

## Uji Algoritma *Stacking Ensemble Classifier* pada Kemampuan Adaptasi Mahasiswa Baru dalam Pembelajaran *Online*

A K Putri<sup>\*1</sup>, H Suparwito<sup>2</sup>

<sup>1,2</sup>Program Studi Informatika, Universitas Sanata Dharma Yogyakarta

E-mail: anastasia.putri162001@gmail.com<sup>1</sup>, shirsj@jesuits.net<sup>2</sup>

**Abstrak.** Perubahan metode pembelajaran dari sistem kelas ke *online* membawa dampak yang sangat signifikan. Mahasiswa dituntut mampu beradaptasi pada perubahan pola belajar mengajar. Penelitian ini bertujuan untuk melakukan klasifikasi kemampuan adaptasi mahasiswa baru dalam pembelajaran *online* dengan pendekatan *machine learning* menggunakan algoritma *stacking ensemble*. Metode penelitian menggunakan penggabungan *single classifier* dengan teknik *ensemble stacking* atau *stacked generalization* menggunakan *Random Forest*, *Decision Tree*, *K-Nearest Neighbor*, *Support Vector Machine*, dan *Neural Network* sebagai *base learner* dan *Logistic Regression* sebagai *meta learner*. Dari penelitian yang dilakukan, didapatkan *f-1 score* pada *Random Forest* sebesar 89.26%, *Decision Tree* 88.58%, *K-NN* 84.25%, *SVM* 88.98%, *Neural Network* 89.06%, *Logistic Regression* 89.07%, dan *Stacking* 88.86%. Meski dibandingkan dengan *single classifier* seperti *Decision Tree* dan *K-NN*, akurasi pada *Stacking* meningkat, akan tetapi tidak lebih optimal dari *Random Forest*, *SVM*, *Neural Network*, maupun *Logistic Regression*. Validasi keakuratan model menggunakan *Cross Validation* menghasilkan *f-1 score* konstan berada pada angka 88% untuk setiap *n-fold* yang menunjukkan bahwa model *stacking* yang diimplementasikan sudah baik dan stabil. Hal tersebut juga ditunjukkan pada hasil uji stabilitas algoritma *stacking* menggunakan *data random* yang berjumlah 10 dan 5 *record* masing-masing sebanyak 5 kali percobaan, hasil yang didapatkan *f-1 score* konsisten berada pada angka 88%.

**Kata kunci:** kemampuan adaptasi; klasifikasi; pembelajaran mesin; pembelajaran online; *stacking ensemble*

**Abstract.** Changes in learning methods from classroom to online systems have had a very significant impact. Students are required to be able to adapt to changes in teaching and learning patterns. This research aims to classify the adaptability of new students in online learning with a machine learning approach using the ensemble stacking algorithm. The research method used the combination of a single classifier with ensemble stacking or stacked generalization techniques using *Random Forest*, *Decision Tree*, *K-Nearest Neighbor*, *Support Vector Machine*, and *Neural Network* as base learner and *Logistic Regression* as meta learner. From the research conducted, the *f-1 score* obtained on *Random Forest* was 89.26%, *Decision Tree* 88.58%, *K-NN* 84.25%, *SVM* 88.98%, *Neural Network* 89.06%, *Logistic Regression* 89.07%, and *Stacking* 88.86%. Although compared to single classifiers such as *Decision Tree* and *K-NN*, the accuracy of *Stacking* increased, it was not more optimal than *Random Forest*, *SVM*, *Neural Network*, or *Logistic Regression*. Validation of the accuracy of the

*model using Cross Validation resulted in a constant f-1 score of 88% for each n-fold which indicated that the stacking model implemented was good and stable. This was also shown in the results of the stability test of the stacking algorithm using random data totaling 10 and 5 records each for 5 times the experiment, the results obtained by the f-1 score are consistently at 88%.*

**Keywords:** *adaptability; machine learning; online learning; classification; stacking ensemble*

## 1. Pendahuluan

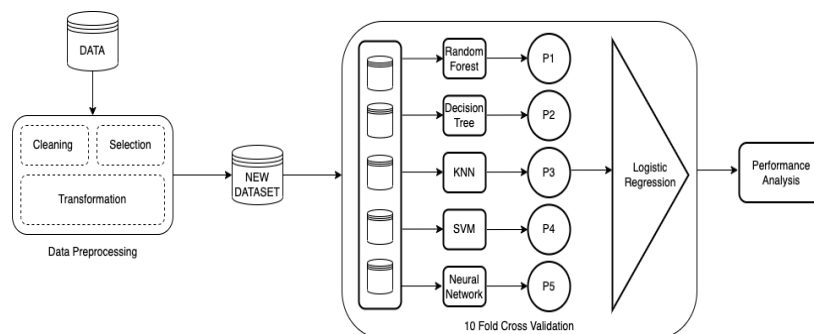
Pandemi Covid-19 berdampak luas pada berbagai bidang kehidupan, salah satunya di bidang pendidikan, yaitu dengan dilaksanakannya pembelajaran *online* bagi para pelajar dan mahasiswa. Adanya perubahan sistem atau transformasi pembelajaran dari yang semula konvensional atau tatap muka di kelas menjadi *online*, tentu saja memerlukan berbagai kesiapan dari peserta didik agar dapat beradaptasi dengan perubahan tersebut. Selain itu, muncul berbagai pro dan kontra yang secara signifikan dialami oleh mahasiswa baru di mana tidak sedikit mahasiswa baru yang merasa kesulitan beradaptasi dari pembelajaran tatap muka di kelas saat SMA/SMK ke pembelajaran *online* saat masuk kuliah [1] dan [2]. Masa-masa adaptasi ini merupakan masa yang tidak mudah dilewati mahasiswa pada semester awal, bahkan dikatakan sebagai masa yang paling sulit [3]. Kemampuan adaptasi mahasiswa baru dalam pembelajaran *online* pun sangat berdampak pada keberhasilan studi, terutama pada hasil belajar dan prestasi mahasiswa yang bisa memengaruhi motivasi atau minat belajar yang dimiliki oleh mahasiswa baru pada semester-semester berikutnya. Dalam penelitian terkait, pendekatan psikologis dengan desain *cross sectional study* atau penggunaan satu variabel independen sebagai fokus studi dan beberapa variabel lain sebagai variabel dependen digunakan untuk menggambarkan adaptasi mahasiswa baru terhadap pembelajaran akademik *online* di Universitas Syiah Kuala [4].

Penelitian ini dilakukan untuk mengklasifikasikan kemampuan adaptasi mahasiswa baru terhadap pembelajaran *online* dengan pendekatan *machine learning* agar dapat mengambil keputusan mengenai faktor-faktor apa saja yang mempengaruhi kemampuan adaptasi mahasiswa baru terhadap pembelajaran *online* termasuk dalam memanfaatkan *Learning Management System* (LMS) sebagai media pendukung pembelajaran. Penelitian yang terkait dengan penggunaan *ensemble classifier* menggunakan metode *stacking* telah banyak dilakukan untuk prediksi atau klasifikasi dalam *machine learning* diantaranya: mengenai klasifikasi penyakit jantung dengan menggunakan metode *ensemble* atau *meta learning* untuk mengatasi ketidakseimbangan kelas pada *single classifier* menggunakan *K-NN*, *SVM*, *C4.5*, dan *Neural Network*. Berdasarkan pengujian yang dilakukan menunjukkan bahwa algoritma *stacking* mampu menghasilkan kinerja dari sisi akurasi [5]. Selain itu, model prediksi berbasis *ensemble classifier* juga dilakukan untuk memprediksi status berlangganan klien bank menggunakan algoritma *Naïve Bayes*, *C4.5* dan *K-NN*. Berdasarkan pengujian yang dilakukan menunjukkan bahwa algoritma *ensemble* mampu menghasilkan kinerja dari sisi *accuracy*, *precision*, dan *recall* yang lebih optimal [6].

Penggabungan metode klasifikasi dengan teknik *ensemble classifier*, yaitu *stacking* atau *stacked generalization* mendapatkan hasil akurasi dan prediksi yang lebih baik [5], [7], [8] dan [9]. Selain itu, dibandingkan dengan metode *ensemble* lainnya yaitu *bagging* dan *boosting*, *stacking* memiliki keunggulan sendiri [10]. Tidak seperti *bagging*, dalam *stacking*, algoritma yang digunakan biasanya berbeda (misalnya tidak semua *decision tree*) dan cocok pada dataset yang sama (misalnya bukan sampel dari dataset *training*). Tidak seperti *boosting*, dalam *stacking*, satu algoritma digunakan untuk mempelajari cara terbaik menggabungkan prediksi dari algoritma yang berkontribusi (bukan urutan algoritma yang mengoreksi prediksi algoritma sebelumnya). Didasarkan pada latar belakang tersebut, dalam penelitian ini akan digunakan algoritma *Random Forest*, *Decision Tree*, *K-NN*, *SVM*, dan *Neural Network* sebagai *base learner* serta *Logistic Regression* sebagai *meta learner* dalam mengimplementasikan salah satu teknik *ensemble*, yaitu *stacking classifier* untuk klasifikasi atau prediksi mengenai kemampuan adaptasi mahasiswa baru terhadap pembelajaran *online*. Data diperoleh dari kuesioner terhadap mahasiswa baru angkatan 2020 dan 2021 di Universitas Sanata Dharma.

## 2. Metode

Tujuan dari penelitian adalah menerapkan pendekatan *machine learning* dengan algoritma *ensemble stacking* untuk mengklasifikasikan mahasiswa baru sebagai mahasiswa adaptif atau non adaptif, menganalisis atribut atribut yang berpengaruh dalam mengklasifikasikan kemampuan adaptasi mahasiswa baru, membandingkan hasil akurasi pada *single classifier* dan hasil akurasi setelah menggunakan *stacking* serta hasil uji stabilitasnya. Data yang digunakan berasal dari jawaban survei kuesioner analisis kemampuan adaptasi mahasiswa baru (angkatan 2020 dan 2021) dalam pembelajaran *online* di Universitas Sanata Dharma, khususnya dalam pemanfaatan LMS-USD. Data berupa skala *Likert* (1 – 4) dengan alternatif pilihan jawaban: Sangat Tidak Setuju, Tidak Setuju, Setuju, dan Sangat Setuju. Data berjumlah sekitar 3438 *record* terdiri dari nim, program studi, tanggal input, 30 pertanyaan kuesioner beserta Indeks Prestasi 2020 Semester 1, indeks prestasi 2020 semester 2, indeks prestasi 2021 semester 1 dan indeks prestasi kumulatif selama mengikuti pembelajaran online, dengan ipk dijadikan sebagai atribut independen atau target label untuk klasifikasi. Selain itu, terdapat atribut nama dan no hp. Untuk dapat memprediksi kemampuan adaptasi mahasiswa tersebut, dilakukan proses klasifikasi seperti gambar di bawah ini.



Gambar 1. Desain Proses Klasifikasi.

Berikut merupakan desain proses klasifikasi beserta penjelasan setiap tahapnya.

### 2.1. Preprocessing

Pada tahap ini dilakukan proses *data cleaning*, *data selection* dan *data transformation* menggunakan modul *preprocessing* dari *library sklearn* menggunakan *Python*.

#### 2.1.1. Data Cleaning

Sebelum dataset digunakan, dilakukan tahap *preprocessing* terhadap data terlebih dahulu, diantaranya adalah pengecekan dan pembersihan pada *missing value*, *incosisten data*, *duplicate data*, diskretisasi dan hal-hal yang perlu dilakukan agar dapat meningkatkan akurasi. Sebelum dilakukan *preprocessing*, dataset awal berjumlah 3437 *record*, kemudian dataset menjadi 3417 *record* yang sudah bersih setelah dilakukan *preprocessing*. Adanya pengurangan data dikarenakan pada data ditemukan *missing value*, *incosisten data*, duplikat data, dan variabel dependen yang yang tidak relevan/berpengaruh terhadap variabel kelas/target juga telah dihapus.

#### 2.1.2. Data Selection

Proses pemilihan data atau atribut yang relevan dilakukan dengan menghapus data *unique* dan perangkan menerapkan Metode Boruta. Untuk memilih data atau atribut yang relevan, maka proses seleksi atribut dilakukan dengan metode Boruta dan jumlah atribut dengan perangkan tertinggi akan digunakan untuk

proses klasifikasi selanjutnya. Boruta sendiri merupakan teknik pemilihan fitur yang menerapkan algoritma *Random Forest* [11].

### 2.1.3. *Data Transformation*

Proses perubahan data-data yang telah dipilih ke dalam bentuk yang sesuai untuk proses *machine learning*, yaitu dengan melakukan transformasi data dari beberapa kolom.

### 2.1.4. *Split Data*

Pembagian data dilakukan sebanyak 3 kali berdasarkan pada *train\_test split* dari *library scikit-learn* menggunakan parameter *test\_size* bernilai 0.2, 0.3, dan 0.4, *random\_state*, dan *shuffle*.

### 2.1.5. *Data Normalization*

Normalisasi data dilakukan pada semua kolom menggunakan *MinMaxScaler*.

## 2.2. *Modelling*

Setelah data melalui proses *preprocessing* maka dilanjutkan dengan *modelling* klasifikasi. Metode yang diusulkan berbasis *ensemble learning* di mana dataset akan diklasifikasikan baik menggunakan masing-masing algoritma klasifikasi *Random Forest*, *Decision Tree*, *K-NN*, *SVM* dan *Neural Network* secara individu juga menggunakan *metode ensemble* menggunakan *stacking* yaitu penggabungan antara algoritma *Random Forest*, *Decision Tree*, *SVM*, *K-NN*, dan *Neural Network* sebagai *base classifier* kemudian hasil dari masing-masing algoritma *base classifier* akan dijadikan input kepada metode selanjutnya yang bertindak sebagai *meta classifier* dalam hal ini adalah *Logistic Regression*.

### 2.2.1. *Machine Learning*

Pembelajaran mesin (*machine learning*) adalah cabang kecerdasan buatan yang tumbuh sangat cepat dan mampu menyelesaikan masalah klasifikasi, regresi, klustering, dan deteksi anomali di berbagai bidang dengan lebih efektif. Misalnya, mengidentifikasi sel abnormal dari *CT scan* untuk membantu dokter mendiagnosis pasien. Identifikasi yang akurat dari jenis dan lokasi objek berdasarkan video digital atau citra satelit telah banyak digunakan di bidang keamanan regional, periklanan, penanggulangan bencana alam dan transportasi cerdas [12].

Metode *machine learning* dapat dilatih dengan berbagai algoritma, masing-masing algoritma memiliki kelebihan dan kekurangan. Metode *machine learning* secara umum dapat dibedakan menjadi tiga jenis, yaitu *Supervised*, *Unsupervised* dan *Reinforcement Learning*.

Langkah-langkah yang dilakukan dalam machine learning adalah sebagai berikut [13]:

- 1) Pengumpulan data dari berbagai sumber;
- 2) Pembersihan data dan rekayasa fitur pada data;
- 3) Pembentukan model menggunakan algoritma yang sesuai;
- 4) Mengevaluasi model;
- 5) Menerapkan model yang telah dievaluasi.

### 2.2.2. *Classification*

Klasifikasi termasuk teknik *supervised learning* di mana variabel dependen diklasifikasikan berdasarkan variabel input lainnya [14]. Klasifikasi adalah proses menemukan model atau fungsi yang dapat membedakan kelas-kelas dari semua data dengan tujuan untuk mengklasifikasikan kelas-kelas objek yang label kelasnya diketahui. Terdapat beberapa algoritma yang umumnya digunakan seperti *Random Forest*, *Decision Tree*, *K-NN*, *SVM*, *Neural Network*, *Logistic Regression* dan lain-lain.

### 2.2.3. Ensemble Method

Metode *ensemble* merupakan penggabungan beberapa individual *classifier* untuk membentuk *classifier* baru sehingga didapatkan hasil yang lebih akurat [15]. Metode *ensemble* telah banyak digunakan dalam beberapa penelitian karena terbukti memberikan hasil akurasi yang lebih baik [16]. Dalam metode *ensemble*, beberapa individual *classifier* digabungkan dengan tujuan untuk menggabungkan kelebihan dari masing-masing *classifier* sedemikian rupa sehingga kinerja yang dicapai dalam menyelesaikan tugas menjadi lebih baik. Contoh metode klasifikasi yang umum adalah: *bagging*, *boosting* dan *stacking* [8].

### 2.2.4. Stacking Ensemble Classifier

*Stacking* atau *stacked generalization* adalah metode lain untuk menggabungkan beberapa pengklasifikasi [8]. Tidak seperti teknik *bagging* dan *boosting*, *stacking* digunakan untuk menggabungkan pengklasifikasi yang berbeda di mana beberapa pengklasifikasi tunggal (*single classifier*) disebut sebagai *base learner* dan pengklasifikasi general (*generalizer*) disebut sebagai *meta learner* [17].

*Stacking* terdiri dari dua tingkatan, yaitu *base learner* pada tingkat nol dan *meta learner* pada tingkat satu. *Base Learner* (tingkat nol) menggunakan model yang berbeda untuk belajar dari kumpulan data. Output dari setiap model dikumpulkan untuk membuat kumpulan data baru. Dalam dataset baru, setiap *instance* sesuai dengan nilai sebenarnya yang seharusnya diprediksi. Model *stacking* kemudian digunakan oleh *meta learner* (tingkat satu) untuk mendapatkan hasil akhir dari dataset [8].

Langkah-langkah *Stacking* dengan *2-fold* adalah sebagai berikut [18]:

1. Pisahkan dataset pelatihan menjadi 2 bagian;
2. Latih setiap estimator pada lapisan pertama menggunakan bagian pertama dari set pelatihan dan buat prediksi untuk bagian kedua;
3. Latih setiap estimator pada lapisan pertama menggunakan bagian kedua dari rangkaian pelatihan dan buat prediksi untuk bagian pertama;
4. Gunakan prediksi ini untuk melatih estimator pada lapisan berikutnya;
5. Latih masing-masing estimator lapisan pertama dengan seluruh set pelatihan.

## 2.3. Performance Analysis

### 2.3.1. Confusion Matrix

*Confusion matrix* merupakan salah satu cara untuk menghitung akurasi dengan memberikan informasi tentang bagaimana hasil klasifikasi yang dibuat oleh sistem dibandingkan dengan klasifikasi yang sebenarnya (*ground truth*). *Confusion matrix* berupa tabel matriks yang menggambarkan performansi model klasifikasi dengan sekumpulan data uji yang nilai sebenarnya diketahui. Tabel 1 merupakan contoh *confusion matrix*.

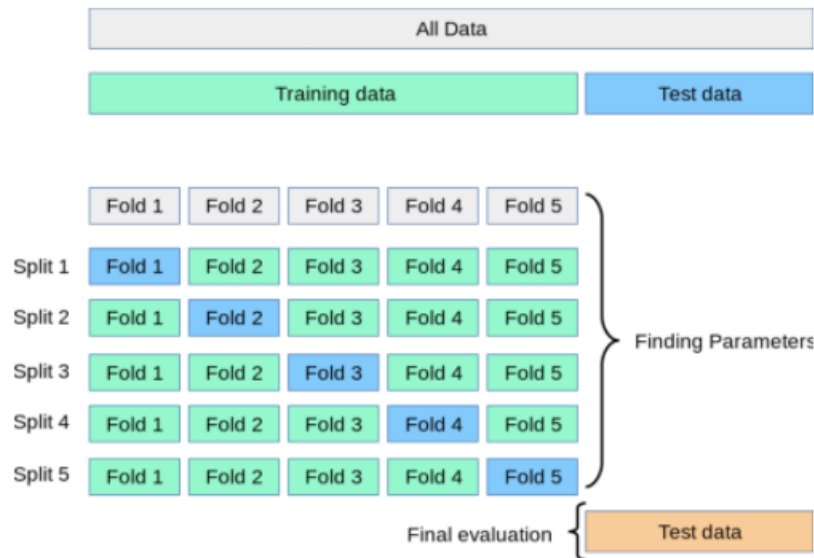
**Tabel 1.** Confusion Matrix

Confusion Matrix		Kelas Prediksi	
		Kelas = Positif	Kelas = Negatif
Kelas Aktual	Kelas = Positif	TP ( <i>True Positive</i> )	FN ( <i>False Negative</i> )
	Kelas = Negatif	FP ( <i>False Positive</i> )	TN ( <i>True Negative</i> )

### 2.3.2. Cross Validation

*Cross-validation* adalah metode untuk mengevaluasi algoritma pembelajaran melalui validasi berulang, di mana kumpulan data dibagi menjadi beberapa himpunan bagian (set) data latih dan validasi. Setiap iterasi memvalidasi satu subset dan subset yang tersisa sebagai data pelatihan. Selain itu, pemilihan jenis validasi silang dapat didasarkan pada ukuran dataset validasi silang, yang bertujuan untuk mengurangi waktu perhitungan dengan tetap menjaga keakuratan estimasi. Misalnya model dilatih dengan *5-fold*, maka

kemudian data dibagi menjadi 5 bagian dengan ukuran yang sama, di mana 4 bagian digunakan sebagai data latih dan 1 bagian digunakan sebagai data uji. Contoh penggunaan validasi silang dapat dilihat pada Gambar 2.



**Gambar 2.** Cross Validation 5-fold.

Sumber: [https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html)

### 3. Hasil dan Pembahasan

#### 3.1. Hasil Training Model

Base model yang dipakai adalah *Random Forest*, *Decision Tree*, *K-NN*, *SVM*, dan *Neural Network*. Pembagian data latih dan data uji menggunakan parameter *test\_size* yang tersedia pada *library Python* yaitu *sklearn.model\_selection.train\_test\_split*. Selanjutnya, masing-masing model pada *base classifiers* akan disimpan hasil prediksinya sebagai input data latih untuk *meta classifier*. Terakhir, hasil prediksi pada *meta classifier* yang telah tersimpan divisualisasikan untuk evaluasi. Setelah evaluasi, kemudian dilakukan *cross validation* untuk memvalidasi keakuratan model. Pada penelitian ini, penulis melakukan 2 kali percobaan dengan 7 dan 10-fold untuk *cross validation*. Tabel 2 di bawah ini merupakan hasil *train model stacking* menggunakan *test\_size* yang berbeda.

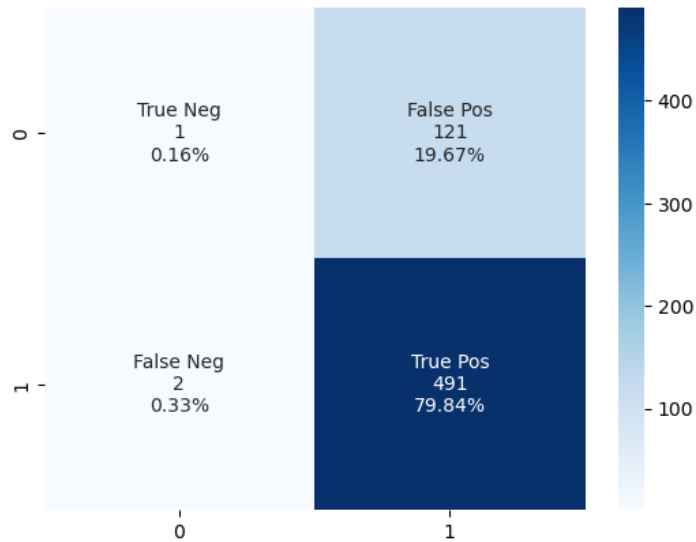
**Tabel 2.** Perbandingan *F-1 Score Training Model Stacking*.

<i>Training</i>	<i>test_size</i>	<i>F-1 Score</i>
I	0.2	88.13%
II	0.3	88.86%
III	0.4	88.26%

Berdasarkan Tabel 2 di atas, nilai *f-1 score* optimal berada pada eksperimen klasifikasi dengan *test\_size* 0.3 yaitu dari *Training II* dengan *score* 88.86%.

### 3.2. Hasil Evaluasi Model

*Confusion Matrix* untuk model *Stacking Training II* ditunjukkan pada Gambar 3 di bawah ini.



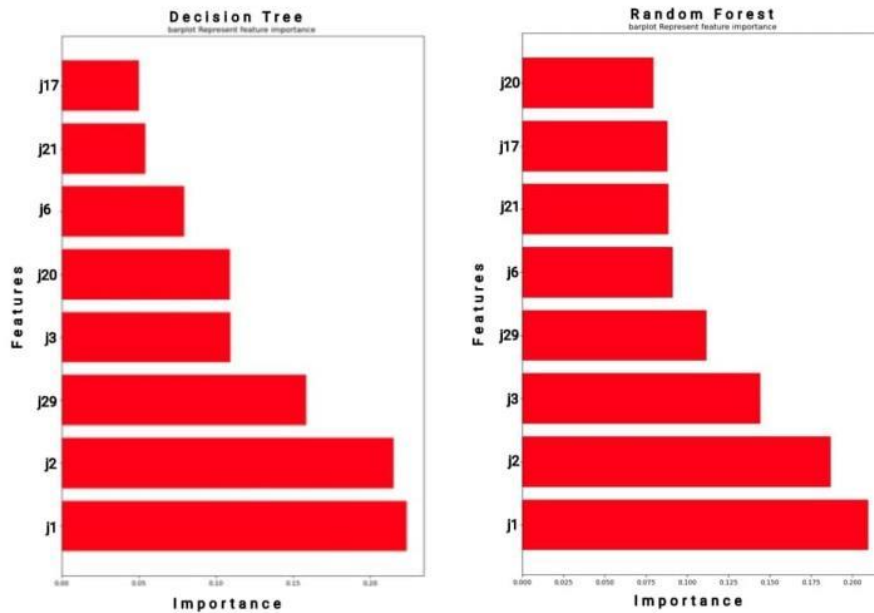
**Gambar 3.** *Confusion Matrix Stacking.*

Pada Gambar 4 matriks *accuracy* digunakan sebagai acuan performansi algoritma jika dataset memiliki jumlah data *False Negative* dan *False Positive* yang sangat mendekati (*symmetric*). Namun berdasarkan *confusion matrix* di atas, karena jumlahnya tidak mendekati, maka penulis menggunakan *f-1 score* sebagai acuan. Hasil perhitungan *confusion matrix* pada tiap model ditunjukkan pada Tabel 3 berikut.

**Tabel 3.** Hasil Perhitungan *Confusion Matrix.*

Algoritma	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F-1 Score</i>
Random Forest	80.81%	80.88%	99.59%	89.26%
Decision Tree	79.83%	81.11%	97.56%	88.58%
K-NN	73.98%	81.83%	86.81%	84.25%
SVM	80.32%	80.69%	99.18%	88.98%
Neural Network	80.32%	80.29%	100%	89.06%
Logistic Regression	80.48%	80.82%	99.18%	89.07%
Stacking	80.00%	80.22%	99.59%	88.86%

Dari beberapa model yang digunakan, *Random Forest* (RF) dan *Decision Tree* (DT) menghasilkan *feature importance* yang ditunjukkan pada Gambar 4 di bawah ini.



**Gambar 4.** Perbandingan *Feature Importance* DT dan RF.

Pada *Random Forest*, atribut j1 menempati posisi sebagai atribut yang paling penting dalam mengklasifikasikan kemampuan adaptasi mahasiswa. Begitu pula pada model *Decision Tree*, atribut j1 sebagai atribut yang paling penting. Untuk memperjelas hasil dari *feature importance* yang dihasilkan dari kedua algoritma tersebut, Tabel 4 berikut merupakan perbandingan *feature importance* dari algoritma *Random Forest* dan *Decision Tree* berdasarkan urutan dari paling tinggi ke rendah nilai *importance* yang dihasilkan.

**Tabel 4.** Perbandingan *Ranking Feature Importance* pada RF dan DT.

<i>Ranking</i>	<i>Random Forest</i>	<i>Decision Tree</i>
1	j1	j1
2	j2	j2
3	j3	j29
4	j29	j3
5	j6	j20
6	j21	j6
7	j12	j21
8	j20	j17

Berdasarkan Tabel 4, meskipun urutan atribut yang dihasilkan pada kedua algoritma berbeda, akan tetapi atribut-atribut yang dihasilkan sebagai *feature importance* sama, misalnya atribut j1 pada *Random Forest* juga dihasilkan dari *Decision Tree*, atribut j2 pada *Random Forest* juga dihasilkan dari *Decision Tree* dan seterusnya. Penjelasan dari masing-masing atribut ditunjukkan oleh Tabel 5 berikut.



**Tabel 5.** Keterangan Atribut.

Atribut	Keterangan
j1	Saya terbiasa menggunakan <i>gadget</i> /teknologi seperti: komputer, HP, tablet, untuk mengakses Google, website pengetahuan, <i>games</i> , dll.
j2	Saya terbiasa menggunakan aplikasi Zoom, WA, email dll. sebagai sarana yang mendukung pembelajaran <i>online</i> .
j3	Fasilitas internet dan teknologi tersedia dengan baik di rumah, dan saya pergunakan untuk belajar <i>online</i> .
j6	Saya merasa mudah beradaptasi dari pembelajaran tatap muka di kelas saat SMA ke pembelajaran <i>online</i> saat masuk kuliah di USD.
j12	Saya senang jika LMS-USD digunakan untuk menyampaikan materi kuliah dan mengumpulkan tugas/ujian.
j20	LMS-USD sangat efektif karena dapat membantu saya dalam proses belajar baik saat kuliah maupun di rumah.
j21	Fitur dan menu yang ada di LMS-USD mudah digunakan dan dipahami.
j29	Setiap semester saya melihat dan mengenali ada perubahan atau penambahan fitur fasilitas di LMS-USD sehingga menjadi lebih baik.

### 3.3. Hasil Testing Model

Setelah dilakukan *training* dan evaluasi pada model menggunakan *test\_size* yang berbeda, didapatkan hasil yang optimal berada pada *test\_size* 0.3 yang ditunjukkan pada Tabel 2 dengan *f-1 score* 88.86%, maka selanjutnya penulis melakukan uji coba menggunakan *data testing* yang berbeda dan hasilnya disajikan pada Tabel 6.

**Tabel 6.** Perbandingan *F-1 Score Testing Model Stacking*.

<i>Testing</i>	<i>F-1 Score</i>
I	88.48%
II	88.60%
III	88.31%

### 3.4. Hasil Validasi Model

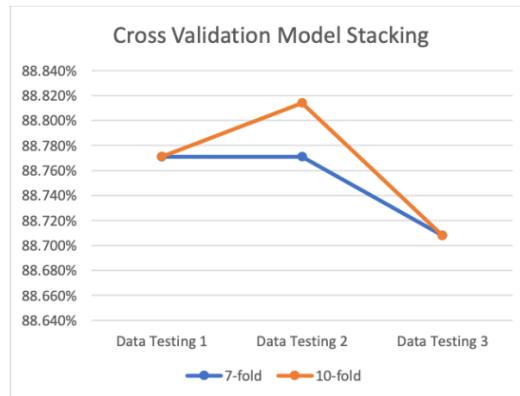
Berdasarkan hasil *testing* di atas, hasil *testing* menggunakan *data testing* yang berbeda menghasilkan akurasi dan *f-1 score* yang berbeda. Tabel 6 menunjukkan *f-1 score* yang paling optimal pada *Testing* II, yaitu sebesar 88.60% dan akan dilakukan validasi keakuratan *model stacking* menggunakan *cross validation*.

**Tabel 7.** Perbandingan *F-1 Score Validasi Model Stacking*.

Validasi	<i>n-fold</i>	<i>F-1 Score</i>
I	7	88.77%
II	10	88.81%

Berdasarkan penerapan *n-fold* untuk validasi pengujian klasifikasi pada Tabel 7 di atas, nilai optimal ditunjukkan pada Validasi II dengan 10-*fold* dan *f-1 score* 88.814%. Validasi model menggunakan

Cross Validation mengalami peningkatan akurasi yang signifikan pada model *stacking* seperti yang ditunjukkan pada Gambar 5 di bawah ini.



**Gambar 5.** Hasil *F-1 Score* Cross Validation Stacking.

### 3.5. Hasil Uji Stabilitas Model

Seperti yang disajikan pada Gambar 5 dari hasil validasi model, *f-1 score* berada pada angka 88% yang menunjukkan bahwa model *stacking* yang diimplementasikan sudah stabil untuk setiap *n-fold*. Selain itu, untuk menunjukkan bahwa model *stacking* yang diimplementasikan sudah stabil, penulis melakukan uji stabilitas menggunakan 10 dan 5 *record random data* dengan menunjukkan baris indeks yang digunakan, kemudian dilakukan eksperimen sebanyak 5 kali percobaan lalu hasil yang disajikan berupa *mean accuracy*. Tabel 8 dan Tabel 9 di bawah ini merupakan hasil dari pengujian *data random* yang telah dilakukan.

**Tabel 8.** Hasil Uji Stabilitas Stacking 10 Random Record.

Pengujian	10 Record Acak	<i>F-1 Score</i>
I	141, 620, 576, 1109, 73, 629, 845, 978, 1183, 820	71.42%
II	1361, 674, 289, 1013, 329, 394, 1133, 279, 951, 660	94.11%
III	371, 105, 1275, 883, 644, 1269, 222, 511, 296, 6	94,73%
IV	73, 428, 468, 105, 1336, 905, 548, 1290, 458, 642	88.88%
V	146, 624, 21, 634, 976, 271, 342, 975, 390, 612	94,73%
	<i>Mean</i>	88.77%

**Tabel 9.** Hasil Uji Stabilitas Stacking 5 Random Record.

Pengujian	5 Record Acak	<i>F-1 Score</i>
I	188, 831, 169, 402, 583	88.88%
II	434, 1088, 631, 1012, 31	75.00%
III	215, 566, 402, 637, 425	88.88%
IV	110, 1073, 710, 987, 1077	100.00%
V	623, 1193, 1027, 187	88.88%
	<i>Mean</i>	88.32%

Setelah dilakukan pengujian dari hasil evaluasi dan validasi model, kemudian dilakukan uji stabilitas dan hasil yang didapatkan menunjukkan bahwa nilai *f-1 score* juga konsisten berada pada angka 88%.

#### 4. Kesimpulan

Pendekatan *machine learning* dengan metode *Ensemble Stacking* berhasil mengklasifikasikan mahasiswa baru sebagai mahasiswa adaptif atau non adaptif saat mengikuti pembelajaran *online* dengan tingkat akurasi sebesar 88.81%.

Atribut yang paling berpengaruh dalam proses klasifikasi kemampuan adaptasi mahasiswa baru terhadap pembelajaran *online* berdasarkan hasil perankingan atribut menggunakan Boruta yaitu atribut j1, j2, j3 yang berkaitan dengan kebiasaan dalam menggunakan teknologi dan fasilitas internet yang dimiliki; atribut j6 yang berkaitan dengan adaptasi mahasiswa dari SMA ke perguruan tinggi; atribut j20, j21, dan 29 yang berkaitan dengan *Learning Management System* kampus sebagai media pembelajaran *online*; dan atribut j17 yang berkaitan dengan kemandirian mahasiswa dalam pembelajaran. Pada *Random Forest* dan *Decision Tree*, atribut j1 menempati atribut teratas untuk *feature importance*.

#### Referensi

- [1] M. W. S. Huda and A. Hidayat, "Quo Vadis Pendidikan di Masa Pandemi: Menyoal Tanggung Jawab Negara terhadap Hak atas Pendidikan Peserta Didik: Quo Vadis Education in a Pandemic Period: Questioning the State's Responsibility for the Right to Education of Students," *Seminar Nasional Hukum Universitas Negeri Semarang*, vol. 7, no. 1, Art. no. 1, Jul. 2021, doi: 10.15294/snhunnes.v7i1.711.
- [2] H. N. Z.r and W. Saugi, "Pengaruh Kuliah Online Terhadap Minat Belajar Mahasiswa Pendidikan Agama Islam (PAI) di IAIN Samarinda," *el-Buhuth: Borneo Journal of Islamic Studies*, pp. 121–131, Jun. 2020, doi: 10.21093/el-buhuth.v2i2.2330.
- [3] R. Setiawan, "Apa itu Data Mining dan Bagaimana Metodenya?," *Dicoding Blog*, Oct. 29, 2021. <https://www.dicoding.com/blog/apa-itu-data-mining/> (accessed Oct. 03, 2022).
- [4] L. Loreni and S. R. Jannah, "Tingkat Adaptasi Mahasiswa Baru Terhadap Pembelajaran Akademik Secara Daring," *Jurnal Ilmiah Mahasiswa Fakultas Keperawatan*, vol. 5, no. 3, Art. no. 3, Dec. 2021, Accessed: Oct. 03, 2022. [Online]. Available: <http://jim.unsyiah.ac.id/FKep/article/view/19074>
- [5] A. Nurmasani and Y. Pristyanto, "Algoritme Stacking Untuk Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class," *pseudocode*, vol. 8, no. 1, Art. no. 1, Mar. 2021, doi: 10.33369/pseudocode.8.1.21-26.
- [6] D. Novianti, "Prediksi Status Berlangganan Klien Bank Menggunakan Algoritma Naive Bayes, C4.5, Dan Knn Berbasis Ensemble Classifier," 2019.
- [7] R. Jayapermana, "Implementasi Stacking Ensemble Classifier Untuk Klasifikasi Multi Kelas Topik Vaksin Covid-19 Pada Twitter," sarjana, Universitas Siliwangi, 2021.
- [8] D. D. Sidik and T. W. Sen, "Penggunaan Stacking Classifier Untuk Prediksi Curah Hujan," 2019, Accessed: Dec. 21, 2022. [Online]. Available: <http://repository.president.ac.id/xmlui/handle/123456789/3598>
- [9] Y. Pristyanto, A. Sidauruk, and A. Nurmasani, "Klasifikasi Penyakit Diabetes Pada Imbalanced Class Dataset Menggunakan Algoritme Stacking," *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, Art. no. 1, Jan. 2022, doi: 10.30865/mib.v6i1.3442.
- [10] A. F. Nugraha, R. F. A. Aziza, and Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan*, vol. 7, no. 1, 2022.
- [11] A. Velia, T. R. Simamora, S. N. Suherman, A. A. Pravitasari, and F. Indrayatna, "Klasifikasi Customer Churn Pada Perusahaan Telekomunikasi Menggunakan Support Vector Machine," *E-*

- Journal BIAStatistics / Departemen Statistika FMIPA Universitas Padjadjaran*, vol. 2022, no. 1, Art. no. 1, Sep. 2022, doi: 10.1234/bias.v2022i1.154.
- [12] Y. Heryadi and T. Wahyono, *Machine Learning: Konsep dan Implementasi*. 2020.
- [13] E. Retnoningsih and R. Pramudita, “Mengenal Machine Learning Dengan Teknik Supervised Dan Unsupervised Learning Menggunakan Python,” *Bina Insani Ict Journal*, vol. 7, no. 2, Art. no. 2, Dec. 2020, doi: 10.51211/biict.v7i2.1422.
- [14] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*. Springer Science & Business Media, 2011.
- [15] I.- Ati and A. Kusyanti, “Metode Ensemble Classifier untuk Mendeteksi Jenis Attention Deficit Hyperactivity Disorder (SDHD) pada Anak Usia Dini,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 3, Art. no. 3, May 2019, doi: 10.25126/jtiik.2019631313.
- [16] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka Jr., “Tweet sentiment analysis with classifier ensembles,” *Decision Support Systems*, vol. 66, pp. 170–179, 2014, doi: 10.1016/j.dss.2014.07.003.
- [17] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.
- [18] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.