BAB II LANDASAN TEORI

A. Deskripsi Teori

1. Performa Siswa

Dengan berjalannya zaman berbagai bidang perlu beradaptasi dengan teknologi yang juga terus berkembang, salah satu bidang tersebut adalah pendidikan. Dunia pendidikan sendiri bertujuan untuk mencetak siswa yang mampu berkembang dan beradaptasi dalam dunia setelah mereka selesai dengan pendidikan formal sekolah. Untuk mengetahui perkembangan siswa perlu adanya suatu penilaian yang menghasilkan laporan mengenai sejauh mana perkembangan siswa.

Performa siswa merupakan sebuah tolak ukur yang dianggap penting guna melihat keefektifan sebuah pembelajaran. Performa siswa merepresentasikan sebuah hasil yang mengindikasikan bahwa siswa telah mampu untuk mencapai suatu tujuan misalnya saja tujuan pembelajaran (Steinmayr et al. 2014).

Performa siswa ada dengan tujuan agar seorang guru dapat mengetahui apakah siswa sudah paham ataupun mengerti dengan apa yang guru tersebut ajarkan. Selain itu ketika mengetahui performa dari siswa, hal tersebut dapat membantu siswa untuk meningkatkan kemampuannya sendiri ketika performanya tergolong rendah

Banyak hal yang mampu mempengaruhi performa dari siswa ketika berada di sekolah. Salah satunya adalah tata letak bangunan sekolah. Terdapat hubungan antara tata letak bangunan sekolah dengan performa dari siswa. Ketika siswa berada di sekolah yang dekat dengan alam terdapat peningkatan pada performa siswa.(Matsuoka 2010). Selain tempat, performa dari siswa juga dipengaruhi oleh keadaan sosial seperti kehadiran teman ketika pembelajaran, guru yang mengajar dan hubungan sosialnya. Hal ini terbukti bahwa ketika dalam kelompok belajar siswa dapat mengembangkan konsep dan ide dari diskusi bersama teman ataupun guru mereka (Qureshi et al. 2023). Faktor eksternal seperti aspek masyarakat dan aspek keluarga juga mempengaruhi hasil belajar dari siswa (Hapnita et al. 2018). Tidak

hanya faktor eksternal yang mempengaruhi performa dari siswa, faktor internal juga memiliki pengaruh dalam performa dari siswa. Salah satu dari faktor internal itu adalah motivasi, kepercayaan siswa pada motivasi mampu untuk mempengaruhi hasil performa dari siswa ketika pembelajaran di sekolah (Taheri 2011).

Performa siswa dapat dinilai dari ujian – ujian yang diberikan oleh guru. Ujian tersebut merepresentasikan hasil belajar yang telah para siswa lalui dalam masa pembelajaran. Salah satu ujian tersebut adalah ujian tengah semester (UTS). UTS menjadi bentuk dari komponen penilaian yang dapat merepresentasikan performa siswa (Abdurrahman 2016).

Pengukuran performa siswa dilakukan untuk membantu siswa itu sendiri meningkatkan performanya. Terdapat teknik yang digunakan untuk mengetahui performa dari siswa beberapa diantaranya adalah penilaian yang dilakukan untuk menentukan hasil belajar dari siswa berdasarkan ranah kognitif (Attamimi, Ahmad, dan Al Fajar 2023). Adapun penggunaan teknik *data mining* untuk memprediksi performa siswa. Teknik ini dinamakan sebagai *Educational Data Mining* (EDM). Dengan menggunakan teknik ini dapat membantu guru atau bagian administrasi pendidikan dalam mengambil keputusan yang berdampak pada siswa (Khan dan Ghosh 2021). Tercapainya sebuah hasil atau nilai uts dapat dipengaruhi berbagai faktor, baik itu secara langsung ataupun tidak langsung. Berikut adalah beberapa faktor yang mempengaruhi nilai uts siswa:

a. Demografi Siswa

Salah satu indicator atau faktor yang mempengaruhi hasil dari performa siswa adalah demografi. Demografi adalah sebuah ilmu yang mempelajari jumlah persebaran, teritorial dan komposisi penduduk serta perubahan-perubahannya juga berhubungan dengan sebab-sebab perubahan tersebut, yang biasanya timbul karena peristiwa kelahiran, kematian dan migrasi (perpindahan dari sebuah tempat ke tempat lain yang dirasa lebih baik dari tempat

sebelumnya) serta mobilitas status., penduduk atau manusia terutama tentang kelahiran, kematian dan perpindahan penduduk yang terjadi (Buku UT). Untuk demografi yang ditinjau pada penelitian ini adalah demografi yang dimiliki oleh siswa. Demografi tersebut terdiri dari jenis kelamin, pekerjaan orang tua, tingkat pendidikan orang tua, pekerjaan orang tua, dan penghasilan orang tua (Jati dan Yoenanto 2013). Pemilihan demografi tersebut dikarenakan terdapat perang orang tua yang memiliki peran penting dalam kehidupan anak (Efria Nusruli 2020)

Dengan demografi yang memiliki peran terhadap perkembangan seorang anak. Secara tidak langsung demografi memiliki peran dalam mempengaruhi hasil dari nilai uts seorang siswa. Terdapat penelitian dimana demografi digunakan sebagai salah satu fitur untuk memprediksi performa dari siswa. Ketika demografi dimasukkan sebagai fitur hasil akurasi model sebesar 92%, dan ketika demografi tidak dimasukkan sebagai fitur hasil akurasi model hanya sebesar 76% (Bilal et al. 2022).

b. Kinerja Guru

Guru merupakan komponen penting dalam pembelajaran yang ada di sekolah. Guru merupakan salah satu faktor yang mempengaruhi tinggi rendahnya performa akademik dari siswa (Kipngeno 2018). Proses pembelajaran akan berlangsung dengan lancar dan baik jika guru memiliki kompetensi juga kinerja yang tinggi (Putro dan Rinawati 2012). Guru berperan dalam pembelajaran sebagai informator/ komunikator, organisator, konduktor, motivator, pengarah dan pembimbing, pencetus ide, penyebar luas, fasilisator, evaluator, dan pendidik (Suwardi dan Farnisa 2018).

Kehadiran guru memberikan efek terpenting dari peningkatan nilai uts atau prestasi siswa. Tingkah laku dari seorang guru pastinya akan selalu menjadi perhatian atau bahkan contoh bagi siswanya sehingga guru harus mengedepankan tingkah laku

yang positif dan memiliki moral serta etika yang tinggi. Selain memiliki tingkah laku yang positif guru harus memiliki kemampuan untuk berkomunikasi yang sopan baik itu terhadap orang laing yang memiliki jabatan lebih tinggi ataupun guru lainnya (Wardany dan Rigianti 2023).

Selain cara bersosial guru memiliki kemampuan didalam kelas atau kemampuan pendagogig. Ketika mengajar di dalam kelas guru harus memerankan peran yang nantinya akan membantu siswa untuk mampu memahami pelajaran yang sedang diajarkannya. Sehingga setiap tingkah atau kegiatan yang dilakukan oleh guru akan membuat presepsi tersendiri oleh siswa. Misalnya ketika guru mengajar dengan cara ketat dan tegas pasti akan mempengaruhi presepsi siswa baik itu positif ataupun negative (Sulfemi dan Supriyadi 2018).

2. Machine Learning

Pada tahun 1950-an seorang psikolog dari Universitas Cornell bernama Frank Rosenblett memiliki ide tentang cara kerja jaringan syaraf manusia. Frank membentuk sebuah tim yang menciptakan mesin untuk mendeteksi huruf dari kata Rosenblett. Mesin tersebut dinamakan *perceptron.* Berawal dari mesin inilah prototype dari ANN (*Artificial Neural Network*) tercipta (Fradkov 2020).

Salah satu orang yang berperan dalam pengembangan *machine learning* adalah Arthur Samuel. Arthur merupakan salah satu dari pioneer kecerdasan buatan. Dalam karyanya Arthur mengembangkan sebuah cara bagi komputer untuk mampu belajar dari pengalamannya. Alat yang digunakan untuk melakukan hal tersebut adalah sebuah permainan bernama *checkers* (McCarthy n.d.).

Machine learning merupakan salah satu bagian dari kecerdasan buatan atau *artificial intelligence.* Prinsip kerja dari *machine learning* yaitu diprogram untuk berperilaku secara cerdas seperti manusia, program yang digunakan memampukan *machine learning* untuk menambah pemahamannya melalui pengalaman yang diperoleh system

secara otomatis (Retnoningsih dan Pramudita 2020). Dengan kata lain pemrograman yang dilakukan dalam *machine learning* tidak perlu secara eksplisit untuk memberi kemampuan belajar pada system (Das dan Nene 2017).

Tahapan-tahapan dalam *machine learning* berbeda menurut penggunaan model masing-masing namun secara umum langkahnya terdiri dari. Pengumpulan data, pembersihan data, pemberian label pada data, menentukan fitur, pelatiah model *machine learning*, pengawasan model.

a. Pengumpulan Data

   Hal yang pertama dilakukan untuk menggunakan *machine learning* adalah mengumpulkan dan mengidentifikasi data-data yang relevan. Data yang diperoleh memiliki peran sangat penting untuk memperoleh hasil prediksi.

b. Pembersihan Data

   Kegiatan selanjutnya adalah mengatur atau membersihkan data. Pada kegiatan ini data-data atau informasi yang tidak dibutuhkan akan dibuang. Kegiatan ini dilakukan untuk memperoleh data yang akurat. Sehingga dapat memperoleh akurasi yang tinggi

c. Memilih model

   Pemilihan model dilakukan dengan memperhatikan tujuan awal dari penggunaan *machine learning.* Banyak model telah dibuat oleh berbagai peneliti. Model dipilih berdasarkan kemampuan dan data yang diperoleh.

d. Pelatihan

   Dari data yang sudah diperoleh dan model yang sudah ditentukan maka proses selanjutnya adalah memasukkan data pada model dan melakukan *training.* Proses dilakukan untuk model mengetahui atau mempelajari pola dari data yang dimasukkan

e. Pengetesan

Ketika *training* sudah selesai dan hasilnya diketahui. Langkah selanjutnya adalah melakukan testing atau evaluasi pada data yang dipisahkan sebelumnya pada proses data preparation. Evaluasi digunakan untuk mengetahui keakuratan model pada data yang belum pernah dilihatnya

f. Meningkatkan *Hyperparamater*

Untuk meningkatkan kemampuan dari model dapat dilakukan tuning pada Hyperparameternya seperti batch, weight, bias dan learning ratenya. Untuk mengetahui perbedaan setelah diterapkanya parameter dapat menggunakan teknik *holdout validation.* Teknik ini menggunakan model yang dilatih pada *train set – validation set*

Pengelompokkan *machine learning* dibedakan menjadi dua kategori besar yaitu *supervised learning* dan *unsupervised learning* (Savitri et al. 2021). *Supervised Learning* pada training setnya diberikan *input* dan *output* yang menggunakan informasi yang sudah diketahui. Sehingga jika menggunakan *supervised learning* yang dicari adalah pola dari data tersebut. Sedangkan *Unsupervised Learning* adalah algoritma yang bertugas dalam mengelompokkan dan mengkategorikan data (Abijono, Santoso, dan Anggreini 2021). Pada *Unsupervised Learning* data yang dimiliki belum berlabel.

Machine learning dapat diterapkan dalam berbagai bidang. Misalnya dalam dunia pendidikan dapat digunakan untuk menilai hasil pekerjaan siswa dengan menghilangkan bias pada manusia. Memprediksi kemampuan atau performa dari siswa. Dengan mempelajari dan menemukan kelemahan dari siswa, system dapat menyarankan cara untuk mengembangkan kemampuan siswa seperti mengerjakan latihan soal (Kucak, Juricic, dan Dambic 2018). *Machine learning* juga dapat membantu menyelesaikan masalah lingkungan khususnya menggunakan *supervised learning.* Karena algoritmanya yang dapat menggunakan input baru dan memprediksi output yang berkoresponden. *Machine learning* dirasa lebih baik dari model

statistic yang tradisional. Karena kemampuannya untuk memperlakukan fitur yang banyak dari sumber data (Zhong et al. 2021).

3. **Supervised Learning**

Salah satu jenis atau kategori dari *machine learning* adalah *supervised learning.* Perbedaan antara kategori dalam *machine learning* terletak pada data dan algoritmanya. *Supervised learning* memiliki algoritma dengan proses memetakan *input* pada *output* yang diinginkan (Nasteski 2017). Data yang dilatih pada *supervised learning* sudah tersedia dan setiap kelas memiliki masing-masing data untuk dilatih meskipun kelas tersebut terbatas (Feldman 2013).

Pada model *Supervised learning, training set* terdiri dari n dengan *order* $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, dimana x_i adalah suatu perhitungan atau kategori dalam data, dan y_i adalah label dari data tersebut. Sebagai sebuah contoh x adalah lima kategori dari penjualan rumah, kategori tersebut terdiri dari, jarak, kondisi, usia, luas, dan fasilitas, kemudian y akan berperan menjadi harga yang akan diprediksi, harga tersebut akan mulai dari 10 juta hingga 100 juta.

Metode-metode yang tergolong sebagai *supervised learning* diantaranya adalah *K-Nearest Neighbor, Naive Bayes Classifier*, *Support Vector Machine*, *Neural Network, Rdanom Forest Classifier*, *Decision Tree*, dan masih banyak lagi. Metode tersebut memiliki kelebihan dan kelemahannya masing-masing tergantung pada penggunanya (Argina 2020).

Model *supervised learning* memiliki banyak kegunaan salah satunya adalah penerapannya dalam memprediksi masa depan. Sebagai contohnya adalah metode *Artificial Neural Network* yang sukses melakukan prediksi dengan data yang bersifat non linear berjenjang waktu (*nonlinear time series*) pada eksperimen yang dilakukan oleh Lapedes (Bontempi, Ben Taieb, dan Le Borgne 2013).

4. **Fungsi Aktivasi**

Fungsi Aktivasi akan dimplementasikan pada setiap neuron. Neuron sendiri adalah media penyampaian informasi layaknya neuron asli didalam otak, namun berbeda dengan neuron yang ada di otak, neuron di dalam jaringan syaraf tiruan akan diaktifkan menggunakan fungsi aktifasi. Fungsi ini memiliki tugas untuk menentukan apakah neuron tersebut harus aktif atau tidak. Pada dasarnya fungsi aktifasi adalah persamaan matematika yang memiliki kemampuan untuk menentukan aktif atau tidaknya suatu neuron dalam jaringan syaraf tiruan. Pada penelitian ini akan digunakan duan fungsi pada LSTM yaitu :

a. Fungsi Sigmoid

Fungsi Sigmoid memiliki kewenangan untuk aktivasi dari neuron. Ketika input dimasukkan makan funsi akan menghasilkan output dengan nilai antara 0 hingga 1. Jika 0 maka tidak akan terjadi aktvasi dan sebaliknya. Fungsi ini memiliki kurva dengan bentuk S

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$x =$ Input

$\sigma(x)$ = Fungsi Aktivasi Sigmoid

b. Fungsi TANH (*Hyperbolic Tangent Function*)

Fungsi TANH memiliki keunggulan dibdaningkan dengan fungsi sigmoid. Dikarenakan fungis ini mempu melakukan konvergensi lebih cepat daripada fungsi sigmoid. Dengan konvergensi yang lebih cepat fungsi ini juga mampu menghasilkan nilai akurasi yang lebih tinggi. Fungsi ini memiliki nilai output antara -1 dan 1.

$$TANH = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$x =$ input

$TANH$ = Fungsi Aktivasi Hyperbolic Tangent

5. **NN (Neural Network) atau ANN (Artificial Neural Network)**

Pada tahun 1943 seorang psikolog bernama Mcculloh dan matematikawan bernama Pitts mengajukan model bernama Multi Layer Perceptron. Model tersebut memiliki algoritma yang bekerja dengan mempertimbangkan *neural network* sebagai suatu mesin dengan fungsi logika.

Kemudian pada tahun 1957 Frank Rosenblett mengajukan sebuah model berdasarkan model M-P dengan sebutan *Perceptron.* Model *Perceptron* menggunakan prinsip dasar dari *neural network* modern yang digunakan saat ini dan struktur yang memiliki alur sama dengan neuropsikolog. Meskipun sangat sederhana *Perceptron* merupakan *neural network* yang pertama (Wu dan Feng 2018).

Neural Network adalah kombinasi paralel raksasa yang berisi unit proses sederhana. Unit tersebut dapat memperoleh pengetahuan menggunakan proses pembelajaran. Ketika unit sudah mendapatkan pengetahuan kemudian pengetahuan tersebut akan disimpan dalam hubungan tiap unit (Taylor 1999).

ANN diciptakan berdasarkan jaringan syaraf manusia namun ANN masih belum mampu untuk meniru secara 100%. Aktivitas *neuron* biologis milik manusia dapat dibdaningkan dengan ANN seperti berikut. Synapses berperan sebagai *weight* dari stimulus yang datang. Hal tersebut memberikan inspirasi untuk memberikan *weight* pada ANN. Dendrit bertugas untuk mengumpul *weighted* dari stimulus. Dalam ANN dendrit menjadi inspirasi untuk pembentukan fungsi penjumlahan. Tubuh sel bertugas untuk mengkonversi stimulus yang sudah dikumpulkan oleh dendrit menjadi stimulus yang baru. Dalam ANN tubuh sel menjadi inspirasi pembentukan fungsi aktivasi. Axon berperan untuk mengantarkan stimulus baru pada neuron yang bersangkutan. Dalam ANN axon adalah output dari ANN. Kemudian terdapat bagian yang memiliki peran untuk mengatur stimulus, baik itu menaikkan maupun menurunkan stimulus. Bagian ini dalam ANN adalah Bias (Guresen dan Kayakutlu 2011).

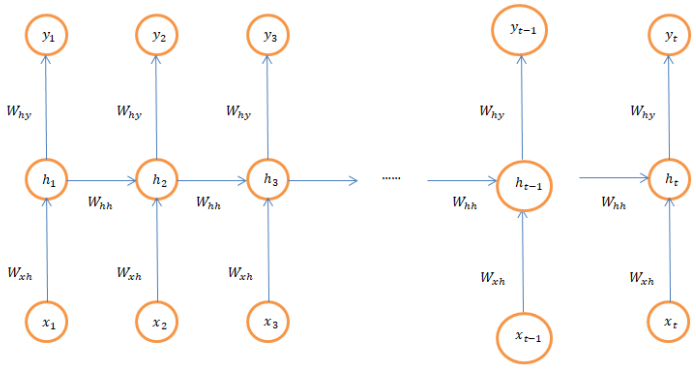
6. **RNN (Recurrent Neural Network)**

Recurrent Neural Network merupakan salah satu cabang dari jaringan syaraf tiruan (*Neural Network*) yang salah satu tujuan pentingnya adalah memproses data bersifat *sequential* (Faturohman, Irawan, dan Si 2020). Arsitektur dari RNN memiliki kemampuan untuk menyimpan memori atau informasi yang lalu. Sehingga RNN cocok digunakan untuk data berjenjang waktu atau *time series* (Pascanu et al., n.d.)

Konsep RNN diperkenalkan pertama kali pada tahun 1986. Model RNN yang mampu menyimpan memori atau informasi yang lalu membuatnya banyak digunakan dalam pemrosesan bahasa alami (*Natural Language Processing*) atau *machine translation.*

Menurut (Hermans & Schrauwen, n.d.) secara umum RNN menerima input dan memproduksi output setiap waktunya. Arsitektur dari model RNN terdiri dari tiga lapisan yaitu *input layer*, *hidden layer*, dan *output layer* (Yin et al. 2017). Ketika terdapat informasi baru yang diberikan, informasi tersebut akan diproses secara satu arah hingga sampai pada *output layer.*

Gambar 1 Diagram RNN



(Sumber : Dokumen Penulis)

$$h_t = tanh(W_{hh}h_{t-1} + W_{hx}x_t)$$

$y_t = W_{hy}h_t$

$x_t =$ *input* pada waktu t

$y_t =$ *output* pada waktu t

$h_t =$ *hidden layer* pada waktu t

W_{xh} = matriks *weight* pada tiap layar input

W_{hy} = matriks *weight* pada tiap layar output

W_{hh} = matriks *weight* pada tiap layar tersembunyi (*hidden layer*)

Pada RNN setiap simpul terhubung satu sama lain dalam satu arah, sehingga dapat dijelaskan bahwa h_2 bergantung dengan x_1 dan h_1. Dapat disimpulkan bahwa simpul pada *hidden layer* merupakan hasil atau keputusan dari simpul pada *hidden layer* sebelumnya

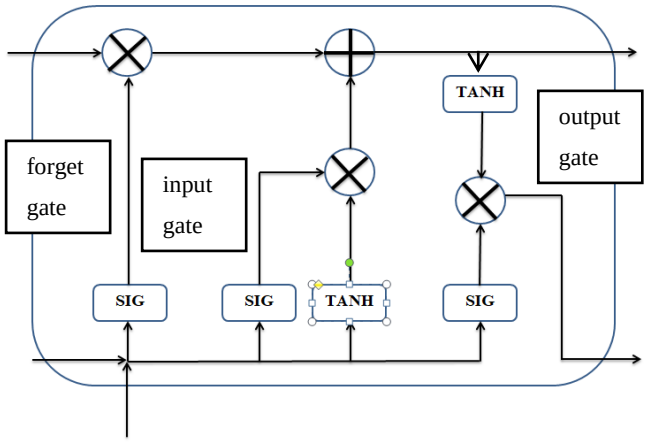
7. **LSTM (Long Short Term Memory)**

Long term dependencies atau masalah ketergantungan jangka panjang merupakan permasalahan jarak antara data yang relevan letaknya terlalu jauh sehingga model tidak mampu memberikan informasi yang relevan. Kelemahan tersebut dimiliki oleh RNN. Berangkat dari masalah tersebut diciptakanlah suatu model yang memiliki kemampuan untuk menyimpan informasi relevan tanpa terciptanya jarak yang terlalu antara input data.

LSTM (*Long Short Term Memory*) dikembangkan oleh Hochreiter dan Schmidhuber pada tahun 1997. Dengan kapasitas belajar yang kuat LSTM banyak digunakan dalam berbagai permasalahan melibatkan data bersifat *sequential,* contohnya adalah *speech recognition, sentence embedding, trajectory prediction* dan masih banyak lagi.

Cara kerja atau struktur LSTM terdiri atas gerbang yang akan mengolah informasi atau data yang dimasukkan. Gerbang tersebut terdiri dari *forget gate, input gate,* dan *output gate*

Gambar 2 LSTM



(Sumber : Dokumentasi Penulis)

Fungsi Sigmoid

Merupakan salah satu fungsi aktivasi

$$sig = \sigma = \frac{1}{1 + e^{-x}}$$

Fungsi TANH

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Forget gate

Merupakan sebuah gerbang dalam arsitektur LSTM yang memutuskan suatu informasi atau data akan dihapus atau tidak. Keputusan diambil berdasarkan *weight* atau matriks bobotnya, *hidden state* atau keluaran sebelumnya (h_{t-1}) dan *input* yang dimasukkan

$$f_t = \sigma\left(W_f x_t + W_f h_{t-1} + b_f\right)$$

Dimana:

f_t = *Forget gate* untuk setiap order t

σ = Fungsi aktivasi *sigmoid*

W_f = *weight* (matriks bobot) pada *forger gate*

x_t = input pada setiap orde t

h_{t-1} = *hidden state* atau output dari proses sebelum orde t

b = bias pada orde t

Input gate

Input gate merupakan gerbang yang memutuskan berapa banyak informasi atau data baru yang dimasukkan pada kondisi sementara *(cell state),* Keputusan tersebut diambil memperhatikan nilai dari matriks bobot pada input gate, *hidden state* atau keluaran sebelumnya (h_{t-1}) dan *input* yang dimasukkan. *Input gate* terdiri dari dua persamaan yaitu :

$$i_t = \sigma(W_i x_t + W_i h_{t-1} + b_{hi})$$

dimana:

$i_t =$ *input gate*

$\sigma =$ fungsi aktivasi sigmoid

$W_i =$ *weight* pada *input gate*

$x_t =$ nilai input untuk setiap orde t

$h_{t-1} =$ *hidden state* atau output dari proses sebelum orde t

$b_{hi} =$ bias pada input gate

$$c_t = tanh(W_c x_t + W_c h_{t-1} + b_c)$$

dimana

$c_t =$ *cell state* untuk setiap orde t, merupakan nilai memori sementara dalam waktu t

$tanh =$ fungsi aktivasi $tanh$

$W_c =$ *weight* pada *cell state*

$x_t =$ nilai input untuk setiap orde t

$h_{t-1} =$ *hidden state* atau output dari proses sebelum orde t

$b_c =$ bias pada cell state

Output gate

Output gate merupakan gerbang yang memutuskan apakah *cell state* dari gerbang sebelumnya akan mempengaruhi *output* yang ada. Keputusan itu diambil berdasarkan nilai dari matriks bobot pada input gate, *hidden state* atau keluaran sebelumnya (h_{t-1}) dan *input* yang dimasukkan.

$$o_t = \sigma(W_o x_t + W_o h_{t-1} + b_o)$$

$o_t =$ *output gate* untuk setiap orde t

$\sigma =$ fungsi aktivasi sigmoid

W_o = *weight* pada *output gate*

x_t = nilai input untuk setiap orde t

h_{t-1} = *hidden state* atau output dari proses sebelum orde t

b_o = bias pada *output gate*

LSTM bekerja dengan memasukkan input baru ke dalam masing - masing gerbang. Terdapat tiga input ketika menggunakan metode LSTM, input pertama merupakan input baru (x_t) atau data baru yang dimasukkan ke dalam metode, input kedua merupakan *hidden state* (h_{t-1}) yang memiliki peran menyimpan memori jangka pendek, untuk penghitungan paling awal *hidden state* akan bernilai 0, input ketiga adalah *cell state* (C_t) yang memiliki peran sebagai penyimpan memori jangka panjang. Dibawah ini merupakan penjelasan mengenai gambar 2

1. Tahap pertama pada alur kerja metode LSTM adalah menghitung *cell state* baru yang merupakan hasil perhitungan dari *forget gate* yang kemudian dikalikan dengan *cell state* yang lama.
2. Tahap kedua adalah melakukan penggabungan (+) antara *cell state* yang telah melalui *forget gate* dengan perhitungan yang dilakukan pada *input gate*, hasil dari penggabungan ini akan menjadi *cell state* yang terbaru atau memori jangka panjang yang akan digunakan dalam penghitungan menggunakan *input* baru.
3. Tahap ketiga adalah menggunakan *cell state* yang terbaru untuk menentukan *hidden state* atau memori jangka pendek yang terbaru, perhitungan tersebut dimulai dengan memasukkan *cell state* pada fungsi TANH, kemudian hasilnya akan dikalikan (×) dengan hasil dari perhitungan *output gate*, sehingga dari perkalian tersebut akan menghasilkan *hidden state* atau memori jangka pendek yang terbaru. Alur kerja pada metode LSTM secara jelas seperti pada gambar 2.

8. **Decision Tree**

Decision tree merupakan salah satu metode yang masuk kategori dari *supervised learning.* Metode ini memetakan domain data menjadi himpunan respon. Metode ini membagi domain menjadi dua subdomain secara rekursif. Pembagian ini menghasilkan *split* yang memiliki *information gain* lebih tinggi daripada domain data. Memperoleh *split* yang memiliki *information gain* secara maximal adalah tujuan optimasi algoritma pada *decision tree* (Suthaharan 2016).

Domain data yang pertama sebelum dipetakan dinamakan *root node.* Cabang dari *node* akan mengarah kekanan maupun kekiri berdasarkan hasil tes. *Decision tree* dengan tipe non binary juga sering digunakan. Pada penggunaanya lebih dari dua cabang boleh dipetakan dari *node* namun hanya boleh satu cabang yang masuk dalam *node* (Jacobsen, Zscherpel, dan Perner 1999).

Penerapan dari *Decision tree* digunakan dalam berbagai bidang contohnya dalam melakukan prediksi dan melakukan klasifikasi. Algoritma dari *decision tree* bersifat *non parametric* sehingga akan mudah untuk menangani data yang berukuran besar. Dengan adanya data yang berukuran besar maka data dapat dibagi menjadi training set dan validation set. Training set digunakan untuk mengembangkan model dari *decision tree* sedangkan *validation set* digunakan untuk pengembangan model yang paling optimal

9. **Extreme Gradient Boost**

Extreme Gradient Boosting pertama kali diperkenalkan pada tahun 2001 oleh Jerome H Friedman dari Universitas Harvard pada waktu seminar yang beliau lakukan(Friedman 2001). Cara kerja dari algoritma Extreme Gradient Boost adalah menggunakan kumpulan *decision tree* untuk membuat sebuah keputusan atau prediksi. Algoritma ini tergolong sebagai *supervised learning* dalam *machine*

learning. Artinya algoritma ini bekerja menggunakan data yang sudah memiliki label.

Extreme Gradient Boost merupakan salah satu algoritma *machine learning.* Extreme Gradient Boost tergolong sebagai klasifikasi ataupun regresi yang menggunakan konsep *boosting.* Konsep tersebut merupakan proses iterative yang dilakukan untuk memperkuat *tree* lemah menjadi lebih baik performanya (Pascanu et al., n.d.).

Extreme Gradient Boost memiliki alur kerja dengan meningkatkan error pada residual atau hasil sebelumnya. Hal ini berbeda dengan *Rdanom Forest* yang hanya menentukan satu yang paling memenuhi persyaratan. Dari penjumlahan nantinya akan ditemukan hasil yang paling maksimal (Wang et al. 2019).

Alur kerja XGBoost

1. Menentukan nilai awal dari *base model* atau prediksi awal, dengan cara menentukan rata – rata dari target yang akan diprediksi

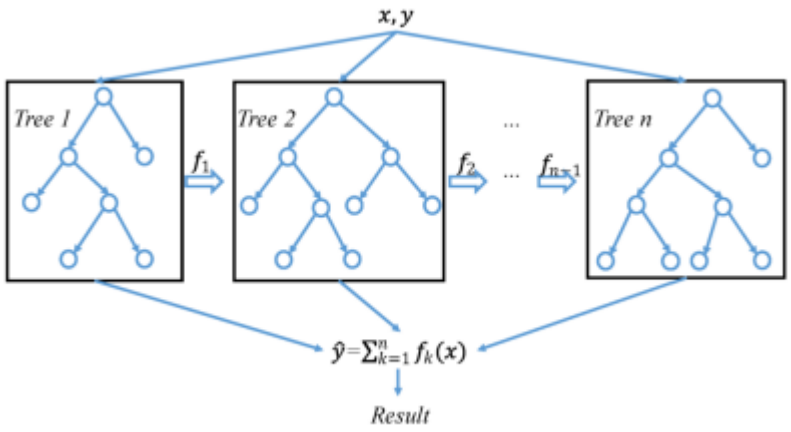
   $$Base\ model = Nilai\ rata - rata\ dari\ target\ prediksi$$

2. Menentukan nilai residual dengan mengurangkan nilai target pada setiap data dengan *base model*

   $$residual = \ Base\ model - nilai\ asli\ target\ prediksi$$

3. Membuat regression tree dengan menggunakan semua residual yang dimiliki diletakkan pada satu *leaf*

Gambar 3 Extreme Gradient Boost



Sumber : Dokumen Penulis

Bentuk lingkatan dalam gambar 3 merupakan *leaf* , dari *leaf* yang paling awal akan dibuat cabang – cabangnya berdasarkan fitur yang dimiliki

4. Menghitung similiarity skor pada setiap leaf

$$SS = \frac{(Sum\ of\ residual)^2}{No.of\ residual + \lambda}$$

5. Menghitung nilai gain pada setiap leaf

$$Gain = (SS_l + SS_r) - SS_{root}$$

6. Menghitung output dari setiap leaf

$$Ov = \frac{Sum\ of\ residual}{No.of\ residual + \lambda}$$

7. Membuat base model baru dengan cara mengalikan nilai output dengan *learning rate* dan ditambahkan dengan base model yang lama

$$Prediction = Base\ score + \rho.(Ov)$$

$\rho = learning\ rate$ merupakan nilai yang berfungsi untuk mengontrol nilai *output*

8. Mengulangi langkah – langkah dari nomer 4 pembuatan *tree* dengan kondisi fitur yang berbeda guna memunculkan nilai residual seminimal mungkin. Seperti yang ada pada gambar 3 dibuat beberapa *tree,* yang akan selalu memiliki nilai lebih baik dari *tree* sebelumnya

Seperti yang telah diterangkan sebelumnya bahwa XGBoost menggunakan prinsip dari pohon keputusan untuk melakukan prediksi atau klasifikasi. Alur kerja dari XGBoost secara sederhana terdiri dari tiga tahap yaitu menentukan *base model* dari target yang akan diprediksi, tahap kedua membuat pohon keputusan, tahap ketiga memperbarui setiap output.

10. **Penghitungan Akurasi**

Pada penelitian ini digunakan penghitungan akurasi yang hanya berfokus pada penyebaran eror tanpa memperhatikan bias (Steurer, Hill, dan Pfeifer 2021). Perhitungan akurasi digunakan untuk mengetahui seberapa besar performa dari model *machine leaerning.*

Dimana kelayakan performa dari model *machine learning* bergantung pada akurasi model (Garosi et al. 2019). Matriks yang digunakan untuk mengetahui performa model adalah akurasi, presisi, dan *recall* (Ming Yin, Jennifer Wortman Vaughan, dan Hanna Wallach 2019). Beberapa matriks yang digunakan pada penelitian ini meliputi, :

a. RMSE (*Root Mean Square Error*)

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}}$$

y_i = Nilai sebenarnya

$\hat{y}_i$ = Nilai hasil peramalan

n = Jumlah data

b. MSE (*Mean Squarred Error*)

$$RMSE = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}$$

y_i = Nilai sebenarnya

$\hat{y}_i$ = Nilai hasil peramalan

n = Jumlah data

11. **Hyperparamater**

Hyperparameter adalah suatu parameter yang memiliki pengaruh untuk meningkatkan performa dari model prediksi. *Hyperparameter* ditetapkan sebelum proses pembelajaran dilangsungkan dalam kata lain yaitu sebelum proses *training* dilaksanakan. *Hyperparameter* digunakan untuk membuat model *machine learing* yang terbaik (Danonie 2019). Berikut beberapa contoh dari *hyperparameter* yang digunakan dalam penelitian ini:

1. Jumlah lapisan

   Menentukan jumlah lapisan pada model penting untuk dilakukan karena sedikit ataupun banyak lapisan yang ada pada model prediksi akan menentukan performa atau tingkat akurasi dari model

2. Jumlah neuron layer per lapisan

Neuron berada di dalam lapisan. Neuron sama pentingnya dengan lapisan. Sehingga jumlahnya menentukan performa dari model prediksi. Neuron pada *machine learning* terinspirasi dari cara kerja neuron yang ada pada otak (Fan, Cong, dan Wang 2018).

3. *Learning rate*

   Setiap waktu ketika *weight* mengalami perubahan maka model prediksi juga akan mengalami perubahan. *Learning rate* lah yang memiliki kemampuan untuk menentukan seberapa jauh model akan berubah untuk merespon eror yang ada ketika *weight* mengalami perubahan

4. *Optimizer*

   Optimizer memiliki kegunaan untuk mengatur *weight* dan *bias* pada model. Optimizer membantu model unutk meningkatkan akurasi serta meminimalisir *loss function* (fungsi yang menghitung performa model ketika pada *training*)

5. *Maximum depth*

   Max depth adalah jumlah cabang yang dapat muncul dari pohon keputusan sampai phon tersebut dihentikan. Jika jumlah dari cabangnya semakin banyak maka pohon keputusan semakin dalam. Hal ini menyebabkan model akan semakin kompleks

6. *N_Estimator*

   N_estimator adalah jumlah iterasi boosting atau jumlah *tree* yang akan digabungkan diakhir penggunaan metode XGBoost. Semakin besar jumlah *n_estimator* yang digunakan akan semakin bagus namun akan memperbesar terjadinya model terlalu baik dalam memahami data sehingga ketika data baru dimasukkan.

12. Google Collaboratory

Google Colaboratory atau sering disebut Colab merupakan salah satu produk dari Google yang menggunakan pelayanan berbasis cloud atau

serverless. Colab sering digunakan untuk penelitian atau pembelajaran yang menggunakan *machine learning.* Colab sering digunakan karena keunggulannya yang memiliki akses GPU yang menjadi dasar dari system komputasi (Carneiro et al. 2018).

Meskipun Colab merupakan platform yang *open source* terdapat batasan dalam penggunaanya dimana, ketika batas waktu yang digunakan telah habis, program yang sudah dijalankan akan direset, sehingga untuk membuat colab tetap menjalankan program yang digunakan harus membayar biaya pada google (Nelson dan Hoover 2020).

13. Tensorflow

Tensorflow adalah sebuah paket atau *library* dari Google yang merupakan platform *open source* atau dapat digunakan oleh berbagai orang dengan gratis. Tensorflow memiliki berbagai fitur yang sangat mendukung pengembang atau peneliti dalam menerapkan *machine learning*, baik itu digunakan dalam lingkungan penelitian atau pengembangan aplikasi