e-ISSN :
p-ISSN :

JCI, Vol. 1 No. 1 (2021) 12 – 19 | https://doi.org/10.54066/jci.v1i1.209

Jurnal Cakrawala Informasi

Journal Homepage: http://www.itbsemarang.ac.id/sijies/index.php/jci
e-Mail: jci@itbsemarang.ac.id

# Komparasi Algoritma Klasifikasi untuk Penentuan Jenis Spesies Tanaman Hutan

**Ari Putra Wibowo**

STIMIK Widya Pratama Pekalongan

**INFO ARTIKEL**

***Histori artikel:***

Diterima : 15 April 2021
Revisi : 25 Mei 2021
Disetujui : 29 Juni 2021
Publikasi : 30 Juni 2021

*Kata kunci:*
*Algoritma Klasifikasi*
*K-Nearest Neighbor*
*Naive Bayes*
*Random Forest*
*Neural Network*
*SVM*
*Spesies Tanaman Hutan*

***ABSTRACT***

*Determining the types of forest plant species in large numbers (many) will take a long time if done manually. For this reason, it is necessary to have a computational model to clarify the types of forest plant species. Several classification models have been applied to predict forest plant species. In this study, a comparison of classification models was carried out to determine the best model in determining forest plant species. There are 5 classification models used in this study, namely K-Nearest Neighbor (KNN), Naive Bayes, Random Forest, Neural Network and Support Vector Machine (SVM). To evaluate the results used accuracy test and parametric difference test with T-test. The results of this study indicate that the K-Nearest Neighbor (KNN) model has the most dominant values than other models. And for the highest accuracy value is the Neural Network model which reaches 96.95%.*

**ABSTRAK**

Menentukan jenis spesies tanaman hutan dalam jumlah yang besar (banyak) akan membutuhkan waktu yang lama apabila dikerjakan secara manual. Untuk itu, perlu adanya model komputasi untuk melakukan pengklarifikasian jenis spesies tanaman hutan. Beberapa model klasifikasi telah diterapkan untuk melakukan prediksi terhadap jenis spesies tanaman hutan. Pada penelitian ini dilakukan perbandian model klasifikasi untuk mengetahui model yang terbaik dalam penentuan jenis spesies tanaman hutan. Ada 5 model klasifikasi yang digunakan dalam penelitian ini yaitu *K-Nearest Neighbor* (KNN), *Naive Bayes*, *Random Forest*, *Neural Network*, dan *Support Vector Machine* (SVM). Untuk evaluasi hasil digunakan uji akurasi dan uji beda parametrik dengan *T-test*. Dari hasil penelitian ini menunjukkan bahwa model *K-Nearest Neighbor*



* Korespondensi penulis:

(KNN) memiliki nilai dominan paling banyak daripada model lainnya. Dan untuk nilai akurasi tertinggi adalah model *Neural Network* yang mencapai 96.95%.

## PENDAHULUAN

Dalam penelitian penginderaan jauh, autokorelasi spasial biasanya dipertimbangkan dalam akurasi skema penilaian pengambilan sampel untuk memastikan bahwa piksel yang digunakan untuk evaluasi akurasi (uji piksel) tidak terletak terlalu dekat dengan piksel *training* atau piksel *test* lainnya [1]. Gambar (foto) udara yang diambil melalui satelit banyak sekali mengandung informasi apabila diolah sesuai dengan karakteristiknya. Dengan menggunakan foto udara, kategori jenis spesies tanaman hutan yang ada di permukaan bumi dapat diprediksi sesuai dengan letak geografis. Namun, penentuan kategori dari gambar dalam jumlah besar sangat sulit untuk dilakukan karena banyak membutuhkan waktu dan tenaga. Untuk itu, perlu dilakukan model komputasi untuk melakukan penentuan kategori jenis spesies tanaman hutan secara otomatis. Untuk menentukan kriteria kategori jenis spesies tanaman hutan dapat dilakukan dengan menggunakan algoritma klasifikasi *data mining*, kemudian untuk mengetahui algoritma yang sesuai maka perlu dilakukan komparasi algoritma yang pernah digunakan untuk melakukan klasifikasi gambar.

Berbagai penelitian mengenai klasifikasi gambar telah dilakukan sebelumnya. Perbedaan dari setiap penelitian yang dilakukan ada pada *dataset* dan atribut yang digunakan. Perbedaan lainnya juga terdapat pada metode klasifikasi yang digunakan untuk melakukan penelitian. Vailaya, *et al* (2001) melakukan penelitian mengenai pengklasifikasian gambar *indoor* dan *outdoor*, kemudian untuk kelas *outdoor* dibagi menjadi dua kategori kota dan pemandangan [2]. Sedangkan untuk pemandangan dikategorikan lebih lanjut menjadi 3 kategori yaitu matahari terbenam, hutan, dan gunung. Dalam penelitiannya ini Vailaya menggunakan model *bayesian* dan diperoleh hasil akurasi dari kategori *indoor* dan *outdoor* adalah 90,5%, akurasi dari kategori kota dan pemandangan adalah 95,3% dan hasil akurasi dari kategori matahari terbenam, hutan, dan gunung adalah 96,6%.

Johnson, *et al* (2012) dalam penelitiannya menggunakan model *Support Vector Machine* (SVM) untuk menentukan kategori jenis tanaman hutan [3]. Ada 2 kategori jenis tanaman yaitu sugi dan hinoki. Dalam melakukan penelitiannya Johnson, *et al* menggunakan informasi spektral untuk mengklasifikasikan jenis tanaman di sebuah hutan campuran. Dan hasil akurasi dari penelitian yang dilakukan adalah 82,2% kemudian dalam penelitian lebih lanjut peneliti menambahkan atribut geografis dan hasil akurasi dari penelitian mengalangi peningkatan sebesar 85,9%. Perbedaan penelitian yang dilakukan dengan penelitian-penelitian yang telah dilakukan sebelumnya adalah melakukan perbandingan akurasi dan uji parametrik model *K-Nearest Neighbor* (KNN), *Naive Bayes*, *Random Forest*, *Neural Network*, dan *Support Vector Machine* (SVM).

## TINJAUAN PUSTAKA

### *Data Mining*

Data mining merupakan proses untuk menemukan pola (*patter*n) dari suatu data. Pola (*pattern*) yang ditentukan harus memiliki arti atau mengandung informasi penting [4]. *Data mining* juga populer disebut sebagai *Knowledge Discovery from Data* (KDD) yaitu penemuan pengetahuan dari data. KDD adalah ekstraksi pola secara otomatis atau mudah yang mewakili pengetahuan

implisit yang disimpan atau ditangkap di *database* besar, gudang data, *web*, repositori informasi besar lainnya, atau data *stream* [5].

***K-Nearest Neighbor* (KNN)**

Algoritma *K-Nearest Neighbor* termasuk dalam kelas yang disebut *lazy learners*. Jenis teknik ini sebenarnya tidak menentukan model dari *data training*. Mereka hanya menyimpan *dataset* ini. Pekerjaan utamanya terjadi pada saat memprediksi. K yang digunakan untuk *training* adalah yang paling mirip dengan k *testing* yang diberikan untuk memperoleh hasil prediksi. Dalam masalah klasifikasi, prediksi ini biasanya diperoleh dengan memilih angka ganjil untuk k yang diinginkan. Namun, mekanisme *voting* yang lebih rumit yang memperhitungkan jarak kasus uji untuk masing-masing k tetangga juga memungkinkan. Untuk regresi, daripada *voting* kita memiliki rata-rata nilai variabel target k tetangga. Jenis model ini sangat tergantung pada konsep kemiripan. Konsep ini biasanya didefinisikan dengan bantuan metrik melalui ruang *input* yang didefinisikan oleh variable prediktor. Metrik ini adalah fungsi jarak yang dapat menghitung nomor yang mewakili "perbedaan" antara dua pengamatan. Ada banyak fungsi jarak, tapi pilihan yang lebih sering digunakan adalah fungsi jarak *Euclidean* yang didefinisikan sebagai berikut:

$$d(\mathrm{x}_i, \mathrm{x}_j) = \sqrt{\sum_{k=1}^{p} (X_{i,k} - X_{j,k})^2}$$

dimana p adalah jumlah prediktor, dan $x_i$ dan $x_j$ adalah dua hasil pengamatan. Metode tersebut jadi sangat sensitif terhadap kedua metrik yang dipilih dan juga adanya variabel yang tidak relevan yang mungkin mengubah konsep kesamaan. Selain itu, skala variabel harus seragam sebaliknya kita mungkin meremehkan beberapa perbedaan dalam variabel dengan nilai rata-rata yang lebih rendah.

Pemilihan jumlah tetangga (k) juga merupakan parameter yang penting dalam metode ini. Nilai-nilai *frequent* berisi angka dalam himpunan {1, 3, 5, 7, 11}, tetapi jelas ini hanya heuristik. Namun, kita dapat mengatakan bahwa nilai-nilai yang lebih besar dari k harus dihindari karena ada risiko terjadinya kesalahan yang jauh dari kasus uji. Jelas, ini tergantung pada kepadatan *data training*. *Dataset* jarang terkena resiko yang lebih tinggi. Seperti model pembelajaran apapun, pengaturan parameter yang "ideal" dapat diperkirakan melalui beberapa metodologi eksperimental [6].

***Naive Bayes***

Merupakan metode klasifikasi dengan menggunakan probabilitas dan statistik untuk menentukan peluang dimasa depan. Dimana antar atribut tidak saling berkaitan atau tidak menentukan [7]. Teorema *Bayes* mengatakan bahwa:

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A)P(A)}}{\mathrm{P(B)}}$$

| | | |
|---|---|---|
| B | : | label atau kelas pada suatu tabel kasus |
| A | : | atribut dalam suatu tabel kasus |
| P(B) | : | probabilitas dari suatu kelas |
| P(A) | : | probabilitas dari suatu atribut |
| P(A\|B) | : | ketentuan dari atribut A yang memberikan probabilitas pada kelas B |
| P(B\|A) | : | probabilitas dari *class* B yang memberikan ketentuan terhadap atribut A |

Tahapan algoritma *Naive Bayes*, sebagai berikut:

1. Hitung P(B), berapa jumlah kemungkinan dari semua data.

2. Tentukan atribut yang memiliki probabilitas yang sama dengan P(A|B).
3. Hitung hasil dengan probabilitas data sesuai hipotesis.
4. Cari nilai maksimum dari hasil hitung.

***Random Forest***

*Random forest* (Breiman, 2001) adalah contoh dari model *ensemble*, yaitu model yang dibentuk oleh satu set model sederhana [8]. Secara khusus, *random forest* terdiri dari satu set pohon keputusan, baik klasifikasi atau regresi pohon, tergantung pada masalah yang ditangani. Pengguna memutuskan jumlah pohon di *ensemble*. Setiap pohon yang dipelajari menggunakan sampel *bootstrap* yang diperoleh secara acak dengan menggambar kasus N dengan penggantian dari *dataset* asli, dimana N adalah jumlah kasus dalam *dataset* itu. Dengan masing-masing *data training*, pohon yang berbeda diperoleh. Setiap *node* dari pohon-pohon ini dipilih berdasarkan prediktor yang diacak. Ukuran subset ini harus jauh lebih kecil dari jumlah prediktor dalam *dataset* [6].

***Neural Network***

Jaringan Saraf Tiruan atau *Neural Network* adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologi di dalam otak [9]. Nama jaringan saraf tiruan merupakan terjemahan dari "*Artificial Neural Network*". Terjemahan yang diambil bukan jaringan saraf buatan seperti dalam menterjemahkan *Artificial Intelligent* (AI). Jaringan saraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi:

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut *neuron*.
2. Isyarat mengalir di antara sel saraf (*neuron*) melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan atau mengalikan isyarat yang dikirim melaluinya.
4. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.

Perhitungan kesalahan merupakan pengukuran bagaimana jaringan dapat belajar dengan baik sehingga jika dibandingkan dengan pola yang baru akan dengan mudah dikenali. Kesalahan pada keluaran jaringan merupakan selisih antara keluaran sebenarnya (*current output*) dan keluaran yang diinginkan (*desired output*). Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan. *Sum Square Error* (SSE) dihitung sebagai berikut:

1. Hitung keluaran jaringan saraf untuk masukan pertama.
2. Hitung selisih antara nilai keluaran jaringan saraf dan nilai target atau yang diinginkan untuk setiap keluaran.
3. Kuadratkan setiap keluaran kemudian hitung seluruhnya. Ini merupakan kuadrat kesalahan untuk contoh latihan. Dinyatakan dalam rumus:

$$SSE = \sum p \sum j \, (T_{jp} - X_{jp})^2$$

$T_{jp}$ : nilai keluaran jaringan saraf
$X_{jp}$ : nilai target atau yang diinginkan untuk setiap keluaran

*Root Mean Square Error* (RMS *Error*):

1. Hitung SSE.
2. Hasilnya dibagi dengan perkalian antara banyaknya data pada pelatihan dan banyaknya keluaran, kemudian diakarkan.

$$RMS\ Error = \sqrt{\frac{\sum p \sum j\ (Tjp - Xjp)2}{n_p\, n_o}}$$

$n_p$ : jumlah seluruh pola
$n_o$ : jumlah keluaran pola

***Support Vector Machine* (SVM)**

Menurut Santoso (2007) *Support Vector Machine* (SVM) adalah suatu teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi [10]. SVM berada dalam satu kelas dengan *Artificial Neural Network* (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*. Teori SVM dimulai dengan kasus klasifikasi yang secara linier bisa dipisahkan. Dalam hal ini fungsi pemisah yang dicari adalah fungsi linier. Fungsi ini bisa didefinisikan sebagai persamaan *hyperplane* (garis):

$$G = w_1 x_1 + w_2 x_2 + b + 0$$

***Decision Stump***

Operator *decision stump* digunakan untuk menghasilkan pohon keputusan dengan hanya satu split tunggal. Pohon yang dihasilkan dapat digunakan untuk mengklasifikasi contoh yang tak terlihat. Operator ini bisa sangat efisien ketika dikuatkan dengan operator seperti operator *ada boost*. Contoh yang diberikan *ExampleSet* memiliki beberapa atribut dan setiap contoh milik kelas (seperti ya atau tidak). *Node* daun dari *decision tree* berisi nama kelas sedangkan *node* non-daun adalah *decision node*. *Decision node* adalah atribut tes dengan masing-masing cabang (*decision tree* lain) menjadi nilai yang mungkin dari atribut [11].

**Model Evaluasi**

Proses validasi merupakan sebuah proses yang penting dalam klasifikasi. Dalam proses ini digunakan *tool RapidMiner* untuk menangani proses validasi. Penelitian ini akan menggunakan proses validasi yang banyak digunakan pada penelitian *data mining* yaitu dengan menggunakan *cross validation*. *Cross validation* yang paling baik dan banyak dipakai dalam proses validasi *data mining* adalah membagi data menjadi 10 bagian secara acak dan 9 bagian menjadi *data training* sedangkan 1 bagian lainnya akan dijadikan sebagai *data testing*. Proses yang sama diulang selama 10 kali sampai dengan semua data dapat bagian menjadi *data testing*. Varian sederhana merupakan dasar dari teknik statistik yang penting yang disebut *cross validation*. Dalam memvalidasi data harus memutuskan jumlah pengulangan [12]. Pada tabel 1 merupakan gambaran dari tabel 10 *fold cross validation*.

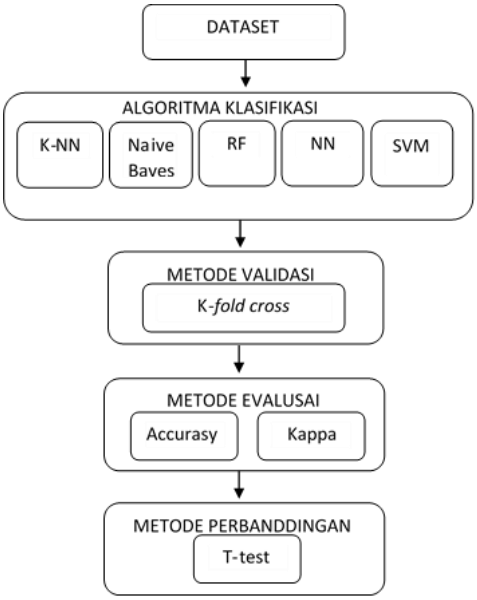
Tabel 1. 10 *Fold Cross Validation*

| Testing | Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ■ | | | | | | | | | |
| 2 | | ■ | | | | | | | | |
| 3 | | | ■ | | | | | | | |
| 4 | | | | ■ | | | | | | |
| 5 | | | | | ■ | | | | | |
| 6 | | | | | | ■ | | | | |
| 7 | | | | | | | ■ | | | |
| 8 | | | | | | | | ■ | | |
| 9 | | | | | | | | | ■ | |
| 10 | | | | | | | | | | ■ |

**METODE PENELITIAN**

*Dataset* yang digunakan pada penelitian adalah *dataset* publik yang diperoleh dari UCI *dataset* kemudian dilakukan proses pemodelan

*data mining* dan validasi, pada penelitian ini penulis menggunakan X-*fold cross validation* dalam melakukan validasi. Selanjutnya dilakukan analisa hasil akurasi masing-masing model untuk dibandingkan model yang terbaik dari penelitian ini. Selain itu dalam penelitian ini juga dilakukan uji parametrik dengan *T-test* untuk membandingkan model, sehingga diharapkan hasil dan uji dari penelitian dapat menjadi referensi dalam menentukan jenis spesies tanaman hutan dengan hasil yang terbaik. Struktur metode penelitian yang dilakukan seperti pada gambar 1.




Gambar 1. Struktur Metode yang Diusulkan

Tabel 2. Struktur *Dataset Forest Type*

| No. | Nama Variabel | Type | Keterangan |
|---|---|---|---|
| 1 | b1 | Integer | Atribut |
| 2 | b2 | Integer | Atribut |
| 3 | b3 | Integer | Atribut |
| 4 | b4 | Integer | Atribut |
| 5 | b5 | Integer | Atribut |
| 6 | b6 | Integer | Atribut |
| 7 | b7 | Integer | Atribut |
| 8 | b8 | Integer | Atribut |
| 9 | b9 | Integer | Atribut |
| 10 | pred_minus_obs_H_b1 | Integer | Atribut |
| 11 | pred_minus_obs_H_b2 | Integer | Atribut |
| 12 | pred_minus_obs_H_b3 | Integer | Atribut |
| 13 | pred_minus_obs_H_b4 | Integer | Atribut |
| 14 | pred_minus_obs_H_b5 | Integer | Atribut |
| 15 | pred_minus_obs_H_b6 | Integer | Atribut |
| 16 | pred_minus_obs_H_b7 | Integer | Atribut |
| 17 | pred_minus_obs_H_b8 | Integer | Atribut |
| 18 | pred_minus_obs_H_b9 | Integer | Atribut |
| 19 | pred_minus_obs_S_b1 | Integer | Atribut |
| 20 | pred_minus_obs_S_b2 | Integer | Atribut |
| 21 | pred_minus_obs_S_b3 | Integer | Atribut |
| 22 | pred_minus_obs_S_b4 | Integer | Atribut |
| 23 | pred_minus_obs_S_b5 | Integer | Atribut |
| 24 | pred_minus_obs_S_b6 | Integer | Atribut |
| 25 | pred_minus_obs_S_b7 | Integer | Atribut |
| 26 | pred_minus_obs_S_b8 | Integer | Atribut |
| 27 | pred_minus_obs_S_b9 | Integer | Atribut |
| 28 | class | Binominal | Label |

Keterangan:

| | | |
|---|---|---|
| b1-b9 | : | informasi spektral |
| pred_minus_obs_H_b1-pred_minus_obs_H_b9 | : | prediksi nilai spektral terhadap kelas ‘s’ |
| pred_minus_obs_S_b1-pred_minus_obs_S_b9 | : | prediksi nilai spektral terhadap kelas ‘h’ |
| class | : | ‘s’ (‘Sugi’ forest), ‘h’ (‘Hinoki’ forest), ‘d’ (‘Mixed deciduous’ forest), ‘o’ (‘Other’ non-forest land) |

## PEMBAHASAN DAN HASIL

Pada penelitian kali ini digunakan *tool Rapidminer Studio* 6.3 yang dijalankan pada sistem operasi *Windows* 8.1 *Enterprise* 32-bit.

### Hasil Akurasi dan *Kappa* Algoritma

Dari hasil penelitian akurasi dan *kappa* dari masing-masing algoritma klasifikasi diperoleh hasil:

Tabel 3. Hasil algoritma *K-Nearest Neighbor*

*Accuracy*: 95,97%; *Kappa*: 0,964

| | true d | true h | true s | true o | class precision |
|---|---|---|---|---|---|
| **pred. d** | 52 | 0 | 0 | 4 | 92.86% |
| **pred. h** | 0 | 48 | 1 | 0 | 97.96% |
| **pred. s** | 1 | 0 | 57 | 0 | 98.28% |
| **pred. o** | 1 | 0 | 1 | 33 | 94.29% |
| **class recall** | 96.30% | 100.00% | 96.61% | 89.19% | |

Tabel 4. Hasil Algoritma *Naive Bayes*

*Accuracy*: 94.97%; *Kappa*: 0,932

| | true d | true h | true s | true o | class precision |
|---|---|---|---|---|---|
| **pred. d** | 51 | 0 | 0 | 2 | 96.23% |
| **pred. h** | 0 | 47 | 4 | 0 | 92.16% |
| **pred. s** | 1 | 1 | 55 | 0 | 96.49% |
| **pred. o** | 2 | 0 | 0 | 35 | 94.59% |
| **class recall** | 94.44% | 97.92% | 93.22% | 94.59% | |

Tabel 5. Hasil Algoritma *Random Forest*

*Accuracy*: 88.29%; *Kappa*: 0,842

| | true d | true h | true s | true o | class precision |
|---|---|---|---|---|---|
| **pred. d** | 51 | 0 | 1 | 8 | 85.00% |
| **pred. h** | 1 | 43 | 6 | 0 | 86.00% |
| **pred. s** | 1 | 5 | 52 | 0 | 89.66% |
| **pred. o** | 1 | 0 | 0 | 29 | 96.67% |
| **class recall** | 94.44% | 89.58% | 88.14% | 78.38% | |

Tabel 6. Hasil Algoritma *Neural Network*

*Accuracy*: 96.95%; *Kappa*: 0,959

| | true d | true h | true s | true o | class precision |
|---|---|---|---|---|---|
| **pred. d** | 53 | 0 | 0 | 1 | 98.15% |
| **pred. h** | 0 | 48 | 1 | 1 | 96.00% |
| **pred. s** | 0 | 0 | 57 | 1 | 98.28% |
| **pred. o** | 1 | 0 | 1 | 34 | 94.44% |
| **class recall** | 98.15% | 100.00% | 96.61% | 91.89% | |

Tabel 7. Hasil Algoritma SVM

*Accuracy*: 84.37%; *Kappa*: 0,790

| | true d | true h | true s | true o | class precision |
|---|---|---|---|---|---|
| **pred. d** | 48 | 4 | 4 | 13 | 69.57% |
| **pred. h** | 0 | 41 | 0 | 0 | 100.00% |
| **pred. s** | 0 | 0 | 54 | 0 | 100.00% |
| **pred. o** | 6 | 3 | 1 | 24 | 70.59% |
| **class recall** | 88.89% | 85.42% | 91.53% | 64.86% | |

Tabel 8. Perbandingan Hasil Pengukuran *Accuracy* dan *Kappa*

| | k-NN | NB | RF | NN | SVM |
|---|---|---|---|---|---|
| *Accuracy* | 95.97 | 94.97 | 88.29 | 96.95 | 84.37 |
| *Kappa* | 0.948 | 0.932 | 0.842 | 0.959 | 0.790 |

Dari hasil pengujian *accuracy* diperoleh bahwa algoritma *Neural Network* memiliki nilai *accuracy* paling tinggi yaitu 96.95% sedangkan untuk nilai *accuracy* terendah dimiliki oleh algoritma *Support Vector Machine* (SVM).

### Hasil *T-Test* Algoritma

Setelah melakukan pengujian *accuracy* dan *kappa*, berikutnya dilakukukan pengujian T-test terhadap algoritma yang digunakan. Untuk hasil pengujian *T-test* dapat dilihat di tabel 9.

Tabel 9. Hasil Uji *T-Test*

| T-TEST | KNN | NB | RF | NN | SVM |
|---|---|---|---|---|---|
| **KNN** | | 0.533 | **0.004** | 0.591 | **0.002** |
| **NB** | | | **0.014** | 0.465 | **0.006** |
| **RF** | | | | **0.004** | 0.454 |
| **NN** | | | | | **0.002** |
| **SVM** | | | | | |

Dari hasil pengujian T-*Test* berdasarkan tabel 9 di atas dapat diketahui, ada perbedaan signifikan antara algoritma *Random Forest* terhadap algoritma K-NN dan algoritma *Naive Bayes*, algoritma Neural Network signifikan terhadap algoritma *Random Forest*, algoritma SVM signifikan terhadap algoritma K-NN, algoritma *Naive Bayes* dan algoritma *Neural Network*. Sehingga didapat urutan algoritma yang baik yaitu algoritma K-NN dan algoritma *Naive Bayes*, selanjutnya algoritma *Random Forest* dan *Neural Network*, terakhir algoritma *Support Vector Machine* (SVM).

## KESIMPULAN

Dari hasil penelitian perbandingan dengan membandingkan 5 algoritma klasifikasi *K-Nearest Neighbor*, *Naive Bayes*, *Random Forest*, *Neural Network*, *Support Vector Machine* (SVM) untuk penentuan jenis tanaman hutan menggunakan uji statistik parametrik (*Paired Sample Test*) terhadap performansi algoritma klasifikasi yang digunakan menunjukan bahwa algoritma *Neural Network* memiliki nilai akurasi tertinggi yaitu 96.95% dan nilai kappa dengan 0.959 dan untuk algoritma dengan performansi terendah adalah *Support Vector Machine* dengan nilai akurasi 84.37% dan nilai *kappa* 0.790.

## DAFTAR PUSTAKA


[1] L. Plourde and R. Congalton, "Sampling Method and Sample Placement: How Do They Affect the Accuracy of Remotely Sensed Maps," *Photogramm. Eng. Remote Sensing*, vol. 69, no. 3, pp. 289–297, 2003.

[2] A. Vailaya, M. Figueiredo, A. K. Jain, and H. J. Zhang, "Image Classification for Content-Based Indexing," *IEEE Trans.*

*Image Process.*, vol. 10, no. 1, pp. 117–130, 2001.

[3] B. Johnson, R. Tateishi, and Z. Xie, "Using Geographically Weighted Variables for Image Classification," *Remote Sens. Lett.*, vol. 3, no. 6, pp. 491–499, 2012.

[4] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining Third Edition*. Elsevier Inc., 2011.

[5] J. Han, M. Kamber, and J. Pei, *Data Mining, Concepts and Techniques*, Third Edit. Massachusetts: Morgan Kaufmann Publishers, 2012.

[6] L. Torgo, *Data Mining with R: Learning with Case Studies*. English: Chapman and Hall, 2010.

[7] T. M. Mitchell, "Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression," in *Machine Learning*, McGraw Hill, 2015, pp. 1–17.

[8] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

[9] A. Kristanto, *Analisa Sistem Informasi*. Yogyakarta: Graha Ilmu, 2004.

[10] B. Santoso, *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*. Yogyakarta: Graha Ilmu, 2007.

[11] M. T. Mohammad *et al.*, "Prediction Of Body Weight from Body Measurements using Regression Tree (RT) Method for Indigenous Sheep Breeds in Balochistan, Pakistan," *J. Anim. Plant Sci.*, vol. 22, no. 1, pp. 20–24, 2012.

[12] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining "Practical Machine Learning Tools and Techniques,"* Third Edit. Burlington: Morgan Kaufmann Publishers, 2011.