

Analisis Performa Random Forest Menggunakan Normalisasi Atribut

By Arie Nugroho, Abdullah Husin

1 Pendahuluan

Dalam data mining menggunakan algoritma *machine learning* untuk membuat model atau pola dari data yang akan digunakan. Dengan *machine learning*, komputer diberi tugas mempelajari data untuk membentuk suatu pola, di mana pola tersebut akan digunakan untuk melakukan prediksi atau klasifikasi pada kasus berikutnya [1]. Tingkat akurasi dari prediksi dan klasifikasi dipengaruhi oleh kualitas dan kuantitas data yang digunakan [2]. Dataset tidak selalu dalam keadaan yang siap diolah. Dataset yang memiliki *missing values*, *noisy data* dan *imbalanced dataset* dapat mempengaruhi hasil klasifikasi dan akurasi[3]. *Missing values* ditemukan jika dalam suatu dataset ada data yang hilang pada suatu atributnya[4]. *Noisy data* ditemukan jika dalam suatu dataset ada data yang di luar range atau tidak berhubungan[5]. Dataset yang mempunyai tipe data nominal cenderung atau rawan dengan inkonsistensi (*noise*) data jika dibandingkan dengan data dengan tipe data numerik[6]. Hal ini terjadi karena isi dari data nominal biasanya lebih dari satu karakter, dimana dalam pengisian datanya akan lebih mempunyai tingkat rawan kesalahan yang lebih, misal penulisan kata “high” tidak sama dengan “tinggi” meskipun mempunyai maksud yang sama, belum lagi jika ada kesalahan dalam penulisan. Untuk mengatasi masalah-masalah tersebut, dibutuhkan *pre-processing* dan pemilihan algoritma *machine learning* untuk klasifikasi yang tepat. *Pre-processing* adalah tahap penting dari proses data mining [7]. Normalisasi atribut termasuk dalam tahap *pre-processing* yang digunakan untuk mempersiapkan data yang digunakan menjadi format tertentu, sehingga proses klasifikasi dapat dikerjakan dengan lebih mudah. Normalisasi atribut adalah proses konversi data untuk suatu atribut dari numerik menjadi skala secara umum dan sebaliknya, tanpa mengganggu variasi datanya[8]. *Imbalanced dataset* terjadi jika dalam suatu dataset perbandingan dari jumlah label yang tidak seimbang atau terlalu jauh[4]. Dalam metode klasifikasi, label dijadikan acuan untuk mengukur performa dalam model yang dihasilkan[9], sehingga jika dalam suatu dataset salah satu label atau class-nya mempunyai jumlah yang lebih besar secara signifikan dibanding label lainnya, maka kemungkinan pengujian atau hasil klasifikasi dengan data uji akan mempunyai kecenderungan lebih dekat pada label yang mempunyai jumlah yang lebih besar pada proses trainingnya. *Pre-processing* juga dapat digunakan untuk dataset yang jumlah labelnya tidakimbang atau *imbalanced dataset* [4]. *Pre-processing* terhadap data yang akan digunakan dapat mempengaruhi tingkat akurasi yang dihasilkan oleh suatu model dari suatu algoritma [5]. Normalisasi atribut pada dataset penting untuk performa model yang lebih baik [6]. Algoritma *machine learning* yang dibahas dalam riset ini adalah salah satu algoritma dalam *ensemble learning method*, yaitu *random forest*. *Random Forest* adalah kombinasi dari beberapa *decision tree*, dimana dapat menyelesaikan permasalahan yang berkaitan dengan klasifikasi dan regresi dan merupakan *ensemble learning method* yang berisi beberapa algoritma dasar, yaitu *decision tree* [10]. *Ensemble learning method* dapat digunakan untuk menangani masalah *imbalanced dataset* [11]. Untuk membuat model dengan *random forest* dataset dipecah menjadi beberapa bagian untuk membangun model *decision tree*, kemudian data testing diuji satu-persatu pada setiap *decision tree*. Setiap *decision tree* akan memberikan hasil atau *vote* berdasarkan model masing-masing. Nilai tertinggi dari *vote* pada setiap *decision tree* akan dijadikan hasil akhir proses klasifikasi [12],[13].

Berdasarkan uraian di atas, tujuan dari penelitian ini adalah membahas pentingnya *pre-processing* dengan normalisasi atribut dan penggunaan *ensemble learning method* yang salah satunya adalah algoritma *random forest* untuk mengatasi masalah *imbalanced dataset*. Diharapkan dengan penelitian ini dapat digunakan sebagai salah satu referensi untuk riset berikutnya dalam implementasi *pre-processing* dan pemilihan algoritma *machine learning* yang sesuai dengan karakter datasetnya, dalam riset ini untuk menangani *imbalanced dataset*.

2 Tinjauan Literatur

Literatur yang terkait dengan penelitian ini adalah beberapa riset tentang klasifikasi evaluasi mobil oleh pelanggan yang juga menggunakan dataset publik yang sama, yaitu data evaluasi mobil dari repository UC Irvine (UCI) *machine learning*, yang mempunyai 1728 data dan 7 fitur. Riset yang pertama adalah membahas perbandingan performa dengan algoritma *decision tree* atau pohon keputusan, *naïve bayes* (NB) dan jaringan saraf tiruan atau *artificial neural network* (ANN) [14]. Dalam riset tersebut menggunakan *pre-processing* data dengan melakukan konversi atribut yang

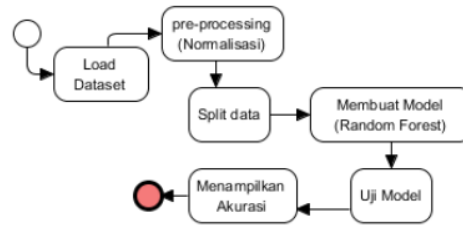
<http://sistemasi.ftik.unisi.ac.id>

sebelumnya berisi nominal menjadi numerik dan proses transformasi data dengan normalisasi min-max. Setelah proses *split* data menjadi data latih atau *training* dan uji atau *testing*, ketiga algoritma tersebut digunakan sebagai membuat model klasifikasi atau *supervised learning* dari data *training* kemudian ketiga model tersebut diuji dengan data testing. Hasilnya adalah ANN memperoleh akurasi tertinggi dari pada kedua algoritma lain yaitu 93 %, akan tetapi memerlukan waktu yang paling lama dalam proses *training* dan *testing*nya. Dari riset tersebut yang belum dilakukan adalah menguji dengan algoritma *random forest* di mana algoritma ini adalah *improvement* dari *decision tree*. Riset yang kedua adalah membahas tentang perbandingan evaluasi performa dari algoritma *Multi Layer Perceptron Neural Network* (MLPNN) dengan *naive bayes* (NB) [15]. Dalam riset tersebut juga menggunakan tahap pre-processing dengan cara konversi dari nominal ke numerik, kemudian menggunakan normalisasi min-max. Tahap berikutnya adalah membagi dataset menjadi 90 dan 10 % sampai 10 *k-folds*, kemudian membuat model dari setiap algoritma. Hasilnya adalah MLPNN mempunyai akurasi yang lebih baik dari NB, akan tetapi juga memerlukan waktu yang lama untuk *training* dan *testing*nya. Riset yang ketiga adalah membahas perbandingan performa dari beberapa 66 algoritma, yang dikategorikan menjadi 7 classifier yang berbeda, yaitu *bayes*, *function*, *lazy*, *meta*, *rules*, *misc* dan *tree* [16]. Hasilnya adalah *classifier tree* dengan algoritma rotation forest mempunyai akurasi tertinggi sebesar 98,61 % akan tetapi memerlukan waktu yang lebih lama dibandingkan classifier yang lain.

Perbedaan riset yang akan dilakukan dengan ketiga riset yang terkait adalah belum menggunakan algoritma *random forest* yang dikombinasikan dengan pre-processing dengan normalisasi atribut pada data yang digunakan. Pre-processing ini dilakukan agar algoritma machine learning dapat bekerja dengan lebih baik. Sebagai indikasi pentingnya pre-processing, akan dibandingkan hasil akurasinya dengan dan tanpa pre-processing menggunakan *random forest* pada dataset atau kasus yang sama.

3 Metode Penelitian

Riset ini menggunakan dataset publik yaitu data evaluasi mobil dari UCI Repository. Dataset ini mempunyai 1728 data, 6 atribut dan 1 label dengan tidak ada nilai yang *null/missing values* [17]. Atribut-atributnya adalah keamanan, ukuran bagasi, jumlah pintu, kapasitas penumpang, perawatan dan harga. Label pada dataset ini adalah class. Atribut keamanan berisi data nominal yang berisi rendah, sedang dan tinggi. Atribut ukuran bagasi juga berisi data nominal, yaitu kecil, sedang dan luas. Atribut jumlah pintu dan kapasitas penumpang berisi 2, 4 dan lebih. Atribut perawatan dan harga berisi rendah, sedang, tinggi dan sangat tinggi. Dataset tersebut mempunyai label yang merupakan ciri dari *supervised learning*. *Supervised learning* membutuhkan manusia untuk memberikan masukan atau input dan output atau hasilnya. Input dan output tersebut akan dijadikan bahan untuk melakukan klasifikasi data berikutnya [18]. Berdasarkan bentuk datasetnya, hasil akhirnya adalah klasifikasi yang nilai setiap data atau datanya adalah salah satu dari keempat pilihan label tersebut. Label dari dataset ini adalah tidak diterima (*un-acc*), diterima (*acc*), baik (*good*) dan sangat baik (*v-good*). Jumlah data untuk label *un-acc* sebanyak 1210 data, label *acc* berjumlah 384 data, label *good* dengan jumlah 69 data dan label *v-good* ada 65 data. Berdasarkan perbandingan jumlah label, ditemukan jumlah label antara yang tidak diterima (*un-acc*) mempunyai jumlah yang lebih tinggi secara signifikan dibandingkan dengan jumlah label yang lain, yaitu banyaknya label *un-acc* lebih dari tiga kali lipat dibandingkan label *acc* dan hampir 18 kali lipat dibandingkan dengan *good* dan *v-good*, sehingga dapat disimpulkan dataset ini mempunyai label yang tidak seimbang atau disebut *imbalanced dataset*. Algoritma yang digunakan dalam riset ini adalah *random forest*, yang merupakan salah satu metode *ensemble learning method* untuk mengatasi *imbalanced dataset*. *Random Forest* merupakan salah satu dari algoritma *machine learning* dengan jenis *supervised learning* untuk kasus klasifikasi yang berbentuk gabungan dari beberapa *decision tree* [19],[20]. Dalam penelitian ini digunakan serangkaian langkah untuk menjelaskan alur penelitian, metode penelitian yang dibahas dalam riset ini ditunjukkan pada Gambar 1.



Gambar 1 Metode Penelitian yang digunakan

Langkah pertama adalah menampilkan data yang akan diproses yaitu data evaluasi mobil yang diambil dari data publik dari web *UCI Repository* dengan alamat <https://archive.ics.uci.edu/ml/datasets/Car+Evaluation>. Pada proses *load data*, hasilnya ditampilkan dengan lima (5) data teratas dari evaluasi mobil ditunjukkan pada Gambar 2. Proses berikutnya adalah *pre-processing* dengan normalisasi atribut, dengan mengubah atau konversi dari tipe data nominal ke numerik. Proses konversi menggunakan *label encoding* dimana setiap data kategori atau nominal diurutkan secara alfabet dan direpresentasikan dengan nilai numerik lebih detailnya ke dalam tipe integer (bilangan bulat). *Label encoding* yang digunakan dalam riset ini memanfaatkan *package* atau modul dari *scikit-learn* pada *class preprocessing*. Berikutnya adalah proses *split data* menjadi *training* dan *testing*. *Split data training* dan *testing* mulai dari 90 % sampai 50 % sebagai data uji dan 10 % sampai 50 % sebagai data uji. Selain *split data*, juga akan dilakukan *cross validation* dengan 8 *k-folds*. Langkah berikutnya adalah membuat model dengan algoritma *random forest*. Berikutnya adalah pengujian model dengan data *testing* dan *cross validation*. Langkah terakhir adalah menampilkan hasil akurasi.

| | buying | maint | door | persons | lug_boot | safety | class |
|---|--------|-------|------|---------|----------|--------|-------|
| 0 | vhigh | vhigh | 2 | 2 | small | low | unacc |
| 1 | vhigh | vhigh | 2 | 2 | small | med | unacc |
| 2 | vhigh | vhigh | 2 | 2 | small | high | unacc |
| 3 | vhigh | vhigh | 2 | 2 | med | low | unacc |
| 4 | vhigh | vhigh | 2 | 2 | med | med | unacc |

Gambar 2 Hasil proses load data

Proses *load* atau menampilkan data menghasilkan *sample* atau contoh dari dataset yang digunakan. Baris teratas adalah atribut-atribut yang digunakan sebagai variabel bebas mulai dari *buying* sampai dengan *safety* dan satu atribut sebagai variabel terikat atau label yaitu *class*. Atribut *buying*, *maint*, *lug_boot*, *safety* dan *class* mempunyai tipe data kategorikal atau nominal, sedangkan atribut *door*, *person* pada hasil di atas mempunyai tipe data numerik, tapi untuk beberapa data yang lain jika isi dari atributnya lebih dari 4 maka isinya adalah *more* atau lebih, sehingga sebenarnya tipe data untuk *door* dan *person* juga adalah nominal.

Setelah proses *load data*, berikutnya adalah tahap *pre-processing*. Pada proses ini dilakukan konversi data dari data nilai kategorikal atau nominal menjadi nilai numerik, disebut juga dengan *categorical encoding*. Tipe *categorical encoding* yang digunakan adalah *label encoding*. Mapping konversi nilai dari nominal ke numerik ditunjukkan pada Tabel 1.

Tabel 1 Mapping Konversi Nilai

| Kategori | Nama | Nilai |
|----------|----------|--------------------------------------|
| Atribut | Safety | low: 1, med : 2, high : 0 |
| | Lug-boot | small: 2, med: 1, big: 0 |
| | Door | 2: 0, 3: 1, 4: 2, 5more: 3 |
| | Person | 0, 4 : 1, more: 2 |
| | Maint | low: 1, med :2, high :0, v-high: 3 |
| | Buying | low: 1, med :2, high :0, v-high: 3 |
| Label | Class | unacc :2, acc :0, good: 1, v-good :3 |

Pada proses normalisasi ini, isi data dari atribut *safety* yang semula mempunyai tipe nominal seperti *low* diganti dengan angka 1, *med* diganti dengan angka 2, *high* dengan angka 0. Pada atribut *lug-boot* yang semula mempunyai tipe data nominal yaitu *small* diganti dengan angka 2, *med* diganti dengan angka 1, *big* diganti dengan angka 0. Pada atribut *door* yang semula mempunyai tipe nominal, yaitu 2 diganti dengan angka 0, 3 diganti dengan angka 1, 4 diganti dengan angka 2, 5more diganti dengan angka 3. Pada atribut *person* yang semula mempunyai tipe data nominal, yaitu 2 diganti dengan angka 0, 4 diganti dengan angka 1, *me* diganti dengan angka 2. Pada atribut *maint* dan *buying* mempunyai tipe data nominal yaitu *low* diganti dengan angka 1, *med* diganti dengan angka 2, *high* diganti dengan angka 0, *v-high* diganti dengan angka 3. Pada label class mempunyai tipe data nominal yaitu *unacc* diganti dengan angka 2, *acc* diganti dengan angka 0, *good* diganti dengan angka 2, *v-good* diganti dengan angka 3. Penentuan konversi ke data numerik menggunakan label encoding berdasarkan urutan alphabet dari data nominal sebelum konversi. Hasil dari proses normalisasi dengan konversi nilai ditampilkan dengan 5 data teratas dan ditunjukkan pada Gambar 3.

| | buying | maint | door | persons | lug_boot | safety | class |
|---|--------|-------|------|---------|----------|--------|-------|
| 0 | 3 | 3 | 0 | 0 | 2 | 1 | 2 |
| 1 | 3 | 3 | 0 | 0 | 2 | 2 | 2 |
| 2 | 3 | 3 | 0 | 0 | 2 | 0 | 2 |
| 3 | 3 | 3 | 0 | 0 | 1 | 1 | 2 |
| 4 | 3 | 3 | 0 | 0 | 1 | 2 | 2 |

Gambar 3 Hasil proses normalisasi

Hasil dari proses normalisasi data menunjukkan perubahan atau transformasi isi data yang semula mempunyai nominal menjadi tipe data numerik atau angka. Hal ini dilakukan untuk konversi. Untuk proses berikutnya yaitu *split* data menggunakan dataset yang telah diproses dengan normalisasi yang semua isi datanya mempunyai tipe numerik.

Pada tahap *split* data, dataset di-split atau dipecah menjadi 2 bagian yaitu data latih dan uji. Data latih atau *training dataset* digunakan dalam membuat model dengan algoritma *random forest* dan data uji atau *testing dataset* digunakan sebagai pengujian dari model yang sudah dibuat. Ada beberapa pembagian untuk *split* data dan *cross validation*, yang ditunjukkan pada Tabel 2. Hal ini dilakukan untuk melihat variasi hasil dari akurasi.

Tabel 2 Split data

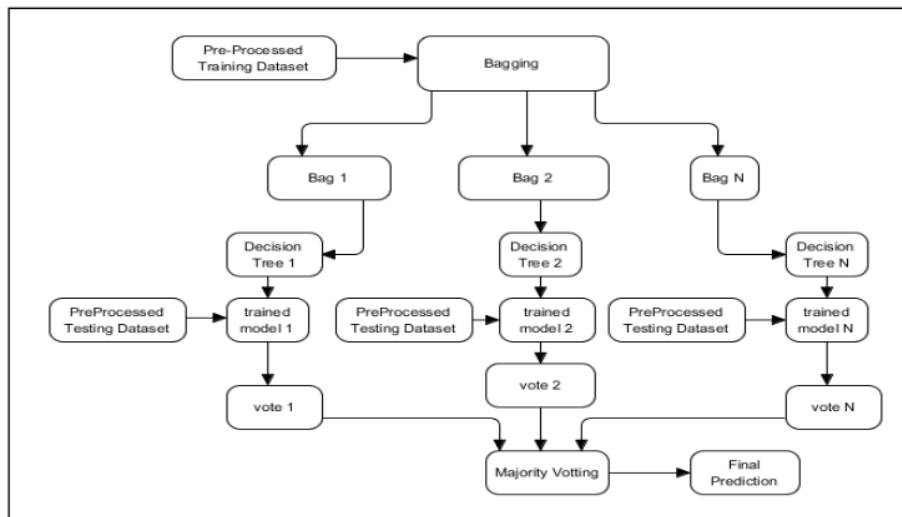
| Training dataset (%) | Testing dataset (%) |
|----------------------|---------------------|
| 90 | 10 |
| 80 | 20 |
| 70 | 30 |
| 60 | 40 |
| 50 | 50 |

8-Fold Cross-Validation

Total data yang digunakan dalam dataset ini adalah 1728 data. Pada training dataset dengan ukuran 90 % mempunyai jumlah data 1555 data, sisanya 10 % dengan jumlah 173 data. Pada training dataset dengan ukuran 80 % mempunyai jumlah data 1382 data, sisanya 20 % dengan jumlah 346 data. Pada training dataset dengan ukuran 70 % mempunyai jumlah data 1209 data, sisanya 30 % dengan jumlah 519 data. Pada training dataset dengan ukuran 60 % mempunyai jumlah data 1036 data, sisanya 40 % dengan jumlah 692 data. Pada training dataset dengan ukuran 50 % mempunyai jumlah data 864 data, sisanya 50 % dengan jumlah yang sama. Pada *cross validation* dengan 8-fold (lapis) berarti dari jumlah seluruh dataset yaitu 1728 data, dibagi menjadi 8 lapis atau bagian, setiap bagiannya berisi 216 data, mulai bagian 1 sampai dengan bagian 8, dimana akan dilakukan 8 kali pengujian. Pada pengujian pertama, bagian 1 sampai dengan bagian 7 dengan jumlah data 1512 dijadikan data latih dan bagian 8 dengan jumlah data 216 data dijadikan data latih. Pada pengujian kedua, semua bagian (1 sampai dengan 8) kecuali bagian ke-7 dijadikan data latih dan bagian ke-7

dijadikan data uji. Pada pengujian ketiga, semua bagian (1 sampai dengan 8) kecuali bagian ke-6 dijadikan data latih dan bagian ke-6 dijadikan data uji. Pengujian ini dilakukan seterusnya sampai dengan pengujian yang ke-8. Nilai akurasi akhir dari *cross validation* didapatkan dari rata-rata akurasi dari 8 kali pengujian.

Langkah berikutnya adalah membuat model dengan algoritma *random forest* dari data *training*. Ukuran data *training* dalam penelitian ini mulai dari 90 % sampai dengan 50 %. Langkah pertama dari *random forest* adalah membuat *bootstrap aggregating (bagging)* dari dataset secara acak. Proses *bagging* pada dasarnya menggunakan *random sampling with replacement* pada dataset yang akan digunakan dan akan menghasilkan sejumlah data *training* baru sejumlah model yang akan di-*training*. Data *training* baru yang dihasilkan dalam proses *bagging* disebut dengan *bag*. Dalam satu *bag* berisi *training* data yang berbeda dengan *bag* lain. Setiap *bag* yang telah dibuat kemudian digunakan untuk men-*training* satu model *machine learning* yang sama, yaitu *decision tree*. Dalam algoritma ini menggunakan 100 pohon atau *tree*, yang merupakan jumlah standart *tree* dalam *random forest*. Dengan menggunakan 100 pohon, berarti dibutuhkan 100 *bag* untuk membuat 100 model *decision tree* yang menghasilkan 100 *trained* model. *Trained* model yang akan dihasilkan akan mempunyai bentuk *decision tree* yang berbeda-beda sesuai *bag* yang sudah ditentukan pada proses *bagging*. Setiap *trained* model digunakan untuk melakukan prediksi terhadap sekumpulan data *testing*. Hasil prediksi dari data *testing* yang dihasilkan oleh setiap *trained* model akan menghasilkan 1 suara (*vote*). Jadi jika ada 100 *trained* model berarti akan menghasilkan 100 *vote*. Proses selanjutnya adalah *majority votting* untuk mengetahui prediksi akhir. *Majority votting* menghitung banyaknya *vote* yang sering muncul sebagai *vote* yang dominan, dimana akan dijadikan hasil klasifikasi akhir atau *final prediction*. Alur algoritma *random forest* ditunjukkan pada Gambar 4.



Gambar 4 Alur Algoritma Random Forest

Pada Gambar 4 dijelaskan alur dari *random forest* yang merupakan salah satu algoritma *ensemble learning methods* dengan teknik *bagging*. *Random forest* yang berarti hutan acak, ditunjukkan pada proses *bagging* yang memilih data dan atribut secara acak dari dataset *training* yang telah di-*preprocessing* dengan normalisasi atribut menjadi *bag-bag* sebanyak jumlah model. Model-model yang telah dihasilkan oleh *decision tree* yang disebut *trained* model diujikan dengan dataset *testing* yang telah di-*preprocessing*. *Vote-vote* yang dihasilkan dari proses *testing* model, kemudian dipertemukan dalam *majority votting* untuk dihitung *vote* mana yang lebih dominan dari *vote* lain, dimana *vote* tersebut akan dijadikan sebagai hasil prediksi akhir.

Berikutnya adalah menguji akurasi dari model yang telah dihasilkan dengan data *testing* dan *cross validation* menggunakan *confusion matrix*. Jumlah *k-folds* dalam *cross validation* yang akan diterapkan adalah 8. Dalam riset ini menggunakan 8 *k-folds* karena berdasarkan riset dengan optimasi

beberapa parameter dari jumlah *tree*, jenis *criterion* dan *k-folds*, didapatkan hasil akurasi tertinggi pada kombinasi jumlah pohon 100, *criterion* gain ratio dan 8 *k-folds*. *Confusion matrix* digunakan untuk evaluasi hasil dari klasifikasi [21]. Dalam *confusion matrix*, setiap kolom atau atribut mewakili kategori data dari yang diprediksi dan setiap barisnya mewakili kategori sebenarnya dari yang diprediksi. Setiap data yang diprediksi akan dihitung jumlah dari prediksi yang sesuai kenyataan benar (True Positive/TP), prediksi yang sesuai kenyataan salah (True Negative/TN), prediksi yang tidak sesuai kenyataan benar (False Positive/FP) dan prediksi yang tidak sesuai kenyataan salah (False Negative/FN). Akurasi dihitung dari hasil penjumlahan TN dan TP dibagi dengan hasil penjumlahan dari TN, TP, FP, FN.

4 Hasil dan Pembahasan

4.1 Hasil Penelitian

Hasil dari riset ini ditunjukkan dengan hasil pengujian dari model yang telah dibuat dengan data *testing* dan *cross validation*. Dalam riset ini ukuran data *testing* yang digunakan mulai dari 10 % sampai dengan 50 % serta 8 *k-folds cross validation*, berdasarkan pembagian data *training* dan *testing* pada penjelasan di Tabel 2. Sebagai pertolongan, ditampilkan hasil pengujian akurasi tanpa dan dengan *pre-processing* normalisasi atribut. Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3 Hasil Pengujian

| Pembagian Data | | Akurasi (%) | |
|----------------|-------------|-------------------|--------------------|
| Training (%) | Testing (%) | Tanpa normalisasi | Dengan normalisasi |
| 90 | 10 | 94,80 | 99,42 |
| 80 | 20 | 93,35 | 99,13 |
| 70 | 30 | 93,63 | 98,26 |
| 60 | 40 | 91,75 | 97,25 |
| 50 | 50 | 91,20 | 96,87 |
| Eight k-folds | | 94,60 | 95,95 |

Pengujian dengan normalisasi dimulai dengan 90 % data *training* dan 10 % data *testing* sejumlah 173 data menghasilkan akurasi sebesar 99,42 %, yaitu 172 data dengan prediksi sesuai dan 1 data dengan prediksi tidak sesuai. Pengujian dengan 80 % data *training* dan 20 % data *testing* sejumlah 346 data menghasilkan akurasi sebesar 99,13 %, yaitu 344 dengan prediksi sesuai dan 2 data dengan prediksi tidak sesuai. Pengujian dengan 70 % data *training* dan 30 % data *testing* sejumlah 519 data menghasilkan akurasi sebesar 98,26 %, yaitu 511 dengan prediksi sesuai dan 8 data dengan prediksi tidak sesuai. Pengujian dengan 60 % data *training* dan 40 % data *testing* sejumlah 692 data menghasilkan akurasi sebesar 97,25 %, yaitu 675 dengan prediksi sesuai dan 17 data dengan prediksi tidak sesuai. Pengujian dengan 50 % data *training* yaitu 864 data dan 50 % data *testing* dengan jumlah yang sama menghasilkan akurasi sebesar 96,87 %, yaitu 832 data dengan prediksi sesuai dan 32 data dengan prediksi tidak sesuai. Pengujian untuk 8 *k-folds cross validation* menghasilkan akurasi sebesar 95,95 %, dengan prediksi yang sesuai kenyataan sejumlah 1174 data untuk *class unacc*, 379 untuk *class acc*, 58 data untuk *class vgood* dan 47 data untuk *class good*. Sedangkan *class* presisi (*class precision*) untuk setiap prediksi adalah 99,19 % untuk *class un-acc*, 87,33 % untuk *class acc*, 85,29 % untuk *class vgood* dan 92,16 % untuk *class good*. *Class recall* dari setiap kenyataan (*true*) adalah 97,02 % untuk *unacc*, 98,70 % untuk *class acc*, 89,23 % untuk *class vgood* dan 68,12 % untuk *class good*.

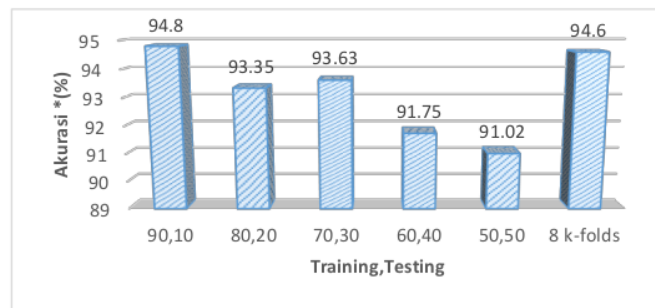
Pengujian tanpa normalisasi dengan 90 % data *training* dan 10 % data *testing* dihasilkan akurasi sebesar 94,80 %, yaitu 164 data dengan prediksi sesuai dan 9 data dengan prediksi tidak sesuai. Pengujian dengan 80 % data *training* dan 20 % data *testing* dihasilkan akurasi sebesar 93,35 %, yaitu 323 dengan prediksi sesuai dan 23 data dengan prediksi tidak sesuai. Pengujian dengan 70 % data *training* dan 30 % data *testing* menghasilkan akurasi sebesar 93,63 %, yaitu 485 dengan prediksi sesuai dan 34 data dengan prediksi tidak sesuai. Pengujian dengan 60 % data *training* dan 40 % data *testing* menghasilkan akurasi sebesar 91,75 %, yaitu 634 dengan prediksi sesuai dan 58 data dengan prediksi tidak sesuai. Pengujian dengan 50 % data *training* dan 50 % data *testing* dihasilkan akurasi

sebesar 91,20 %, yaitu 788 dengan prediksi sesuai dan 76 data dengan prediksi tidak sesuai. Pengujian untuk 8 *k-folds cross validation* menghasilkan akurasi sebesar 94,60 %, dengan prediksi yang sesuai kenyataan sejumlah 1173 data untuk *class unacc*, 362 untuk *class acc*, 58 data untuk *class vgood* dan 52 data untuk *class good*. Sedangkan *class presisi (class precision)* untuk setiap prediksi adalah 99,15 % untuk *class un-acc*, 88,94 % untuk *class acc*, 80,56 % untuk *class vgood* dan 78,79 % untuk *class good*. *Class recall* dari setiap kenyataan(*true*) adalah 96,94 % untuk *unacc*, 94,27 % untuk *class acc*, 89,23 % untuk *class vgood* dan 75,76 % untuk *class good*.

4.2 Pembahasan

Berdasarkan hasil penelitian, model yang dibuat dengan algoritma *random forest* dengan normalisasi atribut pada kasus data evaluasi mobil menghasilkan akurasi yang lebih tinggi daripada tidak menggunakan normalisasi atribut. Model telah diuji dengan pembagian data mulai dari 90 % data *training* dan 10 % data *testing* sampai dengan 8 *k-folds cross validation*.

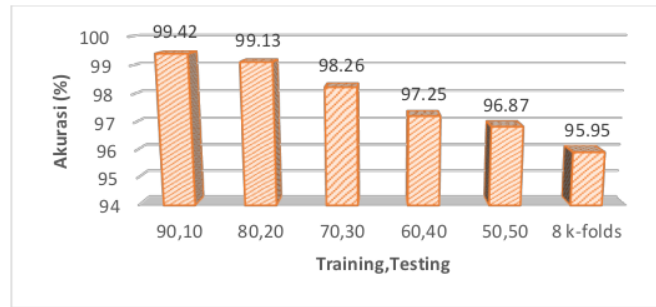
Pada model tanpa normalisasi atribut, akurasi tertinggi ada pada jumlah data *training* sebesar 90 % dan sisanya untuk data *testing*. Untuk pengujian dengan 8 *k-folds cross validation*, data dijadikan *training* dan *testing* secara bergantian hingga 8 lapis atau *folds*. Grafik hasil pengujian tanpa normalisasi ditunjukkan pada Gambar 5.



Gambar 5 Grafik Hasil Pengujian Tanpa Normalisasi

Dari Gambar 5 dapat dijelaskan bahwa jumlah data *training* dan *testing* dapat mempengaruhi nilai akurasi pada model yang dihasilkan. Semakin besar data *training*, akurasi semakin meningkat atau semakin kecil data *training*, akurasi semakin menurun. Dalam hal ini model dapat menghasilkan akurasi yang tinggi jika data yang dijadikan *training* lebih banyak dari data *testing* dengan jarak yang besar. Dengan menggunakan *random forest* pada dataset evaluasi mobil, klasifikasi tanpa normalisasi atribut menghasilkan akurasi di atas 90 %, hal ini menunjukkan meskipun tanpa normalisasi atribut, algoritma *random forest* sebagai *ensemble learning method* dapat menangani dataset yang *imbalanced*.

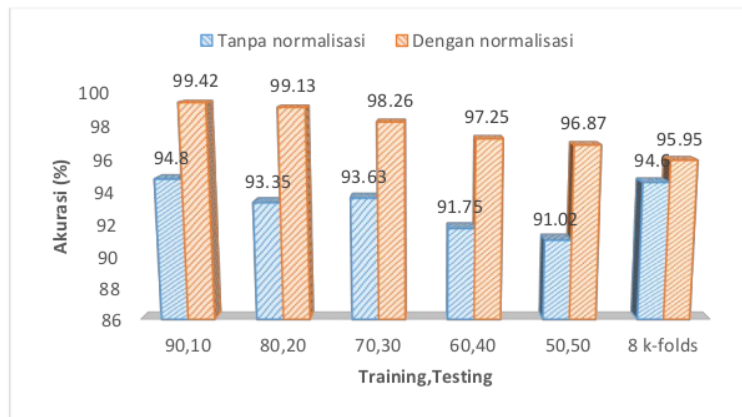
Pada model yang juga dibuat dengan *random forest* dan dikombinasikan dengan normalisasi atribut, akurasi tertinggi juga dihasilkan dengan jumlah data *training* sebesar 90 % dan 10 % untuk data *testing*. Jumlah akurasi terus menurun seiring berkurangnya jumlah data *training* sampai dengan untuk pengujian dengan 8 *k-folds cross validation* dengan akurasi 95,95 %. Grafik hasil pengujian dengan normalisasi ditunjukkan pada Gambar 6.



Gambar 6 Grafik Hasil Pengujian dengan Normalisasi

Dari Gambar 6 dapat dijelaskan bahwa dengan klasifikasi menggunakan algoritma *random forest* dan normalisasi atribut pada dataset yang *imbalanced*, mendapatkan tingkat akurasi yang lebih tinggi daripada klasifikasi tanpa normalisasi atribut dengan algoritma *machine learning* yang sama dimana ditampilkan pada Gambar 5. Hal ini membuktikan bahwa penggunaan algoritma *machine learning* yang sesuai dengan karakter dataset yang digunakan dan dengan normalisasi atribut dapat menambah tingkat akurasi.

Perbandingan hasil dalam bentuk grafik dari penggunaan algoritma *random forest* pada dataset evaluasi penerimaan mobil antara dengan dan tanpa menggunakan normalisasi atribut ditunjukkan pada Gambar 7.



Gambar 7 Perbandingan Hasil Pengujian

Dari Gambar 7 dapat dijelaskan bahwa ada perbedaan hasil akurasi pada algoritma *random forest* dengan dan tanpa menggunakan normalisasi atribut. Warna biru adalah hasil akurasi dari *random forest* tanpa menggunakan normalisasi atribut mulai dari ukuran dataset *training* 90 % sampai dengan 50 % dan 8 *k-folds cross validation*. Warna jingga menunjukkan hasil akurasi *random forest* dengan normalisasi atribut dengan teknik *split* data yang sama. Pada pengaturan ukuran data *training* dan *testing* dapat dijelaskan bahwa semakin besar data *training* maka algoritma *machine learning* dalam hal ini adalah *random forest* dapat melakukan *learning* dengan lebih baik, sehingga dapat menghasilkan akurasi yang lebih tinggi. Penentuan jumlah *k-fold* juga tergantung dari ukuran dataset, semakin besar *k-fold* tidak menjamin hasil akurasi yang lebih baik dan begitu juga sebaliknya, sehingga diperlukan beberapa percobaan untuk mengetahui jumlah *k-fold* yang tepat.

5 Kesimpulan

Berdasarkan hasil riset dan pembahasan, dapat disimpulkan bahwa *pre-processing* dengan normalisasi atribut yaitu mengubah data nominal menjadi data numerik dan penggunaan *ensemble learning method*, yaitu algoritma *random forest* dengan pengaturan parameter tertentu yang

<http://sistemasi.ftik.unisi.ac.id>

diterapkan, dapat menangani *imbalanced dataset* pada dataset evaluasi mobil. Hal ini ditunjukkan dengan hasil akurasi yang tinggi, berdasarkan beberapa percobaan yang telah dilakukan. Dengan menggunakan normalisasi atribut nilai akurasi terendah adalah 95 % dan tertinggi adalah 99 %, sedangkan jika tanpa normalisasi atribut nilai akurasi terendah adalah 91 % dan tertinggi adalah 94 %. Proses *split* data untuk pengaturan ukuran data *training* dan *testing* serta penentuan jumlah *k-fold* dalam *cross validaton* juga mempengaruhi besarnya akurasi. Besarnya akurasi dapat dijadikan acuan bahwa model yang telah dibuat cocok dengan dataset yang digunakan dan dapat digunakan untuk dataset lain yang mempunyai karakteristik yang sama. Pengembangan berikutnya untuk riset ini dapat ditambahkan dengan pelatihan dan pengujian untuk dataset publik lain dalam bidang klasifikasi yang mempunyai *missing values* dan inkonsistensi data. Pengembangan yang lain adalah dengan membandingkan beberapa algoritma klasifikasi atau *ensemble learning method* dan teknik *pre-processing* yang lain untuk mendapatkan variasi hasil atau untuk menghasilkan akurasi yang lebih baik dan sebagai kontribusi ke pengetahuan.

Analisis Performa Random Forest Menggunakan Normalisasi Atribut

ORIGINALITY REPORT

9%

SIMILARITY INDEX

PRIMARY SOURCES

| | | |
|---|--|-----------------|
| 1 | repository.bsi.ac.id Internet | 72 words — 2% |
| 2 | kaksaona.blogspot.com Internet | 64 words — 1% |
| 3 | repo.unikadelasalle.ac.id Internet | 60 words — 1% |
| 4 | sistemasi.ftik.unisi.ac.id Internet | 54 words — 1% |
| 5 | id.scribd.com Internet | 27 words — 1% |
| 6 | docplayer.info Internet | 20 words — < 1% |
| 7 | www.researchgate.net Internet | 18 words — < 1% |
| 8 | Marchell Rianto, Roni Yunis. "Analisis Runtun Waktu Untuk Memprediksi Jumlah Mahasiswa Baru Dengan Model Random Forest", Paradigma - Jurnal Komputer dan Informatika, 2021 Crossref | 14 words — < 1% |

| | | |
|----|--|-----------------|
| 9 | journalhosting.ucalgary.ca Internet | 12 words — < 1% |
| 10 | pt.scribd.com Internet | 11 words — < 1% |
| 11 | text-id.123dok.com Internet | 11 words — < 1% |
| 12 | mirror.its.dal.ca Internet | 10 words — < 1% |
| 13 | repositori.usu.ac.id Internet | 10 words — < 1% |
| 14 | shohib91.blogspot.com Internet | 10 words — < 1% |
| 15 | agungcorneliuspasaribu.wordpress.com Internet | 9 words — < 1% |
| 16 | ejurnal.its.ac.id Internet | 8 words — < 1% |
| 17 | Rivaldi Rivaldi, R Ratianingsih, D Lusiyanti. "APLIKASI MODEL NEURO FUZZY UNTUK PENGONTROL TINGKAT INFLASI DI PROVINSI SULAWESI TENGAH", JURNAL ILMIAH MATEMATIKA DAN TERAPAN, 2017 Crossref | 7 words — < 1% |

EXCLUDE QUOTES OFF

EXCLUDE MATCHES OFF

EXCLUDE BIBLIOGRAPHY OFF