

Nama : Andreas Hasiholan Sinaga

NIM: 110321303

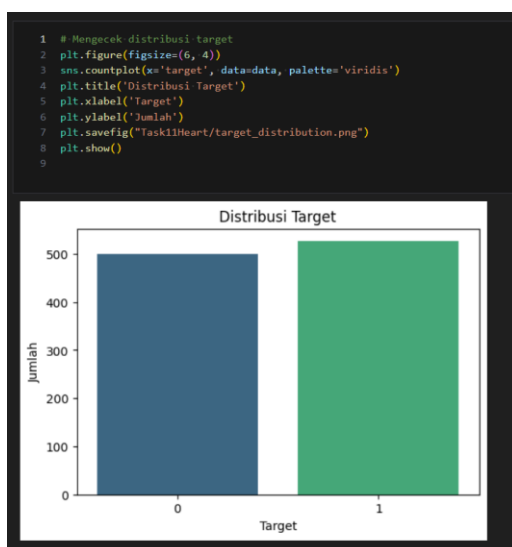
## Deep Learning Model dengan Dataset Heart Disease

```
1 # =====
2 # Eksplorasi Data Awal (EDA)
3 # =====
4 # Menampilkan informasi dataset
5 print("Informasi Dataset:")
6 print(data.info())
```

[3]

```
** Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         1025 non-null   int64
1   sex         1025 non-null   int64
2   cp          1025 non-null   int64
3   trestbps    1025 non-null   int64
4   chol        1025 non-null   int64
5   fbs         1025 non-null   int64
6   restecg     1025 non-null   int64
7   thalach     1025 non-null   int64
8   exang       1025 non-null   int64
9   oldpeak     1025 non-null   float64
10  slope       1025 non-null   int64
11  ca          1025 non-null   int64
12  thal        1025 non-null   int64
13  target      1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
None
```

deep learning model dengan dataset heart disease memiliki column sebanyak 13 yang memiliki tipe data integer dan float .



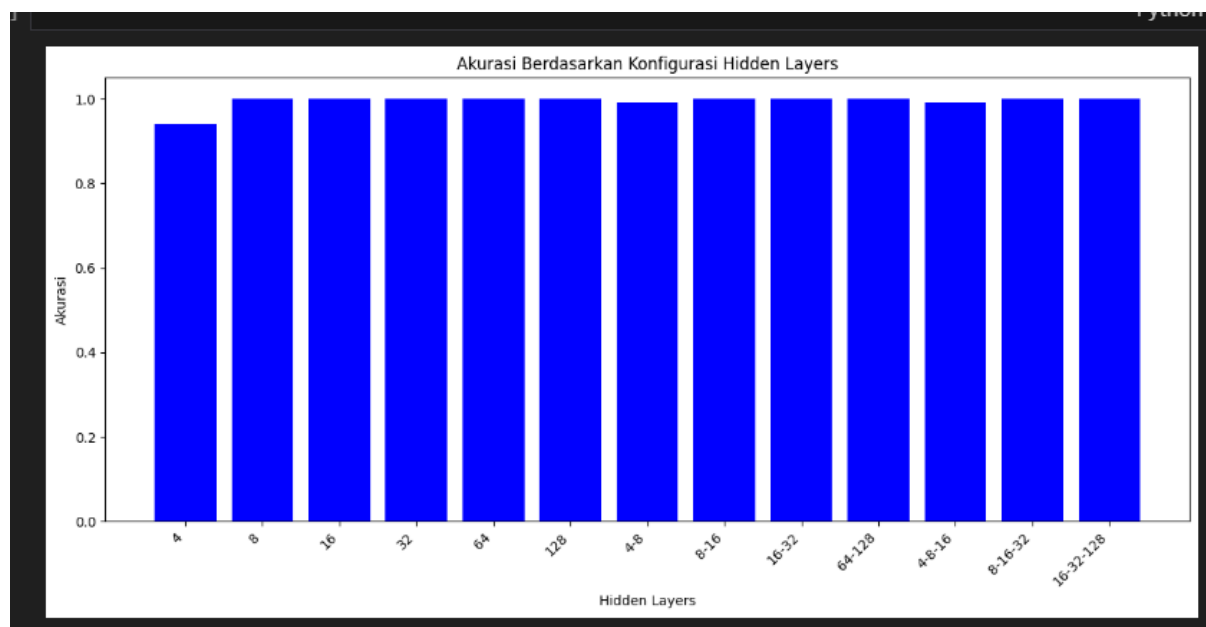
Dapat kita lihat distribusi target 0 sebanyak 500 dan target 1 sebanyak 600

```

1 # =====
2 # Eksperimen Hyperparameter
3 # =====
4 # Konfigurasi parameter untuk eksperimen
5 hidden_layer_configs = [
6     [4], [8], [16], [32], [64], [128],
7     [4, 8], [8, 16], [16, 32], [64, 128],
8     [4, 8, 16], [8, 16, 32], [16, 32, 128]
9 ]
10 activation_functions = [nn.Identity, nn.Sigmoid, nn.ReLU, nn.Softmax, nn.Tanh]
11 epoch_values = [1, 10, 25, 50, 100, 250]
12 learning_rates = [10, 1, 0.1, 0.01, 0.001, 0.0001]
13 batch_sizes = [16, 32, 64, 128, 256, 512]
14

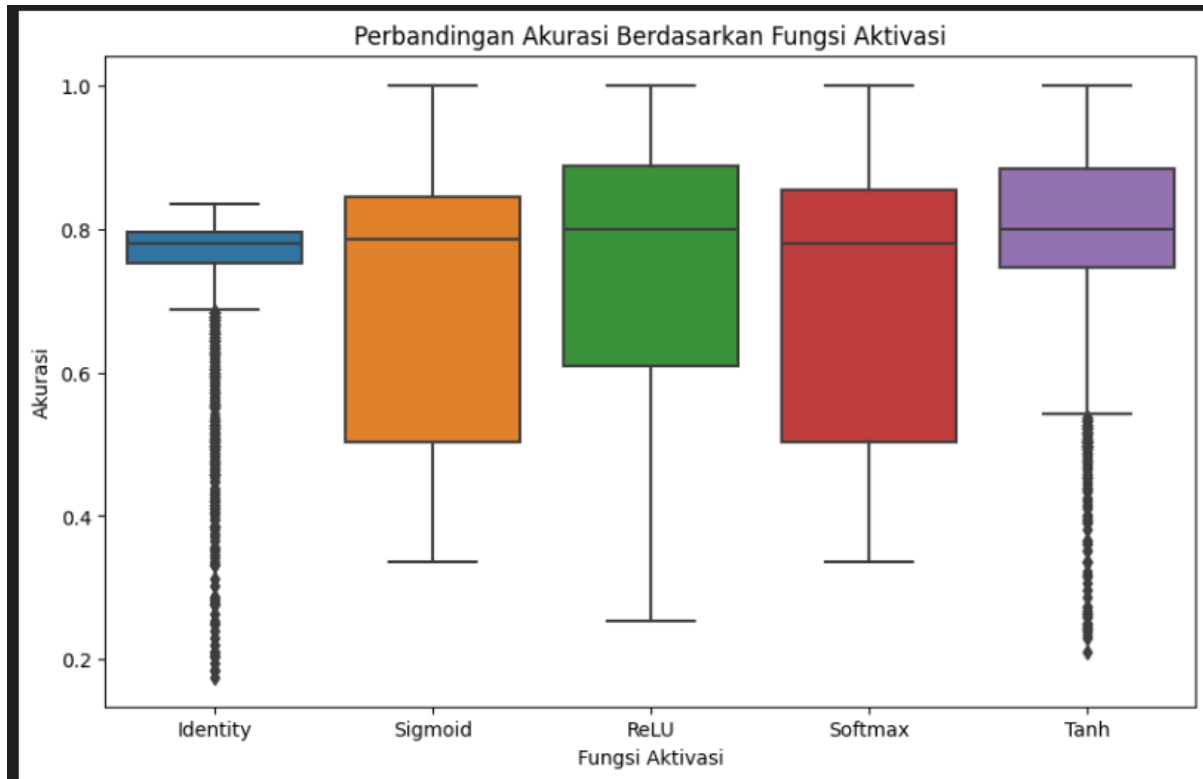
```

Kode ini mendefinisikan konfigurasi hyperparameter untuk eksperimen pelatihan model klasifikasi menggunakan berbagai kombinasi parameter. Hyperparameter yang dicoba meliputi jumlah neuron pada lapisan tersembunyi (hidden\_layers), dengan variasi mulai dari satu lapisan sederhana seperti [4] hingga struktur multi-lapisan seperti [32, 16, 8]. Fungsi aktivasi (activation\_fn) mencakup opsi ReLU, Sigmoid, Tanh, Linear, dan Softmax untuk menangani pola non-linearitas dalam data. Selain itu, nilai learning\_rate diatur dari 10 hingga 0.0001 untuk mengeksplorasi kecepatan pembaruan bobot, batch\_size bervariasi antara 16 hingga 512 untuk mengelola jumlah data yang diproses per iterasi, dan jumlah epochs mulai dari 1 hingga 250 untuk menentukan jumlah iterasi pelatihan. Tujuan dari konfigurasi ini adalah mengevaluasi performa model dengan berbagai kombinasi hyperparameter untuk menemukan pengaturan terbaik yang menghasilkan akurasi tertinggi.

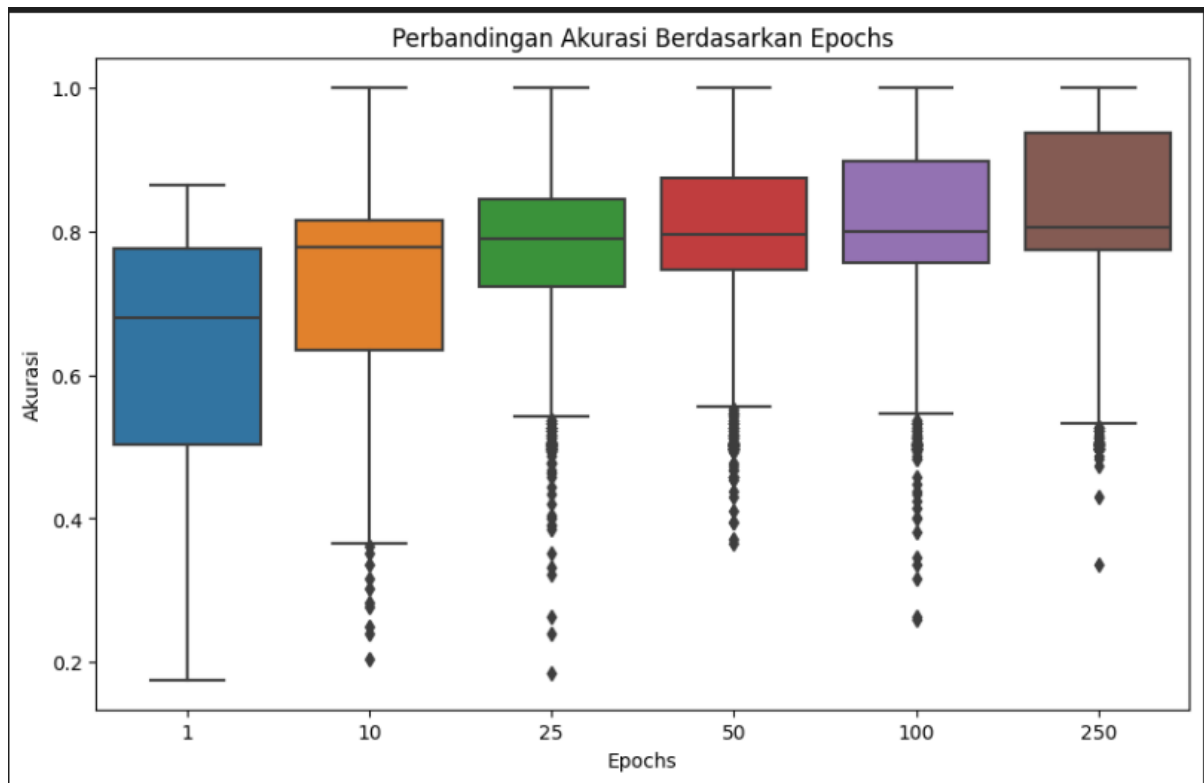


Grafik ini menunjukkan akurasi model berdasarkan konfigurasi jumlah dan struktur hidden layers. Setiap konfigurasi hidden layers, yang direpresentasikan pada sumbu horizontal, memiliki performa akurasi yang hampir merata dengan nilai mendekati 1.0, menunjukkan bahwa variasi jumlah lapisan tersembunyi dan jumlah neuron per lapisan tidak memberikan

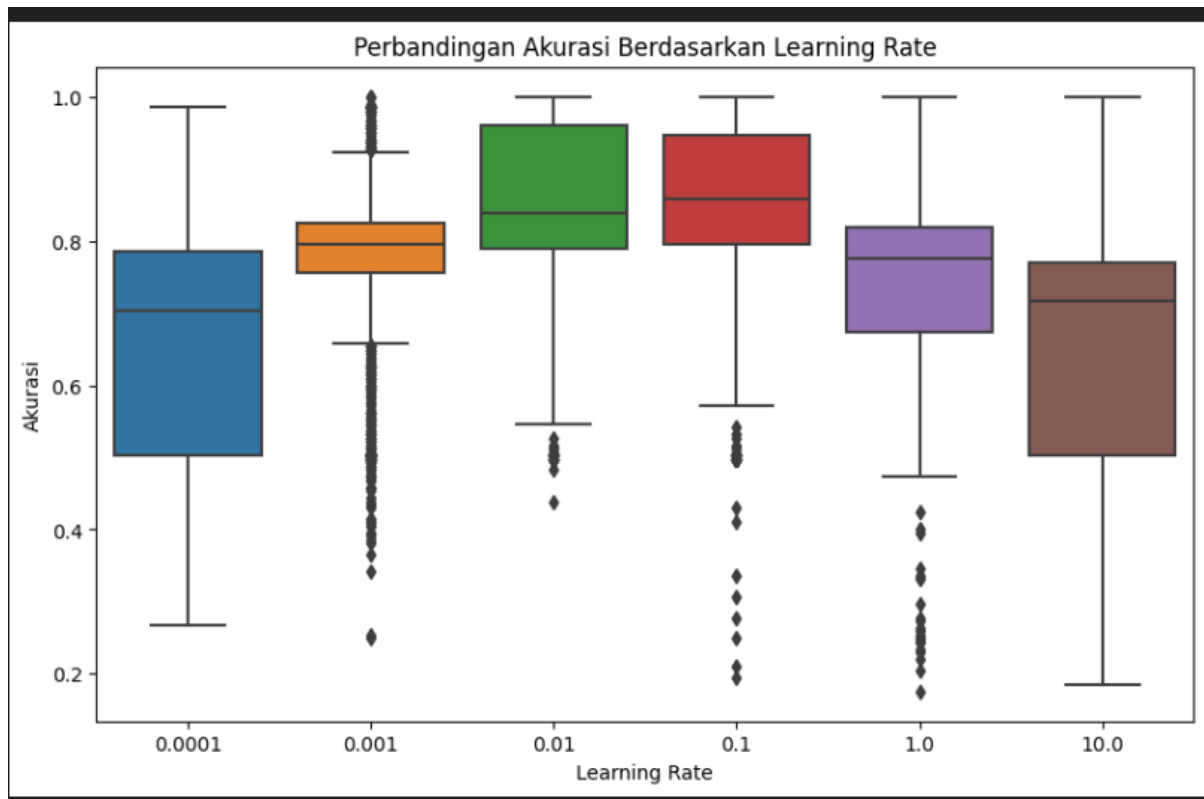
pengaruh signifikan terhadap performa akhir model. Hasil ini mengindikasikan bahwa model MLP memiliki kemampuan yang cukup baik untuk beradaptasi dengan berbagai struktur hidden layers, asalkan parameter lainnya, seperti learning rate dan fungsi aktivasi, dioptimalkan dengan baik. Dengan demikian, pemilihan konfigurasi hidden layers dapat lebih disesuaikan dengan kebutuhan efisiensi komputasi tanpa mengorbankan akurasi secara signifikan.



Grafik scatter plot menunjukkan perbandingan akurasi model berdasarkan fungsi aktivasi yang digunakan, yaitu Identity, Sigmoid, ReLU, Softmax, dan Tanh. Fungsi aktivasi Tanh memiliki performa akurasi tertinggi dengan distribusi yang relatif sempit di sekitar median, menunjukkan konsistensi performa yang baik meskipun terdapat beberapa outlier di akurasi rendah. Sigmoid dan ReLU juga menunjukkan performa yang kompetitif, tetapi dengan distribusi yang lebih lebar, menandakan variabilitas yang lebih tinggi. Fungsi Softmax memiliki performa yang mirip dengan Sigmoid, namun sedikit lebih stabil. Sebaliknya, fungsi Identity menunjukkan median akurasi yang lebih rendah dibandingkan fungsi aktivasi lainnya, dengan jumlah outlier yang signifikan di akurasi rendah. Dari grafik ini, dapat disimpulkan bahwa fungsi aktivasi Tanh memberikan akurasi terbaik dan paling konsisten, sementara fungsi Identity kurang optimal untuk menghasilkan performa model yang tinggi.



Grafik boxplot ini menunjukkan distribusi akurasi model berdasarkan jumlah epochs yang digunakan dalam pelatihan. Pada epochs yang lebih kecil, seperti 1 dan 10, median akurasi lebih rendah dengan distribusi yang lebih luas, mencerminkan bahwa model belum terkonvergensi sepenuhnya dan menghasilkan variasi performa yang lebih besar. Seiring dengan bertambahnya epochs, seperti pada 25, 50, 100, hingga 250, median akurasi meningkat secara konsisten, mendekati 0.9, dan distribusi menjadi lebih sempit, menunjukkan bahwa model mencapai stabilitas dan konsistensi dalam akurasi. Namun, terdapat beberapa outlier pada semua jumlah epochs, terutama pada nilai epochs kecil, yang menunjukkan kombinasi parameter tertentu menghasilkan performa yang buruk. Dari grafik ini, dapat disimpulkan bahwa jumlah epochs yang lebih tinggi, seperti 100 atau 250, memberikan akurasi yang lebih baik dan lebih stabil untuk model.



Dari hasil beberapa kombinasi parameter, diperoleh grafik boxplot yang menunjukkan distribusi akurasi berdasarkan nilai learning rate yang digunakan. Grafik ini menunjukkan bahwa nilai learning rate yang lebih kecil, seperti 0.0001 dan 0.001, menghasilkan median akurasi yang lebih tinggi dan stabil, meskipun terdapat beberapa outlier pada akurasi rendah. Learning rate menengah, seperti 0.01 dan 0.1, menunjukkan performa yang cukup baik, tetapi distribusinya mulai meluas, mengindikasikan adanya peningkatan variabilitas. Sebaliknya, learning rate yang besar, seperti 1.0 dan 10.0, menghasilkan median akurasi yang lebih rendah dan performa yang kurang stabil dengan lebih banyak outlier, menunjukkan bahwa learning rate yang terlalu besar dapat menghambat konvergensi model. Kesimpulannya, learning rate kecil hingga menengah (0.0001 hingga 0.01) adalah pilihan terbaik untuk menghasilkan akurasi yang lebih baik dan konsisten.