**Laporan Case-Based 1**

**Mata Kuliah Pembelajaran Mesin**

**Artificial Neural Network (ANN)**

**Muhamad Azmi Rizkifar – 1301218586 – IFX-45-GAB – IKN**

“Saya mengerjakan tugas ini dengan cara yang tidak melanggar aturan perkuliahan dan kode etik akademisi”

****

**Program Studi Sarjana Informatika**

**Fakultas Informatika**

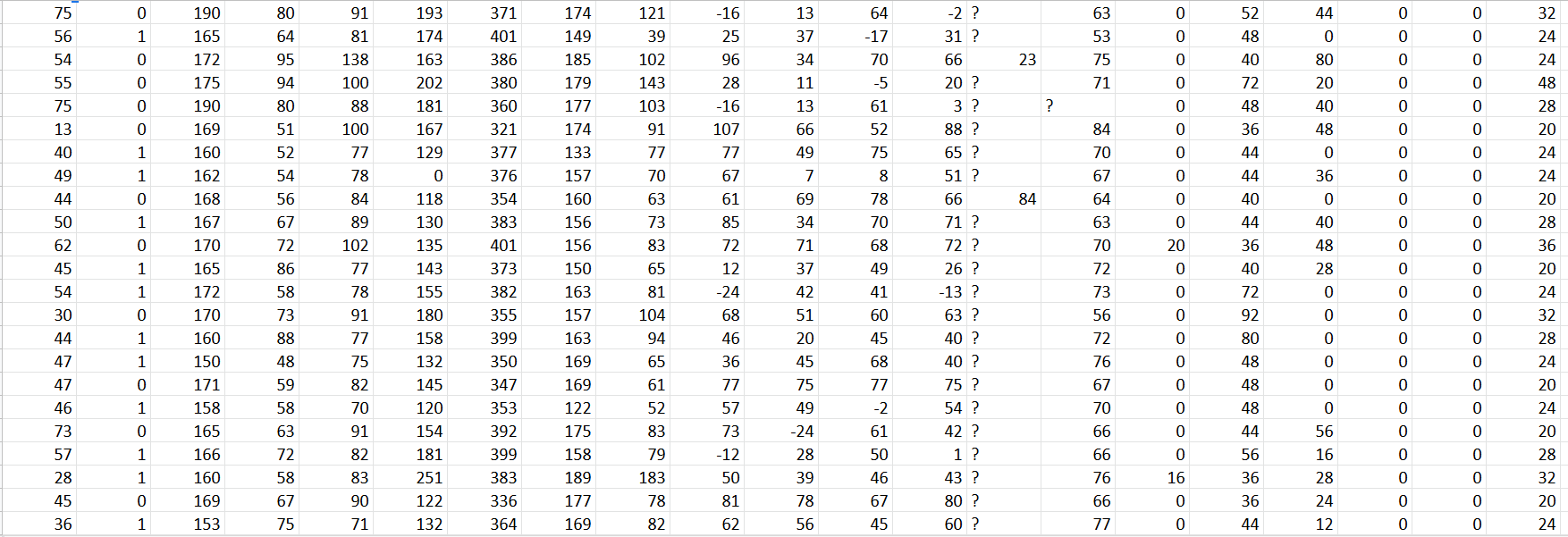
**Universitas Telkom**

**Bandung**

**2022**

# **BAGIAN I Ikhtisar Data Yang Dipilih**

Dataset yang akan dipilih dalam pengimplementasian *ANN* adalah dataset *Arrhythmia* yang didapatkan dari *UCI Machine Learning Repository*. Tujuan penulis adalah untuk membangun sebuah model klasifikasi yang dapat memprediksi antara ada dan tidak adanya aritmia jantung dan mengklasifikasikannya dalam salah satu dari 16 kelompok data. Data yang didapat disajikan dalam bentuk tabulasi yang telah di konversi ke file Excel (xlsx).

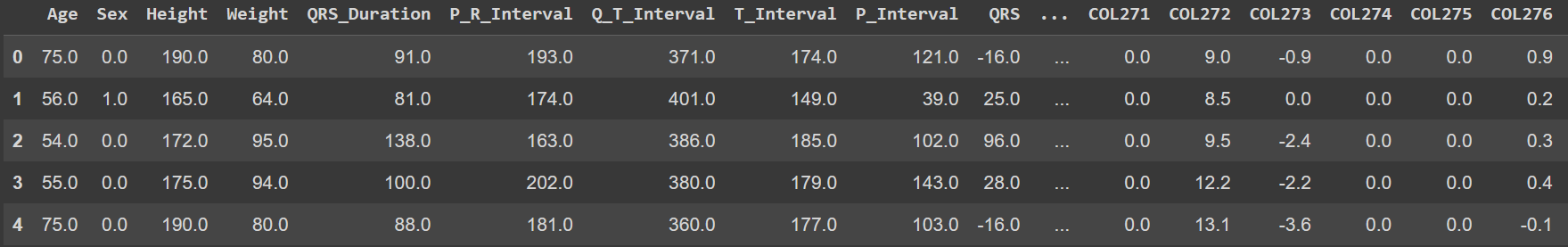
****

Gambar 1 Dataset Arrhythmia Data

Dari keseluruhan data yang ada, terdapat total 452 data pasien yang dilengkapi dengan 279 kolom atau nilai fitur dan 1 kolom target yaitu kategori aritmia jantung. Pada kolom target, Terdapat 16 jenis kelompok data aritmia jantung yang menunjukkan apakah seorang pasien termasuk ke dalam salah satu dari beberapa jenis kelompok tersebut. Data jenis kelompok tersebut berisi :

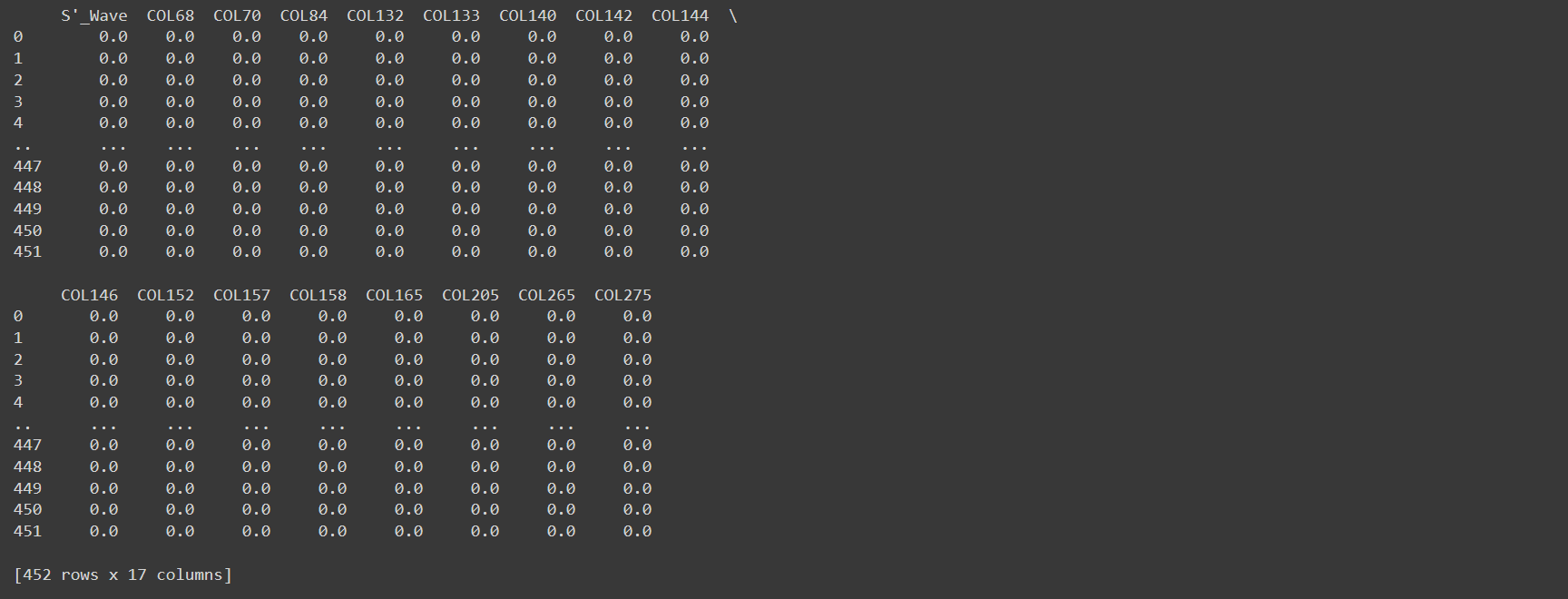
1. Normal
2. Ischemic changes (Coronary Artery Disease)
3. Old Anterior Myocardial Infarction
4. Old Inferior Myocardial Infarction
5. Sinus tachycardy
6. Sinus bradycardy
7. Ventricular Premature Contraction (PVC)
8. Supraventricular Premature Contraction
9. Left bundle branch block
10. Right bundle branch block
11. 1 degree AtrioVentricular block
12. 2 degree AtrioVentricular block
13. 3 degree AtrioVentricular block
14. Left ventricule hypertrophy
15. Atrial Fibrillation or Flutter
16. Lainnya

Dataset Arrhythmia memiliki missing values pada beberapa fiturnya sehingga perlu dilakukannya pra-pemrosesan data dengan menghapus beberapa fitur atau dengan mengisi nilai rata-rata dari suatu fitur dengan bantuan Python. Selain itu juga terdapat beberapa fitur yang seluruh isinya bernilai 0 dan juga terdapat beberapa fitur yang hampir seluruh isinya bernilai 0. Dikarenakan pada dataset tidak terdapat label pada setiap kolomnya, maka penulis memberikan label untuk 20 kolom pertama, dan sisa kolom lainnya diberikan nama dengan prefix COL[nomor urut fitur].



Gambar 2 Head Data Arrhythmia

Adapun beberapa fitur yang seluruh isinya bernilai nol yang ditampilkan pada Gambar 3 berikut :



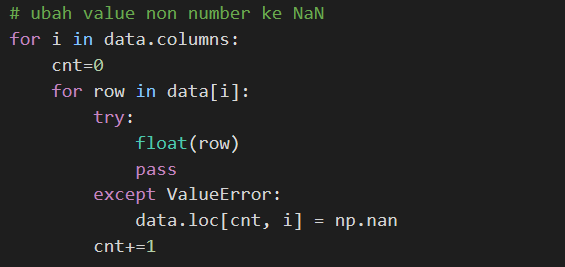
Gambar 3 Daftar Fitur Yang Semua Isinya Bernilai 0

# **BAGIAN II Pra-Pemrosesan Data**

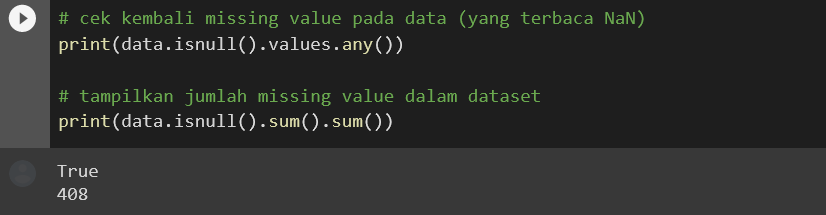
Sebelum dilakukannya pembuatan model dan prediksi data, perlu dilakukannya tahap pra-pemrosesan data dengan harapan untuk mendapatkan hasil prediksi yang lebih akurat. Penulis menemukan bahwa pada data tersebut terdapat beberapa *missing values* pada fiturnya dan perlu untuk dilakukan proses eliminasi dan mengisi nilai rata-rata pada beberapa fitur yang memiliki nilai missing values dengan jumlah yang tidak terlalu banyak. Selain itu juga terdapat beberapa fitur yang semua isinya bernilai 0, dan beberapa fitur lainnya memiliki banyak nilai nol. Berikut merupakan penjabaran tahapan pra-pemrosesan yang dilakukan oleh penulis :

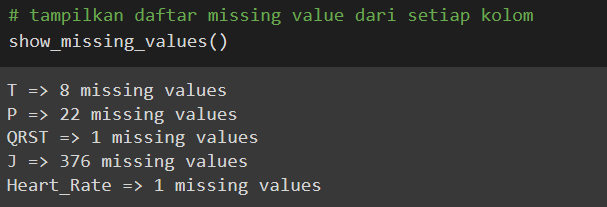
## **Eliminasi fitur dan replace missing values**

Terdapat dua cara untuk penanganan *missing values* yang penulis lakukan, yaitu dengan cara menghapus fitur yang memiliki banyak *missing values* dan mengisi nilai *missing values* dengan nilai rata-rata dari fitur tersebut apabila memiliki jumlah *missing values* yang tidak terlalu banyak. Dikarenakan missing value pada dataset ini didefinisikan dengan karakter “?” yang mengakibatkan library pandas tidak bisa mendeteksi *null value*, maka dilakukan mekanisme pengubahan nilai *Non-Number* ke dalam nilai *NaN (Not a Number)*.

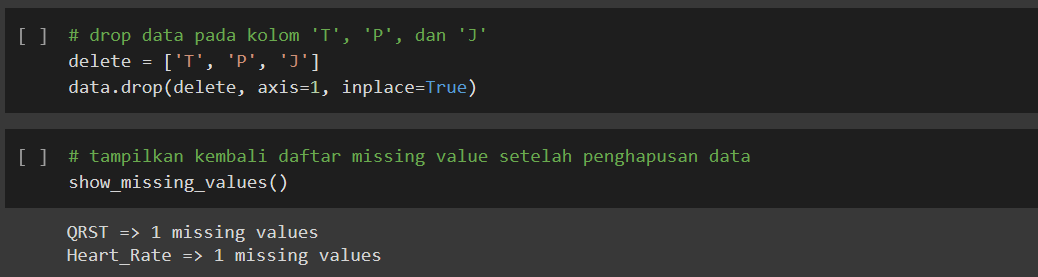


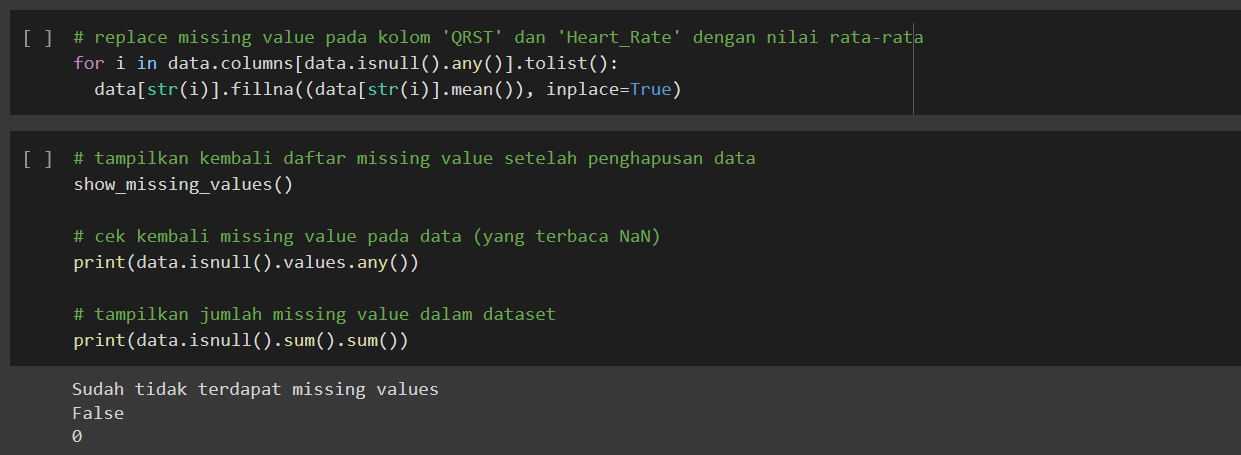
Dari proses pengubahan tersebut, didapatkan hasil sebanyak 408 *missing values* dan selanjutnya dilakukan pencarian lokasi *missing values* tersebut yang berada pada beberapa fitur yang ditampilkan pada gambar dibawah.





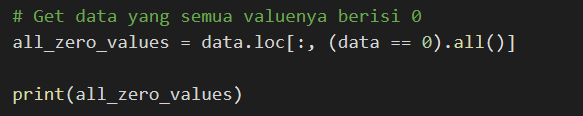
Setelah didapatkan beberapa fitur dengan jumlah missing valuesnya, dilakukan penghapusan pada fitur dengan label **T**, **P**, dan **J** karena memiliki banyak missing values. Kemudian dilakukan pengisian missing values pada sisa fitur lainnya.



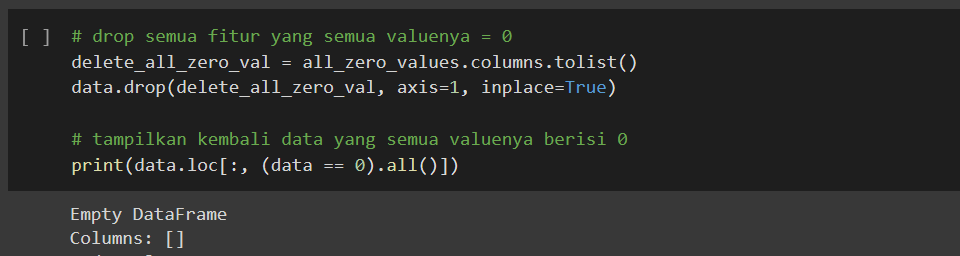


## **Eliminasi fitur yang seluruh valuesnya bernilai 0**

Pada tahapan ini, penulis melakukan pencarian data terlebih dahulu dengan ketentuan semua fitur yang seluruh valuesnya bernilai 0.

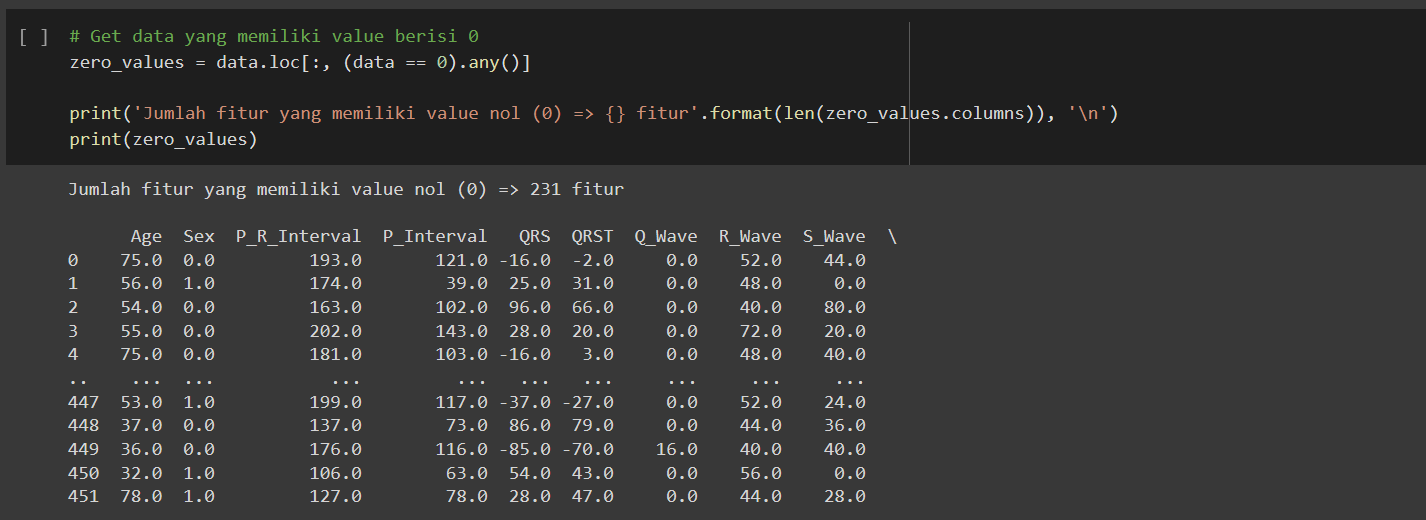


Dari hasil pencarian tersebut, ditemukan sebanyak 17 fitur yang selanjutnya akan dilakukan penghapusan/eliminasi.

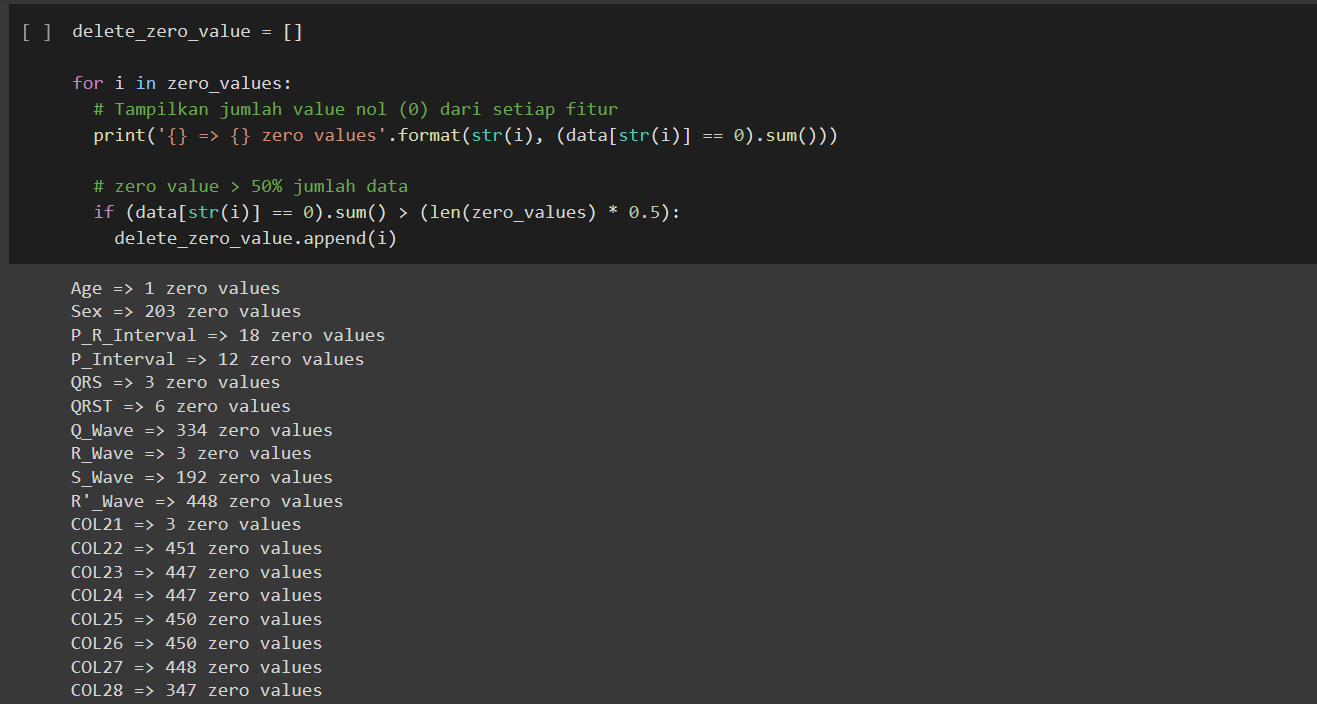


## **Eliminasi fitur dengan rasio jumlah nilai 0 lebih dari 50%**

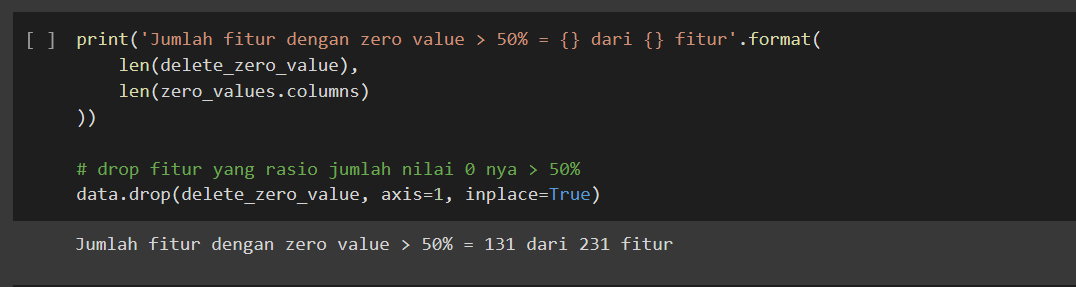
Pada tahapan ini, penulis juga melakukan pencarian data terlebih dahulu dengan ketentuan semua fitur yang memiliki values bernilai 0.



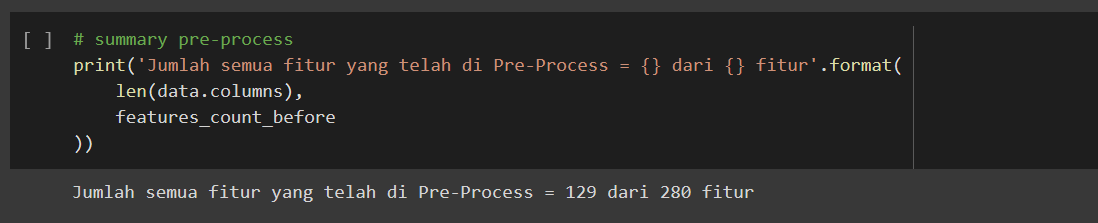
Dari hasil pencarian tersebut, ditemukan sebanyak 231 fitur dimana masing-masing fiturnya memiliki isi yang bernilai 0. Untuk mengetahui jumlah nilai nol dari masing-masing fitur tersebut, dilakukan kembali pencarian jumlah nilai nolnya pada masing-masing fitur seperti gambar dibawah :



Penulis melampirkan screenshot yang hanya sebagian hasilnya saja, karena dirasa terlalu panjang dikarenakan terdapat 231 baris output yang ditampilkan dari hasil pencarian tersebut. Setelah dilakukan pencarian data, penulis melakukan penghapusan fitur dengan ketentuan memiliki jumlah nilai 0 lebih dari 50% jumlah nilai pada setiap fiturnya.



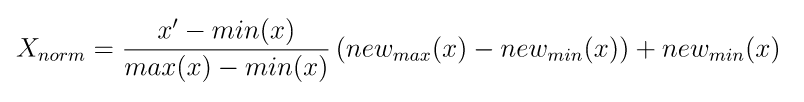
Ditemukan sebanyak 131 dari 231 fitur yang memiliki jumlah nilai 0 yang lebih dari 50% jumlah nilai pada setiap fiturnya, dan kemudian dilakukan eliminasi sebanyak 131 fitur. Dari hasil pra-pemrosesan pada beberapa tahapan diatas, didapatkan sisa sebanyak 129 dari 280 fitur yang telah dilakukan pra-pemrosesan.

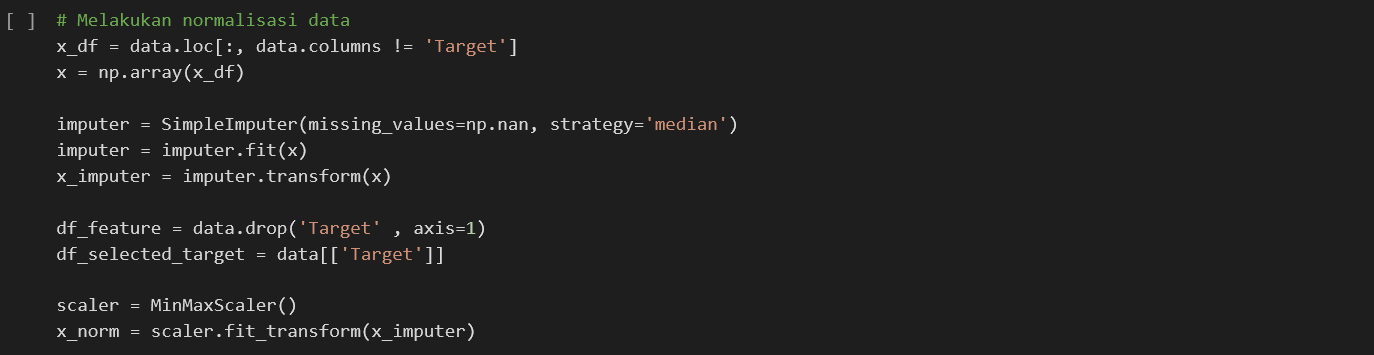


## **Normalisasi data dan feature selection**

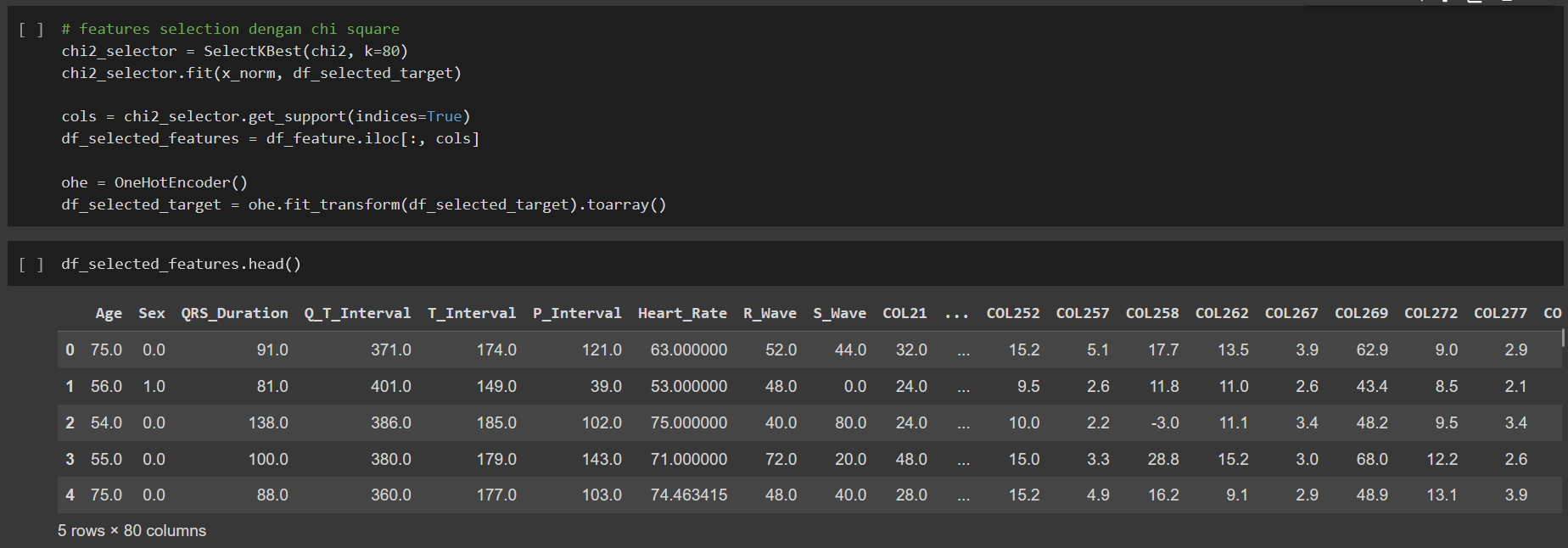
Pada tahapan akhir pra-pemrosesan, penulis melakukan normalisasi data yang bertujuan untuk memastikan record pada dataset tetap konsisten dan menghilangkan redundansi data agar data memiliki kualitas yang lebih baik. Normalisas dibutuhkan karena beberapa atribut dari dataset ini memiliki skala atau rentang yang berbeda-beda sehingga terdapat ketimpangan dimana terdapat data yang terlampau tinggi dan terdapat data yang terlampau rendah.

Penulis menggunakan bantuan library dari sklearn untuk melakukan normalisasi dan feature selection. Pertama, penulis menggunakan function *SimpleImputer()* untuk penanganan nilai data yang kosong/hilang apabila masih terdapat data kosong yang belum tertangani pada tahapan sebelumnya. Kemudian penulis melakukan normalisasi *Min-Max* dengan mengubah sekumpulan data menjadi skalai mulai dari 0 (Min) hingga 1 (Max) dengan bantuan function *MinMaxScaler()*. Berikut terlampir formula perhitungan normalisasi Min-Max dan penerapannya :



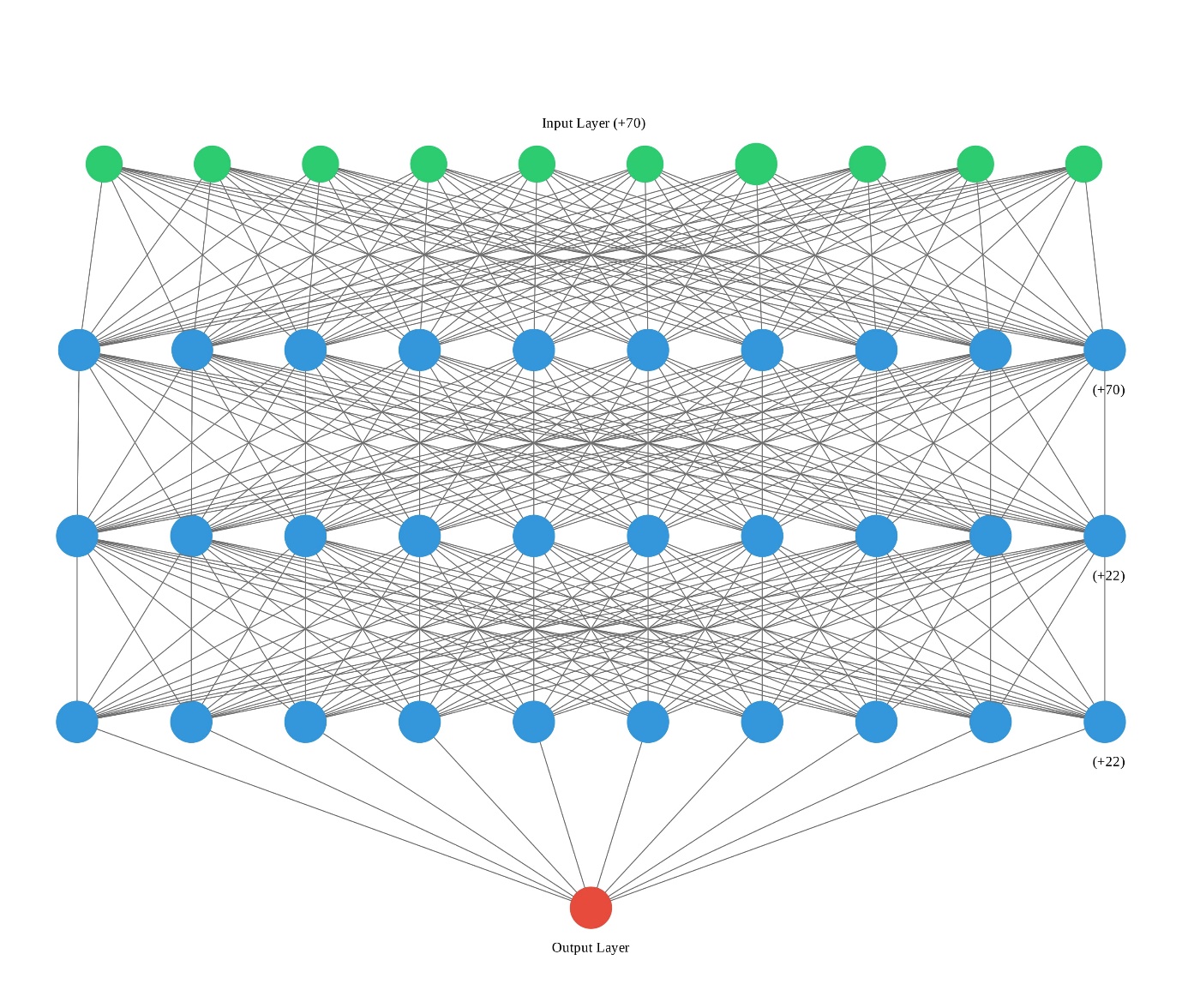


Selanjutnya, penulis melakukan *feature selection* dengan tujuan untuk mendapatkan fitur-fitur optimal yang memiliki korelasi tinggi dan mengeliminasi fitur-fitur yang tidak berpengaruh atau fitur yang tidak relevan. Penulis menggunakan metode feature selection *Chi Square* dan menggunakan bantuan function SelectKBest() dengan memilih 80 fitur terbaik yang akan digunakan pada proses prediksi dataset nanti.



# **BAGIAN III Penerapan Algoritma**

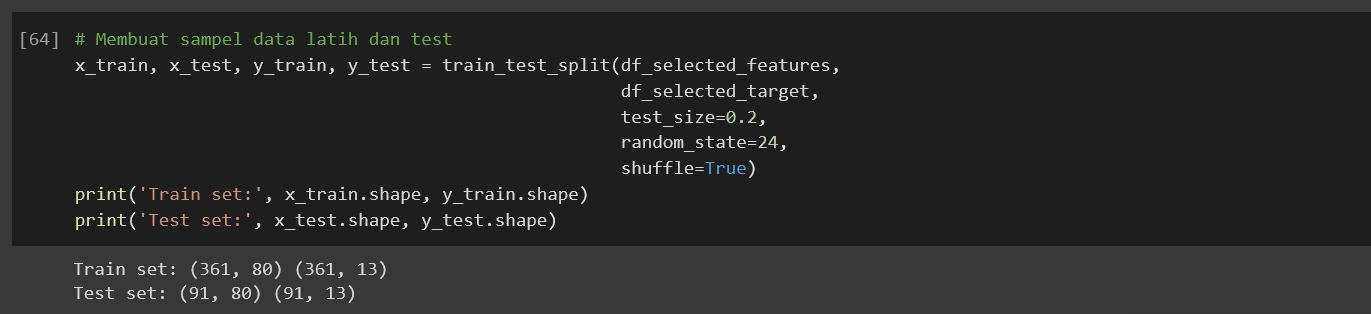
Pada proses penerapan algoritmanya, penulis menggunakan algoritma *Artifical Neural Network* (*ANN*) dikarenakan algoritma ini merupakan algoritma yang sesuai dengan dataset yang didapatkan untuk mengklasifikasikan ada atau tidaknya aritmia jantung pada pasien berdasarkan 16 kelompok data. Algoritma *ANN* ini dibangun dengan 1 input layer sebanyak 80 nodes, 2 hidden layer masin-masing sebanyak 32 nodes, dan 1 output layer sebanyak 1 nodes.



Berikut penjabaran implementasi ANN dalam penyelesaian studi kasus Arrhythmia Data dengan bantuan library *Tensorflow* pada *Python* :

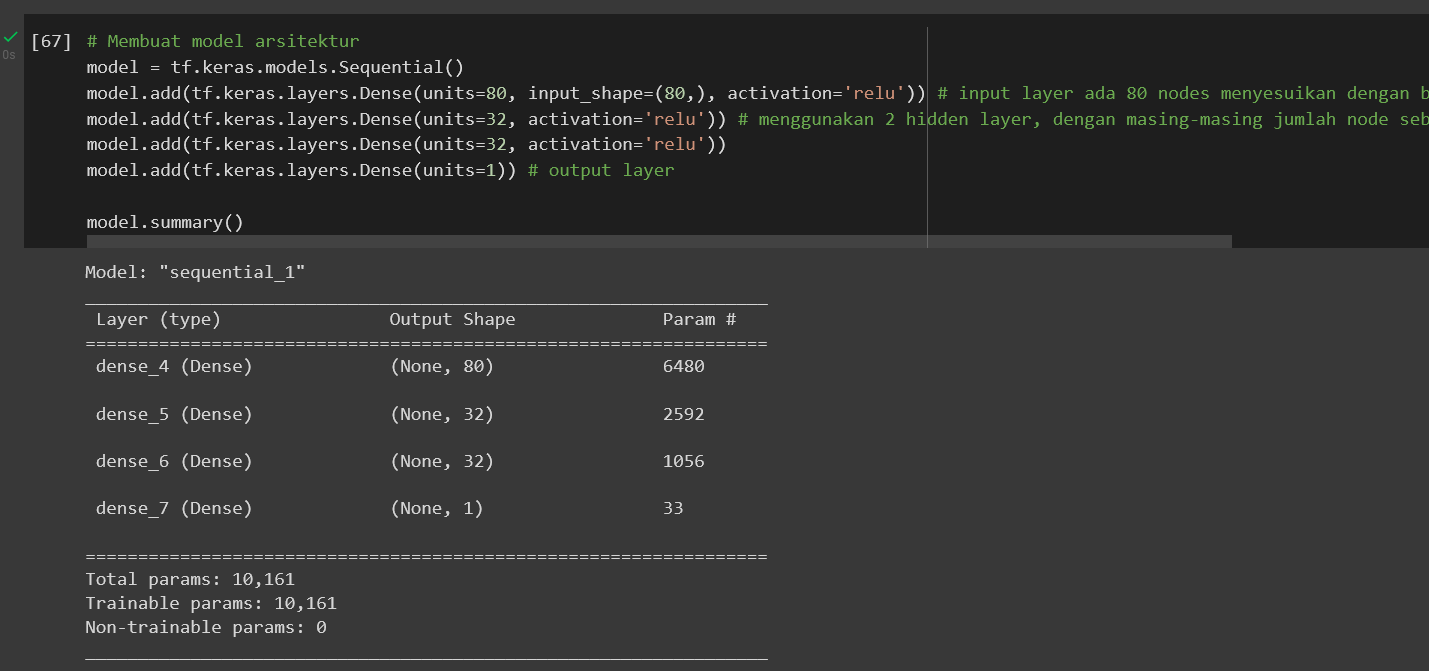
## **Pembuatan sample data latih dan data test**

Pada penerapannya, dilakukan pemisahan dataset untuk data latih dan data test dengan rasio 80:20 dengan detail 80% untuk data latih dan 20% untuk data test.



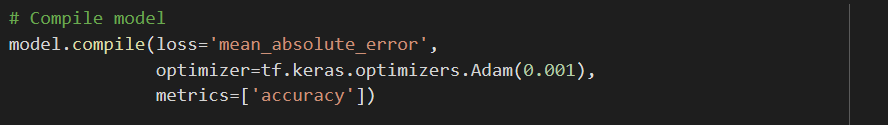
## **Pembuatan model arsitektur**

Model arsitektur yang dibangun menggunakan 1 input layer yang terdiri dari 80 nodes dan 80 input shape yang merepresentasikan 80 fitur yang digunakan, 2 hidden layer dengan masing-masing 32 nodes, dan 1 output layer dengan 1 nodes. Fungsi aktivasi yang digunakan pada input layer dan hidden layer adalah relu.



## **Proses compile model**

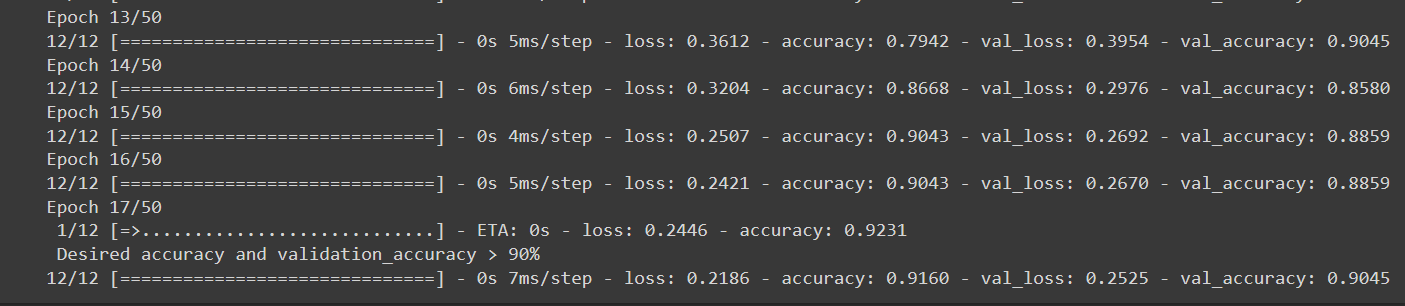
Pada proses compile modelnya, penulis menggunakan *Mean Absolute Error* (*MAE*) sebagai paramter loss dengan menggunakan *Adam* sebagai optimizernya dengan learning rate sebesar 0,001. Untuk metriknya digunakan *accuracy* sebagai acuan akurasi pada saat proses training modelnya.



## **Training Model**

Pada proses training modelnya, penulis mendefinisikan paramter x\_train sebagai fitur yang akan di latih, y\_train sebagai target fitur, epoch sebanyak 50 iterasi, data validation yang digunakan untuk memvalidasi model dan mencegah *overfitting*, dan callbacks untuk menghentikan iterasi apabila nilai akurasi dan akurasi validasinya lebih besar dari 90%.



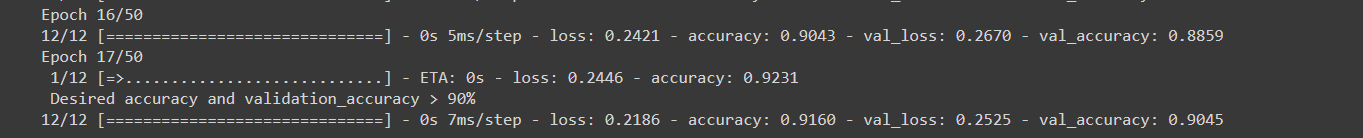


**Link Google Collab :** <https://colab.research.google.com/drive/1mOPZ9qemNPlMcmRo4GOSzXBykN-gE7dW?usp=sharing>

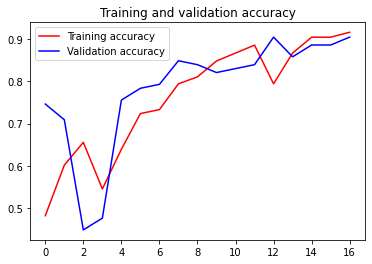
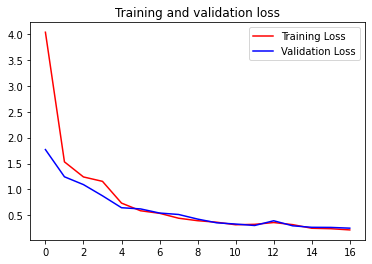
**Link Github :** <https://github.com/azmirizkifar20/Machine-Learning---Implementasi-ANN>

# **BAGIAN IV Evaluasi Hasil**

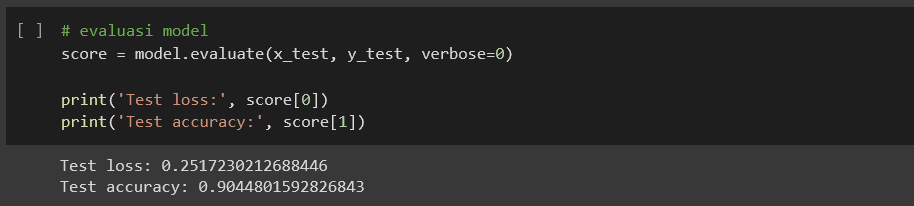
Pengukuran performansi model dilihat dari nilai akurasi yang didapatkan dari epoch ke 17 dengan nilai akurasi sebesar 0,9160 dan nilai akurasi validasi sebesar 0,9045.



Dari hasil akurasi tersebut, penulis memvisualisasikan hasilnya kedalam bentuk grafik yang dapat dilihat pada gambar berikut :

Pada tahap evaluasi model dengan menggunakan data test didapatkan nilai loss sebesar 0,25 dan nilai akurasinya sebesar 0,904.



Dari hasil nilai evaluasi model berdasarkan nilai akurasi dan validasi akurasinya, didapatkan nilai yang diharapkan yaitu nilai akurasi yang lebih dari 90%. Hal ini membuktikan bahwa model bekerja dengan baik dalam mempelajari data-data latih tersebut ketika memvalidasi input dan target pada datanya. hasil akurasi tersebut menunjukkan bahwa model yang dibangun memiliki performansi yang baik untuk digunakan dalam memprediksi data baru. Hasil nilai akurasi dari data training dan data test tersebut tidak terpaut jauh yang dapat menunjukkan bahwa model tidak underfit ataupun overfit.