JIKO (Jurnal Informatika dan Komputer)
Vol. 6, no. 2, September 2022, hal. 251–260
e-ISSN : 2477-3964 — p-ISSN : 2477-4413

OPTIMASI CORRELATION-BASED FEATURE SELECTION UNTUK PERBAIKAN AKURASI RANDOM FOREST CLASSIFIER DALAM PREDIKSI PERFORMA AKADEMIK MAHASISWA

Yoga Priantama[1*], **Taghfirul Azhima Yoga Siswa**[2]
[1,2]Universitas Muhammadiyah Kalimantan Timur
Email: yogapriantama@gmail.com[1], tay758@umkt.ac.id[2],

Abstrak

Kegiatan pembelajaran sejak pandemi covid-19 melanda Indonesia mengharuskan institusi pendidikan melaksakanan kegiatan pembelajaran secara online atau daring. Learning Management System (LMS) menjadi salah satu solusi untuk mendukung proses pembelajaran daring. Universitas Muhammadiyah Kalimantan Timur (UMKT) memanfaatkan plat-form LMS OpenLearning sebagai upayanya untuk menjaga agar kegiatan pembelajaran tetap berlangsung dimasa pan-demi. Tujuan penelitian ini adalah untuk mengidentifikasi indikator atau atribut yang berpengaruh menggunakan metode correlation-based feature selection dan mengevaluasi kinerja algoritma Random Forest Classifier untuk memprediksi per-forma akademik mahasiswa UMKT dalam pembelajaran daring berbasis LMS OpenLearning. Pada penelitian ini, data diperoleh dari bagian administrasi akademik dan LMS OpenLearning sebanyak 2.663 data. Hasil penelitian menunjukan identifikasi korelasi atribut terbaik menggunakan correlation-based feature selection (CFS) adalah pada atribut time spent on course, course completed, tugas, uts, dan quiz. Hasil pemodelan Random Forest Classifier menggunakan optimasi CFS terbukti dapat memperbaiki akurasi pemodelan sebesar 97,22%, sedangkan pemodelan tanpa menggunakan optimasi CFS menghasilkan akurasi sebesar 91,66%.
Kata Kunci: *correlation-based feature selection, data mining, learning management system, performa akademik, random forest classifier*

Abstract

Learning activities since the COVID-19 pandemic hit Indonesia have required educational institutions to carry out online learning. Learning Management System (LMS) is one of the solution to support learning process. The University of Muhammadiyah East Kalimantan (UMKT) utilizes the LMS OpenLearning platform as an effort to keep learning activities going during the pandemic. The purpose of this study was to identify indicators that has influence and develop the Random Forest Classifier algorithm to predict academic performance of UMKT students in LMS-based learning activities. In this study, the data obtained from the academic administration and LMS OpenLearning were 2,663 data. The results showed that the best attribute correlation using CFS was time spent on courses, completed courses, assignments, uts, and quizzes. The results of Random Forest Classifier modeling using CFS optimization are proven to improve modeling accuracy by 97.22%, while modeling without using CFS optimization produces an accuracy of 91.66%.
KeyWords : *correlation-based feature selection , data mining, learning management system, performa akademik, random forest classifier*

I. PENDAHULUAN

Dunia pendidikan di masa pandemi covid-19 yang telah melanda dunia sejak awal tahun 2020 menemui sejumlah tantangan yang harus dihadapi oleh pelaku pendidikan. Kegiatan pembelajaran yang biasanya dilakukan dengan tatap muka secara langsung, kini harus dilakukan secara *online* atau daring. Survey menunjukan bahwa 73,2% peserta didik merasa terbebani dalam melakukan PJJ [1]. Permasalahan ini dapat menjadi sebuah ancaman bagi institusi pendidikan dalam upaya untuk menjaga kualitas pembelajaran untuk mendapatkan lulusan yang berkualitas. Untuk menjaga kualitas pembelajaran khususnya pada kegiatan pembelajaran daring, perlu dilakukan evaluasi dan monitoring sejak dini terhadap performa peserta didik dalam mengikuti pembelajaran. Manfaat utama dari evaluasi pada kegiatan belajar mengajar adalah meningkatkan kualitas pembelajaran yang selanjutnya akan mengakibatkan peningkatan kualitas pendidikan [2]. Demi menjaga kualitas pembelajaran daring, maka dirasa perlu untuk melakukan prediksi terhadap performa akademik peserta didik dalam kegiatan pembelajaran khususnya di Universitas Muhammadiyah Kalimantan Timur (UMKT).

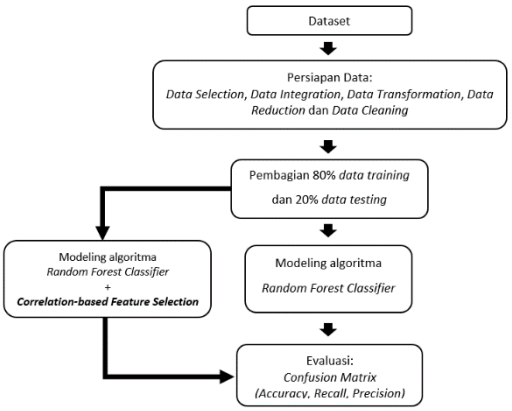
Penyelenggaraan kegiatan pembelajaran di UMKT pada masa pandemi covid-19 mengusung pembelajaran daring dengan menggunakan *platform Learning Management System* (LMS) bernama *OpenLearning* untuk menunjang modul pembelajaran, penugasan, serta penilaian mahasiswa. Salah satu metode yang dapat digunakan untuk melakukan prediksi adalah dengan menggunakan pendekatan *data mining*. *Data mining* adalah proses yang melakukan pekerjaan satu atau lebih teknik *machine learning* atau pembelajaran komputer yang dapat digunakan untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) dari kumpulan data yang dilakukan secara otomatis [3]. Dalam *platform OpenLearning*, terdapat data rekam jejak mahasiswa UMKT yang dapat dimanfaatkan sebagai indikator dalam melakukan prediksi terhadap performa akademik pembelajaran mahasiswa.

Penelitian terkait mengenai prediksi menggunakan indikator *e-learning* pada dasarnya sudah pernah dilakukan oleh Xing et. al. [4] dengan menggunakan indikator *Number of discussion post*, *Number of forum views*, *Number of quiz views*, *Number of module views*, *Number of active days*, dan *Social network degree* pada pembelajaran berbasis MOOCs atau *Massive Open Online Courses* untuk kebutuhan memprediksi tingkat *drop out* mahasiswa menggunakan komparasi metode *Gaussian Bayesian Network* dan C4.5 dengan hasil akurasi sebesar 80.7% dan 96.1%. Kemudian, pada penelitian yang dilakukan Abubakar & Ahmad [5], digunakan indikator *e-learning* lainya seperti *Course View*, *Assign View*, *Assign_submit_update*, *Resource View*, *Forum View*, *Overall Score* PT I (*Programming Technique* I), *Overall Score* PT II (*Programming Technique* II), *Assignment*, *Lab Total*, dan *Mid Term* dalam prediksi terhadap performa peserta didik pada pembelajaran berbasis *e-learning* menggunakan komparasi metode *Naïve bayes*, KNN, *Decision Tree*, dan *Random Forest* dengan hasil akurasi 92.3%, 69.2%, 61.5%, dan 76.9%.

Metode yang akan digunakan pada penelitian ini adalah metode *Random Forest*. *Random Forest* merupakan metode hasil pengembangan dari algoritma *Classification And Regression Tree* (CART) yang pada penerapannya menggunakan metode *bootstrap aggregating* (bagging) dan *random feature selection* [6]. Pada penelitian-penelitian sebelumnya, metode *Random Forest* telah banyak digunakan untuk memprediksi kinerja akademik peseta didik dalam dunia pendidikan [7] [8]. Namun, yang menjadi pembeda antara penelitian ini dengan penelitian sebelumnya adalah studi kasus dan indikator yang digunakan untuk prediksi performa peserta didik. Penelitian ini mengangkat studi kasus pada Universitas Muhammadiyah Kalimantan Timur dan indikator atribut data yang diperoleh dari *platform OpenLearning* dalam memprediksi performa akademik mahasiswa. Sehingga penelitian ini memunculkan pembahasan penelitian baru yang belum pernah diteliti oleh peneliti-peneliti sebelumnya. Oleh karena itu, pada penelitian ini dilakukan prediksi performa akademik mahasiswa menggunakan metode *Random Forest* dengan mengeksplorasi beberapa indikator *e-learning* diluar penelitian-penelitian sebelumnya. Adapun tujuan dari penelitian ini adalah untuk mengidentifikasi indikator atau atribut yang berpengaruh menggunakan metode *correlation-based feature selection* dan mengevaluasi kinerja algoritma pemodelan *Random Forest Classifier* dalam memprediksi baik buruknya performa akademik mahasiswa UMKT pada pembelajaran daring berbasis LMS *OpenLearning*.

II. METODE

Penelitian ini menggunakan data yang diperoleh dari bagian administrasi akademik (BAA) UMKT dan *platform OpenLearning*. Data yang dikumpulkan merupakan data akademik mata kuliah "kewarganegaraan" tahun akademik 2020/2021 dan 2021/2022. Setelah data dikumpulkan, selanjutnya dilakukan proses persiapan data yang meliputi *data selection*, *integration*, *transformation*, *reduction*, dan *cleaning*. Sebelum memasuki pemodelan data akan melalui proses pembagian data masing-masing menjadi 80% *data training* dan 20% *data testing*. Tahapan pemodelan yang dilakukan pada penelitian ini menerapkan algoritma *Random Forest Classifier* untuk menangani data akademik mahasiswa UMKT dalam pembelajaran daring. Pada pemodelan *Random Forest Classifier*, algoritma CART digunakan untuk membangun pohon dengan aturan *Gini Impurity* untuk menentukan pecahan dari pohon keputusan [9]. Proses analisa pemodelan *Random Forest Classifier* yang dilakukan pada penelitian ini menggunakan bahasa pemrograman python. Evaluasi pemodelan yang dilakukan pada penelitian ini akan menggunakan metode *confusion matrix*, pada tahapan evaluasi ini juga akan dibandingkan hasil akurasi pemodelan menggunakan seleksi fitur *correlation-based feature selection* (CFS) dan tanpa seleksi fitur CFS.



Gambar 1: Alur Penelitian

A. Random Forest

Random Forest merupakan metode hasil pengembangan dari algoritma *Classification And Regression Tree* (CART) yang pada penerapannya menggunakan metode *bootstrap aggregating* (*bagging*) dan *random feature selection* [8]. Menurut Jonathan [9],

Random Forest biasa digunakan untuk menyelesaikan masalah yang berhubungan dengan klasifikasi dan regresi. Perumusan untuk *Random Forest* yang terdiri dari *n trees* dinyatakan pada persamaan (1) [10]:

$$I_(y) = argmax_c(\sum_{n=1}^{N} Ih_n(y) = c) \quad (1)$$

Dimana variabel l adalah fungsi indikator dan ffi$_n$ merupakan tree ke-n dari algoritma *Random Forest*.

Metode CART digunakan untuk membangun pohon pada algoritma *Random Forest Classifier*, dengan menggunakan aturan *Gini Impurity* untuk menentukan pecahan dari pohon keputusan [11]. Perhitungan dimulai dengan penentuan nilai *Gini Index* untuk menentukan distribusi probabilitas atribut terhadap kelas target dan dilanjutkan pada perhitungan *Gini Impurity*. Persamaan (2) menyatakan rumus perhitungan *Gini Index*:

$$Gini = \sum_{i=1}^{n} P_{i^2} \quad (2)$$

Dimana:
n = Merupakan jumlah kelas target
i = Merupakan kelas target
P = Merupakan rasio kelas target

Adapun perhitungan Gini Impurity dinyatakan pada persamaan (3):

$$Gini\ Impurity = 1 - \sum_{i=1}^{n} P_{i^2} \quad (3)$$

B. Studi Kasus

Pembentukan algoritma *Random Forest* dilakukan dengan membentuk n pohon keputusan dengan aturan algoritma *Classification and Regression Tree* (CART) dan pemilihan dataset secara acak menggunakan metode *bootstrap aggregating*. Hasil dari masing-masing pohon keputusan yang didapatkan dengan menggunakan algoritma CART akan melalui proses pemungutan suara terbanyak atau *majority voting*. Pada kasus prediksi performa akademik mahasiswa UMKT dalam pembelajaran daring berbasis *OpenLearning* akan diambil 10 sampel dataset untuk pembuktian perhitungan algoritma *Random Forest*. Sampel dataset acak dapat dilihat pada Tabel I.

Tabel I: Sampel dataset acak

Time spent on course	*Course Completed*	Tugas	UTSQuiz	Simbol
5	6	0	00	BURUK
24	18	0	00	BURUK
26	16	14	320	BURUK
56	27	20	00	BURUK
89	89	30	8684	BAIK
112	90	52	7884	BAIK
114	75	78.5	7670	BAIK
155	97	81.5	8470	BAIK
189	91	53	8675	BAIK
338	82	81.7	6466	BAIK

Setelah dataset siap, kemudian dilakukan perhitungan pohon keputusan CART dengan melakukan penentuan *node* akar atau *root node*. Perhitungan pohon keputusan CART dimulai denagn melakukan perhitungan aturan *gini index* sebagai berikut:

Gini index = 1−(Jumlah kelas BAIK/(Total Kelas BAIK dan BURUK))^2−((Jumlah kelas BAIK /(Total Kelas BAIK dan BURUK)))^2

Kemudian, perhitungan dilanjutkan dengan melakukan perhitungan gini impurity sebagai berikut:

Gini Impurity = 1 – (Total nilai calon cabang kiri)^2− (Total nilai calon cabang kanan)^2

Adapun perhitungan *Gini index* dan *Gini Impurity* terhadap data sampel akan disajikan pada Tabel II:

Tabel II: Hasil Perhitungan Gini index dan Gini Impurity

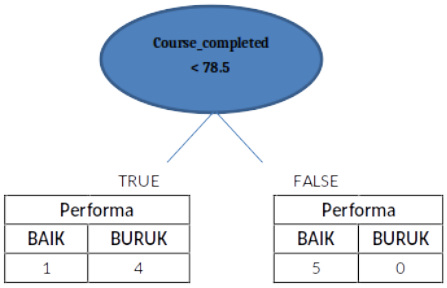
record	nilai ***midpoint***	**<<Kurang dari Performa**		**Lebih dari >> Performa**		*Gini* **(<<Calon Cabang Kiri)**	*Gini* **(Calon Cabang Kanan)**	**Total *Gini Impurity***
		BAIK	**BURUK**	**BAIK**	**BURUK**			
time_spent_on_course								
5/24	14.5	0	1	6	3	0	0.444444444	0.4
26/56	41	0	3	6	1	0	0.244897959	0.171428571
89/112	100.5	1	4	5	0	0.32	0	0.16
114/155	134.5	3	4	3	0	0.489795918	0	0.342857143
189/338	263.5	5	4	1	0	0.49382716	0	0.444444444
course_completed								
6/16	11	0	1	6	3	0	0.444444444	0.4
18/27	22.5	0	3	6	1	0	0.244897959	0.171428571
75/82	78.5	1	4	5	0	0.32	0	0.160000000
89/90	89.5	3	4	3	0	0.489795918	0	0.342857143
91/97	94	5	4	1	0	0.49382716	0	0.444444444
tugas								
0/0	0	0	1	6	4	0	0.48	0.436363636
14/20	17	0	3	6	1	0	0.244897959	0.171428571
30/52	41	1	4	5	0	0.32	0	0.160000000
53/78.5	65.75	3	4	3	0	0.489795918	0	0.342857143
81.5/81.7	81.6	5	4	1	0	0.49382716	0	0.444444444
uts								
0/0	0	0	3	6	4	0	0.48	0.369230769
0/32	16	0	3	6	1	0	0.244897959	0.171428571
64/76	70	1	4	5	0	0.32	0	0.160000000
78/84	81	3	4	3	0	0.489795918	0	0.342857143
86/86	86	6	4	2	0	0.48	0	0.4
quiz								
0/0	0	0	4	6	4	0	0.48	0.342857143
0/0	0	0	4	6	4	0	0.48	0.342857143
66/70	68	1	4	5	0	0.32	0	0.160000000
70/75	72.5	3	4	3	0	0.489795918	0	0.342857143
84/84	84	4	4	2	0	0.5	0	0.4

Hasil keseluruhan perhitungan total *gini impurity* atribut *time_spent_on_course*, *course_completed*, tugas, uts, dan quiz terhadap kelas target simbol untuk menentukan *root node* akan dipaparkan pada Tabel III.

Tabel III: Hasil Perhitungan Total *Gini Impurity*

time_spent_on_course	***course_completed***	**tugas**	**uts**	**quiz**
0.16	0.16	0.16	0.16	0.16
< 100.5	< 78.5	< 41	< 70	< 68

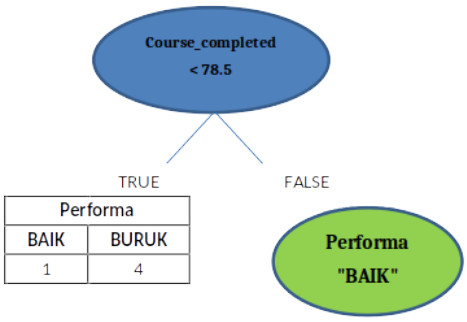
Dari hasil yang diperoleh akan diambil total *gini impurity* paling rendah yang akan menjadi *root node*. Hasil perhitungan pada Tabel III menunjukkan bahwa 4 atribut sama-sama memiliki nilai paling rendah. Pada kasus seperti ini akan dipilih salah satu atribut secara acak dari ke-4 atribut yang memiliki nilai *gini impurity* yang sama.



Gambar 2: Hasil Penentuan *Root Node*

Penentuan *root node* akan yang dipilih pada kasus ini adalah *course_completed* dengan nilai $<$ 78.5. Hasil penentuan *root node* terlihat pada Gambar 2.

Langkah selanjutnya adalah menentukan calon cabang kiri dan kanan pohon keputusan CART. Penentuan calon cabang dilakukan dengan melihat distribusi nilai *TRUE* dan *FALSE* yang melalui ketentuan akar *course_completed* terhadap kelas target. Dapat dilihat pada Gambar 2 distribusi nilai *FALSE* pada menunjukkan bahwa calon cabang kanan tidak perlu melakukan perhitungan kembali. Calon cabang kanan memuat distribusi kelas target BAIK sebanyak 5 dan kelas target BURUK sebanyak 0. Sedangkan calon cabang kiri memuat distribusi kelas target campuran yaitu BAIK sebanyak 1 dan BURUK sebanyak 4. Sehingga calon cabang kanan dapat langsung ditentukan sebagai *node* akhir seperti yang terlihat pada Gambar 3.



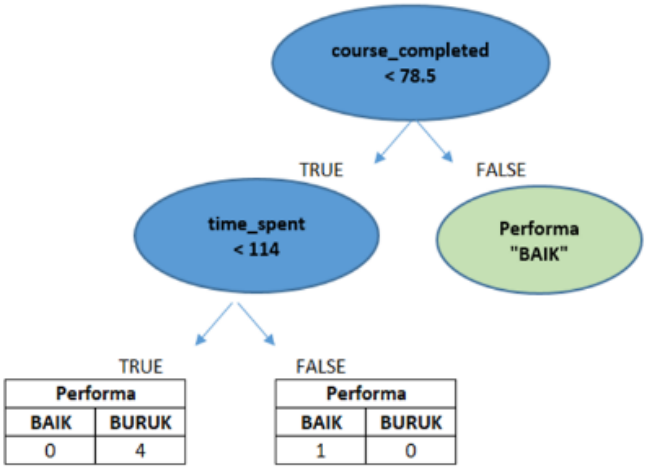
Gambar 3: Hasil Penentuan Cabang Kanan

Untuk pencarian calon cabang kiri, perhitungan *gini impurity* akan terus dilakukan secara rekursif seperti langkah-langkah sebelumnya dalam penentuan *root node*. Perhitungan akan terus dilakukan hingga menemukan atribut dengan nilai *gini impurity* paling rendah hingga akhir pohon keputusan. Tabel IV menunjukan perhitungan *gini impurity* pada pencarian calon cabang kiri.

Tabel IV: Hasil Perhitungan *Gini Impurity* Untuk Cabang Kiri

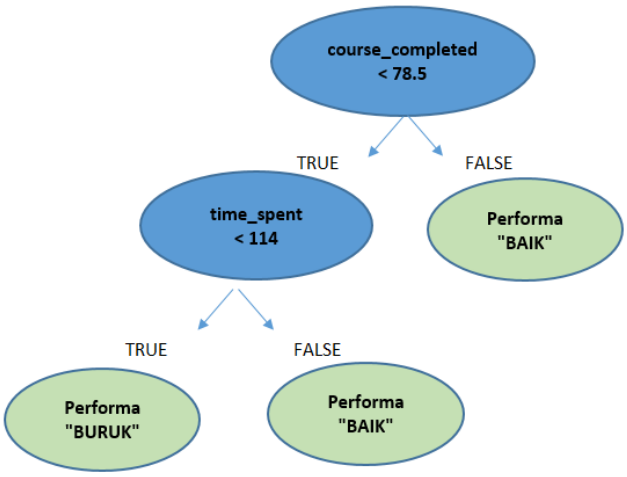
time_spent_on_course	**tugas**	**uts**	**quiz**
0	0.2	0.2	0.177777778
$<$ 114	$<$ 15	$<$ 16	$<$ 0

Pada Tabel IV, perbandingan terhadap hasil perhitungan *gini impurity* atribut *time_spent_on_course*, tugas, uts, dan quiz menunjukkan bahwa atribut *time_spent_on_course* memiliki nilai *gini impurity* paling rendah. Sehingga atribut quiz akan menempati calon cabang kiri seperti yang terlihat pada Gambar 4.



Gambar 4: Hasil Penentuan Cabang Kirin

Pada Gambar 4, dapat dilihat bahwa hasil penentuan cabang kiri menghasilkan distribusi ***time_spent_on_course*** dan quiz terhadap kelas target BAIK dan BURUK tidak memiliki distribusi kelas target campuran. Pecahan calon cabang kiri dan kanan pada node quiz masing-masing menghasilkan distribusi kelas BAIK sebanyak 0 dan BURUK sebanyak 4, serta distribusi kelas BAIK sebanyak 1 dan BURUK sebanyak 0. Sehingga calon cabang kanan dan kiri pada *node quiz* dapat langsung ditentukan sebagai *node* akhir seperti yang terlihat pada Gambar 5.



Gambar 5: Hasil Penentuan Pohon Keputusan CART

Penentuan pohon keputusan CART telah selesai dilakukan dengan hasil atribut *course_completed* sebagai *node* akar, atribut *time_spent_on_course* sebagai cabang kiri, sehingga terbentuklah klasifikasi dengan pohon keputusan CART seperti yang terlihat pada Gambar 5. Langkah terakhir pada perhitungan algoritma *Random Forest Classifier* adalah dengan melakukan pemungutan suara terbanyak atau *majority voting* terhadap pohon keputusan CART yang dibuat. Perhitungan algoritma *Random Forest Classifier* menggunakan aturan pohon keputusan CART akan terus dilakukan hingga jumlah n pohon yang diinginkan. Jumlah prediksi klasifikasi terbanyak yang dihasilkan oleh pohon keputusan CART akan menjadi hasil klasifikasi algoritma *Random Forest Classifier*.

III. HASIL

A. Pengumpulan Data

Dataset yang diperoleh dari *platform OpenLearning* dan BAA UMKT adalah sebanyak 2663 record data. Hasil perolehan dataset kemudian akan melalui proses seleksi, integrasi, transformasi, reduksi dan pembersihan data. Keterangan atribut dataset yang diperoleh dari *platform OpenLearning* dan BAA UMKT disajikan pada Tabel V dan Tabel VI.

Tabel V: Keterangan Atribut Data *OpenLearning*

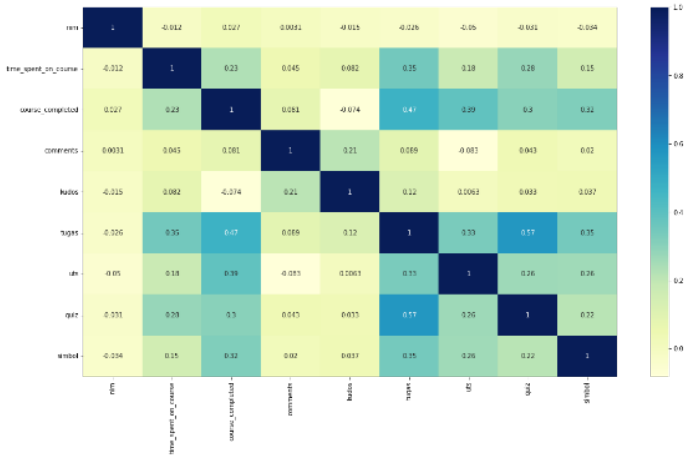
No.	Atribut	Keterangan
1	*Profile name*	Id mahasiswa pada sistem OpenLearning
2	*Learner name*	Nama mahasiswa
3	*Learner email*	Email mahasiswa
4	*Enrolment ID*	Id pendaftaran OpenLearning
5	*Institution Membership ID*	Id anggota institusi
6	*Enrolment date*	Tanggal daftar
7	*Completion date*	Tanggal menyelesaikan mata kuliah
8	*Time spent on course*	Lama waktu mahasiswa berada di mata kuliah
9	*Progress*	Persentase kemajuan mahasiswa
10	*% Course completed*	Persentase kemajuan mahasiswa menyelesaikan mata kuliah
11	*Certificate ID*	Id sertifikat
12	*Comments*	Banyaknya komentar mahasiswa selama perkuliahan
13	*Kudos*	Penghargaan
14	*Enrolment cost*	Biaya pendaftaran mata kuliah
15	*Tugas*	Nilai rata-rata tugas 1 – 10
16	*Quiz*	Nilai quiz
17	*UTS*	Nilai Ujian Tengah Semester

Tabel VI: Keterangan Attribut Data BAA

No.	Uraian	Keterangan
1	NIM	Nomor induk mahasiswa
2	Nama Mahasiswa	Nama mahasiswa
3	Nilai Akhir	Nilai akhir mahasiswa
4	Bobot	Bobot nilai berdasarkan standar penilaian
5	Simbol	Skala peniliaian

B. Seleksi dan Integrasi Data

Proses seleksi dan integrasi tahap pertama adalah dengan menyeleksi atribut yang tidak diperlukan menggunakan metode CFS. CFS merupakan teknik yang melakukan pertimbangan terhadap fitur individual yang berguna pada dugaan kelas label menggunakan tingkat hubungan atau korelasi diantara fitur lainnya [12]. Atribut yang dihapus pada data BAA UMKT adalah Nama dan Bobot. Sementara atribut yang dihapus pada data *OpenLearning* adalah *Profile name*, *Learner name*, *Learner email*, *Enrolment ID*, *Institution Membership ID*, *Enrolment date*, *Completion date*, *Progress*, dan *Certificate ID*. Proses selanjutnya adalah melihat korelasi antara atibut (*nim*, *time spent on course*, *course completed*, *comments*, *kudos*, program studi, tugas, uts, dan *quiz*) terhadap kelas target / target class (simbol).



Gambar 6: Hasil Pencarian Korelasi Atribut Menggunakan CFS

Hasil pencarian korelasi antara atribut terhadap kelas target dapat dilihat pada Gambar 6. Pencarian korelasi antara atribut terhadap kelas target dilakukan dengan menggunakan python. Hasil pencarian korelasi menunjukkan bahwa atribut nim, *comments*, dan *kudos* memiliki nilai korelasi yang sangat rendah terhadap kelas target. Atribut yang dipilih adalah *time spent on course* = 0.15, *course completed* = 0.32, tugas =0.35, uts = 0.26, dan *quiz* = 0.22. Tabel VII menunjukan hasil seleksi atribut berdasarkan nilai korelasi yang diperoleh dengan metode CFS.

Tabel VII: Hasil Seleksi Atribut Berdasarkan Nilai Korelasi

No	*time spent on course*	*course completed*	tugas	UTS	*quiz*	Simbol
1	4 Hrs 56 Mins	69.55	54.1	0	0	AB
2	5 Hrs 45 Mins	80.91	90	0	57	AB
3	17 Hrs 47 Mins	81.36	79.6	0	57	AB
...	...	...	...	...	...	...
2663	4 Hrs 42 Mins	95	72.4	84	0	A

C. Transformasi Data

Pada atribut *time spent on course*, data lamanya mahasiswa berada pada *course* atau mata kuliah memuat dua jenis tipe data dalam *record*-nya, yaitu *integer* dan *string*. Nilai atribut *time spent on course* akan diubah menjadi akumulasi dari lamanya mahasiswa berada di *course* menjadi hitungan menit. Sementara pada atribut simbol akan dirubah dengan ketentuan apabila nilai mahasiswa sama dengan A, AB, atau B maka kelas diubah menjadi BAIK, sementara apabila nilai mahasiswa adalah BC, C, D, atau E maka kelas diubah menjadi BURUK. Gambar 7 dan Gambar 8 menunjukkan hasil transformasi pada atribut *time spent on course* dan simbol.

Time spent on course
4 Hrs 56 Mins
5 Hrs 45 Mins
17 Hrs 47 Mins
⋮
4 Hrs 42 Mins

Time spent on course
296
345
1067
⋮
282

Gambar 7: Hasil Transformasi Atribut *Time Spent On Course*

Simbol
AB
AB
AB
⋮
A

Simbol
BAIK
BAIK
BAIK
⋮
BAIK

Gambar 8: Hasil Transformasi Atribut Simbol

D. Transformasi Data

Tahapan reduksi data terhadap dataset yang telah melalui proses seleksi, integrasi, dan transformasi dilakukan untuk menghindari dataset dengan distribusi kelas target yang tidak seimbang (data *imbalance*). Karena diketahui bahwa setelah dataset yang telah melalui proses seleksi, integrasi dan transformasi memliki kelas target yang tidak seimbang. Kelas target yang tidak seimbang pada dataset dapat mempengaruhi proses *training* data. Dimana dataset yang memiliki kelas target yang paling banyak (mayoritas) akan menyebabkan pemodelan menjadi bias terhadap kelas target mayoritas [13].

Pada tahapan reduksi data ini, juga dilakukan tahapan pembersihan data terhadap data yang tidak konsisten. Ketidakkonsistenan data dapat dilihat pada data mahasiswa memiliki nilai tugas, uts, dan quiz yang buruk namun diklasifikasikan BAIK. Juga pada kasus sebaliknya, mahasiswa memiliki nilai tugas, uts, dan quiz yang baik namun diklasifikasikan BURUK. Maka pada tahapan reduksi ini, juga dilakukan proses pembersihan terhadap data yang tidak konsisten. Tabel VIII menunjukkan hasil akhir tahapan persiapan data.

Tabel VIII: Hasil Seleksi, Integrasi, Transformasi, Pembersihan, dan Reduksi Data

No	time spent on course	*course completed*	tugas	UTS	*quiz*	Simbol
1	0	0	0	0	0	BURUK
2	0	0	0	0	0	BURUK
3	0	0	0	0	0	BURUK
...	...	...	...	...	...	...
178	304	81	80.2	70	60	BAIK

E. PemodelanPemodelan Random Forest Classifier

Dataset yang akan digunakan pada pemodelan adalah sebanyak 178 data. Bahasa pemrograman python digunakan untuk mengetahui tingkat akurasi algoritma *Random Forest Classifier*. Setelah dataset di-*import*, selanjutnya dilakukan pembuatan *data frame* pada variabel data. Pembuatan *data frame* ditujukan untuk membuat sebuah struktur data, agar nantinya mudah untuk digunakan pada tahap pemodelan.

```
#Import library pandas
import pandas as pd
# Melakukan import dataset
ds = pd.read_csv('dataset.csvs')
```

(a)*Import dataset*

```
data=pd.DataFrame({
'time_spent_on_course':ds['time_spent_on_course'],
'course_completed':ds['course_completed'],
'tugas':ds['tugas'],
'uts':ds['uts'],
'quiz':ds['quiz'],
'simbol':ds['simbol']})
```

(b)Membuat *data frame*

Proses selanjutnya adalah membuat pemodelan *Random Forest Classifier*. Modul algoritma *Random Forest Classifier* dimuat kedalam variabel *clf* yang kemudian diikuti dengan *fitting* data *training* kedalam pemodelan dengan menggunakan fungsi *fit()*. Deklarasi variabel *y_pred* dilakukan untuk memuat *data testing* yang telah di bagi sebelumnya dengan menggunakan fungsi *predict()*.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

(c)*Import modul*

```
x=data['time_spent_on_course','course_completed','tugas','uts','quiz',]
y=data['simbol']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
clf=RandomForestClassifier(n_estimators=42)
clf.fit(X_train,y_train)
y_pred=clf.predict(X_test)
print(\Akurasi: ",metrics.accuracy_score(y_test,y_pred))
```

(d)Pengujian Pemodelan dengan 80% *data training* dan 20% *data testing*

IV. PEMBAHASAN

A. Evaluasi Pemodelan

Pada tahapan evaluasi pemodelan, metode evaluasi akan menggunakan confusion matrix. Menurut Suyanto [14], metode evaluasi menggunakan matriks konfusi atau yang biasa disebut dengan *confusion matrix* merupakan perhitungan untuk mengukur seberapa baik proses klasifikasi. Dalam mengukur kategori klasifikasi sebuah pemodelan, terdapat beberapa katergori klasifikasi, yaitu [15]:

a) Akurasi dengan nilai 0,90 - 1,00 = *excellent classification*
b) Akurasi dengan nilai 0,80 - 0,90 = *good classification*
c) Akurasi dengan nilai 0,70 - 0,80 = *fair classification*
d) Akurasi dengan nilai 0,60 - 0,70 = *poor classification*
e) Akurasi dengan nilai 0,50 - 0,60 = *failure classification*

Tabel IX: Hasil *Confusion Matrix* Pemodelan *Random Forest Classifier* Menggunakan CFS

		True Value	
		BAIK	BURUK
Prediction	BAIK	21	0
	BURUK	1	14

$$\begin{aligned} accuracy &= \frac{21+14}{21+14+0+1} \\ &= \frac{35}{36} = 0.97222222222 \\ &= 97,22\% \end{aligned}$$

Tabel IX menunjukkan hasil *confusion matrix* terhadap 36 data testing pada yang telah melalui proses seleksi fitur menggunakan metode CFS pada pemodelan *Random Forest Classifier* menunjukan hasil *accuracy* = 0.97, *precision* = 0.98, dan *recall* = 0.97. Maka pada pemodelan *Random Forest Classifier* menggunakan seleksi fitur CFS diperoleh hasil akurasi sebesar 97,22%, dengan hasil prediksi benar sebanyak 35 data dari 36 jumlah keseluruhan *data testing*. Dari hasil yang diperoleh, menunjukan bahwa pemodelan dinyatakan masuk dalam kategori *excellent classification*.

Tabel X: Hasil *Confusion Matrix* Pemodelan *Random Forest Classifier* Tanpa CFS

		True Value	
		BAIK	BURUK
Prediction	BAIK	15	3
	BURUK	0	18

$$\begin{aligned} accuracy &= \frac{15+18}{18+17+3+0} \\ &= \frac{33}{36} = 0.916666666 \\ &= 91,66\% \end{aligned}$$

Adapun sebagai pembanding pemodelan *Random Forest Classifier* pada Tabel X menunjukkan *dataset* yang tidak melalui proses seleksi fitur menggunakan metode CFS menghasilkan *accuracy* = 0.916, *precision* = 0.93, dan *recall* = 0.92 dengan hasil prediksi benar sebanyak 33 data dari 36 jumlah keseluruhan *data testing*. Dengan perolehan nilai akurasi sebesar 91.66%, pemodelan *Random Forest Classifier* tanpa menggunakan seleksi fitur CFS masih masuk dalam kategori *excellent classification*, namun dengan penambahan metode seleksi fitur CFS, akurasi pemodelan dapat ditingkatkan sebesar 5.56% menjadi 97,22%.

V. Simpulan

Penerapan metode seleksi fitur CFS pada pemodelan *Random Forest Classifier* dilakukan untuk mengetahui atribut-atribut mana saja yang memiliki korelasi paling tinggi terhadap kelas target prediksi. Hasil seleksi fitur menggunakan CFS yang dilakukan terhadap data performa akademik mahasiswa dalam pembelajaran daring menunjukkan bahwa nilai korelasi atribut tertinggi ada pada atribut *time spent on course* = 0.15. *course completed* = 0.32, tugas =0.35, uts = 0.26, dan *quiz* = 0.22. Sedangkan atribut nim = 0.034, *comments* = 0.02, *kudos* = 0.037 memiliki nilai korelasi yang sangat rendah terhadap kelas target. Berdasarkan hasil *confusion matrix* pemodelan algoritma *Random Forest* dengan optimasi CFS yang dilakukan, diketahui bahwa penggunaan metode seleksi fitur CFS terbukti dapat meningkatkan akurasi prediksi pemodelan *Random Forest Classifier*. Dimana hasil pemodelan *Random Forest Classifier* yang tanpa menggunakan metode seleksi fitur CFS hanya memperoleh tingkat akurasi sebesar 91,66%, sedangkan pemodelan *Random Forest Classifier* dengan menggunakan metode seleksi fitur CFS, terbukti meningkatkan akurasi pemodelan sebanyak 5.56%, menjadikan total akurasi pemodelan *Random Forest Classifier* sebesar 97,22%.

VI. Saran

Adapun saran yang dihasilkan dari penelitian ini adalah sebagai berikut:

1) Untuk meningkatkan kualitas dan akurasi pemodelan, disarankan untuk menambah atribut terhadap atribut yang telah diteliti dapat dilakukan seperti menambah nilai tugas-tugas tambahan, absensi kehadiran, serta nilai ujian akhir semester serta jumlah data yang lebih banyak.
2) Disarankan pada algoritma *Random Forest Classifier* juga dapat digunakan algoritma seleksi fitur lainnya seperti *Chi-square*, *Information Gain*, *ANOVA*, *Forward Selection*, dan lain-lain.
3) Disarankan pada pemodelan algoritma *Random Forest Classifier* dapat dilakukan percobaan kembali dengan nilai pohon atau *n_estimator* yang lebih banyak untuk menghasilkan analisis dan kualitas akurasi pemodelan yang baru.

Pustaka

[1] KPAI. 2021. Survei Pelaksanaan Pembelajaran Jarak Jauh (PJJ) dan Sistem Penilaian Jarak Jauh Berbasis Pengaduan KPAI [pdf] Komisi Perlindungan Anak Indonesia. Tersedia di: https://bankdata.kpai.go.id/files/2021/02/Paparan-Survei-PJJ-KPAI-29042020_Final-update.pdf
[2] Magdalena, I., Ridwanita, A., & Aulia, B. (2020). Evaluasi Belajar Peserta Didik. PANDAWA, 2(1), 117-127.
[3] Hermawati, F.A. (2013). Data Mining. Yogyakarta: Penerbit Andi.
[4] Xing, W., Chen, X., Stein, J., & Marcinkowski, M. (2016). Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. Computers in Human Behavior, 58, 119-129.
[5] Abubakar, Y., & Ahmad, N. B. H. (2017). Prediction of Students Performance in E-Learning Environment Using Random Forest. International Journal of Innovative Computing, 7(2).
[6] Batool, S., Rashid, J., Nisar, M. W., Kim, J., Mahmood, T., & Hussain, A. (2021). A Random Forest students' performance prediction (rfspp) model based on students' demographic features. In 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC) (pp. 1-4). IEEE
[7] Linawati, S., Nurdiani, S., Handayani, K., & Latifah, L. (2020). Prediksi Prestasi Akademik Mahasiswa Menggunakan Algoritma Random Forest Dan C4. 5. Jurnal Khatulistiwa Informatika, 8(1).
[8] Breiman, L. (2001). Random Forests. Machine learning, 45(1), 5-32. Springer.
[9] Jonathan. (2021). Implementasi Algoritma Random Forest untuk Klasifikasi Kategori Berita. UMN Knowledge Center, [e-journal] Tersedia melalui : UMN Knowledge Center https://kc.umn.ac.id/id/eprint/16610 [Diakses 25 Januari 2022]
[10] Liparas, D., Ha, Cohen-Kerner, Y., Moumtzidou, A., Vrochidis, S., & Kompatsiaris, I. (2014). News articles classification using random forests and weighted multimodal features. In Information Retrieval Facility Conference (pp. 63-75). Springer, Cham
[11] Daniya, T., Geetha, M., & Kumar, K. S. (2020). Classification and regression trees with Gini index. Advances in Mathematics: Scientific Journal, 9(10), 8237-8247.
[12] Djatna, T, & Yasuhiko M. (2008). "Pembandingan Stabilitas Algoritma Seleksi Fitur Menggunakan Transformasi Ranking Normal." Jurnal Ilmiah Ilmu Komputer, vol. 6, no. 2.
[13] Ali, H., Salleh, M. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: a review. Indonesian Journal of Electrical Engineering and Computer Science, 14(3), 1560-1571.
[14] Suyanto, S. A. N. (2018). Klasifikasi Jenis Infeksi Berdasarkan Hasil Pemeriksaan Leukosit Menggunakan K-Nearest Neighbor (KKN). Tersedia di: https://repositori.usu.ac.id/handle/123456789/11694 [Diakses 28 Januari 2018]
[15] Hariati, H., Wati, M., & Cahyono, B. (2018). Penerapan Algoritma C4. 5 pada Penentuan Penerima Program Bantuan Pemerintah Daerah di Kabupaten Kutai Kartanegara. Jurnal Rekayasa Teknologi Informasi (JURTI), 2(2), 106-114.