



**PROPOSAL
KUALIFIKASI**

**Prediksi Aliran Turbulen Disekitar Airfoil Dengan
*GCNN Graph Convolutional Neural Networks***

Promotor :

Prof. Dr. Ir. Bambang Suryawan, MT.

Disusun Oleh

IRVAN SEPTYAN MULYANA

99218024

**PROGRAM DOKTOR TEKNOLOGI INFORMASI
UNIVERSITAS GUNADARMA
DESEMBER 2021**

DAFTAR PUSTAKA

DAFTAR PUSTAKA	2
BAB I Pendahuluan	3
1.1 Latar belakang	3
1.2 Rumusan Masalah	7
1.3 Batasan Masalah.....	7
1.4 Tujuan.....	8
1.5. Manfaat Penelitian dan Keterbaruan	8
BAB II Tinjauan Pustaka	9
2.1 Airfoil	9
2.2 Model Turbulen.....	9
2.2.1 Model (LES) <i>Large eddy Simulations</i>	15
2.2.2 Model (PDF) <i>Probability Density Function</i>	15
2.2.3 Perbandingan Model RANS dan Model LES.....	16
2.3 <i>Deep Learning</i>	17
2.4 <i>Convolutional Neural Networks</i> (CNN).....	17
2.5 <i>Long Short Term Memory</i> (LSTM).....	18
2.6 <i>Generative Adversarial Networks</i>	19
2.7 Perbandingan CNN,LSTM, GAN dalam bidang dinamika fluida	19
2.8 GNN Graph Convolutional Neural Networks	20
2.9 Perbandingan Tinjauan dari Penelitian Sebelumnya.....	24
BAB III Metode Penelitian	26
3.1 Tahapan Konsep Penelitian	26
3.2 Pengumpulan Data	27
3.3 Airfoil NACA 0012.....	27
3.4 Model CFD.....	28
3.5 <i>Graph Convolutional Neural Networks</i> GCNN	29
3.6 Optimasi Parameter	30
3.7 Road Map Penelitian	30
DAFTAR PUSTAKA	31
LAMPIRAN	34

BAB I

Pendahuluan

1.1 Latar belakang

Selama beberapa tahun terakhir, peneliti dinamika fluida komputasi (CFD) sebagian besar telah diuntungkan dari perkembangan pesat bidang machine learning (ML), dan lebih khusus lagi dari domain jaringan saraf (NN). Dalam banyak kasus, model pengganti berbasis data dibangun dengan melatih jaringan saraf pada tugas prediksi tertentu, untuk melawan biaya komputasi yang tinggi dari pemecah CFD. Dalam konteks ini, jaringan saraf convolutional (CNNs) telah banyak dieksploitasi. Beberapa peneliti menggunakan CNN yang mewakili penelitian bidang aliran dalam prediksi aliran stabil atau tidak stabil.

Alih-alih berusaha memperoleh solusi secara detail dari aliran turbulen, para peneliti dan engineer memiliki ide yang lebih cerdas, yaitu “mengelompokkan” aliran-aliran turbulen yang terjadi menjadi satu paket yang dapat diselesaikan secara matematis ataupun numerik, metode ini dikenal juga dengan permodelan turbulen (*turbulence modelling*), yang tentu saja akan bervariasi berdasarkan karakteristik aliran, geometri, reynold number dan lain-lain sehingga pemilihan model turbulen yang tepat sangatlah penting dalam analisis aliran fluida, biasanya hal ini dilakukan untuk analisis menggunakan metode *Computational Fluid Dynamics* (CFD).

Salah satu penggunaan komputasi numeric turbulen yang telah dilakukan membahas mengenai *Neural Networks* dalam bidang mekanika fluida yang dijelaskan oleh (Ling, Templeton, and Kurzawski n.d.) menggunakan dua metode Tensor Basis Neural Network (TBNN) dan *Linear Eddy Viscosity Model* (LEVM) untuk mempelajari model untuk Reynolds stres anisotropy tensor dari simulasi tingkat ketelitian tinggi Sebuah arsitektur *Neural Networks* baru diusulkan yang menggunakan lapisan perkalian dengan dasar tensor invarian untuk menambahkan invariansi Galilea ke dalam prediksi tensor anisotropi. Prediksi anisotropi TBNN terbukti secara signifikan lebih akurat dibandingkan dengan model LEVM dan model viskositas eddy non-linier. Dalam kasus uji turbulensi menunjukkan bahwa TBNN mempelajari tentang aliran yang mendasari dan memiliki kemampuan

untuk melakukan ekstrapolasi ke kasus aliran baru. TBNN perlu dilatih dan diuji di berbagai aliran yang lebih luas. Pelatihan lebih banyak aliran akan meningkatkan akurasi jaringan saraf, dan pengujian di lebih banyak aliran akan memungkinkan analisis ketidakpastian untuk prediksi TBNN. Dalam penelitian yang dilakukan oleh memiliki kesulitan untuk melatih jaringan saraf dan dibutuhkan data variasi data yang cukup banyak dan apakah mampu memprediksi diluar data pelatihan.

Penelitian yang dikembangkan oleh (Gaitonde 2018) *Reduced Order Modeling* (ROM) untuk kontrol aliran menggunakan *Neural Networks Long Short Term Memory* (LSTM) dengan metode tersebut prediksi aliran turbulen tanpa menghitung persamaan *Navier-Stokes* (NS) penuh. Dilakukan dengan memproyeksikan dinamika dimensi tinggi ke subruang berdimensi rendah, data yang digunakan dua database *Direct Numerical Simulation* (DNS) dari Johns Hopkins turbulence database (JHTB) dan Dataset *Forced Isotropic Turbulence* (ISO) bersumber dari 3-D DNS *Navier Stok*. Metode *deep learning* (ROM-LSTM) menghasilkan prediksi aliran turbulen. Dalam penelitian yang dilakukan oleh (Gaitonde, 2018) memiliki kekurangan yang sama yang dilakukan oleh (Ling, Templeton and Kurzawski, 2016) yaitu melatih kumpulan data yang berada di luar data peatihan.

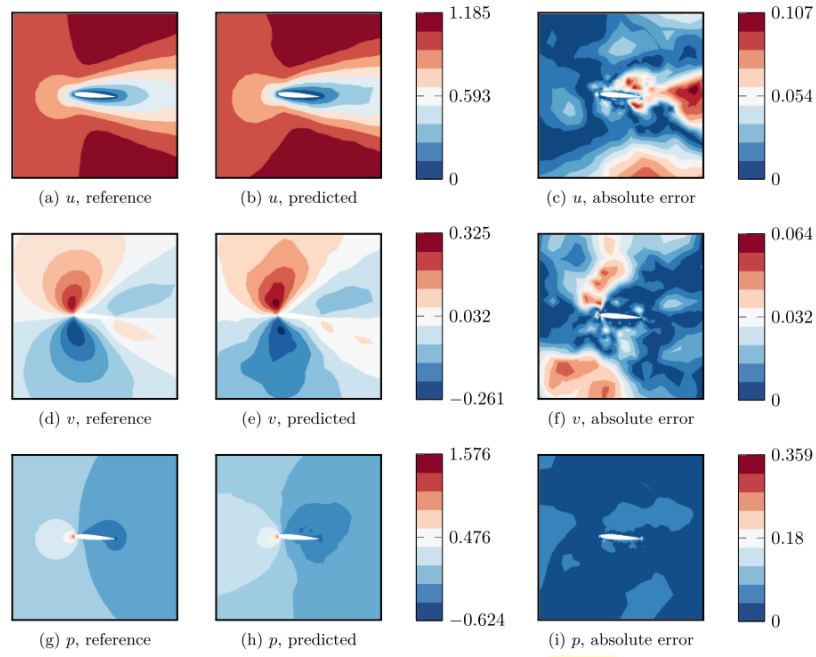
Model turbulensi RANS dengan *One-Equation Models Spalart Allmaras* (SA) juga digunakan untuk memprediksi aliran turbulen yang dijelaskan oleh (Piotrowski 2020) dalam penelitiannya memprediksi aliran turbulen pada Airfoil S809 dan NLF0416, dengan banyak model turbulen atau mengabungkan model turbulen yang ada dengan memodifikasi algoritma untuk memperbaiki pemodelan prediksi turbulen.

Menganalisis kekurangan dari *computational fluid dynamics* (CFD) and *traditional reduced-order models* (ROMs). Selanjutnya, untuk mengklasifikasikan studi berbasis *neural networks* dalam pemodelan data turbulen dan menggambarkan perbandingan pada penelitian yang sudah ada . Menunjukkan bahwa *neural networks* adalah pendekatan yang efektif untuk pemodelan data aliran turbulen (Hu et al. 2020).

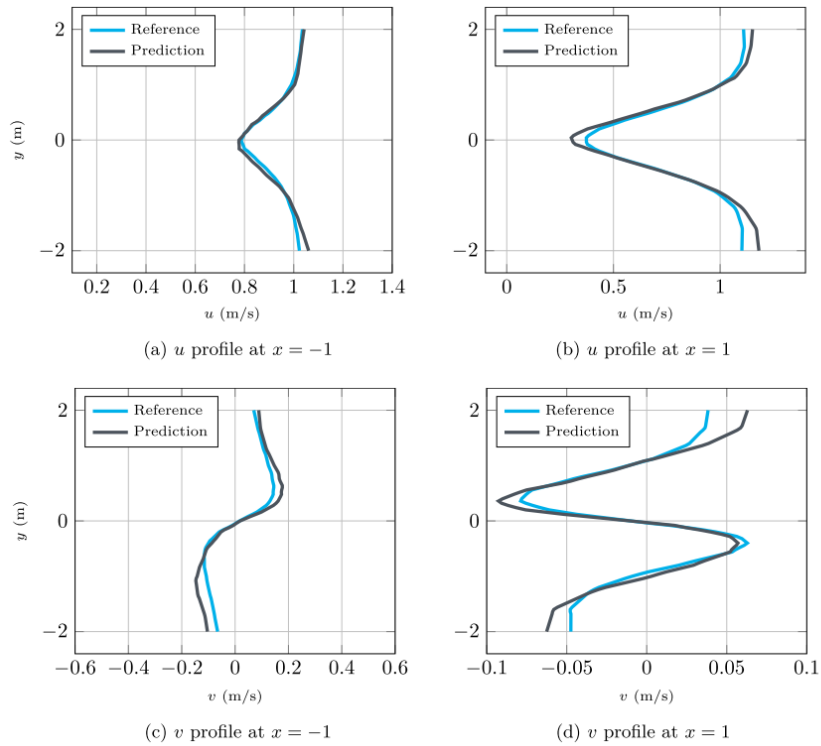
Model turbulen yang digunakan dalam prediksi aliran turbulen diantaranya Metode *Reynolds-averaged Navier Stokes* (RANS) (Zhao et al. 2020), Metode Large Eddy Simulation (LES) dan Metode *Direct numerical simulation* (DNS), Sudah dilakukan dalam pemodelan deep learning namun masih ada kemungkinan menggunakan Model Probability density function (PDF) yang dapat dilakukan dengan sifat dari PDF secara konvensional memiliki kompleksitas prediksi aliran turbulen. Model PDF karna mampu memberikan batasan pada aliran turbulen dan laminar. Hasil prediksi dari PDF memungkinkan dapat di jadikan sebuah parameter untuk memperbaiki pada deep learning untuk prediksi hasil aliran *turbulence*.

radial basis function neural networks (RBFNNs) adalah metode untuk membaca karakteristik nonlinier dari data aliran, dan *convolutional neural networks* (CNNs) diharapkan untuk menggantikan *long short-term memories* (LSTM) untuk membaca karakteristik temporal dari data aliran dan di bidang rekonstruksi bidang aliran yang stabil atau tidak stabil, (Hu et al. 2020) *CNN-based conditional generative adversarial networks* (cGANs) akan menjadi salah satu *framework* untuk menemukan distribusi spasial temporal dari data bidang aliran. (Hu et al. 2020)

Penelitian prediksi aliran trubulent sangat cepat berkembang dalam hal memperkecil nilai error data validasi dengan data hasil traning metode GCNN *Graph Convolutional Neural Networks* pada aliran laminar pada airfoil NACA 0012 dengan rata-rata nilai Error sebesar 3.43% merupakan nilai yang paling baik dalam penelitian beberapa tahun belakang ini. (Chen, Hachem, and Viquerat 2021) berikut adalah hasil penelitian pada airfoil 0012 yang dilakukan oleh (Chen, Hachem, and Viquerat)



Gambar 1.1 Laminar Flow predictions on a NACA0012 airfoil using a GCNN model.(Chen, Hachem, and Viquerat 2021)



Gambar 1.2 Profil kecepatan dan tekanan di sekitar airfoil NACA0012.
(Chen, Hachem, and Viquerat 2021)

Prediksi Aliran disekitar airfoil dengan metode GCNN dengan cara pendekatan menggunakan grafik *node* dan *edge* pada data input yang diperoleh dari hasil Simulasi OpenFOAM dengan pendekatan Mesh pada *Boundary Condition* merupakan pendekatan yang masuk akal karna dengan sifat aliran turbulen yang bersifat acak dengan mempertimbangkan antara *node* dan *edge* sangat memungkinkan untuk memprediksi hasil dengan tingkat kesalahan yang relative kecil.

1.2 Rumusan Masalah

Adapun rumusan masalah yang terjadi dalam penelitian ini dan yang akan dilakukan adalah sebagai berikut :

1. Seberapa besar kumpulan data saya harus benar melatih jaringan ?
2. Apakah benar-benar memprediksi data dengan baik di luar data pelatihan ?
3. Apakah *Graph Convolutional Neural Networks* akan menjadi *framework* terbaik untuk menemukan distribusi aliran turbulen.
4. Apakah model Aliran turbulen akan sama hasilnya dengan model laminar Yang sudah dilakukan oleh (J. Viquerat and E. Hachem) dengan membaca hubungan node dan edge yang pada *Mesh Boundary Condition*

1.3 Batasan Masalah

Adapun batasan masalah yang terjadi dalam penelitian ini dan yang akan dilakukan adalah sebagai berikut :

1. Penggunaan *Graph Convolutional Neural Networks* digunakan untuk memprediksi aliran Turbulen
2. Memprediksi aliran turbulen pada airfoil.
3. Menggunakan data Input dari hasil simulasi Openfoam yang berupa *nodes*, *edges*, *flow* dan Kecepatan

1.4 Tujuan

Adapun dan tujuan penelitian yang akan dilakukan adalah sebagai berikut :

1. Memprediksi aliran turbulen pada airfoil dengan *Graph Convolutional Neural Networks*
2. Memodifikasi Algoritma untuk perediksi aliran turbulen pada airfoil
3. Membuat dataset yang akan digunakan dengan *framework Graph Convolutional Neural Networks*

1.5. Manfaat Penelitian dan Keterbaruan

Untuk mengeksplorasi kemungkinan pengembangan sistem *Machine Learning* yang kuat yang mampu meningkatkan prediksi turbulen dengan Melatih data diluar data Pelatihan. Metode *Graph Convolutional Neural Networks* untuk memprediksi aliran laminar disekitar airfoil dikembangkan oleh (F.Ogoke,K.Meidani,et,al..2020) dan dilanjutkan oleh (J. Viquerat and E. Hachem,2021) dengan metode Graph yang mengoptimasi parameter *node* dan *edge* pada *Mesh Boundary Condition*. Keberhasilan penelitian tersebut belum dicoba pada aliran *turbulen* apakah bisa dilakukan dengan metode GCNN. Pada penelitian prediksi aliran turbulen disekitar airfoil dengan metode GCNN diharapkan mampu mebuktikan bahwa metode GCNN memiliki tingkat eror yang cukup rendah 3.43% pada aliran laminar disekitar airfoil. Diharapkan prediksi aliran pada airfoil dengan metode GCNN menjadi sebuah kemajuan dalam bidang *fluid dynamic* dengan penerapan ilmu teknologi informasi.

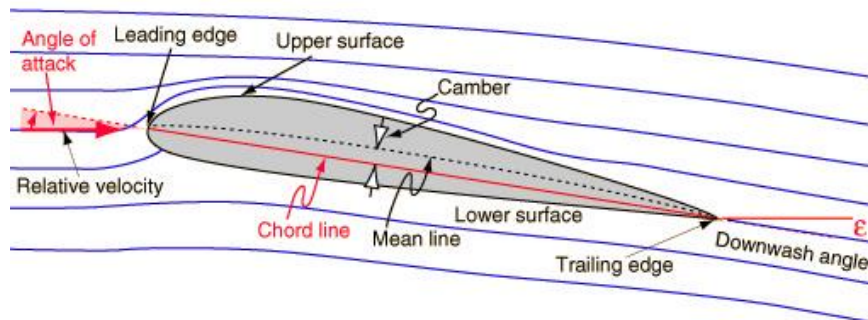
BAB II

Tinjauan Pustaka

2.1 Airfoil

Airfoil atau aerofoil adalah suatu bentuk geometri yang apabila ditempatkan di suatu aliran fluida akan memproduksi gaya angkat (lift) lebih besar dari gaya hambat (*drag*). Pada airfoil terdapat bagian-bagian seperti berikut :

- a. *Leading Edge* adalah bagian yang paling depan dari sebuah airfoil.
- b. *Trailing Edge* adalah bagian yang paling belakang dari sebuah airfoil.
- c. *Chamber line* adalah garis yang membagi sama besar antara permukaan atas dan permukaan bawah dari airfoil mean *chamber line*.
- d. *Chord line* adalah garis lurus yang menghubungkan *leading edge* dengan *trailing edge*.
- e. *Chord (c)* adalah jarak antara *leading edge* dengan *trailing edge*.
- f. Maksimum *chamber* adalah jarak maksimum antara mean *chamber line* dan *chord line*. Posisi maksimum *chamber* diukur dari *leading edge* dalam bentuk persentase *Chord (c)*
- g. Maksimum *thickness* adalah jarak maksimum antara permukaan atas dan permukaan bawah airfoil yang juga diukur tegak lurus terhadap *chord line*.



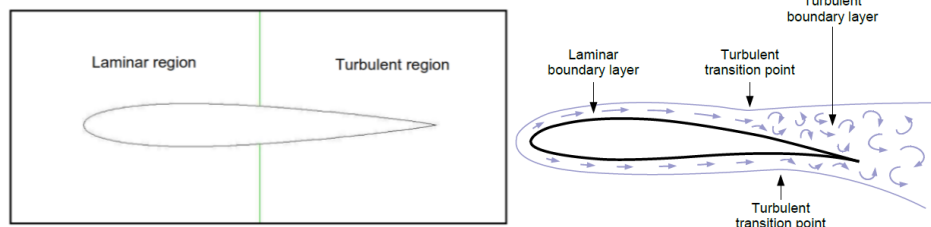
Gambar 2.1 Bagian-bagian airfoil (R Nave n.d.2021)

2.2 Model Turbulen

Mekanika fluida merupakan ilmu yang sangat mendasar dalam dunia fisika dan engineering, penerapannya sangatlah banyak dan variatif, mulai dari desain

roket dan pesawat terbang hingga analisis biomedis. Meskipun ilmu ini sudah cukup lama berkembang dan digunakan, namun formulasi-formulasi dari hukum fluida ini masih banyak yang belum terpecahkan, misalkan persamaan *Navier-stokes* yang merupakan persamaan fluida dengan bentuk diferensial non-linear. Salah satu penyebab tidak dapat diselesaikannya persamaan ini adalah sifat alami dari fluida yang pada kondisi tertentu bersifat sangatlah random, unsteady dan dinamis sehingga tidak dapat diprediksi dengan baik, kondisi ini dikenal juga dengan istilah turbulen.

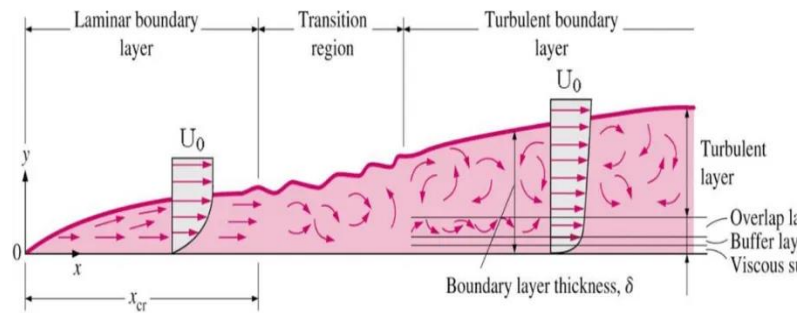
Secara definisi, aliran turbulen adalah aliran dengan pola yang random dan kacau yang mengandung eddie, swirl, serta ketidakstabilan aliran didalamnya. Sedangkan lawan kata dari turbulen adalah laminar, yaitu aliran dengan pola yang halus dan terprediksi tanpa adanya gangguan antar path. Pada aliran yang laminar, persamaan Navier-stokes terkadang mudah untuk diselesaikan misalkan disederhanakan menjadi persamaan Bernoulli. Karena kesulitannya dalam menyelesaikan masalah turbulensi ini secara matematis, bahkan matematikawan terkenal peraih nobel, Richard Feynman menyatakan bahwa “turbulen adalah masalah paling penting dan belum terselesaikan dalam fisika klasik”



Gambar 2.2 Ilustrasi aliran laminar dan trubulen (Anam, Budiarto, and Budiyono 2015)

Karena secara matematik analitis kasus ini belum terselesaikan, muncul beberapa gagasan untuk mengkuantifikasi turbulensi ini berdasarkan eksperimen, dan metode yang paling terkenal adalah yang dikemukakan oleh *Osborne Reynold* (1883), yang menemukan sebuah rasio non-dimensional yang mampu memprediksi apakah aliran tersebut akan laminar atau turbulen, nilai ini dikenal juga dengan Reynold Number, Re = rasio antara gaya

internal dengan gaya eksternal. Menggunakan bilangan *Reynold* ini, dapat diprediksi dengan baik terjadinya aliran laminar, turbulen atau transisi (perubahan dari laminar ke turbulen). Misalkan untuk aliran di dalam pipa, untuk $Re = 0-2300$ aliran adalah laminar, kemudian $Re = 2300-4000$ aliran transisi, dan $Re > 4000$ aliran adalah turbulen, tidak peduli fluida apa yang digunakan dan berapapun kecepatan dan diameter pipa tersebut. Bilangan ini menyatakan bahwa semakin mendominasi gaya viskos dari fluida maka aliran akan laminar, sedangkan semakin mendominasi gaya internal maka aliran akan turbulen.



Gambar 2.2 Batas lapisan Aliran laminar dan Turbulen (Keith Agedy 2021)

Persamaan Navier-stokes merupakan sistem persamaan differensial non-linier yang mendeskripsikan bagaimana fluida mengalir. Persamaan ini menjelaskan keseimbangan gaya-gaya yang bekerja pada fluida yaitu gaya-gaya benda dan gaya-gaya permukaan. Dalam menentukan solusi persamaan numerik, model turbulensi adalah prosedur komputasional untuk mendekatkan sistem persamaan aliran sehingga masalah aliran yang bervariasi dapat dihitung. Secara garis besar, model turbulensi digunakan untuk memodifikasi persamaan Navier Stokes dengan memperkenalkan kuantitas rata-rata dan besar fluktuasi untuk menghasilkan persamaan *Reynolds Averaged Navier Stokes*. Dalam persamaan CFD persamaan RANS dirumuskan (Anand and Srinivasan n.d.)

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \delta_{ij} \frac{\partial u_l}{\partial x_l} \right) \right] + \frac{\partial}{\partial x_j} \left(-\rho \overline{u_i u_j} \right)$$

Model turbulensi yang sering digunakan untuk memodifikasikan persamaan RANS diantaranya *Spalart-Almarast*, *k-epsilon*, *k-omega* dan *shear stress transport (SST)*.

A. *Spalart-Allmaras*

Spalart-Allmaras merupakan model turbulen dengan satu persamaan yang menyelesaikan model persamaan transport untuk viskositas kinematik turbulen. Model ini dapat digunakan untuk simulasi yang relatif kasar dengan ukuran mesh yang besar dimana perhitungan aliran turbulen bukan merupakan hal yang bersifat kritis. Pemodelan yang akurat di dekat dinding merupakan hal yang penting untuk kebanyakan aplikasi teknik karena perkiraan penurunan tekanan (pressure drop), separasi aliran, lapisan batas bergantung kepada kebenaran prediksi gaya geser lokal pada dinding tersebut, oleh karena itu diperlukan treatment tambahan untuk membuat kedua model tersebut lebih akurat untuk memprediksikan perilaku aliran di dekat dinding, terutama pada aliran yang sangat terpengaruh oleh dinding (wall-bonded flow). Persamaan transport untuk model turbulen *Spalart-Almarast* (SA) dinyatakan sebagai berikut (Piotrowski 2020; Sulistiya and K. 2018):

$$\frac{\partial}{\partial t} (\rho \tilde{\nu}) + \frac{\partial}{\partial x_i} (\rho \tilde{\nu} u_i) = G_v + \frac{1}{\sigma_{\tilde{\nu}}} \left[\frac{\partial}{\partial x_j} \left\{ (\mu + \rho \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right\} + C_{b2} \rho \left(\frac{\partial \tilde{\nu}}{\partial x_j} \right)^2 \right] - Y_v + S_{\tilde{\nu}}$$

dimana G_v adalah viskositas turbulen produksi, Y_v adalah viskositas turbulen destruksi pada daerah dekat dinding karena adanya dinding penghalang dan redaman viskos, ν adalah viskositas kinematik molekul fluida, $S_{\tilde{\nu}}$ adalah sumber yang didefinisikan oleh pengguna.

B. K-Epsilon

Model ini merupakan model semi empiris yang dikembangkan oleh Launder & Spalding. Model kepsilon merupakan model turbulensi yang cukup lengkap dengan dua persamaan yang memungkinkan kecepatan turbulen (*Turbulent velocity*) dan skala panjang (length scales) ditentukan secara independen. Kesetabilan, ekonomis (dari sisi komputasi) dan akurasi yang memadai untuk berbagai jenis aliran turbulen membuat model k-epsilon sering digunakan pada simulasi fluida dan perpindahan panas model k-epsilon standar didasarkan pada persamaan transport untuk energi kinetik turbulen (k) dan tingkat disipasinya (ϵ). Energi kinetik turbulen dan laju disipasinya dapat diperoleh dari dua persamaan transport berikut :

$$\begin{aligned}\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho k u_j) &= \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + G_k + G_b - \rho \epsilon - Y_M + S_k \\ \frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_j}(\rho \epsilon u_j) &= \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + C_{1\epsilon} \frac{\epsilon}{k} (G_k + C_{3\epsilon} G_b) - C_{2\epsilon} \rho \frac{\epsilon^2}{k} + S_\epsilon\end{aligned}$$

dimana G_k merepresentasikan energi kinetik turbulensi dari gradient kecepatan rata-rata, G_b merepresentasikan energi kinetik turbulensi dari daya apung, Y_M merepresentasikan kontribusi dari fluktuasi dilatasi pada aliran turbulen kompresibel terhadap keseluruhan tingkat disipasi. $C_{1\epsilon}$, $C_{2\epsilon}$ dan $C_{3\epsilon}$ adalah konstanta. σ_k dan σ_ϵ adalah turbulent Prandtl number untuk masing-masing σ_k dan σ_ϵ . S_k dan S_ϵ adalah sumber yang ditentukan oleh pengguna.

C. Model k-omega

Model k-omega standar merupakan model turbulensi berdasarkan *model Wilcox k-omega* yang memasukkan beberapa modifikasi untuk menghitung efek aliran pada bilangan Reynolds rendah,

kompresibilitas dan penyebaran aliran geser (*shear flow*). Model ini dapat diaplikasikan pada aliran dalam maupun aliran bebas geseran (*free shear flow*). Model k-omega standar didasarkan model empiris berdasarkan model persamaan transport untuk energi kinetik turbulen k dan tingkat spesifik ω , atau disebut juga rasio antara ϵ terhadap k. pada model ini, istilah produksi telah ditambahkan dalam

persamaan untuk meningkatkan akurasi prediksi penyebaran aliran geser(Sulistiya and K. 2018) .

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j} \left(\Gamma_k \frac{\partial k}{\partial x_j} \right) + G_k - Y_k + S_k$$

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_i}(\rho \omega u_i) = \frac{\partial}{\partial x_j} \left(\Gamma_\omega \frac{\partial \omega}{\partial x_j} \right) + G_\omega - Y_\omega + S_\omega$$

dimana G_k merepresentasikan energi kinetik turbulensi dari gradient kecepatan rata-rata, G_ω merepresentasikan tingkat disipasi energi ω , Γ_k dan Γ_ω merepresantasikan difusivitas efektif dari k dan ω , Y_k dan Y_ω merepresentasikan disipasi turbulensi dari k dan ω . S_k dan S_ω adalah sumber yang ditentukan oleh pengguna.

D. Model *Shear Stress Transport*

Salah satu pemasalahan utama dalam memodelkan turbulensi adalah perdiksi yang akurat terhadap separasi aliran di permukaan yang halus. Model turbulen dua equation sering gagal dalam dalam memprediksi letak awal dan jumlah separasi aliran didalam kondisi *adverse pressure gradient*. Hal ini adalah fenomena yang penting dalam berbagai aplikasi teknis khususnya aerodinamika pesawat karena karakteristik stallnya diatur oleh separasi aliran pada sayap. Oleh karena itu dikembangkan model yang dapat memprediksi separasi aliran dengan lebih akurat.

Model *Shear-Stress-Transport* (SST) didesain untuk memberikan akurasi yang tinggi terhadap letak awal dan jumlah aliran dalam separasi didalam adverse pressure gradient dengan memasukkan efek transport ke dalam formulasi eddy viscosity. Model ini disarankan untuk simulasi lapisan batas yang akurat. Oleh karena itu dibutuhkan resolusi yang tinggi pada lapisan batas. Model SST memiliki kelebihan dapat memprediksi separasi aliran dengan lebih akurat Sangat cocok untuk mendapatkan akurasi tinggi di lapisan batas, akurasi dalam menangani komputasi near wall treatment pada bilangan Reynolds rendah. Selain kelebihan model SST juga mempunyai kekurangan yaitu dibutuhkan grid yang bagus dan rapat di sekitar *wall* yang berarti dibutuhkan usaha lebih besar pada saat

grid generation. Persamaan transport untuk model turbulensi SST didefinisikan sebagai berikut (Sulistiya and K. 2018):

$$\frac{\partial(\rho\gamma)}{\partial t} + \frac{\partial(\rho U_j \gamma)}{\partial x_j} = P_{\gamma 1} - E_{\gamma 1} + P_{\gamma 2} - E_{\gamma 2} + \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\gamma} \right) \frac{\partial \gamma}{\partial x_j} \right]$$

2.2.1 Model (LES) *Large eddy simulations*

Large eddy simulations (LES) mengurangi beban komputasi dari simulasi aliran reaksi turbulen. LES bekerja dengan variabel keadaan yang disaring secara spasial, yang menunjukkan struktur temporal dan spasial yang jauh lebih sedikit dan dengan demikian membutuhkan resolusi numerik yang jauh lebih sedikit.

Resolusi simulasi bergantung pada grid dan perbaikan langkah waktu. Karena LES hanya Seharusnya untuk menangkap pusaran terbesar, grid hanya diperhalus untuk menangkap pusaran yang mengandung energi. Dalam persamaan CFD persamaan RANS dirumuskan (Matai 2018) :

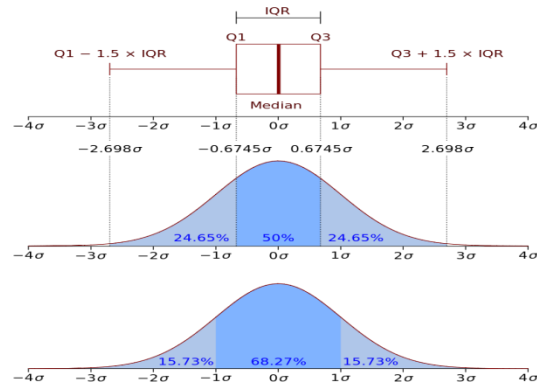
$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}$$

2.2.2 Model (PDF) *Probability density function*

PDF digunakan untuk menentukan probabilitas variabel acak yang berada dalam rentang nilai tertentu, sebagai lawan untuk mengambil satu nilai. (Raman, Pitsch, and Fox 2006) persamaan model PDF gabungan dari kecepatan dan hilangnya energi mekanik karena gesekan atau gaya mirip gesekan dapat dikembangkan, dengan membangun model stokastik untuk kecepatan dan disipasi mengikuti partikel fluida. (Raman, Pitsch, and Fox 2006)

Probabilitas ini diberikan oleh integral PDF variabel ini pada rentang tersebut yaitu, diberikan oleh area di bawah fungsi *density*. Jika variabel acak kontinu, maka probabilitas dapat dihitung melalui fungsi PDF. Bentuk *Probability density function* di seluruh domain untuk variabel acak disebut sebagai distribusi probabilitas dan distribusi probabilitas umum memiliki nama, seperti seragam, normal, eksponensial, dan seterusnya. Dengan metode

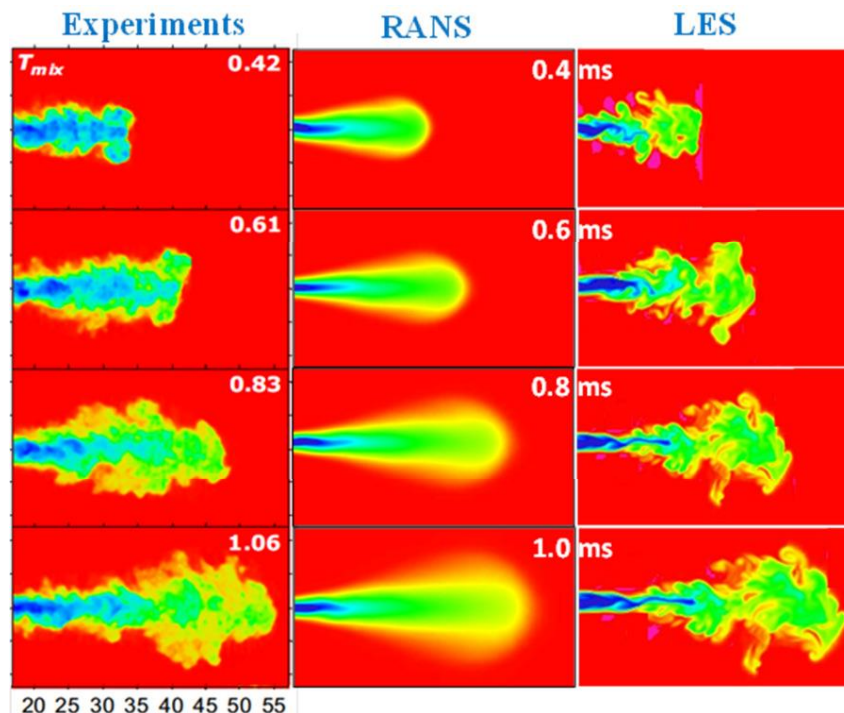
Probabilitas untuk menentukan batas suatu nilai pada density dimungkinkan dilakukan dalam penelitian prediksi aliran turbulen.



Gambar 2.1 Grafik *probability density function of a normal distribution*

2.2.3 Perbandingan Model RANS dan Model LES

Gambar 2.2 adalah perbandingan model RANS dan model LES pada ruang pembakaran mesin diesel dalam keadaan normal. Terlihat perbedaan yang lebih mendekati hasil percobaan. Yang dilakukan oleh Sandia National Laboratory (Som, Senecal, and Pomraning 2012)



Gambar 2.2 *Images comparing the calculated temperature contours using RANS and LES models*(Som, Senecal, and Pomraning 2012)

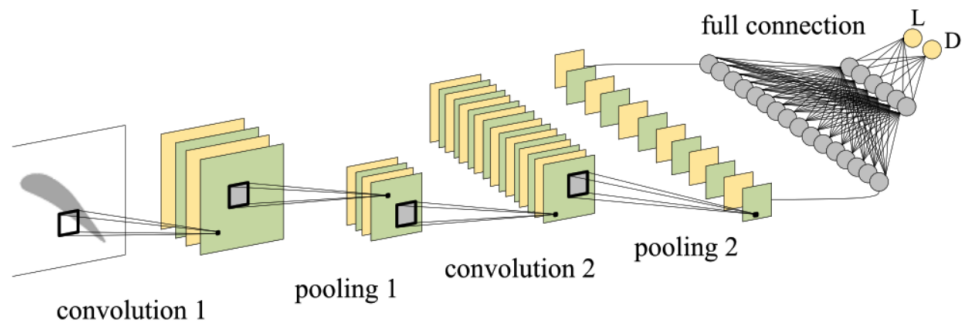
2.3 *Deep Learning*

Fitur yang menonjol dari arsitektur deep learning, terutama deep neural networks (DNNs), adalah kemampuannya untuk mempelajari hubungan spasial atau temporal antara input data mentah. Oleh karena itu, DNN adalah alternatif yang menjanjikan untuk memodelkan sistem dinamis kompleks yang fisiknya tidak diketahui secara akurat atau mengandung stokastisitas yang melekat. Sifat kacau aliran fluida turbulen, dan upaya komputasi besar yang diperlukan untuk mensimulasikannya secara akurat, menawarkan kemungkinan bagi model DNN untuk secara potensial mencapai peningkatan kinerja dibandingkan pendekatan simulasi konvensional.

Convolutional neural networks (CNN) adalah jenis DNN khusus dengan koneksi lokal antar lapisan. Jaringan ini telah digunakan untuk mempercepat langkah yang memakan waktu dalam simulasi fluida, yang memiliki aplikasi dalam grafik komputer dan animasi untuk memberikan efek asap dan fluida yang realistis. (Tompson et al. 2017) Selain itu, kemampuan recurrent neural networks (RNNs) untuk prediksi data sekuensial membawa ide penggunaan RNNs untuk learning dynamics fluid dan estimasi keadaan masa depan dari systems dynamics fluid. long short-term memory (LSTM) digunakan untuk memprediksi evolusi aliran, misalnya *learning dynamic* dan prediksi aliran turbulen.

2.4 **Convolutional neural networks (CNN)**

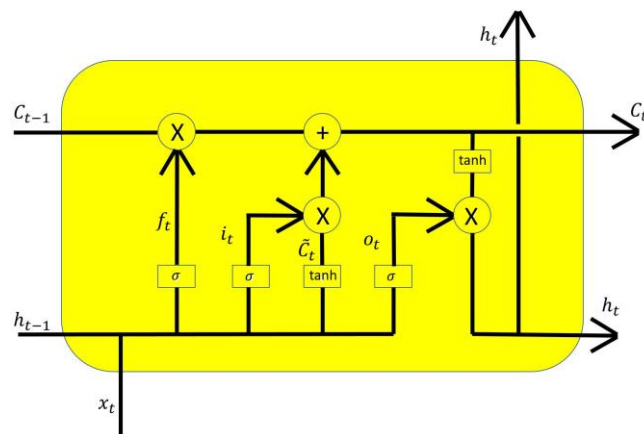
CNN memungkinkan untuk mempelajari fitur-fitur tingkat tinggi invarian ketika data memiliki korelasi spasial dan / atau temporal yang kuat. Kemampuan ini sangat penting untuk banyak tugas pemodelan aerodinamis telah menunjukkan bahwa CNN memungkinkan estimasi cepat untuk bidang aliran, dan studi difokuskan pada replikasi visual bidang kecepatan di sekitar sekelompok geometri kategori. CNN dapat mengatasi pemrosesan data urutan waktu, bahkan lebih baik daripada RNNs



Gambar 2.3 CNN klasik dalam prediksi respons aerodinamis stabil nonlinier. Jaringan terdiri dari dua lapisan konvolusional, dua lapisan penyatuan, dan dua lapisan yang terhubung sepenuhnya(Hu et al. 2020)

2.5 Long Short Term Memory (LSTM)

Jaringan LSTM berbeda dari jaringan dalam lainnya arsitektur pembelajaran seperti jaringan saraf konvolusional (CNN), di mana sel LSTM biasanya berisi tiga gerbang: gerbang input, gerbang output, dan gerbang lupa. LSTM mengatur aliran informasi pelatihan melalui gerbang ini dengan menambahkan informasi secara selektif (input gate), menghapus (forget gate) atau membiarkannya masuk ke sel berikutnya (output gate).



Gambar 2.4 Arsitektur Sel LSTM dengan Berbagai Gerbang
(Hu et al. 2020)

Jaringan LSTM sangat cocok untuk mengklasifikasikan, memproses, dan membuat prediksi berdasarkan data deret waktu, karena mungkin ada jeda durasi yang tidak diketahui antara peristiwa penting dalam deret waktu. LSTM

dikembangkan untuk menangani masalah gradien lenyap yang dapat ditemukan saat melatih RNN tradisional. Ketidakepekaan relatif terhadap panjang celah adalah keunggulan LSTM dibandingkan RNN, model Markov tersembunyi, dan metode pembelajaran urutan lainnya dalam berbagai aplikasi.

2.6 Generative Adversarial Networks (GANs)

Hasil prediksi yang terbaik itu yang diharapkan, dengan adanya *Generative Adversarial Networks* yang memikirkan output dengan menambahkan syarat tertentu baik berupa variable. Pada Kausus Penelitian prediksi aliran lapisan aliran trubulen yang sudah dilakukan oleh (Lee, Lee, and You 2018). penerapan generative adversarial networks (GANs) untuk subgrid pemodelan dalam aliran trubulen, karena akan menjadi alat yang fleksibel, juga menjanjikan untuk simulasi aliran trubulen reaktif. GAN milik kelas model generatif tertentu yang bertujuan untuk memperkirakan probability density yang tidak diketahui yang mendasari data yang diamati.(Bode et al. 2019)

Dalam GAN memungkinkan berbagai aplikasi seperti interpolasi antara titik data dan modifikasi yang berarti dari titik data yang ada. modifikasi dapat dilakukan dengan menentukan sebuah karakter aliran fluida pada airfoil misalnya kecepatan aliran, density, reynold number dengan menentukan karakter ini diharapkan dapat menjawab permasalahan yang ada pada prediksi aliran trubulen

2.7 Perbandingan CNN,LSTM, GAN dalam bidang dinamika fluida

Interpretabilitas jaringan saraf di bidang aerodinamika juga akan menjadi topik hangat di masa mendatang. Dalam prediksi respons aerodinamis yang tidak stabil, CNN dan LSTM biasanya digunakan untuk target yang berbeda.

CNN fokus pada nonlinier spasial dari pemodelan aerodinamis yang tidak stabil. Model hybrid dan model multikernel akan menjadi topik penelitian nonlinier di masa depan. LSTM dan CNN fokus pada kesinambungan temporal pemodelan aerodinamis yang tidak stabil. Bahwa CNN dapat memproses data urutan waktu dengan cara input rekursif, dan menunjukkan memori efektif yang lebih lama daripada LSTM. Namun, fitur CNN, saat ini, masih tidak dapat

dijelaskan dan membutuhkan studi lebih lanjut. Di bidang rekonstruksi bidang aliran, rekonstruksi bidang aliran stabil dan tidak stabil dianalisis. Dalam rekonstruksi medan aliran yang stabil, akurasi prediksi dan stabilitas model adalah indikator utama.

Generative adversarial networks (GANs) untuk subgrid pemodelan dalam aliran turbulen, karena akan menjadi alat yang fleksibel, juga menjanjikan untuk simulasi aliran turbulen reaktif. Dalam (GANs) mampu menginterpolasikan parameter Probability density yang tidak di ketahui. Parameter ini diharapkan menjadi sebuah solusi yang mengabungkan antra CNN berbasis GAN karna CNN mampu memodelkan aliran turbulen yang tidak stabil kemudian GAN menjadikan Pelengkap untuk memperkuat hasil dari prediksi trubulen dengan menambahkan parameter probability density.

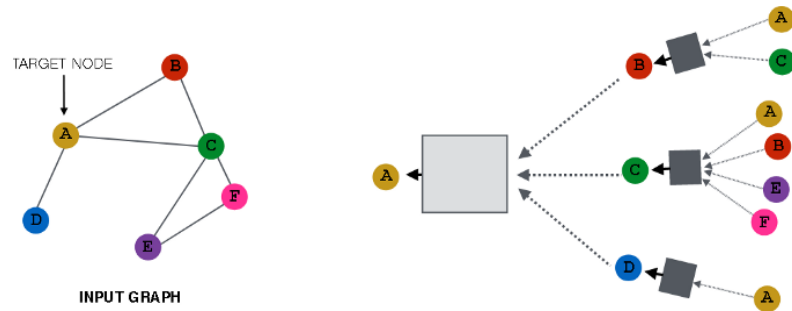
2.8 GNN *Graph Convolutional Neural Networks*

Graph adalah struktur data non-linier yang terdiri dari *node* dan *edge*. Kadang-kadang simpul juga disebut simpul dan sisi-sisinya adalah garis atau busur yang menghubungkan graf dengan dua simpul. Data berstruktur grafik dapat ditemukan hampir di mana-mana. Masalah yang digunakan untuk memecahkan GNN dapat dibagi ke dalam kategori berikut:

- A. Klasifikasi Node: Tujuan dari tugas ini adalah untuk menentukan pelabelan sampel (diwakili sebagai node) dengan memeriksa label tetangga terdekat mereka (yaitu, label tetangga mereka). Biasanya, masalah seperti ini dilatih dengan cara semi-diawasi, dengan hanya sebagian dari grafik yang diberi label selama proses pelatihan.
- B. Klasifikasi Grafik: Tujuannya di sini adalah untuk mengkategorikan seluruh grafik ke dalam berbagai kategori. Ini mirip dengan klasifikasi gambar, kecuali bahwa target sekarang berada dalam domain grafik. Aplikasi klasifikasi grafik sangat banyak, dan berkisar dari menentukan apakah suatu protein adalah enzim atau tidak dalam bioinformatika hingga mengkategorikan dokumen dalam pemrosesan bahasa alami (NLP) atau analisis jejaring sosial, antara lain.

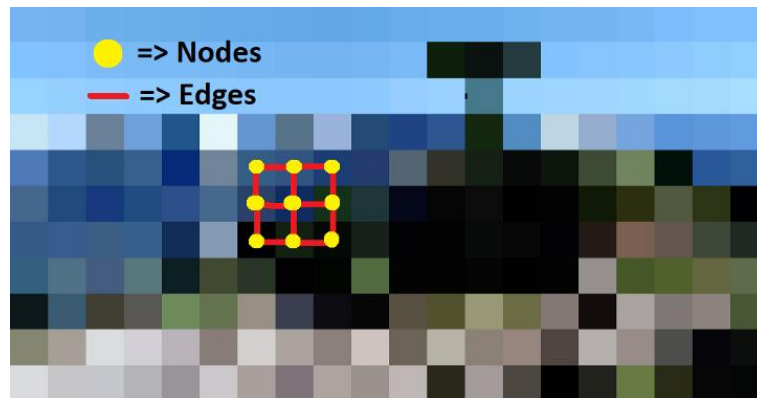
- C. Visualisasi grafik: Visualisasi informasi adalah cabang matematika dan ilmu komputer yang ada di persimpangan teori grafik geometris dan ilmu komputer. Hal ini berkaitan dengan representasi visual dari grafik yang mengungkapkan struktur dan anomali yang mungkin ada dalam data dan membantu pengguna dalam memahami grafik yang disajikan.
- D. Prediksi tautan : Secara khusus, algoritme harus memahami hubungan antara entitas dalam grafik, serta mencoba memperkirakan kemungkinan adanya koneksi antara dua entitas. Dalam jejaring sosial, perlu untuk menyimpulkan interaksi sosial atau menyarankan teman potensial untuk pengguna agar dapat berfungsi dengan baik. Ini juga telah diterapkan pada masalah yang melibatkan sistem pemberi rekomendasi dan prediksi asosiasi kriminal.
- E. Pengelompokan grafik: Visualisasi data dalam bentuk grafik disebut sebagai pengelompokan. Pada data graf, ada dua jenis pengelompokan yang berbeda yang dapat dilakukan. Pengelompokan simpul mencoba untuk mengelompokkan simpul-simpul dari suatu graf ke dalam kelompok-kelompok daerah yang terhubung secara padat berdasarkan bobot tepi atau jarak tepi antara simpul-simpul grafik. Saat menggunakan bentuk kedua dari pengelompokan grafik, grafik diperlakukan sebagai objek yang akan dikelompokkan, dan objek dikelompokkan berdasarkan kesamaannya.

GNN adalah hibrida dari mekanisme difusi informasi dan jaringan saraf yang digunakan untuk memproses data, mewakili satu set fungsi transisi dan satu set fungsi output. Mekanisme difusi informasi didefinisikan oleh node yang memperbarui statusnya dan bertukar informasi dengan meneruskan "pesan" ke node tetangganya hingga mencapai keseimbangan yang stabil. Proses ini melibatkan pertama fungsi transisi yang mengambil sebagai masukan fitur dari setiap node, tepi fitur dari setiap node, negara node tetangga, dan node tetangga fitur dan keluaran negara baru node.

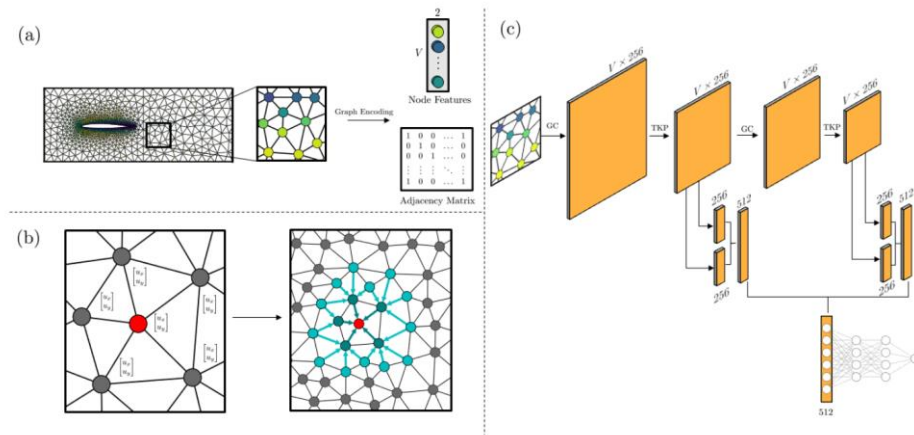


Gamabar 2.5 Target node yang memberikan informasi dari node tetanganya (Baldassarre and Azizpour 2019)

Generalisasi grafik suatu gambar, kita dapat mengatakan bahwa metode penjelasan CNN tidak terlalu memperhatikan *edge* (koneksi antar piksel) dan lebih fokus pada node (nilai piksel). Masalahnya di sini adalah bahwa data grafik mengandung banyak informasi berguna di *edge* (Baldassarre and Azizpour 2019)



GAMBAR 3. Sebuah gambar dapat digeneralisasikan sebagai grafik berbentuk kisi. Palang merah hanya menggambarkan sebagian kecil gambar yang dapat direpresentasikan sebagai grafik. Setiap piksel dapat dianggap sebagai node V yang memiliki 3 fitur (nilai RGB) dan yang terhubung ke piksel tetangga dengan tepi E . (Baldassarre and Azizpour 2019)



Gambar (a) Informasi kecepatan yang didefinisikan pada mesh tidak terstruktur direpresentasikan sebagai grafik, di mana node mesh adalah simpul dari grafik, dan konektivitas mesh diambil sebagai tepi grafik. Matriks $2 \times N$ fitur simpul berisi kecepatan di setiap dimensi, dan matriks kedekatan $N \times N$ mengkodekan konektivitas matriks. (b) Operasi Konvolusi Graf. (kiri) Grafik sebelum operasi konvolusi dilakukan pada simpul tengah (merah). (kanan) Selama konvolusi graf, informasi di setiap cincin tetangga orde- N , di mana $N \leq k$, dikumpulkan ke simpul pusat. Pada aplikasi ini, $k = 2$. (c) Arsitektur Graph Convolutional Neural Network. 'GC' mengacu pada operasi konvolusi grafik di b), 'TKP' mengacu pada Top-K Pooling. Output peta fitur dari setiap top-K pooling layer mengalami mean pooling dan max pooling, dan output dari setiap operasi digabungkan bersama. Output gabungan dari setiap lapisan ditambahkan bersama-sama dan diteruskan ke lapisan yang terhubung sepenuhnya untuk regresi. (Ogoke et al. 2020)

2.9 Perbandingan Tinjauan dari Penelitian Sebelumnya

Pada tabel 2.1 adalah penelitian yang sudah dilakukan dalam 5 tahun terakhir.

Tabel 2.1. Penelitian Terdahulu

No	Tahun / penelitian	Judul	Metode	HASIL
1	J. Viquerat and E. Hachem, 1 Oktober 2021	Graph neural networks for laminar flow prediction around random two-dimensional shapes A	Graph Convolutional Neural Networks dengan hasil simulasi Openfoam Menggunakan Spalart–Allmaras pada airfoil Serta Jumlah node 2000 sampai 3000 node	Mampu Memperediksi aliran laminar pada sekitar Airfoil dengan Metode GCNN dengan tingkat eror hasil prediksi dengan data validasi sebesar 3.43%.
2	F.Ogoke,K.Meidani,A.Hashemi, andA.B.Farimani. 3 Desember 2020	Graph Convolutional Neural Networks for Body Force Prediction	Graph Convolutional Neural Networks dengan hasil simulasi Openfoam Menggunakan Spalart–Allmaras pada airfoil	Mampu Memperediksi aliran pada sekitar Airfoil dengan Metode GCNN
3	Piotrowski, Michael G H 22Oktober 2020	Smooth Local Correlation-Based Transition Model for the Spalart–Allmaras Turbulence Model	Model transisi LM2015 dimodifikasi dan digabungkan dengan model turbulensi Spalart–Allmaras	Mampu memprediksi transi alami dan pemisahan yang diinduksi secara akurat.
4	Nils Thuerey,* Konstantin Weißenow, 16 November 2019	Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows	dimodernisasi arsitektur U-net yang dan mengevaluasi sejumlah besar neural networks terlatih sehubungan dengan keakuratannya untuk penghitungan distribusi tekanan dan kecepatan	tekanan relatif rata-rata dan kesalahan kecepatan kurang dari 3% di berbagai bentuk airfoil yang sebelumnya tidak terlihat

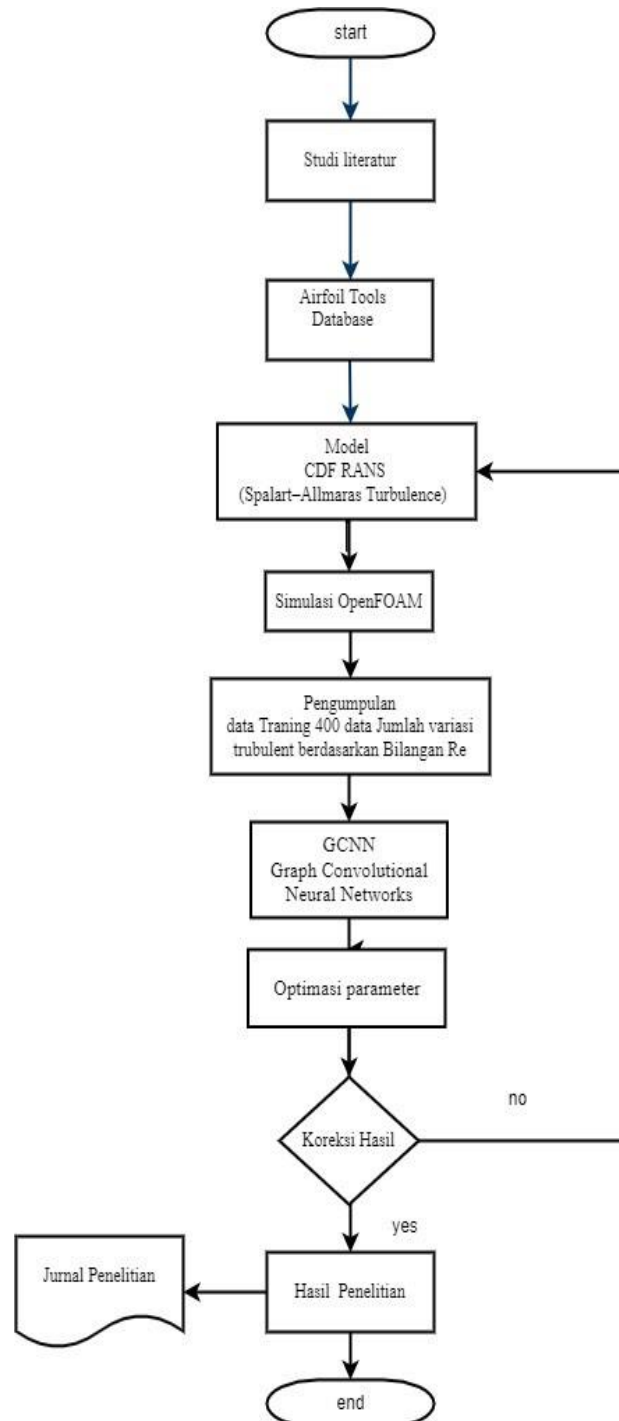
5	Arvind T. Mohan & Datta V. Gaitonde 24 April 2018	A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks	Pendekatan berbasis pembelajaran mendalam untuk membangun ROM menggunakan basis Proper Orthogonal Decomposition (POD) dari kumpulan data DNS, untuk aplikasi kontrol aliran turbulen dengan menambahkan eksponen Hurst	Hasil prediksi aliran turbulent baik dengan memanfaatkan eksponen Hurst untuk mempelajari penerapan pendekatan pemodelan prediktif untuk mekanika fluida.
6	M. T. Henry de Frahan 8 juli 2019	Deep learning for presumed probability density function models	Menggunakan tiga perwakilan algoritma ML yang berbeda jenis ML (traditional methods, deep learning, and generative learning) untuk merancang model PDF yang dianggap untuk aplikasi ruang pembakaran	menunjukkan bahwa model yang dirancang melalui teknik ML lanjutan lebih mampu menangkap kompleksitas FDF ini daripada model analitik atau model regresi linier
7	Julia Ling dan Jeremy Templeton 24 Juli 2016	Reynolds Averaged Turbulence Modeling using Deep Neural Networks with Embedded Invariance	Penggunaan deep neural networks untuk mempelajari model tensor anisotropi tegangan Reynolds dari data simulasi	Peningkatan yang signifikan model viskositas eddy linier RANS dasar dan model viskositas eddy non-linier ditunjukkan.

Bab III

Metode Penelitian

3.1 Tahapan Konsep Penelitian

Tahapan Penelitian prediksi aliran turbulen sebagai berikut :



Gambar 3.1 Alur Penelitian

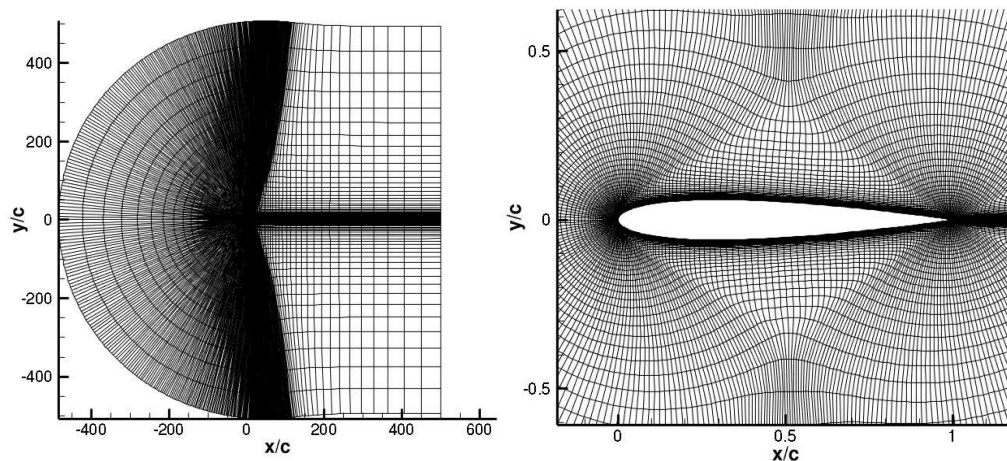
3.2 Pengumpulan Data

Data yang digunakan dalam penelitian data CFD dari OpenFOAM seperti yang dilakukan oleh (J. Viquerat, 2021) atau menggunakan data *Johns Hopkins Turbulence database* JHTB. JHTB sudah banyak dilakukan untuk penelitian prediksi aliran turbulen yang sudah dilakukan oleh (Gaitonde 2018) (Liu et al. 2020). Data bisa berupa data kecepatan, tekanan, viskositas, begitu juga dengan menggunakan data gambar aliran. Selain data dari JHTB data hasil experiment yang dapat dipercaya

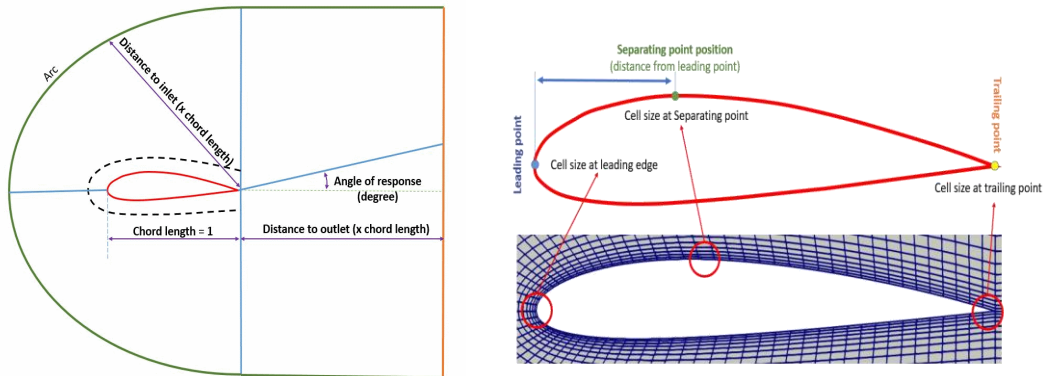
Untuk dataset yang dibutuhkan training data penelitian yang dilakukan oleh (J. Viquerat, 2021) sebanyak 2000 sampai 3000 node computational domain untuk aliran laminar pada airfoil NACA 0012 untuk mencapai nilai eror antara data training dengan data validation.

3.7 Airfoil NACA 0012

Airfoil NACA dari database *Airfoil Tools* adalah salah satu geometri yang paling banyak dipelajari di komunitas CFD. Airfoil NACA 0012 dipilih memudahkan untuk membandingkan dan memvalidasi karna sudah banyak digunakan dalam penelitian prediksi aliran turbulen atau laminar memudahkan dalam pengumpulan data yang sudah diteliti terdahulu. Berikut adalah *Blok mesh* dari NACA 0012



Gambar 3.2 Mesh Komputasi Untuk Aliran NACA 0012(Tracey, Duraisamy, and Alonso 2015)

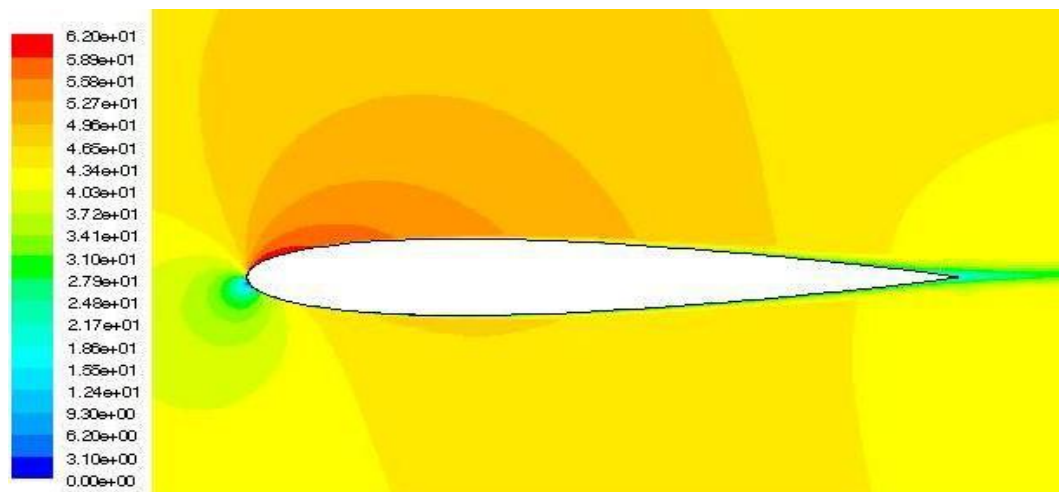


Gambar 3.4 Define domain Meshing airfoil boundary (Phan n.d.)

Lapisan batas pada airfoil dipastikan dengan *edgeGrading* yang kompleks sehingga kita tidak perlu memisahkan lapisan batas menjadi blok yang berbeda.

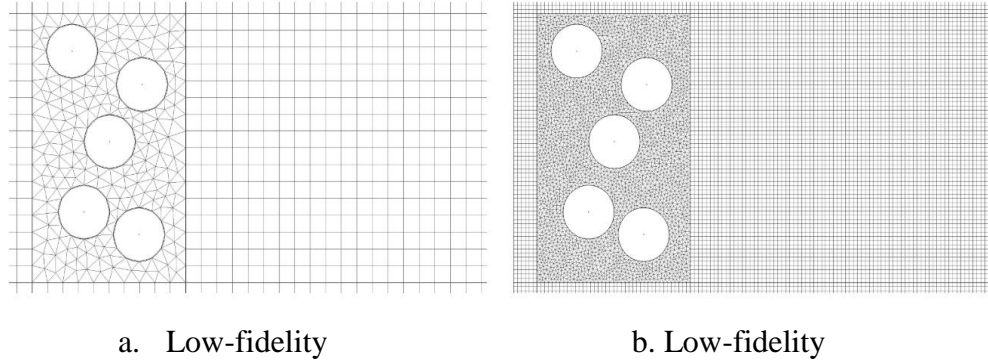
3.7 Model CFD

Untuk mendapatkan data yang akan di gunakan dalam GCNN yang akan diinput pada traning data menggunakan software CFD OpenFOAM. OpenFOAM memiliki berbagai fitur untuk menyelesaikan apa pun mulai dari aliran fluida kompleks yang melibatkan reaksi kimia, turbulensi dan perpindahan panas, hingga akustik, mekanika padat, dan elektromagnetik.(OpenFOAM n.d.) Pada penelitian yang akan dijadikan dalam dataset menggunakan NACA 0012 berikut Hasil simulasi yang akan di training.



Gambar 3.4 Kontur kecepatan pada sudut serang 3° dengan model *turbulen Spalart-Allmaras. (SA)*(Douvi C. Eleni 2012)

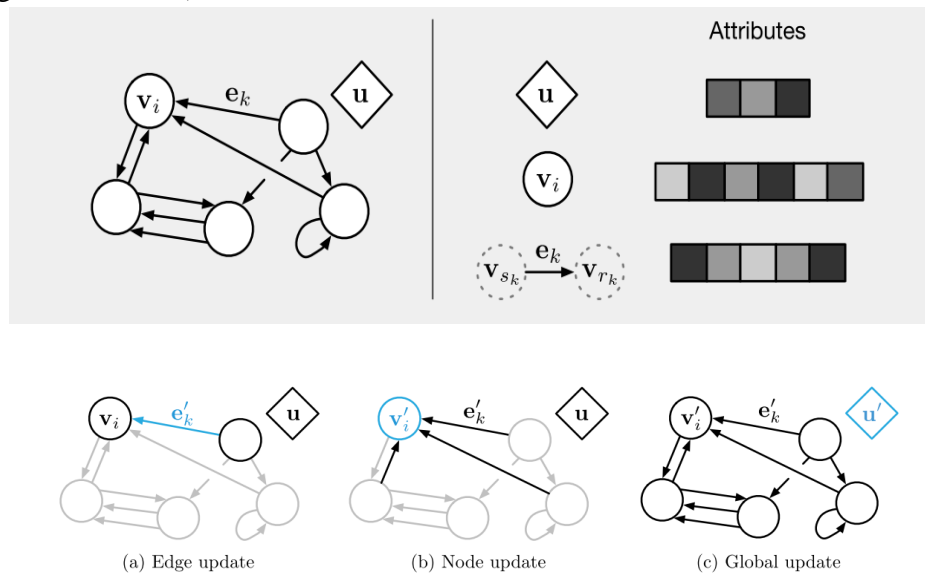
Untuk mendapatkan hasil yang optimal di perlukan Blok Mesh dalam melakukan simulasi pada Openfoam dipengaruhi oleh Block Mesh generator OpenFOAM



Gambar 3.5 low- and high-fidelity CFD simulasi pada OpenFOAM
.(Geneva and Zabaras 2020)

3.7 Graph Convolutional Neural Networks GCNN

Grap terdiri dari tiga stuktur data yang digunakan untuk menyimpan sekumpulan data. $G = (u, V, E)$, u menggambarkan medan gravitasi, V himpunan dari posisi, kecepatan, dan massa. E terdiri dari (e_k, r_k, s_k) e_k , atribut tepi. R_k adalah indeks dari node penerima dan S_k adalah indeks dari node pengirim. (Battaglia et al. 2018)



Gambar 3.6 Graph network (GN) (Battaglia et al. 2018)

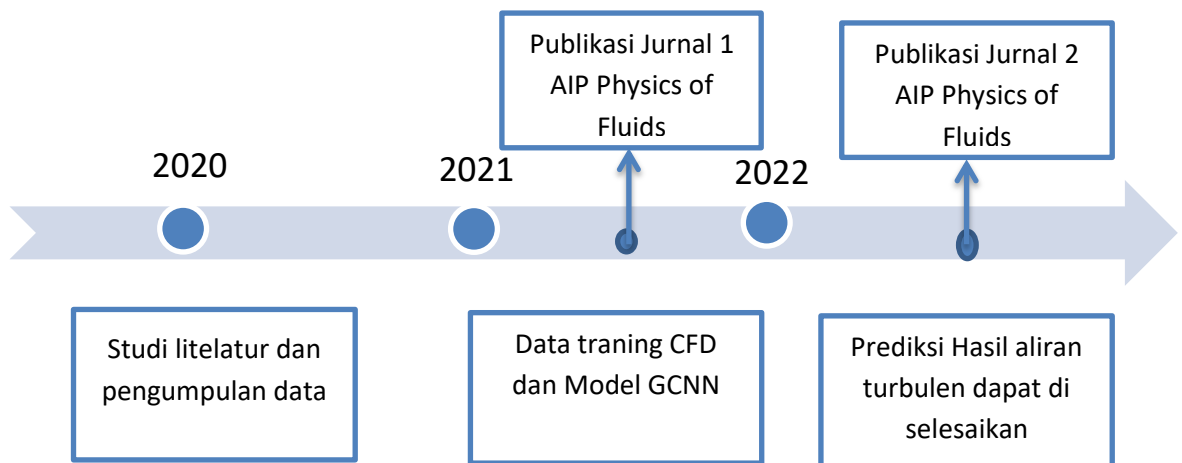
Pada pengumpulan data yang akan dilakukan untuk memprediksi aliran turbulen pada Airfoil didapatkan blok mesh serta data disetiap node yang berupa kecepatan fluida yang dapat di jadikan data input pada GCNN. Metode Graph diharapkan dapat menjawab serta memberikan kontribusi pada penelitian dengan memikirkan hubungan interpolasi setiap node pada Blok Mesh dari hasil simulasi OpenFoam

3.6 Optimasi Parameter

Penelitian yang dilakukan oleh (Chen, Hachem, and Viquerat 2021) pada Prediksi aliran laminar Memiliki beberapa Parameter yang di Optimasi adalah *nodes, edges, flow*, Kecepatan yang di dapatkan dari Hasil simulasi Open Foam.

3.7 Road Map Penelitian

Pada Road Map penelitian prediksi aliran trubulen yang akan dilaksanakan adalah sebagai berikut.



Gambar 3.7 Road Maps Penelitian

DAFTAR PUSTAKA

- Anam, Hujjatul, Aris Budiarto, and Agus Budiyo. 2015. "Design of Diver Propulsion Vehicle Ganendra RI-1 Using SolidWorks Flow Simulation." (December).
- Anand, Prem, and Alathur Srinivasan. *Deep Learning Models for Turbulent Shear Flow*.
- Baldassarre, Federico, and Hossein Azizpour. 2019. "Explainability Techniques for Graph Convolutional Networks."
- Battaglia, Peter W. et al. 2018. "Relational Inductive Biases, Deep Learning, and Graph Networks." : 1–40.
- Bode, Mathis et al. 2019. "Using Physics-Informed Super-Resolution Generative Adversarial Networks for Subgrid Modeling in Turbulent Reactive Flows." *arXiv*.
- Chen, Junfeng, Elie Hachem, and Jonathan Viquerat. 2021. "Graph Neural Networks for Laminar Flow Prediction around Random 2D Shapes." (MI).
- Douvi C. Eleni. 2012. "Evaluation of the Turbulence Models for the Simulation of the Flow over a National Advisory Committee for Aeronautics (NACA) 0012 Airfoil." *Journal of Mechanical Engineering Research* 4(3): 100–111.
- Gaitonde, Arvind T. Mohan* & Datta V. 2018. "A Deep Learning Based Approach to Reduced Order Modeling for Turbulent Flow Control Using LSTM Neural Networks."
- Geneva, Nicholas, and Nicholas Zabar. 2020. "Multi-Fidelity Generative Deep Learning Turbulent Flows."
- Hu, Liwei, Jun Zhang, Yu Xiang, and Wenyong Wang. 2020. "Neural Networks-Based Aerodynamic Data Modeling: A Comprehensive Review." *IEEE Access* 8: 90805–23.
- Keith Agedy. 2021. "No Title." <https://www.aeroengineering.co.id/2020/08/teori-lapisan-batas-boundary-layer/> (November 23, 2021).
- Lee, Jinu, Sangseung Lee, and Donghyun You. 2018. "Deep Learning Approach in Multi-Scale Prediction of Turbulent Mixing-Layer."

- Ling, Julia, Jeremy Templeton, and Andrew Kurzawski. *Reynolds Averaged Turbulence Modeling Using Deep Neural Networks with Embedded Invariance*.
- Liu, Bo, Jiupeng Tang, Haibo Huang, and Xi Yun Lu. 2020. "Deep Learning Methods for Super-Resolution Reconstruction of Turbulent Flows." *Physics of Fluids* 32(2).
- Matai, Racheet. 2018. "LES of Flow over Bumps and Machine Learning Augmented Turbulence Modeling."
- Ogoke, Francis, Kazem Meidani, Amirreza Hashemi, and Amir Barati Farimani. 2020. "Graph Convolutional Neural Networks for Body Force Prediction." (DL): 1–20.
- "OpenFOAM." <https://www.openfoam.com/>.
- Phan, Thien. "Thien Phan's Website." <https://www.phanquochien.org/mesh-geometry/blockmesh/airfoil> (November 20, 2021).
- Piotrowski, Michael G H. 2020. "Smooth Local Correlation-Based Transition Model for the Spalart – Allmaras Turbulence Model." *AIAA Journal*: 1–19.
- R Nave. "No Title." <http://hyperphysics.phy-astr.gsu.edu/hbase/Fluids/airfoil.html> (November 23, 2021).
- Raman, Venkatramanan, Heinz Pitsch, and Rodney O. Fox. 2006. "Eulerian Transported Probability Density Function Sub-Filter Model for Large-Eddy Simulations of Turbulent Combustion." *Combustion Theory and Modelling* 10(3): 439–58.
- Som, S, P K Senecal, and E Pomraning. 2012. "Comparison of RANS and LES Turbulence Models against Constant Volume Diesel Experiments." *24th Annual Conference on Liquid Atomization and Spray Systems, ILASS Americas* Di(May): 1–11.
- Sulistiya, and Alief Sadlie K. 2018. "Validasi M Odel T Urbulensi P Ada S Imulasi N Umerik M Enggunakan S Oftware F Luent D Engan S Ayap O Nera M6 V Alidation Of T Urbulence M Odel In N Umerical S Imulation U Sing F Luent S Oftware W Ith W Ings O Nera M6." *Journal of Aero Technology* 2(1): 21–25.

- T. Henry de Frahan, Marc et al. 2019. “Deep Learning for Presumed Probability Density Function Models.” *Combustion and Flame* 208: 436–50.
- Tompson, Jonathan, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. 2017. “Accelerating Eulerian Fluid Simulation with Convolutional Networks.” *34th International Conference on Machine Learning, ICML 2017* 7: 5258–67.
- Tracey, Brendan, Karthik Duraisamy, and Juan J. Alonso. 2015. “A Machine Learning Strategy to Assist Turbulence Model Development.” In *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics Inc, AIAA.
- Viquerat, Jonathan, and Elie Hachem. 2020. “A Supervised Neural Network for Drag Prediction of Arbitrary 2D Shapes in Laminar Flows at Low Reynolds Number.” *Computers and Fluids* 210.
- Zhao, Yaomin et al. 2020. “RANS Turbulence Model Development Using CFD-Driven Machine Learning.” *Journal of Computational Physics* 411: 109413.

LAMPIRAN

Lampiran 1 Blok Mesh OpenFOAM

```

/*-----*Thien Phan*-----*\
|=====|
|\\ / F ield | OpenFOAM: phanquocthien.org |
|\\ / O peration | Files are generated by Thien Phan |
|\\ / A nd | Web: www.OpenFOAM.com |
|\\ M anipulation | PhanQuocThien@gmail.com |
\*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    object blockMeshDict;
}
//*****//
convertToM 1 ;

geometry
{
}

vertices
(
    ( 0 0 0 ) // 0
    ( 1 0 0 ) // 1
    ( 1 12 0 ) // 2
    ( -11 0 0 ) // 3
    ( 0 0 0,3 ) // 4
    ( 1 0 0,3 ) // 5
    ( 1 12 0,3 ) // 6
    ( -11 0 0,3 ) // 7
    ( 21 1,830271 0 ) // 8
    ( 21 12 0 ) // 9
    ( 21 1,830271 0,3 ) // 10
    ( 21 12 0,3 ) // 11
    ( 1 -12 0 ) // 12
    ( 1 -12 0,3 ) // 13
    ( 21 -12 0 ) // 14
    ( 21 -12 0,3 ) // 15
    ( 1 0 0 ) // 16
    ( 1 0 0,3 ) // 17
);

```

blocks

```

(
    hex (0 1 2 3 4 5 6 7) (
        edgeGrading
        (
            // x-direction expansion ratio
            (
                ( 0,4 0,512195 3,5 )
                ( 0,6 0,487805 0,857143 )
            )
            0,03 0,03
        (
            ( 0,4 0,512195 3,5 )
            ( 0,6 0,487805 0,857143 )
        )
        // y-direction expansion ratio
        (
            ( 0,041667 0,354167 22,18611 )
            ( 0,958333 0,645833 9,014649 )
        )
        (
            ( 0,041667 0,354167 22,18611 )
            ( 0,958333 0,645833 9,014649 )
        )
        (
            ( 0,041667 0,354167 22,18611 )
            ( 0,958333 0,645833 9,014649 )
        )
        (
            ( 0,041667 0,354167 22,18611 )
            ( 0,958333 0,645833 9,014649 )
        )
        // z-direction expansion ratio
        1 1 1 1
    )

    hex (1 8 9 2 5 10 11 6) (
        edgeGrading
        (
            // x-direction expansion ratio
            33,33333 33,33333 33,33333 33,33333
            // y-direction expansion ratio
            (
                ( 0,041667 0,354167 22,18611 )
                ( 0,958333 0,645833 9,014649 )
            )
            13,95817 13,95817
        (
            ( 0,041667 0,354167 22,18611 )
            ( 0,958333 0,645833 9,014649 )
        )
        // z-direction expansion ratio
        1 1 1 1
    )

```

```

hex      (3      12      16      0      7      13      17 4)      (
edgeGrading
(
//      x-direction expansion ratio
0,03
(
(      0,4  0,512195      3,5 )
(      0,6  0,487805  0,857143 )
)
(
(      0,4  0,512195      3,5 )
(      0,6  0,487805  0,857143 )
)
0,03
//      y-direction expansion ratio
(
(      0,958333  0,645833  0,110931 )
(      0,041667  0,354167  0,045073 )
)
(
(      0,958333  0,645833  0,110931 )
(      0,041667  0,354167  0,045073 )
)
(
(      0,958333  0,645833  0,110931 )
(      0,041667  0,354167  0,045073 )
)
(
(      0,958333  0,645833  0,110931 )
(      0,041667  0,354167  0,045073 )
)
//      z-direction expansion ratio
1      1      1      1
)

hex      (12      14      8      16      13      15      10 17)      (
edgeGrading
(
//      x-direction expansion ratio
33,333333  33,333333  33,333333  33,333333
//      y-direction expansion ratio
(
(      0,958333  0,645833  0,110931 )
(      0,041667  0,354167  0,045073 )
)
0,071643  0,071643
(
(      0,958333  0,645833  0,110931 )
(      0,041667  0,354167  0,045073 )
)
//      z-direction expansion ratio
1      1      1      1
)
);

```

edges

```
(
    arc      3 2      (      -7,485281  8,485281      0 )
    arc      7 6      (      -7,485281  8,485281      0,3 )
```

```
spline      1      0
(
    (      0,999035  0,000297      0 )
    (      0,996145  0,001185      0 )
    (      0,991338  0,002653      0 )
    (      0,984631  0,004682      0 )
    (      0,976046  0,007248      0 )
    (      0,965613  0,010321      0 )
    (      0,953367  0,013863      0 )
    (      0,939351  0,017833      0 )
    (      0,923612  0,022187      0 )
    (      0,906208  0,026874      0 )
    (      0,887198  0,031843      0 )
    (      0,866653  0,03704      0 )
    (      0,844647  0,042411      0 )
    (      0,82126  0,047899      0 )
    (      0,796579  0,053449      0 )
    (      0,770697  0,059005      0 )
    (      0,743712  0,064511      0 )
    (      0,715726  0,069913      0 )
    (      0,686848  0,075158      0 )
    (      0,657189  0,080194      0 )
    (      0,626866  0,084971      0 )
    (      0,595998  0,089439      0 )
    (      0,564709  0,093554      0 )
    (      0,533123  0,097269      0 )
    (      0,50137  0,100545      0 )
    (      0,469578  0,103344      0 )
    (      0,437879  0,105631      0 )
    (      0,406404  0,107376      0 )
    (      0,37482  0,108468      0 )
    (      0,343599  0,108714      0 )
    (      0,313001  0,10811      0 )
    (      0,283167  0,10667      0 )
    (      0,254236  0,104416      0 )
    (      0,226341  0,101384      0 )
    (      0,19961  0,097618      0 )
    (      0,174166  0,093172      0 )
    (      0,150121  0,088108      0 )
    (      0,127583  0,082496      0 )
    (      0,106649  0,07641      0 )
    (      0,087405  0,06993      0 )
    (      0,069929  0,063138      0 )
    (      0,054289  0,056114      0 )
    (      0,04054  0,048938      0 )
    (      0,028731  0,041689      0 )
    (      0,018897  0,034435      0 )
    (      0,011066  0,027241      0 )
    (      0,005256  0,020161      0 )
    (      0,001476  0,013241      0 )
    (      -0,000274  0,006513      0 )
)
```

```

spline          5          4
(
(          0,999035  0,000297  0,3 )
(          0,996145  0,001185  0,3 )
(          0,991338  0,002653  0,3 )
(          0,984631  0,004682  0,3 )
(          0,976046  0,007248  0,3 )
(          0,965613  0,010321  0,3 )
(          0,953367  0,013863  0,3 )
(          0,939351  0,017833  0,3 )
(          0,923612  0,022187  0,3 )
(          0,906208  0,026874  0,3 )
(          0,887198  0,031843  0,3 )
(          0,866653  0,03704  0,3 )
(          0,844647  0,042411  0,3 )
(          0,82126  0,047899  0,3 )
(          0,796579  0,053449  0,3 )
(          0,770697  0,059005  0,3 )
(          0,743712  0,064511  0,3 )
(          0,715726  0,069913  0,3 )
(          0,686848  0,075158  0,3 )
(          0,657189  0,080194  0,3 )
(          0,626866  0,084971  0,3 )
(          0,595998  0,089439  0,3 )
(          0,564709  0,093554  0,3 )
(          0,533123  0,097269  0,3 )
(          0,50137  0,100545  0,3 )
(          0,469578  0,103344  0,3 )
(          0,437879  0,105631  0,3 )
(          0,406404  0,107376  0,3 )
(          0,37482  0,108468  0,3 )
(          0,343599  0,108714  0,3 )
(          0,313001  0,10811  0,3 )
(          0,283167  0,10667  0,3 )
(          0,254236  0,104416  0,3 )
(          0,226341  0,101384  0,3 )
(          0,19961  0,097618  0,3 )
(          0,174166  0,093172  0,3 )
(          0,150121  0,088108  0,3 )
(          0,127583  0,082496  0,3 )
(          0,106649  0,07641  0,3 )
(          0,087405  0,06993  0,3 )
(          0,069929  0,063138  0,3 )
(          0,054289  0,056114  0,3 )
(          0,04054  0,048938  0,3 )
(          0,028731  0,041689  0,3 )
(          0,018897  0,034435  0,3 )
(          0,011066  0,027241  0,3 )
(          0,005256  0,020161  0,3 )
(          0,001476  0,013241  0,3 )
(          -0,000274  0,006513  0,3 )
)

```

```

arc      3 12      (      -7,485281 -8,485281      0 )
arc      7 13      (      -7,485281 -8,485281      0,3 )

spline          0          16
(
(      0,002247 -0,006119      0 )
(      0,006409 -0,011672      0 )
(      0,012457 -0,016658      0 )
(      0,020351 -0,021081      0 )
(      0,030046 -0,024945      0 )
(      0,041493 -0,02826      0 )
(      0,054633 -0,031036      0 )
(      0,069404 -0,033287      0 )
(      0,085743 -0,035032      0 )
(      0,103578 -0,036293      0 )
(      0,122838 -0,037096      0 )
(      0,143448 -0,037472      0 )
(      0,165332 -0,037456      0 )
(      0,18841 -0,037089      0 )
(      0,212604 -0,036415      0 )
(      0,237832 -0,035481      0 )
(      0,26401 -0,034339      0 )
(      0,291053 -0,033042      0 )
(      0,318875 -0,031644      0 )
(      0,347385 -0,0302      0 )
(      0,37649 -0,028765      0 )
(      0,406215 -0,027385      0 )
(      0,436788 -0,02594      0 )
(      0,467631 -0,02439      0 )
(      0,49863 -0,022768      0 )
(      0,529667 -0,021106      0 )
(      0,560624 -0,019434      0 )
(      0,591383 -0,017776      0 )
(      0,621824 -0,016155      0 )
(      0,651828 -0,014589      0 )
(      0,681277 -0,01309      0 )
(      0,710053 -0,011669      0 )
(      0,738042 -0,010332      0 )
(      0,76513 -0,009085      0 )
(      0,791206 -0,007927      0 )
(      0,816164 -0,006859      0 )
(      0,839901 -0,005879      0 )
(      0,862316 -0,004983      0 )
(      0,883315 -0,00417      0 )
(      0,902809 -0,003435      0 )
(      0,920716 -0,002777      0 )
(      0,936956 -0,002191      0 )
(      0,95146 -0,001676      0 )
(      0,964163 -0,001232      0 )
(      0,97501 -0,000856      0 )
(      0,983952 -0,000548      0 )
(      0,990949 -0,000308      0 )
(      0,99597 -0,000137      0 )
(      0,998991 -0,000034      0 )
)

```



```

spline          4          17
(
(          0,002247 -0,006119          0,3 )
(          0,006409 -0,011672          0,3 )
(          0,012457 -0,016658          0,3 )
(          0,020351 -0,021081          0,3 )
(          0,030046 -0,024945          0,3 )
(          0,041493 -0,02826          0,3 )
(          0,054633 -0,031036          0,3 )
(          0,069404 -0,033287          0,3 )
(          0,085743 -0,035032          0,3 )
(          0,103578 -0,036293          0,3 )
(          0,122838 -0,037096          0,3 )
(          0,143448 -0,037472          0,3 )
(          0,165332 -0,037456          0,3 )
(          0,18841 -0,037089          0,3 )
(          0,212604 -0,036415          0,3 )
(          0,237832 -0,035481          0,3 )
(          0,26401 -0,034339          0,3 )
(          0,291053 -0,033042          0,3 )
(          0,318875 -0,031644          0,3 )
(          0,347385 -0,0302          0,3 )
(          0,37649 -0,028765          0,3 )
(          0,406215 -0,027385          0,3 )
(          0,436788 -0,02594          0,3 )
(          0,467631 -0,02439          0,3 )
(          0,49863 -0,022768          0,3 )
(          0,529667 -0,021106          0,3 )
(          0,560624 -0,019434          0,3 )
(          0,591383 -0,017776          0,3 )
(          0,621824 -0,016155          0,3 )
(          0,651828 -0,014589          0,3 )
(          0,681277 -0,01309          0,3 )
(          0,710053 -0,011669          0,3 )
(          0,738042 -0,010332          0,3 )
(          0,76513 -0,009085          0,3 )
(          0,791206 -0,007927          0,3 )
(          0,816164 -0,006859          0,3 )
(          0,839901 -0,005879          0,3 )
(          0,862316 -0,004983          0,3 )
(          0,883315 -0,00417          0,3 )
(          0,902809 -0,003435          0,3 )
(          0,920716 -0,002777          0,3 )
(          0,936956 -0,002191          0,3 )
(          0,95146 -0,001676          0,3 )
(          0,964163 -0,001232          0,3 )
(          0,97501 -0,000856          0,3 )
(          0,983952 -0,000548          0,3 )
(          0,990949 -0,000308          0,3 )
(          0,99597 -0,000137          0,3 )
(          0,998991 -0,000034          0,3 )
)

);

faces
(

);

faces
(

);

defaultPatch
{
    name frontAndBack;
    type empty;
}

```

```

boundary
(
inlet      // patch name
{
    type patch; // patch type for patch 0
    faces
    (
        (9 2 6 11) // block face in this patch
        (2 3 7 6 ) // block face in this patch
        (3 12 13 7 ) // block face in this patch
        (12 15 14 13 ) // block face in this patch
    );
}

outlet      // patch name
{
    type patch; // patch type for patch 0
    faces
    (
        (8 9 10 11) // block face in this patch
        (15 8 10 14 ) // block face in this patch
    );
}

walls      // patch name
{
    type wall; // patch type for patch 0
    faces
    (
        (0 1 5 4) // block face in this patch
        (0 4 17 16 ) // block face in this patch
    );
}

interface1      // patch name
{
    type patch; // patch type for patch 0
    faces
    (
        (1 8 10 5) // block face in this patch
    );
}

interface2      // patch name
{
    type patch; // patch type for patch 0
    faces
    (
        (16 17 10 8) // block face in this patch
    );
}

);

mergePatchPairs
(
    ( interface1 interface2 )
);
//*****

```