ANALISIS TINGKAT KEJAHATAN MENGGUNAKAN MODEL RANDOM FOREST

OLEH :

PARHAN HAMBALI

RONI HABIBI S.KOM.,MT.,SFPC

Judul :

Analisis Tingkat Kejahatan Menggunakan Model Random Forest
2021,

Penulis :
PARHAN HAMBALI
RONI HABIBI., S.Kom., M.T., SFPC

Penyunting : CAHYO PRIANTO, S.Pd., M.T.,CDSP, SFPC
Layout : PARHAN HAMBALI

penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

Kata Sambutan

Puji syukur kehadirat Allah Subhanahu Wa Ta'ala, atas rahmat dan hidayat-Nya penulis dapat menyelesaikan buku yang berjudul "Analisis Tingkat Kejahatan Menggunakan Model Random Forest". Buku ini disusun dalam konteks mendorong masyarakat khususnya mahasiswa yang sedang mencari referensi kebutuhan laporan, baik laporan magang ataupun tugas akhir. Semoga buku ini bisa memberikan banyak manfaat kepada kalian.

Kata Pengantar

Puji syukur kehadirat Allah Subhanahu Wa Ta'ala, atas rahmat dan hidayat-Nya penulis dapat menyelesaikan buku yang berjudul "Analisis Tingkat Kejahatan Menggunakan Model Random Forest" .

Dalam kesempatan ini penulis tidak lupa menyampaikan ucapan terima kasih kepada semua pihak yang telah memberikan bantuan moral dan spiritual, langsung maupun tidak langsung dalam menyelesaikan buku ini. Semoga buku ini bisa bermanfaat bagi penulis dan para pembaca.

Daftar Isi

Daftar Gambar

Daftar Singkatan

OSEMN	= Obtain, Scrub, Explore, Model
OOP	= Object Oriented Programming
TP	= True Positif
TN	= True Negatif
FP	= False Positif
FN	= False Negatif
URL	= Uniform Resource Locator
CSV	= Comma Separated Values

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kriminalitas merupakan tindakan yang bisa merugikan jiwa orang lain dan pelaku itu sendiri. Kejahatan yang dilakukan sangat berdampak pada kehidupan sosial. Tindakan ini bisa menyebabkan seseorang mengalami trauma atau pun hal- hal yang lain.[1]

Kejahatan bisa seperti pencurian di pinggir jalan atau sama berbahayanya dengan pembunuhan, dan sangat sering sebagian besar kejahatan muncul dari berbagai penyebab seperti masa anak – anak yang bermasalah, pengaruh teman seumuran yang nakal, kemiskinan, pengangguran, penyalahgunaan narkoba, dan lain – lain.[2]

Pada tindak kejahatan suatu daerah, bisa dilakukan dengan memprediksi peramalan suatu kejadian, permalan merupakan upaya memperkirakan kemungkinan peristiwa yang mungkin terjadi selama beberapa jangka waktu. peramalan adalah tentang upaya memprediksi masa depan seakurat mungkin, mengingat semua informasi yang tersedia, dengan menggunakan data historis dan pengetahuan tentang masa depan ataupun segala peristiwa yang mungkin akan mempengaruhi peramalan. Peramalan juga dapat diartikan sebagai sebuah cara untuk mendapatkan interval berdasarkan data saat ini dan masa lalu dimana pada umumnya bertujuan untuk mengetahui tren masa depan. Hasil peramalan biasanya diterapkan untuk membantu proses pengambilan keputusan, perencanaan, dan alokasi sumber daya.[3]

Beberapa penelitian telah dilakukan untuk memprediksi jenis kejahatan, tingkat kejahatan dan hot spot kejahatan dengan menggunakan dataset kejahatan untuk wilayah yang berbeda, misalnya di Korea Selatan, dan Amerika Serikat, India. [4]

Tujuan penelitian ini dalam ruang lingkup jenis kejahatan pencurian menggunakan dataset Chicago, untuk mencari tau range kejahatan dari rendah sampai tinggi berdasarkan jumlah kejadian dan waktu kejadian menggunakan metode *OSEMN*.

Metode *OSEMN* adalah tahapan tahapan yang dilakukan mulai dari pengumpulan data sampai dengan penentuan model yang digunakan untuk melakukan tingkat accuracy data yang akan digunakan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, maka penulis merumuskan permasalahannya sebagai berikut :

> Identifikasi tingkat kejahatan menggunakan model Random Forest berdasarkan data pada website *data.cityofchicago.org*.

1.3 Tujuan

Adapun tujuan melakukan penelitian diantaranya :

> Untuk mengetahui tingkat kejahatan di suatu tempat tersebut menggunakan model Random Forest.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian tersebut bermanfaat untuk membantu masyarakat umum dan pihak keamanan dalam melakukan pengawasan terhadap beberapa tempat, terutama untuk tenpat yang memiliki tingkat kejahatan yang rendah sampai dengan tempat yang memiliki tingkat kejahatan tinggi

BAB II

LANDASAN TEORI

2.1 Penelitian Yang Berhubungan

Penelitian ini tidak terlepas dari hasil penelitian yang pernah dilakukan dan dijadikan sebagai tolak ukur serta acuan. Adapun hasil penelitian yang pernah dilakukan dan berhubungan dengan topik penelitian adalah sebagai beriku :

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1.	Raihan Virgatama, Andri Suprayogi dan Hana Sugiastu Firdaus	Identifikasi Pengaruh Sistem Keamanan Lingkunagn Terhadap Tingkat Kejahatan Pencurian di Kota Surakarta Dengan Metode Sistem Informasi Geografis	Penelitian ini Menggunakan metode sistem informasi geografis dimana datanya di *clustering* terlebih dahulu, kemudian setelah dilakukan clustering dan klasifikasi data, maka hasil pengolahan datanya dan koordinatnya ditampilkan menggunakan software ArcGIS 10.4 Berdasarkan jumlah kejadian tindak kejahatan pencurian yang terjadi di Kota Surakarta selama tahun 2016-2017, terdapat 228 kasus tindak pencurian yang terdiri dari 110 kasus pencurian pada tahun 2016 dan 118 kasus pada tahun 2017. Berdasarkan jenis kejadian tindak pencurian yang terjadi selama tahun 2016-

			2017 terdapat 32 kasus pencurian biasa, dan 196 kasus pencurian dengan pemberatan. Kecamatan Contoh Judul : Pembelajaran Abad 21: dari *E-learning* sampai *Blended Learning* Banjarsari adalah wilayah dengan jumlah tindak pencurian terbanyak yaitu 112 kasus yang terdiri pencurian biasa sebanyak 17 kasus, dan pencurian dengan pemberatan sebanyak 95 kasus. Sedangkan Kecamatan Pasar Kliwon dan Serengan merupakan wilayah dengan jumlah tindak pencurian paling sedikit, yaitu sembilan kasus. 2. Berdasarkan pengolahan data tindak kejahatan pencurian dengan pemilihan jarak 750m, dihasilkan data raster yang terbagi menjadi empat kelas tingkat kerawanan dimana wilayah yang tingkat kerawanannya sangat tinggi terletak pada Kecamatan Banjarsari di kawasan GOR Manahan. Sedangkan dari data fasilitas keamanan dengan pemilian jarak 1200m,

			dihasilkan data raster yang terbagi menjadi empat kelas ketersediaan fasilitas keamanan dimana wilayah yang fasilitas keamanannya sangat memadai terletak pada wilayah Kecamatan Pasar Bab 1 Pendidikan di Era Digital Kliwon yang merupakan kawasan pusat kota dan perkantoran di Kota Surakarta. 3. Berdasarkan analisis hubungan sistem keamanan lingkungan terhadap tindak kejahatan pencurian di Kota Surakarta dalam waktu dua tahun, dapat ditarik kesimpulan bahwa ketersediaan sistem keamanan lingkungan mempunyai pengaruh terhadap tingkat kerawanan pencurian di Kota Surakarta tetapi tidak mempunyai korelasi yang kuat, yaitu dengan nilai keofisiens korelasi sebesar 0,36982. [5]
2.	Wajiha Safat, Sohail Asghar, Saira Andleeb Gillani	Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques.	Berdasarkan hasil penelitian dan pembahasan, maka didapatkan kesimpukan sebagai berikut : Pada model Logistic Regression mendapatkan accuracy

			sebesar 90% pada dataset Chicago, sedangkan pada Los Angeles mendapatkan accuracy sebesar 48%. Pada model XGBoost, model KNN dan model MLP mendapatkan accuracy yang optimal.[6]
3.	Marchell Rianto, Roni Yunis	Analisis Runtun Waktu Untuk Memprediksi Jumlah Mahasiswa Baru Dengan Model Random Forest	Pada penelitian ini menjelaskan tentang peramalan prediksi jumlah mahasiswa yang mendaftar berdasarkan runtun waktu, dengan menggunakan metode random forest dapat memprediksi seberapa besar jumlah mahasiswa yang masuk berdasarkan tahun. Berdasarkan hasil analisis peramalan yang telah dilakukan maka dapat diperoleh kesimpulan sebagai berikut: Berdasarkan hasil penelitian didapatkan hasil forecasting dengan model random forest 3 memiliki variabel penting dalam prediksi yaitu Jumlah.Grade.B, Lalu parameter terbaik dari model terdapat pada maxnodes 100 dan ntree 900 yang menunjukkan parameter tersebut adalah parameter yang sangat

			akurat. Peramalan jumlah mahasiswa baru dalam beberapa tahun ke depan akan menurun dengan tingkat akurasi sebesar 98% karena nilai MSE dan MAE dari model sangat baik, yaitu di bawah 5%. Dari hasil peramalan disarankan bagi penelitian selanjutnya dapat mengkombinasi model Random Forest dengan model lain yang mendukung agar tingkat keakuratan dan visualisasi data yang lebih baik dan jelas lagi. [7]

2.2 Landasan Teori

2.2.1 Kriminalitas

Kriminalitas merupakan tindakan yang bisa merugikan jiwa orang lain dan pelaku itu sendiri. Kejahatan yang dilakukan sangat berdampak pada kehidupan sosial. Tindakan ini bisa menyebabkan seseorang mengalami trauma atau pun hal- hal yang lain. Kejahatan yang dilakukan bisa berupa perjudian ,perampokan, pembunuhan, dan lain-lain. Pelaku tindak pidana harus diproses oleh pihak yang berwajib dan dihukum sesuai dengan kejahatan yang dilakukan. Secara hukum, kejahatan didefinisikan sebagai tindakan atau kelalaian yang dilarang oleh hukum yang dapat dihukum dengan pidana penjara dan atau denda. Pembunuhan, perampokan, pencurian, pemerkosaan, mengemudi, mabuk, pembuangan anak dan sebagainya. [1]

2.2.2 Pencurian

Pencurian merupakan kejahatan yang ditujukan terhadap harta benda dan paling sering terjadi di dalam masyarakat. Kejahatan ini merupakan tindakan kejahatan yang dapat menggoncangkan stabilitas keamanan baik terhadap

harta maupun terhadap jiwa masyarakat. Oleh karena itu, baik dalam Kitab Undang - Undang Hukum Pidana (KUHP) maupun dalam Kitab Suci melarang keras tindakan kejahatan tersebut dan menegaskan ancaman hukuman secara rinci dan berat atas diri pelanggarnya. Hal ini dapat dilihat dari bentuk hukuman dan ancaman hukuman yang dijatuhkan.[8]

Adapun mengenai ancaman hukuman tentang kejahatan pencurian dalam hukum pidana positif di Indonesia diatur dalam Kitab Undang-Undang Hukum Pidana (KUHP) Buku Kedua Bab XXII tentang kejahatan terhadap harta benda dari Pasal 362 sampai dengan Pasal 367 KUHP. [8]

2.2.3 Python

Python merupakan bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintak kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar. Python mendukung konsep paradigma *object oriented programming* (OOP). pemrograman *imperatif*, dan pemrograman fungsional.

Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah:

a. Linux/Unix
b. Windows
c. Mac OS
d. Java Virtual Machine
e. Amiga
f. Palm
g. Symbian (untuk produk-produk Nokia)

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi, namun pada prinsipnya python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan baik menurut definisi *open source* maupun *General Public License* (GPL). [9]

Gambar 2. 1 Logo Python

Selain itu, python mempunyai beberapa kelebihan dibandingkan dengan bahasa pemrograman lain diantaranya :

a. Readability

Kelebihan yang pertama yaitu *readability* atau keterbacaan. Artinya bahasa pythob mudah dibaca karena memiliki *cript code* yang sederhana dan mudah ditulis. Hal ini akan memudahkan dalam tahap *development* aplikasi dan juga mudah dalam hal pemeliharaan.[10]

b. Efisien

Selain memiliki kelebihan keterbacaan yang baik, python juga memiliki banyak *library* yang lengkap sehingga membuat bahasa python menjadi efisien ketika membutuhkan sebuah *library* yang dibutuhkan.[10]

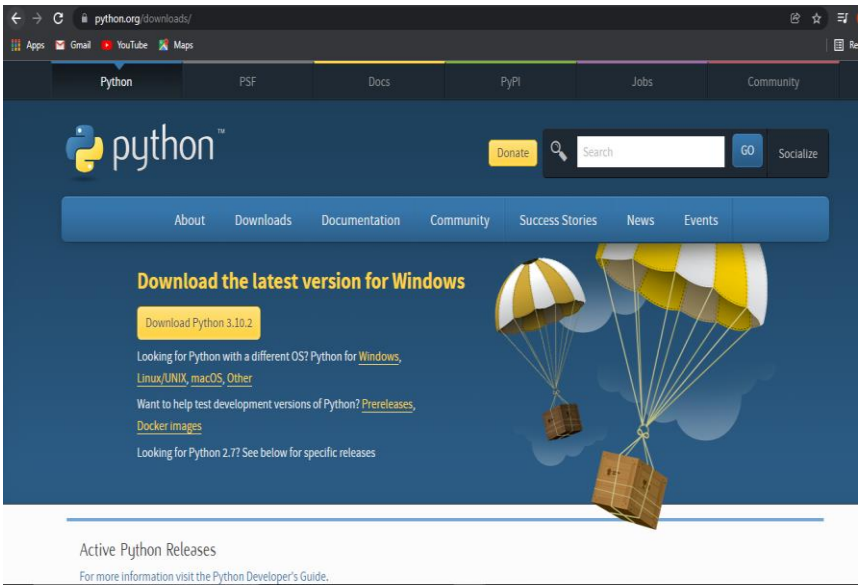
c. Multifungsi

Python merupakan bahasa pemrograman multifungsi, yaitu python bisa digunakan untuk berbagai keperluan dalam pembuatan dan pengembangan aplikasi, seperti pembuatan *website* , pembuatan aplikasi *machine learning*, dan pembuatan *internet of think's*.[10]

2.2.4 Instalasi Python Menggunakan Installer

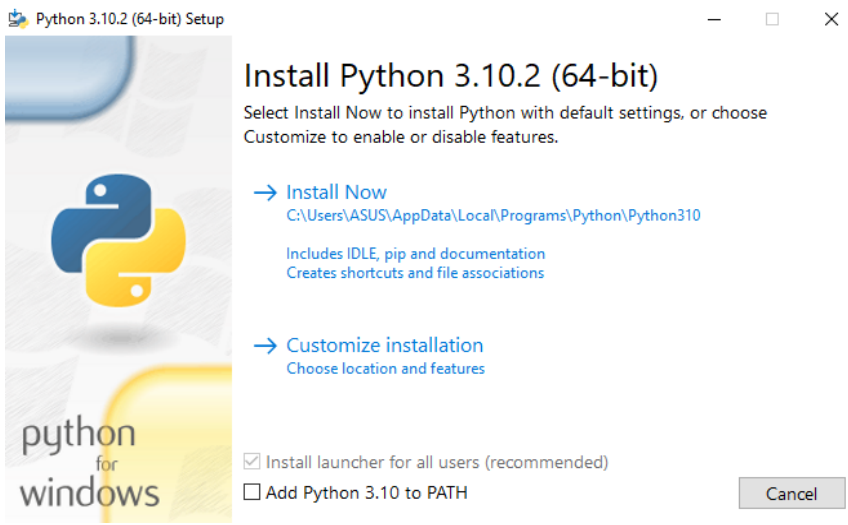
Cara melakukan instalasi python dengan installer yaitu :

1. Kunjungi situs https://www.python.org/downloads/. Kemudian unduh installer python tersebut, untuk versi terbaru pada saat buku ini dibuat terdapat pada versi 3.10.2.



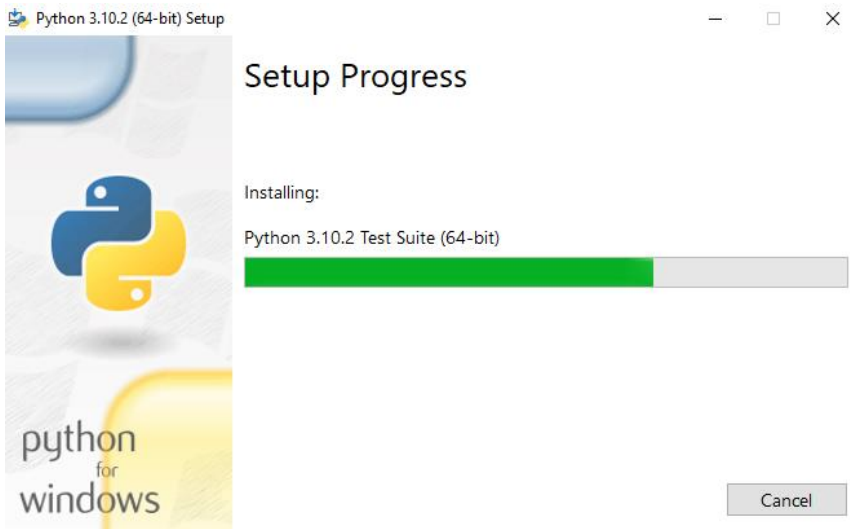
Gambar 2. 2 Website Resmi Python

2. Setelah di download, klik hasil download python dan tampilannya akan seperti ini. Kemudian klik install now.



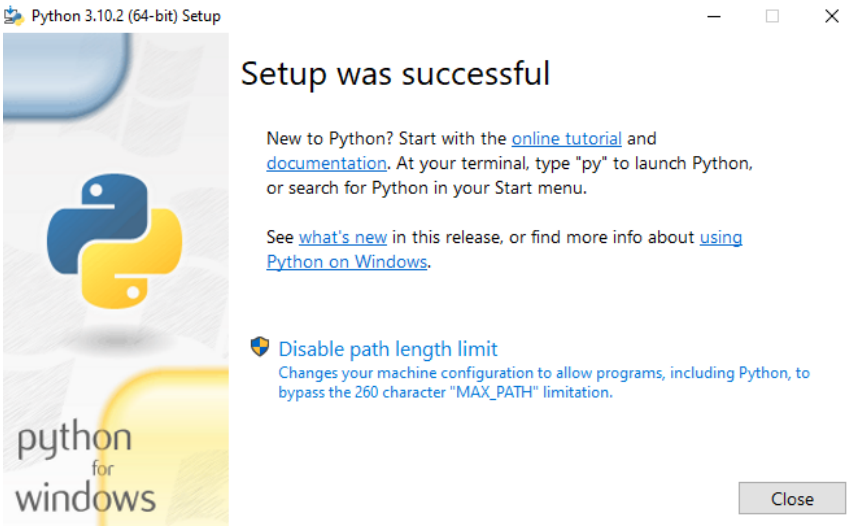
Gambar 2. 3 Instalasi Python Dengan Installer

3. Kemudian akan muncul tampilan seperti ini, tunggu hingga instalasi python selesai.



Gambar 2. 4 Instalasi Python Dengan Installer

4. Instalasi python telah berhasil, kemudian klik close.



Gambar 2. 5 Instalasi Python Dengan Installer

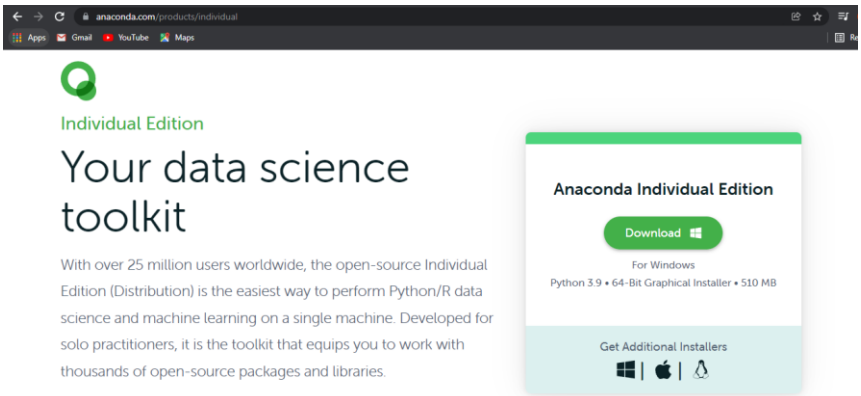
2.2.5 Instalasi Python dengan Anaconda

Cara selanjutnya yaitu dengan menggunakan paket distribusi Anaconda, keuntungan dengan menggunakan Anaconda yaitu telah tersedianya beberapa tools yang digunakan untuk menjalankan python seperti Spyder dan Jupyter Notebook. Sehingga tidak harus menginstalnya secara terpisah.

Untuk Anaconda sendiri tersedia dalam bentuk gratis sehingga memudahkan dalam penggunaaan python. Selain itu, Anaconda tersedia dalam beberapa sistem operasi seperti Windows, Linux dan Mac OS.

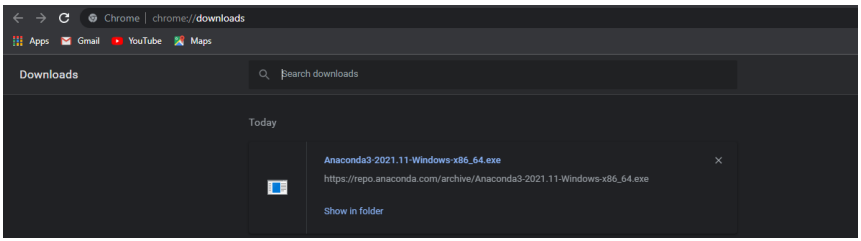
Cara – cara melakukan instalasi Anaconda diantaranya :

1. Masuk ke situs https://www.anaconda.com/products/individual, kemudian klik *Download*.



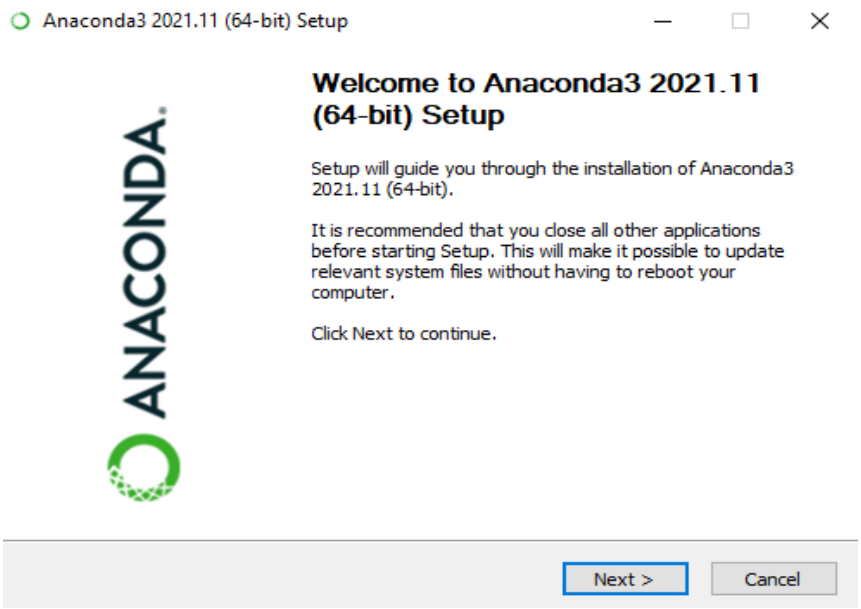
Gambar 2. 6 Website Anaconda

2. Selanjutnya cek hasil download pada folder Downloads, atau bisa juga dengan mengklik ctrl + j untuk melihat hasil download nya.



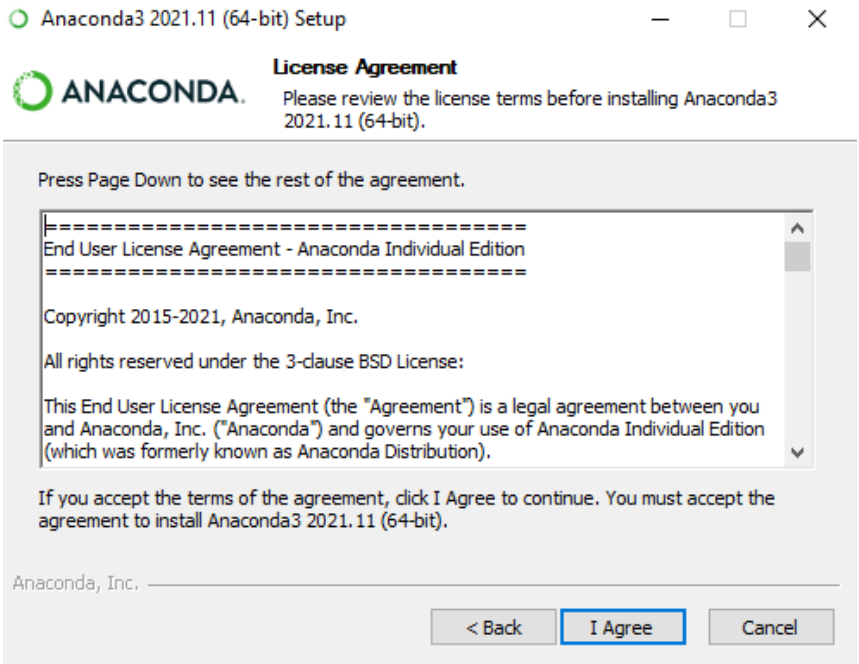
Gambar 2. 7 Hasil Download Anaconda

3. Selanjutnya klik hasil download Anaconda nya sampai muncul tampilan seperti ini. Kemudian klik ***Next***.



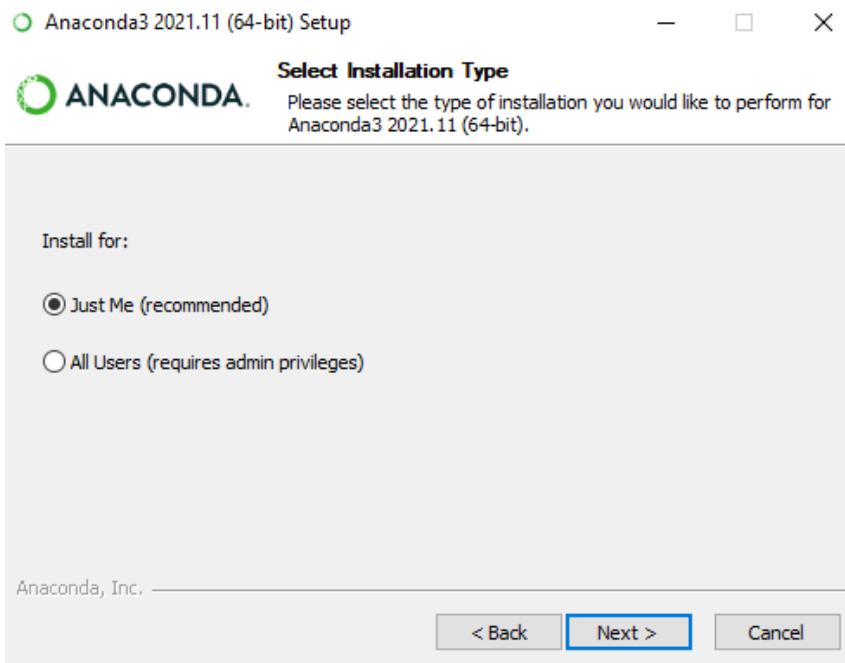
Gambar 2. 8 Instalasi Anaconda

4. Kemudian bacalah poin-poin terlebih dahulu, jika setuju, *klik* ***I Agree***



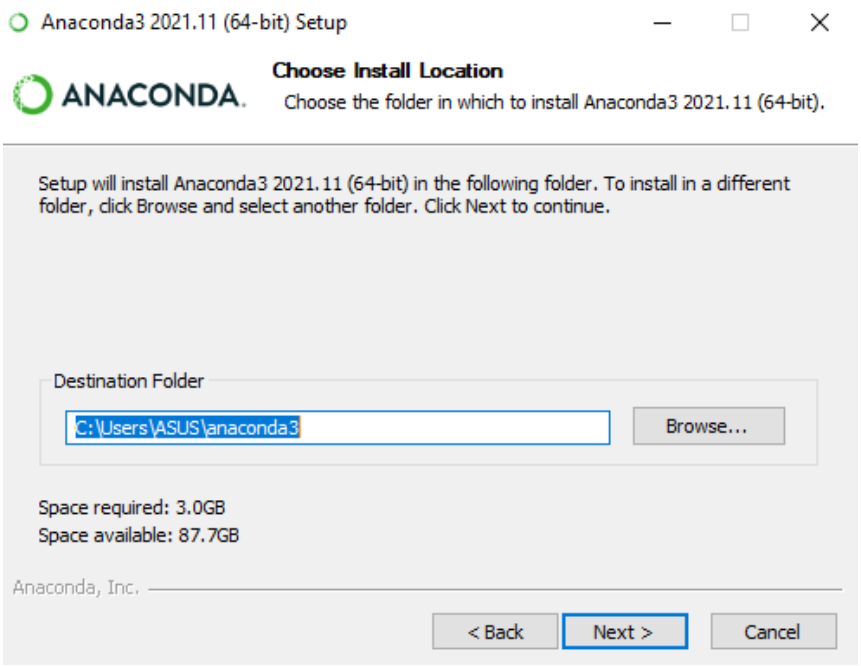
Gambar 2. 9 Instalasi Anaconda

5. Selanjutnya disini terdapat 2 kotak dialog, di proses instalasi disini menggunakan ***Just Me (recommended)*** sesuai dengan rekomendasi pada Anaconda tersebut. klik **Next.**



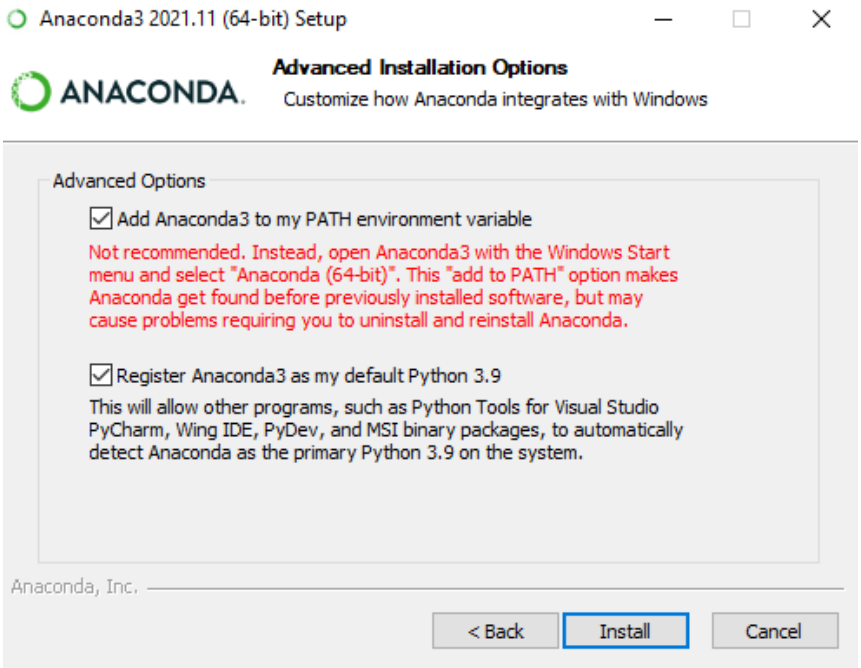
Gambar 2. 10 Instalasi Anaconda

6. Selanjutnya disini muncul ***Destination Folder*** yang bisa digunakan untuk mengatur tempat folder itu disimpan, tetapi disini saya menggunakan versi defaultnya saja seperti pada gambar. Kemudian klik ***Next.***



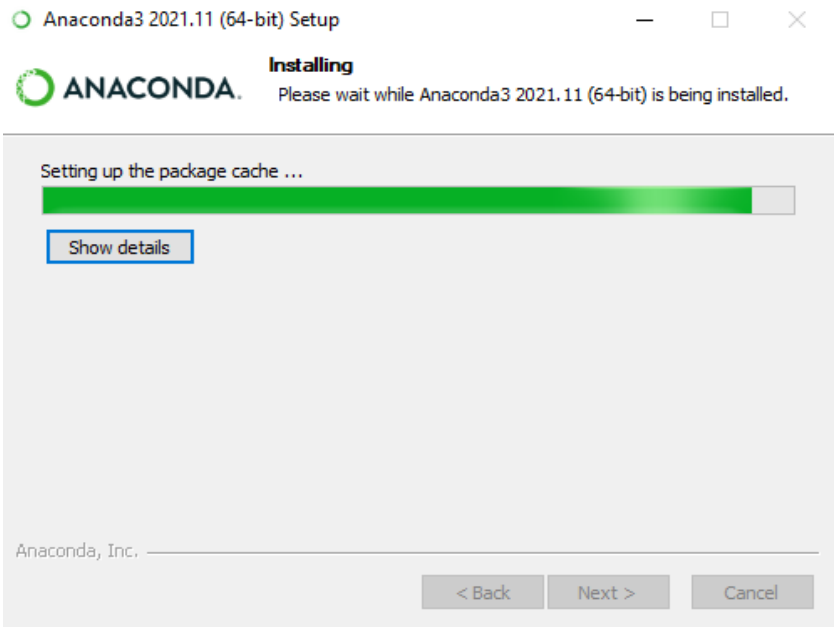
Gambar 2. 11 Instalasi Anaconda

7. Selanjutnya akan muncul ***Advanced Instalation Options,*** disini berilah tanda ceklis untuk kedua form tersebut, tujuannya agar tidak perlu menginstall lagi path yang dibutuhkan, kecuali path tersebut tidak tersedia pada saat proses instalasi. Klik ***Install.***



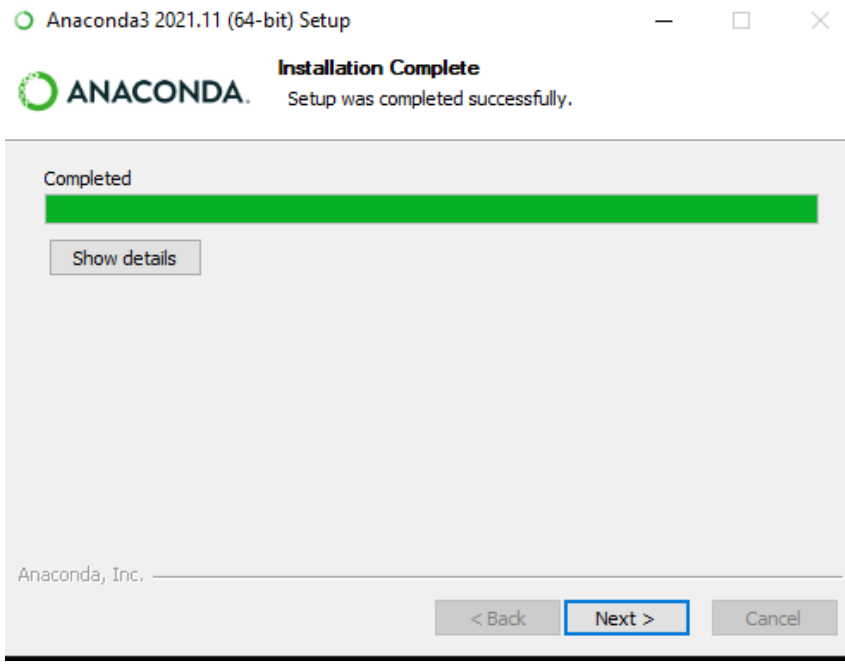
Gambar 2. 12 Instalasi Anaconda

8. Proses instalasi berjalan, tunggu hingga instalasi selesai.



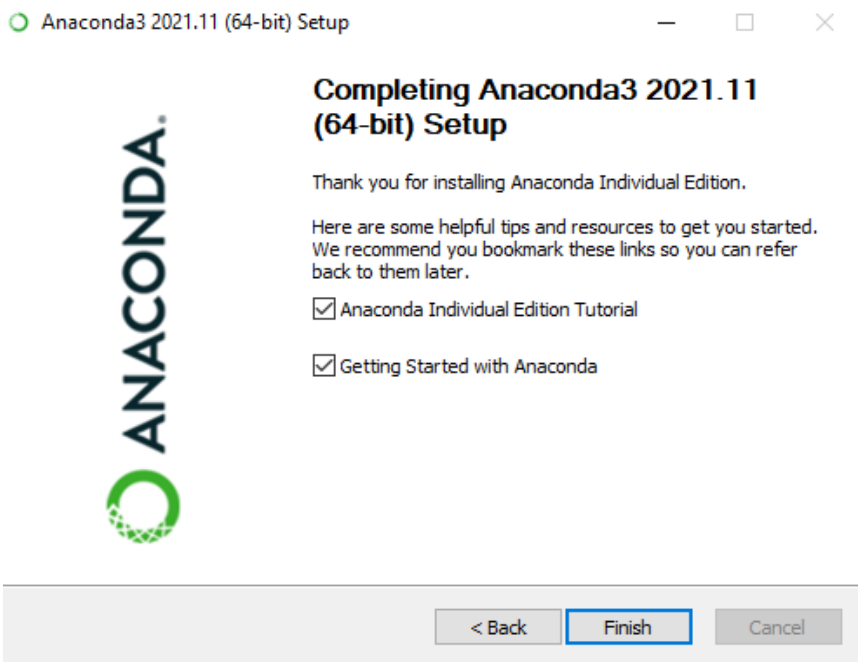
Gambar 2. 13 Instalasi Anaconda

9. Setelah prosesya ***Completed***. Klik ***Next.***



Gambar 2. 14 Instalasi Anaconda

10. Proses Instalasi Anaconda selesai. Klik ***Finish***



Gambar 2. 15 Instalasi Anaconda Selesai

2.2.6 Machine Learning

Maching Learning adalah cabang aplikasi dari *Artificial Intelligence* (Kecerdasan Buatan) yang fokus pada pengembangan sebuah sistem yang mampu belajar "sendiri" tanpa harus berulang kali di program oleh manusia. Aplikasi *Machine Learning* membutuhkan data sebagai bahan belajar (*training*) sebelum mengeluarkan *output*. Aplikasi sejenis ini juga biasanya berada dalam domain spesifik alias tidak bisa diterapkan secara general untuk semua permasalahan. [11]

Machine learning mempunyai fokus pada pengembangan sistem yang mampu belajar sendiri untuk memutuskan sesuatu tanpa harus berulang kali di program oleh manusia. Hal tersebut menjadikan mesin tidak hanya mampu berperilaku mengambil keputusan, namun juga dapat beradaptasi dengan perubahan yang terjadi. *Machine learning* bekerja apabila tersedia data sebagai

input untuk dilakukan analisis terhadap kumpulan data besar (*big data*) sehingga menemukan pola tertentu. [12]

Sejarah *Machine Learning*

Arthur Samuel, seorang pionir dalam pengembangan permainan komputer dan kecerdasan buatanlah yang pertama kali mengeluarkan istilah "*Machine Learning*" ke publik pada tahun 1959. Perkembangan pembelajaran mesin tumbuh berkat berkembangnya bidang kecerdasan buatan atau *Artificial Intelegence* (AI). Banyak peneliti di bidang AI tertarik untuk memiliki mesin yang dapat belajar dari data. Para peneliti berusaha untuk mendekati masalah dengan berbagai metode simbolik, serta apa yang kemudian disebut *Neural Network*, Penalaran probabilistik dan berbagai model statistik. [13]

Era sebelum 1920an Thomas Bayes, Adrien-Marie Legendre, Andrey Markov, dan matematikawan lainnya memulai penelitian yang mejadi teknik dasar pada konsep-konsep ML. Sejarah *Machine learning* dapat dikatakan dimulai oleh penelitian Thomas Bayes pada tahun 1763 yang dipublikasikan oleh temannya Richard Prince. Penelitian tersebut disempunakan oleh Pierre Simon Laplace sehingga sekarang dikenal dengan Theorema Bayes (1812). Selanjutnya penelitian penting lainnya dilakukan oleh Adrien-Marie Legendre matematikawan asal Prancis, dimana dia mengembangkan metode Least Squares yang bertujuan untuk melakukan data fitting menggunakan pendekatan aljabar (1805), serta Andrey Markov mendeskripsikan sebuah teknik yang ia gunakan untuk menganalisis puisi. Teknik tersebut dikenal dengan Markov Chains (1913). Teori-teori mereka tersebut menjadi landasan utama untuk *machine learning*. [13]

Era 1940-1950an Pada tahun 1943, McMulloh dan Pitts mengembangkan model matematis yang terinspirasi dari bagaimana otak manusia bekerja, dimana dalam model tersebut terdapat neuron-neuron yang dapat menjadi aktif atau nonaktif seperti saklar on-off. Selain itu neuron-neuron tersebut memiliki kemampuan untuk belajar dan menghasilkan respons yang berbeda berdasarkan input yang diberikan. Sumbangan besar lainnya diberikan oleh Alan Turing, pada tahun 1950 yang menciptakan mesin turing untuk menjawab pertanyaan "Dapatkan Komputer Berfikir?". Paper Alan Turing pada tahun 1950 berjudul "*Computing Machinery* and *Intelligence*" mendiskusikan syarat sebuah mesin dianggap cerdas. Dia beranggapan bahwa jika mesin dapat dengan sukses berprilaku seperti manusia, kita dapat menganggapnya cerdas. Pengembangan selanjutnya, pada tahun 1959, Allen

Newell dan Herbert Simon mengembangkan sebuah proyek disebut General Problem Solver (GPS). GPS telah berhasil menyelesaikan permasalahan manusia menggunakan teknik mean-ends analysis. [13]

Era 1960-1990 an Selanjutnya penemuan fenomenal lainnya pada tahun 1962, dimana Frank Rosenblatt membuktikan teorema Perceptron dan berhasil mengklasifikasi objek dari dua kelompok menggunakan classification rules. Perkembangan selanjutnya dibuat oleh Rumelhart (1986) dengan mencoba mengembangkan sistem layar tunggal (single layer) pada Perceptron menjadi sistem layar jamak (multilayers), yang kemudian disebut dengan sistem back-propagation. Konsep back-propagation telah ada sebelumnya, namun Rumelhart dan Hinton-lah yang pertama kali mempopulerkan dan membuktikan metode ini dapat dilakukan. Setelah itu, muncul beberapa model jaringan saraf tiruan lain yang dikembangkan oleh Kohonen (1972), Hopfield (1982), dan lain-lain. Selain algoritma berbasis Neural Network, di Rusia berkembang algoritma-algoritma Generalized Portrait pada tahun 1960-an (1963, 1964). Algoritma tersebut berakar dari teori pembelajar statistika yang telah dikembangkan selama 3 dekade oleh Vapnik dan Chervonenkis. SVM (Support Vector Machines) diperkenalkan oleh Vapnik, Boser, dan Guyer dalam konferensi COLT tahun 1992 dalam sebuah jurnal dan berkembang dengan sangat cepat. SVM pertama kali diperkenalkan ke dunia sejak akhir tahun 1970-an (Vapnik,1979), Support Vector Machines tidak mendapat banyak perhatian oleh para peneliti. Kemudian Vladimir Vapnik kembali menerbitkan bukunya pada tahun 1990-an (Vapnik, 1995; Vapnik, 1998). Sejak saat itu, SVM menjadi metode yang popular dan aktif diteliti pada bidang Machine Learning.[13]

Era 2000 an di akhir 90 - an hingga pertengahan tahun 2000 an, neural networks sempat "nyaris dilupakan" dikarenakan muncul berbagai algoritma seperti Support Vector Machines, AdaBoost yang dapat dieksekusi lebih cepat dengan performa yang lebih baik pada waktu itu. Neural networks kembali mendapatkan perhatian ketika Deep Belief Networks (DBN) membuat terobosan dengan menjadi model handwritten digit recognition yang paling akurat, yang pada akhirnya memunculkan istilah Deep Learning. Istilah Deep Learning makin popular dengan diperkenalkannya Convolution Neural Network (CNN). CNN dengan arsitektur tertentu yang dipadukan dengan berbagai trik (misalnya, dropout regularization, pemanfaatan *Rectified Linear Unit* (ReLU) sebagai fungsi aktivasi, data augmentation) sehingga telah mampu melakukan klasifikasi pada data gambar yang berjumlah sangat besar

(ImageNet). ImageNet yang memiliki 1000 kategori objek dan dengan jumlah 1 juta gambar, hal ini melebihi performa manusia. [13]

Machine learning memiliki 3 jenis teknik yaitu teknik *supervised learning* , *unsupervised learning* dan *reinforcement learning*. Beberapa praktisi dari *machine learning* menggunakan *supervised learning*. *Supervised learning* adalah salah satu tipe *machine learning* yang menggunakan dataset yang dikenal (*training dataset*) untuk membuat prediksi. *Unsupervised learning* adalah salah satu *tipe machine learning* yang digunakan untuk menarik kesimpulan dari datasets yang terdiri dari *input data labeled response*.[14].

Reinforcement learning yaitu teknik ini bekerja dalam lingkungan yang *dinamis* di mana konsepnya harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai. [15]

Machine learning berdasarkan bagaimana cara kerjanya dapat dikolompokkan menjadi dua yaitu :

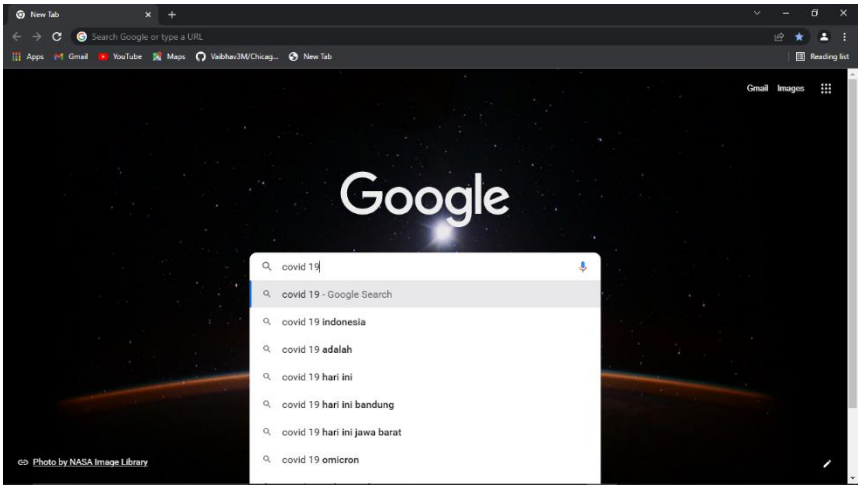
a. *Instance-based learning* (atau sering disebut *memory-based learning*) adalah sebuah kelompok model *machine learning* yang bekerja dengan membandingkan data *testing* dengan data yang telah dipelajari pada proses *training*.
b. *Model based learning* adalah kebalikan dari *intance-based* dimana menggunakan memori untuk melakukan pemecahan masalah, algoritma ini membuat sebuah model yang bersifat generik. [13]

Machine learning dan ilmu statistik sangat erat kaitannya, suatu ide dalam *machine learning* berawal dari prinsip – prinsip metodologis ke alat teoritis, dan memiliki sejarah keterkaitan yang erat dengan ilmu statistik.[16]

Machine leaning tersebut dimanfaatkan dalam beberapa aplikasi yang digunakan diantaranya :

a. Web Searching

Pada saat melakukan pencarian pada *website* pencarian, sepert google, Mozilla, opera dan sebagainya. Maka *website* tersebut akan menampilkan halaman web yang paling relevan dan mendekati dengan pencarian yang diinginkan. Dalam hal ini *machine learning* bekerja membantu menangani data yang besar dengan cara yang cerdas, dalam hal ini untuk memberikan hasil pencarian yang tepat kepada pengguna web tersebut..[10]



Gambar 2. 16 Machine Learning Pada Web Searching

b. Filter Spam

Machine learning berperan sangat besar untuk melakukan penyaringan (filter) pesan spam (menganalisa, menilai dan menyaring spam berdasarkan isi) terutama yang sering di terjadi pada email. model pohon keputusan (*decision tree*) merupakan cikal bakal dari algoritma spam filtering, untuk mentukan suatu pesan termasuk spam atau bukan. [13]

c. Penerapan pada *Marketplace*

Pada penerapannya dengan memprediksi pergerakan dari pelanggan, dari mulai melakukan pencarian sampai dengan mendeteksi validasi dari sebuah transaksi. Sehingga pada saat kegiatan tersebut bisa memunculkan sebuah

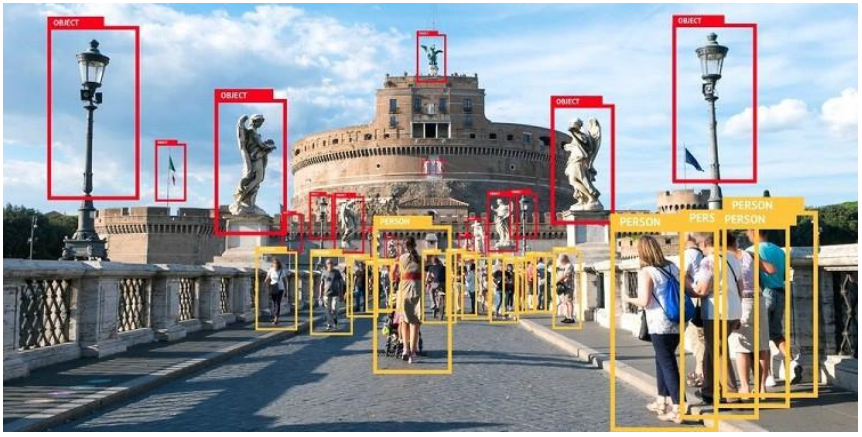
rekomendasi produk sehingga sistem bisa memberikan rekomendasi produk kepada setiap akun *marketplace*. [13]

d. Virtual Assistant

Berbagai teknologi seperti *handphone* dan laptop telah dilengkapi dengan *virtual assistant*, misalnya: Cortana di Microsoft Windows, Siri di Iphone, dan Google Now di Android. *Virtual assistant* dapat membantu penggunanya untuk melakukan pencarian di internet, menanyakan jalan, melihat prakiraan cuaca, melakukan panggilan telepon dan lain – lain. Di Windows, Cortana bahkan mempelajari penggunanya untuk memberikan rekomendasi perintah yang akan dilakukan. [13]

e. Image Recognition

Machine learning juga bisa berfungsi sebagai fitur untuk mengenali gambar yang diberikan. Pada *image recognition* bisa mengenali berbagai macam objek yang tersedia dalam sebuah gambar seperti berikut. [10]



Gambar 2. 17 (Sumber : www.bing.com)

f. Mobil Kendali Otomatis

Mobil kendali otomatis merupakan penerapan serta pengembangan dari machine learning yaitu machine vision. Mobil kendali otomatis merupakan penerapan machine learning yang kompleks dan dengan resiko langsung yang tinggi. Banyak hal yang harus dipelajari oleh mobil, mulai dari rambu-rambu

lalulintas, arah dan tujuan, kondisi jalan, traffic light, kondisi manusia sekitarnya, dan sensor lainnya yang terintegrasi. [17]

g. Videos Surveillance

Videos surveillance atau pengawasan video merupakan teknologi baru yang merupakan penerapan dari *machine learning* yang disematkan pada CCTV untuk mendeteksi suatu tindak kejahatan atau kecelakaan. Di negara-negara maju, CCTV sudah digunakan untuk melakukan pencarian penjahat yang masih buron. [17]

h. Online Fraud Detection

Online fraud detection adalah metode yang digunakan untuk mendeteksi suatu transaksi digital sah atau tidak. *Online fraud detection* digunakan oleh semua bank baik bank umum maupun bank virtual seperti paypal. *Online fraud detection* menggunakan *machine learning* untuk melakukan perlindungan (*cybersecurity*) terhadap pencucian uang, pendeteksian transaksi palsu, hingga deteksi pembobolan akun bank *digital*. [17]

2.2.7 Data Science

Data science atau ilmu data adalah suatu disiplin ilmu yang khusus mempelajari data, khususnya data kuantitatif (data numerik), baik yang terstruktur maupun tidak terstruktur. Bidang ilmu data telah muncul dalam menanggapi peningkatan jumlah data. Sejumlah besar data telah tersedia untuk orang-orang di semua lapisan masyarakat, melalui jejaring sosial, perangkat seluler, dan berbagai perangkat sensor Internet of Things. [18]

Berikut beberapa pilar yang ada pada data *science*

a. Bisnis

Seorang data scientist yang mengolah data berdasarkan ilmu data science harus bisa mengolah data menjadi informasi yang bisa dipahami untuk membantu perancangan strategi guna menyelesaikan masalah bisnis. Untuk bisa melakukan ini, keahlian data *science* pun harus disertai pemahaman bisnis sehingga penyelesaian yang diusulkan berdasarkan data mungkin untuk dilakukan sebuah bisnis untuk mencapai tujuannya. [19]

b. Matematika dan Statistika

Data science sangat membutuhkan ilmu matematika, karena data harus diolah secara kuantitatif. Banyak permasalahan dalam bisnis yang dapat diselesaikan dengan membuat model analitik dengan dasar matematika. Untuk membuatnya, dibutuhkan pemahaman matematika yang mendalam.

Contohnya, model untuk merancang *machine learning* sebagai salah satu aplikasi ilmu data science sangat lekat dengan matematika. Statistik untuk data science adalah hal yang tak kalah penting. Tidak hanya mengerti statistika klasik, seorang data scientist juga perlu memahami statistika Bayes. [19]

c. Teknologi

Data science tidak bisa lepas dari teknologi dan kreativitas serta kecerdasan dalam menggunakan keahlian teknis untuk menyelesaikan suatu permasalahan. Data science adalah keilmuan yang menggunakan data dalam jumlah besar dan model yang rumit, sehingga butuh keahlian ilmu komputer yang mendalam. Seorang data scientist perlu menguasai bahasa pemrograman seperti SQL, Phyton, R, SAS, Java, Scala, Julia, dan masih banyak lagi. Seorang data scientist harus mampu berpikir layaknya algoritma dalam memecahkan permasalahan yang paling sulit sekalipun. [19]

Selanjutnya ada beberapa penerapan yang digunakan dalam data *science* diantaranya :

a. Bidang Kesehatan

Pada tahun 2008, data *science* membuat tanda besar pertamanya di industri perawatan kesehatan. Staff Google menemukan bahwa mereka dapat memetakan wabah flu secara real time dengan melacak data lokasi pada pencarian terkait flu. Peta CDC yang ada tentang kasus flu yang terdokumentasi, FluView, diperbarui hanya sekali seminggu. Google dengan cepat meluncurkan alat yang bersaing dengan pembaruan yang lebih sering: Google Flu Trends. Tapi itu tidak berhasil. Pada tahun 2013, Google memperkirakan sekitar dua kali kasus flu yang benar-benar diamati. Meski begitu, ini menunjukkan potensi serius ilmu data dalam perawatan kesehatan.[20]

b. Bidang Keuangan

perusahaan berbasis *finance* atau keuangan kini menggunakan data *science* untuk mengklasifikasikan, mengelompokkan, dan menyegmentasikan data yang mungkin menandakan pola penipuan. Hal ini diperlukan guna menghindari terjadinya proses kriminalisasi berkelanjutan di dalam sistem milik perusahaan. Sistem pakar dalam perusahaan finansial juga bisa mengkodekan data yang mampu mendeteksi penipuan dalam bentuk-bentuk yang tak terduga. [19]

2.2.8 Data Analystic

Data Analytics merupakan proses memeriksa kumpulan data untuk menarik kesimpulan tentang informasi yang dikandungnya. *Data Analystic* digunakan dalam industri komersial untuk memungkinkan organisasi membuat keputusan bisnis yang lebih tepat dan cepat. Oleh para ilmuwan atau peneliti untuk memverifikasi atau menyangkal model, teori, dan atau hipotesis ilmiah.[21]

2.2.9 Scikit Learn

Sklearn merupakan library open-source yang dapat digunakan sebagai Machine Learning pada bahasa pemrograman python . scikit-learn atau sklearn adalah modul untuk bahasa pemograman python yang dibangun diatas numpy, scipy, dan matplotlib, fungsinya dapat membantu melakukan processing data ataupun melakukan *training* data untuk kebutuhan *machine learning* Sklearn merupakan *library* simple dan efisien unruk melakukan data mining serta analisis data . Sklearn memiliki fitur regresi , klasifikasi dan pengelompokkan model contohnya seperti support vector machine , nearest neighbors, naive bayes dan lain – lain . library Scikit-learn dibangun menggunakan librray lainnya seperti Numpy, Scipy dan matplotlip. Ada banyak fitur yang dapat digunakan dengan sklearn ini, seperti *Classification*, *Regression*, *Clustering*, *Dimensionality reduction*, Model selection, dan *Preprocessing* data [11]

Gambar 2. 18 Logo Scikit Learn

2.2.10 Confusion Matrix

Confusion Matrix adalah tabel yang digunakan untuk menggambarkan kinerja model klasifikasi pada set data pengujian yang nilainya telah diketahui.

Confusion matrix sendiri relatif sederhana dan mudah untuk mengerti, tetapi terminologi yang terkaitdapat membingungkan apabila belum memahami konsep dari *confusion matrix*. [22]

2.2.11 Balance Dataset

Balance Dataset adalah *dataset* yang berisi jumlah sampel yang sama atau hampir sama dari kelas positif dan negative. [23]

2.2.12 Imbalace Dataset

Imbalance Dataset adalah salah satu dari dua kelas lebih tinggi dari yang lain, dengan cara lain, jumlah pengamatan tidak sama untuk semua kelas dalam dataset klasifikasi. [23]

2.2.13 Visualisasi Data

Visualisasi adalah suatu metode untuk memrepresentasikan suatu data atau permasalahan ke dalam format grafik atau bentuk gambar yang mudah untuk dipahami. Visualisasi data dalam bentuk gambar dan grafik akan memberikan kemudahan dalam membaca dan memahami data tersebut. Salah satu bentuk visualisasi yaitu pemetaan. [24]

Visualisasi data merupakan era baru yang merupakan sumber yang muncul dari kecerdasan, perkembangan teoretis dan kemajuan dalam pencitraan

multidimensi dengan membentuk kembali nilai potensial yang analitik dan wawasan dapat memberikan peran visualisasi. [25]

2.2.14 Numpy

Numpy merupakan *library* python yang berfokus pada *scientific computing*. Numpy mempunyai kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada Python. Keunggulan Numpy array dibandingkan dengan list pada python yaitu konsumsi memory yang lebih kecil serta runtime yang lebih cepat. [26]

Numpy mempunyai beberapa fungsi diantaranya :

a. Numpy Array

Untuk menggunakan numpy, pertama – tama *import* terlebih dahulu *library* yang dibutuhkan.

```
import numpy as np
```

Gambar 2. 19 Mengimport Library Numpy

Kemudian untuk membuat array, gunakan fungsi array(). Didalam *library* Numpy, terdapat *upcasting*, yaitu ketika tipe data *element* array tidak sama, dilakukan penyamaan tipe data pada yang lebih tinggi.

Berikut contoh numpy array.

```
import numpy as np
x = np.array([[1,2,3], [7,8,9]])
x
```

```
array([[1, 2, 3],
       [7, 8, 9]])
```

Gambar 2. 20 Contoh Numpy Array

Selanjutnya untuk mengecek tipe array gunakan perintah type() yang diikuti oleh nama variable yang digunakan.

Berikut untuk memanggil tipe numpy array.

```
import numpy as np
x = np.array([[1,2,3], [7,8,9]])
type(x)
```

```
numpy.ndarray
```

Gambar 2. 21 Cek Type Numpy Array

Numpy array adalah objek ndarray, yang merupakan singkatan dari *n-dimensional array*.

Selanjutnya untuk mengecek sebuag tipe data element pada array, gunakan fungsi dtype().

```
import numpy as np
x = np.array([[1,2,3], [7,8,9]])
x.dtype
```

```
dtype('int32')
```

Gambar 2. 22 Cek Type Element Array

Disini terlihat bahwa untuk tipe data array tersebut merupakan *integer*.

Numpy mempunyai fungsi shape yang berfungsi untuk menghasilkan sebuah tuple yang berisikan panjang sebuah array pada tiap dimensi. Jadi artinya fungsi shape ini bisa menghitung jumlah baris pada array.

Numpy bisa digunakan untuk membuat array multi dimensi, contohnya membuat array 2 dimensi seperti ini.

```
import numpy as np
x = np.array([[1,2,3], [7,8,9]])
x
```

```
array([[1, 2, 3],
       [7, 8, 9]])
```

Gambar 2. 23 Array 2 Dimensi

2.2.15 Matplotlib

Matplotlib merupakan *library* python yang berfokus pada visualisasi data seperti membuat plot grafik. Matplotlib pertama kali diciptakan oleh John D. Hunter dan sekarang telah dikelola oleh tim developer yang besar. Matplotlib dapat digunakan dalam code python, IPython shell, server aplikasi web, dan beberapa toolkit graphical user interface (GUI) lainnya.[26]

Visualisasi dari matplotlib merupakan sebuah gambar grafik yang terdapat satu sumbu atau lebih. Setiap sumbu memiliki sumbu horizontal (x) dan sumbu vertikal (y), dan data yang direpresentasikan menjadi warna dan glyphs seperti marker (contohnya bentuk lingkaran) atau lines (garis) atau poligon [26].

Gambar 2. 24 Logo Matplotlib

Untuk menggunakan *library* pada matplotlib gunakan perintah seperti berikut.

```
import matplotlib.pyplot as plt
```

Gambar 2. 25 Import Library Matplotlib

Matplotlib mempunyai *magic command* %matplotlib inline, untuk pengaturan pada backend matplotlib agar setiap grafik ditampilkan secara *inline*, yaitu akan ditampilkan langsung pada cell notebook.

Berikut perintahnya seperti ini.

```
import matplotlib.pyplot as plt
%matplotlib inline
```

Gambar 2. 26 Penggunaan Magic matplotlib

Berikut beberapa diagram yang bisa dibuat pada matplotlib.

a. Membuat Bar Plot

Bar plot berfungsi untuk membandingkan perubahan tiap waktu pada beberapa kelompok data. Bar plot sangat bagus digunakan dalam visualisasi ketika perubahan data sangat besar dibandingkan dengan line plot. Bar plot biasanya memiliki dua sumbu yaitu sumbu x untuk jenis kelompok dan sumbu y untuk proporsi kelompok. Matplotlib menyediakan fungsi bar() untuk mempermudah dalam visualisasi bar plot.

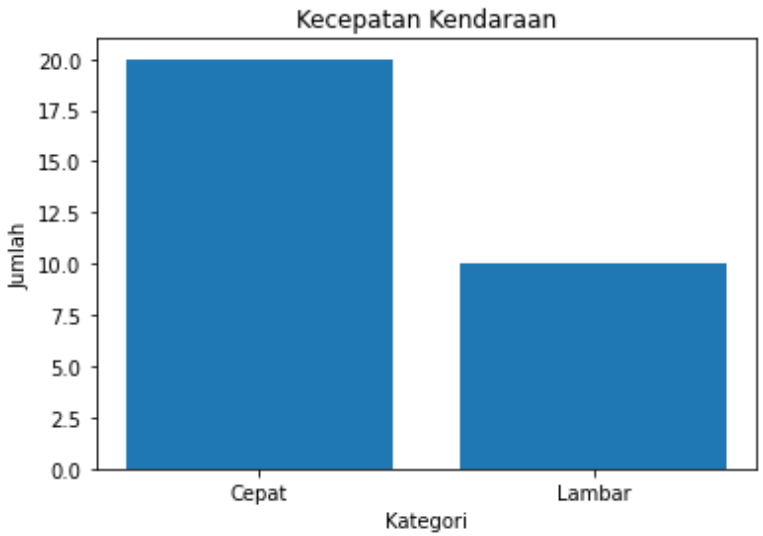
Berikut perintah untuk menampilkan bar plot.

```
import matplotlib.pyplot as plt
%matplotlib inline

x = ['Cepat', 'Lambar']
y = [20,10]

fig, ax = plt.subplots()
ax.bar(x, y)
ax.set_xlabel('Kategori')
ax.set_ylabel('Jumlah')
ax.set_title('Kecepatan Kendaraan')
```

Gambar 2. 27 Perintah Membuat Bar Plot

Maka akan menampilkan bar plot seperti ini.

```
Text(0.5, 1.0, 'Kecepatan Kendaraan')
```



Gambar 2. 28 Gambar Bar Plot

b. Scatter Plot

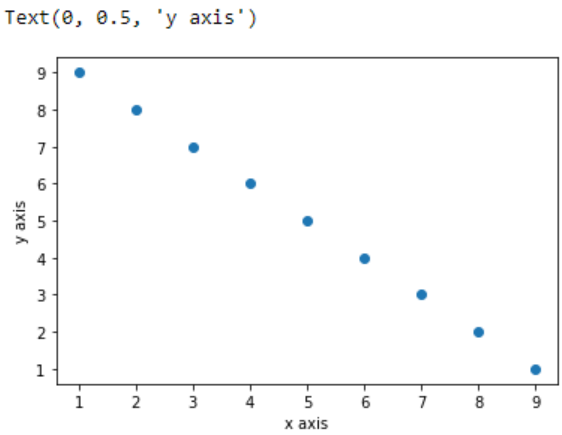
Scatter plot berfungsi untuk melakukan observasi dan menunjukkan hubungan relasi antara dua variabel *numeric*. Titik-titik pada scatter plot juga dapat menggambarkan pola dari data secara keseluruhan. Matplotlib menyediakan fungsi scatter() untuk mempermudah dalam visualisasi scatter plot.

Berikut perintah untuk menampilkan scatter plot.

```
import matplotlib.pyplot as plt
%matplotlib inline

x = np.array([1,2,3,4,5,6,7,8,9])
y = np.array([9,8,7,6,5,4,3,2,1])

fig, ax = plt.subplots()
ax.scatter(x,y)
ax.set_xlabel('x axis')
ax.set_ylabel('y axis')
```

Gambar 2. 29 Perintah Menampilkan Scatter Plot

Berikut tampilan dari scatter plot.



Gambar 2. 30 Tampilan Scatter Plot

c. Line Plot

Line plot berguna untuk melacak perubahan pada periode waktu pendek dan panjang. Ketika terdapat perubahan kecil, line plot lebih baik dalam melakukan visualisasi dibandingkan grafik bar. [26]

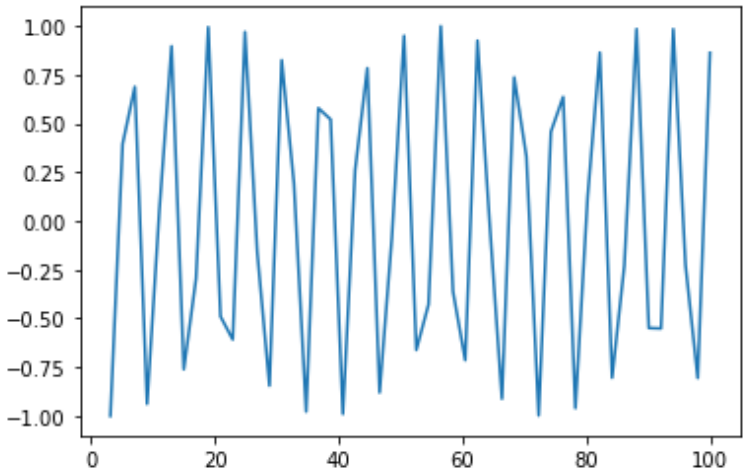
Berikut perintah untuk menampilkan *line plot*.

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

x = np.linspace(1*np.pi, 100)
y = np.cos(x)

fig, ax = plt.subplots()
ax.plot(x, y)
```

Gambar 2. 31 Perintah Menampilkan Line Plot

Berikut tampilan dari *line plot.*

```
[<matplotlib.lines.Line2D at 0x1e16e2388e0>]
```



Gambar 2. 32 Tampilan Line Plot

2.2.16 Pandas

Pandas merupakan *library* python yang berfokus untuk proses analisis data seperti manipulasi data, persiapan data, dan pembersihan data. Pandas

menyediakan struktur data dan fungsi high-level untuk membuat pekerjaan dengan data terstruktur/tabular lebih cepat, mudah, dan ekspresif. [26]

Pandas memadukan *library* numpy yang memiliki kemampuan manipulasi data yang fleksibel dengan database relasional (seperti SQL). Sehingga memudahkan kita untuk melakukan *reshape*, *slice* dan dice, agregasi data, dan mengakses subset dari data.

Untuk menggunakan *library* pandas gunakan perintah seperti ini.

```
import pandas as pd
```

Gambar 2. 33 Mengimport Library Pandas

Series adalah objek 1-dimensi yang berisi *sequence* nilai dan berasosiasi dengan label data, yang disebut indeks. Berikut contoh penggunaan series.

```
import pandas as pd
x = pd.Series([1,2,3,4,5,6,7,8,9])
x
```

Gambar 2. 34 Penggunaan Fungsi Series

Berikut tampilan fungsi series.

```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
7    8
8    9
dtype: int64
```

Gambar 2. 35 Tampilan Fungsi Series

Selanjutnya untuk mengecek type dari series pandas gunakan perintah type().

```
import pandas as pd
x = pd.Series([1,2,3,4,5,6,7,8,9])
type(x)
```

```
pandas.core.series.Series
```

Gambar 2. 36 Cek Tipe Pandas Series

DataFrame merupakan objek yang memiliki struktur data tabular, berorientasi pada kolom dengan label baris dan kolom. Series adalah objek array 1-dimensi yang memiliki label. DataFrame merupakan tabel data yang terdapat kolom dan baris, dimana nilai-nilai yang terdapat di dalamnya dapat berupa tipe berbeda seperti *numeric*, *string*, *boolean*, dan sebagainya.. Dataframe mirip dengan data 2 dimensi dengan adanya baris dan kolom. Selain itu, dataFrame bisa dikatakan gabungan dari *dictionary* objek series yang memiliki *indeks* yang sama. Terdapat berbagai macam cara untuk membentuk objek DataFrame. Salah satu cara yang biasa dilakukan untuk membentuk objek DataFrame dengan menggunakan data masukan berupa *dictionary* [26]

Berikut contoh dalam penggunaan DataFrame.

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
       'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
       'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.info()
```

Gambar 2. 37 Penggunaan DataFrame

Maka hasil tampilan dari dataframe nya seperti berikut.

	warna	sepatu	tahun
0	merah	nike	2017
1	kuning	adidas	2018
2	hijau	puma	2019
3	biru	kompas	2020
4	ungu	swallow	2021

Gambar 2. 38 Tampilan DataFrame

Selanjutnya ada fungsi shape yaitu fungsi untuk mengetahui jumlah baris dan kolom dari DataFrame.

Berikut contoh penggunaan shape.

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
      'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
      'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.shape
```

```
(5, 3)
```

Gambar 2. 39 Penggunaan Fungsi Shape dan Hasilnya

Selanjutnya ada fungsi info() yang berfungsi untuk mengetahui keterangan dari objek DataFrame yang dibuat seperti index dari DataFrame lengkap dengan range dari index, jumlah kolom beserta informasi tiap kolom untuk null data dan tipe data, dan jumlah total penggunaan memory pada tiap kolom dalam satuan bytes.

Berikut contoh dalam penggunaan fungsi info().

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
      'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
      'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.info()
```

Gambar 2. 40 Penggunaan Fungsi Info

Berikut tampilan hasil fungsi info() pada DataFrame.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   warna   5 non-null      object
 1   sepatu  5 non-null      object
 2   tahun   5 non-null      int64
dtypes: int64(1), object(2)
memory usage: 248.0+ bytes
```

Gambar 2. 41 Tampilan Info DataFrame

Maka hasilnya bisa didapat seperti pada diatas, dimana terdapat info tipe data apa saja yang digunakan, diatas didapat bahwa tipe data yang digunakan yaitu object dan integer.

Selanjutnya fungsi describe() yang berfungsi mengetahui statistika data untuk data *numeric* seperti *count*, *mean*, *standard deviation*, *maximum*, minimum, dan *quartile*. Untuk data *string*, misalkan data tersebut adalah kategori, maka dapat menggunakan fungsi value_counts() untuk mengetahui jumlah tiap kategori pada data.

Berikut contoh penggunaan fungsi *describe*.

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
       'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
       'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.describe()
```

Gambar 2. 42 Penggunaan Fungsi Describe

Berikut tampilan dari fungsi *describe*().

	tahun
count	5.000000
mean	2019.000000
std	1.581139
min	2017.000000
25%	2018.000000
50%	2019.000000
75%	2020.000000
max	2021.000000

Gambar 2. 43 Tampilan Fungsi Describe

Selanjutnya fungsi value_counts() yang berfungsi untuk mengetahui jumlah tiap kategori pada data.

Berikut contoh penggunaan value_counts().

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
      'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
      'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.value_counts()
```

Gambar 2. 44 Penggunaan Value_Counts

Maka tampilan dari fungsi value_counts seperti berikut.

```
warna   sepatu   tahun
biru    kompas   2020     1
hijau   puma     2019     1
kuning  adidas   2018     1
merah   nike     2017     1
ungu    swallow  2021     1
dtype: int64
```

Gambar 2. 45 Tampilan Fungsi Value Counts

Selanjutnya untuk mengakses DataFrame pada suatu data, alamat data tersebut adalah pada nama kolom sebagai petunjuk lokasi kolom dan indeks sebagai petunjuk lokasi baris.

Berikut contoh untuk mengakses DataFrame.

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
      'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
      'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data['sepatu']
```

Gambar 2. 46 Mengakses DataFrame

Maka hasilnya seperti berikut.

```
0       nike
1     adidas
2       puma
3     kompas
4    swallow
Name: sepatu, dtype: object
```

Gambar 2. 47 Tampilan Akses DataFrame

Sedangkan mengakses data pada baris tertentu, kita menggunakan fungsi loc[indeks]. Indeks disini menunjukan baris pada DataFrame.

Berikut contoh penggunaan fungsi loc.

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
       'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
       'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.loc[4]
```

Gambar 2. 48 Penggunaan Fungsi Loc

Berikut tampilan fungsi loc

```
warna        ungu
sepatu    swallow
tahun        2021
Name: 4, dtype: object
```

Gambar 2. 49 Tampilan Fungsi Loc

DataFrame bisa diakses lebih dari 1 baris dengan menggunakan titik dua ':', seperti ingin mengakses indeks 2–3, maka menggunakan perintah loc[2:3].

Berikut contoh penggunaan fungsi loc dengan akses lebih dari 1 baris.

```
import pandas as pd
data = { 'warna' : ['merah', 'kuning', 'hijau', 'biru', 'ungu'],
       'sepatu' : ['nike', 'adidas', 'puma', 'kompas', 'swallow'],
       'tahun' : [2017, 2018, 2019, 2020, 2021]
}

main_data = pd.DataFrame(data)
main_data.loc[2:3]
```

Gambar 2. 50 Penggunaan Loc Lebih Dari 1 Baris

Berikut tampilan penggunaan loc lebih dari 1 baris.

	warna	sepatu	tahun
2	hijau	puma	2019
3	biru	kompas	2020

Gambar 2. 51 Tampilan Loc Lebih dari 1 Baris

2.2.17 Seaborn

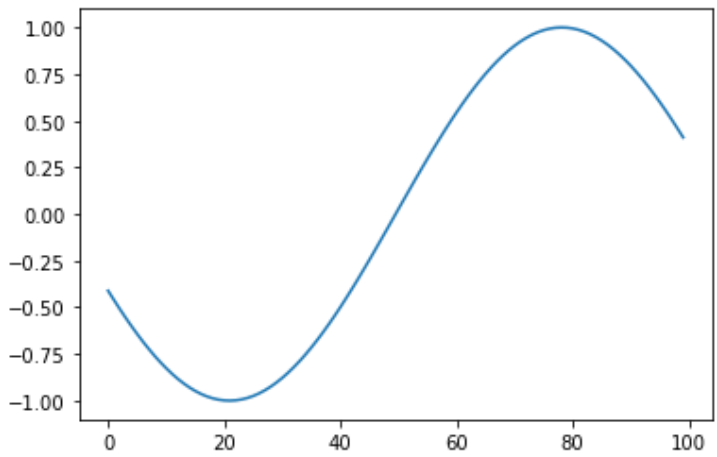
Seaborn berfungsi untuk memproduksi visualisasi dengan Python dan memiliki beberapa kelebihan dibandingkan dengan Matplotlib yaitu, hasil visualisasi Seaborn diklaim lebih bagus dan indah juga menggunakan serangkaian kode yang lebih mudah.[27]

Untuk menggunakan seaborn yaitu harus menggunakan library numpy, pandas dan matplotlib.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

data = np.linspace(-np.e, np.e, 100, endpoint=True)

plt.plot(np.sin(data))
plt.show()
```

Gambar 2. 52 Perintah Menggunakan Seaborn

Berikut tampilan dari seaborn.



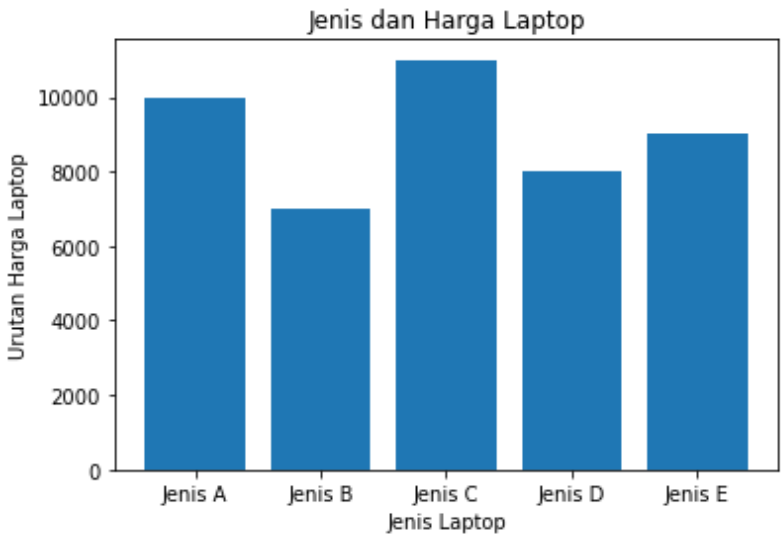
Gambar 2. 53 Tampilan Seaborn

Selanjutnya bandingkan tampilan bar plot dengan matplotlib dan seaborn, maka hasilnya seperti berikut.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

laptop = ["Jenis A", "Jenis B", "Jenis C", "Jenis D", "Jenis E"]
harga_laptop = [10000,7000,11000 ,8000, 9000]

plt.bar(laptop, harga_laptop)
plt.title("Jenis dan Harga Laptop")
plt.xlabel("Jenis Laptop")
plt.ylabel("Urutan Harga Laptop")
plt.show()
```

Gambar 2. 54 Bar Plot Dengan Matplotlib

Maka hasil tampilan bar plot dengan matplotlib seperti berikut.



Gambar 2. 55 Tampilan Bar Plot Dengan Matplotlib

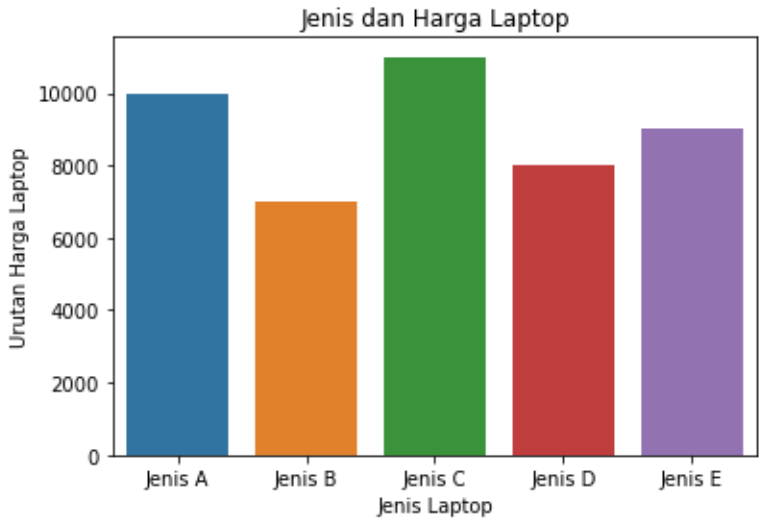
Selanjutnya perintah bar plot dengan seaborn maka seperti ini.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

laptop = ["Jenis A", "Jenis B", "Jenis C", "Jenis D", "Jenis E"]
harga_laptop = [10000,7000,11000 ,8000, 9000]

sns.barplot(x=laptop, y=harga_laptop)
plt.title("Jenis dan Harga Laptop")
plt.xlabel("Jenis Laptop")
plt.ylabel("Urutan Harga Laptop")
plt.show()
```

Gambar 2. 56 Bar Plot Dengan Seaborn

Berikut tampilan dari bar plot dengan seaborn.

Gambar 2. 57 Tampilan Bar Plot Dengan Matplotlib



Gambar 2. 58 Tampilan Bar Plot Seaborn

Terdapat perbedaan antara seaborn dengan matplotlib dimana jika menggunakan matplotlib, tampilan bar plot hanya berwarna biru sedangkan

dengan seaborn terlihat bahwa tampilan bar plotnya lebih berwarna, tidak hanya warna biru tetapi terdapat dengan warna lain.

2.2.18 Classification Report

Classification Report berfungsi untuk mengukur kualitas prediksi dari model yang digunakan. Ada prediksi yang benar dan ada juga prediksi yang salah. Lebih lanjut, ada prediksi dengan hasil *True Negatif*, *True Positif*, *False Negatif*, dan *False Positif* digunakan untuk memprediksi metric classification report. Beberapa penjelasan terkait True Negatif, True Positif, False Negatif dan False Positif.

a. TN / *True Negative* = ketika sebuah kasus negatif dan diprediksi negatif
b. TP / *True Positive*: ketika kasus positif dan diprediksi positif
c. FN / *Negatif Palsu*: ketika sebuah kasus positif tetapi diprediksi negatif
d. FP / *False Positive*: ketika suatu kasus negatif tetapi diprediksi positif. [28]

2.2.19 Jupyter Notebook

Gambar 2. 59 Logo Jupyter Notebook

Jupyter notebook merupakan aplikasi berbasis web yang bisa digunakan untuk membuat dan membagikan dokumen. Dokumen tersebut berisi kode, persamaan matematika, visualisasi data maupun text. Jupyter notebook ini di kelola oleh orang-orang yang tergabung pada Project Jupyter.

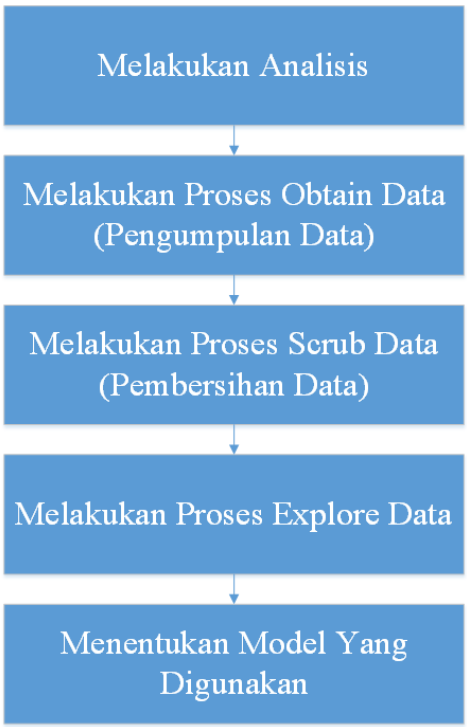
Jupyter notebook merupakan project spin-off dari IPython, yang pada mulanya memiliki proyek tersendiri yaitu Notebook IPyhton. Memiliki nama Jupyter karena dapat mendukung bahasa pemrograman Julia, Python dan R. Jupyter disajikan dengan kernel IPython, sehingga memungkinkan untuk menulis program dengan menggunakan Python. Namun, pada saat ini ada lebih dari 100 kernel lainnya yang dapat digunakan. [29].

BAB III

METODE PENELITIAN

3.1 Metodologi Penelitian

Pada penelitian ini, penulis menggunakan metode *OSEMN* yang digunakan untuk mendapatkan analisis terbaik dari data yang disajikan. Untuk menyelesaikan permasalahan yang ada, diperluakan suatu metode penelitian sebagai berikut :



Gambar 3. 1 Metodologi Penelitian

3.2 Tahapan – Tahapan Diagram Alur Metodologi Penelitian

Tahapan – tahapan dari metodologi penelitian dapat diuraikan dan dijelaskan lebih detail seperti berikut :

3.2.1 Melakukan Analisis

Pada tahap ini, akan dibahas mengenai penelitian yang dilakukan, dari analisis kebutuhan data. Untuk data disini menggunakan data yang disediakan dari *website* Chicago. Berikut *link url* nya ((https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/data).

Kemudian melakukan proses pengumpulan data dengan cara mengunduh data pada *website* tersebut. setelah itu masuk ke proses *scrub data* (pembersihan data) , proses pembersihan data dilakukan dengan 3 tahapan yaitu menghilangkan data kosong, melakukan rekayasa fitur dan melakukan kompres data.

Kemudian melakukan explore data pada data yang ada untuk digunakan sesuai dengan kebutuhan.

Kemudian melakukan penentuan model untuk mencari tingkat akurasi dari model yang digunakan. Pada tahap ini menggunakan model *random forest.*

3.2.2 Obtain Data (Mengumpulkan Data)

Dalam bidang *data science*, proses mengumpulkan data wajib dan harus menggunakan data yang usable atau dapat digunakan untuk tujuan analisis dengan melihat jenis dan bentuk datanya, cara memperoleh datanya dan sumber data.[30]

Dalam tahap awal melakukan penelitian. penulis mengidentifikasi dan mengumpulkan data yang tepat dalam penelitian. Jika tidak memiliki data maka tidak ada yang dapat dilakukan. Banyak cara untuk mendapatkan dataset yaitu dari situs kaggle.com, data.cityofchicago.org, dan masih banyak lagi repository dataset lainnya. Dalam penelitian ini, penulis menggunakan dataset yang disediakan oleh data.cityofchicago.org yang membahas tentang criminal.

3.2.3 Scrub Data (Pembersihan Data)

Scrubbing Data adalah kegiatan melakukan konversi atau merapikan data dari satu format ke format lain dan menggabungkan semuanya ke dalam satu format standar di agar memberikan gambar yang akurat pada hasil akhir.[7]

Tahapan selanjutnya yaitu melakukan *scrub data* (pembersihan data) dimana pada bagian ini penulis menguraikan dan beberapa subbab diantaranya :

a. *Getting rid of missing value* (menghilangkan nilai yang kosong)
   Pada tahap ini akan memeriksa data dan membersihkan data dari data yang tidak diperlukan dengan menggunakan Bahasa pemrograman Python.

b. *Feature Engineering* (Rekayasa Fitur)
   Pada tahap ini peneliti akan merekayasa beberapa fitur yang sesuai dengan bahasa yang digunakan, disini menggunakan Python sebagai bahasa pemrograman yang digunakan.

c. *Compressing the Dataset* (Melakukan kompres data)
   Tahapan selanjutnya yaitu melakukan kompres dataset yang didapat menjadi dataset yang baru.[2]

3.2.4 Explore

Explore data adalah langkah di mana kita membiasakan diri dengan data dan memungkinkan kita untuk mencari tahu sub kumpulan data mana yang akan digunakan untuk pemodelan lebih lanjut serta membantu dalam pembuatan hipotesis untuk dieksplorasi.[7]

3.2.5 Model

Pada Tahap akhir ini dilakukan proses *modeling* data dengan menggunakan beberapa model yang digunakan untuk mengklasifikan data. Tahap *Modelling* data adalah proses memvisualisasikan, mengelompokkan, dan melakukan pengurangan dimensi model dari data. Dengan melihat pola dan tren data yang unik dapat dijadikan model terbaik di mana model terbaik adalah model yang prediktif hasilnya.[30]

3.2.6 Random Forest

Random Forest merupakan salah satu *ensamble learning* yang dibangun dari pohon keputusan . Beberapa keuntungan menggunakan pendekatan *ensamble learning* adalah dapat digunakan untuk kasus klasifikasi dan regresi, mampu memperoleh akurasi tinggi, cocok untuk analisis ukuran dataset yang besar

dengan banyak dimensi. Selain itu ensamble learning seperti Random Forest juga cocok digunakan untuk menangani *imbalanced* data.[31]

Random forest mempunyai beberapa keunggulan,yaitu dapat meningkatkan akurasi apabila terdapat data yang hilang serta untuk resisting outliers, dan juga efisien untuk penyimpanan data. Tidak hanya itu, pada Random Forest terdapat proses seleksi fitur dimana mampu mengambil fitur terbaik sehingga meningkatkan performa pada model klasifikasi. Dengan adanya fitur seleksi tentunya Random Forest mampu bekerja pada data yang besar dengan parameter yang kompleks secara efektif. Selain itu, Random Forest juga mampu bekerja secara parallel yang dikenal dengan multiple random forest. Namun, Random Forest terkadang memiliki nilai yang tidak diharapkan dan juga tidak mempredikasi rangedari responsenilai pada data latih. [32]

Random Forest adalah model yang dapat meningkatkan hasil akurasi karena dalam membangkitkan simpul anak untuk setiap node dilakukan secara acak. Algoritma klasifikasi Random Forest merupakan pengembangan dari Decision Tree, dimana menghasilkan pohon gabungan yang memberikan tingkat akurat yang lebih tinggi dibandingkan dengan pohon tunggal. [33]

3.3 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan pada penelitian ini, penulis menggunakan data yang terdapat pada web Chicago. Dataset yang berisi tentang data criminal yang dilakukan pada daerah tersebut.

3.4 Metode Analisis Data

Analisis data kuantitatif merupakan sebuah teknik analisis yang digunakan pada data kuantitatif. Data kuantitatif merupakan data yang dapat dibentuk dengan simbol angka atau bilangan. Metode ini merupakan pendekatan pengolahan data melalui metode statistik atau matematik yang terkumpul dari data sekunder. Kelebihan dari metode ini adalah kesimpulan yang lebih terukur dan komprehensif. Hasil dari analisis kuantitatif biasanya dalam bentuk angka yang kemudian akan diinterpretasikan dalam uraian-uraian kalimat yang dapat dipahami oleh pengguna. Analisis data dalam penelitian ini biasanya menggunakan dua macam teknik analisis statistik, yaitu statistik deskriptif dan statistik inferensial.

Metode analisis data dalam penelitian kuantitatif menggunakan statistik. Metode analisis data kuantitatif adalah metode yang bergantung kepada kemampuan untuk menghitung data secara akurat. Selain itu, metode ini juga memerlukan kemampuan untuk menginterpretasikan data yang kompleks. Beberapa contoh metode analisis kuantitatif, seperti analisis deskriptif, regresi, dan faktor. Metode analisis data kuantitatif mempunyai berbagai macam jenis analisis seperti teknik korelasional, regresi, komparasi, deskriptif dan sejenisnya. [34]

BAB IV

ANALISIS HASIL DAN PEMBAHASAN

4.1 Analisis

Pada analisis ini akan menjelaskan tentang Analisis tentang tingkat kejahatan menggunakan model *random forest*. Pada tahapan penelitian dilakukan analisis kebutuhan data, selanjutnya melakukan analisis data yang digunakan pada penelitian. Pada analisis data yang digunakan, dilakukan proses klasifikasi data dari urutan jenis kejahatan rendah – tinggi, menentukan data apa saja yang digunakan.

obtain data (pengumpulan data) yang dimana data tersebut didapatkan dalam *website* Chicago yaitu (https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/data).

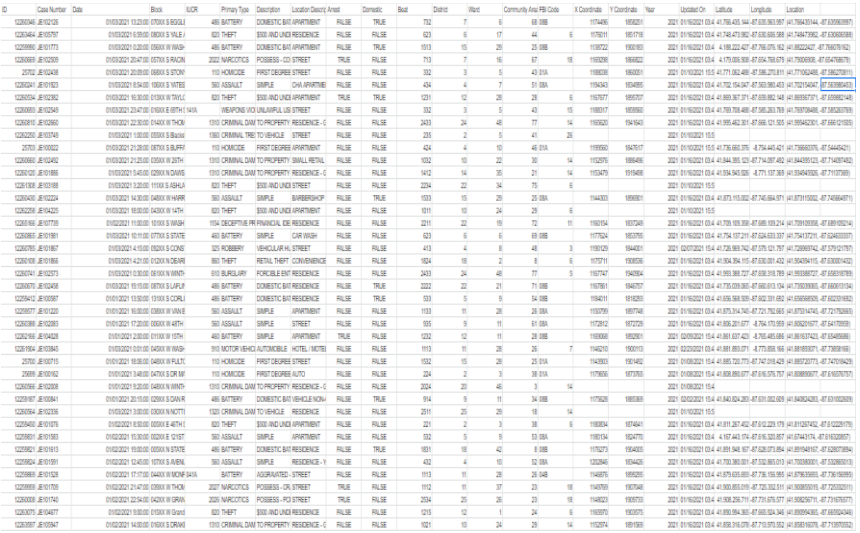
Setelah itu masuk ke proses *scrub data* (pembersihan data) , proses pembersihan data dilakukan dengan 3 tahapan yaitu menghilangkan data kosong, melakukan rekayasa fitur dan melakukan kompres data.

Kemudian melakukan explore data pada data yang ada untuk digunakan sesuai dengan kebutuhan.

Kemudian melakukan penentuan model untuk mencari tingkat akurasi dari model yang digunakan. Pada tahap ini menggunakan model *random forest*.

4.4.1 Data Yang Digunakan

Data yang digunakan pada penelitian ini menggunakan dataset kejahatan yang berada di daerah chicago. Untuk mendapatkan dataset tersebut bisa mengunjungi web **https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/data** . Data yang digunakan penulis data rentan tahun 2019 – 2021 yang sudah dalam bentuk kompres data. Berikut gambaran dari tampilan data yang digunakan.



ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Descrip	Arrest	Domestic	Beat	District	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location
12382348	JE107235	01/03/2021 13:23:00	070XX S EGGLI	486	BATTERY	DOMESTIC BAT	APARTMENT	FALSE	TRUE	732	7	6	68	08B	1174406	1856251	2021	01/16/2021 03:	41.766.435.144	-87.635.563.997	(41.766435144, -87.635563997)
12383464	JE105707	01/03/2021 6:09:00	080XX S VALE	820	THEFT	$500 AND UNDE	RESIDENCE	FALSE	FALSE	623	6	17	44	6	1176011	1851716	2021	01/16/2021 03:	41.748.473.992	-87.639.606.588	(41.748473992, -87.639606588)
12385990	JE101773	01/03/2021 0:20:00	056XX W WASH	486	BATTERY	DOMESTIC BAT	APARTMENT	FALSE	TRUE	1513	15	29	25	08B	1139722	1900183	2021	01/16/2021 03:	4.188.222.427	-87.756.976.162	(41.88222427, -87.756976162)
12385669	JE102509	01/03/2021 20:47:00	057XX S RACIN	2022	NARCOTICS	POSSESS - CO	STREET	TRUE	FALSE	713	7	16	67	18	1169206	1866822	2021	01/16/2021 03:	4.179.026.936	-87.654.768.079	(41.79006936, -87.654768079)
25702	JE102438	01/03/2021 20:09:00	068XX S STONY	110	HOMICIDE	FIRST DEGREE	STREET	FALSE	FALSE	332	3	5	43	01A	1188308	1860051	2021	01/16/2021 15:5	41.771.082.488	-87.586.270.811	(41.771082488, -87.586270811)
12385241	JE101823	01/03/2021 6:54:00	100XX S YATES	560	ASSAULT	SIMPLE	CHA APARTME	FALSE	FALSE	434	4	7	51	08A	1194343	1834085	2021	01/16/2021 03:	41.702.154.567	-87.563.560.453	(41.702154567, -87.563560453)
12383754	JE102362	01/03/2021 16:30:00	013XX W TAYLO	820	THEFT	$500 AND UNDE	APARTMENT	TRUE	TRUE	1231	12	28	28	6	1167577	1895707	2021	01/16/2021 03:	41.869.367.391	-87.659.862.148	(41.869367391, -87.659862148)
12383850	JE102549	01/03/2021 23:47:00	016XX E 69TH (	1410	WEAPONS VIO	UNLAWFUL USE	STREET	FALSE	FALSE	332	3	5	43	15	1185317	1858582	2021	01/16/2021 03:	41.769.758.488	-87.585.263.769	(41.769758488, -87.585263769)
12382810	JE102660	01/03/2021 22:30:00	014XX W THOM	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - G	FALSE	FALSE	2433	24	49	77	14	1165520	1941643	2021	01/16/2021 03:	41.9955.482.301	-87.668.121.565	(41.995482301, -87.668121565)
12382250	JE101749	01/03/2021 1:00:00	055XX S Blackst	1360	CRIMINAL TRES	TO VEHICLE	STREET	FALSE	FALSE	235	2	5	41	26			2021	01/16/2021 15:5			
25703	JE100022	01/03/2021 21:28:00	087XX S BURY	110	HOMICIDE	FIRST DEGREE	APARTMENT	FALSE	FALSE	424	4	10	46	01A	1199550	1847017	2021	01/16/2021 15:5	41.736.693.376	-8.764.465.421	(41.736693376, -87.64465421)
12380660	JE102402	01/03/2021 21:25:00	035XX W 26TH	1310	CRIMINAL DAM	TO PROPERTY	SMALL RETAIL	FALSE	FALSE	1032	10	22	30	14	1152976	1886486	2021	01/16/2021 03:	41.844.395.123	-87.714.097.482	(41.844395123, -87.714097482)
12381208	JE101886	01/03/2021 5:45:00	029XX N CAMIS	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - G	FALSE	FALSE	1412	14	35	21	14	1153479	1915488	2021	01/16/2021 03:	41.934.545.026	-8.771.157.369	(41.934545026, -87.71157369)
12381308	JE101888	01/03/2021 3:20:00	111XX S ASHLA	820	THEFT	$500 AND UNDE	STREET	FALSE	FALSE	2234	22	34	75	6			2021	01/16/2021 15:5			
12380436	JE102224	01/03/2021 14:30:00	048XX W HARR	560	ASSAULT	SIMPLE	BARBERSHOP	TRUE	TRUE	1033	15	28	25	08A	1144333	1896901	2021	01/16/2021 03:	41.873.110.002	-87.745.664.971	(41.873110002, -87.745664971)
12382258	JE104225	01/03/2021 18:00:00	043XX W 14TH	820	THEFT	$500 AND UNDE	APARTMENT	FALSE	FALSE	1011	10	24	29	6			2021	01/16/2021 15:5			
12380166	JE101739	01/02/2021 11:00:00	101XX S WASH	1154	DECEPTIVE PR	FINANCIAL IDE	RESIDENCE	FALSE	FALSE	2211	22	19	72	11	1183154	1837245	2021	01/16/2021 03:	41.709.109.358	-87.688.109.214	(41.709109358, -87.688109214)
12382863	JE101891	01/03/2021 10:11:00	077XX S STATE	460	BATTERY	SIMPLE	CAR WASH	FALSE	FALSE	623	6	6	69	08B	1177524	1853755	2021	01/16/2021 03:	41.754.107.211	-87.624.633.337	(41.754107211, -87.624633337)
12382755	JE101867	01/03/2021 4:15:00	052XX S CICE	320	ROBBERY	VEHICULAR HI	STREET	FALSE	FALSE	413	4	9	48	3	1181120	1844081	2021	01/16/2021 03:	41.726.983.742	-87.676.121.797	(41.726983742, -87.676121797)
12382188	JE101866	01/03/2021 4:21:00	012XX N DEARB	860	THEFT	RETAIL THEFT	CONVENIENCE	FALSE	FALSE	1824	18	2	8	6	1175711	1908356	2021	01/16/2021 03:	41.904.354.115	-87.630.081.432	(41.904354115, -87.630081432)
12382741	JE102573	01/03/2021 0:30:00	081XX N WINTH	810	BURGLARY	FORCIBLE ENT	RESIDENCE	FALSE	FALSE	2433	24	48	77	5	1167747	1942854	2021	01/16/2021 03:	41.993.388.727	-87.658.318.789	(41.993388727, -87.658318789)
12380679	JE102458	01/03/2021 19:15:00	087XX S LAFLI	486	BATTERY	DOMESTIC BAT	RESIDENCE	FALSE	TRUE	2222	22	21	71	08B	1167861	1845757	2021	01/16/2021 03:	41.735.039.060	-87.660.913.154	(41.735039060, -87.660913154)
12382412	JE100587	01/01/2021 13:50:00	131XX S CORLI	486	BATTERY	DOMESTIC BAT	RESIDENCE	FALSE	TRUE	533	5	9	54	08B	1184311	1818283	2021	01/16/2021 03:	41.656.568.928	-87.552.331.682	(41.656568928, -87.552331682)
12385677	JE101220	01/01/2021 16:00:00	038XX W VAN E	560	ASSAULT	SIMPLE	APARTMENT	FALSE	FALSE	1133	11	28	26	08A	1150799	1897748	2021	01/16/2021 03:	41.875.314.745	-87.721.782.665	(41.875314745, -87.721782665)
12383388	JE102063	01/03/2021 17:20:00	030XX W 48TH	560	ASSAULT	SIMPLE	STREET	FALSE	FALSE	935	9	11	61	08A	1172812	1872729	2021	01/16/2021 03:	41.806.281.677	-8.764.170.959	(41.806281677, -87.64170959)
12382166	JE104228	01/03/2021 2:00:00	011XX W 15TH	460	BATTERY	SIMPLE	APARTMENT	TRUE	FALSE	1232	12	11	28	08B	1169068	1892061	2021	01/16/2021 15:4	41.861.837.423	-8.766.485.686	(41.861837423, -87.66485686)
12381954	JE100845	01/03/2021 0:01:00	040XX W WASH	910	MOTOR VEHICL	AUTOMOBILE	HOTEL / MOTEL	FALSE	FALSE	1113	11	28	26	7	1149210	1900113	2021	02/23/2021 03:	41.881.893.071	-8.775.858.166	(41.881893071, -87.75858166)
25702	JE102715	01/03/2021 18:06:00	048XX W FILLT	110	HOMICIDE	FIRST DEGREE	STREET	FALSE	FALSE	1032	10	28	25	01A	1142803	1901462	2021	01/16/2021 15:4	41.885.720.775	-87.747.018.429	(41.885720775, -87.747018429)
25699	JE100162	01/01/2021 3:48:00	047XX S DR MF	110	HOMICIDE	FIRST DEGREE	AUTO	FALSE	FALSE	224	2	3	38	01A	1179656	1873765	2021	01/16/2021 15:4	41.808.899.077	-87.616.376.757	(41.808899077, -87.616376757)
12382566	JE102008	01/03/2021 9:20:00	049XX W WINTH	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - G	FALSE	FALSE	2024	20	45	3	14			2021	01/16/2021 15:4			
12385187	JE100841	01/01/2021 20:15:00	029XX S DAN R	486	BATTERY	DOMESTIC BAT	VEHICLE NON-I	FALSE	TRUE	914	9	11	34	08B	1177526	1885369	2021	02/02/2021 15:4	41.840.824.280	-87.631.092.609	(41.840824280, -87.631092609)
12382564	JE102356	01/03/2021 3:00:00	030XX N NOTTI	1320	CRIMINAL DAM	TO VEHICLE	RESIDENCE	FALSE	FALSE	2511	25	29	18	14			2021	01/16/2021 15:5			
12385400	JE101876	01/02/2021 6:50:00	005XX E 40TH (	820	THEFT	$500 AND UNDE	APARTMENT	FALSE	FALSE	221	2	3	38	6	1180834	1874841	2021	01/16/2021 03:	41.811.267.452	-87.612.229.179	(41.811267452, -87.612229179)
12385691	JE101583	01/02/2021 15:30:00	052XX E 12ST	560	ASSAULT	SIMPLE	APARTMENT	FALSE	FALSE	532	5	9	53	08A	1180134	1824770	2021	01/16/2021 03:	4.167.443.174	-87.616.303.857	(41.67443174, -87.616303857)
12385621	JE101613	01/02/2021 19:00:00	005XX N STATE	486	BATTERY	DOMESTIC BAT	RESIDENCE	TRUE	TRUE	1651	18	42	8	08B	1175273	1904305	2021	01/16/2021 03:	41.891.546.167	-87.628.073.854	(41.891546167, -87.628073854)
12385624	JE101591	01/02/2021 12:45:00	107XX S AVEN	560	ASSAULT	SIMPLE	RESIDENCE - P	FALSE	FALSE	432	4	10	52	08A	1182846	1834426	2021	01/16/2021 03:	41.700.380.001	-87.592.860.013	(41.700380001, -87.592860013)
12385669	JE101528	01/02/2021 17:17:00	044XX W MONR	0410	BATTERY	AGGRAVATED	STREET	FALSE	FALSE	1113	11	28	26	04B	1149870	1895255	2021	01/16/2021 03:	41.879.655.693	-87.736.155.995	(41.879655693, -87.736155995)
12385668	JE101705	01/02/2021 21:47:00	035XX W 79TH	2027	NARCOTICS	POSSESS - CR	STREET	FALSE	FALSE	1112	11	37	23	18	1149769	1857346	2021	01/16/2021 03:	41.750.350.511	-87.708.350.913	(41.750350511, -87.708350913)
12382008	JE101740	01/02/2021 22:54:00	042XX W GRAN	2026	NARCOTICS	POSSESS - PO	STREET	TRUE	FALSE	2034	25	26	23	18	1148323	1909733	2021	01/16/2021 03:	41.908.256.711	-87.731.675.077	(41.908256711, -87.731675077)
12383075	JE104877	01/03/2021 9:00:00	015XX W Grand	820	THEFT	$500 AND UNDE	RESIDENCE	FALSE	FALSE	1215	12	1	24	6	1165970	1903075	2021	01/16/2021 03:	41.890.994.365	-87.665.924.348	(41.890994365, -87.665924348)
12383597	JE103947	01/02/2021 14:00:00	016XX S CRAWI	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - G	FALSE	FALSE	1021	10	24	29	14	1152974	1891599	2021	01/16/2021 03:	41.858.516.076	-87.719.970.552	(41.858516076, -87.719970552)

Gambar 4. 1 Dataset Chicago

Berikut script code untuk membaca dataset tersebut.

```
DATA_PATH = "E:/Poltekpos/TI/Tingkat 4/Intership/# Source Code Khusus Jurnal/Data Crime Internship/data/"

file_names = ['crimes_2019.csv','crimes_2020.csv','crimes_2021.csv']
file_names = [DATA_PATH+x for x in file_names]

main_df = us.create_df(file_names)
orig_shape = main_df.shape
print("Jumlah Kejahatan: "+ str(main_df.shape[0]))
print("\n Jumlah Kolom: "+ str(main_df.shape[1]))
```

Gambar 4. 2 Membaca Dataset

DATA_PATH = digunakan untuk mengetahui posisi tempat penyimpanan data csv yang digunakan.

file_names = variable yang digunakan untuk menampung data array.

main_df = variable yang digunakan untuk mengeksekusi parameter file_names.

orig_shape = variable yang digunakan untuk menampung fungsi shape, disini main_df.shape.

shape = digunakan untuk untuk mengetahui jumlah baris dan kolom dari DataFrame

print = fungsi untuk menampilkan hasil dari suatu parameter.

Berikut penjelasan tentang dataset Chicago sebagai berikut :

1. *ID* : Merupakan nomor id dari dataset
2. *Case Number* : Merupakan case number atau number perkara pada dataset
3. *Date* : Merupakan tanggal kejadian dari sebuah peristiwa pada dataset.
4. *Primary Type* : Merupakan daftar kejahatan yang tersedia pada dataset.
5. *Description* : Merupakan deskripsi kejahatan tersebut.
6. *Location Description* : Merupakan deskripsi lokasi kejadian.
7. *Arrest* : Merupakan kode penangkapan, apakah dilakukan penangkapan atau tidak, dengan kode *TRUE* atau *FALSE*
8. *Domestic* : Merupakan kode domestic
9. *Beat* : Merupakan kode jumlah kejahatan
10. *District* : Merupakan kode distrik atau tempat
11. *Ward* : merupakan kode bangsal atau tempat tinggal
12. *Community Area* : Merupakan kode area dari suatu wilayah
13. *FBI Code* : Merupakan kode kejahatan yang kode tersebut dibawah fbi
14. *X Coordinate* : Merupakan titik koordinat X
15. *Y Coordinate* : Merupakan titik koordinat Y
16. *Year* : Merupakan tahun kejadian
17. *Update On* : Merupakan Update tanggal kejadian.
18. *Latitude* : Merupakan kode Lintang.
19. *Longtitude* : Merupakan kode bujur.
20. *Location* : Merupakan kode lokasi.

Setelah didapatkan data, maka selanjutnya data tersebut di bersihkan dan direkayasa beberapa fitur, sehingga data yang tersedia terdapat *Month*, *Day*, *Hour*, *Distric*t, Jumlah Kejahatan, *Alarm* (Jenis Kejahatan).

	Month	Day	Hour	District	Jumlah_Kejahatan	Alarm
25709	7	6	10	31	1	0
36724	10	6	3	31	1	0
35145	10	3	23	31	1	0
28336	8	4	0	31	1	0
36199	10	5	2	31	1	0
36200	10	5	7	31	1	0
22568	7	0	12	31	1	0
13676	4	4	15	31	1	0
528	1	0	0	31	1	0
7360	2	6	12	31	1	0
3156	1	5	11	31	1	0
42461	12	3	11	31	1	0
40377	11	6	12	31	1	0
23090	7	1	9	31	1	0
21517	6	5	7	31	1	0

11568	4	0	16	25	10	0
27269	8	2	3	25	2	0
11569	4	0	17	25	18	1
11567	4	0	15	25	11	0

Berikut *script code* untuk melakukan proses kompres dataset.

```
def crime_rate_assign(x):
    if(x<=14):
        return 0
    elif(x>14 and x<=33):
        return 1
    else:
        return 2
cri6['Alarm'] = cri6['Jumlah_Kejahatan'].apply(crime_rate_assign)
cri6 = cri6[['Month','Day','Hour','District','Jumlah_Kejahatan','Alarm']]
cri6.head()
```

Gambar 4. 3 Kompres Dataset

def = merupaka sebuah fungsi dalam python diikuti nama fungsinya, disini nama fungsi nya yaitu crime_rate_assign.

If else = merupakan kondisi perulangan ya dan tidak.

Cri6 = variable yang digunakan untuk menampung nilai array.

Cri6.head = merupakan variable yang mempunyai fungsi head yang digunakan untuk menampilkan *value* dari variable cri6.

Selanjutnya untuk melakukan proses klasifikasi, penulis membuat pelabelan tingkat kejahatan menjadi 3 level.

0 = tingkat kejahatan rendah.

1 = tingkat kejahatan sedang.

2 = tingkat kejahatan tinggi.

Untuk penentuan tingkat kejahatan, diukur dari seberapa banyak jumlah kejahatan yang terdapat pada waktu tersebut. Jumlah kejahatan dibagi menjadi 3 bagian.

Kejahatan rendah memiliki jumlah kejahatan sebanyak 0 – 14 kejadian.

Kejahatan sedang memiliki jumlah kejahatan sebanyak 15 – 33 kejadian.

Kejahatan tinggi memiliki jumlah kejahatan sebanyak 34 kejadian keatas.

Untuk kategori kejahatan, penulis menggunakan jenis kejahatan pencurian untuk penelitian ini, karena kejahatan tersebut adalah yang tertinggi dari kejahatan yang lain.

Untuk memprediksi tingkat kejahatan antara jumlah kejahatan dan jenis kejahatan, maka disini menggunakan percabangan if else. [2]

```
if(x<=14):
    return 0
elif(x>14 and x<=33):
    return 1
else:
        return 2
    end if
```

Berikut script code untuk melakukan kondisi perulangan.

```
def crime_rate_assign(x):
    if(x<=14):
        return 0
    elif(x>14 and x<=33):
        return 1
    else:
        return 2
```

Gambar 4. 4 Kondsi IF Else

Untuk menetukan accuracy dari model yang digunakan, disini menggunakan beberapa rumus.

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

$$\textit{Precision} = \frac{TP}{TP + FP}$$

$$\textit{Recall} = \frac{TP}{TP + FN}$$

$$\textit{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\textit{UAR} = \frac{R1 + R2 + R3}{3}$$

Beberapa penjelasan diatas diantaranya :

TP = True Positif (Mengacu pada jumlah sampel positif yang telah diidentifikasi dengan benar sebagai positif). [2]

FP = False Positif (Mengacu pada jumlah sampel negatif yang salah diidentifikasi sebagai positif) [2]

TN = True Negatif (Mengacu pada jumlah sampel negatif yang telah diidentifikasi dengan benar sebagai negatif). [2]

FN = False Negatif (Mengacu pada jumlah sampel positif yang salah diidentifikasi sebagai negatif). [2]

Accuracy = Akurasi merupakan tingkat ketepatan atau seberapa dekat suatu pengukuran dengan nilai sebenarnya. [2]

Presisi = Mendeskripsikan kemungkinan suatu sampel berada di kelas X jika sudah diprediksi masuk ke dalam kelas X. [2]

Recall = Menjelaskan kemungkinan model untuk mengklasifikasikan sampel dengan benar sebagai milik kelas X jika itu benar benar milik kelas X. [2]

F1 = Ukuran yang menggambarkan hubungan antara presisi dan mengingat. Semakin tinggi skor F1 untuk suatu kelas, semakin baik kinerja model dalam mengklasifikasi kelas tersebut. [2]

4.2 Hasil dan Pembahasan

4.2.1 Mengimport Library

Pada tahap ini, persiapkan library yang akan digunakan.

```
import pandas as pd
import sys
sys.path.append("utils")
import matplotlib.pyplot as plt
import numpy as np
from datetime import datetime
import seaborn as sns

import util_script as us
```

Gambar 4. 5 Import Library

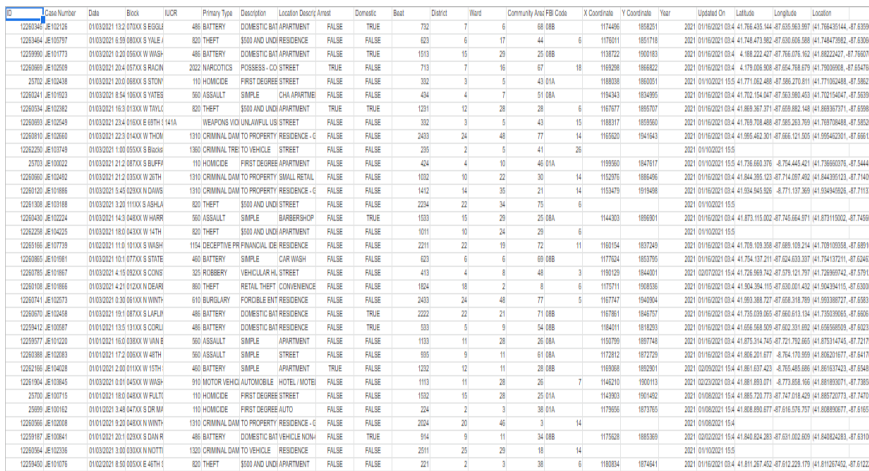
Pada code util_script tersebut, buat sebuah fungsi dengan menggunakan bahasa python untuk melakukan proses load data yang akan digunakan.

```
import pandas as pd

def create_df(filenames):

    main_df = pd.read_csv(filenames[0])
    print("Proses Loading Dataset "+filenames[0][-8:4]+".")
 main_df = main_df[list(main_df.columns[:22])]
    for file in filenames[1:]:
        print("Proses Loading Dataset "+file[-8:-4]+".")
        df_temp = pd.read_csv(file)
        df_temp = df_temp[list(df_temp.columns[:22])]
        main_df
main_df.append(df_temp,ignore_index=True)
print("Prses Loading Dataframe Selesai.\n\n")
```

```
return main_df
```

4.2.2 Melakukan Proses Obtain Data

Untuk melakukan proses *obtain data* ini dimana proses ini untuk mengumpulkan data, penulis menggunakan dataset criminal yang berada di kota Chicago, dataset tersebut berbentuk csv. Disini penulis menggunakan dataset rentang waktu tahun 2019, 2020 dan 2021 dimana pada dataset tersebut disediakan oleh data.cityofchicago.org yang membahas tentang criminal. Berikut contoh data yang tersedia dalam bentuk csv.



ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Descrip	Arrest	Domestic	Beat	District	Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location
12263940	JE102126	01/03/2021 13:0	070XX S EGGLE	486	BATTERY	DOMESTIC BAT	APARTMENT	FALSE	TRUE	732	7	6	68	08B	1174496	1858251	2021	01/16/2021 03:4	41.766.435.144	-87.635.963.997	(41.766431944, -87.635963997)
12263464	JE105797	01/03/2021 6:59	083XX S VALE /	820	THEFT	$500 AND UNDI	RESIDENCE	FALSE	FALSE	623	6	17	44	6	1176011	1851718	2021	01/16/2021 03:4	41.748.473.982	-87.630.606.588	(41.748473982, -87.630606588)
12259990	JE101773	01/03/2021 0:20	056XX W WASH	486	BATTERY	DOMESTIC BAT	APARTMENT	FALSE	TRUE	1513	15	29	25	08B	1138722	1900183	2021	01/16/2021 03:4	4.188.222.427	-87.766.076.162	(41.88222427, -87.766076162)
12265969	JE102569	01/03/2021 20:4	057XX S RACIN	2022	NARCOTICS	POSSESS - CO	STREET	TRUE	FALSE	713	7	16	67	18	1169298	1866822	2021	01/16/2021 03:4	4.179.006.908	-87.654.768.679	(41.79009008, -87.654768679)
25702	JE102438	01/03/2021 20:0	068XX S STONY	110	HOMICIDE	FIRST DEGREE	STREET	FALSE	FALSE	332	3	5	43	01A	1188038	1860051	2021	01/10/2021 15:5	41.771.082.488	-87.586.270.811	(41.771082488, -87.586270811)
12265241	JE101923	01/03/2021 8:54	100XX S YATES	560	ASSAULT	SIMPLE	CHA APARTME	FALSE	FALSE	434	4	7	51	08A	1194343	1834995	2021	01/16/2021 03:4	41.702.154.047	-87.563.980.453	(41.70215047, -87.563980453)
12265034	JE102382	01/03/2021 16:5	013XX W TWLC	820	THEFT	$500 AND UNDI	APARTMENT	TRUE	TRUE	1231	12	28	28	6	1167677	1895707	2021	01/16/2021 03:4	41.869.367.371	-87.659.882.148	(41.869367371, -87.659882148)
12263955	JE102549	01/03/2021 23:4	016XX E 69TH (	141A	WEAPONS VIO	UNLAWFUL USI	STREET	FALSE	FALSE	332	3	5	43	15	1188317	1859560	2021	01/16/2021 03:4	41.768.708.488	-87.585.283.769	(41.76870848, -87.585283769)
12263910	JE102660	01/03/2021 22:3	014XX W THOM	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - C	FALSE	FALSE	2433	24	48	77	14	1165520	1941643	2021	01/16/2021 03:4	41.885.462.301	-87.666.121.505	(41.88546301, -87.666121505)
12262250	JE103749	01/03/2021 1:00	055XX S Blackst	1360	CRIMINAL TRES	TO VEHICLE	STREET	FALSE	FALSE	235	2	5	41	26			2021	01/10/2021 15:5			
25703	JE100022	01/03/2021 21:2	087XX S BUFFA	110	HOMICIDE	FIRST DEGREE	APARTMENT	FALSE	FALSE	424	4	10	46	01A	1189560	1847917	2021	01/10/2021 15:5	41.736.660.376	-87.5444.5421	(41.736660376, -87.54445421)
12265960	JE102492	01/03/2021 21:2	035XX W 26TH	1310	CRIMINAL DAM	TO PROPERTY	SMALL RETAIL	FALSE	FALSE	1032	10	22	30	14	1152976	1886496	2021	01/16/2021 03:4	41.844.395.123	-87.714097452	(41.844395123, -87.714097452)
12265126	JE101886	01/03/2021 5:45	029XX N DAWS	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - C	FALSE	FALSE	1412	14	35	21	14	1153479	1919498	2021	01/16/2021 03:4	41.934.945.926	-87.7113769	(41.934945926, -87.7113769)
12261308	JE103188	01/03/2021 3:20	111XX S ASHLA	820	THEFT	$500 AND UNDI	STREET	FALSE	FALSE	2234	22	34	75	6			2021	01/10/2021 15:5			
12265430	JE102224	01/03/2021 14:3	048XX W HARR	560	ASSAULT	SIMPLE	BARBERSHOP	FALSE	TRUE	1533	15	29	25	08A	1144303	1896901	2021	01/16/2021 03:4	41.873.115.002	-87.745664871	(41.873115002, -87.745664871)
12262258	JE104225	01/03/2021 18:0	043XX W 14TH	820	THEFT	$500 AND UNDI	APARTMENT	FALSE	FALSE	1011	10	24	29	6			2021	01/10/2021 15:5			
12265166	JE107739	01/02/2021 11:0	101XX S WASH	1154	DECEPTIVE PR	FINANCIAL IDE	RESIDENCE	FALSE	FALSE	2211	22	19	72	11	1160154	1837249	2021	01/16/2021 03:4	41.709.109.358	-87.689.109.214	(41.709109358, -87.689109214)
12265965	JE101981	01/03/2021 10:1	077XX S STATE	460	BATTERY	SIMPLE	CAR WASH	FALSE	FALSE	623	6	6	69	08B	1177024	1853795	2021	01/16/2021 03:4	41.754.137.211	-87.624.633.337	(41.754137211, -87.624633337)
12265785	JE101867	01/03/2021 4:15	052XX S CONS	325	ROBBERY	VEHICULAR HI	STREET	FALSE	FALSE	413	4	8	48	3	1190129	1844001	2021	02/07/2021 15:4	41.726.569.742	-87.579.121.797	(41.726569742, -87.579121797)
12265106	JE101866	01/03/2021 4:21	012XX N DEARB	860	THEFT	RETAIL THEFT	CONVENIENCE	FALSE	FALSE	1824	18	2	8	6	1175711	1908336	2021	01/16/2021 03:4	41.904.394.115	-87.630.081.432	(41.904394115, -87.630081432)
12265741	JE102573	01/03/2021 9:30	061XX N WINTH	610	BURGLARY	FORCIBLE ENT	RESIDENCE	FALSE	FALSE	2433	24	48	77	5	1167747	1940804	2021	01/16/2021 03:4	41.993.388.727	-87.668.318.789	(41.993388727, -87.668318789)
12265976	JE102458	01/03/2021 19:1	087XX S LAFLI	486	BATTERY	DOMESTIC BAT	RESIDENCE	FALSE	TRUE	2222	22	21	71	08B	1167861	1846757	2021	01/16/2021 03:4	41.735.039.065	-87.660.613.134	(41.735039065, -87.660613134)
12265412	JE100587	01/01/2021 13:5	131XX S CORLI	486	BATTERY	DOMESTIC BAT	RESIDENCE	FALSE	TRUE	533	5	9	54	08B	1184011	1818293	2021	01/16/2021 03:4	41.656.568.509	-87.602.331.652	(41.656568509, -87.602331652)
12265977	JE101220	01/01/2021 16:0	038XX W VAN B	560	ASSAULT	SIMPLE	APARTMENT	FALSE	FALSE	1133	11	28	26	08A	1150799	1897748	2021	01/16/2021 03:4	41.875.314.745	-87.721.792.665	(41.875314745, -87.721792665)
12265388	JE102083	01/01/2021 17:2	035XX W 48TH	560	ASSAULT	SIMPLE	STREET	FALSE	FALSE	935	9	11	61	08A	1172812	1872729	2021	01/16/2021 03:4	41.808.201.677	-87.714.709.69	(41.808201677, -87.71470969)
12262166	JE104028	01/01/2021 2:00	011XX W 115TH	460	BATTERY	SIMPLE	APARTMENT	TRUE	FALSE	1232	12	11	28	08B	1169068	1892901	2021	02/09/2021 15:4	41.681.637.423	-87.654.858.686	(41.681637423, -87.654858686)
12261904	JE103845	01/03/2021 0:01	045XX W WASH	910	MOTOR VEHICL	AUTOMOBILE	HOTEL / MOTEI	FALSE	FALSE	1113	11	28	26	7	1148210	1900113	2021	02/23/2021 03:4	41.881.893.071	-8.773.858.166	(41.881893071, -87.73858166)
25705	JE100715	01/01/2021 18:0	048XX W FULTO	110	HOMICIDE	FIRST DEGREE	STREET	FALSE	FALSE	1532	15	28	25	01A	1143903	1901482	2021	01/08/2021 15:4	41.885.720.773	-87.747.018.429	(41.885720773, -87.747018429)
25695	JE100162	01/01/2021 3:48	047XX S DR MA	110	HOMICIDE	FIRST DEGREE	AUTO	FALSE	FALSE	224	2	3	38	01A	1179906	1873765	2021	01/08/2021 15:4	41.808.890.877	-87.616.576.757	(41.808890877, -87.616576757)
12265950	JE102008	01/01/2021 9:20	046XX N WINTH	1310	CRIMINAL DAM	TO PROPERTY	RESIDENCE - C	FALSE	FALSE	2024	20	46	3	14			2021	01/08/2021 15:4			
12259167	JE100841	01/01/2021 21:5	029XX S DAN R	486	BATTERY	DOMESTIC BAT	VEHICLE NON-	FALSE	TRUE	914	9	11	34	08B	1179528	1885369	2021	02/02/2021 15:4	41.840.824.283	-87.631.002.609	(41.840824283, -87.631002609)
12260564	JE102336	01/03/2021 3:00	030XX N NOTTI	1320	CRIMINAL DAM	TO VEHICLE	RESIDENCE	FALSE	FALSE	2511	25	29	16	14			2021	01/10/2021 15:5			
12263450	JE101078	01/02/2021 8:50	005XX E 49TH (	820	THEFT	$500 AND UNDI	APARTMENT	FALSE	FALSE	221	2	3	38	6	1180934	1874641	2021	01/16/2021 03:4	41.811.267.452	-87.612.235.179	(41.811267452, -87.612235179)

Gambar 4. 6 Dataset Chicago

Pada tahap ini, data yang disediakan akan dikumpulkan dan dilakukan untuk membaca data yang digunakan.

```
DATA_PATH = "E:/Poltekpos/TI/Tingkat 4/Intership/# Source Code Khusus Jurnal/Data Crime Internship/data/"

file_names = ['crimes_2019.csv','crimes_2020.csv','crimes_2021.csv']
file_names = [DATA_PATH+x for x in file_names]

main_df = us.create_df(file_names)
orig_shape = main_df.shape
print("Jumlah Kejahatan: "+ str(main_df.shape[0]))
print("\n Jumlah Kolom: "+ str(main_df.shape[1]))
```

Gambar 4. 7 Membaca Dataset

DATA_PATH = digunakan untuk mengetahui posisi tempat penyimpanan data csv yang digunakan.

file_names = variable yang digunakan untuk menampung data array.

main_df = variable yang digunakan untuk mengeksekusi parameter file_names.

orig_shape = variable yang digunakan untuk menampung fungsi shape, disini main_df.shape.

print = fungsi untuk menampilkan hasil dari suatu parameter.

Berikut tampilan dari perintah tersebut seperti ini.

```
Proses Loading Dataset 2019.
Proses Loading Dataset 2020.
Proses Loading Dataset 2021.
Prses Loading Dataframe Selesai.


Jumlah Kejahatan: 492007

 Jumlah Kolom: 22
```

Gambar 4. 8 Hasil Pembacaan Dataset

Pada tampilan gambar tersebut menjelaskan bahwa :

1. Tiga **Proses Loading** diatas menjelaskan bahwa proses dataset sedang di load untuk ditampilkan.
2. **Proses Loading Dataframe Selesai** menjelaskan bahwa proses untuk meload dataset telah selesai.

```
Proses Loading Dataset 2019.
Proses Loading Dataset 2020.
Proses Loading Dataset 2021.
Prses Loading Dataframe Selesai.


Jumlah Kejahatan: 492007

 Jumlah Kolom: 22
```

Gambar 4. 9 Hasil Pembacaan Dataset

3. **Jumlah Kejahatan** menjelaskan tentang total kasus kejahatan yang terjadi dari tahun 2019 – 2021.
4. **Jumlah Kolom** menjelaskan total kolom yang tersedia pada dataset tersebut.
5. Untuk melihat info tipe data yang digunakan pada dataset tersebut menggunakan fungsi main_df.info().

```
main_df.info()
```

Gambar 4. 10 Melihat Informasi Dataframe

Berikut tampilan dari fungsi tersebut.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 492007 entries, 0 to 492006
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   ID                    492007 non-null  int64
 1   Case Number           492007 non-null  object
 2   Date                  492007 non-null  object
 3   Block                 492007 non-null  object
 4   IUCR                  492007 non-null  object
 5   Primary Type          492007 non-null  object
 6   Description           492007 non-null  object
 7   Location Description  489790 non-null  object
 8   Arrest                492007 non-null  bool
 9   Domestic              492007 non-null  bool
 10  Beat                  492007 non-null  int64
 11  District              492007 non-null  int64
 12  Ward                  491985 non-null  float64
 13  Community Area        492006 non-null  float64
 14  FBI Code              492007 non-null  object
 15  X Coordinate          486901 non-null  float64
 16  Y Coordinate          486901 non-null  float64
 17  Year                  492007 non-null  int64
 18  Updated On            492007 non-null  object
 19  Latitude              486901 non-null  float64
 20  Longitude             486901 non-null  float64
 21  Location              486901 non-null  object
dtypes: bool(2), float64(6), int64(4), object(10)
memory usage: 76.0+ MB
```

Gambar 4. 11 Hasil Informasi Dataframe

Terdapat beberapa tipe data yang digunakan pada dataset tesebut diantaranya integer, object, boolean, dan float.

6. Untuk menghapus data NaN atau kosong, null data tidak ada, menggunakan fungsi dropna(), maka akan menghapus data yang kosong.

```
main_df = main_df.dropna()
main_df.isna().sum()
```

Gambar 4. 12 Menghapus Data Kosong

Maka tampilannya seperti ini :

```
ID                      0
Case Number             0
Date                    0
Block                   0
IUCR                    0
Primary Type            0
Description             0
Location Description    0
Arrest                  0
Domestic                0
Beat                    0
District                0
Ward                    0
Community Area          0
FBI Code                0
X Coordinate            0
Y Coordinate            0
Year                    0
Updated On              0
Latitude                0
Longitude               0
Location                0
dtype: int64
```

Gambar 4. 13 Tampilan Data Kosong

Hasil yang diperoleh 0 karena kekosongan data tidak terlalu banyak.

7. Untuk melihat data yang telah dibersihkan

```
print("Data Setelah Di Cleaning / Dibersihkan:",round(((main_df.shape[0]/orig_shape[0]) * 100),2),"%")
```

Gambar 4. 14 Data Setelah Dibersihkan

Untuk melihatnya hasilnya menggunakan perintah print, kemudian gunakan fungsi round, fungsi round ini digunakan untuk mengembalikan nilai dalam bentuk decimal. Kemudian lakukan pembagian pada *variable*

((main_df.shape[0]/orig_shape[0]) * 100),2)"%", berfungsi untuk membangi nilai array, seperti nilai dari data dari dataset, dimana data yang tersedia, dibagi dengan data yang dibutuhkan pada saat proses pembersihan data tersebut. Angka 2 disini berfungsi untuk menampilkan berapa banyak angka dibelakang koma, disini ditampilkan yaitu 2 angka, symbol persen disini sebagai string untuk menambahkan symbol di akhir pada saat program telah selesai di jalankan.
maka akan mendapatkan hasil 98, 57%.

```
Data Setelah Di Cleaning / Dibersihkan: 98.57 %
```

8. Untuk melihat data pada dataset menggunakan fungsi head(), secara default fungsi ini akan menampilkan 5 data pada dataset, bilangannya dimulai dari 0 – 4.

```
main_df.head()
```

Gambar 4. 15 Melihat Informasi Data

Berikut tampilannya seperti ini :

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	FBI Code
0	11675533	JC248984	05/03/2019 11:40:00 PM	027XX W NORTH AVE	1812	NARCOTICS	POSS: CANNABIS MORE THAN 30GMS	STREET	True	False	...	1.0	24.0	18
1	11675453	JC248966	05/03/2019 11:33:00 PM	051XX W MADISON ST	1330	CRIMINAL TRESPASS	TO LAND	PARKING LOT/GARAGE(NON.RESID.)	True	False	...	28.0	25.0	26
2	11675494	JC248999	05/03/2019 11:30:00 PM	011XX W LAWRENCE AVE	0810	THEFT	OVER $500	MOVIE HOUSE/THEATER	False	False	...	46.0	3.0	06
3	11675481	JC248956	05/03/2019 11:30:00 PM	121XX S HARVARD AVE	0810	THEFT	OVER $500	RESIDENTIAL YARD (FRONT/BACK)	False	False	...	34.0	53.0	06
4	11675910	JC248965	05/03/2019 11:25:00 PM	025XX N MILWAUKEE AVE	5011	OTHER OFFENSE	LICENSE VIOLATION	BAR OR TAVERN	False	False	...	32.0	22.0	26

5 rows × 22 columns

Gambar 4. 16 Tampilan Informasi Data

X Coordinate	Y Coordinate	Year	Updated On	Latitude	Longitude	Location
1157682.0	1910520.0	2019	05/10/2019 04:20:42 PM	41.910225	-87.696173	(41.910224908, -87.696172663)
1142285.0	1899542.0	2019	05/10/2019 04:20:42 PM	41.880400	-87.753009	(41.880399914, -87.753008553)
1167886.0	1932034.0	2019	05/10/2019 04:20:42 PM	41.969046	-87.658065	(41.969046206, -87.658064717)
1176145.0	1824335.0	2019	05/10/2019 04:20:42 PM	41.673328	-87.630934	(41.67332819, -87.630934077)
1155343.0	1916744.0	2019	05/10/2019 04:20:42 PM	41.927351	-87.704598	(41.92735143, -87.704597631)

Gambar 4. 17 Tampilan Informasi Data

4.4.2 Scrub Data

Tahapan selanjutnya yaitu melakukan scrub data (pembersihan data) pada dataset, tahapan tersebut diantanya :

1. Untuk melihat nama kolom pada tabel tersebut menggunakan fungsi **columns.**

```
print(main_df.columns)
```

Gambar 4. 18 Melihat Informasi Kolom

Berikut tampilan informasi kolom tersebut.

```
Index(['ID', 'Case Number', 'Date', 'Block', 'IUCR', 'Primary Type',
       'Description', 'Location Description', 'Arrest', 'Domestic', 'Beat',
       'District', 'Ward', 'Community Area', 'FBI Code', 'X Coordinate',
       'Y Coordinate', 'Year', 'Updated On', 'Latitude', 'Longitude',
       'Location'],
      dtype='object')
```

Gambar 4. 19 Tampilan Informasi Kolom

2. *Getting rid of missing value* (menghilangkan nilai yang kosong)
   Pada tahap ini akan memeriksa data dan membersihkan data dari data yang tidak diperlukan dengan menggunakan Bahasa pemrograman Python. Data yang dibersihkan terletak pada kolom Date.
   Berikut fungsi yang digunakan untuk membersihkan data pada kolom *Date* tersebut.

```
def time_convert(date_time):
    s1 = date_time[:11]
    s2 = date_time[11:]

    month = s1[:2]
    date = s1[3:5]
    year = s1[6:10]

    hr = s2[:2]
    mins = s2[3:5]
    sec = s2[6:8]
    time_frame = s2[9:]
    if(time_frame == 'PM'):
        if (int(hr) != 12):
            hr = str(int(hr) + 12)
    else:
        if(int(hr) == 12):
            hr = '00'

    final_date = datetime(int(year), int(month), int(date), int(hr), int(mins), int(sec))
    return final_date
```

Gambar 4. 20 Konversi Bulan, Hari, Jam

Kemudian pada kolom **Date** tersebut dilakukan convert dengan menggunakan fungsi **apply(time_convert)**

```
main_df['Date'] = main_df['Date'].apply(time_convert)
```

Gambar 4. 21 Fungsi Convert

Untuk menampilkan tanggal hasil *convert* tersebut bisa menggunakan perintah ***head***, maka akan ditampilkan 5 data teratas.

```
main_df['Date'].head()
```

Gambar 4. 22 Tampilan Informasi Data

Berikut hasil dari *convert* tersebut.

```
0   2019-05-03 23:40:00
1   2019-05-03 23:33:00
2   2019-05-03 23:30:00
3   2019-05-03 23:30:00
4   2019-05-03 23:25:00
Name: Date, dtype: datetime64[ns]
```

Gambar 4. 23 Hasil Pembersihan Tabel Date

3. *Feature Engineering* (Rekayasa Fitur)
   Pada tahap ini akan merekayasa beberapa fitur yang sesuai dengan bahasa yang digunakan, disini akan melakukan convert field ***Date*** **(Tanggal)** pada dataset yang dimana yang direkayasa agar dipisahka tiap datanya yaitu ***Month*** **(Bulan)**, ***Day*** **(Hari)** dan ***Hour*** **(Jam)** menggunakan library pandas dan datetime dari python.

   Proses *feature engineering* pada pyhon dilakukan seperti ini:

```
# Feature Engineering 1 : Month
def month_col(x):
    return int(x.strftime("%m"))
main_df['Month'] = main_df['Date'].apply(month_col)

# Feature Engineering 2 : Day
def day_col(x):
    return int(x.strftime("%w"))
main_df['Day'] = main_df['Date'].apply(day_col)

# Feature Engineering 3 : Hour
def hour_col(x):
    return int(x.strftime("%H"))
main_df['Hour'] = main_df['Date'].apply(hour_col)
```

Gambar 4. 24 Rekayasa Fitur Date

Pada tahap ini terdapat penambahan kolom sebanyak 3 kolom untuk menampung nilai dari Bulan (*Month*), Hari (*Day*) dan Jam (*Hour*).

Untuk melihat hasilnya gunakan fungsi **head.**

```
main_df.head()
```

Gambar 4. 25 Menampilkan Informasi Dataframe

Pada hasilnya terdapat penambahan kolom, dimana yang pertama terdapat 22 kolom, dan untuk sekarang bertambah menjadi 25 kolom seperti ini.

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	X Coordinate	Y Coordinate
0	11675533	JC248984	2019-05-03 23:40:00	027XX W NORTH AVE	1812	NARCOTICS	POSS: CANNABIS MORE THAN 30GMS	STREET	True	False	...	1157682.0	1910520.0
1	11675453	JC248966	2019-05-03 23:33:00	051XX W MADISON ST	1330	CRIMINAL TRESPASS	TO LAND	PARKING LOT/GARAGE(NON.RESID.)	True	False	...	1142285.0	1899542.0
2	11675494	JC248999	2019-05-03 23:30:00	011XX W LAWRENCE AVE	0810	THEFT	OVER $500	MOVIE HOUSE/THEATER	False	False	...	1167886.0	1932034.0
3	11675481	JC248956	2019-05-03 23:30:00	121XX S HARVARD AVE	0810	THEFT	OVER $500	RESIDENTIAL YARD (FRONT/BACK)	False	False	...	1176145.0	1824335.0
4	11675910	JC248965	2019-05-03 23:25:00	025XX N MILWAUKEE AVE	5011	OTHER OFFENSE	LICENSE VIOLATION	BAR OR TAVERN	False	False	...	1155343.0	1916744.0

5 rows × 25 columns

Gambar 4. 26 Tampilan Informasi Dataframe

Disini terlihat ada 3 kolom yang bertambah yaitu kolom *Month, Day, Hour*.

Year	Updated On	Latitude	Longitude	Location	Month	Day	Hour
2019	05/10/2019 04:20:42 PM	41.910225	-87.696173	(41.910224908, -87.696172663)	5	5	23
2019	05/10/2019 04:20:42 PM	41.880400	-87.753009	(41.880399914, -87.753008553)	5	5	23
2019	05/10/2019 04:20:42 PM	41.969046	-87.658065	(41.969046206, -87.658064717)	5	5	23
2019	05/10/2019 04:20:42 PM	41.673328	-87.630934	(41.67332819, -87.630934077)	5	5	23
2019	05/10/2019 04:20:42 PM	41.927351	-87.704598	(41.92735143, -87.704597631)	5	5	23

Gambar 4. 27 Tampilan Informasi Dataframe

Selanjutnya memilih salah satu kejahatan yang digunakan untuk penelitian ini, peneliti memilih kejahatan pencurian sebagai tingkat kejahatan yang akan dianalisis, kemudian memfilter jenis kejahatan hanya tindak pidana pencurian.

```
top_10 = list(main_df['Primary Type'].value_counts().head(10).index)

def filter_top_10(df):
    df2=df[df['Primary Type']=='THEFT']
    for crime in top_10[1:]:
        temp=df[df['Primary Type']==crime]
        df2 = df2.append(temp, ignore_index=True)
    return df2

df2=filter_top_10(main_df) # the dataframe with all the data of only the top 10 crimes
df2.shape
```

Gambar 4. 28 Klasifikasi Kejahatan Pencurian

Tampilan hasil kejahatan dengan kejahatan pencurian (***THEFT***) sebagai berikut.

```
(330221, 22)
```

Gambar 4. 29 Hasil Kejahatan Pencurian

Untuk menampilkan dataset baru untuk pencurian, gunakan fungsi head.

```
df2.head()
```

Gambar 4. 30 Melihat Informasi Dataframe

Hasil dari dataset untuk pencurian (***THEFT***) seperti ini.

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	FBI Code	X Coordinate
0	11675494	JC248999	2019-05-03 23:30:00	011XX W LAWRENCE AVE	0810	THEFT	OVER $500	MOVIE HOUSE/THEATER	False	False	...	46.0	3.0	06	1167886.0
1	11675481	JC248956	2019-05-03 23:30:00	121XX S HARVARD AVE	0810	THEFT	OVER $500	RESIDENTIAL YARD (FRONT/BACK)	False	False	...	34.0	53.0	06	1176145.0
2	11675513	JC248946	2019-05-03 23:11:00	066XX W GRAND AVE	0810	THEFT	OVER $500	RESTAURANT	True	False	...	29.0	18.0	06	1131423.0
3	11675615	JC249145	2019-05-03 23:00:00	079XX S ST LAWRENCE AVE	0820	THEFT	$500 AND UNDER	RESIDENCE-GARAGE	False	False	...	6.0	44.0	06	1181614.0
4	11676084	JC249692	2019-05-03 23:00:00	032XX N LAMON AVE	0820	THEFT	$500 AND UNDER	STREET	False	False	...	31.0	15.0	06	1143136.0

Gambar 4. 31 Tampilan Informasi Dataframe

Kemudian kelompokan beberapa kolom menjadi sebuah group. Ini berfungsi sebagai patokan dalam penelitian, dalam menentukan bulan, Hari dam Jam.

```
cri5 = df2.groupby(['Month','Day','District','Hour'], as_index=False).agg({"Primary Type":"count"})
cri5 = cri5.sort_values(by=['District'], ascending=False)
cri5.head()
```

Gambar 4. 32 Pengelompokan Data kolom

Berikut hasil tampilan dari pengelompokan yang telah dibuat.

	Month	Day	District	Hour	Primary Type
25709	7	6	31	10	1
36724	10	6	31	3	1
35145	10	3	31	23	1
28336	8	4	31	0	1
36199	10	5	31	2	1

Gambar 4. 33 Tampilan Pengelompokan Data

```
cri6=cri5.rename(index=str, columns={"Primary Type":"Jumlah_Kejahatan"})
cri6.head()
```

Gambar 4. 34 Ubah Nama Kolom

Berikut tampilan hasil dari perubahan nama kolom tersebut.

	Month	Day	District	Hour	Jumlah_Kejahatan
25709	7	6	31	10	1
36724	10	6	31	3	1
35145	10	3	31	23	1
28336	8	4	31	0	1
36199	10	5	31	2	1

Gambar 4. 35 Tampilan Perubahan Nama Kolom

4. *Compressing the Dataset* (Melakukan kompres pada dataset)
   Tahapan selanjutnya yaitu melakukan kompres dataset yang didapat menjadi dataset yang baru, dimana data yang sebelumnya tidak dibutuhkan, dihilangkan dan data ini yang akan digunakan

untuk melakukan proses prediksi tersebut. data yang di kompres merupakan data dalam bentuk csv.

Hasil dari compress data tersebut.

	Month	Day	Hour	District	Jumlah_Kejahatan	Alarm
25709	7	6	10	31	1	0
36724	10	6	3	31	1	0
35145	10	3	23	31	1	0
28336	8	4	0	31	1	0
36199	10	5	2	31	1	0
36200	10	5	7	31	1	0
22568	7	0	12	31	1	0
13676	4	4	15	31	1	0
528	1	0	0	31	1	0
7360	2	6	12	31	1	0
3156	1	5	11	31	1	0
42461	12	3	11	31	1	0
40377	11	6	12	31	1	0
23090	7	1	9	31	1	0
21517	6	5	7	31	1	0
11568	4	0	16	25	10	0
27269	8	2	3	25	2	0
11569	4	0	17	25	18	1
11567	4	0	15	25	11	0

Data hasil kompres yang ditampilkan hanya sebagian data. Data yang be rhasil dikompres sebanyak **44015 data** dan mempunyai **7 kolom.**

4.4.3 *Explore Data*

Pada tahap ini dilakukan *explore* data dimana pada bagian ini jumlah kejahatan dibagi berdasarkan bulan, hari, tempat dan waktu.

Untuk pertama mengetahui jumlah dataset yang telah hasil di kompres

```
cri6 = cri6[['Month','Day','District','Hour','Jumlah_Kejahatan']]
cri6.head()
print("Jumlah Dataset:", cri6.shape)
```

Gambar 4. 36 Explore Data

Untuk tampilan jumlah hasil nya seperti berikut

```
Jumlah Dataset: (44015, 5)
```

Gambar 4. 37 Jumlah Data

Untuk melihat rata rata jumlah kejahatan yang dilakukan.

```
print("Rata-rata kejahatan per distrik per titik waktu
      :",round(cri6['Jumlah_Kejahatan'].sum()/cri6.shape[0],
      2),".")
```

Untuk melihat jumlah hasil rata rata seperti berikut.

```
Rata-rata kejahatan per distrik per titik waktu : 7.59 .
```

Gambar 4. 38 Melihat Rata Rata Kejahatan Per Waktu

Selanjutnya pada tahap ini dilakukan percabangan if else, pada tahap analisis dijelaskan bahwa pada proses terdapat 3 label, yaitu 0,1 dan 2 yang artinya tingkat kejahatan rendah, sedang dan tinggi.

Untuk percabangannya seperti ini.

```
def crime_rate_assign(x):
    if(x<=14):
        return 0
    elif(x>14 and x<=33):
        return 1
    else:
        return 2
cri6['Alarm'] = cri6['Jumlah_Kejahatan'].apply(crime_rate_assign)
cri6 = cri6[['Month','Day','Hour','District','Jumlah_Kejahatan','Alarm']]
cri6.head()
```

Gambar 4. 39 Kondisi IF Else

Buat fungsi terlebih dahulu, disini terdapat fungsi crime_rate_assign dengan nilai x, selanjutnya didalam fungsi berikan beberapa kondisi, disini menggunakan kondisi if else.

Untuk hasil nya seperti ini.

	Month	Day	Hour	District	Jumlah_Kejahatan	Alarm
21329	6	5	7	31	1	0
28101	8	4	0	31	1	0
22895	7	1	9	31	1	0
7306	2	6	12	31	1	0
35861	10	5	2	31	1	0

Gambar 4. 40 Tampilan Hasil Explore Data

Terdapat penambahan kolom yaitu district untuk menjelaskan kode daerah tersebut.

Untuk melihat hasil kompres dataset tersebut dapat menggunakan perintah.

```
cri6.to_csv("E:/Poltekpos/TI/Tingkat 4/Intership/# Source Code Khusus Jurnal/Data Crime Internship/data/Hasil_Kompress.csv")
```

Terdapat variable yang digunakan untuk menampung data penyimpanan ke lokasi yang dituju. Untuk penyimpanan file tersebut bisa berbeda beda tergantung selera anda untuk menyimpan file tersebut.

Kemudian untuk mengecek seberapa banyak data yang digunakan untuk klasifikasi gunakan fungsi **value_counts(),** berfungsi untuk melihat nilai hitungannya.

```
cri6['Alarm'].value_counts()
```

Gambar 4. 41 Value Counts

Berikut hasil dari fungsi tersebut.

```
0    39933
1     3589
2       14
Name: Alarm, dtype: int64
```

Gambar 4. 42 Hasil Value Counts

Kemudian untuk melihat presentase tingkat kejahatan, baik rendah sedang dan tinggi.

```
print("Persentase Kejahatan Rendah:", round(cri6['Alarm'].value_counts()[0]/cri6['Alarm'].value_counts().sum()*100,2))
print("Persentase Kejahatan Sedang:", round(cri6['Alarm'].value_counts()[1]/cri6['Alarm'].value_counts().sum()*100,2))
print("Persentase Kejahatan Tinggi:", round(cri6['Alarm'].value_counts()[2]/cri6['Alarm'].value_counts().sum()*100,2))
```

Gambar 4. 43 Presentase Tingkat Kejahatan

Untuk hasilnya seperti berikut.

```
Persentase Kejahatan Rendah: 91.72
Persentase Kejahatan Sedang: 8.24
Persentase Kejahatan Tinggi: 0.03
```

Gambar 4. 44 Tampilan Presentase

Untuk melihat kejahatan berdasarkan tempat kejadian, dilakukan perintah seperti berikut :

Selanjutnya melakukan explore data berdasarkan nama lokasi seperti berikut:

```
def crime_rate_assign(x):
    if(x<=14):
        return 0
    elif(x>14 and x<=33):
        return 1
    else:
        return 2
cri6['Alarm'] = cri6['Jumlah_Kejahatan'].apply(crime_rate_assign)
cri6 = cri6[['Month','Day','Hour','Location Description','Jumlah_Kejahatan','Alarm']]
cri6.head(60)
```

Gambar 4. 45 Kondisi IF Else

Untuk melihat kejadian berdasarkan tahun

```
%matplotlib inline

main_df.resample('M').size().plot(legend=True)
plt.title('Jumlah Kejadian')
plt.xlabel('Data KeJadian Bulan ')
plt.ylabel('Bulan dan Tahun Kejadian')
plt.show()
```

Gambar 4. 46 Melihat Data Kejadin

Berikut penjelasannya ,

%matplotlib inline = berfungsi agar hasil visualisasi bisa langsung tercetak di Jupyter Notebook.

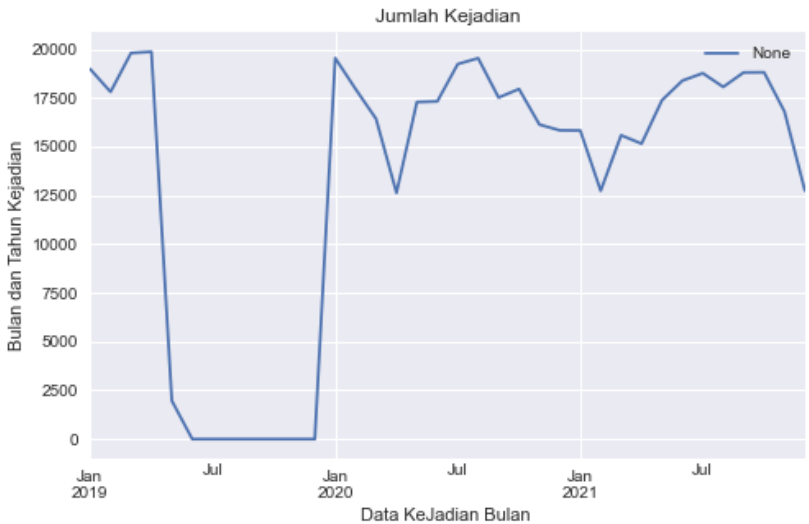
plt.title = digunakan untuk menentukan judul visualisasi yang digambarkan dan menampilkan judul menggunakan berbagai atribut.title()

plt.xlabel = digunakan untuk memberikan nama label untuk garis x.

plt.ylabe = digunakan untuk memberikan nama label untuk garis y.

plt.show = digunakan untuk menampilkan plot.

Beriku tampilan kejadian berdasarkan tahun.



Gambar 4. 47 Data Kejadian Berdasarkan Tahun

Pada gambar diatas, yang paling banyak kejadian kejahatan adalah tahun 2019 dari januari sampai juli dengan kejadian 20000 kejadian.

Selanjutnya untuk melihat kejadian berdasarkan bulan

```
%matplotlib inline

main_df['Date'].apply(lambda x: x.month).value_counts(sort=False).plot(kind='barh')
plt.xlabel("Jumlah Kejadian Per Bulan")
plt.ylabel("Bulan Kejadian")
plt.title("Data Kejadian Per Bulan")

plt.savefig("E:/Poltekpos/TI/Tingkat 4/jumlahdataperbulan.png")
plt.show()
```

Gambar 4. 48 Berdasarkan Bulan

Berikut penjelasannya ,

%matplotlib inline = berfungsi agar hasil visualisasi bisa langsung tercetak di Jupyter Notebook.

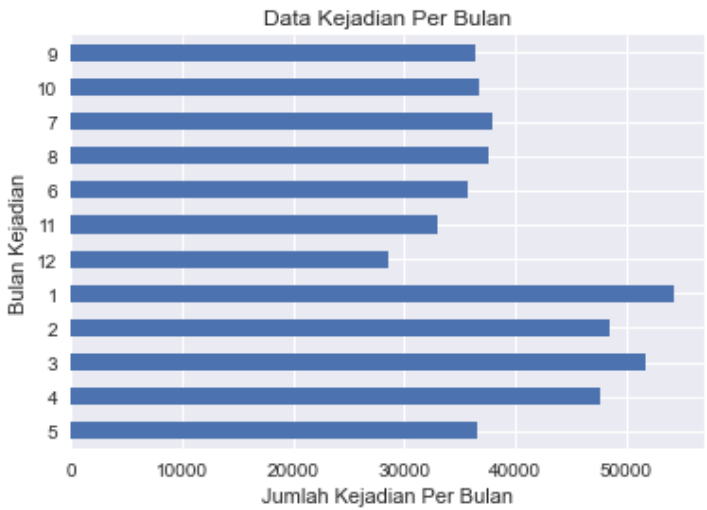
plt.title = digunakan untuk menentukan judul visualisasi yang digambarkan dan menampilkan judul menggunakan berbagai atribut.title()

plt.xlabel = digunakan untuk memberikan nama label untuk garis x.

plt.ylabe = digunakan untuk memberikan nama label untuk garis y.

plt.show = digunakan untuk menampilkan plot.

Berikut tampilan kejadian berdasarkan bulan



Gambar 4. 49 Hasil Kejadian Berdasarkan Bulan

Selanjutnya untuk melihat kejadian berdasarkan hari

Pada gambar diatas, yang paling banyak kejadian kejahatan yaitu pada bulan 1 dengan kejadian sebanyak 50000 lebih kejadian.

```
%matplotlib inline

main_df['Date'].apply(lambda x: x.weekday()).value_counts(sort=False).plot(kind='barh')
plt.xlabel("Jumlah Kejadian Per Per Hari")
plt.ylabel("Hari Kejadian")
plt.title("Data Kejadian Per Hari")
plt.show()
```

Gambar 4. 50 Berdasarkan Hari

%matplotlib inline = berfungsi agar hasil visualisasi bisa langsung tercetak di Jupyter Notebook.

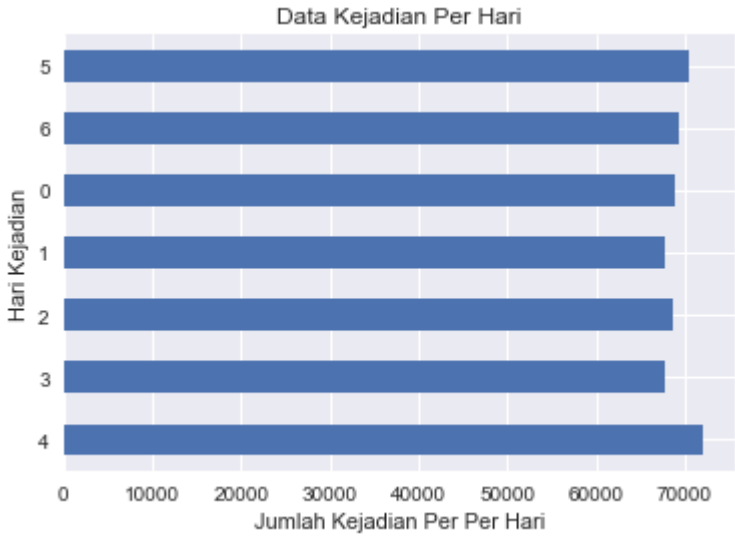
plt.title = digunakan untuk menentukan judul visualisasi yang digambarkan dan menampilkan judul menggunakan berbagai atribut.title()

plt.xlabel = digunakan untuk memberikan nama label untuk garis x.

plt.ylabe = digunakan untuk memberikan nama label untuk garis y.

plt.show = digunakan untuk menampilkan plot.

Berikut tampilan kejadian berdasarkan hari



Gambar 4. 51 Hasil Kejadian Berdasarkan Hari

Pada gambar diatas, yang paling banyak kejadian terdapat pada hari ke 4 dengan kejadian sebanyak 70000 lebih kejadian.

Selanjutnya untuk melihat kejadian berdasarkan Jam.

```
%matplotlib inline
main_df['Date'].apply(lambda x: x.hour).value_counts(sort=False).plot(kind='bar')
plt.xlabel("Jumlah Kejadian Per Jam")
plt.ylabel("Jam Kejadian")
plt.title("Data Kejadian Per Jam")
plt.show()
```

Gambar 4. 52 Berdasarkan Jam

%matplotlib inline = berfungsi agar hasil visualisasi bisa langsung tercetak di Jupyter Notebook.

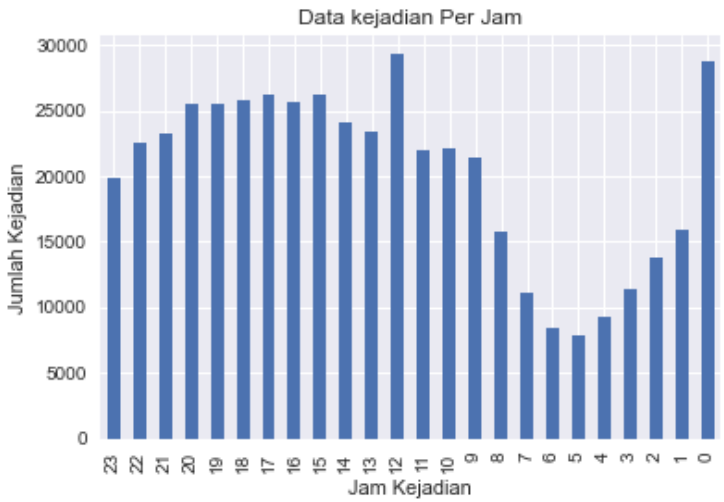
plt.title = digunakan untuk menentukan judul visualisasi yang digambarkan dan menampilkan judul menggunakan berbagai atribut.title()

plt.xlabel = digunakan untuk memberikan nama label untuk garis x.

plt.ylabe = digunakan untuk memberikan nama label untuk garis y.

plt.show = digunakan untuk menampilkan plot.

Berikut tampilan kejadian berdasarkan Jam



Gambar 4. 53 Hasil Kejadian Berdasarkan Jam

Pada gambar diatas, yang paling banyak kejadian terdapat pada jam ke 12 dengan kejadian sebanyak 30000 kejadian.

Selanjutnya untuk melihat kejadian berdasarkan lokasi.

```
%matplotlib inline
main_df['Location Description'].value_counts()
```

Gambar 4. 54 Berdasarkan Lokasi

%matplotlib inline = berfungsi agar hasil visualisasi bisa langsung tercetak di Jupyter Notebook.

Main_df = digunakan untuk menampung nilai array.

Value_counts() = merupakan fungsi untuk menghitung nilai dari array.

Berikut tampilan kejadian berdasarkan lokasi.

```
STREET                 118128
APARTMENT               89202
RESIDENCE               81858
SIDEWALK                30433
SMALL RETAIL STORE      12604
                        ...
CTA PROPERTY                1
CHA ELEVATOR                1
DRIVEWAY                    1
ELEVATOR                    1
STAIRWELL                   1
Name: Location Description, Length: 172, dtype: int64
```

Pada gambar diatas, yang paling banyak kejadian yaitu di tempat *street* (Jalanan) dengan kejadian sebanyak 118128.

4.4.4 Model

Pada tahap model ini, akan dilakukan proses pemodelan dengan menggunakan metode random forest untuk menentukan tingkat akurasi dari penelitian ini, untuk menentukan accuracy berikut.

Untuk pengujian dataset dengan model, sebelumnya akan meload terlebih dahulu data data yang akan dibutuhkan. Disini akan melakukan pengujian dari tahapan yang sebelumnya telah dilakukan.

```
# Load the Dataset
test_files = ['crimes_2019.csv','crimes_2020.csv','crimes_2021.csv']
test_files = [DATA_PATH+x for x in test_files]
test_df = us.create_df(test_files)
```

Gambar 4. 55 Meload Dataset

Pada tahap ini berfungsi untuk meload dataset yang digunakan. Pada tahap ini bernama test_file dimana nama variable ini berfungsi untuk mengetes dataset yang digunakan.

Tahap selanjutnya untuk menghilangkan nilai kosong.

```
# Drop missing values
test_df = test_df.dropna()
```

Gambar 4. 56 Menghilangkan Nilai Kosong

Pada tahap ini, nilai yang kosong akan dihapus menggunakan fungsi dropna().

Tahap selanjutnya yaitu *feature engineering*.

```
# Feature Engineering our columns
test_df['Month'] = test_df['Date'].apply(month_col)
test_df['Day'] = test_df['Date'].apply(day_col)
test_df['Hour'] = test_df['Date'].apply(hour_col)
```

Gambar 4. 57 Feature Engineering

Pada tahap ini dilakukan rekayasa fitur, dimana pada kolom *date* dipecah menjadi 3 bagian, yaitu bulan, hari dan jam.

Tahap selanjutnya yaitu compres dataset

```
# Compressing
df7 = filter_top_10(test_df)
cri7 = df7.groupby(["Month", "Day", "District", "Hour"], as_index=False).agg({"Primary Type" : "count"})
cri7 = cri7.sort_values(by=["District"], ascending=False)
cri8 = cri7.rename(index=str, columns={"Primary Type" : "Jumlah_Kejahatan"})
cri8 = cri8[["Month", "Day", "District", "Hour", "Jumlah_Kejahatan"]]
cri8['Alarm'] = cri8['Jumlah_Kejahatan'].apply(crime_rate_assign)
cri8 = cri8[['Month','Day','Hour','District','Jumlah_Kejahatan','Alarm']]
print(cri8.head(90))
```

Pada tahap ini dilakukan kompres data, artinya hanya menggunakan kolom kolom yang akan digunakan.

Berikut tampilan hasil dari test data diatas tersebut.

```
Proses Loading Dataset 2019.
Proses Loading Dataset 2020.
Proses Loading Dataset 2021.
Prses Loading Dataframe Selesai.


       Month  Day  Hour  District  Crime_Count  Alarm
21329      6    5     7        31            1      0
28101      8    4     0        31            1      0
22895      7    1     9        31            1      0
7306       2    6    12        31            1      0
35861     10    5     2        31            1      0
Class Imbalance

0    39933
1     3589
2       14
Name: Alarm, dtype: int64
```

Gambar 4. 58 Hasil Testing

Membuat *balance* dataset, membuat data seimbang.

```
from sklearn.utils import resample # for upsampling

# Set individual classes
cri6_low = cri6[cri6['Alarm']==0]
cri6_medium = cri6[cri6['Alarm']==1]
cri6_high = cri6[cri6['Alarm']==2]

# Upsample the minority classes to size of class 1 (medium)
cri6_low_upsampled = resample(cri6_low,
                              replace=True,     # sample with replacement
                              n_samples=22640,    # to match majority class
                              random_state=101)

cri6_high_upsampled = resample(cri6_high,
                              replace=True,     # sample with replacement
                              n_samples=22640,    # to match majority class
                              random_state=101)

# Combine majority class with upsampled minority class
cri6_upsampled = pd.concat([cri6_medium, cri6_low_upsampled, cri6_high_upsampled])
```

Gambar 4. 59 Balanced Dataset

Untuk melakukan prediksi, diperlukan beberapa library seperti berikut

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report
import joblib
import matplotlib.pyplot as plt
```

Gambar 4. 60 Library

Train_test_split = merupakan *library* yang digunakan untuk membagi dataset menjadi train dan test.

Standarscaller = untuk mengubah data sedemikian rupa sehingga distribusinya akan memiliki nilai rata-rata 0 dan deviasi standar 1.

RandomForestClassfier = merupakan model yang digunakan dalam prediksi ini.

Metrics = merupakan ukuran penilaian kuantitatif untuk menilai, membandingkan, dan melacak kinerja.

Confusion matrix = tabel yang digunakan untuk menggambarkan kinerja model klasifikasi pada set data pengujian yang nilainya telah diketahui.

Classification_report = untuk memberikan laporan hasil prediksi.

Matplotlib = digunakan untuk membuat diagram.

Selanjutnya lakukan proses prediksi dengan model random forest.

```
X_train, X_test, y_train, y_test = train_test_split (X, y, test_size=0.25, random_state=101)



classifier = RandomForestClassifier(n_estimators = 1000, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)
```

Gambar 4. 61 Proses Prediksi

X_train = adalah variable dataset yang digunakan untuk proses training. Pada proses training dilakukan sebanyak 75 % dataset yang digunakan.

X_test = adalah variable dataset yang digunakan untuk proses test dataset. Pada test dataset tersebut dilakukan sebanyak 25% dari data yang digunakan.

Train_test_split = adalah memecah dataset training dan test.

Test_size = adalah ukuran dataset yang diuji, dimana disini ada 0.25 berarti 25% dataset yang diuji.

n_estimators = merupakan jumlah pohon yang diuji, disini ada 1000 pohon.

random_state = adalah status acak dari random forest tersebut.

kemudian untuk menampilkan hasil dari prediksi.

```
y_pred = classifier.predict(X_test)
print(y_pred)
```

Gambar 4. 62 Untuk Menampilkan Hasil Prediksi

Nilai y_pred diambil dari nilai classifier, selanjutnya classifier tersebut diberi fungsi predict untuk memberikan hasil prediksi. Hasil prediksi sendiri diambil dari classifier.predict(X_test), jadi data yang ditampilkan adalah data dari nilai X_test. Untuk menampilkan hasil prediksi tersebut, maka disini menggunakan

perintah *print* diikuti dengan variable y_pred untuk mengeluarkan hasil dari nilai prediksi y_pred tersebut.

Pada saat tampil hasil random forest tersebut berbentuk array seperti ini. Hasil ini didapat setelah mengeluarkan hasil prediksi dari y_pred.

```
[2 0 0 ... 2 0 0]
```

Gambar 4. 63 Hasil Random Forest Array

Selanjutnya untuk menghasilkan nilai hasil prediksi tersebut.

```
cm = pd.crosstab(y_test, y_pred, rownames=['Kejahatan Sebenarnya'], colnames=['Prediksi Kejahatan'])
print(cm)
```

Gambar 4. 64 Hasil Random Forest

cm = merupakan variable untuk menampung nilai.

pd = merupakan library dari pandas.

Crosstab = merupakan fungsi yang tersedia di dalam *library* pandas untuk menghitung tabulasi silang dari dua (atau lebih) factor.

y_test = merupakan variabel untuk menampung nilai test.

y_pred = merupakan variable untuk menampung nilai prediksi.

print = digunakan untuk menampilkan nilai. Disini nilai yang ditampilkan yaitu cm.

berikut tampilan hasil dari prediksi nilai cm.

Prediksi Kejahatan	0	1	2
Kejahatan Sebenarnya			
0	5499	177	0
1	478	422	2
2	0	0	5640

Gambar 4. 65 Hasil Prediksi

Untuk menampilkan nilai accuracy dari hasil model yang diprediksi.

```
print("Accuracy:",(metrics.accuracy_score(y_test, y_pred)*100),"\n")
```

Gambar 4. 66 Menampilkan Hasil Accuracy

Berikut tampilan hasil *accuracy*

```
Accuracy: 94.62268783761662
```

Gambar 4. 67 Hasil Accuracy

Untuk menampilkan hasil *classification report* tersebut.

```
print("\n----------Classification Report----------------------------------")
print(classification_report(y_test,y_pred))
```

Gambar 4. 68 Classification Report

Berikut tampilan hasil *classification report.*

```
----------Classification Report-------------------------------------
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      5676
           1       0.70      0.47      0.56       902
           2       1.00      1.00      1.00      5640

    accuracy                           0.95     12218
   macro avg       0.87      0.81      0.84     12218
weighted avg       0.94      0.95      0.94     12218
```

Gambar 4. 69 Hasil Classification Report

Disini terdapat beberapa keterangan.

Accuracy = Akurasi merupakan tingkat ketepatan atau seberapa dekat suatu pengukuran dengan nilai sebenarnya.

Precision = Mendeskripsikan kemungkinan suatu sampel berada di kelas X jika sudah diprediksi masuk ke dalam kelas X.

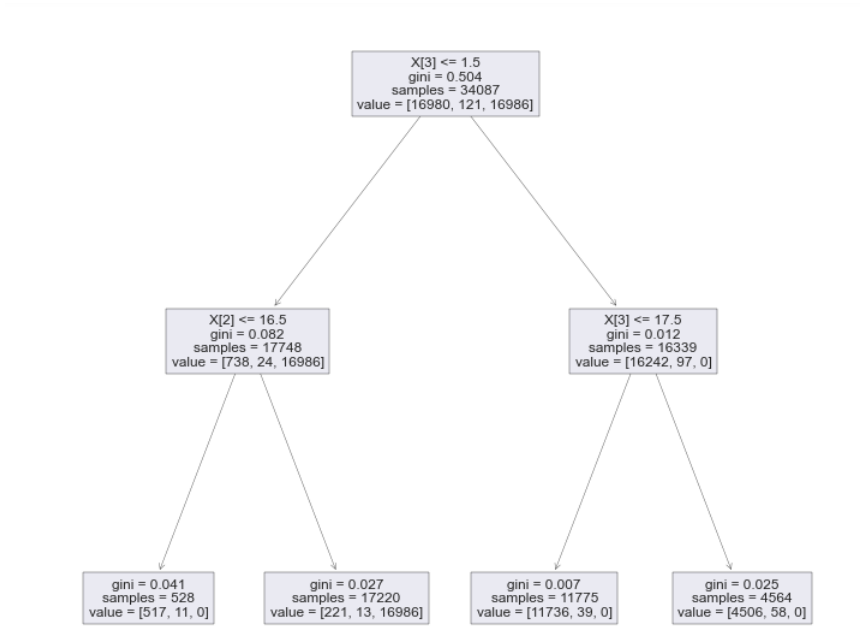
Recall = Menjelaskan kemungkinan model untuk mengklasifikasikan sampel dengan benar sebagai milik kelas X jika itu benar benar milik kelas X.

F1 = Ukuran yang menggambarkan hubungan antara presisi dan mengingat. Semakin tinggi skor F1 untuk suatu kelas, semakin baik kinerja model dalam mengklasifikasi kelas tersebut.

macro avg = merupakan nilai rata rata macro

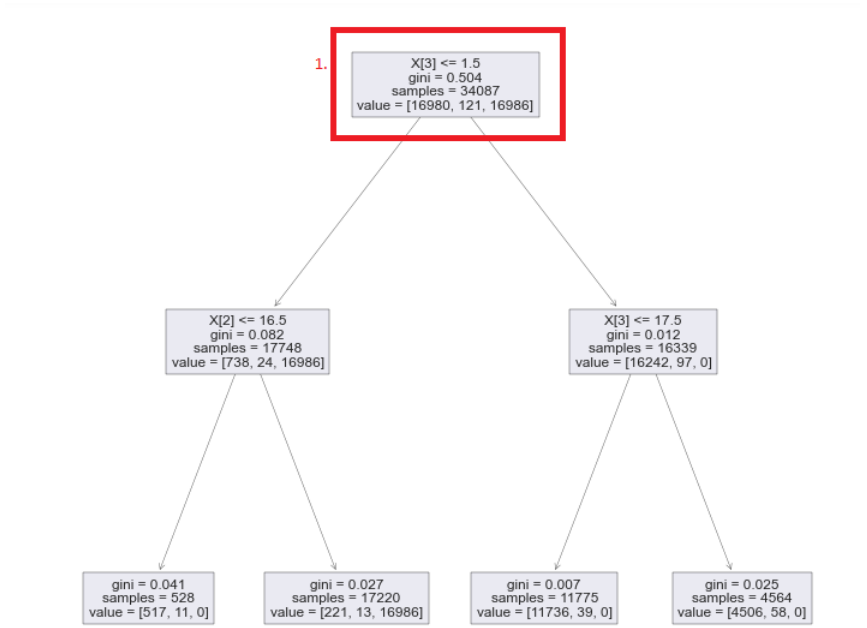
weight avg = rata-rata yang memperhitungkan pentingnya setiap angka dalam pembuatan rata-rata.

Selanjutnya Berikut adalah tampilan pohon keputusan



Gambar 4. 70 Pohon Keputusan

Berikut penjelasan dari pohon keputusan

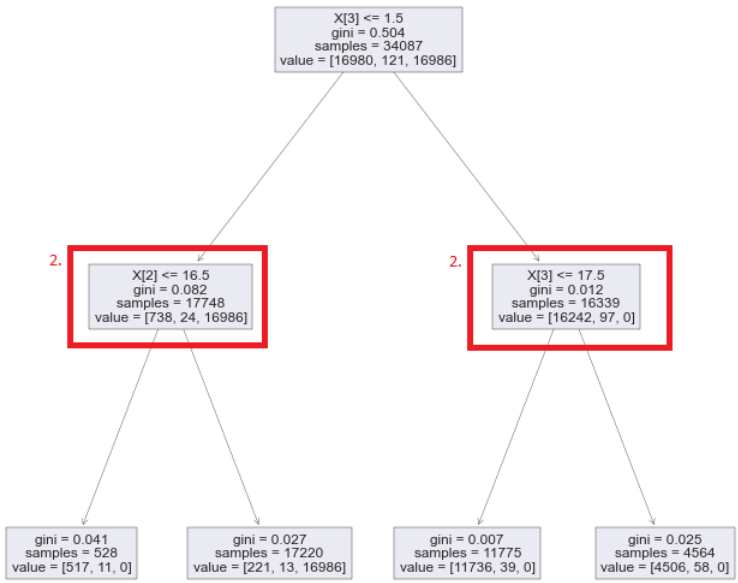


Gambar 4. 71 Root Node

Pada bagian no 1 yang ditandai dinamakan dengan *root node* (akar),

root node (akar) merupakan tujuan akhir atau keputusan besar yang ingin diambil. [35]

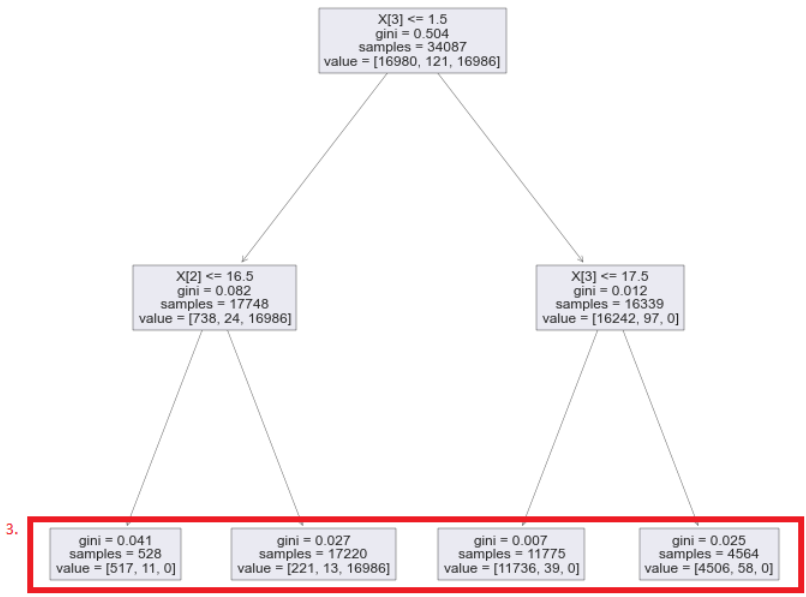
Selanjutnya terdapat bagian *branches* (ranting)



Gambar 4. 72 Branches

Branches (ranting) merupakan berbagai pilihan tindakan. Tindakan yang diambil dalam pohon keputusan setelah dari akar. [35]

Selanjutnya ada bagian *leaf node* (daun)



Gambar 4. 73 Leaf Node

Leaf node (daun) merupakan berbagai pilihan tindakan. Setelah dari *branches* (ranting) selanjutnya dalam pohon keputusan memilih tindakan apa yang dilakukan untuk proses selanjutnya.[35]

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada hasil evaluasi menggunakan model random forest, didapatkan *accuracy* pengujian model tersebut sebesar 94% dari hasil pengujian data.

Pada kejadian diatas, didapatkan kesimpulan bahwa tempat yang sering kejadian berada di *street* (jalanan), pada bulan ke 1 (Januari), hari ke 4 (Kamis) dan jam 12.00.

5.2 Saran

Berikut beberapa saran yang bisa dilakukan pada penelitian ini diantaranya :

Pada sistem ini dapat dikembangkan untuk melakukan prediksi pada tingkat lebih dari satu jenis kejahatan yang lain dan ruang lingkup yang lebih luas dalam konteks jenis kejahatan.

DAFTAR PUSTAKA

[1] D. Winarti, M. Kom, E. Revita, dan M. Kom, "Penerapan Data Mining untuk Analisa Tingkat Kriminalitas Dengan Algoritma Association Rule Metode FP-Growth," *J. SIMTIKA*, vol. 4, no. 3, hal. 8–22, 2021.

[2] R. Yadhunath, S. Srikanth, A. Sudheer, dan S. Palaniswamy, "Identification of Criminal Activity Hotspots using Machine Learning to Aid in Effective Utilization of Police Patrolling in Cities with High Crime Rates," *CSITSS 2019 - 2019 4th Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. Proc.*, 2019, doi: 10.1109/CSITSS47250.2019.9031057.

[3] N. O. Syamsiah dan I. Purwandani, "Penerapan Ensemble Stacking untuk Peramalan Laba Bersih Bank Syariah Indonesia (BSI)," *Build. Informatics, Technol. Sci.*, vol. 3, no. 3, hal. 295–301, 2021, doi: 10.47065/bits.v3i3.1017.

[4] K. Kumari, M. Bhardwaj, dan S. Sharma, "OSEMN Approach for Real Time Data Analysis," *Int. J. Eng. Manag. Res.*, vol. 10, no. 02, hal. 107–110, 2020, doi: 10.31033/ijemr.10.2.11.

[5] I. Pengaruh *et al.*, "Identifikasi Pengaruh Sistem Keamanan Lingkungan Terhadap Tingkat Kejahatan Pencurian Di Kota Surakarta Dengan Metode Sistem Informasi Geografis," *J. Geod. Undip*, vol. 8, no. 1, hal. 398–407, 2019.

[6] W. Safat, S. Asghar, dan S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," *IEEE Access*, vol. 9, hal. 70080–70094, 2021, doi: 10.1109/ACCESS.2021.3078117.

[7] M. Rianto dan R. Yunis, "Analisis Runtun Waktu Untuk Memprediksi Jumlah Mahasiswa Baru Dengan Model Random Forest," *Paradig. - J. Komput. dan Inform.*, vol. 23, no. 1, 2021, doi: 10.31294/p.v23i1.9781.

[8] R. Rusmiati, S. Syahrizal, dan M. Din, "Konsep Pencurian Dalam

Kitab Undang-Undang Hukum Pidana dan Hukum Pidana Islam," *Syiah Kuala Law J.*, vol. 1, no. 1, hal. 339–352, 2018, doi: 10.24815/sklj.v1i1.12318.

[9] A. N. Syahrudin dan T. Kurniawan, "Input Dan Output Pada Bahasa," *J. Dasar Pemrograman Python STMIK*, no. January, hal. 1–7, 2018.

[10] T. Wahyono, *Fundamental Of Python For Machine Learning*, Cetakan I. Yogyakarta: Gava Media, 2018.

[11] R. I. F. Ibadurrohman, D. R. Wijaya, dan E. Hernawati, "Pengembangan Aplikasi Machine Learning Menggunakan Algoritma Support Vector Regression Dan Statistical-based Feature Selection Untuk Memprediksi Kemiskinan," *eProceedings Appl. Sci.*, vol. 6, no. 2, hal. 1910–1917, 2020, [Daring]. Tersedia pada: https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/view/12271.

[12] E. Retnoningsih dan R. Pramudita, "Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python," *Bina Insa. Ict J.*, vol. 7, no. 2, hal. 156, 2020, doi: 10.51211/biict.v7i2.1422.

[13] I. D. Id, *MACHINE LEARNING : Teori, Studi Kasus dan Implementasi Menggunakan Python*, Edisi I. Riau: UR PRESS, 2021.

[14] N. Buslim dan R. P. Iswara, "Pengembangan Algoritma Unsupervised Learning Technique Pada Big Data Analysis di Media Sosial sebagai media promosi Online Bagi Masyarakat," *J. Tek. Inform.*, vol. 12, no. 1, hal. 79–96, 2019, doi: 10.15408/jti.v12i1.11342.

[15] A. Roihan, P. A. Sunarya, dan A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 5, no. 1, hal. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.

[16] A. R. Muslikh, H. A. Santoso, A. Marjuni, P. Teknik, I. Universitas, dan D. Nuswantoro, "Klasifikasi Data Time Series Arus Lalu Lintas," vol. 14, hal. 24–38, 2018.

[17] Adversenia, "10 Manfaat dan Contoh Penerapan Machine Learning," *www.advernesia.com*. https://www.advernesia.com/blog/data-science/10-manfaat-dan-contoh-penerapan-machine-learning/.

[18] M. Harahap, F. Rozi, Y. Yennimar, dan S. D. Siregar, "Analisis Wawasan Penjualan Supermarket dengan Data Science," *Data Sci. Indones.*, vol. 1, no. 1, hal. 1–7, 2021, doi: 10.47709/dsi.v1i1.1173.

[19] N. Rahmalia, "Data Science: Arti, Manfaat, Proses, dan Contoh Penerapannya," *glints.com*, 2020. https://glints.com/id/lowongan/data-science-adalah/.

[20] "17 Data Science Applications and Examples," *builtin.com*, 2019. https://builtin.com/data-science/data-science-applications-examples.

[21] J. Sanjaya, E. Renata, V. E. Budiman, F. Anderson, dan M. Ayub, "Prediksi Kelalaian Pinjaman Bank Menggunakan Random Forest dan Adaptive Boosting," *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 1, hal. 50–60, 2020, doi: 10.28932/jutisi.v6i1.2313.

[22] A. Rosadi *et al.*, "Analisis Sentimen Berdasarkan Opini Pengguna pada Media Twitter Terhadap BPJS Menggunakan Metode Lexicon Based dan Naïve Bayes Classifier," *J. Ilm. Komputasi*, vol. 20, no. 1, hal. 39–52, 2021, doi: 10.32409/jikstik.20.1.401.

[23] S. Ghosh, "Imbalanced vs Balanced Dataset in Machine Learning," *medium.com*, 2019. https://medium.com/open-datascience/imbalanced-vs-balanced-dataset-in-machine-learning-4faec5629b7e.

[24] K. Kurniawan dan D. Antoni, "Visualisasi Data Penduduk Dalam Membangun E-government Berbasis Sistem Informasi Geografis (GIS)," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, hal. 310–316, 2020, doi: 10.32736/sisfokom.v9i3.828.

[25] R. E. Nalawati dan D. Y. Liliana, "Visualisasi Data Program Vaksinasi Covid-19 di Kota Depok dengan Big Data Analytics," *J. Media Inform. …*, vol. 5, hal. 1570–1579, 2021, doi: 10.30865/mib.v5i4.3330.

[26] Yasir Abdur Rohman, "Pengenalan NumPy, Pandas, Matplotlib," *medium.com*, 2019. https://medium.com/@yasirabd/pengenalan-numpy-pandas-matplotlib-b90bafd36c0.

[27] H. Mccaslin, "Visualisai Data Menggunakan Seaborn dan Matplotlib | Python," *medium.com*, 2020. https://medium.com/@HakimMccaslin/apa-itu-seaborn-234a224d946f.

[28] Muthukrishnan, "Understanding the Classification report through sklearn," *medium.com*, 2018. https://muthu.co/understanding-the-classification-report-in-sklearn/.

[29] M. Z. Asy'ari, "Cara Menggunakan Jupyter Notebook dengan Mudah," *auftechnique.com*, 2020. https://auftechnique.com/cara-menggunakan-jupyter-notebook/.

[30] A. U. Jamila, B. M. Siregar, dan R. Yunis, "Analisis Runtun Waktu Untuk Memprediksi Jumlah Mahasiswa Baru Dengan Model Arima," *Paradig. - J. Komput. dan Inform.*, vol. 23, no. 1, hal. 99–105, 2021, doi: 10.31294/p.v23i1.9758.

[31] F. Hamami dan I. A. Dahlan, "Klasifikasi Cuaca Provinsi Dki Jakarta Menggunakan Algoritma Random Forest Dengan Teknik Oversampling," *J. Teknoinfo*, vol. 16, no. 1, hal. 87, 2022, doi: 10.33365/jti.v16i1.1533.

[32] S. Devella, Y. Yohannes, dan F. N. Rahmawati, "Implementasi Random Forest Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 2, hal. 310–320, 2020, doi: 10.35957/jatisi.v7i2.289.

[33] F. Dan, R. Forest, dan R. F. Method, "(Cow Weight Classification Based on Digital Image Using Fractal and," vol. 8, no. 2, hal. 1472–1480, 2021.

[34] S. M. Rezkia, "Macam-Macam Metode Analisis Data: 2 Macam Metode Penting dalam Mengolah Data," 2020. https://dqlab.id/data-analisis-pahami-2-metode-analisis-data.

[35] G. N. Arviana, "Coba Metode Decision Tree bagi Kamu yang Sulit Ambil Keputusan," *glints.com*, 2020. https://glints.com/id/lowongan/decision-tree-adalah/#.YiBk-KtBzIU.

LAMPIRAN – LAMPIRAN

FORMAT PENILAIAN INTERNSHIP
PROGRAM STUDI D4 TEKNIK INFORMATIKA
POLITEKNIK POS INDONESIA

NAMA	NPM	Tempat Tgl Lahir
PARHAN HAMBALI	1184042	Bandung, 15 Oktober 2000

JUDUL INTERNSHIP : Rancang Bangun Sistem Sinkronisasi Surat Keputusan Direksi (Studi Kasus PT Pos Indonesia)

PEMBIMBING EKSTERNAL : Suprapto

NO	KOMPONEN YANG DI NILAI	NILAI MAKS	PENILAIAN (ANGKA)	RATA - RATA
1.	PENAMPILAN INDIVIDUAL			
	A. Penampilan Berpakaian	7	6	
	B. Sikap Terhadap Orang Lain	8	6	
	C. Semangat Bekerja	7	6	
	D. Kematangan Dalam Bertindak	6	6	
	E. Adaptasi Tempat Kerja	6	6	
	F. Pengetahuan Yang Mendukung Pekerjaan	6	6	
	G. Kehadiran Ditempat Kerja	8	6	
2.	KINERJA PKL			
	A. Ketelitian & Ketepatan Dalam Bekerja	8	6	
	B. Kwalitas Produk / Kerja	8	6	
	C. Kemandirian Dalam Melaksanakan Pekerjaan	7	6	
	D. Kemampuan Bekerjasama	7	6	
	E. Kemampuan Berkomunikasi	8	6	
	F. Manajemen Waktu	7	6	
	G. Kemampuan Menganalisa Masalah	7	6	
TOTAL			84	

BANDUNG, 31 Desember 2021

(SUPRAPTO)
NIPPOS : 969349427

Sinopsis

Pencurian merupakan kejahatan yang ditujukan terhadap harta benda dan paling sering terjadi di dalam masyarakat. Kejahatan ini merupakan tindakan kejahatan yang dapat menggoncangkan stabilitas keamanan baik terhadap harta maupun terhadap jiwa masyarakat. Oleh karena itu, baik dalam Kitab Undang - Undang Hukum Pidana (KUHP) maupun dalam Kitab Suci melarang keras tindakan kejahatan tersebut dan menegaskan ancaman hukuman secara rinci dan berat atas diri pelanggarnya. Hal ini dapat dilihat dari bentuk hukuman dan ancaman hukuman yang dijatuhkan.Untuk meminimalisir pencurian, dari daerah yang mempunyai tingkat kejahatan tinggi. Maka pada tahap ini membuat klasifikasi pelabelan untuk mengetahui tingkat keamanan suatu daerah, selanjutnya dilakukan prediksi seberapa akurat prediksi tersebut menggunakan model OSEMN. Penentuan model menggunakan Random Forest dipilih karena Random Forest merupakan salah satu metode yang digunakan untuk klasifikasi dan regresi. Random Forest dapat digunakan untuk time series dengan cara meningkatkan keakuratan metode klasifikasi dengan menggabungkan metode klasifikasi dalam kata lain Random Forest dilakukan secara ensemble.