PENERAPAN DATA MINING UNTUK KLASIFIKASI PENJUALAN BARANG TERLARIS MENGGUNAKAN METODE *DECISION TREE C4.5*

Ni Wayan Wardani[1)] Putu Gede Surya Cipta Nugraha[2)] Eddy Hartono[3)] I Wayan Dharma Suryawan[4)] Ayu Manik Dirgayusari[5)] I Wayan Darmadi[6)] Gede Surya Mahendra[7)]
Program Studi Teknik Informatika, Fakultas Teknologi dan Informatika[1)2)3)4)6)]
Program Studi Sistem Komputer, Fakultas Teknologi dan Informatika[5)]
Institut Bisnis Dan Teknologi Indonesia[1) 2) 3) 4) 5) 6)]
Program Studi Sistem Informasi, Universitas Pendidikan Ganesha[7)]
niwayan.wardani@instiki.ac.id

ABSTRACT

This research aims to find the best accuracy from the decision tree model so that the model can perform well for bestselling sales classification and make the model usable and integrated with other systems through the Application Programming Interface (API). This analysis uses the decision tree method and the Cross-Industry Standard Process for Data Mining (CRISP-DM) as the research process flow. The research results that have been obtained in the early stages of the model can produce an accuracy of 90.85% in RapidMiner modeling. In comparison, in python, the resulting accuracy is 92.83%, but when the parameter tuning process is carried out the highest accuracy produced reaches 95.68% on RapidMiner while in modeling using python the quality of accuracy is 95.09% and based on the deployment process, the prediction function of the model can be accessed properly through the Application Programming Interface (API).

Keywords: *Data Mining, Decision Tree C4.5, Classification*

ABSTRAK

Penelitian ini bertujuan untuk mencari akurasi terbaik dari model *Decision Tree* sehingga model dapat menghasilkan performa yang baik untuk tujuan klasifikasi penjualan terlaris dan juga membuat model dapat digunakan dan terintegrasi pada sistem lain melalui *Application Programming Interface* (API). Analisis ini menggunakan metode *decision tree*, dan *Cross-Industry Standard Process for Data Mining* (CRISP-DM) sebagai alur proses penelitian. Hasil penelitian yang telah didapatkan pada tahap awal model dilatih dapat menghasilkan akurasi 90.85% pada pemodelan RapidMiner, sedangkan pada python akurasi yang dihasilkan 92.83%, akan tetapi pada saat proses tuning parameter dilakukan akurasi paling tertinggi yang dihasilkan mencapai 95.68% pada RapidMiner sedangkan pada pemodelan menggunakan python menghasilkan akurasi sebesar 95.09%, dan berdasarkan proses *deployment*, fungsi prediksi model dapat dengan baik diakses melalui *Application Programming Interface* (API).

Kata Kunci: *Data Mining*, *Decision Tree* C4.5, Klasifikasi

PENDAHULUAN

Usaha Ritel Usaha yang bergerak di bidang ritel merupakan jenis usaha yang melibatkan penjualan barang kepada konsumen dalam jumlah satuan. Pada zaman modern seperti sekarang banyak usaha ritel yang sudah menggunakan teknologi, hal ini dikarenakan sistem informasi dapat memberikan kemudahan dan kecepatan dalam memberikan pelayanan kepada para konsumen. Biasanya sistem informasi yang sering diimplementasikan pada usaha ritel yaitu sistem informasi penjualan dan sistem *inventory*.

Usaha ritel biasanya akan terus memastikan ketersediaan dari produk-produk yang dijual. Biasanya acuan informasi untuk menambah stok barang yang sudah habis didapatkan melalui sistem informasi *inventory*.

Pada saat sistem informasi *inventory* memberikan pemberitahuan bahwa stok suatu barang akan habis, dan bagian penggudangan barang akan segera melakukan *restock* produk. Hal ini menimbulkan suatu masalah jika usaha ritel melakukan *restock* produk begitu saja tanpa mengetahui apakah produk tersebut termasuk pada produk yang laris atau tidak laris. Jika usaha ritel melakukan *restock* produk yang kurang diminati (tidak laris) maka kemungkinan perputaran modal usaha akan mengendap terlalu lama sehingga keuntungan yang didapatkanpun menjadi tidak maksimal. Tetapi dilain sisi jika usaha ritel dapat mengetahui produk mana yang cepat terjual (laris), dan memfokuskan *restock* produk lebih banyak pada barang-barang tersebut, maka perputaran modal menjadi lebih cepat dan usaha ritel mendapatkan keuntungan lebih cepat. Cara untuk mengetahui apakah suatu produk laris atau tidak yaitu dengan cara menganalisis data penjualan yang sudah terjadi pada beberapa tahun yang lalu. Analisis ini berguna untuk mengklasifikasi barang laris dan tidak laris. Proses analisis suatu data yang sangat besar dapat menimbulkan masalah jika dilakukan dengan cara manual. Maka dari itu diperlukan implementasi *data mining* yang dapat mengekstrak suatu informasi baru pada kumpulan data yang berjumlah sangat banyak [1].

UD. Mawar Sari merupakan usaha ritel yang menjual barang kebutuhan masyarakat sehari-hari yang telah berdiri sejak tahun 1976 dan berlokasi di Jalan Ahmad Yani No.5, Subagan, Amlapura, Karangasem. UD. Mawar Sari telah menggunakan sistem informasi penjualan dan *inventory* untuk mendukung proses bisnis yang ada. Sejak tahun 2019 sampai 2020, sistem informasi penjualan di UD. Mawar Sari menyimpan data penjualan sebanyak 1.104.897 data transaksi penjualan dan 12.177 data produk yang dijual. Sumber informasi yang digunakan UD.Mawar Sari saat ini untuk melakukan manajemen stok yaitu dilakukan secara konvensional dan pada saat melakukan order akan dilakukan pengecekan kembali ke sistem *inventory*, rak barang, dan stok digudang. Dilihat dari cara manajemen stok yang dilakukan saat ini maka dapat disimpulkan bahwa modal usaha dari UD.Mawar Sari digunakan secara merata untuk melakukan *restock* produk yang akan habis sesuai dengan informasi yang didapatkan pada rak barang, gudang barang dan sistem *inventory*. UD.Mawar Sari dapat memanfaatkan data penjualan dan data barang yang dijual untuk mendapatkan pengetahuan tentang klasifikasi produk yang laris dan tidak laris sehingga manajemen stok bisa dapat difokuskan pada barang yang laris. Metode yang digunakan dalam penelitian ini adalah metode *decision tree* C4.5 metode ini dipilih karena, berdasarkan penelitian yang berjudul "Implementasi Algoritma *Decision Tree* Untuk Klasifikasi" oleh Nasrullah, 2018 dimana model pohon keputusan yang dibuat menghasilkan akurasi sebesar 90% dengan nilai *Area Under the Curve* (AUC) sebesar 0.709 dan nilai tersebut masuk kedalam kategori klasifikasi yang cukup baik[2]. Penelitian yang berjudul "Analisis Penjualan Barang Laris dan Kurang Laris Terhadap Percetakan AWFA Digital Printing Menggunakan Metode *Decision Tree* Dengan Optimasi Algoritma Genetika" yang dilakukan oleh *Suherman dan Muzaky, 2019* menghasilkan akurasi paling optimal yaitu 85.15%, tetapi pada saat metode *decision tree* C4.5 dioptimasi dengan menggunakan metode algoritma genetika, akurasi yang dicapai meningkat hingga 93.09% [3]. Berdasarkan penelitian diatas yang telah menggunakan metode *decision tree* C4.5 di berbagai *case*, menunjukkan metode tersebut menghasilkan akurasi yang cukup baik untuk tujuan klasifikasi.

METODE PENELITIAN

Metode yang digunakan pada penelitian ini yaitu CRISP-DM (*Cross Industry Standard Process for Data MIning*) sebagai alur proses utama untuk mendukung proses pembuatan model *data mining*. Metode CRISP-DM merupakan suatu standar proses yang sering digunakan untuk memecah permasalahan pada suatu bisnis degan menggunakan *data mining* [4], [5]. Metode CRISP-DM terdiri atas enam tahapan yaitu:

Gambar 1 Alur CRISP-DM
[Sumber: Wardani, 2020]

1) *Business Understanding*

Pada *business understanding* merupakan tahap pertama untuk memahami tujuan dan kebutuhan dari sudut pandang bisnis dan kemudian menerjemahkan pengetahuan ini kedalam pendefinisian masalah dalam *data mining* dan dilanjutkan menentukan rencana dan strategi untuk mencapai tujuan tersebut.

2) *Data Understanding*

Setelah mengetahui dan memahami permasalahan pada proses *business understanding* maka tahap selanjutnya yaitu mengumpulkan data yang dilanjutkan dengan proses pemahaman terhadap data yang didapatkan.

3) *Data Preparation*

Tahap ini meliputi semua keinginan untuk membangun data yang akan diproses pada tahap modeling dari data mentah. Pada tahap ini mencakup pemilihan *table, record*, atribut-atribut data, termasuk proses *cleaning* dan *transformation data*, sehingga dapat dijadikan masukan dalam tahap pemodelan (*modeling*).

4) *Modeling*

Proses modeling merupakan langkah setelah proses *data preparation* dilakukan dan telah memastikan bahwa data cukup bersih dan siap digunakan untuk proses pelatihan model. Pada proses modelling akan digunakan bantuan *tools* seperti RapidMiner dan python. Adapun metode yang digunakan yaitu *decision tree* dengan algoritma C4.5.

5) *Evaluation*

Setelah model dilatih diperlukan suatu proses untuk mengevaluasi performa dari model yang telah dilatih. Adapun metode yang digunakan untuk evaluasi yaitu dimulai evaluasi sederhana menggunakan *confussion matrix*, mengunakan metode *cross validation* untuk menghilangkan faktor beruntung dan ketidak beruntungan dari model, dan terakhir menggunakan metode *tuning parameter* untuk mencari nilai akurasi terbaik dari model yang dilatih dengan menggunakan dan membandingkan beberapa parameter yang digunakan pada proese modeling.

6) *Deployment*

Pada tahap deployment merupakan proses untuk menentukan bagaimana cara agar model dapat diakses dengan mudah dan digunakan oleh pihak UD.Mawar Sari. Model harus dapat dimengerti dan digunakan oleh pihak yang tidak terlalu mengerti hal teknis.

HASIL DAN PEMBAHASAN

Untuk hasil penelitian penulis akan membahas tentang hasil dari setiap proses pada CRISP-DM. dimulai dari:

1) *Business Understanding*

UD. Mawar Sari merupakan sebuah usaha ritel yang menjual barang-barang kebutuhan masyarakat sekitar. UD.Mawar Sari memiliki masalah dalam alokasi modal dalam memilih restok produk laris untuk menghindari pengendapan modal yang terlalu lama pada produk-produk yang tidak laris. Jadi UD.Mawar Sari ingin mengklasifikasi dan mengetahui pola dari produk yang laris dan tidak laris, karena dengan mengetahui pola tersebut, pihak UD.Mawar Sari dapat mengklasifikasikan produk-produk yang termasuk klasifikasi laris, sehingga pihak manajemen UD.Mawar Sari dapat memfokuskan modal untuk melakukan *restock* produk yang diklasifikasikan sebagai laris sehingga perputaran modal menjadi lebih lancar dan terhindar dari pengendapan modal pada produk-produk yang sulit dijual.

Dari permasalahan diatas diketahui bahwa UD.Mawar Sari ingin agar penggunaan modal

dalam manajemen stok produk dapat difokuskan pada produk terlaris dengan tujuan untuk mengurangi resiko modal mengendap pada produk-produk yang sulit untuk terjual (tidak laris) dengan begitu tujuan dari penelitian ini yaitu mengklasifikasikan dan menemukan pola dari produk laris dan tidak laris.

2) *Data Understanding*

Untuk menyelesaikan masalah pada business understanding yang terkait dengan produk terlaris, data yang digunakan untuk penelitian ini yaitu data penjualan produk yang didapatkan dari sistem penjualan yang digunakan oleh UD.Mawar Sari. Data yang didapatkan yaitu data penjualan 2019 – 2020 (Januari - Agustus), dengan 25 atribut, dan memiliki total 1.104.897 *record*. Adapun atribut yang terdapat pada data penjualan diantaranya yaitu: TGL, JAM, NO_EJ, BARA, NAMA, HARGA, QTY, DISC1, DISC2, DISCRP, AVER, AVER_KONSI, BRUTO, NETTO, BKP, ID, BO, TGLC, FEE, CASHBACK, STN, STNSTD, ISI, INV, KONSI.

3) *Data Preparation*

- *Data cleaning*

Data *cleaning* merupakan suatu metode yang bertujuan untuk memastikan kebersihan data agar terhindar dari value yang hilang (*missing value)* atau kesalahan dalam proses input (*noise value*)

Untuk proses penemuan *missing value* penulis menggunakan formula excel (counta) untuk menghitung dan memastikan jumlah *value* setiap atribut sesuai dengan jumlah *record* yang dimiliki pada dataset.

Untuk proses penemuan *noise value* dibutuhkan pemahaman dari setiap atribut pada data, misalnya pada data penjualan atribut *quantity* (QTY) merupakan atribut yang memiliki makna suatu produk berhasil terjual (barang keluar) jadi akan sangat tidak memungkinkan jika *value* pada atribut *quantity* kurang atau sama dengan 0. Maka untuk mencari *noise value* yaitu memahami makna atribut dan sesuaikan dengan *value* lainnya.

Solusi penulis pada saat mendapatkan *missing/noise value* pada beberapa atribut yaitu dikarenakan jumlah data cukup banyak dan jumlah *missing/noise value* sedikit maka penulis menghapus baris data yang hilang atau salah tersebut.

- *Data Transformation*

Pada tahap ini penulis akan menyeleksi atribut yang akan digunakan, dimana dari 25 atribut pada data penjualan, penulis akan memilih 3 atribut utama yang akan memiliki keterkaitan terhadap penjualan suatu produk. Adapun atribut yang dipilih yaitu TGL, NAMA, QTY.

Untuk atribut TGL dan QTY penulis akan gunakan sebagai atribut utama untuk membentuk atribut baru, dimana penulis akan mengelompokan kembali jumlah penjualan berdasarkan nama produk pada jangka waktu setiap empat bulan sekali (caturwulan). Jadi pada dataset akan membentuk data C1 sampai C-n, dimana n merupakan jumlah caturwulan berdasarkan jangka waktu dari data yang didapatkan, pada penelitian ini penulis menggunakan data 2019 – 2020 (Agustus).

Untuk atribut NAMA penulis gunakan untuk membentuk atribut kategori dimana atribut tersebut penulis isi secara manual berdasarkan informasi yang penulis dapat dari *google*. Atribut kategori produk diperlukan karena setiap jenis produk memiliki kriteria laris yang berbeda-beda.

Tahap terakhir yaitu membentuk atribut target pembelajaran dari model. Atribut target sangat diperlukan pada dataset penjualan, hal ini dikarenakan metode *decision tree* C4.5 merupakan metode *supervised learning*, dimana metode yang memerlukan target untuk proses pembelajarannya. Untuk acuan laris dan tidak laris, penulis melakukan diskusi dengan pihak UD.Mawar Sari dan menyepakati bahwa untuk menentukan suatu produk laris penulis menggunakan persamaan sebagai berikut:

$$\text{target} = \text{if}(x > y):'\text{Laris}';'\text{Tidak Laris}' \quad (1)$$

Dimana x merupakan nilai rata-rata penjualan produk x pada kategori y, dan y merupakan total rata-rata penjualan pada kategori y.

Contoh penentuan target, jika diketahui untuk seluruh produk pada kategori rokok total

penjualan rata-rata pada sepanjang data 128.5. dan jika produk rokok gudang gula penjualan disepanjang data mencapai 325 maka produk gudang gula memiliki target laris, sedangkan jika kurang atau sama dengan dari 128.5 maka produk akan diidentifikasi sebagai tidak laris.

	NAMA	KATEGORI	C1	C2	C3	C4	C5	TARGET
0	SAMPOERNA A MILD 16 BKS	Rokok	1301	1455	1405	1714	1332	Laris
1	SAMPOERNA U MILD 12	Rokok	69	177	200	105	52	Tidak Laris
2	U MILD 16 BKS	Rokok	101	82	71	194	163	Laris
3	IN MILD NEW	Rokok	982	941	1074	1233	1349	Laris
4	GG FILTER MRH 12 BKS	Rokok	268	311	292	307	293	Laris
...	...	...	...	...	...	...	...	...
3152	PUYER BINTANG 7 NO 16	Kesehatan	0	0	12	72	30	Laris
3153	TOLAK ANGIN ANAK+MADU 10ML	Kesehatan	0	0	8	38	45	Tidak Laris
3154	BETADINE SOLUTION 30ML	Kesehatan	0	0	1	1	0	Tidak Laris
3155	BIOGESIC TABLET 4`S	Kesehatan	0	0	2	13	1	Tidak Laris
3156	MASKER SPON SCUBA	Kesehatan	0	0	0	172	0	Laris

3157 rows × 8 columns

Gambar 2 Hasil Data Preparation

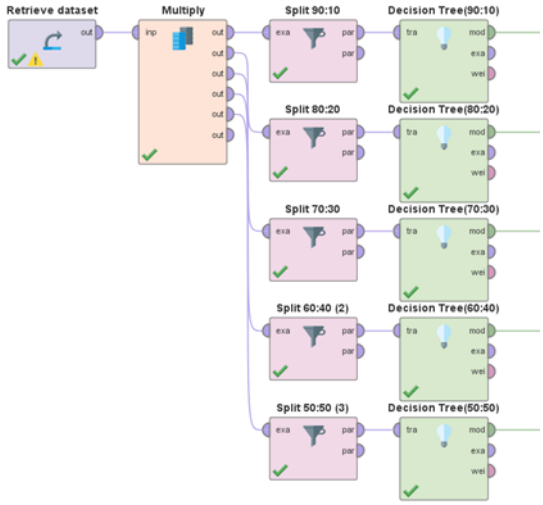
Berdasarkan gambar 2, data tersebut merupakan hasil dari proses *data preparation*, dimana dataset sudah ditransform untuk menyesuaikan pola penjualan setiap produk yang tersedia.

4) *Modeling*

Pada proses pemodelan penulis menggunakan algoritma *decision tree C4.5* pada tools RapidMiner dimana merupakan sebuah alat untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi [6]. dan python. Untuk penentuan data latih dan data uji penulis menggunakan fungsi *split data* dengan menyesuaikan rasio untuk memisah antara data latih dan data uji dari total dataset yang digunakan.

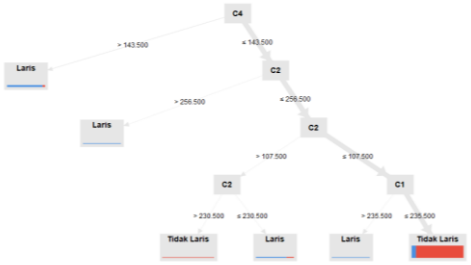
- Pemodelan Dengan RapidMiner

Proses penentuan data latih dan data uji akan dilakukan melalui operator *split data* dimana data akan dipisah menjadi data latih dan data uji dengan rasio mulai 90:10, 80:20, 70:30, 60:40, dan 50:50. Proses akan dilakukan sekaligus menggunakan operator *multiply* untuk menduplikat proses menjadi 5 bagian.

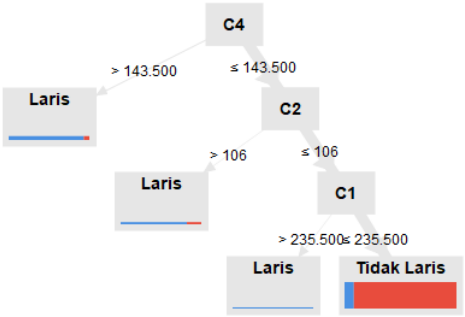


Gambar 3 Pemodelan *Split Data*

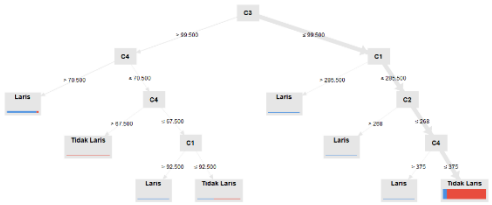
Pada gambar 3, merupakan pemodelan yang dilakukan pada *tool* rapidMiner, dimana data latih dibagi dalam 5 proses dengan bobot data latih dan data uji yang berbeda-beda. Dari pemodelan diatas akan menghasilkan pohon keputusan sebagai berikut.



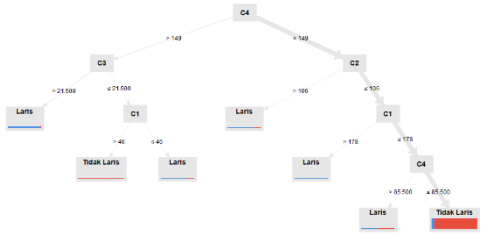
Gambar 4 *Tree* 90% Data Latih



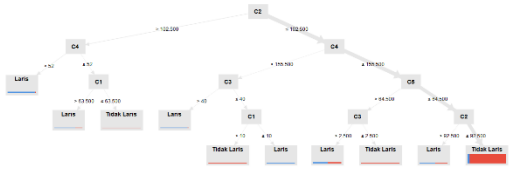
Gambar 5 *Tree* 80% Data Latih



Gambar 6 *Tree* 70% Data Latih



Gambar 7 *Tree* 60% Data Latih



Gambar 8 *Tree* 50% Data Latih

Pada gambar 4 sampai gambar 8, merupakan hasil dari pemodelan menggunakan *decision tree* parameter criterion(gain_ratio) dan maximal_depth(5).

- Pemodelan Dengan Python

Proses pemodelan menggunakan bahasa pemprograman python akan menggunakan library scikit-learn dan pandas sebagai modul utama untuk melakukan proses pelatihan model.

Gambar 9 *Tree* Pemodelan Dengan Python

Pada gambar 9, merupakan hasil visualisasi model dengan parameter bawaan dari scikit-learn, yaitu criterion(gini) dan maximal_depth(5)

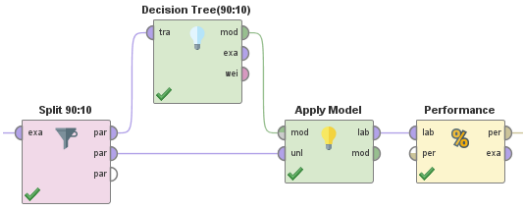
5) *Evaluation*

Proses akan dilakukan pada tiga proses yang berbeda, dimana proses pertama mengevaluasi model yang telah dilatih pada saat proses modeling, dilanjutkan untuk memvaludasi performa menggunakan metode *cross validation* untuk memastikan akurasi yang dihasilkan terhindar dari bias, dan proses evaluasi terakhir yaitu *tuning parameter* pada metode *decision tree* untuk mencari set parameter yang dapat menghasilkan performa model terbaik.

- **Evaluasi Model *Decision Tree***

1) Evaluasi model pada RapidMiner

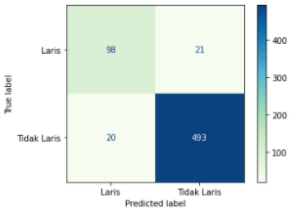
Evaluasi model pada RapidMiner akan dibantu dengan operator *apply model* untuk menggunakan model yg telah dilatih untuk memprediksi data uji. Tahap selanjutnya akan menggunakan operator *performance* untuk mendapatkan akurasi dari model yang telah di latih.



Gambar 10 Evaluasi Pada RapidMiner

Performa terbaik yang dapat dihasilkan yaitu pada pengujian dengan bobot data latih sebesar 90% dan data uji 10%, dengan akurasi sebesar 92.06%, *average precision* 94.31%, *average recall* 79.47%, *dan f1-score 86.89%.*

2) Evaluasi model pada Python



	precision	recall	f1-score	support
Laris	0.83	0.82	0.83	119
Tidak Laris	0.96	0.96	0.96	513
accuracy			0.94	632
macro avg	0.89	0.89	0.89	632
weighted avg	0.93	0.94	0.94	632

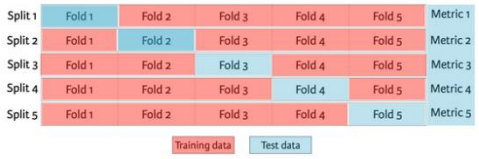
Gambar 11 Evaluasi Pada Python

Untuk mendapatkan akurasi pada pemodelan menggunakan python penulis menggunakan fungsi-fungsi khusus untuk

tujuan evaluasi yang sudah tersedia pada modul scikit-learn untuk mendapatkan *report* klasifikasi. Adapun performa yang dihasilkan dengan pemodelan menggunakan python yaitu akurasi yang dihasilkan diangka 94%, dengan *average precision 89%, average recall 89%, dan f1-score 89%*.

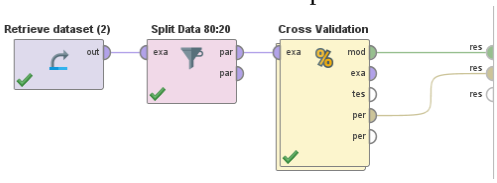
- **Evaluasi Dengan Metode *Cross Validation***

Evaluasi menggunakan split data merupakan metode yang paling sederhana untuk mengevaluasi sebuah model, tetapi metode ini dapat menimbulkan permasalahan ketika dilakukan pelatihan ulang, hasil yang didapat bisa berbeda-beda. Selain itu menggunakan split data dapat terjadi bias ketika hasil yang didapat bagus karena data uji yang terlalu mudah untuk diprediksi, atau hasil yang didapat tidak bagus karena data uji yang terlalu sulit untuk diprediksi sehingga menghasilkan nilai yang kurang optimal. Maka oleh sebab itu solusi yang dapat digunakan untuk memperkecil kemungkinan akurasi yang dihasilkan lebih stabil yaitu dengan menggunakan metode *cross validation* [7]. Berikut ilustrasi dari evaluasi menggunakan *cross validation*.



Gambar 12 Ilustrasi *Cross Validation*
(Sumber: datacamp.com)

1) *Cross Validation* Pada RapidMiner



Gambar 13 *Cross Validation Operator*

Untuk melakukan *cross validation* pada rapidminer dibutuhkan operator *cross validation*, pada operator ini akan menerima input berupa data latih dan akan dilanjutkan pada sub proses pada cross validation berupa proses training menggunakan operator *decision tree*, dan evaluasi menggunakan operator *apply model* dan *performance*.

accuracy: 90.85% +/- 1.63% (micro average: 90.85%)

	true Laris	true Tidak Laris	class precision
pred. Laris	285	41	87.42%
pred. Tidak Laris	190	2009	91.36%
class recall	60.00%	98.00%	

Gambar 14 Performa *Cross Validation*

Pada gambar 14, merupakan performa yang dihasilkan *cross validation* sebanyak 5k-*fold*, yang menghasilkan nilai akurasi rata-rata 90.85%, dengan *precision* 89.39%, *recall* 79% dan *f1-score* 84.20%.

2) *Cross Validation* Pada Python

Untuk melakukan *cross validation* menggunakan python diperlukan *library* dari scikit-learn untuk model *selection* yaitu cross_val_score. Fungsi ini yang akan melakukan evaluasi *cross validation* terhadap model yang sebelumnya telah dilatih pada tahap pemodelan.

Hasil yang dapatkan pada *cross validation* model pada python rata-rata menghasilkan akurasi sebesar 92.83% dengan nilai *precision* 88.51%, *recall* 88.30% dan *f1-score* 88.30%.

- ***Tuning Parameter Decision Tree***

Tahap selanjutnya penulis akan melakukan optimasi atau *tuning* pada parameter *decision tree* untuk mencari parameter terbaik dan terburuk untuk melatih model. Tuning parameter pada data mining merupakan suatu proses pemodelan yang dilakukan secara berulang pada suatu metode dengan menggunakan parameter yang berbeda-beda [8]. Untuk kebutuhan *tuning parameter* penulis akan menggunakan metode *cross validation* sebagai metode evaluasi model, dan acuan dalam proses *tuning* algoritma *decision tree* penulis akan menggunakan acuan nilai *accuracy* untuk memilih set parameter terbaik.

1) *Tuning Parameter* Pada RapidMiner

Tuning parameter pada RapidMiner akan difokuskan pada parameter *criterion*, *maximal depth*, dan *minimal leaf size* dengan batasan sebagai berikut:

- Criterion = [Gain ratio, Information Gain, Gini Index]
- Maximal Depth = [5-6]

- Minimal Leaf Size = [1-2]

Tabel 1 Hasil *Tuning Parameter* Pada RapidMiner

Criterion	*Maximal Depth*	*Minimal Leaf Size*	*Accuracy*	*Rank*
Information Gain	6	1	95.68%	1
Gini Index	6	1	94.97%	2
Information Gain	6	2	94.61%	3
Information Gain	5	1	94.46%	4
Gini Index	6	2	94.18%	5
Gini Index	5	1	94.10%	6
Information Gain	5	2	93.74%	7
Gini Index	5	2	93.58%	8
Gain ratio	5	2	91.13%	9
Gain ratio	6	1	90.73%	10
Gain ratio	6	2	90.61%	11
Gain ratio	5	1	90.26%	12

2) *Tuning Parameter* Pada Python

Tuning parameter pada python pengujian akan difokuskan pada tiga parameter utama yaitu *criterion*, *maximal depth*, dan *minimal leaf size*. Adapun batasan yang akan diuji akan lebih luas dikarenakan pada python terdapat fungsi gridCV yang memungkinkan untuk melakukan *tuning parameter* secara otomatis. Berikut batasan value parameter yang akan diuji.

- Criterion = [‘gini, ‘entropy’]
- Maximal Depth = [1-10]
- Minimal Leaf Size = [1-2]

Tabel 2 Hasi Tuning Parameter Pada Python

Criterion	*Max Depth*	*Min Samples Leaf*	*Accuracy*	*Rank*
entropy	8	1	95.09%	1
gini	9	1	95.01%	2
gini	8	1	95.01%	2
entropy	9	1	94.97%	4
gini	6	1	94.77%	5
entropy	9	2	94.73%	6
entropy	8	2	94.73%	6
entropy	7	1	94.73%	6
gini	8	2	94.73%	6
gini	9	2	94.65%	10
gini	7	1	94.53%	11
entropy	7	2	94.46%	12
gini	6	2	94.38%	13
gini	7	2	94.38%	13
entropy	6	1	93.82%	15
entropy	6	2	93.62%	16
gini	5	1	92.83%	17
entropy	5	2	92.79%	18
entropy	5	1	92.75%	19
gini	5	2	92.71%	20
entropy	4	2	91.88%	21

Criterion	*Max Depth*	*Min Samples Leaf*	*Accuracy*	*Rank*
entropy	4	1	91.84%	22
gini	4	1	91.68%	23
gini	4	2	91.64%	24
entropy	3	1	90.85%	25
entropy	3	2	90.85%	25
gini	3	1	90.73%	27
gini	3	2	90.73%	27
gini	2	2	90.14%	29
gini	2	1	90.14%	29
entropy	2	2	89.43%	31
entropy	2	1	89.43%	31
gini	1	2	89.35%	33
gini	1	1	89.35%	33
entropy	1	2	89.07%	35
entropy	1	1	89.07%	35

Berdasarkan hasil tuning pada tabel 2, akurasi tertinggi dihasilkan pada set parameter criterion(entropy), max_depth(8), dan min_sample_leaf(1) mampu mengasilkan akurasi paling optimal di angka 95.09%.

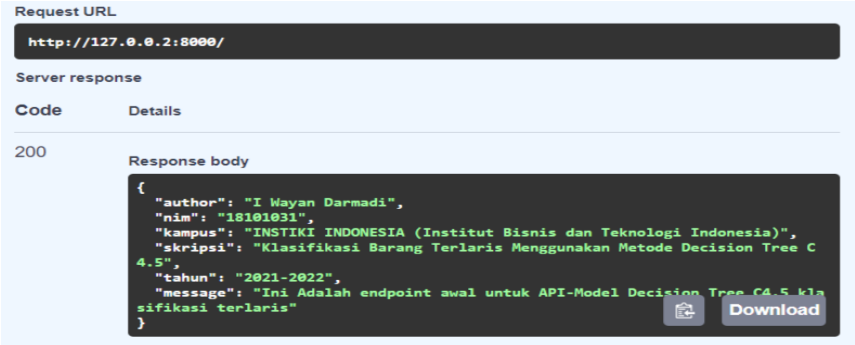
6) Deployment

Pada proses deployment penulis akan menggunakan model dengan akurasi terbaik yang didapatkan melalui hasil *tuning parameter* menggunakan python. Tahap awal penulis akan mengekspor model yang telah dilatih menjadi file *binary* dengan ekstensi .pkl yang berisi aturan pohon keputusan dari model yang telah dilatih menggunakan dataset penjualan barang. File *binary* yang berisi model tersebut hanya dapat diakses menggunakan bahasa python dengan bantuan *library* joblib [9], maka dari itu penulis akan membuat suatu API untuk mengintegrasikan fitur prediksi barang terlaris yang dimiliki oleh model agar dapat diakses pada sistem yang menggunakan bahasa pemprograman selain python. API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan [10].

Tahap pembentukan API dilakukan menggunakan *framework* FastAPI dari python, dimana akan dibuatkan empat *endpoint* untuk mengakses fungsi dari model yang telah dilatih. Adapun empat *endpoint* yang akan dibuat yaitu:

- http://localhost/, Merupakan *base root* dari API, yang menampilkan informasi dari penulis.
- http://localhost/akurasi, *Request type* GET, bertujuan untuk mengakses akurasi dari model yang digunakan.
- http://localhost/kategori, *Request type* GET, ketika pengguna melakukan request terhadape *endpoint* kategori, akan mengembalikan suatu *array* yang berisi kategori barang yang telah dilatih.
- http://localhost/predict, *Request type* POST, pengguna dapat melakukan *request* untuk melakukan klasifikasi/prediksi menggunakan model yang telah dilatih, dengan mengirimkan data produk berdasarkan skema data yang telah ditentukan, dan server akan mengembalikan *response* berupa hasil prediksi berdasarkan data yang dikirim pada saat melakukan post *request* ke *endpoint predict*.

Berikut merupakan hasil *deployment* model kedalam bentuk web API.



Gambar 15 *Response Endpoint* HomePada

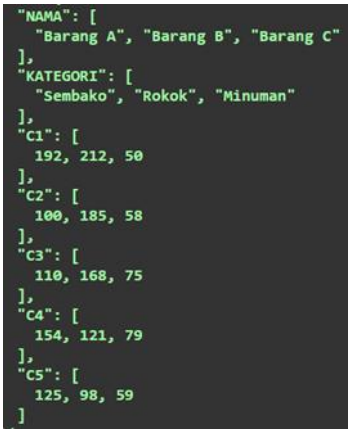
gambar 15, merupakan *response* yang ditampilkan ketika *user* mengirimkan *request* pada *endpoint home* (http://localhost/). Tampilan yang dihasilkan berupa data json dengan identitas diri penulis seperti nama author, nim, asal kampus, judul skripsi, tahun pembuatan, dan pesan.

Gambar 16 *Response Endpoint* Akurasi

Pada gambar 16 merupakan hasil yang ditampilkan ketika *user* mengirim *request* pada endpoint get akurasi (http://localhost/akurasi), data yang didapatkan berupa akurasi seberapa baik performa model yang sedang digunakan.

Gambar 17 *Response Endpoint* Kategori

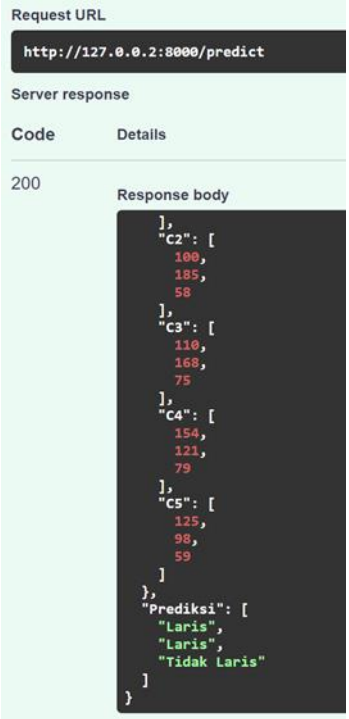
Pada gambar 17, merupakan hasil yang ditampilkan pada saat pengguna melakukan *request* pada *endpoint* get kategori (http://localhost/kategori). Data yang ditampilkan merupakan kumpulan kategori produk yang telah di latih pada model saat ini.



Gambar 18 *Post* Data Prediksi

Pada gambar 18, merupakan data yang akan diprediksi menggunakan bantuan API pada *endpoint* POST *predict* untuk mengakses fungsi prediksi yang dimiliki model. Data

dikirim menggunakan fungsi POST yang akan disematkan pada saat *user* melakukan *request* pada *endpoint ini* (http://localhost/predict).



Gambar 19 *Return Endpoint Predict*

Pada gambar 19, merupakan hasil yang diberikan pada saat *user* mengirim data prediksi melalui *endpoint predict* (http://localhost/predict). Pada bagian akhir objek, terdapat *property* prediksi, dimana *property* ini berisi hasil prediksi barang yang dikirimkan pada API.

SIMPULAN

Hasil evaluasi terhadap model *decision tree* C4.5 dapat disimpulkan bahwa rata-rata akruasi yang dihasilkan menggunakan *cross validation* dengan pemodelan pada RapidMiner yaitu 90.85%, sedangkan pada python akurasi yang dihasilkan diangka 92.83%. Pada proses *tuning parameter* model pada rapidMiner mencapai akurasi tertinggi sebesar 85.68% sedangkan pada python akurasi tertinggi pada proses *tuning parameter* mencapai 95.09%. Untuk tahap *deployment* model kedalam bentuk web API (*Application Programming Interface*) membuat fungsi prediksi dari model yang telah dilatih dapat diakses pada aplikasi lain dengan berbegai teknologi yang mendukung proses integrasi web API.

DAFTAR PUSTAKA

[1] D. T. Larose and C. D. Larose, *Discovering knowledge in data: an introduction to data mining*, vol. 4. John Wiley & Sons, 2014.

[2] A. H. Nasrullah, "IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI PRODUK LARIS," *J. Pilar Nusa Mandiri*, vol. 14, no. 2, p. 217, 2018, doi: 10.33480/pilar.v14i2.926.

[3] Suherman and I. Muzaky, "Analisis Penjualan Barang Laris Dan Kurang Laris Terhadap Percetakan Awfa Digitl Printing Menggunakan Metode Decision Tree Dengan Optimasi Algoritma Genetika," *J. Teknol. Pelita Bangsa*, vol. 10, no. 9–1 (87), pp. 153–167, 2019.

[4] N. Wardani, *Penerapan Data Mining Dalam Analytic CRM*, no. May. 2020.

[5] M. J. Zaki and M. J. Meira, "Data Mining and Analysis: Fundamental Concepts and Algorithms," p. 562, 2013, [Online]. Available: https://books.google.com.tr/books?id=Gh9GAwAAQBAJ&lpg=PR9&dq=Data Mining and Analysis: Foundations and Algorithms&hl=tr&pg=PR9#v=onepage&q=Data Mining and Analysis: Foundations and Algorithms&f=false.

[6] Aprilla Dennis, "Belajar Data Mining dengan RapidMiner," *Innov. Knowl. Manag. Bus. Glob. Theory Pract. Vols 1 2*, vol. 5, no. 4, pp. 1–5, 2013, [Online]. Available: http://esjournals.org/journaloftechnology/archive/vol1no6/vol1no6_6.pdf%5Cnhttp://www.airccse.org/journal/nsa/5413nsa02.pdf.

[7] A. W. Wardhana, E. Patimah, A. I. Shafarindu, Y. M. Siahaan, B. V. Haekal, and D. S. Prasvita, "Klasifikasi Data Penjualan pada Supermarket dengan Metode Decision Tree," *Senamika*, vol. 2, no. 1, pp. 660–667, 2021, [Online]. Available:

https://conference.upnvj.ac.id/index.php/senamika/article/view/1389.

[8] R. G. Mantovani, T. Horváth, R. Cerri, J. Vanschoren, and A. C. de Carvalho, "Hyper-parameter tuning of a decision tree induction algorithm," in *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, 2016, pp. 37–42.

[9] Purnendu Shukla, "How to save and load machine learning models using Pickle." https://practicaldatascience.co.uk/machine-learning/how-to-save-and-load-machine-learning-models-using-pickle (accessed Jul. 29, 2022).

[10] Anugrah Sandi, "Mengenal Apa itu Web API," *Codepolitan*, 2017. https://www.codepolitan.com/mengenal-apa-itu-web-api-5a0c2855799c8 (accessed Jul. 29, 2022).