Perbandingan Algoritma *Machine Learning* dalam Menilai Sebuah Lokasi Toko Ritel

http://dx.doi.org/10.28932/jutisi.v7i1.3182
Riwayat Artikel
Received: 7 Desember 2020 | *Final Revision*: 20 Februari 2021 | *Accepted*: 1 Maret 2021

Kristiawan[#1], Andreas Widjaja[✉*2]

Magister Ilmu Komputer, Universitas Kristen
Jl. Surya Sumantri No. 65, Sukawarna, Kec. Sukajadi, Kota Bandung, Jawa Barat 40164
[1]kristiawan.indi@gmail.com
[2]andreas.widjaja@it.maranatha.edu

Abstract — **The application of machine learning technology in various industrial fields is currently developing rapidly, including in the retail industry. This study aims to find the most accurate algorithmic model so that it can be used to help retailers choose a store location more precisely. By using several methods such as Pearson Correlation, Chi-Square Features, Recursive Feature Elimination and Tree-based to select features (predictive variables). These features are then used to train and build models using 6 different classification algorithms such as Logistic Regression, K Nearest Neighbour (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM) and Neural Network to classify whether a location is recommended or not as a new store location.**

Keywords— **Application of Machine Learning, Pearson Correlation, Random Forest, Neural Network, Logistic Regression.**

I. PENDAHULUAN

Industri retail adalah industry yang sangat menarik bagi banyak pembisnis. Menurut majalah Forbes dari daftar 10 keluarga terkaya di Amerika Serikat, yang menduduki peringkat nomor satu adalah keluarga Waltons yang merupakan pengusaha jaringan retail *Walmart* [1]. Selain mendatangkan keuntungan yang besar industri retail adalah industry yang paling tahan banting, di saat perusahaan industri lain harus mengalami kerugian dan harus menutup bisnisnya, perusahaan retail tetap bertahan, contoh nyata saat pandemi *covid 19*, di saat semua perusahaan harus menutup kantor dan tempat usahanya, toko-toko retail tetap buka dan menjalankan bisnisnya sekalipun dengan berbagai macam aturan dan batasan.

Maka dari itu tidak heran banyak pengusaha-pengusaha besar yang ingin berbisnis di bisnis retail, tetapi industry retail bukan bisnis yang mudah, khususnya untuk pemain-pemain baru. Industry retail memiliki *entry barrier* yang tinggi karena harus menghadapi peritel-peritel lama yang sudah berpengalaman dan juga karena sifat bisnis retail yang padat karya dan memiliki margin yang rendah membuat banyak pengusaha harus benar-benar efektif dan efisien dan tidak boleh salah dalam membuat keputusan.

Salah satu faktor utama yang mempengaruhi keberhasilan bisnis retail adalah penentuan lokasi, salah memilih lokasi bisa berakibat fatal, selain membuat toko tidak berkembang karena marketnya tidak tumbuh juga bisa menyebabkan toko tersebut tutup karena merugi.

Banyak cara dalam memilih sebuah lokasi dari yang menggunakan intuisi, menggunakan metode statistik sederhana sampai menggunakan sistem pendukung keputusan seperti *naïve bayes* [2] [3]. Penelitian ini bermaksud untuk memberikan cara lain yang lebih akurat dalam menentukan lokasi toko ritel dengan menggunkan algoritma *machine learning*.

Rumusan masalah dalam penelitian ini adalah bagaimana mendapatkan algoritma *machine learning* yang menghasilkan model terbaik, yang dapat memberikan rekomendasi apakah sebuah lokasi tepat untuk dijadikan toko baru atau tidak?

Penelitian ini akan dibatasi pada algoritma *machine learning* dengan pendekatan supervised learning. Pendekatan supervised learning adalah salah satu cabang besar dari machine learning yang membuat sebuah fungsi atau model dari data pelatihan yang sudah diberi label. Data-data pelatihan tersebut bentuknya berupa pasangan *input, output (label)*. Jenis algoritma yang digunakan adalah *Classification Algorithm* dengan menggunakan metode *binary classification* untuk membuat sebuah fungsi atau model yang dapat melakukan klasifikasi apakah sebuat lokasi direkomendasi atau tidak untuk dijadikan toko baru.

Pembuatan model menggunakan enam macam algoritma klasifikasi yang berbeda, yang nantinya setiap model yang dihasilkan oleh masing-masing algoritma tersebut akan dibandingkan. Adapun algoritma klasifikasi yang akan dibandingkan adalah *Logistic Regression, Lasso, Decision Tree, Random Forest, Support Vector Machine* dan *Neural Network.*

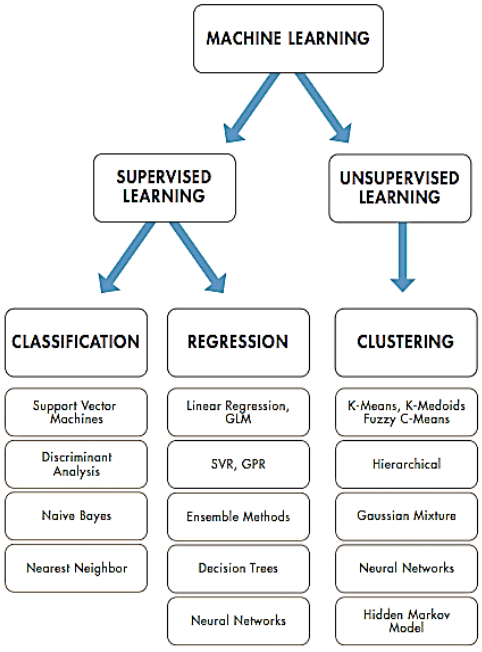
p-ISSN : 2443-2210
e-ISSN : 2443-2229

II. KAJIAN LITERATUR

Dalam penelitian ini, menggunakan literatur-literatur yang terkait seputar sub bagian machine learning yaitu *Supervised, Unsupervised Learning,* Algoritma klasifikasi seperti *Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM) dan Neural Network*. Selain itu ada beberapa metode *Features Selection* seperti *Lasso, Pearson Correlation* dan juga ada tentang *Cross Validation* untuk mencegah *Overfitting* serta terakhir metode untuk melakukan evaluasi model menggunakan *Confusion Matrix* dan *kurva ROC* dan AUC.

A. Supervised dan Unsupervised Learning

Supervised Learning adalah Teknik melatih mesin menggunakan data yang diberi label [4]. Maksud dari pembelajaran yang diawasi adalah data label atau target ikut berperan sebagai 'supervisor' atau 'guru' yang mengawasi proses pembelajaran dalam mencapai tingkat akurasi atau presisi tertentu. Artinya beberapa data sudah ditandai dengan jawaban yang benar dan mesin belajar untuk membuat sebuah model yang akan menghasilkan hasil yang benar atau mendekati bila diberikan data baru. Misalkan kita ingin memprediksi harga sebuah rumah maka dengan memasukkan variabel input luas rumah, jumlah kamar dan variabel harga sebagai variable ouput yang merupakan label maka kita bisa membuat sebuah model yang bisa memprediksi harga sebuah rumah. Contoh algoritma yang termasuk *Supervised Learning* adalah Regresi dan klasifikasi lihat Gambar 1.



Gambar 1. *Supervised* and *Unsupervised learning* [4]

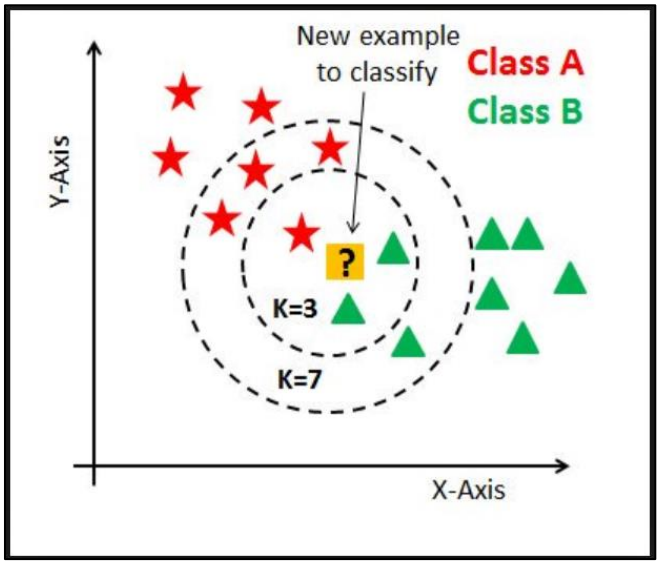
Unsupervised machine learning adalah teknik pembelajaran mesin, di mana machine tidak perlu diawasi diberi jawaban yang benar. Sebaliknya, mesin diijinkan bekerja sendiri untuk menemukan informasi. Hal Ini terutama berkaitan dengan data tidak berlabel. Misalkan kita ingin mengelompokkan rumah berdasarkan jumlah kamar dan luas rumah, maka dengan memberikan data rumah dengan variabel jumlah kamar dan luas rumah mesin akan membuat model yang bisa mengelompokan rumah menjadi beberapa kelompok berdasarkan kedua variabel tersebut, sehingga jika nanti ada data baru maka mesin bisa menebak bahwa rumah tersebut akan termasuk kelompok yang mana. Contoh algoritma yang termasuk *Unsupervised Learning* adalah: *Clustering, Association dan Dimensionality Reduction* [2].

B. Logistic Regression

Logistic Regression atau Regresi logistik adalah kasus khusus dari regresi linier di mana variabel target bersifat kategorikal. Regresi logistic dipakai untuk memprediksi kelas biner 0 atau 1 [4]. Variabel hasil atau target bersifat dikotomis. Dikotomis berarti hanya ada dua kemungkinan, ya dan tidak, dengan hanya dua kemungkinan, regresi logistik, dapat digunakan untuk mendeteksi kanker, email spam atau bukan spam [4].

C. K-Nearest Neighbor (KNN)

Secara Sederhana *K-nearest neighbor* atau KNN adalah algoritma yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (*train data sets*), yang diambil dari *k* tetangga terdekatnya (*nearest neighbors*). Dengan *k* merupakan banyaknya tetangga terdekat. Hasil klasifikasinya berdasarkan mayoritas kedekatan jarak dari tetangga terdekatnya [5]



Gambar 2. *K-Nearest Neighbor* [5]

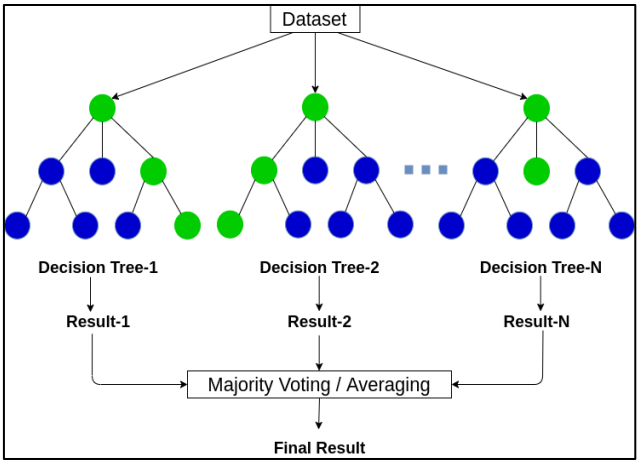
D. Decision Tree

Decision Tress atau Pohon Keputusan adalah jenis algoritma untuk pembelajaran mesin pemodelan prediktif. Representasi untuk model *Decision Trees* adalah pohon biner.

Setiap node mewakili satu variabel input (*x*) dan cabangnya merepresentasikan nilai dari variabel input tersebut, sedangkan simpulnya merepresentasikan variable output (*y*) atau kelas. Node teratas dari decision tree ini disebut root. Dinamakan pohon keputusan karena aturan yang terbentuk mirip dengan bentuk pohon [6].

E. Random Forest

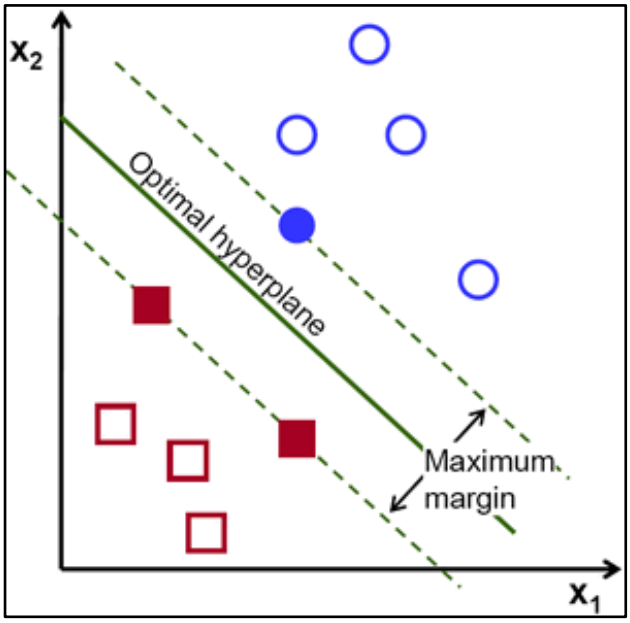
Random Forest adalah salah satu algoritma pembelajaran mesin yang paling populer dan termasuk salah satu yang paling kuat. *Random Forests* adalah perbaikan dari pohon keputusan. Disebut *Random Forest* atau hutan "acak"? Karena hutan pohon keputusan yang dibuat secara acak. Setiap node di pohon keputusan bekerja pada *subset fitur* acak (bukan *greedy* algoritma) untuk menghitung *output. Random Forest* kemudian menggabungkan keluaran dari pohon keputusan individu untuk menghasilkan keluaran akhir [6].



Gambar 3. *Random Forest* [6]

F. Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu metode dalam *supervised learning* yang biasanya digunakan untuk klasifikasi (seperti *Support Vector Classification*) dan regresi (*Support Vector Regression*) baik untuk data *linear* maupun *nonlinear*. SVM melakukan klasifikasi dengan cara memilih batas keputusan yang memaksimalkan (*maximum Margin Classifier*) jarak dari titik data terdekat dari semua kelas. Batas keputusan yang dibuat oleh SVM disebut pengklasifikasi margin maksimum(*maximal margin classifier*) atau bidang hiper (*Hyperplane*) margin maksimum [7].

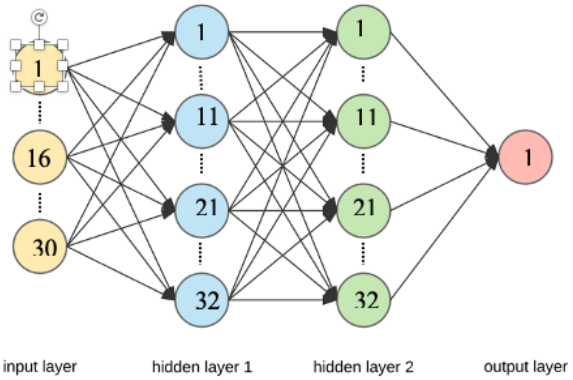


Gambar 4. *SVM Maximum Margin Classifier* [7]

G. Neural Network

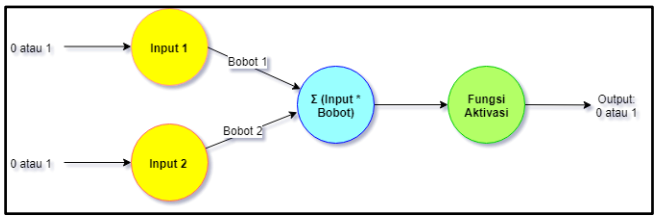
Neural Network atau Jaringan Saraf adalah pembentuk atau tulang punggung dari *Algoritma Deep Learning. Deep Learning* adalah bagian sub bagian dari *machine learning* dan machine learning adalah sub bagian dari bidang *Artificial Intelligent. Neural network* (jaringan saraf) bekerja dengan cara meniru kerja *neuron* (saraf) dalam otak manusia.

Secara sederhana neural network dibentuk dari dari 3 macam layer, input layer berfungsi menerima masukkan, output layer berfungsi memprediksi hasil akhir keluaran dan di tengah-tengahnya terdapat *hidden layer* yang berfungsi melakukan sebagian besar komputasi yang dibutuhkan oleh jaringan [8].



Gambar 5. *Neural Network Architecture*

p-ISSN : 2443-2210
e-ISSN : 2443-2229



Gambar 6. *Neural Network sederhana*

H. Features Selection

Data yang digunakan untuk melatih model pembelajaran mesin memiliki pengaruh besar pada performa yang dapat dicapai. *features* yang tidak relevan atau sebagian relevan dapat berdampak negatif pada performa model.
Ada tiga manfaat melakukan pemilihan *features* sebelum membuat model:

- Mengurangi *Overfitting* (kondisi di mana bekerja dengan sangat baik *te*rhadap data training tetapi berkinerja buruk terhadap data real): lebih sedikit data yang berlebihan berarti lebih sedikit peluang untuk membuat keputusan berdasarkan bias.
- Meningkatkan Akurasi: lebih sedikit data yang tidak relevan berarti akurasi pemodelan meningkat.
- Mengurangi waktu pelatihan: Lebih sedikit data berarti algoritma berlatih lebih cepat.

Beberapa cara atau metode untuk mengaplikasikan *features selection*, di antaranya adalah:

- ***Filter Method*** melakukan *filtering* pada dataset dan mengambil hanya bagian data yang mengandung semua fitur yang relevan (Contoh: *Matrix Correlation* dengan menggunakan metode Pearson) [9].

- **Wrapper Method** mengikuti cara seperti metode filter tetapi memakai *machine learning* model sebagai kriteria evaluasinya (contoh. *Forward/Backward /Bidirectional/ Recursive Feature Elimination*) [9]. Memasukkan beberapa fitur ke model *machine learning*, mengevaluasi kinerjanya dan kemudian memutuskan apakah menambah atau menghapus fitur untuk meningkatkan akurasi. Metode ini bisa lebih akurat daripada pemfilteran tetapi lebih banyak melakukan proses komputasi.

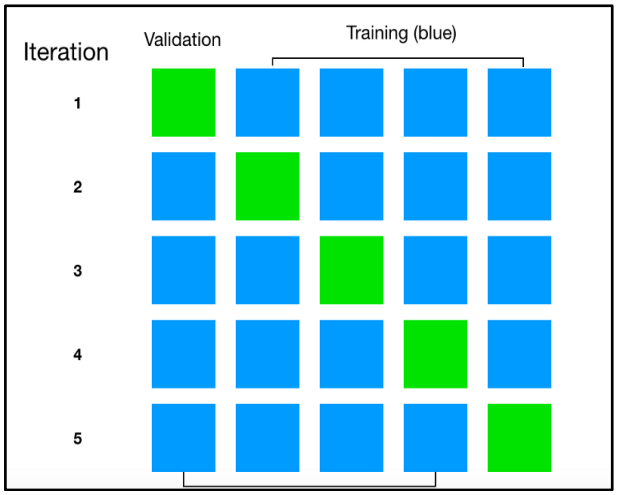
- ***Embedded Method*** metodenya seperti metode wrapper, Embedded *Method* juga menggunakan *machine learning* model. Perbedaannya pada *wrapper* fitur diintegrasikan ke model sedangkan pada *embedded* methode fiturnya di embedded, dengan melakukan proses iterasi pelatihan model *machine learning* dan kemudian membuat peringkat tingkat kepentingan dari setiap fitur berdasarkan berapa banyak masing-masing fitur berkontribusi pada pelatihan model *machine learning* (mis. regresi LASSO) [9].

I. Cross Validation

Cross Validation digunakan untuk mencegah overfitting, Ada berbagai jenis Teknik *Cross Validation* (Validasi Silang) tetapi konsep keseluruhannya tetap sama, yaitu:

- Mempartisi data menjadi beberapa subset
- Menyimpan satu set pada satu waktu dan melatih model pada set yang tersisa
- Uji model pada data yang tadi disimpan

Contoh metode *Cross Validation yang umumnya digunakan adalah Fold Cross Validation* [10].



Gambar 7. 5-*Fold Cross Validation* [10]

III. METODOLOGI PENELITIAN

A. Hipotesis-Hipotesis

Penelitian ini didasarkan pada hipotesis-hipotesis berikut ini:

- Tidak semua *features* penting atau memiliki peranan dalam pembuatan model, ada *features* yang perannya tidak penting sehingga boleh dihilangkan.
- Setiap algoritma memiliki karakteristik karakteristik yang berbeda-beda tergantung dari jenis atau type dari datanya sehingga setiap algoritma akan memiliki tingkat akurasi yang berbeda-beda juga.

B. Tahapan Penelitian

Proses penelitian tesis ini dapat dilihat pada Langkah-langkah berikut ini:

1. Preprocessing data
2. Seleksi fitur
3. Membuat Model
4. Evaluasi
5. Membandingkan model

Langkah pertama dengan preprocesssing data:

a) **Proses Encoding**

Encoding adalah salah satu tahap preprocessing data adalah proses merubah tipe data kategori ke numerik sebelum diproses dengan algoritma *machine learning*. Dalam mengerjakan projek *data science* ataupun *machine learning*, akan sangat mungkin menemukan satu atau beberapa fitur yang bertipe kategori, misalnya salah satu kategori dalam tesis ini adalah kebiasaan_belanja 'Harian, 'Mingguan', 'Bulanan'. Algoritma machne learning klasifikasi tidak dapat memproses data bertipe kategori sehingga data harus diubah menjadi berbentuk bilangan. Proses ini perubahan tersebut disebut dengan *encoding*. Proses *encoding yang dilakukan* menggunakan metode *One-Hot encoding*. Metode ini merepresentasikan data bertipe kategori sebagai vektor biner yang bernilai integer, 0 dan 1, di mana semua elemen akan bernilai 0 kecuali elemen yang memiliki nilai kategori nilainya 1. Fungsi yang digunakan adalah fungsi *pd.get_dummies* yang berfungsi mengubah data kategori menjadi *vector biner*. Salah satu contoh data yang dikonversi adalah: data kebiasaan belanja seperti pada tabel 1:

TABEL I
KEBIASAAN BELANJA

kebiasaan_belanja
Mingguan
Harian
Mingguan
Harian
Harian
Mingguan
Harian
Harian
Harian
Harian
Bulanan

Dengan menggunakan fungsi *pd.get_dummies* dari *library panda*, Tabel 1 diencode.

TABEL II
TABEL KEBIASAAN BELANJA YANG SUDAH DIENCODE

	kebiasaan_belanja	Bulanan	Harian	Mingguan
0	Mingguan	0	0	1
1	Harian	0	1	0
2	Mingguan	0	0	1
3	Harian	0	1	0
4	Harian	0	1	0
...	...	...	...	...
252	Harian	0	1	0
253	Mingguan	0	0	1
254	Mingguan	0	0	1
255	Harian	0	1	0
256	Harian	0	1	0

257 rows × 4 columns

Berikut adalah tampilan potongan dataset yang sudah melalui proses encoding

TABEL III
POTONGAN DATASET YANG SUDAH DIENCODE

	Bis	Angkot	Ojek	roda2	roda4	pejalan_kaki	BCA	BNI	Ibadah	pasar	...
0	0	1	0	3012	1452	65	1	0	0	0	...
1	0	0	1	882	108	135	1	0	0	0	...
2	0	0	0	20	10	15	1	0	0	0	...
3	1	0	0	20	120	30	1	1	1	0	...
4	0	1	1	1908	419	64	0	0	0	1	...
...	...	...	...	...	...	...	...	...	...	...	...
249	0	1	1	1322	495	75	0	0	0	1	...
250	0	1	1	3600	720	300	1	0	1	0	...
251	1	1	1	2700	1300	360	1	0	0	1	...
252	0	1	0	624	600	24	0	0	1	0	...
253	0	0	0	100	200	5	1	0	0	0	...

254 rows × 31 columns

b) Pengecekan Missing Data

Proses ini untuk memastikan tidak ada data yang invalid atau kosong. Menset *features* (variable praduga) dan label (variabel target). Dengan menggunakan fungsi *is. null ()*, program akan melakukan pengecekan apakah ada kolom-kolom yang datanya tidak lengkap.
Hasil pengecekannya seperti di bawah ini

p-ISSN : 2443-2210
e-ISSN : 2443-2229

Tabel 1 Hasil pengecekan dengan fungsi is.null()

```
jml_lantai          0
Bis                 0
Angkot              0
Ojek                0
lainnya             0
..
lebar_besar6m       0
lebar_kecil3m       0
lebar_sedang3-6m    0
jalur_cepat         0
jalur_lambat        0
Length: 63, dtype: int64.
```

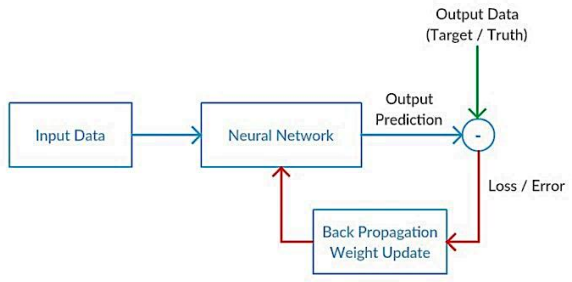
Terlihat semua data tidak ada yang kosong atau invalid.

c) *Feature Scaling*

Beberapa algoritma machine learning memerlukan tambahan *preprocessing data* sebelum melakukan proses training model. Algoritma MACHINE LEARNING seperti KNN dan SVM adalah algoritma berbasis jarak yang sangat dipengaruhi oleh besar dan nilai dari fitur-fiturnya. Hal ini terjadi karena di balik layar algoritma-algoritma tersebut menggunakan jarak antar titik data untuk menentukan kesamaannya. Oleh karena itu sebelum mentraining data, data harus di *features scaling* dulu agar semua fiturnya memiliki berkontribusi sama pada hasil dan tidak menjadi bias terhadap satu fitur saja.

d) *Standarization*

Algoritma lain seperti *neural network*, membutuhkan perlakuan yang berbeda, *neural network* membutuhkan proses standarisasi data sebelum melakukan training model. Alasan pertama untuk menghilangkan pengaruh satu faktor di atas faktor lainnya (yaitu untuk memberikan fitur peluang yang sama), alasan kedua metode gradient descent untuk *mekanisime backpropagasi*, *normalization* akan mempercepat pembelajaran dan mengarah ke konvergensi yang lebih cepat dibandingkan dengan data yang tidak distandarisasi.

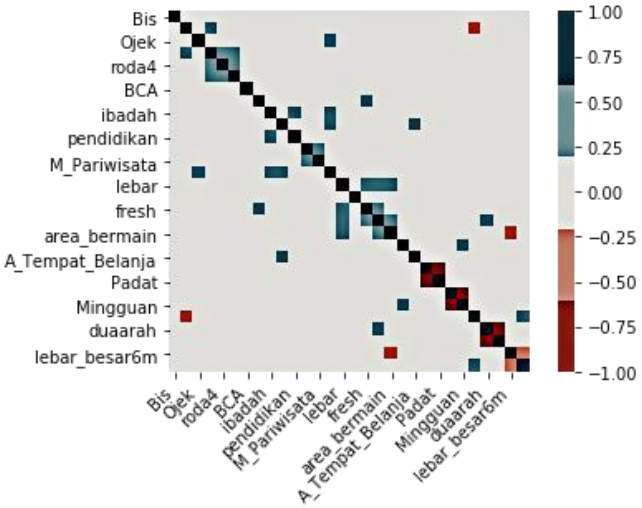


Gambar 9 Back Propagation

$$X_{Stand} = \frac{X - Mean(x)}{standard\ deviation(x)}$$

Gambar 8. Standardization equation

Langkah kedua setelah *preprocessing data* adalah melakukan seleksi fitur. Seleksi fitur dilakukan dengan menggunakan berbagai macam metode seperti metode filtering dengan *Pearson Correlation*, metode *Recursive Feature Elimination* dengan *random forest* dan, *metode wrapper* dengan *Lasso* [9]. Berikut adalah salah satu contoh seleksi fitur dengan menggunakan *Pearson Correlation* dan menampilkannya dengan *library Seaborn.*



Gambar 9. *Matrix Correlation* (*Pearson Correlation*)

Dari Gambar Matrix Correlation ada beberapa pasangan *features* yang memiliki nilai korelasi yang tinggi, yaitu :

```
Kurang_Padat    Padat           -1.000000
Padat           Kurang_Padat    -1.000000
duaarah         satuarah        -0.802657
satuarah        duaarah         -0.802657
```

Features yang memiliki pasangan tinggi bisa didrop, dalam kasus ini features kurang padat dan satu arah bisa didrop karena sudah bisa diwakilkan oleh *feature* padat dan dua arah.

Langkah ketiga membuat model. Setelah data set siap saatnya untuk membuat model dengan menggunakan *k-fold 10 cross validation*. Pembuatan model dilakukan dengan menggunakan 6 macam algoritma seperti: *Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM) dan Neural Network.*

Langkah keempat setelah model dibuat maka dilakukan proses evaluasi untuk setiap model. Evaluasi dilakukan dengan menggunakan 2 alat ukur:

1. Confusion Matrix
2. Kurva ROC dan AUC

confusion matrix akan menghitung nilai *accuracy, sensitivity, precession, recall* dan *F1 score*. Kurva ROC dan AUC akan mevisualisasikannya.

Langkah kelima membandingkan hasil evaluasi dari satu model dengan model lainnya dan memilih algoritma yang menghasilkan model yang terbaik.

IV. Evaluasi Model Hasil dan Pembahasan Model

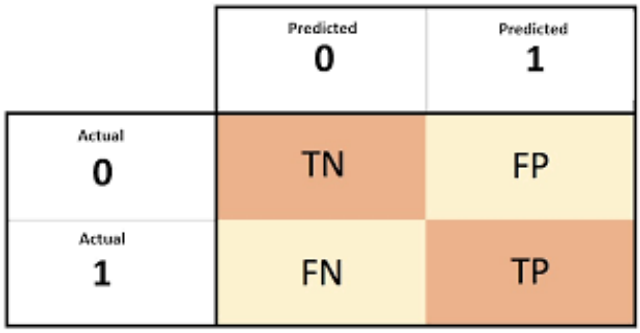
Dalam melakukan evaluasi model digunakan dua macam metode yaitu confusion matrix dan kurva ROC dan AUC.

A. Confusion Matrix

Confusion Matrix adalah tabel yang sering digunakan untuk mendeskripsikan performa model *machine learning. Confusion Matrix* merepresentasikan prediksi dan kondisi sebenarnya(aktual) dari data yang dihasilkan oleh algoritma *machine learning* khususnya model klasifikasi.[3]

Ada 4 kondisi dalam *Confusion Matrix:*

- *True Positive (TP)* prediksi *positive, real positive*
- *True Negative (TN)* prediksi *negative, real negative*
- *False Positive (FP)* prediksi *positive, real negative*
- *False Negative (FN)* prediksi *negative, real positive*



Gambar 9. *Confusion Matrix*

Dari hasil *confusion matrix* didapatkan *Tabel performance* sebagai berikut (Tabel 4). Berdasarkan *Confusion Matrix,* kita bisa menentukan *Accuracy*, *Precision*, *Recall*, *Specificity* dan F1 *Score* [11].

TABEL IV
PERFORMANCE CONFUSION MATRIX DARI SETIAP MODEL

Algoritma	Accuracy	Sensitivity (Recall)	Specificity	Precision	F_1 Score
Logistic Regression	90%	97%	21%	92%	95%
K Nearest Neighbor (KNN)	90%	99%	0.0	90%	95%
Decision Tree	84%	91%	17%	91%	91%
Random Forest	91%	99%	79%	98%	98%
Support Vector Machine (SVM)	93%	100%	25%	93%	96%
*Neural Network	97%	99%	79%	98%	98%

Berdasarkan hasil dari Tabel performance ada 5 pilihan:

- **Pilih algoritma yang memiliki *accuracy* tinggi** jika yang dipentingkan adalah seberapa akurat sistem mengklasifikasi dengan benar, *accuracy* merupakan rasio prediksi benar (positif dan negatif) dengan keseluruhan data. Dari Tabel terlihat algoritma yang memiliki akurasi tertinggi adalah *neural network* disusul oleh *svm* dan *random forest.*
- **Pilih algoritma yang memiliki *recall* tinggi jika**, pengambil keputusan lebih memilih *False Positive* terjadi daripada *False Negative* Dalam penelitian ini, lebih baik algoritma salah memprediksi lokasi bagus tetapi sebenarnya buruk, daripada algoritma salah memprediksi bahwa lokasinya buruk padahal sebenarnya bagus.
- **Pilih algoritma yang memiliki *precission* tinggi jika,** pengambil keputusan lebih menginginkan terjadinya *True Positive* dan sangat tidak menginginkan terjadinya *False Positive* Pada penelitian ini lebih baik algoritma salah memprediksi lokasi buruk padahal sebenarnya bagus daripada salah memprediksi lokasi baik padahal sebenarnya buruk.
- **Pilih algoritma yang memiliki *specificity* tinggi jika,** pengambil keputusan tidak menginginkan terjadinya false positif. Sistem sangat mengharapkan tidak terjadi salah mendeteksi lokasi yang sebenarnya buruk tapi diprediksi bagus.
- **Pilih algoritma dengan *F1 Score* tertinggi jika,** pengambil keputusan lebih mementingkan recall dan precision yang tinggi. Artinya yang dipilih adalah algoritma yang memberikan nilai *False Positive* kecil dan *False Negative* kecil juga.

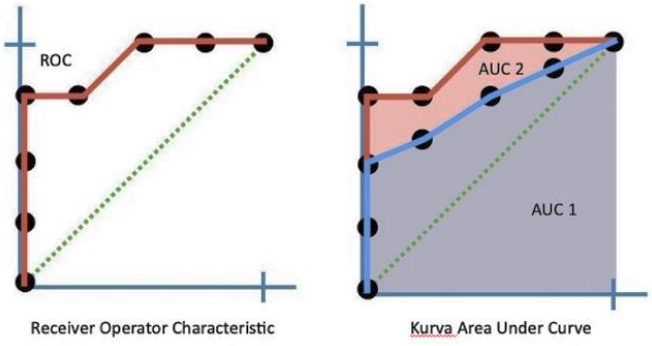
p-ISSN : 2443-2210
e-ISSN : 2443-2229

Dari kelima pengukuran evaluasion matrix di atas, bagi peritel algoritma yang dipilih adalah algoritma yang memiliki *F1 Score* tertinggi, yaitu algoritma yang memiliki *false positive* paling sedikit dan false negative sedikit. Sehingga peritel memiliki resiko mengalami kerugian yang kecil (*false positive* rendah) dan memiliki resiko kehilangan keuntungan yang kecil (*false negative* rendah). Dari Tabel di atas sistem yang memiliki *F1 Score* yang tinggi 98%, adalah algoritma *neural network*, *random forest* atau *svm.* Selain algoritma yang memiliki *F1 Score* tertinggi, bisa juga dipilih algoritma yang memiliki *false positive* paling sedikit yaitu algoritma yang memiliki nilai precision tinggi seperti neural *network*, *svm* dan *random forest*.

B. Kurva *ROC* dan *AUC*

Pada Confusion matrix, performa informasi hanya disajikan dalam bentuk angka. Untuk menampilkan informasi kinerja algoritma klasifikasi dalam bentuk grafik dapat digunakan *Receiver Operating Characteristic (ROC)* atau *Precision-Recall Curve.*

Kurva *ROC* dibuat berdasarkan nilai yang telah didapatkan dari perhitungan dengan confusion matrix, yaitu antara *False Positive Rate* dengan *True Positive Rate*. Untuk membandingkan nilai kinerja masing-masing algoritma dapat dilakukan dengan membandingkan luas di bawah kurva atau *AUC (Area Under Curve).*

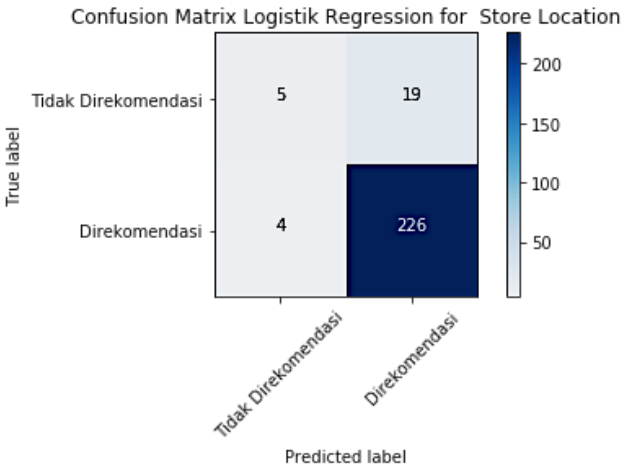


Gambar 10. Kurva ROC dan AUC [11]

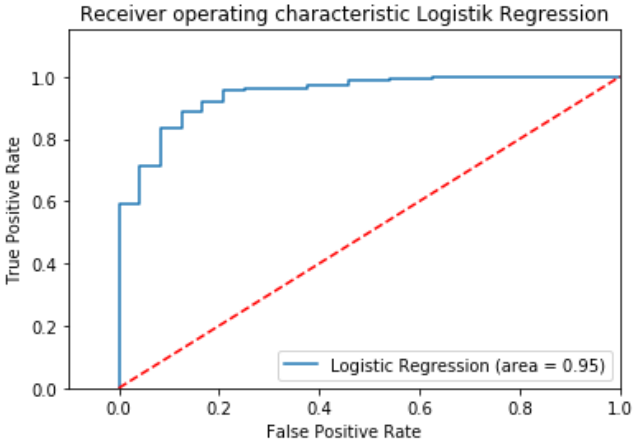
Kelebihan dari penggunaan kurva *ROC* untuk mengevaluasi klasifikasi adalah *ROC* bukan sekedar untuk mencari rata-rata akurasi tetapi *ROC* memvisualisasikan semua *threshold* klasifikasi yang mungkin, sedangkan *error rate classifier* hanya mewakili tingkat kesalahan, akurasi untuk satu *threshold* saja [11].

Berikut adalah hasil dari pengujian dari 6 Algorima klasifikasi.

A. Hasil *Confusion Matrix* dan Kurva *ROC* dan *AUC Logistic Regression*.



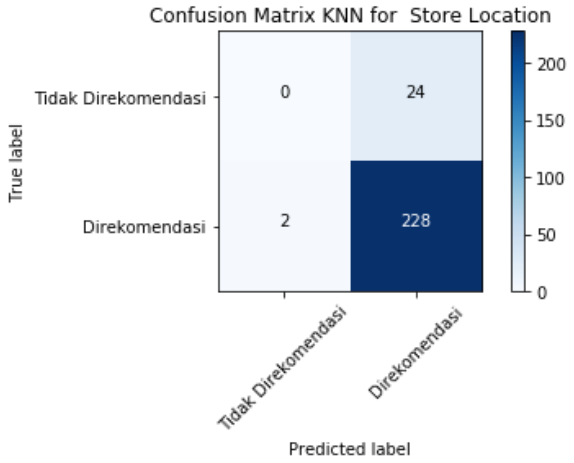

```
True Positives: 226
True Negatives: 5
False Positives: 19
False Negatives: 4
----------------------------------------
Accuracy: 0.91
Sensitivity: 0.98
Specificity: 0.21
Precision: 0.92
f_1 Score: 0.95
```

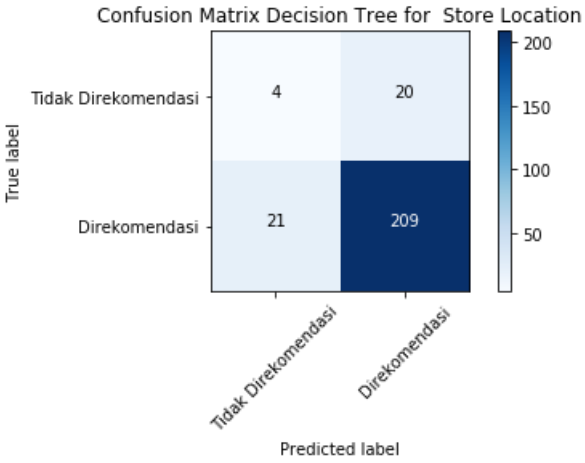


Gambar 11. Confusion Matrix dan *kurva ROC-AUC Logistic Regression.*

Berdasarkan pada Gambar 12, hasil pengujian di atas, menunjukan bahwa akurasi *logistic regression* sebesar 91% dan *AUC* sebesar 0.95.

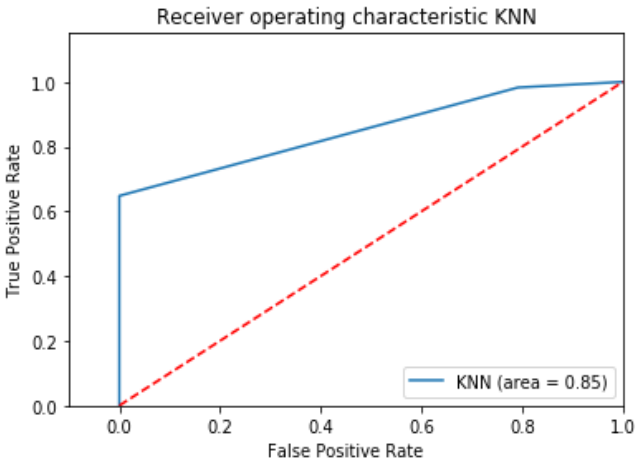
B. Hasil *Confusion Matrix* dan Kurva *ROC* dan *AUC K-Nearest Neigbor (KNN)*



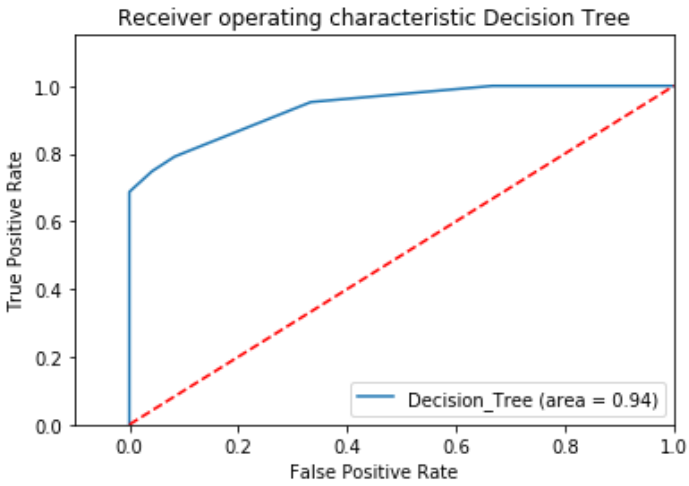



```
True Positives: 228
True Negatives: 0
False Positives: 24
False Negatives: 2
----------------------------------------
Accuracy: 0.9
Sensitivity: 0.99
Specificity: 0.0
Precision: 0.9
f_1 Score: 0.95
```

```
True Positives: 209
True Negatives: 4
False Positives: 20
False Negatives: 21
----------------------------------------
Accuracy: 0.84
Sensitivity: 0.91
Specificity: 0.17
Precision: 0.91
f_1 Score: 0.91
```



Gambar 12. Confusion Matrix dan *kurva ROC-AUC KNN*



Gambar 13. Confusion Matrix dan *kurva ROC-AUC Decision Tree*

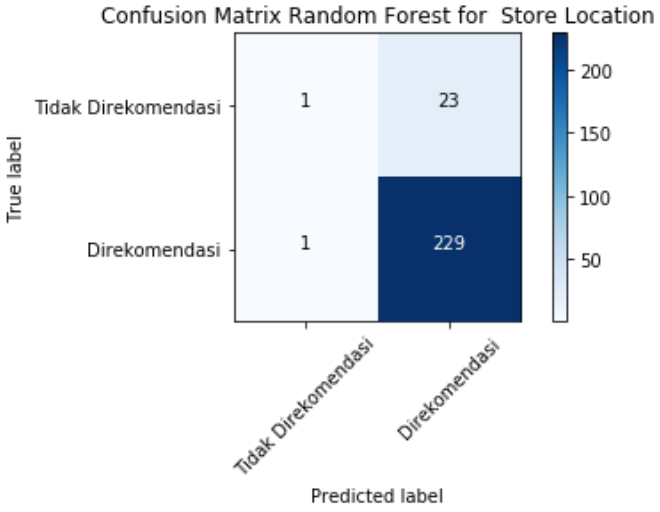
Berdasarkan pada Gambar 13, hasil pengujian di atas, menunjukan bahwa akurasi model *KNN* sebesar 90% dan *AUC* sebesar 0.85.

Berdasarkan pada Gambar 14, hasil pengujian di atas, menunjukan bahwa akurasi model *Decision Tree* sebesar 84% dan *AUC* sebesar 0.94.

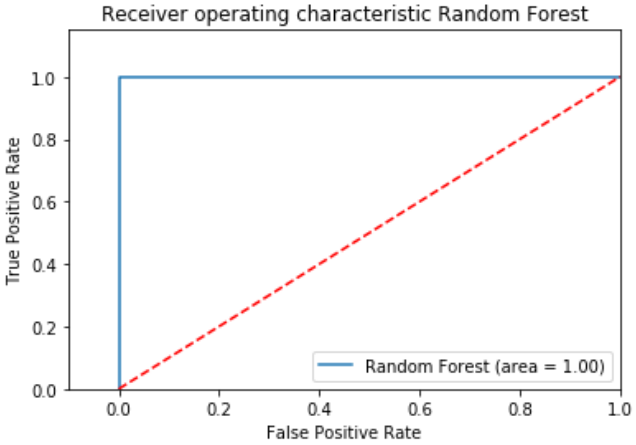
C. Hasil *Confusion Matrix* dan Kurva *ROC* dan *AUC Decision Tree.*

D. Hasil *Confusion Matrix* dan Kurva *ROC* dan *AUC Random Forest.*

p-ISSN : 2443-2210
e-ISSN : 2443-2229




```
True Positives: 229
True Negatives: 1
False Positives: 23
False Negatives: 1
----------------------------------------
Accuracy: 0.91
Sensitivity: 1.0
Specificity: 0.04
Precision: 0.91
f_1 Score: 0.95
```

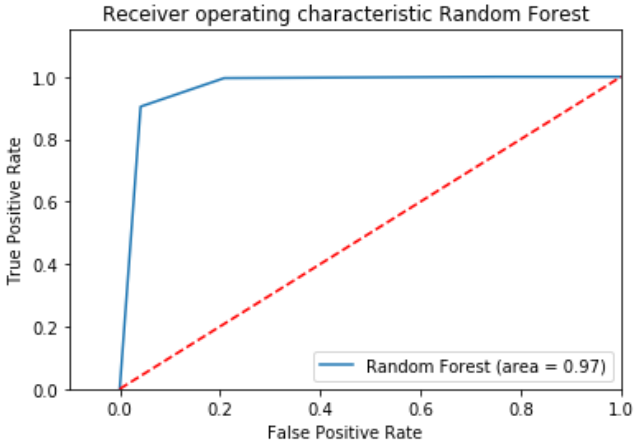


Gambar 14. Confusion Matrix dan *kurva ROC-AUC Random Forest* dengan *n_estimators=100*

Berdasarkan pada gambar 14, hasil pengujian di atas, menunjukan bahwa akurasi model *Random Forest* sebesar 91% dan *AUC* sebesar 1.

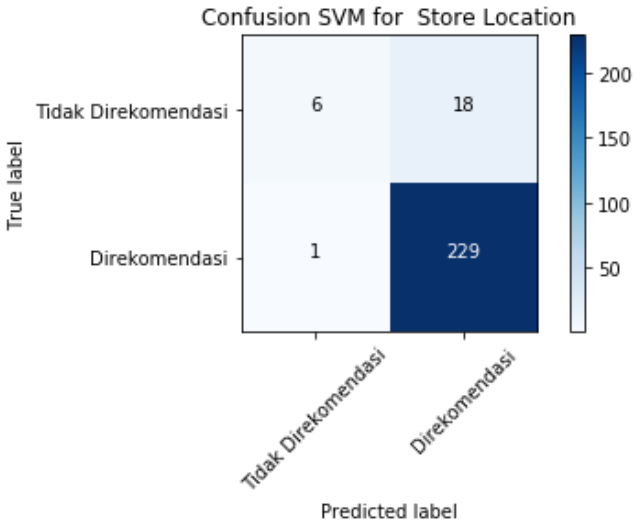
AUC = 1 artinya hasil *True Positive Rate* selalu angka 1 berapapun nilai *False Positive* Ratenya. Maka artinya Pengklasifikasian Random Forest dapat dengan sempurna membedakan antara semua poin kelas Positif dan Negatif dengan benar. Semakin tinggi *AUC*, semakin baik kinerja model dalam membedakan kelas positif dan negatif. Random Forest mendapatkan *AUC* = 1, karena parameter *n_estimators* diset cukup tinggi 100, *n_estimators* menunjukkan jumlah pohon yang digunakan, semakin tinggi nilai *n_estimators* akan memberikan nilai akurasi dan nilai dari kurva *AUC* yang semakin baik tetapi nilai *n_estimators* yang tinggi akan memperlambat proses perhitungan.

Sebagai perbandingan Gambar 15 adalah kurva *AUC* untuk *n_estimators* = 3, dengan *n_estimators* = 3 nilai kurva *AUC* turun menjadi 0.97



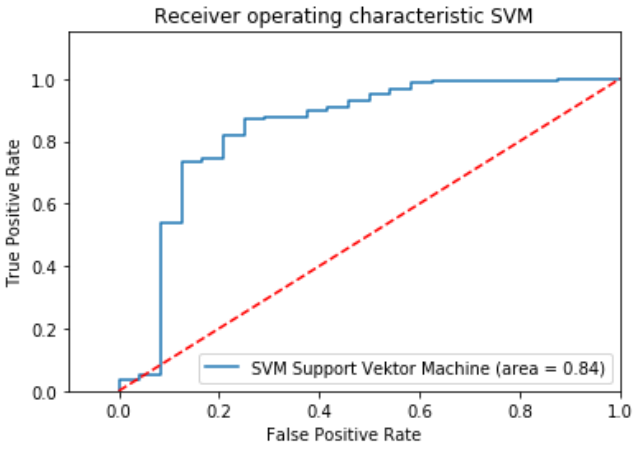
Gambar 15. Confusion Matrix dan *kurva ROC-AUC Random Forest* dengan *n_estimaotors =3*

E. Hasil *Confusion Matrix* dan Kurva *ROC* dan *AUC Support Vector Machine (SVM).*




```
True Positives: 229
True Negatives: 6
False Positives: 18
False Negatives: 1
----------------------------------------
Accuracy: 0.93
```

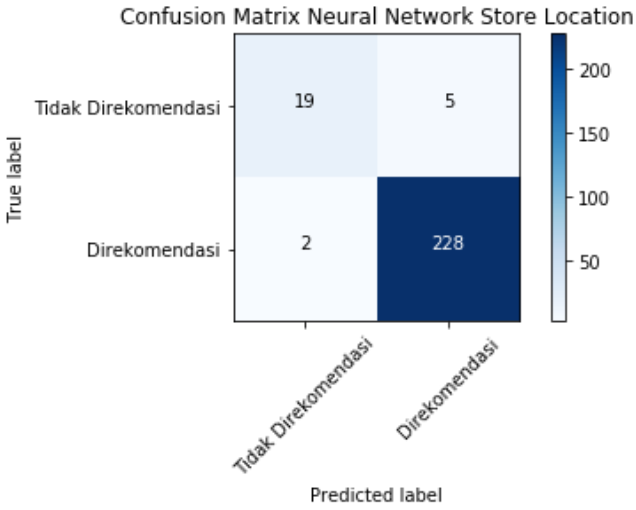
```
Sensitivity: 1.0
Specificity: 0.25
Precision: 0.93
f_1 Score: 0.96
```



Gambar 16. Confusion Matrix dan *kurva ROC-AUC SVM*

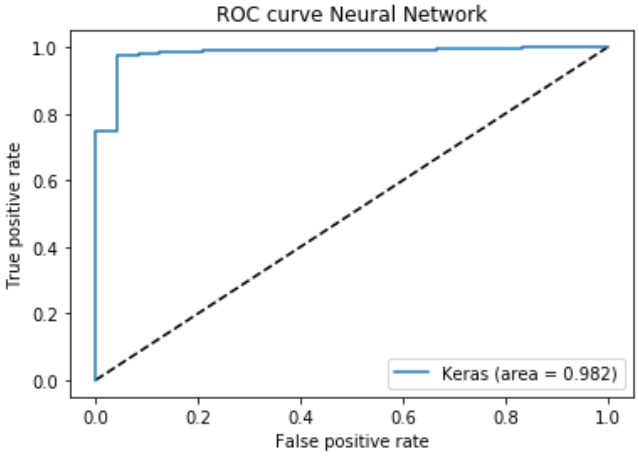
Berdasarkan pada Gambar 16, menunjukan bahwa akurasi *Support Vector Machine (SVM)* sebesar 93% dan *AUC* sebesar 0.84.

F. Hasil *Confusion Matrix* dan Kurva *ROC* dan *AUC Neural Network.*




```
True Positives: 228
True Negatives: 19
False Positives: 5
False Negatives: 2
----------------------------------------
Accuracy: 0.97
```

```
Sensitivity: 0.99
Specificity: 0.79
Precision: 0.98
f_1 Score: 0.98
```



Gambar 17. Confusion Matrix dan *kurva ROC-AUC Neural Network*

Berdasarkan pada Gambar 17, menunjukan bahwa akurasi Neural *Network* sebesar 94 % dan *AUC* sebesar 0.97.

Dari hasil evaluasi keenam algortma tersebut didapat Tabel perbandingan seperti di bawah ini.

TABEL II
PERBANDINGAN HASIL EVALUASI

Algoritma	Accuracy Conf. Matrix	AUC	False Positive
Logistic Regression	91 %	0.95	19
K Nearest Neighbor (KNN)	90%	0.85	24
Decision Tree	84 %	0.94	20
Random Forest	91 %	1	23
Support Vector Machine (SVM)	93 %	0,84	18
Neural Network	97 %	0.98	5

V. SIMPULAN

Dalam Penelitian ini telah dilakukan penelitian menggunakan data lokasi toko untuk membuat model rekomendasi biner (*binary classification*) untuk lokasi toko baru, menggunakan metode klasifikasi dari 6 algoritma yang berbeda yaitu *Logistic Regression, K-Nearest Neighbor (KNN), Decision Tree, Random Forest, Support Vector Machine (SVM) dan Neural Network.*

Berdasarkan hasil evaluasi didapatkan algoritma *neural network* memiliki akurasi paling baik yaitu sebesar 97%, disusul algoritma *SVM* sebesar 93%, kemudian diikuti oleh algoritma *logistic regression* dan *random forest* sebesar 91%, lalu algoritma *KNN* sebesar 90% dan yang terakhir *decision tree* sebesar 84%. Nilai 97% menunjukkan bahwa ketepatan model *neural network* dalam memprediksi satu lokasi direkomendasi atau atau tidak dari total 254 prediksi hanya 7 prediksi yang salah.

Sementara jika dilihat dari kurva *ROC* dan *AUC* yang terbaik adalah algoritma *Random Forest* sebesar 1, disusul oleh *Neural Network* sebesar 0.97, disusul oleh *Logistic Regression sebesar* 0.95, lalu diikuti oleh *Decision Tree* 0.94, setelahnya *KNN* sebesar 0.85 dan terakhir adalah *SVM* 0.84%.

Penelitian uji diagnostic akan semakin baik bila nilai AUC mendekati 1. _Nilai AUC: $0.5 \leq AUC < 0.6$ sangat lemah, $0.6 \leq AUC < 0.7$ lemah, $0.7 \leq AUC < 0.8$ sedang, $0.8 \leq AUC < 0.9$ baik dan $0.9 \leq AUC \leq 1$ sangat baik.

Jadi jika berdasarkan hasil evaluasi, kurva ROC-AUC, model terbaik adalah *Neural Network* lalu *logistic regression*.

Nilai AUC memberikan gambaran tentang keseluruhan perngukuran atas kesesuaian dari model yang digunakan. Semakin besar Area Under Curve (AUC) maka semakin baik model dalam memprediksi lokasi.

Dari sisi bisnis memilih lokasi yang tepat berdasarkan akurasi dan ROC-AUC memang penting, tetapi menghindari pemilihan lokasi yang salah (*false positive*) juga penting bahkan lebih penting, maka dari itu, nilai dari *false positive* pada *confusion matrix* sangat penting untuk juga diperhatikan. Nilai *false positive* penting karena nilai tersebut menunjukkan seberapa banyak sistem melakukan kesalahan dalam memilih lokasi, memprediksi bahwa lokasi tersebut baik padahal kenyataannya buruk. Dibandingkan dengan nilai *fales negative, False positive* jauh lebih penting karena jika satu lokasi diprediksi buruk padahal kenyataannya baik (*false negative)* perusahaan hanya kehilangan kesempatan untuk untung tetapi secara keuangan tidak dirugikan, sebaliknya jika satu lokasi diprediksi bagus padahal kenyataan jelek, maka perusahaan akan mengalami kerugian yang cukup besar bahkan bisa mengganggu keuangan keseluruhan perusahaan.

Berdasarkan hasil evaluasi algoritma yang memiliki nilai *false positive* paling kecil adalah *neural network* yaitu 5, nilainya jauh lebih kecil jika dibandingkan dengan hasil alogritma dari algoritma lainnya. Tapi jika peritel ingin tidak mengalami kerugian dan tidak ingin kehilangan keutungan maka algoritma yang harus dipilih adalah algoritma yang memiliki *F1 Score* tertinggi. Dari Tabel performance terlihat bahwa sistem yang memiliki *F1 Score* tertinggi adalah *neural network* dan random forest.

Jadi jika disimpulkan *neural network* adalah pilihan terbaik untuk pemilihan toko disusul oleh *random forest, SVM* dan *terakhir logistik regression*.

Untuk keperluan penelitan lebih lanjut agar supaya mendapatkan nilai akurasi yang lebih baik maka diusulkan untuk mencoba menggunakan algoritma-algoritma lain seperti *Deep Learning*, menambahkan jumlah data *training* serta menambah fitur-fitur lain seperti pendapatan penduduk di daerah tersebut, jumlah keluarga, fasilitas parkir dan lain-lain.

DAFTAR PUSTAKA

[1] K. Dolan, "Forbes.com," [Online]. Available: https://www.forbes.com/sites/kerryadolan/2020/12/17/billion-dollar-dynasties-these-are-the-richest-families-in-america/?sh=600339da772c.. [Accessed 1 December 2020].

[2] I. Rish, "An Empirical Study of the Naive Bayes Classifier," in *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, IBM, Vol 3*, New York, 2001.

[3] Diana, "Sistem Pendukung Keputusan Menentukan Lokasi Usaha Waralaba Menggunakan Metode Bayes.," *Jurnal Ilmiah MATRIK* , vol. 19, 2017.

[4] J. Brownlee, Master Machine Learning Algorithms, Melbourne, 2017.

[5] D. Solyali, "A Comparative Analysis of Machine Learning Approaches for Short-/Long-Term Electricity Load Forecasting in Cyprus," *Sustainability*, vol. 12, 2020.

[6] J. Ali, R. Khan, N. Ahmad and I. Maq, "Random Forests and Decision Trees," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 5, 12 May 2012.

[7] C. V. L. M. Premalatha, "SVM Trade-Off Between Maximize the Margin and Minimize the Variables Used for Regression," *International Journal of Pure and Applied Mathematics*, vol. 87 , no. 6, December 2013.

[8] B. N. S. C. S. K. S. Harsh Kukreja, "An Introduction to Artificial Neural Network," *International Journal Of Advance Research And Innovative Ideas In Education*, vol. 1, no. 5, April 2016.

[9] A. E. Isabelle Guyon, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research 3*, vol. 11, no. 2, p. 26, November 2003.

[10] D. Berrar, "Cross Validation," *Tokyo Institute of Technology, Tokyo, Japan*, November 2018.

[11] C. G. Gaussier, "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation," *European Conference on Information Retrieval Advances in Information Retrieval*, vol. 3408, pp. 345-359, June 2005.