Prediksi Jumlah Kasus Diabetes Tahunan Menggunakan *Neural Network* Berbasis Regresi

Disusun oleh :

Muhammad Fakhri Ruslan	(2206029935)
Bryan Jonathan	(2206052780)
Muhammad Faris Naufaldi	(2306153843)
Natalius Desta Riyanto	(2306202624)

UNIVERSITAS INDONESIA
DEPOK
2025

DAFTAR ISI

1. PENDAHULUAN

1.1 Latar Belakang

Diabetes merupakan salah satu penyakit tidak menular yang terus mengalami peningkatan secara global, termasuk di Indonesia. Berdasarkan laporan dari International Diabetes Federation (IDF), jumlah penderita diabetes terus meningkat dari tahun ke tahun seiring dengan pertumbuhan populasi, perubahan gaya hidup, dan peningkatan prevalensi obesitas serta penuaan penduduk. Kondisi ini tidak hanya menimbulkan beban bagi sektor kesehatan, tetapi juga menimbulkan dampak ekonomi yang signifikan.

Dengan tren peningkatan yang terus terjadi, kemampuan untuk melakukan prediksi jumlah kasus diabetes di masa mendatang menjadi sangat penting. Prediksi ini dapat digunakan sebagai dasar dalam menyusun kebijakan pencegahan, perencanaan layanan kesehatan, serta alokasi sumber daya secara lebih efektif dan efisien.

Dalam konteks analisis data, pendekatan konvensional seperti regresi linier sering kali tidak mampu menangkap hubungan yang kompleks dan nonlinier antara faktor-faktor risiko dan jumlah kasus diabetes. Oleh karena itu, pendekatan berbasis machine learning, khususnya *neural network*, menjadi alternatif yang menjanjikan. *Neural network* memiliki kemampuan untuk mempelajari pola yang kompleks dari data dan melakukan prediksi secara lebih fleksibel.

Penelitian ini bertujuan untuk membangun model prediksi jumlah kasus diabetes di Indonesia dengan menggunakan data historis dari tahun 2010 hingga 2024. Model dikembangkan menggunakan *neural network* dengan beberapa variabel input, seperti jumlah populasi, populasi usia lanjut (>40 tahun), konsumsi gula, dan harga gula. Melalui penelitian ini, diharapkan dapat diperoleh gambaran seberapa baik model dapat memprediksi jumlah kasus diabetes, serta mengidentifikasi fitur-fitur yang paling berpengaruh dalam proses prediksi tersebut.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian kami adalah sebagai berikut.

1. Bagaimana membangun model *neural network* untuk memprediksi kenaikan kasus diabetes per tahun?
2. Seberapa akurat model yang telah dibuat dalam melakukan prediksi kenaikan kasus diabetes per tahun?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, tujuan penelitian kami adalah sebagai berikut.

1. Membuat model untuk memprediksi jumlah kenaikan kasus diabetes per tahun dengan *neural network*.

2. Mengukur akurasi model dan mengetahui tren kenaikan kasus diabetes per tahun yang diprediksi.

2. ISI

2.1 *Dataset*

Penelitian ini menggunakan *dataset* yang terdiri dari data tahunan mulai dari tahun 2010 hingga 2024 yang disusun sendiri berdasarkan informasi yang didapat dari berbagai sumber, seperti Kemenkes RI dan International Diabetes Federation. *Dataset* kami mencakup variabel-variabel utama yang dianggap relevan terhadap jumlah kasus diabetes antara lain:

- Populasi (Juta) : total populasi di Indonesia per tahun
- Rata-rata kadar gula per minggu (ons) : estimasi konsumsi gula per orang
- Total populasi umur >40 tahun (Juta) : populasi berisiko tinggi terkena diabetes
- Harga Gula per 1 kg : sebagai proksi ekonomi dan konsumsi
- Total Kasus Diabetes (Juta) : data target (label) yang akan diprediksi

Berikut *dataset* yang kami gunakan dalam penelitian ini:

Tahun	Populasi (Juta)	Rata-rata kadar gula per minggu (ons)	Total populasi umur >40 (Juta)	Harga Gula per 1kg	Total Kasus Diabetes (Juta)
2010	237,6	1,617	70,73121	12.000	6,9
2011	249,5	1,555	71,441722	12.000	7,3
2012	252,7	1,344	74,370283	12.170	7
2013	255,9	1,38	77,305066	13.041	7,6
2014	258,9	1,328	79,642965	11.782	9
2015	261,8	1,441	81,397514	12.570	9,1
2016	264,6	1,581	83,947193	15.715	9,6
2017	266,9	1,462	86,492771	14.378	10,3
2018	267,3	1,433	88,6273	13.010	10,2
2019	268	1,391	91,1508	12.917	10,7
2020	270,2	1,376	93,6748	13.596	18,69
2021	272,6	1,415	96,3164	13.236	19,47
2022	275,7	1,337	99,1444	13.392	19,5
2023	278,6	1,228	101,6368	14.327	20,4
2024	281,6	1,133	104,3365	18.628	20

Target variabel yang akan diprediksi oleh model adalah Total Kasus Diabetes (Juta). Variabel-variabel input dipilih karena dianggap relevan secara teoritis maupun praktis dalam menjelaskan penyebab atau faktor risiko dari peningkatan jumlah kasus diabetes. Untuk memastikan keakuratan pemodelan, seluruh fitur numerik (kecuali target) kemudian mengalami

proses standardisasi (*standardization*) agar memiliki skala yang sebanding dan tidak mendominasi model secara numerik.

2.2. Metodologi Penelitian

Penelitian ini bertujuan untuk memprediksi jumlah kasus diabetes di Indonesia berdasarkan data tahunan menggunakan algoritma *Neural Network*. Metodologi yang digunakan meliputi beberapa tahapan utama, yaitu *preprocessing* data, perancangan arsitektur model, pelatihan model, serta evaluasi performa model.

1. *Preprocessing* Data

Sebelum dilakukan pemodelan, data terlebih dahulu diproses agar layak digunakan oleh model *machine learning*. Tahapan *preprocessing* meliputi:

- Pembersihan data numerik: konversi nilai desimal dari format koma (,) menjadi titik (.) agar dapat dibaca sebagai *float*.
- Pemisahan data *train-test*: data dibagi berdasarkan waktu, yaitu:
    - Data tahun 2010–2021 digunakan untuk pelatihan (*training set*).
    - Data tahun 2022–2024 digunakan sebagai data uji (*testing set*).
- Standardisasi fitur (StandardScaler): seluruh variabel input (fitur) distandarkan agar memiliki mean = 0 dan standar deviasi = 1. Ini penting agar model *neural network* dapat melakukan pembaruan bobot secara stabil dan adil antarfitur.

2. Arsitektur Model *Neural Network*

Model yang digunakan adalah *feedforward neural network* (*fully connected*) yang dibangun menggunakan *library* Keras dengan *backend* TensorFlow. Arsitektur terdiri dari:

- *Input layer* sebanyak 4 neuron (untuk 4 variabel input)
- *Hidden layer* 1 : 10 neuron, aktivasi *ReLU*
- *Dropout layer* : p = 0.5 (untuk mencegah *overfitting*)
- *Hidden layer* 2 : 5 neuron, aktivasi *ReLU*
- *Dropout layer* p : 0.5
- *Output layer* : 1 neuron, aktivasi linier, karena *output* berupa nilai kontinu (jumlah kasus)

3. Proses Pelatihan Model

Model dilatih menggunakan:

- *Loss function* : *Mean Squared Error* (*MSE*)
- *Optimizer* : *Stochastic Gradient Descent* (*SGD*) dengan *learning rate* awal 0.003
- *Batch size* : 1
- *Epoch* maksimum : 1000

- EarlyStopping : untuk menghentikan pelatihan jika tidak ada peningkatan val_loss selama 10 *epoch*.
- ReduceLROnPlateau : menurunkan *learning rate* sebesar 50% jika val_loss stagnan selama 5 *epoch*, dengan minimum *learning rate* 1e−10.

4. Evaluasi Model

Kinerja model diuji pada data *test* (2022–2024) menggunakan beberapa metrik evaluasi regresi, yaitu:

- *Mean Absolute Error* (*MAE*) : rata-rata deviasi absolut antara hasil prediksi dan aktual.
- *Mean Squared Error* (*MSE*) : rata-rata kuadrat dari error prediksi.
- *R^2 Score* (Koefisien Determinasi) : mengukur seberapa besar variasi target yang dapat dijelaskan oleh model.

Model juga divisualisasikan melalui grafik berikut:

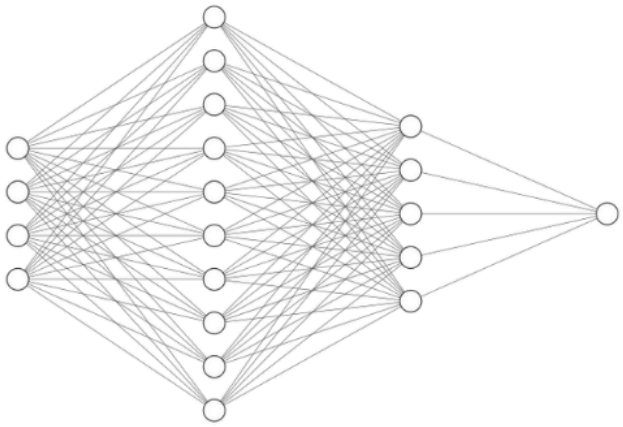
- Grafik *loss training* vs. validasi
- Grafik prediksi vs. aktual (*test set*)

2.3 Implementasi Model

Model *neural network* diimplementasikan menggunakan pustaka Keras dengan *backend* TensorFlow. Proses implementasi meliputi tahapan sebagai berikut:

- Inisialisasi dan *seed*: dilakukan *set seed* untuk memastikan hasil *reproducible*.
- *Import* data: file CSV dimuat, dan nilai desimal disesuaikan dengan mengganti koma (,) menjadi titik (.).
- Standardisasi: fitur diubah menggunakan StandardScaler() agar memiliki skala seragam.
- Pemisahan data: data dilatih menggunakan tahun 2010–2021 dan diuji menggunakan tahun 2022–2024.

Arsitektur model disusun secara berlapis (*fully connected*):



- *Input layer* : 4 neuron, masing-masing mewakili 1 fitur:
    1. Populasi (Juta),
    2. Rata-rata kadar gula per minggu,
    3. Total Populasi usia > 40 Tahun,
    4. Harga gula per 1 kg.

- *Hidden layer* 1 : 10 neuron, aktivasi *ReLU*. Lapisan pertama kali yang memproses input dan mulai belajar pola nonlinier.

- *Hidden layer* 2 : 5 neuron, aktivasi *ReLU*. Bertugas untuk menyempurnakan representasi fitur sebelum dipetakan ke *output*.

- *Output layer* : 1 neuron, aktivasi linier untuk masalah regresi.

  Model dikompilasi menggunakan:

    - Optimizer : SGD (lr=0.003)
    - Loss function : mean squared error
    - Metrik : mae

  Untuk mencegah *overfitting* dan menstabilkan pelatihan:

    - EarlyStopping digunakan (patience=10, restore_best_weights=True)
    - ReduceLROnPlateau (patience=5, factor=0.5, min_lr=1e−10)

Model dilatih dengan *batch size* 1 dan maksimal 1000 *epoch*, namun biasanya berhenti lebih awal karena *early stopping*.

Dalam proses ini kami menggunakan dua fungsi aktivasi berikut.

- *Rectified Linear Unit* (*ReLU*)

  ReLU adalah fungsi aktivasi yang digunakan secara luas dalam jaringan saraf, terutama untuk *hidden layer*. Fungsi ini didefinisikan sebagai:

  $$\text{ReLU}(x) = \max(0, x)$$

  Artinya, *output* yang didapat adalah nilai input itu sendiri jika inputnya positif, dan nol jika inputnya negatif. Penggunaan *ReLU* sangat populer karena sifatnya yang sederhana dan kemampuannya untuk mempercepat konvergensi model. Selain itu, *ReLU* dapat membantu mengatasi permasalahan *vanishing gradient*, yaitu kondisi di mana nilai turunan (gradien) dari fungsi aktivasi menjadi sangat kecil (hampir nol), sehingga menghambat proses pembelajaran dan menghalangi pembaruan bobot secara signifikan pada lapisan terdalam. Dengan menerapkan *ReLU* pada lapisan tersembunyi, model menjadi lebih mampu untuk menangkap hubungan nonlinier antarfitur dalam *dataset*.

- Linier

  Fungsi aktivasi linier digunakan pada *output layer* untuk prediksi regresi. Fungsi ini sangat sederhana:

  $$f(x) = x$$

  Fungsi linier tidak melakukan transformasi pada *output*, sehingga nilai yang dihasilkan model berada dalam rentang kontinu dan tidak dibatasi. Hal ini sesuai dengan karakteristik tugas regresi, di mana target (jumlah kasus diabetes) bersifat numerik dan kontinu. Fungsi linier pada *output layer* memastikan model dapat memetakan input ke nilai prediksi numerik yang realistis tanpa distorsi.

  Dengan kombinasi *ReLU* di *hidden layers* dan linier di *output layer*, model diharapkan mampu menangkap pola kompleks dalam data sekaligus menghasilkan prediksi nilai kontinu yang akurat.

2.4 Analisis Hasil

2.4.1 Standardisasi *Dataset*

Dalam proses pemodelan, setiap fitur input memiliki skala yang berbeda. Sebagai contoh, fitur “Populasi” memiliki rentang nilai ratusan juta, sementara fitur “Harga Gula” hanya berada pada ribuan. Ketidakseimbangan skala antarfitur ini dapat menyebabkan model *neural network* memberikan bobot yang terlalu besar pada fitur berskala besar (seperti populasi), dan mengabaikan fitur lain yang juga penting. Oleh karena itu, diperlukan proses standardisasi untuk menyeimbangkan skala seluruh fitur input. Standardisasi adalah teknik transformasi data di mana setiap fitur diubah sehingga memiliki:

- Rata-rata (mean) = 0
- Standar deviasi (std) = 1

  Secara matematis, standardisasi menggunakan rumus:

Rumusnya:

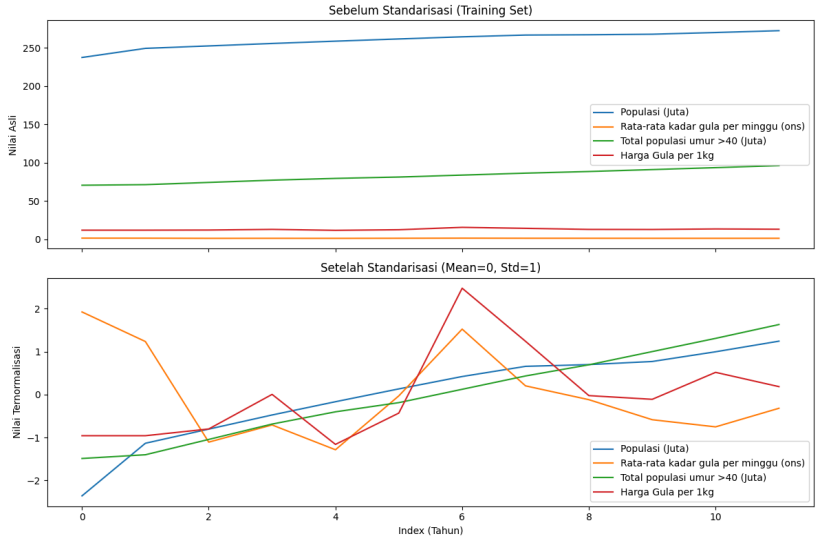
$$X_{\text{scaled}} = \frac{X - \mu}{\sigma}$$

Di mana:

- X = nilai asli
- μ = rata-rata (mean) dari training set
- σ = standar deviasi dari training set

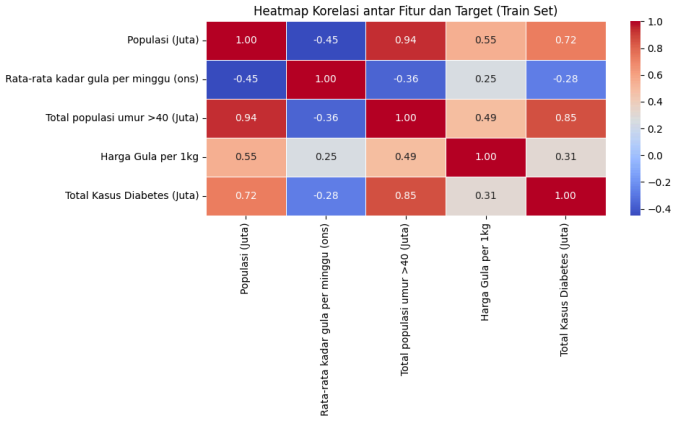
Proses standardisasi ini dilakukan menggunakan fungsi StandardScaler dari pustaka scikit-learn. Setelah standardisasi, semua fitur berada dalam skala yang setara, yang membuat model *neural network*:

- Dapat "mempelajari" seluruh fitur secara adil.
- Tidak terpaku hanya pada fitur yang memiliki angka absolut besar.
- Lebih stabil dan efisien dalam proses pelatihan (konvergensi).



Secara visual, perbedaan antarfitur sebelum dan sesudah standardisasi dapat dilihat pada plot standar *dataset*. Sebelum standardisasi, fitur "Populasi" mendominasi karena nilainya jauh lebih besar, sementara fitur lain nyaris tidak terlihat. Setelah standardisasi, seluruh fitur memiliki distribusi dengan skala yang sama, sehingga semua fitur dapat berkontribusi secara proporsional pada hasil prediksi.

2.4.2 Korelasi Antarfitur



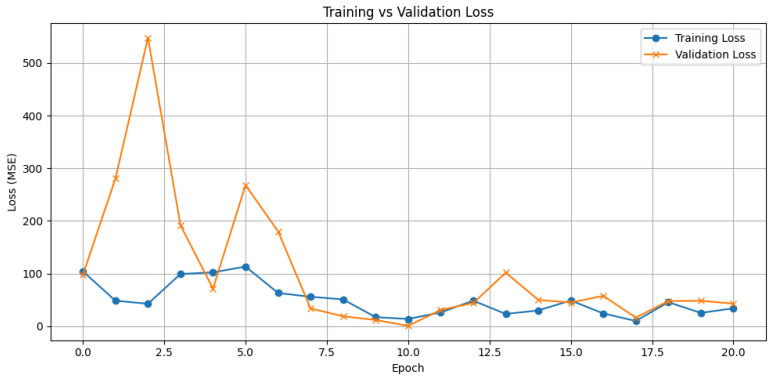
Gambar di atas menunjukkan *heatmap* korelasi antara semua fitur numerik dalam *dataset* terhadap target variabel, yaitu total kasus diabetes. Analisis ini dilakukan pada data latih (*training set*) untuk melihat kekuatan hubungan linier antara variabel. Hasil korelasi mengindikasikan bahwa variabel total populasi umur >40 tahun memiliki korelasi paling tinggi terhadap total kasus diabetes, dengan nilai korelasi sebesar 0.85. Hal ini wajar mengingat diabetes lebih umum terjadi pada kelompok usia lanjut. Selain itu, populasi total juga menunjukkan korelasi yang kuat (0.72), yang berarti pertambahan jumlah penduduk cenderung diikuti oleh meningkatnya jumlah kasus diabetes. Variabel harga gula per 1 kg memiliki korelasi sedang terhadap target (0.31). Ini bisa ditafsirkan sebagai hubungan tidak langsung; harga gula dapat mempengaruhi konsumsi, tetapi bukan satu-satunya faktor penentu jumlah penderita diabetes. Sebaliknya, rata-rata kadar gula per minggu justru memiliki korelasi negatif terhadap jumlah kasus diabetes (−0.28), yang bisa disebabkan oleh ketidaksesuaian skala atau representasi data historis dalam fitur tersebut, atau karena konsumsi tidak secara langsung mencerminkan prevalensi penyakit dalam populasi.

Korelasi antarfitur juga penting untuk menghindari multikolinieritas yang tinggi dalam model prediktif. Terlihat bahwa terdapat korelasi sangat kuat antara populasi total dan populasi umur >40 tahun (0.94), yang menunjukkan adanya potensi redundansi informasi jika keduanya digunakan tanpa regularisasi yang tepat.

Secara keseluruhan, *heatmap* korelasi ini membantu mengidentifikasi fitur-fitur yang paling relevan terhadap target prediksi, serta memberikan gambaran awal mengenai struktur hubungan dalam data sebelum proses pelatihan model dilakukan.

2.4.3 Grafik *Loss* (*Training* vs. *Validation*)

Dalam kasus ini, kami menggunakan *loss function* berupa *mean squared error*. Ditampilkan pada grafik berikut:



1. Pola Awal (*Epoch* 0–5)

   Terlihat *training loss* turun cepat, sedangkan *validation loss* fluktuatif namun masih tinggi. Ini menunjukkan model masih adaptasi awal pada data latih, namun belum mampu generalisasi ke data validasi.

2. Tanda *Overfitting* (Epoch 6–10)

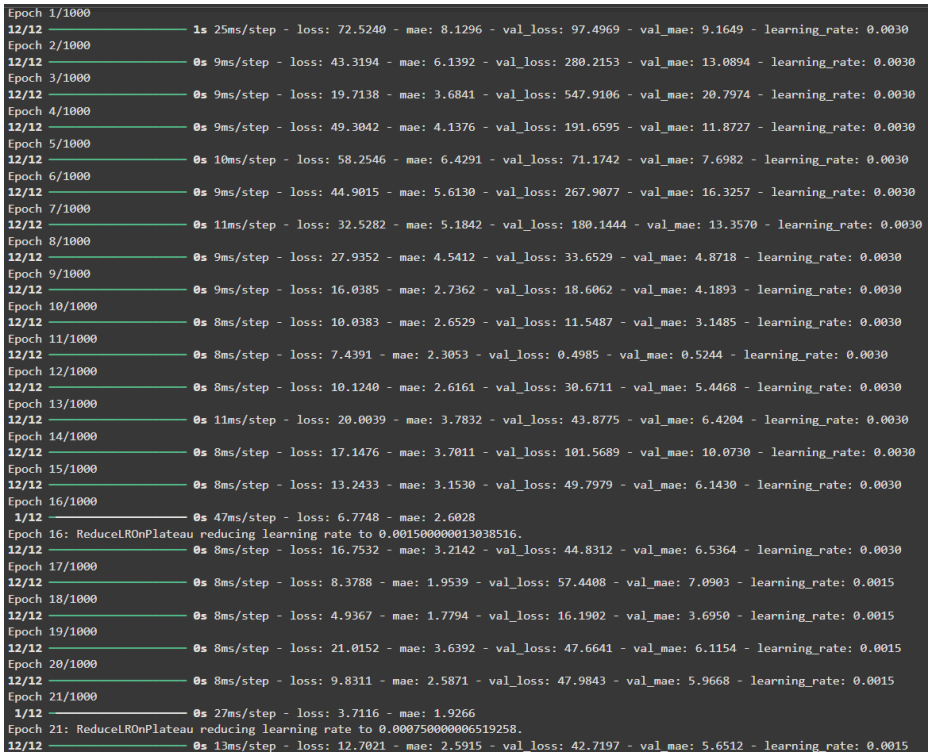
   Training loss terus menurun, namun *validation loss* meningkat tajam di beberapa titik (terlihat *spike* besar). Hal ini menunjukkan model mulai menghafal data *training* (*overfitting*) dan gagal generalisasi ke data baru. Model menjadi sensitif terhadap *noise* pada data latih.

3. Penyesuaian *Learning Rate* (*Epoch* 11–14)

   Setelah ReduceLROnPlateau aktif, *validation loss* mulai menurun kembali. Ini mengindikasikan bahwa *learning rate* yang lebih kecil membantu model keluar dari kondisi stagnan dan memperbaiki performa validasi.

4. Stabilisasi Akhir (*Epoch* 15–20)

   Di akhir, baik *training loss* maupun *validation loss* menjadi lebih stabil. Tidak ada lagi penurunan besar, menandakan model mulai mencapai titik optimal dan *early stopping* akan segera aktif jika tidak ada perbaikan lagi.



Jika melihat hasil per *epoch*, aplikasi ReduceLROnPlateau dapat dijelaskan sebagai berikut:

- Awal *Training* (*Epoch* 1–15)

  Pada tahap awal pelatihan, *learning rate* tetap pada nilai awal yaitu 0.003. Hal ini karena model masih mampu menurunkan val_loss (validation loss) dengan stabil di setiap *epoch*, sehingga tidak ada pengurangan *learning rate* yang diperlukan.

- *Reduce Learning Rate* Pertama (*Epoch* 16)

  Pada *epoch* ke-16, terlihat bahwa val_loss sudah tidak mengalami penurunan signifikan selama beberapa *epoch* sebelumnya (patience=5). Karena itu, ReduceLROnPlateau secara otomatis mengurangi *learning rate* dari 0.003 menjadi 0.0015 (faktor pengurangan = 0.5).

- *Training* Lanjutan (*Epoch* 17–20)

  Setelah pengurangan *learning rate* pertama, model kembali mengalami penurunan val_loss yang lebih baik. Hal ini menunjukkan bahwa *learning rate* yang lebih kecil membantu model untuk melakukan pembelajaran lebih hati-hati dan stabil.

- *Reduce Learning Rate* Kedua (*Epoch* 21)

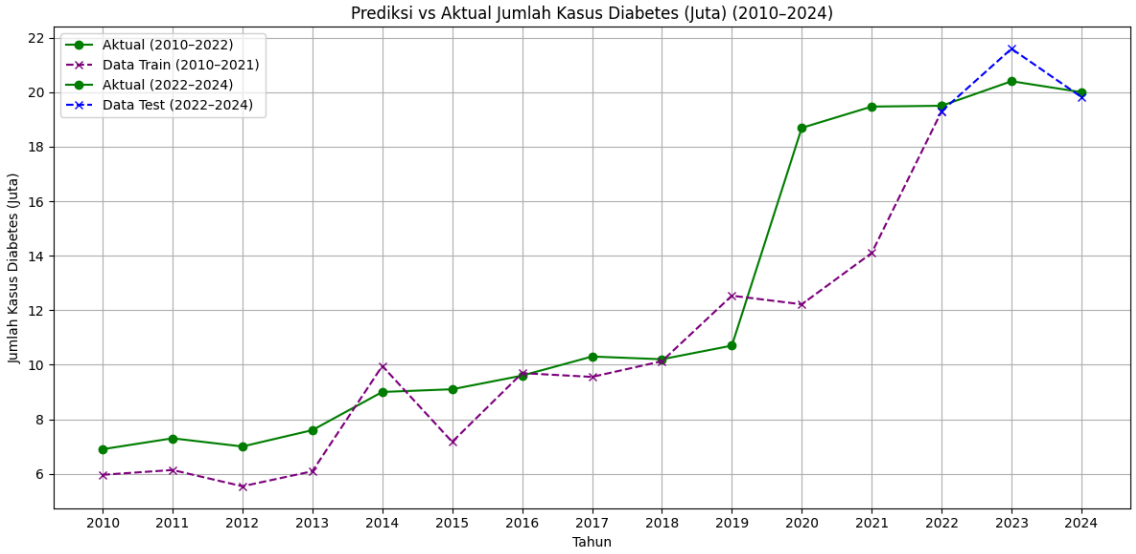
  Meskipun sempat terjadi perbaikan, val_loss kembali stagnan. ReduceLROnPlateau kembali mengurangi *learning rate* menjadi 0.00075, membantu model untuk melakukan penyesuaian bobot secara lebih halus.

- Efek Pengurangan *Learning Rate*

  Dari grafik *training* dan *loss*, terlihat bahwa pengurangan *learning rate* ini membantu model untuk keluar dari area stagnasi dan tetap berusaha mencari solusi terbaik. Langkah-langkah lebih kecil pada learning rate membantu model menghindari "*overshooting*" (melewati titik minimum) dan menemukan solusi yang lebih stabil.

Secara keseluruhan, penggunaan ReduceLROnPlateau ini terbukti bermanfaat dalam mengoptimalkan pembelajaran, khususnya pada tahap-tahap akhir pelatihan. Ini membantu model untuk tetap fokus menyempurnakan prediksi pada data validasi, meskipun sudah mendekati konvergensi.

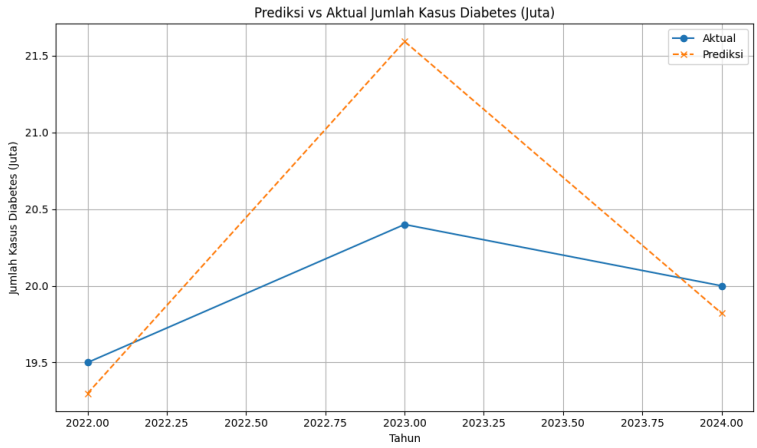
2.4.4 *Output* Model



Grafik di atas menunjukkan perbandingan antara data aktual jumlah kasus diabetes di Indonesia dengan hasil prediksi model *neural network* selama periode 2010 hingga 2024. Dalam grafik tersebut, data aktual ditunjukkan oleh garis hijau solid, sedangkan hasil prediksi pada data latih (2010–2021) ditunjukkan oleh garis ungu putus-putus. Adapun hasil prediksi pada data uji (2022–2024) divisualisasikan dengan garis biru putus-putus bertanda silang.

Secara umum, model berhasil mengikuti pola tren dari data historis dengan cukup baik. Pada data pelatihan (*train*), model mampu merekonstruksi arah pertumbuhan jumlah kasus diabetes dari tahun ke tahun, meskipun terdapat sedikit deviasi pada beberapa titik, seperti tahun

2014 dan 2015. Hal ini menunjukkan bahwa model cukup mampu menangkap hubungan nonlinier dari fitur input terhadap target, meskipun belum sepenuhnya presisi.



Gambar di atas menunjukkan hasil prediksi model *neural network* dibandingkan dengan data aktual jumlah kasus diabetes pada tiga tahun terakhir (2022–2024). Pada grafik tersebut, garis biru merepresentasikan data aktual, sementara garis oranye menampilkan hasil prediksi model. Untuk memperjelas analisis, nilai numerik dari data aktual dan prediksi disajikan kembali dalam bentuk tabel di bawah grafik.

Tahun	Aktual	Prediksi
2022	19.5	19.298351
2023	20.4	21.592810
2024	20.0	19.821127

Model menunjukkan performa yang cukup stabil dalam memprediksi arah tren kenaikan dan penurunan jumlah kasus. Pada tahun 2022, model sedikit *underestimate* (19.30 dibandingkan 19.5 juta), sedangkan pada tahun 2023 terjadi *overestimate* yang paling besar, yaitu sekitar 1.19 juta kasus lebih tinggi dibandingkan aktual. Namun, pada tahun 2024, prediksi kembali mendekati nilai aktual dengan selisih kurang dari 0.2 juta kasus. Dari hasil visualisasi dan tabel tersebut, dapat disimpulkan bahwa model berhasil memetakan tren secara keseluruhan, meskipun masih terdapat deviasi prediksi pada titik-titik tertentu, khususnya pada tahun 2023. Secara umum, model menunjukkan kemampuan generalisasi yang cukup baik, terutama mengingat ukuran *dataset* yang relatif kecil dan keterbatasan fitur yang digunakan.

2.4.5 Evaluasi Model

Evaluasi kinerja model dilakukan menggunakan tiga metrik regresi, yaitu *Mean Squared Error* (*MSE*), *Mean Absolute Error* (*MAE*), dan R^2 *Score* (Koefisien Determinasi). Ketiga metrik ini memberikan gambaran dari sudut pandang yang berbeda mengenai performa model dalam memprediksi jumlah kasus diabetes pada data uji (tahun 2022–2024).

1. *Mean Squared Error* (*MSE*)

Nilai *MSE* pada model ini adalah 0.50. *MSE* mengukur rata-rata kuadrat dari selisih antara nilai aktual dengan nilai prediksi. Karena bersifat kuadrat, metrik ini sensitif terhadap *outlier*. Dalam konteks ini, nilai *MSE* yang cukup rendah mengindikasikan bahwa secara umum kesalahan kuadrat prediksi tidak terlalu besar. Dengan kata lain, model mampu memberikan prediksi yang relatif dekat dengan nilai aktual, meskipun tetap ada perbedaan yang perlu diperhatikan pada titik tertentu (seperti tahun 2023).

2. *Mean Absolute Error* (*MAE*)

Model menghasilkan nilai *MAE* sebesar 0.52, yang berarti secara rata-rata model melakukan kesalahan sebesar 0.52 juta kasus (±520.000 kasus) terhadap nilai aktual. *MAE* lebih mudah diinterpretasikan dibanding *MSE* karena berada dalam satuan yang sama dengan target (jumlah kasus diabetes dalam juta). Dengan *MAE* yang cukup rendah, model dapat dianggap memiliki tingkat akurasi prediksi yang cukup baik secara absolut.

3. *R² Score* (Koefisien Determinasi)

R² Score yang diperoleh sebesar −2.6773, yang menunjukkan bahwa model memiliki kemampuan yang buruk dalam menjelaskan variasi data target. Secara teori, nilai *R²* idealnya mendekati 1 (semakin baik), sedangkan nilai di bawah 0 mengindikasikan bahwa model justru memiliki performa lebih buruk daripada prediksi berbasis rata-rata sederhana. Meskipun error absolut (*MSE* dan *MAE*) rendah, nilai *R²* negatif menandakan bahwa model belum mampu menangkap hubungan antar fitur dan target secara menyeluruh dalam konteks regresi.

Proses didapatnya nilai *R² Score:*

- Langkah 1: Hitung rata-rata dari nilai aktual

$$\bar{y} = \frac{19.5 + 20.4 + 20.0}{3} = \frac{59.9}{3} = 19.9667$$

- Langkah 2: Hitung *Residual Sum of Squares* (SS_res)

$$SS_{res} = \sum (y_i - \hat{y}_i)^2 = (19.5 - 19.298351)^2 + (20.4 - 21.592810)^2 + (20.0 - 19.821127)^2$$
$$= (0.201649)^2 + (-1.192810)^2 + (0.178873)^2 = 0.0407 + 1.4228 + 0.0320 = 1.4955$$

- Langkah 3: Hitung *Total Sum of Squares* (SS_tot)

$$SS_{tot} = \sum (y_i - \bar{y})^2 = (19.5 - 19.9667)^2 + (20.4 - 19.9667)^2 + (20.0 - 19.9667)^2$$
$$= (-0.4667)^2 + (0.4333)^2 + (0.0333)^2 = 0.2178 + 0.1878 + 0.0011 = 0.4067$$

- Langkah 4: Hitung R^2

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{1.4955}{0.4067} = 1 - 3.678 \approx -2.678$$

Berdasarkan ketiga metrik tersebut, dapat disimpulkan bahwa Model memiliki akurasi numerik yang cukup baik, karena nilai *MAE* dan *MSE* tergolong kecil. Namun, kemampuan generalisasi model terhadap variasi data masih lemah, sebagaimana ditunjukkan oleh nilai R^2 yang negatif. Hal ini menunjukkan bahwa meskipun model cukup baik dalam menebak nilai rata-rata, ia belum berhasil menangkap kompleksitas hubungan antar fitur terhadap target secara menyeluruh.

3. PENUTUP

3.1 Kesimpulan

Penelitian ini bertujuan untuk membangun model prediksi jumlah kasus diabetes di Indonesia menggunakan pendekatan *Neural Network* berbasis data historis tahun 2010 hingga 2024. Model dilatih menggunakan data dari tahun 2010–2021 dan diuji pada tahun 2022–2024. Berdasarkan hasil implementasi dan evaluasi, dapat diambil beberapa kesimpulan berikut:

Model menunjukkan kemampuan yang cukup baik dalam memetakan tren kenaikan dan penurunan jumlah kasus diabetes, khususnya pada data uji. Meskipun prediksi pada tahun 2023 menunjukkan deviasi terbesar (*overestimate* sekitar 1.19 juta kasus), prediksi pada tahun 2022 dan 2024 relatif mendekati nilai aktual, dengan rata-rata meleset sekitar 0.2 juta kasus. Hal ini tercermin dari nilai *Mean Absolute Error* (*MAE*) sebesar 0.52 juta dan *Mean Squared Error* (*MSE*) sebesar 0.50 juta.

Namun demikian, nilai R^2 Score yang negatif (−2.68) menunjukkan bahwa model belum mampu menjelaskan variasi target secara optimal. Dengan kata lain, meskipun eror absolutnya kecil, model belum sepenuhnya mampu menangkap hubungan kompleks antarfitur dan target dalam konteks regresi.

Dari sisi kekuatan, model ini cukup baik dalam menangkap arah tren makro, mudah diimplementasikan, serta responsif terhadap pola input. Namun, kelemahannya terletak pada keterbatasan ukuran *dataset*, potensi *overfitting*, dan sensitivitas terhadap fitur tertentu yang belum cukup menjelaskan variasi target secara menyeluruh.

3.2 Saran

- Jumlah data yang digunakan dalam penelitian ini masih terbatas (15 tahun). Untuk meningkatkan performa model, disarankan penggunaan data yang lebih panjang dan bervariasi.
- Penambahan fitur lain, seperti aktivitas fisik, indeks massa tubuh (BMI), atau prevalensi penyakit lain yang berkaitan dapat meningkatkan akurasi prediksi.
- Perlu eksplorasi terhadap arsitektur model lain (seperti *LSTM* atau R*andom Forest*) untuk membandingkan performa prediksi secara menyeluruh.

DAFTAR PUSTAKA

1. Badan Pusat Statistik. (2024). *Jumlah Penduduk Menurut Kelompok Umur dan Jenis Kelamin, 2024*. https://www.bps.go.id/id/statistics-table/3/WVc0MGEyMXBkVFUxY25KeE9HdDZkbTQzWkVkb1p6MDkjMw==/jumlah-penduduk-menurut-kelompok-umur-dan-jenis-kelamin--2023.html?year=2024
2. Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.
3. G´eron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor-Flow*. O'Reilly Media.
4. International Diabetes Federation. (2025). *Indonesia Diabetes Trends & Prevalence*. https://diabetesatlas.org/data-by-location/country/indonesia/
5. Kementerian Kesehatan Republik Indonesia. (2023). *Profil Kesehatan Indonesia 2022*. https://www.kemkes.go.id
6. Pedregosa, F., *et al*. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
7. Smith, L. N. (2017). *Cyclical Learning Rates for Training Neural Networks*. IEEE Winter Conference on Applications of Computer Vision (WACV)

LAMPIRAN PROGRAM DAN *DATASET*

https://drive.google.com/drive/folders/1WjSjOQlujvVKSHBmrmTp0QWc1OenwJQf?usp=sharing