LAPORAN AKHIR

KLASIFIKASI KONDISI TUBUH MAHASISWA TEKNOLOGI SAINS DATA ANGKATAN 2021 UNIVERSITAS AIRLANGGA BERDASARKAN SURVEI KESEHATAN MENGGUNAKAN MODEL MACHINE LEARNING

KELOMPOK: A1 - 03

1. Diaz Arvinda Ardian	162112133009
2. Vaness Nakanaori T.	162112133067
3. Fransiscus Ernest O.	162112133081
4. Aditya Ananda	162112133095
5. Stevanus Sembiring	162112133099

TEAM BASED PROJECT
MATA KULIAH DATA MINING I
PROGRAM STUDI S1 TEKNOLOGI SAINS DATA
FAKULTAS TEKNOLOGI MAJU DAN MULTIDISIPLIN
UNIVERSITAS AIRLANGGA
2023

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

Kesehatan merupakan hal yang sangat penting bagi setiap orang, termasuk mahasiswa Teknologi Sains Data Angkatan 2021 Universitas Airlangga. Kesehatan tubuh yang baik dapat memengaruhi kinerja dan produktivitas mahasiswa dalam menyelesaikan tugas-tugas akademik. Oleh karena itu, penelitian tentang faktor-faktor yang berhubungan dengan kesehatan tubuh mahasiswa sangat penting untuk dilakukan. Beberapa faktor seperti pola makan, tingkat kebugaran fisik, manajemen stres, dan pola tidur dapat menjadi penentu bagaimana kondisi kesehatan seorang mahasiswa. Salah satu bidang yang dapat digunakan untuk meneliti bagaimana faktor-faktor tersebut berhubungan satu sama lain adalah *data science*. *Data science* merupakan bidang ilmu yang menggabungkan pengetahuan di bidang ilmu tertentu dengan keahlian pemrograman, matematika, serta statistika. Adapun peran dari *data science* adalah untuk mengekstrak sebuah pengetahuan atau informasi dari sebuah data yang dalam dunia nyata mayoritas tidak bermakna dan masih sangat kotor.

Dalam penelitian ini, objek penelitian adalah mahasiswa Teknologi Sains Data Angkatan 2021 Universitas Airlangga. Penelitian ini akan berbicara tentang klasifikasi biner dari kategori tubuh mahasiswa Teknologi Sains Data menggunakan beberapa model klasifikasi *machine learning*. Model klasifikasi machine learning yang akan dibangun seperti Regresi Logistik, Decision Tree Classifier, Random Forest Classifier, Naïve Bayes Classifier, Support Vector Machines (SVM), atau Artificial Neural Network (ANN). Mengingat pada zaman sekarang terdapat banyak tantangan untuk dapat menjalankan hidup sehat dan terkadang secara tidak sadar pola hidup yang dijalani sehari-hari tidaklah memberi efek yang baik terhadap kesehatan tubuh, dari situlah muncul inisiasi ditulisnya penelitian ini untuk mengetahui faktor apa saja yang dapat menentukan kondisi tubuh seseorang masih tergolong ideal atau tidak ideal dengan bantuan *machine learning*.

Penelitian ini bertujuan untuk mengetahui apakah model klasifikasi *machine learning* yang digunakan dapat menghasilkan klasifikasi terhadap kategori tubuh mahasiswa yang akurat. Selain itu, penelitian ini juga bertujuan untuk mengetahui faktor-faktor yang berhubungan dengan kategori tubuh mahasiswa Teknologi Sains Data Angkatan 2021 Universitas Airlangga. Penelitian yang dilakukan merupakan salah satu bentuk penerapan *data science* di bidang kesehatan yang berkaitan dengan kesehatan mahasiswa. Penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan ilmu *data science* dan juga bermanfaat sebagai pengetahuan akan kesehatan, selain itu penelitian ini diharapkan dapat memberikan manfaat bagi mahasiswa Teknologi Sains Data Angkatan 2021 Universitas Airlangga dalam menjaga kesehatan tubuh.

BAB 2
TINJAUAN PUSTAKA

2.1 Kesehatan dan Faktor-Faktor yang Berpengaruh

Analisis kesehatan ini akan memberikan pemahaman yang berharga bagi mahasiswa Teknologi Sains Data Angkatan 2021 mengenai pentingnya menjaga kesehatan secara menyeluruh. Dengan memahami kondisi kesehatan mereka, mereka dapat mengambil langkah-langkah pencegahan yang tepat dan mengadopsi gaya hidup yang sehat. Menurut kemenkes kesehatan adalah proses dinamis dalam mempertahankan dan mendukung keutuhan integritas manusia (keseimbangan fisik dan mental) dan adaptasinya dengan lingkungan sekitar secara optimal.

Terdapat faktor-faktor yang mempengaruhi kesehatan juga yaitu kurangnya pengetahuan mengenai gizi dan pola pengasuhan;asupan makanan yang kurang; keberadaan ancaman penyakit infeksi yang berulang; akses air bersih yang tidak memadai; higienis dan sanitasi yang buruk; keterbatasan (sulit) untuk mengakses pelayanan kesehatan; ketersediaan pangan; kondisi sosial dan pendapatan (ekonomi); hingga ketersediaan stok bahan bakar minyak (Kemkes.go.id, 2018). selain itu terdapat juga faktor genetik atau keturunan dapat mempengaruhi kecenderungan seseorang terhadap penyakit tertentu sehingga membutuhkan pengobatan lebih.

2.2 Survei Kesehatan

2020-2022 | 2017-2019 | 2014-2016 | 2011-2013 | 2008-2010 | 2005-2007 | 2002-2004 | 2000-2001

Provinsi	Persentase Penduduk yang Mempunyai Keluhan Kesehatan Selama Sebulan Terakhir (Persen)		
	2020	2021	2022
SUMATERA SELATAN	29,32	27,91	32,30
BENGKULU	30,23	26,63	26,66
LAMPUNG	31,35	28,44	32,52
KEP. BANGKA BELITUNG	31,25	26,16	35,11
KEP. RIAU	18,21	14,72	18,41
DKI JAKARTA	33,80	25,98	16,76
JAWA BARAT	32,04	29,74	31,17
JAWA TENGAH	35,63	29,81	35,34
DI YOGYAKARTA	38,07	30,20	35,73
JAWA TIMUR	32,80	28,55	32,14
BANTEN	32,22	28,41	24,09
BALI	25,48	23,62	20,45
NUSA TENGGARA BARAT	44,00	42,15	43,62
NUSA TENGGARA TIMUR	34,44	30,14	29,06

Figure 1. Survei kesehatan masyarakat Indonesia (2020-2022)

Survei kesehatan adalah metode pengumpulan data yang digunakan untuk memperoleh informasi tentang status kesehatan dan faktor-faktor terkait kesehatan di suatu populasi dalam periode waktu tertentu. Survei ini dapat dilakukan oleh pemerintah, organisasi kesehatan, atau lembaga penelitian untuk memahami tren kesehatan, menganalisis masalah kesehatan, dan merencanakan intervensi yang tepat.

Pada *figure* 1. terlihat bahwa banyak masyarakat mengalami keluhan dan beberapa provinsi banyak mengalami kenaikan. terutama beberapa provinsi yang tentunya mengalami kekurangan tenaga kesehatan. selain itu survei kesehatan tidak hanya dipantau dari kesehatan jasmani saja. namun, kesehatan rohani yang dimana pemerintah juga memantau prevalensi gangguan kesehatan mental, seperti depresi, kecemasan, stres, dan tingkat kesejahteraan mental secara umum. Informasi ini penting dalam merancang intervensi dan dukungan yang sesuai untuk meningkatkan kesehatan mental masyarakat.

2.3 Klasifikasi Kesehatan

Klasifikasi kesehatan mengacu pada proses penggunaan metode analisis data untuk mengelompokkan individu atau sampel ke dalam kategori kesehatan tertentu berdasarkan informasi yang ada dalam data. Tujuan utama klasifikasi kesehatan pada penelitian ini adalah untuk mengembangkan model prediktif yang dapat mengklasifikasikan dengan akurat individu baru berdasarkan data yang tersedia. Klasifikasi kesehatan penting untuk membantu pemerintah, peneliti, dan profesional kesehatan dalam memahami keadaan kesehatan masyarakat, mengidentifikasi masalah kesehatan yang perlu ditangani, serta merencanakan dan melaksanakan intervensi yang sesuai untuk meningkatkan kesehatan individu dan populasi.

2.4 Data Preprocessing

Data preprocessing merupakan langkah awal dan sangat penting dalam data mining terutama pembangunan model klasifikasi *machine learning*. Memiliki sumber data yang baik tidak hanya dapat meningkatkan akurasi dalam proses mining, tetapi juga secara signifikan meningkatkan efisiensi algoritma yang digunakan (Xiang-wei and Qi Yian-fang, 2012). Pada umumnya, data preprocessing mencakup proses pembersihan data (*cleaning*), penggabungan data (*integration*), transformasi data (*transformation*), pengurangan data (*reduction*), dan sebagainya.

432.5 Machine Learning

Machine learning merupakan penerapan komputer dan algoritma matematika yang menggunakan data sebagai dasar pembelajaran untuk menghasilkan prediksi di masa depan. Terdapat tiga kategori utama dalam machine learning, yaitu Supervised Learning, Unsupervised Learning, dan Reinforcement Learning. Supervised Learning menggunakan data dengan label, sementara Unsupervised Learning tidak memerlukan label dan berfokus pada pengelompokan data. Reinforcement Learning berada di tengah-tengah antara keduanya. Metode Supervised Learning melibatkan algoritma seperti Back-propagation, Linear Regression, dan Neural Network. Terdapat juga algoritma lain seperti Support Vector Machines dan Decision Tree. (Roihan, Abas Sunarya and Rafika, 2019)

2.6 Model Klasifikasi Machine Learning (ML)

2.6.1 Logistic Regression

Logistic regression adalah sebuah pendekatan untuk membuat model prediksi seperti halnya regresi linear atau yang biasa disebut dengan istilah Ordinary Least Squares (OLS) regression. Perbedaannya adalah pada regresi logistik, peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori, misalnya: Ya dan Tidak, Baik dan Buruk atau Tinggi dan Rendah.(Ratnawati,2008)

2.6.2 Decision Tree Classifier

Decision tree classifier adalah algoritma non-parametrik, yang digunakan untuk tugas klasifikasi dan regresi. Ini memiliki hierarki, struktur pohon, yang terdiri dari simpul akar, cabang, simpul internal, dan simpul daun. Pohon keputusan dimulai dengan simpul akar, yang tidak memiliki cabang masuk. Cabang keluar dari simpul akar kemudian masuk ke simpul internal, juga dikenal sebagai simpul keputusan. Berdasarkan fitur yang tersedia, kedua tipe simpul melakukan evaluasi untuk membentuk himpunan bagian yang homogen, yang dilambangkan dengan simpul daun, atau simpul terminal (*What is a Decision Tree | IBM* 2023).

2.6.3 Naive Bayes

Model klasifikasi Naive Bayes adalah salah satu metode populer dalam analisis data yang digunakan untuk melakukan klasifikasi. Model ini didasarkan pada teorema Bayes dan mengasumsikan independensi kondisional antara fitur-fitur yang digunakan dalam klasifikasi.

Model klasifikasi Naive Bayes menggunakan Teorema Bayes, yang menyediakan kerangka kerja untuk menghitung probabilitas posterior berdasarkan probabilitas sebelumnya. Teorema Bayes dinyatakan sebagai berikut:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

- P(A|B) adalah probabilitas posterior dari A terjadi jika B terjadi.
- P(B|A) adalah probabilitas likelihood dari B terjadi jika A terjadi.
- P(A) adalah probabilitas sebelumnya dari A terjadi.
- P(B) adalah probabilitas dari B terjadi.

Model klasifikasi Naive Bayes adalah metode yang digunakan untuk klasifikasi data berdasarkan teorema Bayes dengan asumsi "kesederhanaan naif" (naive simplicity) dalam hal independensi fitur. Secara singkat, metode ini mengasumsikan bahwa setiap fitur dalam data independen secara statistik, meskipun dalam kenyataannya tidak selalu terjadi.

Model klasifikasi Naive Bayes biasanya digunakan dalam kasus dengan dataset yang relatif besar dan dimensi fitur yang tinggi. Meskipun asumsi yang sederhana, model ini sering memberikan hasil yang baik dan cocok untuk banyak masalah klasifikasi yang berbeda. Namun, perlu dicatat bahwa performanya dapat terpengaruh jika asumsi independensi fitur tidak terpenuhi secara signifikan.

2.6.4 Random Forest

Random Forest merupakan model ensemble (*ensemble learning*) berdasarkan decision tree (Curry, 2021). *Ensemble learning* sendiri merupakan metode dimana terdapat banyak algoritma *machine learning* yang dipadupadankan menjadi satu dalam satu waktu, adapun tujuan dari *ensemble learning* yakni memeroleh ukuran performa model yang lebih tinggi dibandingkan dengan model *machine learning* secara ‘individual’.

Algoritma yang ada dalam model Random Forest yaitu melibatkan banyak decision tree. Dalam pembuatan setiap decision tree, random forest menggunakan metode *bagging* atau *bootstrap aggregating* dan pengacakan fitur. *Bagging* sendiri merupakan proses di mana terdapat banyak model *machine learning* yang dilatih (*trained*) dalam algoritma yang sama dengan menggunakan sampel hasil resampling bootstrap (resampling menggunakan *random sampling with replacement*) dari dataset asli yang ada. Setelah model-model berhasil di-*train*, hasil prediksi dari tiap tree akan disimpan dan kumpulan prediksi tersebut akan dihitung dengan menggunakan metode; *vote* (untuk model klasifikasi) atau rata-rata (untuk model regresi).

2.6.5 K-Nearest Neighbor (KNN)

Algoritma KNN memiliki kelebihan dalam menangani data training dengan banyak noise dan jumlah data yang besar. Namun, kelemahan dari algoritma ini adalah perlu menentukan jumlah tetangga terdekat (nilai K), hasil perhitungan jarak kurang akurat, dan biaya komputasi yang tinggi karena memerlukan perhitungan jarak pada setiap query instance terhadap seluruh data training. KNN merupakan metode supervised learning yang mengklasifikasikan query instance berdasarkan mayoritas kategori KNN, dan pengumpulan data dilakukan melalui metode observasi (Cholil et al., 2020).

2.6.6 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) atau jaringan syaraf tiruan adalah sistem pemrosesan informasi yang menyerupai sistem saraf manusia dan mampu memecahkan masalah serupa SVM dan KNN melalui pelatihan data yang luas. ANN memiliki kemampuan untuk mentoleransi kesalahan sehingga menghasilkan prediksi yang akurat. Selain itu, metode ini juga dapat digunakan untuk memodelkan hubungan kompleks antara masukan (input) dan keluaran (output), serta mengidentifikasi pola pada data. Namun, kelemahan ANN adalah sulitnya menentukan jumlah optimal neuron dan lapisan, serta kemungkinan mengalami perlambatan dalam proses pembelajaran (Putra and Nabilah Ulfa Walmi, 2020).

2.6.7 Support Vector Machine (SVM)

Model klasifikasi SVM (Support Vector Machine) adalah metode yang mencari hiperplane pemisah terbaik antara dua kelas data. Hiperplane adalah bidang pembagi dalam ruang fitur. SVM berusaha memaksimalkan margin, yaitu jarak terdekat antara hiperplane dan data pelatihan. Prediksi kelas dilakukan dengan memasukkan data baru ke dalam rumus SVM. Jika hasilnya positif, data diklasifikasikan ke kelas satu; jika negatif, ke kelas dua. SVM juga bisa menggunakan kernel untuk mengklasifikasikan data yang tidak dapat dipisahkan secara linier di ruang fitur asli. Kernel memetakan data ke ruang fitur yang memiliki dimensi lebih tinggi, di mana data menjadi linier terpisah. Contoh kernel yang umum digunakan adalah kernel linear, kernel polinomial, dan kernel RBF.

2.7 Hyperparameter Tuning

Hyperparameter tuning adalah proses mencari kombinasi parameter terbaik dalam suatu model machine learning untuk meningkatkan kinerjanya. Hyperparameter merupakan parameter yang tidak dipelajari oleh model melalui pembelajaran, tetapi harus ditentukan oleh pengguna atau diatur secara manual.

Proses hyperparameter tuning umumnya dilakukan dengan membagi data menjadi data latih dan data validasi. Data latih digunakan untuk melatih model dengan setiap kombinasi hyperparameter, sedangkan data validasi digunakan untuk mengevaluasi performa model dan memilih kombinasi hyperparameter terbaik. Tujuan utama dari hyperparameter tuning adalah untuk mencapai model yang memiliki kinerja terbaik pada data yang belum pernah dilihat sebelumnya (data uji).

2.8 Ukuran Performa Model ML

Penilaian performa model *machine learning* direpresentasikan dalam suatu matriks bernama *confusion matrix*, dari matriks tersebut dapat dihitung ukuran performa akurasi, presisi, recall (sensitivitas), dan f1-score. Berikut ini merupakan isi dari *confusion matrix*;

Actual Vs Predicted	Positif	Negatif
Positif	*True Positive* (TP)	*False Negative* (FN)
Negatif	*False Positive* (FP)	*True Negative* (TN)

Table 1. Confusion Matrix

ukuran performa model dapat diperoleh dari matriks ini. Adapun detail formula untuk menentukan performa model dapat dicari sebagai berikut;

Pengukuran Performa	Formula
Accuracy (Akurasi)	$\frac{TP}{TP+TN+FP+FN}$
Recall (Sensitivitas)	$\frac{TP}{TP+FN}$
Precision (Presisi)	$\frac{TP}{TP+FP}$
F1-Score	$\frac{2*recall*precision}{recall+precision}$

Table 2. Performance Measures Formulas

terdapat suatu konsep yakni *micro*, *macro*, dan *weighted* untuk menghitung performa model dari semua kelas secara keseluruhan. Metode *micro* menghasilkan ukuran performa model yang seragam dengan akurasi, sedangkan metode *macro* akan menghitung rata-rata dari ukuran performa model seperti jika f1-score untuk prediksi; kelas 0 sebesar 0,8 dan kelas 1 sebesar 0,9 maka f1-score *macro* yang dihasilkan dari gabungan kedua kelas ini adalah sebesar (0,8+0,9) / 2 = 0,85. Untuk metode *weighted* sendiri didapatkan melalui formula penjumlahan dari ukuran performa model kelas i * proporsi jumlah kelas sampel (*support*), misalkan diambil contoh f1-score sebelumnya jika diketahui *support*; kelas 0 adalah 8 dan kelas 1 adalah 9, maka perhitungan f1-score metode *weighted* yakni; (0,8*(8/17)) + (0,9*(9/17)) = 0,3764 + 0,4764 = 0,8528.

Performa model tidak hanya dapat dilihat dari ukuran performa akurasi, sensitivitas, presisi, dan f1-score namun terdapat cara lain untuk mengetahui kebaikan dari model yakni dengan *Area Under the ROC Curve* atau kerap disebut AUC (Burkov, 2019, pp.16–19). Kurva ROC (*receiver operating characteristic*) merupakan kurva antara *true positive rate* (TPR sama seperti *recall*) dengan *false positive rate* (FPR = 1 - Spesifitas) atau secara matematis dapat dituliskan; $TPR = recall$ dan $FPR = \frac{FP}{(FP+TN)}$. Model klasifikasi yang sempurna akan memiliki nilai AUC 1.

2.9 Proses Pemilihan dan Evaluasi Model ML

Pemilihan model dapat didasarkan dari beberapa metode, salah satu metode yang dapat digunakan yaitu *cross-validation*. *Cross-validation* memberikan kemampuan untuk memperkirakan performa model pada data tak terlihat (*unseen data*) yang tidak digunakan saat pelatihan atau *model training* (Alhamid, 2020). Metode ini cocok digunakan ketika data yang digunakan terbatas karena pembagian data menjadi dua yakni *data training* dan *data testing* sangat berpengaruh terhadap akurasi *training*. Adapun cara kerja *cross-validation* sendiri yakni terbagi menjadi dua tahapan; yang pertama membagi *data training* menjadi bagian-bagian (disebut K-*fold*) kemudian yang kedua dari tiap bagian akan dipilih satu sebagai *data validation* dan sisanya, K-1 *fold*, menjadi *data training*, proses kedua ini akan diulang-ulang hingga seluruh *fold* telah menjadi *data validation*. Setelah selesai perulangan, maka dari setiap model tadi dihitung rata-rata ukuran performa modelnya misal akurasi dan hasil akurasi *cross-validation* inilah yang menjadi pertimbangan untuk pemilihan model. Dalam setiap *fold* mengandung jumlah data yang sama, misalkan terdapat 60 *data training* dan dipilih K = 5 maka setiap *fold* akan berisi 12 data serta pengelompokannya secara acak.

Jenis *cross-validation* (CV) sendiri beragam seperti K-Fold CV, Stratified K-Fold CV, Leave-one-out CV, Leave-pair-out CV, dan sebagainya. Secara garis besar algoritma CV sama seperti yang telah disebutkan sebelumnya, namun untuk leave-one-out dan leave-pair-out sedikit berbeda di mana metode leave-one-out akan menghasilkan model sebanyak observasi atau objek dengan mengambil tiap observasi menjadi *data validation* dan sisa observasi lainnya sebagai *data training*. Kemudian untuk leave-pair-out memiliki algoritma yang sama dengan leave-one-out namun dapat ditentukan berapa jumlah observasi yang akan menjadi *data validation* (leave-one-out merupakan kasus leave-pair-out dengan p = 1), kedua jenis CV ini *computationally intense* jika dibandingkan dengan K-Fold CV ataupun Stratified K-Fold CV karena membangun model yang lebih banyak dan oleh sebab itu jika jenis CV ini diterapkan ketika berhadapan dengan minimnya *data training* akan sesuai.

Proses untuk melakukan evaluasi terhadap model yang dipilih dapat digunakan visualisasi *learning curves*, di mana plot ini menyajikan informasi mengenai perbandingan ukuran performa seperti akurasi; antara *data training* (secara keseluruhan) dengan *data validation* (cross-validation) pada setiap ukuran sampel (dari 1 sampai n-sampel *data training*) dan dari plot tersebut dapat ditentukan apakah model yang dipilih mengalami kasus *underfitting* (terlalu sederhana dalam membaca pola), *overfitting* (terlalu menghafalkan pola data) atau sudah baik (*good-fit*).

BAB 3
METODOLOGI

3.1 Metode Pengambilan Sampel

Dalam survei ini, kami menggunakan populasi mahasiswa Teknologi Sains Data 2021 yang berjumlah 108 orang, dan sampel yang kami ambil sebanyak 61 orang. Kami menggunakan metode pengambilan sampel non probabilitas, di mana tidak semua anggota populasi memiliki peluang yang sama untuk dipilih sebagai sampel. Kami menyebarkan Google Form kepada seluruh mahasiswa Teknologi Sains Data 2021 untuk melakukan pengambilan sampel.

Pemilihan sampel kami mempertimbangkan beberapa faktor seperti jenis kelamin, kebiasaan makan yang teratur, kebiasaan merokok, program diet, konsumsi obat atau suplemen secara rutin, kebiasaan makan di atas jam 7 malam, kebiasaan mengonsumsi buah dan sayur, tingkat stres, tinggi badan, berat badan, jumlah jam tidur, frekuensi berolahraga, frekuensi makan dalam sehari, frekuensi membeli makan di luar dan memasak sendiri, serta frekuensi konsumsi susu.

Kami menggunakan metode sampling Quota Sampling, yang merupakan langkah sederhana namun efektif dalam penelitian tahap awal. Prosesnya dilakukan dengan memilih jumlah populasi yang telah ditentukan, lalu peneliti dapat memilih dua variabel penelitian yang ingin diteliti pada kelompok tertentu. Kami memilih metode ini karena pemilihan sampel dengan metode quota sampling meningkatkan efektivitas penelitian dan hasilnya dapat digeneralisasi untuk seluruh populasi. Oleh karena itu, kami memilih populasi untuk pengambilan sampel berdasarkan karakteristik dan sifat khusus anggota populasi.

3.2 Variabel Penelitian

Variabel yang digunakan pada penelitian terdiri dari 17 variabel respon dan mengambil variabel berat_badan dan tinggi_badan untuk dimasukkan ke rumus *Body Mass Index* atau BMI untuk mendapatkan 1 variabel target feature yaitu variabel kategori tubuh, rumus BMI sebagai berikut :

$$BMI = \frac{Berat\ Badan\ (kg)}{Tinggi\ Badan\ (m)^2}$$

Jika hasil dari perhitungan rumus BMI tersebut di antara rentang 17 - 25 maka kategori tubuh termasuk ideal sebaliknya jika diluar rentang tersebut termasuk tidak ideal. Variabel-variabel yang kami gunakan sebagai berikut :

1. jenis_kelamin, variabel yang merepresentasikan jenis kelamin dari responden & termasuk jenis data biner.
2. jam_makan_teratur, variabel yang merepresentasikan apakah responden memiliki jam makan yang teratur atau tidak & termasuk jenis data biner.
3. perokok, variabel yang merepresentasikan apakah responden merupakan perokok atau tidak & termasuk jenis data biner.
4. diet, variabel yang merepresentasikan apakah responden mengikuti program diet atau tidak & termasuk jenis data biner.
5. konsumsi_obat_rutin, variabel yang merepresentasikan apakah responden mengkonsumsi obat dengan rutin atau tidak & termasuk jenis data biner
6. makan_malam, variabel yang merepresentasikan apakah responden makan malam atau tidak & termasuk jenis data biner
7. frekuensi_buah_mingguan, variabel yang merepresentasikan jumlah frekuensi buah yang dikonsumsi responden dalam rentang waktu seminggu & termasuk jenis data ordinal.
8. frekuensi_sayur_mingguan, variabel yang merepresentasikan jumlah frekuensi sayur yang dikonsumsi responden dalam rentang waktu seminggu & termasuk jenis data ordinal.

9. tingkat_stress, variabel yang merepresentasikan tingkat stress dari responden & termasuk jenis data ordinal.
10. tinggi_badan, variabel yang merepresentasikan tinggi badan dari responden & termasuk jenis data numerik.
11. berat_badan, variabel yang merepresentasikan berat badan dari responden & termasuk jenis data numerik.
12. jumlah_makan_harian, variabel yang merepresentasikan jumlah makan responden dalam rentang waktu hari & termasuk jenis data numerik
13. jumlah_olahraga_mingguan, variabel yang merepresentasikan jumlah kegiatan olahraga responden dalam rentang waktu minggu & termasuk jenis data numerik.
14. jumlah_jam_tidur, variabel yang merepresentasikan jumlah tidur dari responden dalam waktu dalam satuan jam & termasuk jenis data numerik.
15. jum_makan_luar_mingguan, variabel yang merepresentasikan jumlah makan di luar dari responden dalam rentang waktu minggu & termasuk jenis data numerik
16. jum_masak_mingguan, variabel yang merepresentasikan jumlah memasak dari responden dalam rentang waktu minggu & termasuk jenis data numerik.
17. jumlah_gelas_susu, variabel yang merepresentasikan jumlah gelas susu yang dikonsumsi oleh responden dalam rentang waktu minggu & termasuk jenis data numerik.
18. kategori tubuh, variabel yang merepresentasikan.

3.3 Alur Pembangunan Model Klasifikasi Machine Learning

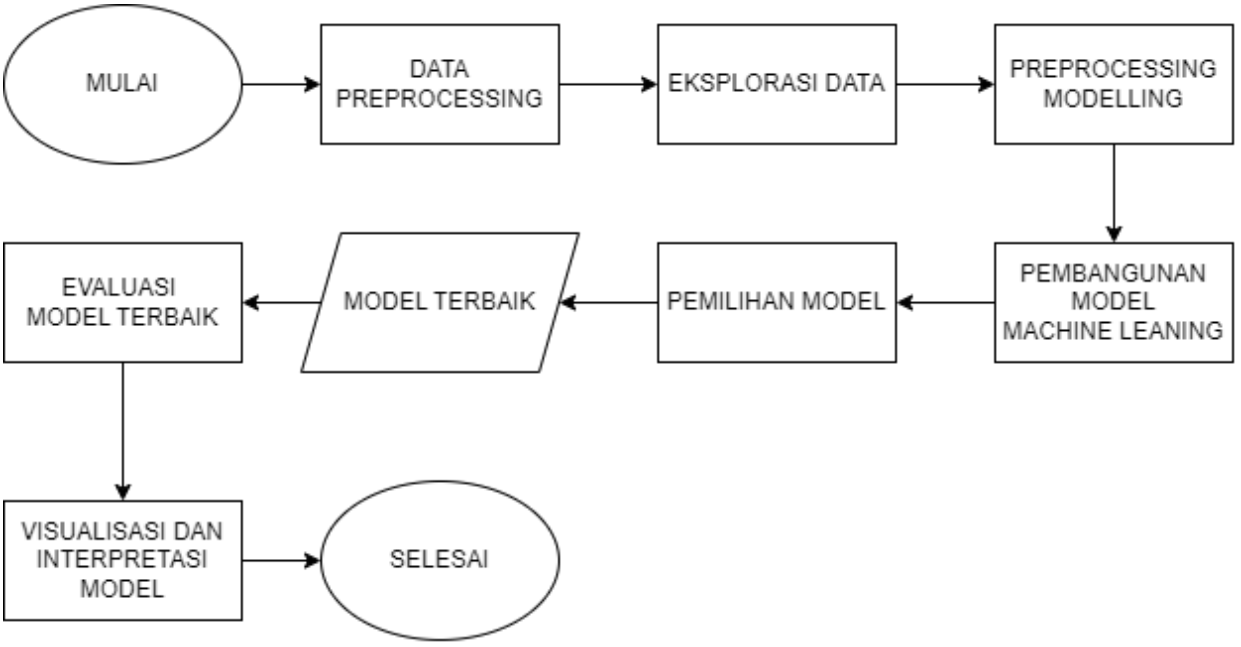

Figure 2. Flowchart Pembangunan Model Klasifikasi

BAB 4
HASIL DAN PEMBAHASAN

4.1 Data Preprocessing

4.1.1 Pengubahan Nama Features

Pertanyaan yang diajukan dalam survei kesehatan yang akan menjadi *features* dalam pembuatan model klasifikasi belum dapat diproses oleh bahasa pemrograman Python, oleh karena itu akan dilakukan pengubahan format setiap pertanyaan sesuai ketentuan Python. Berikut ini detail perubahan pertanyaan yang dilakukan menggunakan Google Spreadsheet:

1. Nama (Anda dapat merahasiakan identitas): nama
2. Jenis Kelamin: jenis_kelamin
3. Apakah jam makan Anda teratur?: jam_makan_teratur
4. Apakah Anda seorang perokok? (termasuk rokok elektrik): perokok
5. Apakah Anda sedang menjalani program diet?: diet
6. Apakah Anda sedang mengonsumsi obat atau suplemen secara rutin?: konsumsi_obat_rutin
7. Apakah Anda makan di atas jam 7 malam?: makan_malam
8. Seberapa sering Anda mengonsumsi buah dalam seminggu?: frekuensi_buah_mingguan
9. Seberapa sering Anda mengonsumsi sayur dalam seminggu?: frekuensi_sayur_mingguan
10. Menurut Anda, seberapa tingkat stres yang dialami?: tingkat_stress
11. Berapa tinggi badan Anda? (dalam cm): tinggi_badan
12. Berapa berat badan Anda? (dalam kg): berat_badan
13. Berapa kali Anda makan dalam sehari?: jumlah_makan_harian
14. Berapa kali Anda berolahraga dalam seminggu? (ex: 2, 3 atau 0 jika tidak pernah): jumlah_olahraga_mingguan
15. Berapa jumlah jam tidur Anda?: jumlah_jam_tidur
16. Berapa kali Anda membeli makan di luar dalam seminggu?: jum_makan_luar_mingguan
17. Berapa kali Anda memasak sendiri dalam seminggu (isi 0 jika tidak pernah): jum_masak_mingguan
18. Berapa gelas Anda mengonsumsi susu dalam seminggu? (isi 0 jika tidak mengonsumsi susu): jumlah_gelas_susu.

4.1.2 Pengecekan Missing-Values (NA)

Setelah dilakukan pengecekan terhadap missing-values, tidak ditemukan adanya nilai yang hilang sehingga tidak diperlukan *handling* atau penanganan lebih lanjut. Berikut *output* dari Python;

#	Column	Non-null Count	Dtype
0	no	61 non-null	int64
1	nama	61 non-null	object
.	.	.	.
.	.	.	.
18	jumlah_gelas_susu	61 non-null	int64
19	kategori_tubuh	61 non-null	object

Table 3. Output Python; Missing-Values

61 non-null menandakan tidak terdapat nilai null (NA) dari 61 observasi.

4.1.3 Features Engineering

Dalam tahapan ini, kami melakukan modifikasi terhadap *features* yang ada dalam *data frame* untuk kepentingan pembangunan model lebih lanjut. Beberapa hal yang dilakukan sebagai berikut;

1. Melakukan penghapusan terhadap *features* yang tidak diperlukan: *Feature* `no` dan `nama` akan dihapus dalam *data frame* karena keduanya hanya sebagai *identifier*,
2. Melakukan perubahan jenis data pada *non-numerical features* menjadi *numeric features* dengan *label encoder*: *Feature* `jenis_kelamin`, `jam_makan_teartur`, `perokok`,`diet`, `konsumsi_obat_rutin`, `makan_malam`, dan `kategori_tubuh`, dan
3. Membuat *data frame* baru yakni; `df_numerik` (berisi delapan fitur numerik),`df_kualitatif` (berisi sembilan fitur non-numerik), dan `df_target` (berisi satu fitur) untuk kepentingan pembersihan data lanjutan dan proses sebelum pembangunan model klasifikasi.

4.1.4 Pengecekan dan Handling Outliers (Pencilan)

Metode yang digunakan untuk pengecekan outliers adalah dengan menampilkan visualisasi boxplot dari df_numerik yang berisi hanya fitur numerik. Dari boxplot di bawah menunjukkan dari 8 fitur numerik terdapat dua fitur yang tidak mengandung outliers yakni tinggi_badan dan jumlah_jam_tidur.

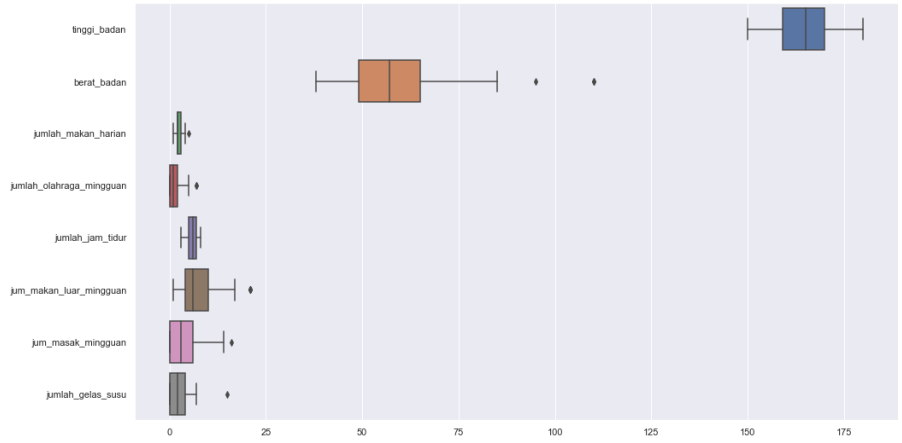

Figure 3. Boxplot - Deteksi Outliers

Setelah mengetahui fitur apa saja yang memiliki outlier, dilakukan handling outlier dengan nilai batas atas atau bawah (*whisker*) boxplot sesuai lokasi outlier. Berikut tampilan boxplot setelah dilakukan handling; (sudah tidak memuat outlier).

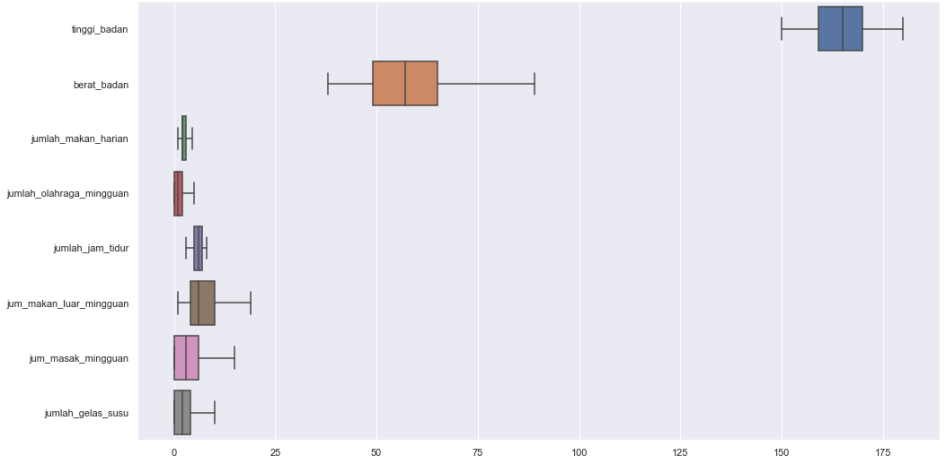

Figure 4. Boxplot - After Handling Outliers

4.2 Exploratory Data Analysis (EDA)

Pada bagian ini dilakukan proses analisis data untuk mendapatkan karakteristik pola dan hubungan dalam dataset secara deskriptif dan visual sebelum dilakukan analisis berikutnya. Adapun visualisasi data yang dilakukan sebagai berikut;

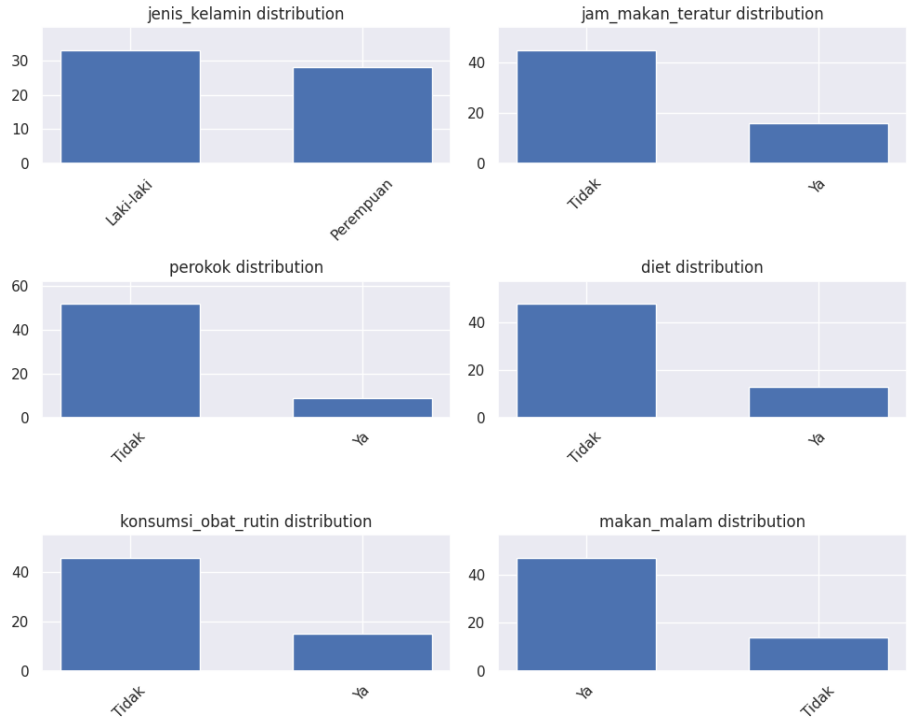

Figure 5. Barplot - Atribut Biner (Nominal)

Disini kami melakukan visualisasi distribusi menggunakan barplot terhadap variabel yang memiliki jenis data biner dalam dataset yang digunakan, dapat disimpulkan dari visualisasi diatas distribusinya tidak merata.

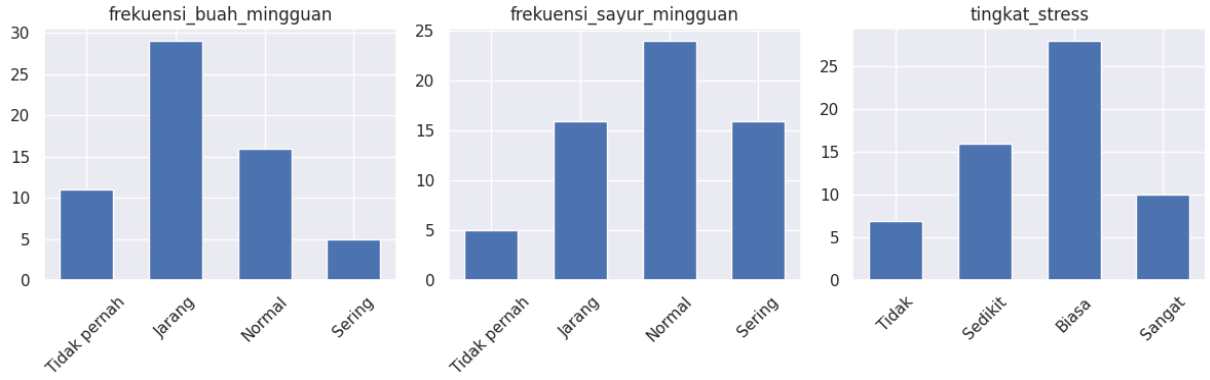

Figure 6. Barchart - Atribut Ordinal

Pada bagian ini dilakukan visualisasi menggunakan barchart terhadap variabel yang memiliki jenis data ordinal dalam dataset yang digunakan, dapat disimpulkan dari visualisasi diatas terdapat perbedaan signifikan antar variabel frekuensi_buah_mingguan dengan frekuensi_sayur_mingguan dan tingkat_stress.

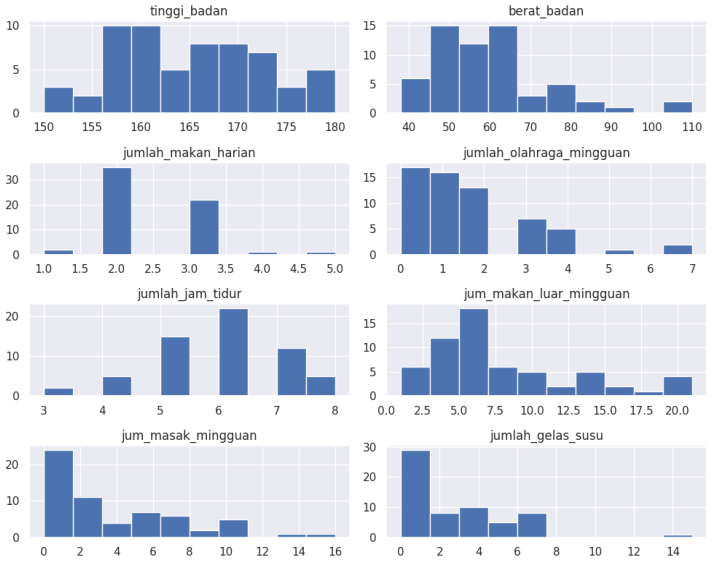

Figure 7. Histogram - Atribut Numerik

Disini kami melakukan visualisasi distribusi menggunakan histogram terhadap variabel yang memiliki jenis data numerik dalam dataset yang digunakan, dapat disimpulkan dari visualisasi di atas memiliki perbedaan distribusi yang signifikan antara variabelnya.

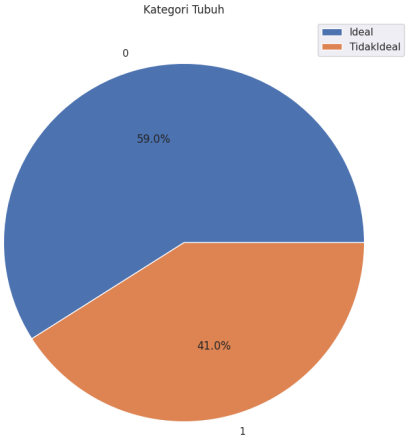

Figure 8. Pie Chart - Atribut Target

Disini kami melakukan visualisasi distribusi menggunakan pie chart terhadap target variabel yaitu kategori_tubuh, dapat disimpulkan dari visualisasi diatas lebih banyak responden yang memiliki kategori tubuh yang ideal (59%) dibanding dengan yang tidak ideal (41%).

4.3 Preprocessing Modelling

Sebelum melakukan pembangunan model *machine learning*, terlebih dahulu dilakukan *preprocessing modelling*. Berikut tahapan yang dilakukan;

4.3.1 Imbalanced Data Checking

Data yang tidak seimbang antara kelas target fitur negatif dan positif akan menghasilkan ukuran performa model yang buruk. Kasus data tidak seimbang terjadi ketika rasio antara kelas; minoritas dan mayoritas kurang dari atau sama dengan 0,1 (hanya terdapat 10% data kelas minoritas) dan terjadi kasus *imbalanced* yang parah ketika rasio 0,05 (terdapat 5% data kelas minoritas). Berdasarkan survei yang dilakukan, jumlah sampel kelas mayoritas sebanyak 36 orang sedangkan kelas minoritas 25 orang sehingga rasio yang dihasilkan yakni 0,694. Berdasarkan rasio tersebut, dapat disimpulkan bahwa data hasil survei yang akan digunakan untuk membangun model ML tidak mengalami masalah keseimbangan kelas (*imbalanced data*).

4.3.2 Data Normalization

Beberapa fitur numerik dalam data hasil survei memiliki nilai yang beragam, sehingga diperlukan perubahan skala data dalam bentuk *z-score* atau yang kerap dinamakan normalisasi. Skala data numerik setelah dilakukan normalisasi berada pada rentang 0 sampai 1, alasan dilakukan normalisasi yakni menyamakan skala data dengan fitur non-numerik yang kebanyakan bernilai 0 atau 1.

4.3.3 Feature Selection

Dalam penentuan fitur mana saja yang akan dipakai dalam membangun model ML, digunakan nilai korelasi pearson. Berikut nilai korelasi pearson tiap fitur terhadap target fitur *kategori_tubuh*;

Biner		***Ordinal***		jumlah_makan_harian	-0,13
jenis_kelamin	-0,1	*frekuensi_buah_mingguan*	*-0,085*	jumlah_olahraga_mingguan	0,18
jam_makan_teratur	*-0,042*	frekuensi_sayur_mingguan	0,11	jumlah_jam_tidur	-0,1
perokok	0,12	*tingkat_stress*	*0,083*	jum_makan_luar_mingguan	0,16
diet	*0,055*	***Numerik***		jum_masak_mingguan	-0,22
konsumsi_obat_rutin	0,14	tinggi_badan	-0,15	*jum_gelas_susu*	*-0,081*
makan_malam	-0,18	*berat_badan*	*0,067*		

Table 4. Korelasi Pearson seluruh Feature terhadap Kategori Tubuh

fitur yang memiliki nilai korelasi pearson <0,1 tidak akan diikutsertakan dalam pembangunan model ML, terdapat enam fitur yang dihapus dan ditandai dengan warna coklat.

4.3.4 Data Splitting

Setelah melakukan seleksi fitur, selanjutnya dilakukan pembagian data menjadi dua bagian yaitu untuk pelatihan model atau *data training* dan pengetesan model atau *data testing*. Ukuran pembagian yakni 20% *data testing* (13 objek) dan 80% *data training* (48 objek). Adapun parameter tambahan yang digunakan yakni *random state* 42.

4.3.5 Leave-One-Out Cross-Validation

Sampel yang digunakan dalam pembangunan model klasifikasi tergolong rendah (48 objek) sehingga diputuskan untuk menggunakan cross-validation untuk menilai ukuran performa model secara lebih baik. Jenis cross-validation yang cocok digunakan untuk data yang terbatas yakni *leave-one-out* karena algoritma dari cross-validation ini menjadikan setiap observasi atau objek sebagai *data testing*. Dari *data training* sejumlah 48 objek maka akan dibangun 48 model, kemudian performa dari tiap model akan dirata-ratakan dan perlu diketahui bahwa pada setiap model akan menghasilkan ukuran performa 0 atau 1 (*testing* data hanya 1).

4.4 Model Building

Library yang digunakan untuk pembangunan model klasifikasi *machine learning* yakni dengan `sklearn`.

4.4.1 Logistic Regression

- Hyperparameter Tuning

Parameter yang akan di-tuning meliputi '*penalty*' dan '*C'*. Setelah proses tuning didapatkan hyperparameter terbaik yang ditemukan adalah {'*C'* = 0,01 ; '*penalty'* = 12}

- Model Fitting

```
clf_lr = LogisticRegression(C=0.01,penalty='l2',class_weight='balanced',random_state=42)
clf_lr.fit(X_train,y_train)
```

Figure 9. Model Fitting - Logistic Regression

Logistic Regression hyperparameter yang telah dibuat dengan hyperparameter {C=0.01, penalty='l2', class_weight='balanced', random_state=42}, selanjutnya dilakukan model *fitting* dengan parameter dari informasi *tuning* menggunakan data train.

- Cross-Validation

```
Cross Validation Scores:  [0. 1. 1. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 1.
 0. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 1. 1. 0. 1. 1. 0. 0. 1. 0. 0. 0.]
Average CV Score:  0.5625
Number of CV Scores used in Average:  48
```

Figure 10. Rata-rata akurasi LOOCV - Logistic Regression

Dari hasil evaluasi performa model Logistic Regression dengan hyperparameter yang ditentukan menggunakan metode cross-validation. Setiap skor akurasi pada setiap iterasi cross-validation dicetak (Cross Validation Scores), skor akurasi rata-rata dicetak (Average CV Score), dan jumlah skor akurasi yang digunakan dalam menghitung rata-rata dicetak (Number of CV Scores used in Average). Dan skor akurasi rata-rata yang dicapai adalah 0.5625

4.4.2 Decision Tree Classifier

- Hyperparameter Tuning

Parameter yang akan di-tuning meliputi *'criterion'*,*'max_depth'*,*'max_leaf_nodes'* dan *'min_samples_split'*. Setelah proses tuning hyperparameter terbaik yang ditemukan adalah {'*criterion*' = '*gini*' ; '*max_depth*' = 6 ; '*max_leaf_nodes*' = 10 ; '*min_samples_split*' = 2}

- Model Fitting

```
clf_dt = DecisionTreeClassifier(random_state=42,criterion='gini',max_depth=6,max_leaf_nodes=10,
min_samples_split=2)
clf_dt.fit(X_train,y_train)
```

Figure 11. Model Fitting - Decision Tree

Decision Tree Classifier yang telah dibuat dengan hyperparameter {max_depth = 6 ; max_leaf_nodes = 10 ; random_state = 42}. selanjutnya dilakukan model *fitting* dengan parameter dari informasi *tuning* menggunakan data train.

- Cross-Validation

```
Cross Validation Accuracy:  [1. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 1. 1.
 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1.]
Average CV Accuracy:  0.7708333333333334
Number of CV Accuracy used in Average:  48
```

Figure 12. Rata-rata akurasi LOOCV - Decision Tree

Dari hasil evaluasi performa model Decision Tree Classifier dengan hyperparameter yang ditentukan menggunakan metode cross-validation. Setiap skor akurasi pada setiap iterasi cross-validation dicetak (Cross Validation Scores), skor akurasi rata-rata dicetak (Average CV Score), dan jumlah skor akurasi yang digunakan dalam menghitung rata-rata dicetak (Number of CV Scores used in Average). Dan skor akurasi rata-rata yang dicapai adalah 0.7708333333333334.

4.4.3 Naive Bayes

- Hyperparameter Tuning

parameter tuning yang dilakukan adalah untuk memperoleh nilai optimal dari parameter *var_smoothing* pada model Gaussian Naive Bayes. Parameter *var_smoothing* mengontrol smoothing atau penghalusan yang diterapkan pada probabilitas kelas dan fitur dalam model. Output tersebut menunjukkan bahwa hyperparameter terbaik yang ditemukan adalah *var_smoothing* dengan nilai 7.891847103414872e-06, dan skor akurasi terbaik yang dicapai oleh model dengan hyperparameter tersebut adalah 0.6666666666666666.

- Model Fitting

Setelah menemukan hyperparameter dari model Naive Bayes, selanjutnya melakukan fitting model dengan parameter dari informasi tuning menggunakan data training sebagai berikut;

```
clf_nb = GaussianNB(var_smoothing= 4.680993769091634e-06)
clf_nb.fit(X_train,y_train)
```

Figure 13. Model Fitting - Naive Bayes

- Cross-Validation

```
Cross Validation Scores:  [1. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 1. 0. 1.]
Average CV Score:  0.6666666666666666
Number of CV Scores used in Average:  48
```

Figure 14. Rata-rata akurasi LOOCV - Naive Bayes

Dari hasil leave-one-out cross-validation didapatkan rata-rata akurasi model Naive Bayes sebesar 0,67.

4.4.4 Random Forest

- Hyperparameter Tuning

Parameter yang akan di-*tuning* meliputi n_estimators, max_features, max_depth, min_samples_leaf, min_samples_split. Setelah proses tuning, dihasilkan urutan dari parameter berturut-turut 100, auto, 10, 1, dan 2.

- Model Fitting

Setelah menemukan hyperparameter dari model Random Forest, selanjutnya melakukan *fitting* model dengan parameter dari informasi *tuning* menggunakan *data training* sebagai berikut;

```
clf_rf = RandomForestClassifier(random_state=42,n_estimators=100, min_samples_split=2, min_samples_leaf=1,
max_features= 'auto', max_depth=10)
clf_rf.fit(X_train,y_train)
```

Figure 15. Model Fitting - Random Forest

- Cross-Validation

```
Cross Validation Accuracy: [1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1.
1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0.]
Average CV Accuracy: 0.6666666666666666
Number of CV Accuracy used in Average: 48
```

Figure 16. Rata-rata akurasi LOOCV - Random Forest

Dari hasil leave-one-out cross-validation didapatkan rata-rata akurasi model Random Forest sebesar 0,67.

4.3.5 K-Nearest Neighbors (KNN)

- Hyperparameter Tuning

Parameter yang akan di-*tuning* adalah "n_neighbors", "weights", dan "p" untuk model KNN. Dicoba beberapa nilai untuk masing-masing parameter dan evaluasi dilakukan menggunakan metrik akurasi dengan metode validasi silang Leave-One-Out (LOO). Hasil terbaik berdasarkan penyetelan hiperparameter diperoleh melalui atribut "best_score_" dan kombinasi terbaik ditampilkan melalui atribut "best_params_".

- Model Fitting

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=13, p=1)
```

Figure 17. Model Fitting - KNN

Dari output di atas, dapat disimpulkan bahwa model akan menggunakan 13 tetangga terdekat dalam proses klasifikasi dan menggunakan jarak (p=1) untuk mengukur kedekatan antara data.

- Cross-Validation

```
Cross Validation Scores:  [1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 1. 0. 1. 1. 0. 1. 0. 0. 1. 1.
1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 0.]
Average CV Score:  0.6875
Number of CV Scores used in Average:  48
```

Figure 18. Rata-rata akurasi LOOCV - KNN

Dari hasil leave-one-out cross-validation didapatkan rata-rata akurasi model Random Forest sebesar 0,6875 atau 68.75% yang termasuk baik dengan jumlah skor validasi yang digunakan dalam menghitung rata-rata sebesar 48 .

4.4.6 Artificial Neural Network (ANN)

- Hyperparameter Tuning

Dilakukan penggunaan MLPClassifier dari modul sklearn.neural_network untuk melakukan klasifikasi menggunakan jaringan syaraf tiruan (Artificial Neural Network/ANN). Model MLPClassifier diinisialisasi dengan hyperparameter "random_state" yang diatur ke 42 untuk

memastikan hasil yang konsisten, dan "solver" yang diatur ke 'adam', yang merupakan metode optimisasi yang umum digunakan dalam pelatihan jaringan saraf. Kemudian, model dilatih menggunakan data pelatihan (X_train dan y_train).

- Model Fitting

```
MLPClassifier
MLPClassifier(random_state=42)
```

Figure 19. Model Fitting - ANN

Model MLPClassifier diinisialisasi dengan hyperparameter "random_state" yang diatur ke 42 untuk memastikan hasil yang konsisten, dan "solver" yang diatur ke 'adam', yang merupakan metode optimisasi yang umum digunakan dalam pelatihan jaringan saraf. Kemudian, model dilatih menggunakan data train (X_train dan y_train).

- Cross-Validation

```
Cross Validation Scores:  [1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 0. 0. 1. 1.
1. 0. 1. 1. 0. 0. 1. 1. 0. 1. 0. 0. 0.]
Average CV Score:  0.6041666666666666
Number of CV Scores used in Average:  48
```

Figure 20. Rata-rata akurasi LOOCV - ANN

Rata-rata skor validasi silang yang diperoleh sekitar 0.604, menunjukkan akurasi sekitar 60.4% dalam memprediksi klasifikasi dengan data cross-validation dan didapatkan skor validasi silang yang digunakan dalam perhitungan rata-rata sebesar 48.

4.4.7 Support Vector Machines (SVM)

- Hyperparameter Tuning

Output menunjukkan skor akurasi terbaik yang ditemukan adalah 0.7083333333333334, dan kombinasi hyperparameter terbaik yang menghasilkan skor akurasi tersebut adalah {'C': 1000, 'gamma': 0.7, 'kernel': 'rbf'}.

- Model Fitting

Setelah menemukan hyperparameter dari model Naive Bayes, selanjutnya melakukan fitting model dengan parameter dari informasi tuning menggunakan data training sebagai berikut;

```
clf_svm = SVC(C=1000, gamma=0.7, kernel= 'rbf')
clf_svm.fit(X_train, y_train)
```

Figure 21. Model Fitting - SVM

- Cross-Validation

```
Cross Validation Scores:  [1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0.
 1. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0.]
Average CV Score:  0.7083333333333334
Number of CV Scores used in Average:  48
```

Figure 22. Rata-rata akurasi LOOCV -SVM

Dari hasil leave-one-out cross-validation didapatkan rata-rata akurasi model Support Vector Machine sebesar 0,71.

4.5 Model Selection

Dalam menentukan model mana yang akan dipilih sebagai model terbaik, digunakan ukuran performa model akurasi dari leave-one-out cross-validation. Berikut ini rangkuman akurasi kebaikan model;

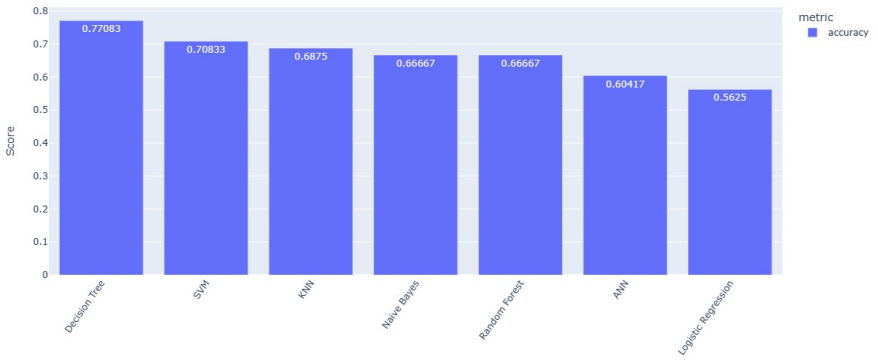

Figure 23. Boxplot - Rata-rata akurasi LOOCV All Models

Dari barplot terlihat bahwa model Decision Tree-lah yang memiliki akurasi tertinggi (0,77083) dibandingkan dengan enam model klasifikasi lainnya, sehingga diputuskan untuk memilih Decision Tree sebagai model terbaik.

4.6 Model Terbaik

Setelah mengetahui bahwa data hasil survei kesehatan untuk pembangunan model klasifikasi kategori tubuh mahasiswa TSD ‘21 paling cocok dimodelkan dengan Decision Tree, berikut akan dilakukan penjelasan lebih mendalam terkait model terbaik.

4.6.1 Learning Curves

Untuk mengecek apakah model yang telah dipilih yaitu Decision tree terjadi kasus *overfitting* ataupun *underfitting*, maka akan dilihat menggunakan Learning-Curve;

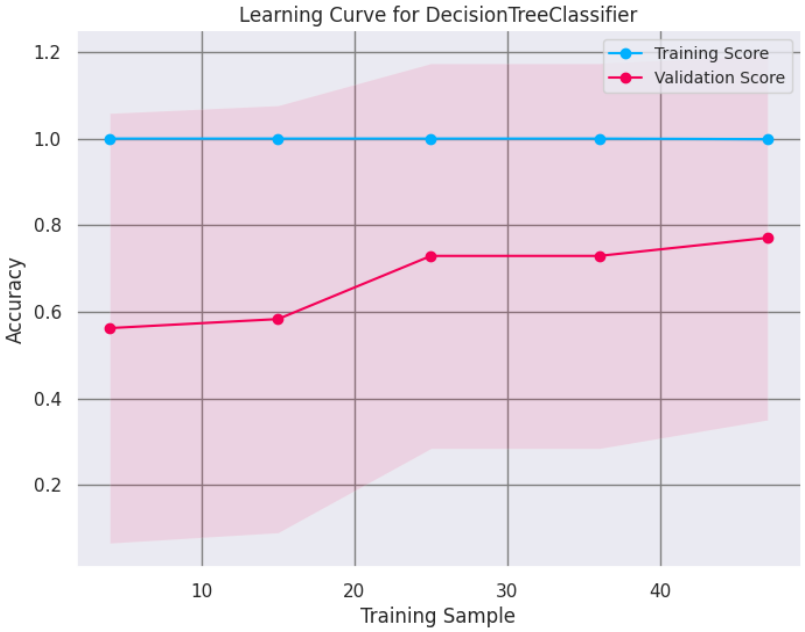

Figure 24. Learning Curves - Decision Tree

Dilihat dari visualisasi Learning-Curve bahwa jarak dari kedua kurva tidak terlalu jauh dan juga berada di akurasi yang tinggi, yang berarti model pada Decision Tree tidak terjadi *overfitting* ataupun *underfitting*.

4.6.2 Model Evaluation (*Performance Measures*)

Untuk mengevaluasi atau melihat performa model pada data baru (data test), maka akan dilihat menggunakan Confusion-Matrix;

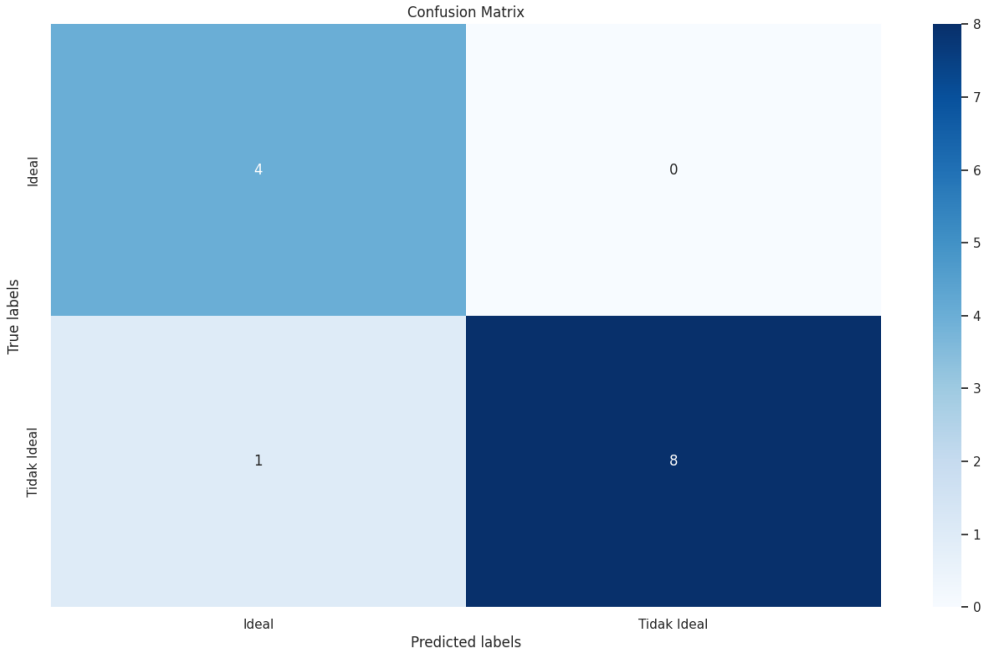

Figure 25. Confusion Matrix from Data Testing

Elemen *Confusion Matrix*	**Jumlah Data**
True Positive (Prediksi Tidak Ideal Benar)	8
False Positive (Prediksi Tidak Ideal Salah)	0
True Negative (Prediksi Ideal Benar)	4
False Negative (Prediksi Ideal Salah)	1

Table 5. Elemen Confusion Matrix - Decision Tree

Performance Measure	***Value***
Akurasi	0.9230769230769231
Presisi (*weighted*)	0.9384615384615385
Recall (*weighted*)	0.9230769230769231
F1-Score (*weighted*)	0.9250879839115134

Table 6. Performance Measures - Decision Tree

Berdasarkan ukuran performa model untuk *data testing*, didapatkan seluruh ukuran performa lebih dari 90% artinya model decision tree telah dapat dianggap sebagai model yang mumpuni.

4.6.3 Model Evaluation (AUC [*Area Under ROC Curve*])

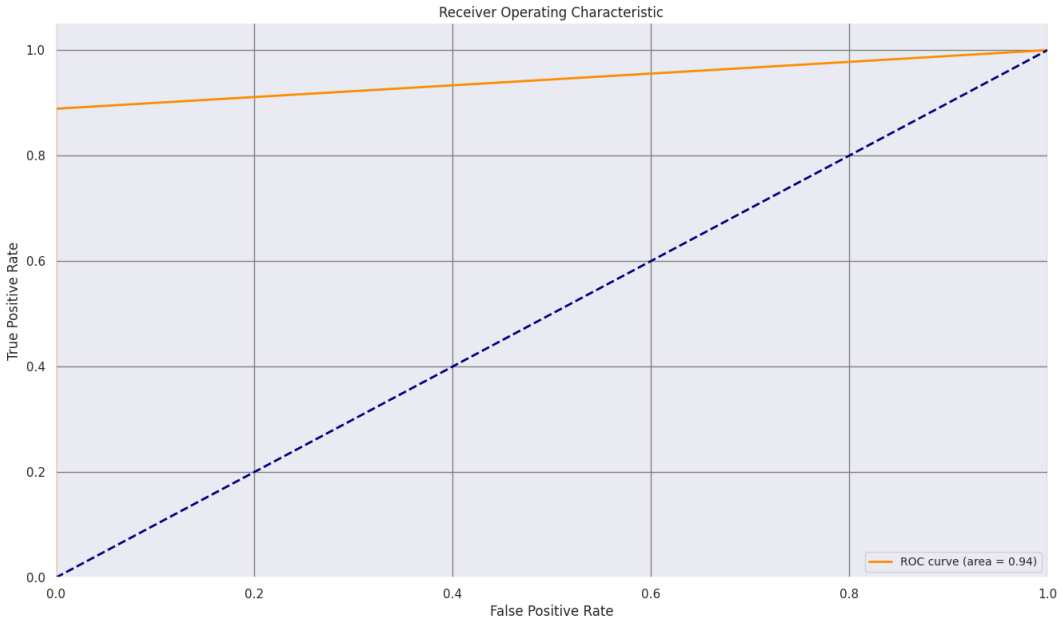

Figure 26. ROC (Receiver Operating Characteristic) Curve - Decision Tree

Dari kurva ROC di atas, terlihat bahwa didapatkan nilai luas di bawah kurva atau AUC sebesar 0,94 artinya model klasifikasi decision tree dianggap sebagai model yang baik (mendekati *classifier* sempurna 1).

4.6.4 Model Visualization

Figure 27. Visualisasi Decision Tree - Sklearn (Graphviz)

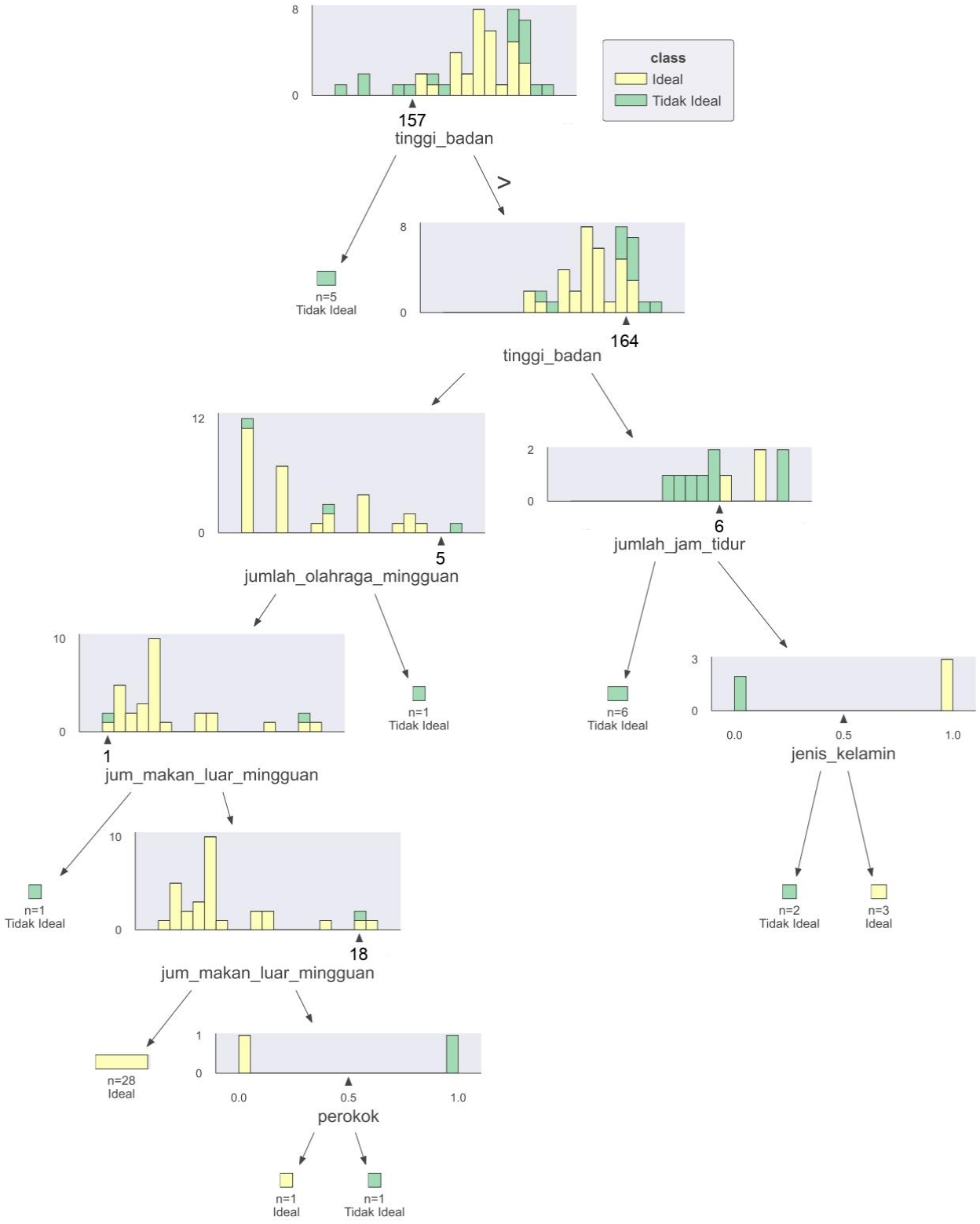

Figure 28. Visualisasi Decision Tree - DtreeViz

4.6.5 Model Interpretation

Visualisasi model Decision-Tree yang telah dibangun adalah seperti gambar di atas. Model Decision-Tree dapat dijelaskan dengan mudah karena memiliki struktur yang intuitif dan mudah dipahami. Sebagai contoh, terdapat data yang memiliki tinggi badan kurang dari 157 maka model akan langsung mengklasifikasikan bahwa orang tersebut tidak ideal, kemudian jika tinggi badannya tidak kurang dari 157 maka akan diseleksi lagi berdasarkan tinggi badan apakah kurang dari 164 atau tidak, jika True maka akan diseleksi dari feature jumlah olahraga mingguan apakah jumlah olahraga mingguannya kurang dari 5 atau tidak, jika False maka model akan mengklasifikasikan bahwa orang tersebut Tidak Ideal, jika True maka akan lanjut ke node selanjutnya dan diseleksi pada feature jumlah makan luar per minggu apakah kurang dari 1 atau tidak, jika True maka model akan mengklasifikasikan orang tersebut Ideal, dan jika False maka akan lanjut ke node selanjutnya yang dimana pada node tersebut menyeleksi berdasarkan jumlah makan luar mingguan juga apakah kurang dari 18 atau tidak, jika True maka orang tersebut diklasifikasikan Ideal, jika False maka akan diseleksi lagi dengan melihat feature merokok apakah orang tersebut merokok atau tidak, jika True maka akan diklasifikasikan dengan Ideal, jika False maka akan diklasifikasikan Tidak Ideal. Oleh karena node tinggi badan yang kurang dari 164 bernilai True sudah mencapai *leaf node* maka selanjutnya akan dijelaskan bagaimana model melakukan klasifikasi node tinggi badan yang kurang dari 164 bernilai False. Jika False, maka akan masuk ke dalam node jumlah jam tidur apakah kurang dari 6 atau tidak, jika True maka model akan mengklasifikasikan Tidak Ideal. Jika False maka akan diseleksi lagi atau masuk kedalam node jenis kelamin, apakah laki-laki atau tidak, jika True maka akan diklasifikasikan Tidak Ideal, jika False (Perempuan) maka model akan mengklasifikasikan Ideal.

BAB 5
KESIMPULAN

Dalam proyek ini, dilakukan pembangunan model *machine learning* untuk mengklasifikasikan kategori tubuh mahasiswa Teknologi Sains Data angkatan 2021 yakni tergolong ideal atau tidak ideal berdasarkan kebiasaan dan aktivitas mahasiswa TSD'21. Berdasarkan jumlah *target feature* kategori tubuh mahasiswa TSD'21 yang didapatkan dari hasil perhitungan BMI (*body mass index*) tidak mengalami kasus *imbalanced data* (*data tidak seimbang*). Adapun *feature* yang didapatkan dari hasil survei sebanyak 18 (termasuk *target feature*), kemudian akan dipilih 11 fitur prediktif berdasarkan kekuatan hubungan dengan *target feature* (kategori tubuh) yang memiliki koefisien korelasi pearson > 0,1. Dilakukan pembagian 20% *data testing* (13 observasi) dan 80% *data training* (48 observasi) serta normalisasi untuk fitur numerik.

Setelah melakukan pembagian data, *data training* dilatih ke dalam beberapa model *machine learning* di antaranya regresi logistik, decision tree, naive bayes, random forest, k-nearest neighbors (KNN), artificial neural network (ANN), dan support vector machines (SVM) di mana tujuh model tersebut sudah memiliki parameter yang optimal.

Pemilihan model terbaik ditentukan berdasarkan nilai akurasi dengan metode *leave-one-out cross-validation* (LOOCV) yang tertinggi. Dari ketujuh model klasifikasi, didapatkan akurasi tertinggi pada model klasifikasi decision tree yakni sebesar 0,77083. Setelah memilih decision tree sebagai model terbaik, dilakukan pengecekan berdasarkan *learning curve* dan didapatkan tidak terdapat kasus *underfitting* atau *overfitting* pada model (*good-fit*). Kemudian untuk mengetahui seberapa baik model bekerja pada data baru, maka akan dievaluasi menggunakan *data testing*. Dari hasil evaluasi, model decision tree dapat memprediksi; kategori tubuh ideal dengan benar (*true negative*) sebanyak 4 orang, kategori tubuh tidak ideal dengan benar (*true positive*) sebanyak 8 orang. Model melakukan kesalahan prediksi tubuh ideal (*false negative*) sebanyak 1 orang sehingga pada kesalahan prediksi tubuh tidak ideal (*false positive*) sebanyak 0 orang. Dari elemen *confusion matrix* tersebut, didapatkan akurasi sebesar 0.9230769230769231, presisi 0.9384615384615385, recall 0.9230769230769231, dan f1-score 0.9250879839115134. Dari kurva ROC juga didapatkan nilai AUC sebesar 0,94. Dilihat dari *performance measures*, dapat disimpulkan bahwa model decision tree dapat bekerja dengan baik untuk mengklasifikasikan kategori tubuh mahasiswa TSD'21.

SARAN

Mengingat terbatasnya waktu dan juga biaya untuk melaksanakan *project* ini, perlu dilakukan survei yang lebih luas, tidak hanya terbatas pada mahasiswa Teknologi Sains Data angkatan 2021 Universitas Airlangga dan juga perlu adanya eksplorasi model yang lebih kompleks sehingga mendapatkan akurasi yang lebih baik serta dapat menambahkan fitur-fitur baru yang sekiranya memiliki hubungan dengan target fitur.

DAFTAR PUSTAKA

Kemkes.go.id, (2018). *Kementerian Kesehatan Republik Indonesia*. [online] Available at: https://www.kemkes.go.id/article/print/18012900004/bersama-selesaikan-masalah-kesehatan.html#:~:text=Bloom%20menyatakan%20bahwa%20ada%204 [Accessed 9 Jun. 2023].

Kemkes.go.id, (2022), *Direktorat Jenderal Pelayanan Kesehatan* [online] Available at; https://yankes.kemkes.go.id/view_artikel/119/kesehatan-dan-makna-sehat. [Accessed 7 June 2023]

Bps.go.id, (2020), *Badan Pusat Statistik* , <https://www.bps.go.id/indicator/30/222/1/persentase-penduduk-yang-mempunyai-keluhan-kesehatan-selama-sebulan-terakhir.html>. [Accessed 7 June 2023]

Ibm.com , (2023), *What is a Decision Tree| IBM , <https://www.ibm.com/topics/decision-trees>,* [Accessed 10 June 2023]

Ratnawati, R, (2008), *'Kajian Principal Component Logistic Regression (PCLR) Dan Principal Component Logistic Regression Stepwise (PCLR(S)) Dalam Pemilihan Regresi Logistik Terbaik Pada Kasus MultikolinieritasBrawijayaKnowledgeGarden*',[online] http://repository.ub.ac.id/id/eprint/151820/. [Accessed 10 June 2023]

World Health Organization 2022, who.int, viewed 9 June 2023, <https://www.who.int/standards/classifications/classification-of-diseases> .

DQ Lab AI-Powered Learning 2022, dqlab.id, viewed 9 June 2023, <https://dqlab.id/mengenal-naive-bayes-sebagai-salah-satu-algoritma-data-science> .

Binus University 2022, sis.binus.ac.id, viewed 9 June 2023, <https://sis.binus.ac.id/2022/02/14/support-vector-machine-algorithm/> .

Xiang-wei, L. and Qi Yian-fang (2012). A Data Preprocessing Algorithm for Classification Model Based On Rough Sets. *Physics Procedia*, [online] 25, pp.2025–2029. doi:https://doi.org/10.1016/j.phpro.2012.03.345.

Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. First ed. 1005 Gravenstein Highway North, Sebastopol, CA 95472.: O'Reilly Media, Inc., 1005.

Burkov, A. (2019). *The Hundred Page Machine Learning Book*. Quebec City, Canada: Andriy Burkov.

Alhamid, M. (2020). *What is Cross-Validation? - Towards Data Science*. [online] Medium. Available at: https://towardsdatascience.com/what-is-cross-validation-60c01f9d9e75 [Accessed 9 Jun. 2023].

Roihan, A., Abas Sunarya, P. and Rafika, A. (2019). IJCIT (Indonesian Journal on Computer and Information Technology) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, [online] 5(1), pp.75–82. Available at:

http://download.garuda.kemdikbud.go.id/article.php?article=1617577&val=10500&title=Pemanfaatan%20Machine%20Learning%20dalam%20Berbagai%20Bidang%20Review%20paper.

Cholil, S., Handayani, T., Prathivi, R. and Ardianita, T. (2020). IJCIT (Indonesian Journal on Computer and Information Technology) Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa. *IJCIT (Indonesian Journal on Computer and Information Technology)*, [online] 6(2). Available at: https://ejournal.bsi.ac.id/ejurnal/index.php/ijcit/article/viewFile/10438/pdf.

Putra, H. and Nabilah Ulfa Walmi (2020). Penerapan Prediksi Produksi Padi Menggunakan Artificial Neural Network Algoritma Backpropagation. *Jurnal Nasional Teknologi dan Sistem Informasi*, [online] 6(2), pp.100–107. Available at: https://teknosi.fti.unand.ac.id/index.php/teknosi/article/view/1642/pdf [Accessed 9 Jun. 2023].

Curry, R. (2021). *The Complete Guide to Random Forests: Part 1 - Rowan Curry - Medium*. [online] Medium. Available at: https://medium.com/@curryrowan/the-complete-guide-to-random-forests-part-1-427c9cba8336 [Accessed 10 Jun. 2023].

LAMPIRAN

Google Collaboratory;

https://colab.research.google.com/drive/1YYCnsgaI1DmBaDVqr5ON7_GNcCj-MDhH?usp=sharing

Dataset;

https://docs.google.com/spreadsheets/d/1LMWZTAoB227DhU6nvU2LqsxnsQW6uGslmPLxkE37940/edit?usp=sharing

Pertanyaan survei melalui Google Form;

Nama (*Anda dapat merahasiakan identitas*) *

Teks jawaban singkat

Jenis Kelamin *

- Laki-laki
- Perempuan

Apakah jam makan Anda teratur? *

- Ya
- Tidak

Apakah Anda seorang perokok? (termasuk rokok elektrik) *

- Ya
- Tidak

Apakah Anda sedang menjalani program *diet*? *

- Ya
- Tidak

Apakah Anda sedang mengonsumsi obat atau suplemen secara rutin? *

- Ya
- Tidak

Apakah Anda makan di atas jam 7 malam? *

- Ya
- Tidak

Seberapa sering Anda mengonsumsi buah dalam seminggu? *

	1	2	3	4	
Tidak pernah	○	○	○	○	Sering

Seberapa sering Anda mengonsumsi sayur dalam seminggu?

	1	2	3	4	
Tidak pernah	○	○	○	○	Sering

Menurut Anda, seberapa tingkat stres yang dialami? *

	1	2	3	4	
Tidak stres	○	○	○	○	Sangat stres

Berapa tinggi badan Anda? (dalam cm) *

Teks jawaban singkat

Berapa berat badan Anda? (dalam kg) *

Teks jawaban singkat

Berapa kali Anda makan dalam sehari? *

Teks jawaban singkat

Berapa kali Anda berolahraga dalam seminggu? (ex: 2, 3 atau 0 jika tidak pernah) *

Teks jawaban singkat

Power Point;

PPT - KELOMPOK 3 - Presentation (canva.com)