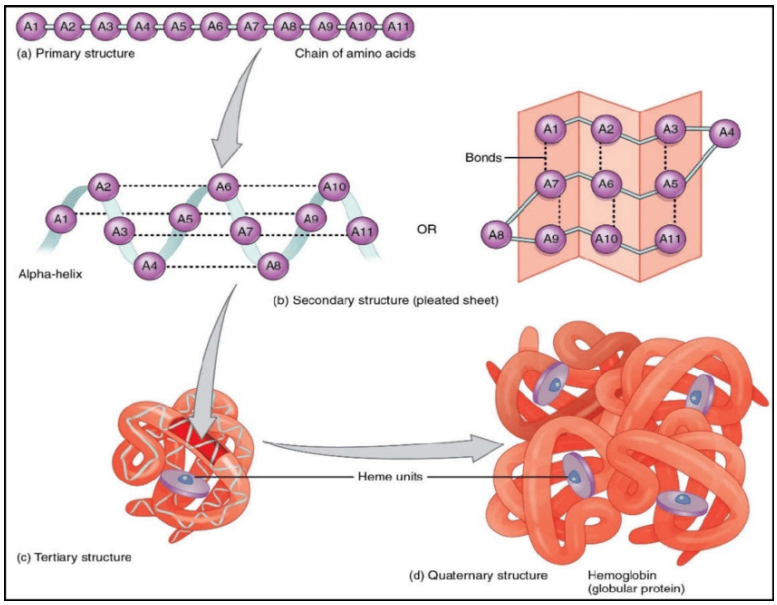
LAPORAN KECERDASAN BUATAN BIOMEDIS
PREDIKSI 3-STATE STRUKTUR SEKUNDER PROTEIN MENGGUNAKAN BI-LSTM DAN AUTOGLUON

Oleh :
Nama : Kayla Queenazima Santoso
NIM : 20/456076/TK/50206
Prodi : Teknik Biomedis

PROGRAM STUDI TEKNIK BIOMEDIS
FAKULTAS TEKNIK
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2023

Latar Belakang

Protein berperan penting bagi kehidupan, dari perbaikan DNA hingga katalisator enzim. Protein berfungsi untuk membangun dan memperbaiki sel dalam tubuh serta produksi energi. Protein memiliki struktur kompleks yang terdiri dari rangkaian asam amino. Struktur protein terdiri dari 4 struktur utama yaitu struktur primer (rangkaian asam amino dari ikatan peptida), struktur sekunder (rangkaian asam amino membentuk struktur melingkar), struktur tersier (penggabungan hasil pelipatan beberapa struktur sekunder berbeda), dan *quarternary*. Struktur sekunder protein merupakan salah satu bagian penting dalam bioinformatika sebagai langkah awal prediksi struktur tersier protein. Melalui prediksi struktur sekunder protein, dapat diketahui aktivitas protein, relasi, dan fungsi protein.



Gambar 1. Struktur Protein [1]

Struktur sekunder protein dapat dikalkulasi berdasarkan koordinat 3D atom struktur tersier protein. Struktur tersier protein didapatkan melalui X-Ray kristalografi atau NMR [2]. Dalam penentuan struktur sekunder suatu protein, digunakan DSSP (Dictionary of Secondary Structure) yang merangkum tipe-tipe struktur sekunder terdiri dari 8-state dan 3-state. Algoritma DSSP mengolah struktur tersier menjadi beberapa pengelompokan struktur sekunder dengan diketahui pula struktur primer protein tersebut. Namun, pengambilan sampel dengan X-Ray kristalografi dan NMR cukup mahal. Oleh karena itu, diperlukan prediksi struktur protein yang lebih komprehensif dan terjangkau.

Melalui data tersier yang ada kemudian didapatkan data struktur sekunder protein melalui DSSP. Data struktur sekunder protein tersebut lebih dekat pada struktur primer protein sehingga melalui data yang ada, dapat diprediksi lebih banyak struktur sekunder protein dari struktur primer protein.

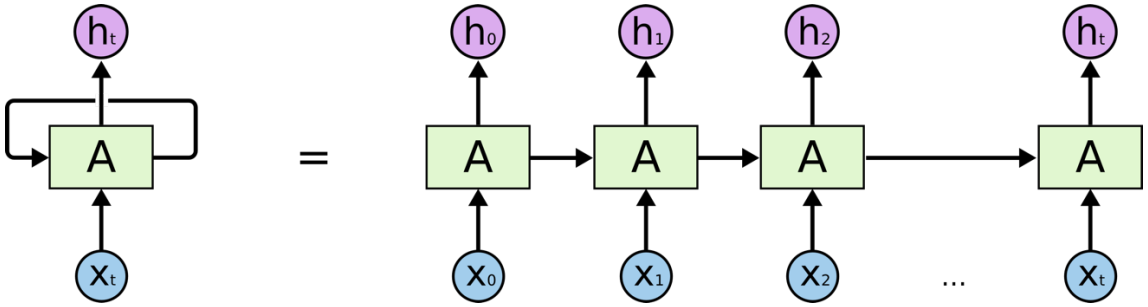
Dengan memprediksi struktur sekunder protein, didapat prediksi struktur tersier yang lebih baik tanpa perlu menggunakan instrumen X-Ray kristalografi maupun NMR.

Algoritma DSSP [3] bekerja dengan menghitung struktur sekunder paling memungkinkan apabila terdapat suatu struktur tersier protein dengan mengetahui posisi atom-atom pada protein diikuti kalkulasi ikatan Hidrogen antar semua atom. Algoritma mendiskualifikasi Hidrogen pada struktur primer yang masuk kemudian didapat posisi Hidrogen paling optimal dengan meletakkannya pada 1000 Avogadro dari *backbone* Nitrogen pada arah berlawanan terhadap *backbone* ikatan ganda karbon-oksigen.

Teknik yang Digunakan

1. Bi-LSTM

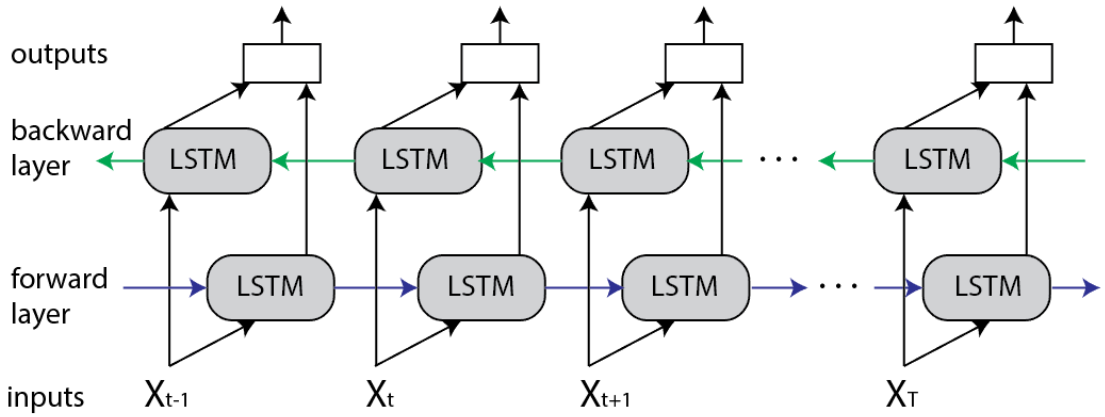
   Bi-LSTM (Bidirectional Long Short Term Memory) [4] merupakan pengembangan arsitektur RNN (Recurrent Neural Network). Jaringan RNN mencoba menyelesaikan tantangan untuk memahami konteks kata berdasarkan kata/sekuens sebelumnya. Hal ini diselesaikan dengan cara memberikan output dari satu layer ke layer itu sendiri (feedback) sehingga mengulang operasi layer tersebut beberapa kali. Hal ini menyebabkan informasi dari data yang masuk akan terus bertahan di seluruh model. Jika proses feedback dibuka, pada unrolled RNN dapat dilihat bahwa tiap sek pada RNN memiliki 2 input yaitu past (layer sebelumnya) dan present (X). Namun, terdapat gap informasi yang bisa terjadi dengan penggunaan network RNN. Gap informasi tersebut terjadi disebabkan masalah gradien yang terus mengalami loop sehingga menjadi sangat kecil (hilang) atau sangat besar (meledak).



Gambar 2. Skema Arsitektur RNN [4]

Dalam fungsi LSTM, terdapat penambahan cell state untuk meneruskan/menghentikan informasi dengan regulator dari berbagai gate. Gate pada LSTM terdiri dari 3 yaitu input gate yang menentukan seberapa banyak informasi baru yang masuk, output gate yang menghapus informasi lama menjadi baru, dan forget gate yang menentukan seberapa banyak informasi lama ingin dihapus. Melalui regulator gate tersebut, informasi yang dianggap kurang bernilai akan dihapus dengan meregulasi alur informasi pada cell state sehingga menyelesaikan masalah gradien yang sangat kecil atau sangat besar.

Selain memahami konteks sekuens dari depan, konteks sekuens dari belakang juga dapat menghasilkan informasi yang penting. Bi-directional LSTM menghadirkan ide tersebut dengan menggunakan 2 layer LSTM dengan arah berlawanan satu sama lain dan data masuk pada tiap layer. Namun, penggunaan Bi-LSTM membutuhkan komputasi yang lebih besar daripada LSTM disebabkan penggunaan 2 layer LSTM. Dalam prediksi struktur sekunder protein, terdapat beberapa konteks yang ingin ditangkap [5] yaitu konteks lokal berupa asam amino yang berdekatan serta konteks jarak jauh berupa interaksi asam amino yang bisa berinteraksi dan berpengaruh satu sama lain dalam struktur protein 3 dimensi (indeks jauh tetapi dalam model 3 dimensi berdekatan). Kedua hal ini dapat diselesaikan dengan LSTM sebab kemampuannya dalam *long term dependencies*. Penggunaan Bi-LSTM memperkuat konsep tersebut dengan mempelajari sekuens primer protein dari depan maupun belakang.



Gambar 3. Skema Arsitektur Bi-LSTM [4]

2. AutoGluon

AutoGluon [6] merupakan *library* untuk *automated machine learning* (AutoML) yang dikembangkan oleh tim dari AWS. AutoGluon memungkinkan kita untuk mencoba berbagai model machine learning maupun deep-learning populer. AutoGluon menggunakan strategi multi-layer stack ensembling untuk menggabungkan model yang telah dilatih. Multi-Layer Stack Ensembling akan menggabungkan output base model yang dilatih pada layer pertama dan akan dijadikan input pada stack model di layer berikutnya. Stack model kemudian akan dilatih lagi dan outputnya akan dilakukan agregasi menggunakan weighting untuk menghasilkan output final model pada layer kedua. AutoGluon juga menggunakan teknik k-fold bagging untuk memastikan tidak terjadi overfitting. Berikut berbagai base model machine learning yang dapat secara otomatis dilatih menggunakan AutoGluon untuk task klasifikasi: NeuralNetFastAI, NeuralNetTorch, CatBoost, LightGBM, LightGBMLarge, LightGBMXT, XGBoost, ExtraTreesMSE, dan RandomForestMSE.

3. LightBM
   LightGBM [7] adalah kerangka kerja peningkatan gradien yang menggunakan algoritma pembelajaran berbasis pohon. Algoritma ini menggunakan dua teknik baru: Gradient-based One Side Sampling (GOSS) dan Exclusive Feature Bundling (EFB) untuk memenuhi keterbatasan algoritma berbasis histogram yang terutama digunakan di semua kerangka kerja GBDT (Gradient Boosting Decision Tree)

4. Random Forest
   Random Forest adalah metode pembelajaran ensemble yang menggunakan pohon keputusan. Metode ini menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi dan mengurangi overfitting.

5. Support Vector Machine Classifier
   Support Vector Machine Classifier (SVC) bekerja dengan menemukan hyperplane yang paling baik memisahkan data ke dalam kelas-kelas yang berbeda.

6. Optuna Hyperparameter Optimization
   Optuna [8] adalah kerangka kerja pengoptimalan hyperparameter yang dirancang khusus untuk machine learning. Ini adalah framework optimalisasi hyperparameter otomatis yang memiliki fitur API pengguna yang sangat penting dan bergaya define-by-run. Optuna adalah kerangka kerja yang agnostik, yang berarti dapat digunakan dengan kerangka kerja pembelajaran mesin atau pembelajaran mendalam apa pun. Optuna memiliki kelebihan berupa paralelisasi pencarian hyperparameter pada beberapa thread atau proses tanpa memodifikasi kode.

Bagaimana Data Diambil

Dataset yang digunakan diakses melalui Kaggle (https://www.kaggle.com/datasets/alfrandom/protein-secondary-structure/data) yang merupakan data diambil dari RCSB (Research Collaboratory for Structural Bioinformatics) PDB (Protein Data Bank) dengan update yang digunakan per tahun 2018. Data dikurasi dan anotasi berdasarkan standar oleh Worldwide Protein Data Bank (wwPDB). Data primer, sekunder, maupun tersier dapat diperoleh melalui beberapa alat seperti X-Ray kristalografi, Nucleur Magnetic Resonance (NMR), dan mikroskop elektron (EM). Berikut daftar kolom pada dataset yang digunakan:

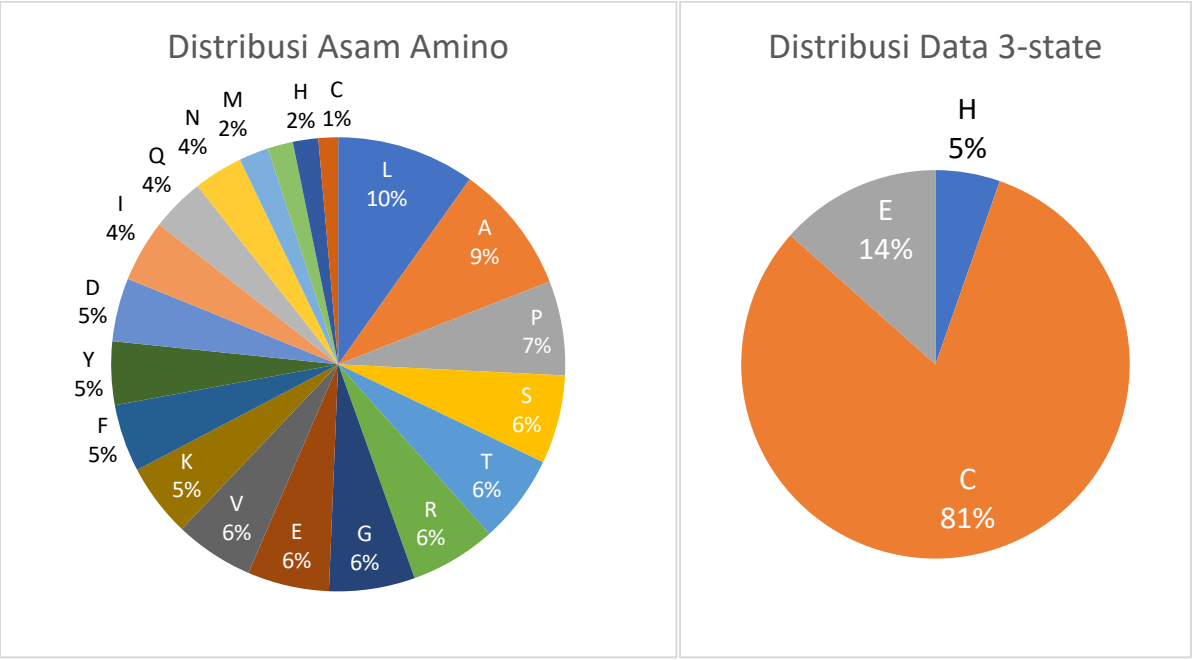
1. pdb_id: id yang digunakan pada https://www.rcsb.org/
2. chain_code: Ketika suatu protein terdiri dari beberapa rantai peptida, kode rantai diperlukan untuk mengetahui rantai spesifik.
3. seq: sekuens dari peptida .
4. sst8: struktur sekunder eight-state (Q8).
5. sst3: struktur sekunder three-state (Q3).
6. len: panjang peptida.
7. has_nonstd_aa: apakah peptida memiliki amino non-standar (B, O, U, X, Z).

Tabel 1. Tabel Reduksi State Protein [9]

Simbol 8-state	Definisi	Simbol 3-state
C	Loop dan elemen tak beraturan (blank character hasil DSSP)	C
E	β-strand	E
H	α-helix	H
B	β-bridge	C
G	3-helix	H
I	π-helix	H
T	Turn	C
S	Bend	C

Bagaimana Data Diolah

Data yang diolah merupakan 5001 data pertama disebabkan keterbatasan komputasi yang ada dengan data representatif menunjukkan imbalance yang signifikan, hal yang juga terdapat pada data sumbernya. Distribusi asam amino pada struktur primer dan 3-state pada struktur sekunder protein sesuai Gambar x.



Gambar 4. Distribusi Asam Amino dan Struktur 3-State

1. Skema Machine Learning (AutoGluon)
   - Filter data tidak mengandung asam amino non-standar

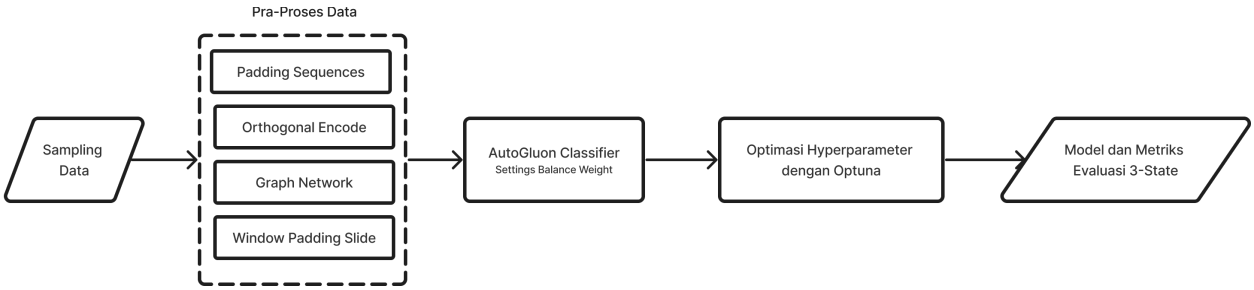
Data yang masuk kemudian disaring untuk menghilangkan sekues yang mengandung asam amino non-standar dengan menyaring data dengan nilai True pada kolom has_nonstd_aa.

- Assign struktur primer menjadi sekuens input dan struktur sekunder sebagai target
  Struktur primer dan struktur sekunder dipisah menjadi 2 list berbeda untuk memudahkan pengolahan.
- Memecah tiap row dengan sekuens lebih dari 128 menjadi beberapa row
  Tiap list tersebut kemudian dipotong-potong kembali berdasarkan panjang sekuens maksimal 128 pada skema ini.
- Filter keluar anomali apakah terdapat panjang struktur primer yang tidak sama dengan panjang struktur sekunder
  Untuk menghindari error dan ketidaktelitian pra-proses sebelumnya, dicek kembali panjang tiap sekuens primer terhadap sekunder.
- Orthogonal encode
  Mengubah struktur primer menjadi array berbentuk matriks orthogonal one-hot encode 20x20 sementara struktur sekunder menjadi array 1x3 (label encode).
- GNN (Graph Neural Network)
  GNN diaplikasikan dengan edge berupa interaksi antar nodes yang merupakan asam amino. Dilakukan dengan menambahkan 2-3 node asam amino struktur primer dengan node di pinggir array menggunakan penambahan 2 node, sementara lainnya penambahan 3 node.
- Window Padding Slide
  Windows padding slide diawali dengan proses padding yaitu menambahkan sekuens dengan padding zeros kemudian diekstrak dengan sliding window sesuai ukuran window yaitu 11.

2. Skema Deep Learning (Bi-LSTM)
   - Filter data tidak mengandung non-standar asam amino dengan panjang kurang dari sama dengan 500 sekuens
   - Tokenisasi sekuens dengan N-Grams dengan n = 3 dengan tokenizer menggunakan library Keras
   - Padding post sekuens

Pipeline

1. Skema Machine Learning (AutoGluon)



Gambar 5. Skema Machine Learning

Pipeline pertama dengan input berupa sampling data sebanyak 5001 row kemudian diolah melalui pra-proses data sehingga didapatkan struktur primer dalam bentuk orthogonal dan struktur sekunder dalam bentuk matriks label encoded. Data yang telah diproses kemudian menjadi input bagi framework AutoGluon untuk komparasi model machine learning yang paling bagus dalam mengklasifikasikan tiap sekuens struktur primer. AutoGluon dilatih menggunakan CPU berbasis Kaggle tanpa ada batasan waktu dengan setting balance weights sebagai mitigasi terhadap imbalance data pada label struktur sekunder. Dari proses latih menggunakan AutoGluon, didapatkan metriks F1 score macro, inference time model, dan fit time model (Gambar x). Beberapa model pada leaderboard kemudian diolah pada Optuna untuk mengoptimasi hyperparameter tiap model. Model yang dioptimasi pada Optuna yaitu Support Vector Machine, Random Forest, dan LightBM dengan trial 5. Selain itu, diujicoba beberapa hyperparameter yaitu nilai C pada SVC, nilai num_leaves pada LightBM, dan max_depth pada random forest. Optuna mengambil kombinasi secara acak untuk diujicoba dalam 5 trials.

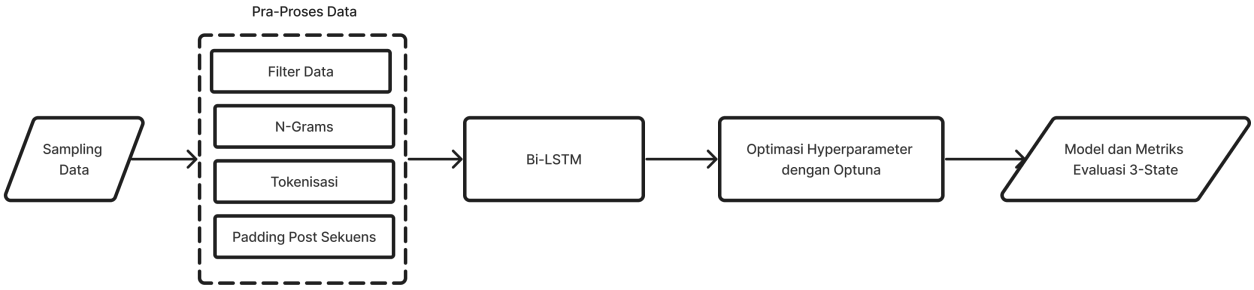
	model	score_val	pred_time_val	fit_time
0	WeightedEnsemble_L2	0.889473	1.608875	211.016472
1	NeuralNetFastAI	0.881319	0.081677	43.426268
2	NeuralNetTorch	0.870685	0.216227	86.952941
3	KNeighborsDist	0.869064	0.313376	0.204979
4	LightGBM	0.867140	0.423944	20.133467
5	LightGBMXT	0.862219	0.625274	28.482292
6	LightGBMLarge	0.860799	1.258068	54.802163
7	XGBoost	0.857574	0.575089	59.884746
8	CatBoost	0.853067	0.067452	511.485576
9	ExtraTreesGini	0.851756	0.209334	7.868995
10	ExtraTreesEntr	0.851091	0.207675	8.124666
11	RandomForestGini	0.851004	0.213061	9.012483
12	RandomForestEntr	0.850823	0.210710	9.862044
13	KNeighborsUnif	0.844011	0.350266	2.761729

Gambar 6. Leaderboard F1 Score Macro AutoGluon

Gambar 7. History Plot tiap Trials dan Slice Plot dari Kombinasi Optuna

Berdasarkan hasil pada AutoGluon (Gambar 6), didapatkan model kombinasi memiliki metriks F1 score macro paling tinggi (88.9%) namun dengan waktu prediksi 20x model maupun durasi latih yang 5x model NeuralNet FastAI. Hal ini menunjukkan bahwa model kombinasi atau ensemble memiliki keuntungan berupa kombinasi dari beberapa model dengan weight masing-masing yaitu 'KNeighborsDist': 0.07692307692307693, 'NeuralNetFastAI': 0.23076923076923078, 'RandomForestEntr': 0.07692307692307693, 'ExtraTreesGini': 0.07692307692307693, 'XGBoost': 0.15384615384615385, dan 'NeuralNetTorch': 0.38461538461538464. Berbagai kombinasi model baik model neural network maupun berbasis decison tree dikombinasikan sehingga meningkatkan F1 skor. Namun, menjadi wajar apabil waktu inferensi dan waktu latih signifikan melonjak disebabkan kombinasi beberapa model sekaligus. Hal ini dapat menjadi pertimbangan dalam pengembangan selanjutnya untuk memahami use case yang dibutuhkan seperti kecepatan prediksi yang diinginkan sangat prediksi struktur sekunder protein. Sementara, hasil dari optimasi hyperparameter (Gambar 7) justru stagnan pada metriks 83.5% yang justru lebih rendah dari ujicoba AutoGluon (Gambar 6) sehingga direkomendasikan untuk mengujicoba parameter pada skema ensemble dengan trials yang lebih besar.

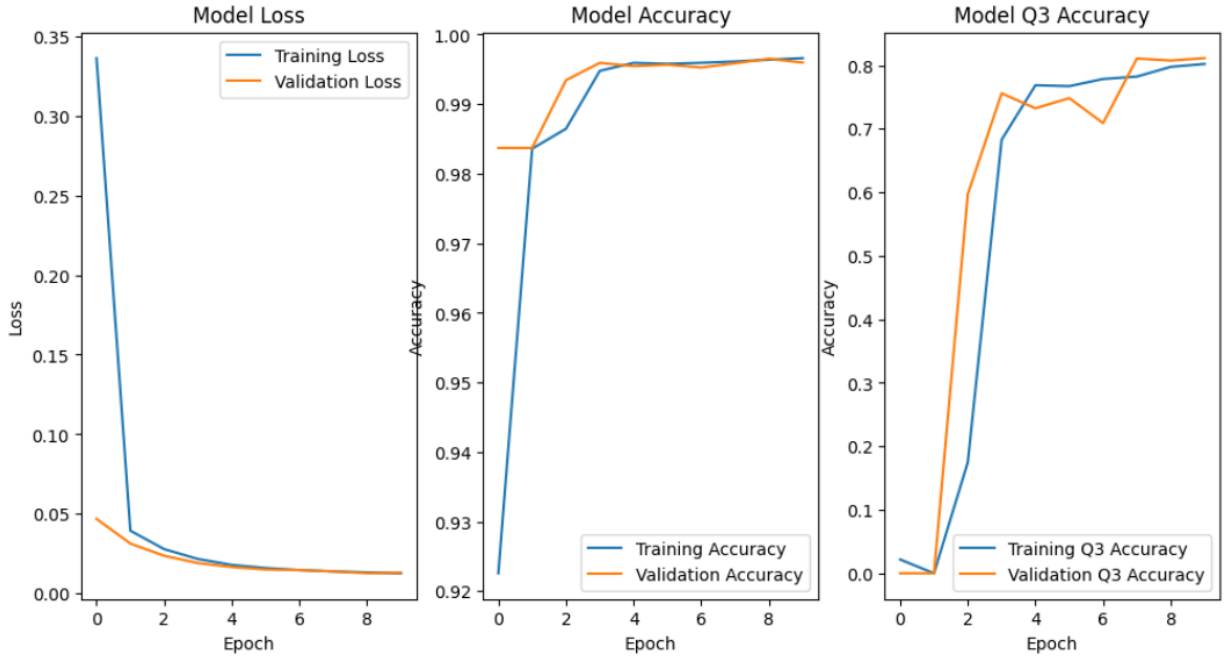
2. Skema Deep Learning (Bi-LSTM)



Gambar 8. Skema Deep Learning

Skema 2 menggunakan sampling data sebanyak 5001 row data kemudian dilakukan pengolahan pra-proses data sehingga didapatkan hasil pad tokenisasi dengan tiap pad sepanjang 500 sekuens.

Data hasil pra-proses dilatih pada CPU Kaggle dengan library Keras. Dengan dropout rate sebesar 0.1 dan fungsi aktivasi softmax, didapatkan data seperti pada Gambar 9. Gambar 9 menunjukkan penurunan loss signifikan pada epoch pertama diikuti peningkatan akurasi dan akurasi Q3-state yang meningkat. Selisih train dengan validasi juga tidak signifikan berbeda. Hal ini menunjukkan proses latih sudah fit dan tidak mengalami underfit maupun overfit. Bi-LSTM juga memiliki kelebihan dengan dapat memahami konteks lokal maupun jarak jauh dengan baik dalam waktu singkat (iterasi kecil). Pada iterasi terakhir, didapatkan performa loss validasi 0.0131, akurasi validasi 0.9960, dan akurasi Q3 state validasi 0.8114.



Gambar 9. Hasil Latih pada 10 Epoch dengan Bi-LSTM

Future Development

Terdapat beberapa potensi yang dikembangkan yaitu evaluasi dengan metriks yang sama antara machine learning dan deep learning. Selain itu, dapat juga dikembangkan dengan membandingkan penggunaan pra-proses data yang sama pada satu algoritma untuk mengetahui efektivitas proses pengolahan data yang dilakukan.

Daftar Pustaka

[1] Alhalmi, Abdulsalam & Ali, Nafaa & Abdulrahman, Amer. (2020). Intracellular Protein Biosynthesis: A Review. Asian Journal of Biochemistry Genetics and Molecular Biology. 2. 10-18. 10.9734/AJBGMB/2020/v5i230125.

[2] https://www.kaggle.com/datasets/alfrandom/protein-secondary-structure/data

[3] https://2struc.cryst.bbk.ac.uk/about/

[4] https://dagshub.com/blog/rnn-lstm-bidirectional-lstm/

[5] H. Hu, Z. Li, A. Elofsson, and S. Xie, “A Bi-LSTM Based Ensemble Algorithm for Prediction of Protein Secondary Structure,” Applied Sciences, vol. 9, no. 17, p. 3538, Aug. 2019, doi: 10.3390/app9173538.
[6] Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., & Smola, A. (2020). AutoGluon-Tabular: Robust and accurate AutoML for structured data. arXiv preprint arXiv:2003.06505.
[7] https://lightgbm.readthedocs.io/en/latest/index.html
[8] https://optuna.org
[9] Singh, Manpreet et al. “Protein Secondary Structure Prediction.” World Academy of Science, Engineering and Technology, International Journal of Biological, Biomolecular, Agricultural, Food and Biotechnological Engineering 2 (2008): 108-111.