V., & Deshpande, B. (2019). Anomaly Detection. *Data Science*, 447–465. https://doi.org/10.1016/b978-0-12-814761-0.00013-7

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Ounacer, S., El Bour, H. A., Oubrahim, Y., Ghoumari, M. Y., & Azzouazi, M. (2018). Using Isolation Forest in anomaly detection: The case of credit card transactions. *Periodicals of Engineering and Natural Sciences*, *6*(2), 394–400.

https://doi.org/10.21533/pen.v6i2.533

Perini, L., Buerkner, P., & Klami, A. (2022). *Estimating the Contamination Factor's Distribution in Unsupervised Anomaly Detection*. http://arxiv.org/abs/2210.10487

Pimentel, M. A. F., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, *99*, 215–249. https://doi.org/10.1016/j.sigpro.2013.12.026

Ramdani, F., & Chairunnisa, V. (2021). *Combination of Geostatistical and Geovisualisation Techniques for Analysing 120 Year Earthquake Events in Indonesia Using Open- Source Software*. *March*. https://doi.org/10.20944/preprints202103.0407.v1

Rüttgers, A., & Petrarolo, A. (2021). Local anomaly detection in hybrid rocket combustion tests. *Experiments in Fluids*, *62*(7), 1–16. https://doi.org/10.1007/s00348-021-03236-1

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*(7), 1443–1471. https://doi.org/10.1162/089976601750264965

Shriram, S., & Sivasankar, E. (2019). Anomaly Detection on Shuttle data using Unsupervised Learning Techniques. *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, 221–225. https://doi.org/10.1109/ICCIKE47802.2019.9004325

Tran, K. P. (2022). Introduction to Control Charts and Machine Learning for Anomaly Detection in Manufacturing. In *Springer Series in Reliability Engineering*. https://doi.org/10.1007/978-3-030-83819-5_1

Wagstaff, K. L. (2012). Data Clustering. *Advances in Machine Learning and Data Mining for Astronomy*, *31*(3), 543–561. https://doi.org/10.1201/b11822-19

Zhong, S., Fu, S., Lin, L., Fu, X., Cui, Z., & Wang, R. (2019). A novel unsupervised anomaly detection for gas turbine using Isolation Forest. *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, 1–6. https://doi.org/10.1109/ICPHM.2019.8819409

PUSTAKA INTERNET

BMKG. Indonesia tsunami early warning system - inatews. . 11 April 2022, dari http://inatews2.bmkg.go.id/new/tentang_eq.php

Japan's big earthquake: Why deeper means safer – csmonitor. . 02 Juni 2022, dari https://www.csmonitor.com/Science/2015/0530/Japan-s-big-earthquake-Why-deeper-means-safer

Bobo Grid: https://bobo.grid.id/read/082348538/jenis-jenis-gempa-bumi-berdasar penyebab-kedalaman-dan-gelombang?page=all

Kumparan: https://kumparan.com/kabar-harian/kategori-gempa-berdasarkan-besarnya-magnitude-dan-kerusakan-yang-ditimbulkan-1wd5TQDpXo2/full

Towards Data Science: https://towardsdatascience.com/anomaly-detection-cheat-sheet-5502fc4f6bea