BAB II
TINJAUAN PUSTAKA

2.1 Kajian Terkait

Penelitian yang dilakukan oleh Solechan & Shinta (2012) dengan judul ”Kajian Komparasi Artificial Neural Network dan Regresi Linear Dalam Memprediksi Harga Saham Dengan Mempertimbangkan Faktor Fundamental Pada Sektor Industri” melakukan perbandingan hasil antara model yang menggunakan *linear regresion* dan *neural network*. Penelitian ini menggunakan tiga variabel bebas (profitabilitas, rasio pasar dan solvabilitas) untuk menganalisis pengaruhnya terhadap *label* (harga saham) yang akan diprediksi. Pada penelitian ini, hasil yang diperoleh adalah model yang dibangun dengan menggunakan *Neural Network* memiliki hasil yang lebih akurat dari pada model linear regresion. Hal ini dapat dilihat dari nilai MSE sebesar 2,03808 lebih kecil apabila dibandingkan dengan menggunakan linear regression yang menghasilkan nilai MSE sebesar 2,46.

Penelitian lainnya yang dilakukan oleh Masruroh (2020) dengan judul ”Perbandingan Metode Regresi Linear dan *Neural Network Backpropagation* Dalam Prediksi Nilai Ujian Nasional Siswa SMP Menggunakan *Software R*” melakukan perbandingan nilai akurasi yang dihasilkan oleh model yang dibangun dengan menggunakan algoritma *Multiple Linear Regression* dan *Neural Network*. Model prediksi tersebut dibangun untuk memprediksi nilai ujian nasional siswa SMP. Dari penelitian ini, didapat bahwa model yang dibangun menggunakan metode *Neural Network* memiliki tingkat akurasi yang lebih baik dibanding model yang dibangun menggunakan *Multiple Linear Regression*. Dengan nilai RMSE sebesar 7.28 dan MAPE sebesar 0.55% untuk model yang dibangun dengan *Neural Network*. Sedangkan, model yang dibangun menggunakan *Multiple Linear Regression* menghasilkan nilai RMSE sebesar 9.04 dan MAPE sebesar 3.94%.

Penelitian dengan judul ”*Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model*” yang

dilakukan oleh Rath dkk (2020) melakukan analisa terhadap statistik harian orang yang terpapar covid-19 dan menghitungnya untuk melakukan prediksi terhadap angka *new active cases* di Odisa dan India menggunakan *Linear Regression* dan *Multiple Linear Regression*. Pada penelitian ini, *feature* yang digunakan untuk memprediksi *label* pada model *Linear Regression* adalah kasus positif harian. Model ini menghasilkan nilai R2 sebesar 0.995588 (Odisha) dan 0.974855 (India). Sedangkan, pada *Multiple Linear Regression, feature* yang diguanakan untuk memprediksi *label* adalah kasus positif harian, kesembuhan dan kematian. Model ini menghasilkan nilai R2 sebesar 0.9985 (Odisha) dan 1.0 (India). Dari kedua skenario yang dilakukan, model yang dibangun diindikasi dapat memprediksi angka *new active cases* pada hari berikutnya dengan akurat. Dari kedua model tersebut, kemudian dilakukan perbandingan antara nilai akurasi yang dihasilkan. Pada penelitian ini, model *Multiple Linear Regression* dianggap lebih baik jika dibandingkan dengan model *Linear Regression* dalam kasus prediksi angka *new active cases* di Odisha dan India.

Penelitian dengan judul ”*Prediction of COVID-19 New Cases Using Multiple Linear Regression Model Based on May to June 2020 Data in Ethiopia*” yang dilakukan oleh Argawu, dkk (2021). Dimana pada penelitian tersebut dilakukan analisa terhadap *feature* yang berkorelasi dengan *label* menggunakan *pearson’s correlation coefficient.* Dari analisa tersebut didapat bahwa terdapat korelasi positif antara *new cases* dengan *number of days, daily laboratory tests, new cases of males, new cases of females, new cases from* ADDIS ABABA *city* dan *new cases from foreign natives*. Model *Multiple Linear Regression* yang dibangun pada penelitian ini memiliki nilai *score F* yang tinggi yaitu sebesar 120.7 dan nilai MSE sebesar 637.4.

Penelitian dengan judul ”Perbandingan Algoritma Regresi Linier dan Regresi *Random Forest* Dalam Memprediksi Kasus Positif Covid-19” yang dilakukan oleh Fachid & Triayudi (2022). Penelitian ini menerapkan metode *Random Forest Regression* dan *Linear Regression* untuk memprediksi angka kasus positif Covid-19. Metode *feature corellation* digunakan untuk menentukan *feature* yang akan digunakan pada penelitian ini. Dari pemilihan *feature* tersebut

terdapat 6 variabel yang digunakan untuk membangun model, yaitu kasus aktif, kasus aktif (%), sembuh, sembuh (harian), meninggal, dan meninggal (harian). Hasil dari penelitian yang dilakukan yaitu model *Regresi Linear* menghasilkan tingkat akurasi sebesar 94% dengan RMSE sebesar 3031.127 dan MAPE sebesar 47.66. sedangkan pada model *Random Forest* yang dilakukan *tuning* pada *hyperparameter max_depth* = 10 dan *n_estimator* = 100 menghasilkan tingkat akurasi sebesar 97.7% dengan RMSE sebesar 1886.555 dan MAPE sebesar 14.85.

Tabel 2. 1 Rangkuman Kajian Terkait

Peneliti	Judul	Metode	Kesimpulan
Solechan & Shinta (2012)	Kajian Komparasi Artificial Neural Network dan Regresi Linear Dalam Memprediksi Harga Saham Dengan Mempertimbangkan Faktor Fundamental Pada Sektor Industri	*Multiple Linear Regression* dan *Neural Network*	Menggunakan 3 variabel bebas untuk membangun model prediksi dan menghasilkan nilai MSE sebesar 2,03808 untuk *Multiple Linear Regression*, sedangkan untuk *Neural Network* nilai MSE sebesar 2.46
Masruroh (2020)	Perbandingan Metode Regresi Linear Dan Neural Network Backpropagation Dalam Prediksi Nilai Ujian Nasional Siswa Smp Menggunakan Software R	*Multiple Linear Regression* dan *Neural Network*	Nilai akurasi model Neural Network, RMSE sebesar 7.28 dan MAPE sebesar 0.55% . Nilai akurasi model *Multiple Linear Regression*, RMSE sebesar 9.04 dan MAPE sebesar 3.94%.
Rath et al. (2020)	*Prediction of new active cases of coronavirus disease (COVID-19) pandemic using multiple linear regression model*	*Multiple Linear Regression*	Membangun model yang dgunakan untuk memprediksi angka *new active cases* di Odisha dan India, menggunakan *Linear Regression* dan *Multiple Linear Regression*. Model *Linear Regression* memiliki nilai R2 sebesar 0.995588 (Odisha) dan 0.974855 (India). Sedangkan, model *Multiple Linear Regression* memiliki nilai R2 sebesar 0.9985 (Odisha) dan 1.0 (India). Model *Multiple Linear Regression* dianggap lebih baik daripada model *Linear Regression*.

Argawu et al. (2021)	*Prediction of COVID-19 New Cases Using Multiple Linear Regression Model Based on May to June 2020 Data in Ethiopia*	*Multiple Linear Regression*	Menggunakan 6 variabel bebas yang dipilih berdasarkan *pearson corellation.* Model ini menghasilkan nilai MSE sebesar 637.4.
Fachid & Triayudi (2022)	Perbandingan Algoritma Regresi Linier dan Regresi Random Forest Dalam Memprediksi Kasus Positif Covid-19	*Linear Regression* dan *Random Forest*	Menggunakan dua variabel yang dipilih berdasarkan *feature corellation.* Tingkat akurasi model Regresi Linear yaitu 94%, MAPE sebesar 47.66 dan RMSE sebesar 3031.127. Sedangkan, tingkat akurasi model *Random Forest* yang dilakukan *hyperparameter tuning* sebesar 97.7% dengan RMSE sebesar 1886.555 dan MAPE sebesar 14.85.

2.2 Landasan Teori

2.2.1 Machine Learning

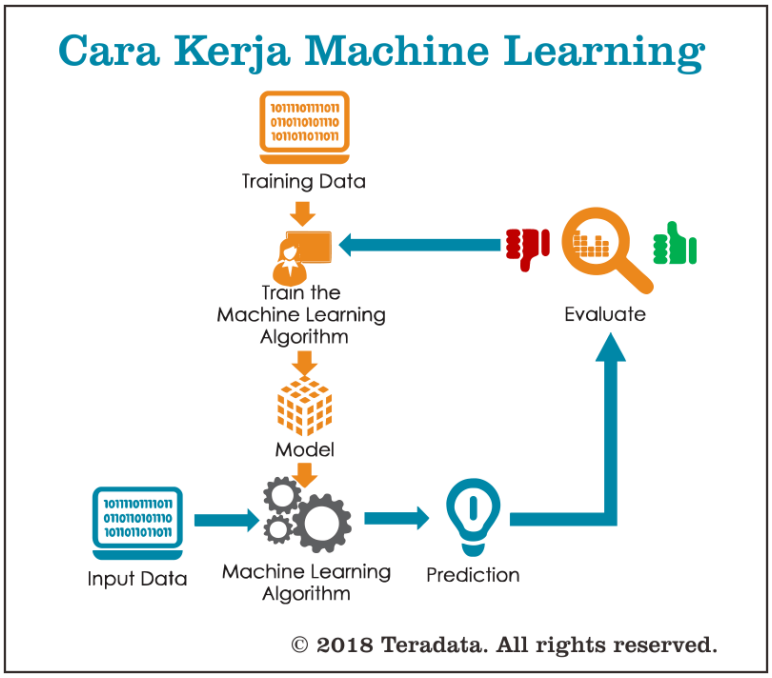
Machine learning merupakan salah satu bentuk penerapan *Artificial Intelegent* yang mempelajari tentang bagaimana membuat mesin belajar untuk menangani data dengan lebih efisien dan mengubahnya menjadi suatu aksi yang nyata tanpa campur tangan manusia (Batta, 2020). *Machine learning* umumnya digunakan untuk menyelesaikan masalah *big data* dan *dataset* yang sulit diselesaikan hanya dengan menggunakan *Excel* saja (Bi et al., 2019). Saat ini, *machine learning* banyak diterapkan pada berbagai bidang kehidupan untuk membantu manusia dalam melakukan prediksi, pengklasifikasian, dan pengelompokan. Contohnya, pada bidang kesehatan *machine learning* digunakan untuk membangun sebuah sistem yang dapat memprediksi apakah pasien tersebut terkena diabetes atau tidak. Terdapat dua tipe umum *machine learning*, yaitu *supervised learning* dan *unsupervised learning*(Kourou et al., 2015).

Supervised learning biasanya digunakan untuk membuat model prediktif menggunakan *machine learning*. Dalam *supervised learning*, sekumpulan *dataset input* dengan *target feature* yang akan diprediksi (*label*) beserta *feature* lainnya

yang memiliki hubungan dengan *target feature* akan digunakan untuk membangun model prediksi (Bi et al., 2019). Dengan kata lain, pemrosesan dalam *supervised learning* dilakukan dengan pemberian intruksi yang jelas mengenai apa yang akan diprediksi dan bagaimana cara model belajar. Terdapat beberapa algoritma yang dapat diimplementasikan pada *supervised learning*, contohnya *linear regression* dan *neural network.*

Pada *Unsupervised learning*, algoritma akan mengindentifikasi hubungan alami yang terbentuk antar *feature* dan mengelompokkannya tanpa merujuk kepada hasil lainnya(Kourou et al., 2015). Berbeda dengan *supervised learning*, dalam *unsupervised learning* tidak ada yang disebut *label*. Hal ini dikarenakan, *unsupervised learning* menggunakan pendekatan statistik yang mencoba untuk mengidentifikasi subkelompok yang sebelumnya tidak ditentukan karakteristiknya(Bi et al., 2019). Algoritma yang dapat diimplementasikan pada *unsupervised learning* adalah algoritma *clustering*, seperti *K-means*.

Secara umum, *machine learning* bekerja dengan cara belajar seperti manusia menggunakan contoh-contoh yang diberikan sebelumnya agar mendapatkan pengetahuan untuk dapat menjawab pertanyaan terkait dengan contoh yang diberikan. *Machine learning*, belajar menggunakan data, dimana proses pembelajaran tersebut dikenal dengan sebutan *train* data. Proses pelatihan ini biasanya menggunakan suatu algoritma tertentu. Berikut cara kerja *machine learning* yang dapat dilihat pada Gambar 2.1.



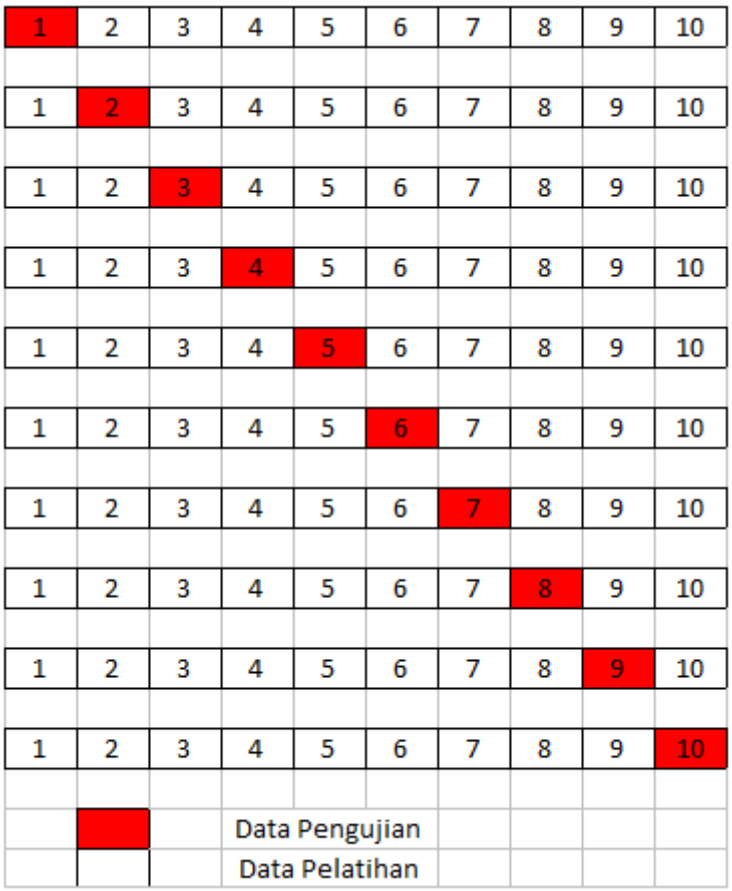
Gambar 2. 1 Cara Kerja *Machine Learning*

Machine learning akan mendapatkan *input* berupa data yang sudah dilatih dan data yang belum dilatih (data uji). Umumnya, jumlah data latih selalu lebih banyak dari data uji, misalnya dengan rasio 9:1. Data latih akan dilatih oleh sebuah algoritma *machine learning* dan menjadi sebuah model. Model ini akan di uji menggunakan data uji untuk menghitung seberapa efisien model yang dihasilkan. Hasil tersebut akan di evaluasi, apakah hasil yang didapat sesuai dengan yang diinginkan, atau tidak. Jika tidak, maka proses pelatihan akan dilakukan lagi.

2.2.2 Cross Validation

Cross validation merupakan suatu metode pengambilan ulang sampel data untuk menilai kemampuan model prediktif dan memeriksa terjadinya *overfitting* pada model klasifikasi. Pada *cross validation*, dilakukan pembagian data dengan ukuran yang hampir sama menjadi himpunan bagian K, kemudian, model akan dilatih dan diuji sebanyak K. Pengujian dilakukan terhadap salah satu bagian, sedangkan bagian lainnya digunakan untuk proses pelatihan (Nurhayati, 2014). Proses ini disebut *K-fold cross validation*.

K-fold cross validation merupakan metode yang biasanya digunakan dalam melakukan *cross validation. K-fold cross validation* yang biasanya digunakan untuk melakukan uji validasi adalah *10-fold cross validation*. Alur pengujian *10-fold cross validation* ditunjukkan oleh Gambar 2.2.



Gambar 2. 2 *K-fold Cross Validation*

Pada *10-fold cross validation*, satu himpunan bagian digunakan untuk melakukan pengujian dan sembilan himpunan bagian lainnya, digunakan untuk proses pelatihan. Contohnya, asumsikan bahwa terdapat 100 data yang dibagi kedalam *10-fold cross validation*. *Fold-1* akan digunakan untuk proses pengujian, lalu sisanya akan digunakan untuk proses pelatihan. Kemudian, masuk ke proses selanjutnya, dimana *fold-2* akan digunakan untuk proses pengujian, lalu sisanya digunakan untuk proses pelatihan. Begitu seterusnya hingga ke *fold-10* (Hulu, 2020).

2.2.3 Model Regresi

Model ini biasanya digunakan untuk mengetahui hubungan antara variabel bebas (prediktor) dengan satu atau lebih variabel tidak bebes (*label*) serta peramalan terhadap *label*(Fathurahman, 2009). Dalam penggunaannya, perlu adanya keyakinan bahwa variabel – variabel yang terdapat didalam suatu dataset memiliki keterkaitan secara teoritis atau dapat diestimasi sebelumnya. Hal ini dikarenakan, hubungan yang terjadi antar variabel membentuk hubungan sebab akibat (*causal relationship*)(Ningsih & Dukalang, 2019). Model regresi, biasanya digunakan pada dataset *time series*. Umumnya terdapat dua jenis model regresi, yaitu *simple linear regression* dan *multiple linear regression*. Namun, pada pengaplikasiannya metode regresi juga bisa digunakan pada algorima lainnya seperti *neural network.*

Untuk mendapatkan model regresi yang optimal, perlu dilakukan analisis terhadap parameter secara tepat menggunakan *hyperparameter tuning*. *Hyperparameter tuning* merupakan langkah yang krusial dalam membangun sebuah *machine learning*(Bardenet et al., 2013).

2.2.4 Multiple Linear Regression

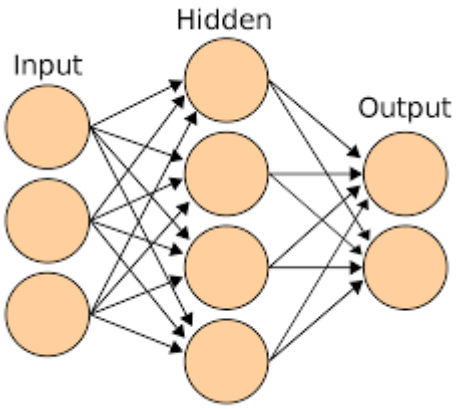
Linear regression merupakan teknik analisis data yang biasanya digunakan untuk melakukan prediksi. Fokus utama dari metode ini adalah untuk menganalisa hubungan antara satu variabel dependen dengan satu variabel independen (Uyanık & Güler, 2013). Model regresi yang memiliki lebih dari satu variabel independen adalah *Multiple Linear Regression*. *Multiple Linear regression* yang digunakan untuk menggambarkan suatu hubungan dari satu dependen variabel (y) terhadap 2 atau lebih *predictor variable* (x_1, x_2, ..., x_n) sehingga menghasilkan model yang *multivariate* (Tranmer et al., 2020). Model *multiple linear regression* direpresentasikan dengan persamaan umum sebagai berikut(LIU, 2020):

$$y = \beta 0 + \beta 1 x 1 i + \beta 2 x 2 i + \cdots + \beta p x p i \qquad (2.1)$$

Dimana y adalah dependen variabel, $\beta 0$, $\beta 1$, $\beta 2$, …. , βp adalah koefisisen regresi dan x_1, x_2, ... , x_n adalah variabel independen dari model *Multiple Linear Regression*. Koefisien regresi menggambarkan kontribusi yang tidak terkait dari masing-masing variabel independen terhadap hasil prediksi variabel dependen.

2.2.5 Neural Network

Artificial Neural Network atau biasa juga disebut *Neural Network* merupakan salah satu metode *deep learning* yang bekerja dengan cara mensimulasikan pendekatan jaringan syaraf manusia ke dalam perangkat lunak (Solechan & Shinta, 2012). Cara kerja dari *Neural Network* ini bersifat *black block*. Dimana, proses kerja di dalam kotak setelah *input* tidak bisa diketahui dengan jelas. Hal ini dikarenakan, *Neural Network* dibangun berdasarkan kalkulasi matematis yang rumit agar dapat bekerja mengikuti sistem kerja otak manusia yang kompleks (Kurniawan, 2020). Dengan kemampuan yang dibuat berdasarkan cara kerja otak manusia, metode ini dapat digunakan untuk mengatasi masalah – masalah yang kompleks.



Gambar 2. 3 Arsitektur *Neural Network*

Dalam *Neural Network*, terdapat *neuron-neuron* yang saling terhubung dan berkumpul dalam lapisan-lapisan yang disebut *neuron layer*. Terdapat 3 lapisan penyusun *Neural Network*, yaitu *input layer*, *hidden layer* dan *output layer* (Bell, 2014).

1. Lapisan *input* (*Input Layer*)

   Lapisan ini bertugas sebagai penerima pola *input*-an dari luar yang menggambarkan suatu permasalahan tertentu.

2. Lapisan tersembunyi (*Hidden Layer*)

   Pada lapisan ini, unit yang bekerja didalamnya tidak dapat dilihat atau diamati secara langsung, ini lah mengapa *neural network* disebut memiliki sifat *blackbox*.

3. Lapisan *output* (*Output Layer*)

   Lapisan *output* memiliki unit-unit yang disebut dengan unit *output* yang merupakan bentuk solusi *Neural Network* terhadap permasalahan tertentu berdasarkan *input* yang diberikan pada lapisan *input*.

Umumnya, *neural network* digunakan pada *Deep Learning*, karena metode ini memiliki *perceptron* sebagai asumsi dari *neuron* yang ada pada otak manusia, sehingga memiliki kemampuan untuk belajar dari pengalaman-pengalaman masalalu. Selain digunakan pada *Deep Learning*, *Neural network* juga dapat digunakan untuk berbagai keperluan, seperti melakukan klasifikasi, prediksi numerik (regresi) dan juga pengenalan pola yang tidak terlalu kompleks (Kurniawan, 2020).

Pada penelitian ini, metode *Neural Network* yang digunakan adalah *backpropagation*. *Backpropagation* merupakan suatu metode yang digunakan untuk melakukan *training* terhadap jaringan. Pada umumnya, *backpropagation* terdiri dari 3 tahap, yaitu *feedforward*, *backpropagation*, dan penyesuaian bobot. Berikut langkah-langkah dalam algoritma *backpropagation* (Nurmila et al., 2010).

1. Inisialisasi bobot (pengambilan bobot awal dengan nilai *random* yang kecil),

2. Menetapkan maksimum *Epoch*, target *error*, learning rate dan momentum

3. Lakukan langkah ke-4 sampai ke-8 selama kondisi *epoch* < maksimum *epoch* dan *error* model > Target *Error*.

4. Lakukan langkah ke-6 sampai ke-10 selama proses *training*

5. Untuk pengujian, lakukan langkah ke-6 dan lagkah ke-7

***Feedforward* (Perambatan maju)**

6. Melakukan penjumlahan bobot pada *hidden layer*

$$Z\_in(j) = b_{ij} + \sum_{i=1}^{n} X_i . V_{ij} \quad (2.2)$$

Mengaplikasikan fungsi aktivasi signoid untuk menghitung signal *output*

$$Z_j = \frac{1}{1+e^{-Z\_in(j)}} \quad (2.3)$$

7. Menghitung nilai *output layer*

$$Y\_in(k) = b_{jk} + \sum_{i=1}^{n} Z_j . W_{jk} \quad (2.4)$$

Menghitung sinyal ouyput dengan fungsi aktvasi signoid.

$$Y_k = \frac{1}{1+e^{-Y\_in(k)}} \quad (2.5)$$

Backpropagation

8. Hitung nilai *error* berdasarkan nilai *error* dan nilai target

$$\delta k = (tk - Yk) \quad (2.6)$$

Hitung nilai koreksi setiap bobot W_{jk}

$$\Delta W_{jk}(t) = \alpha \delta_k Z_j + \Delta W_{jk}(t-1)\mu \quad (2.7)$$

Hitung nilai koreksi bias b_{jk}

$$\Delta b_{jk}(t) = \square\square_k + \Delta b_{jk}(t-1)\mu \quad (2.8)$$

9. Menjumlahkan *delta unit* pada setiap *neuron hidden layer*

$$\delta in\ j = \sum_{k=1}^{m} \delta_k W_{jk} \quad (2.9)$$

Hitung nilai *error*

$$\delta j = \delta in\ j(Zj)(1 - Zj) \quad (2.\ 10)$$

Koreksi bobot :

$$\Delta Vij(t) = \alpha \delta j Xi + \Delta Vij(t-1)\mu \quad (2.\ 11)$$

Koreksi bias :

$$\Delta V0j(t) = \alpha \delta j + \Delta V0j(t-1)\mu \quad (2.\ 12)$$

Update bobot dan bias

10. Melakukan update pada nilai bobot dan nilai bias

- *Update* bobot *hidden layer*

$$V_{ij}(baru) = V_{ij}(lama) + \Delta V_{ij} \quad (2.\ 13)$$

- *Update* bias *hidden layer*

$$b_{ij}\ (baru) = b_{ij}\ (lama) + \Delta b_{ij} \quad (2.\ 14)$$

- *Update* bobot *output layer*

$$W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk} \quad (2.\ 15)$$

- *Update* bias *output layer*

$$b_{jk}\ (baru) = b_{jk}\ (lama) + \Delta b_{jk} \quad (2.\ 16)$$

11. Tes kondisi berhenti.

Dimana, V = Bobot *input*, b_{0j} = Bias *hidden*, W = Bobot *neuron* pada *hidden layer*, b_{0k} = Bias *output*, X = Data, Y_k = *Output neuron* k, Z_j = *Neuron hidden* j, i = Data ke- , j = Jumlah *neuron* pada *hidden layer*, k = Jumlah target, dan μ = Momentum.

2.2.6 Uji T

Uji T merupakan salah satu metode yang digunakan untuk menguji hipotesis dari penelitian analisis regresi. Metode ini sering digunakan untuk

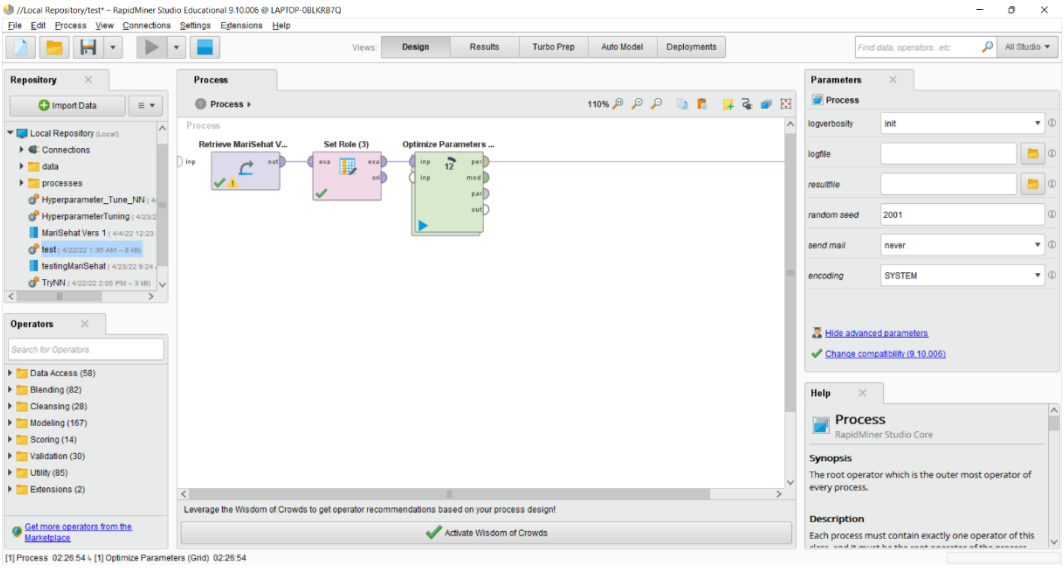
menguji kebenaran dari sebuah hipotesis dalam menyatakan apakah terdapat berbedaan yang yang signifikan antara dua buah sampel (Aditya et al., 2021). Adanya kebenaran atau kepalsuan pada suatu hipotesis dilihat dari suatu nilai yang disebut T-statistics (Ghozali, 2016). Pengambilan keputusan biasanya dilakukan dengan dasar pengujian hasil regresi yang dilakukan dengan tingkat kepercayaan sebesar 95% ataupun dengan taraf signifikan atau nilai alpha sebesar 5% (0,05). Berikut kriteria dari uji T (Ghozali, 2016):

1. Jika nilai signifikansi dari uji t > 0.05, maka H0 diterima dan Ha ditolak, hal ini berarti bahwa tidak ada perbedaan yang berarti antara model yang dibandingkan.

2. Jika nilai signifikansi dari uji t < 0.05, maka H0 ditolak dan Ha diterima, yang berarti bahwa terjadi perbedaan yang berarti antara model yang di bandingkan.

2.2.7 RapidMiner

RapidMiner merupakan perangkat lunak yang digunakan sebagai solusi untuk permasalahan analisa terhadap *data mining*, *text mining* dan analisis prediksi. *Tools* ini biasa digunakan untuk membangun model prediktif pada *machine learning*. Dalam membangun model *machine learning* menggunakan *RapidMiner* dilakukan dengan cara membuat *workflow* yang mencakup proses – proses yang diperlukan untuk membangun model *machine learning*. Proses tersebut meliputi *data preprocessing*, *training*, *testing* hingga *deployment*. Untuk membuat sebuah *workflow,* pengguna hanya perlu melakukan *drag and drop* operator - operator ke canvas yang sudah disediakan.



Gambar 2. 4 *RapidMiner*