Prediksi Kenaikan Jabatan Pranata Komputer pada Kementerian X dengan Menggunakan Model Algoritma Klasifikasi *Linear Discriminant Analysis* (LDA)

Savira Rahmania Putri Ariyanto[1], Wiyli Yustanti[2]

[1,2] Sistem Informasi, Teknik Informatika, Universitas Negeri Surabaya
[1]savira.19039@mhs.unesa.ac.id
[2]wiyliyustanti@unesa.ac.id

***Abstrak* — Kementerian X memiliki data mengenai riwayat kenaikan jabatan pegawainya, khususnya pada pegawai Pranata Komputer. Kementerian X merasa kesulitan untuk mendata pegawainya yang sudah saatnya untuk naik jabatan. Kementerian X membutuhkan sebuah model algoritma untuk memprediksi lama kenaikan jabatan pranata komputer di instansinya. Hasil prediksi yang diinginkan berupa label / kelas yakni label A, B dan C dengan durasi lama waktu yang berbeda. Label A untuk pegawai yang diprediksi dapat naik jabatan antara 0 – 2 tahun, Label B untuk pegawai yang diprediksi dapat naik jabatan antara 3 – 5 tahun dan Label C untuk pegawai yang diprediksi dapat naik jabatan lebih dari 5 tahun. Dalam penelitian ini, peneliti menggunakan salah satu model algoritma klasifikasi yaitu *Linear Discriminant Analysis* dengan *fold 4* untuk proses prediksi. Peneliti menggunakan metode CRISP-DM (*Cross Industry Standard Process for Data Mining*) dan *tools* JupyterLab. Dataset yang digunakan untuk proses prediksi terdiri dari 15 variabel independen dan 1 variabel dependen (target) yang bersifat *multiclass.* Hasil penelitian ini menunjukkan bahwa model *Linear Discriminant Analysis* (LDA) menghasilkan *f1 score training* sebesar 0,9345 dan *f1 score testing* sebesar 0,9765**

***Kata Kunci* — Klasifikasi, Prediksi, *Linear Discriminant Analysis*, JupyterLab, F1 Score**

I. PENDAHULUAN

Pegawai adalah aset paling berharga bagi instansi maupun perusahaan. Peran mereka penting karena dapat menjaga keberlangsungan operasional, membantu mencapai visi dan misi serta menciptakan lingkungan kerja yang produktif. Di Indonesia, umumnya terdapat dua jenis pegawai yakni pegawai swasta dan pegawai negeri. Pegawai swasta bekerja pada perusahaan atau lembaga swasta. Sedangkan pegawai negeri bekerja pada negara dan pemerintahan. Pegawai negeri sipil atau PNS merupakan bagian dari Aparatur Sipil Negara (ASN).

Setiap instansi pemerintahan termasuk di Kementerian X, mulanya terdapat dua jenis jabatan yakni jabatan struktural atau jabatan administrasi dan jabatan fungsional. Tetapi dengan adanya peraturan baru dijelaskan bahwa Jabatan Administrasi disetarakan dengan Jabatan Fungsional. Penyetaraan Jabatan ini merupakan pengangkatan Jabatan Administrasi ke dalam Jabatan Fungsional melalui penyesuaian / *inpassing* pada Jabatan Fungsional yang setara [1]. Jabatan Fungsional erat kaitannya dengan Angka Kredit. Angka Kredit merupakan nilai kuantitatif dari hasil kerja Pejabat Fungsional. Angka Kredit Kumulatif merupakan akumulasi nilai Angka Kredit yang harus dicapai oleh Pejabat Fungsional sebagai salah satu syarat kenaikan jabatan [2]. Sehingga pegawai Jabatan Fungsional harus mengumpulkan nilai Angka Kredit kemudian akan dilakukan penilaian oleh Tim Penilai Angka Kredit Jabatan Fungsional.

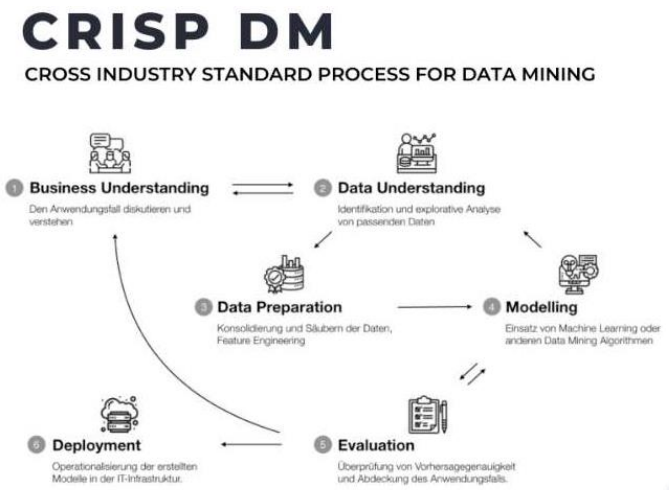
Kementerian X memiliki data mengenai riwayat kenaikan jabatan pegawainya, khususnya pada pegawai Jabatan Fungsional Pranata Komputer (PK). Jabatan Fungsional Pranata Komputer (PK) adalah jabatan yang mempunyai ruang lingkup, tugas, tanggungjawab, wewenang dan hak untuk melaksanakan kegiatan sistem teknologi informasi berbasis komputer yang meliputi tata kelola dan tata laksana teknologi informasi, infrastruktur teknologi informasi, serta sistem informasi dan multimedia [3]. Kementerian X merasa kesulitan untuk mendata pegawainya yang sudah saatnya untuk naik jabatan. Kementerian X membutuhkan sebuah model algoritma untuk membantu memprediksi durasi kenaikan jabatan dari pranata komputer di instansinya.

Berdasarkan masalah diatas peneliti berusaha untuk menyelesaikan permasalahan yang dihadapi oleh Kementerian X untuk membuat sebuah model klasifikasi. Model klasifikasi dipilih karena *output* yang diharapkan berupa kelas atau label. Peneliti menggunakan salah satu model algoritma klasifikasi yakni *Linear Discriminant Analysis* (LDA) dengan *fold* 4. Dari model tersebut akan dilakukan evaluasi metrik pada data *training* dan data *testing*, dilakukan analisis apakah model tersebut terindikasi *overfitting* ataupun *underfitting*. Model LDA juga akan dievaluasi dengan beberapa plot model.

Hasil prediksi akan berupa label / kelas yakni label A, B dan C dengan durasi kenaikan jabatan yang berbeda-beda. Label A untuk pegawai dengan kenaikan jabatan antara 0 – 2 tahun, Label B untuk pegawai dengan kenaikan jabatan antara 3 – 5 tahun sedangkan untuk Label C untuk pegawai dengan kenaikan jabatan diatas 5 tahun. Dengan pengklasifikasian tersebut, diharapkan Kementerian X dapat mengetahui prediksi pegawai untuk naik jabatan berdasarkan lama waktu kenaikan jabatannya.

II. METODE PENELITIAN

Pada penelitian ini menggunakan metode CRISP-DM (*Cross Industry Standard Process for Data Mining*) yang diusulkan oleh Tuga dan Faisal [4]. Berikut ini grafik alur tahapannya :



Gbr. 1 Grafik alur tahapan CRISP-DM

A. *Business Understanding*

Business Understanding adalah tahapan yang memerlukan pemahaman dari sisi objek bisnis, cara membangun dan memperoleh data serta menyesuaikan dengan tujuan pemodelan untuk mencapai tujuan bisnis sehingga model terbaik dapat ditemukan.

B. *Data Understanding*

Data Understanding merupakan tahapan untuk memahami kondisi dataset serta mengidentifikasi masalah yang mungkin ada dalam dataset tersebut. Data yang digunakan dalam penelitian ini yakni data jafung_jfpk-v4.csv Dalam dataset ditentukan batasan *scope* pegawai dan periode waktu.

C. *Data Preparation*

Data Preparation adalah tahapan data *treatment* menuju model berkualitas yang berguna [2]. Berikut adalah tahapan Data Preparation yang dilakukan pada penelitian ini :

1) Menyesuaikan tipe data pada kolom
2) Mengubah nilai '(NULL)' menjadi nol (0)
3) Memeriksa baris duplikat
4) Memeriksa *missing value*
5) Memeriksa *outlier*
6) Memberikan label pada dataset
7) Melakukan proses *encoding*
8) Melakukan *grouping*

Hasil *Data Preparation* diatas akan dihasilkan variabel dependen atau kelas/target serta variabel independen untuk pemodelan klasifikasi.

D. *Modelling*

Modelling adalah tahapan membuat model prediktif dan deskriptif, pada tahap ini dilakukan penggunaan *Machine Learning* untuk menggunakan model atau algoritma untuk proses prediksi. Pada penelitian ini, proses *modelling* akan dilakukan beberapa tahapan berikut ini :

1) Melakukan *SetUp Environment*

Dalam proses ini, menentukan kolom yang menjadi target, tipe target kolom tersebut (*multiclass* atau *binary class*), mengubah tipe data pada kolom apabila masih belum sesuai, menghapus atau mengabaikan kolom yang tidak ingin digunakan dalam proses *modelling*, mengetahui jumlah *original* data, mengubah banyaknya *fold* yang ingin digunakan, mengelompokkan kolom kolom yang termasuk numerik atau kategorik, mengatur *fold generator*, pembagian antara *train* dan *test set* yang *default* nya memiliki komposisi *train set* 70% dan *test set* 30%.

2) Menggunakan Model Algoritma Klasifikasi

Setelah proses *SetUp Environment*, data *train set* dimasukkan ke model algoritma klasifikasi *Linear Discriminant Analysis* dan dimunculkan skor menggunakan *cross validation stratifiedkfold* untuk *evaluation metric*. Hasil keluarannya berupa nilai *Accuracy, Recall, Precision, F1-Score*, Kappa, MCC (*Matthews Correlation Coefficient*) serta *training times* pada setiap *fold* nya.

E. *Evaluation*

Tahapan *Evaluation* juga merupakan tahapan melakukan interpretasi terhadap hasil dari data mining yang dihasilkan dari model algoritma yang digunakan. *Evaluation* yang dilakukan di penelitian ini antara lain :

1) Membuat plot model

Plot model dapat digunakan untuk menganalisa *performance* dari setiap aspek yang berbeda pada model *Linear Discriminant Analysis*, antara lain *Confusion Matrix, Area Under Curve, Precision Recall Curve, Class Prediction Error* dan *Classification Report*. Dalam plot model, data yang diambil adalah *train set* dan dikembalikan / hasil *output* dalam bentuk plot adalah hasil *test set*.

2) Melakukan Prediksi pada *Test Set*

Dalam tahapan ini dilakukan prediksi menggunakan model *Linear Discriminant Analysis*, data yang digunakan adalah data *test set* dengan komposisi 30% dari data awal. Apabila nilai evaluasi sudah muncul, dapat dibandingkan dengan nilai pada *training set*. Jika nilai *training set* jauh lebih tinggi dari pada nilai *test set* maka terindikasi *overfitting*. Sedangkan, jika nilai *test set* jauh lebih tinggi dari pada nilai *training set* maka terindikasi *underfitting*. Nilai yang baik yakni antara nilai *training set* dengan *test set* tidak terlalu besar perbedaannya.

F. *Deployment*

Tahap *Deployment* adalah tahap rencana penggunaan model dan menggabungkan dengan keputusan dalam sistem operasional. Meskipun model sudah digunakan tetapi model juga perlu untuk dipantau dan diganti dengan model model yang lebih baik lagi di masa mendatang. Finalisasi model adalah tahap terakhir untuk menyesuaikan model terbaik

kedalam dataset yang utuh, termasuk test set. Tujuannya adalah melatih model pada dataset yang lengkap sebelum disebarluaskan untuk diproduksi.

III. HASIL DAN PEMBAHASAN

Bagian ini membahas mengenai hasil dari setiap tahapan yang sudah dijelaskan pada bagian sebelumnya mulai dari *Business Understanding* sampai dengan *Deployment.*

A. *Business Understanding*

Penelitian ini mengangkat permasalahan bisnis yang ada pada Kementerian X. Kementerian X khususnya Pimpinan Pranata Komputer ingin mengetahui berapa lama pegawai Pranata Komputer dapat mencapai kenaikan jabatan. Dengan adanya suatu model algoritma *Data Mining* yang dapat memprediksi lama kenaikan jabatan, pihak Kementerian X dapat mengetahui apakah pegawai Pranata Komputer banyak yang mencapai kenaikan jabatan secara cepat, sedang atau lambat berdasarkan *range* tahun yang sudah disesuaikan. Apabila model algoritma klasifikasi *Linear Discriminant Analysis* menghasilkan nilai evaluasi yang tinggi, maka akan memudahkan pimpinan untuk mengambil tindakan terhadap pegawai tersebut melalui proses prediksi.

B. *Data Understanding*

Data yang digunakan dalam penelitian ini yakni data *jafung_jfpk-v4.csv* Dalam dataset tersebut terdapat beberapa batasan :

1) *Scope* Pegawai : Seluruh Pegawai PraKom (Pranata Komputer) di Kementerian X
2) Periode Waktu : Tahun 2014 hingga Tahun 2021

Beberapa informasi mengenai dataset tersebut antara lain :

1) Dataset terdiri dari 1 tabel yang berisikan 35.463 baris dan 22 kolom
2) Jenis jabatan fungsional pranata komputer dibedakan menjadi 2 yakni pranata komputer keterampilan dan pranata komputer keahlian
3) Skema tingkatan jabatan fungsional pranata komputer (dari terendah hingga tertinggi) yang terdapat dalam dataset tersebut yakni :

TABEL I
TINGKATAN JABATAN FUNGSIONAL PRANATA KOMPUTER

No	Jabatan Fungsional PK Keterampilan	Jabatan Fungsional PK Keahlian
1.	Calon JF Terampil	Calon JF Ahli
2.	Terampil	Ahli Pertama
3.	Mahir	Ahli Muda
4.	Penyelia	Ahli Madya

4) Berikut ini deskripsi setiap kolom dalam dataset :

TABEL II
DESKRIPSI KOLOM DATASET

No	Nama Kolom	Deskripsi
1.	t_dupak_id	Nomor ID pengusulan Daftar Usulan Penetapan Angka Kredit (DUPAK)
2.	r_pegawai_id	Nomor ID pranata komputer
3.	r_nama	Nama pranata komputer
4.	r_nip	Nomor Identitas Pegawai Negeri Sipil dari pranata komputer
5.	r_pangkat	Pangkat pranata komputer
6.	r_golongan	Golongan pranata komputer
7.	jabatan	Jabatan pranata komputer
8.	r_tmt_jabatan	Terhitung mulai tanggal jabatan
9.	r_tahun	Tahun pranata komputer mengusulkan penetapan angka kredit
10.	r_semester	Semester saat pengusulan penetapan angka kredit diajukan
11.	unsur	Item kegiatan yang diusulkan oleh pranata komputer
12.	sub_unsur	Sub-item kegiatan yang diusulkan oleh pranata komputer
13.	kegiatan	Kegiatan yang telah dilakukan oleh pranata komputer (dalam lingkup unsur)
14.	t_keterangan	Keterangan tambahan atau penjelasan terkait kegiatan yang telah dilakukan oleh pranata komputer
15.	t_volume	Jumlah item kegiatan yang diusulkan oleh pranata komputer
16.	t_usulan_ak	Angka kredit yang diusulkan oleh pranata komputer
17.	t_volume_pleno	Jumlah item kegiatan yang dinilai oleh tim Penilai
18.	t_ak_pleno	Angka kredit yang dinilai oleh tim Penilai
19.	nama_penilai1	Nama Penilai 1
20.	nip_penilai1	Nomor Identitas Pegawai Negeri Sipil dari Penilai 1
21.	nama_penilai2	Nama Penilai 2
22.	nip_penilai2	Nomor Identitas Pegawai Negeri Sipil dari Penilai 2

C. *Data Preparation*

1) Menyesuaikan tipe data pada kolom

Dalam dataset masih terdapat beberapa kolom yang belum sesuai tipe datanya. Maka, dilakukan:

- Mengubah tipe data kolom t_dupak_id, r_pegawai_id, r_nip, r_tahun, r_semester, nip_penilai1, dan nip_penilai2 menjadi '*object'*
- Mengubah tipe data kolom t_volume dan t_volume_pleno menjadi 'float' serta t_usulan_ak dan t_ak_pleno menjadi '*int'*.

2) Mengubah nilai '(NULL)' menjadi nol (0)

Saat akan mengubah tipe data kolom numerik menjadi *integer* atau *float*, *output* yang dihasilkan *error* karena terdapat nilai '(NULL)' dalam kolom-kolom tersebut. Maka, dilakukan pengubahan nilai '(NULL)' menjadi nol (0) dalam kolom t_volume dan t_volume_pleno, t_usulan_ak, dan t_ak_pleno. Hal ini dikarenakan nilai '(NULL)' berarti usulan volume atau angka kredit dan volume pleno atau angka kredit pleno tersebut ditolak

3) Memeriksa baris duplikat

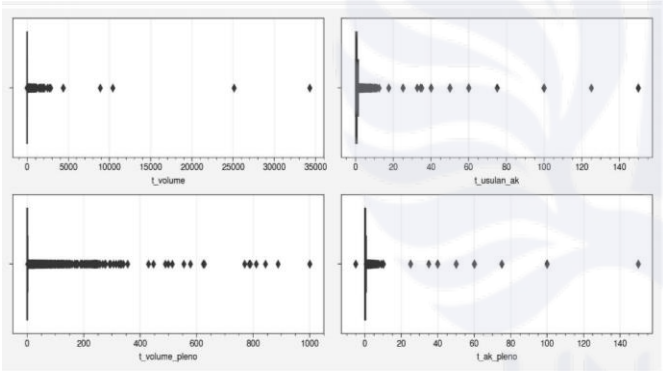
Terdapat 1.921 baris data yang duplikat. Tetapi, data yang duplikat tidak dihapus karena dalam *use case* ini sangat memungkinkan bagi pegawai mengajukan volume dan angka kredit yang sama, dengan unsur sub unsur yang sama serta di tahun dan semester yang sama.

4) Memeriksa *missing value*

Tidak ada baris yang *missing value*. Semua baris terisi sesuai dengan tipe data masing masing kolom.

5) Memeriksa *outlier*

Menggunakan *Box Plot* untuk analisis *outlier* pada kolom numerik, tetapi data yang *outlier* tidak di lakukan *remove*, hanya di cek saja.



Gbr. 2 Hasil analisis *outlier*

6) Memberikan label pada dataset

Pada dataset tersebut terdapat kolom 'selisih_jabatan', dengan menggunakan kolom tersebut setiap baris dikelompokkan dalam label tertentu sesuai lama kenaikan jabatannya. Label A untuk pegawai dengan lama kenaikan jabatan 0 – 2 tahun. Label B untuk pegawai dengan lama kenaikan jabatan 3 – 5 tahun dan Label C untuk pegawai dengan lama kenaikan jabatan diatas 5 tahun. Pemberian label menggunakan *Cut Point* dari *Binning Function*.

vol_akumulasi	vol_pleno_akumulasi	usulan_ak_akumulasi	ak_pleno_akumulasi	Label
8	8	11.000	11.000	Label A
65	65	17.576	17.576	Label A
6	3	4.500	4.000	Label A
24	21	13.129	10.104	Label A

Gbr. 3 Hasil pemberian label pada dataset

7) Melakukan proses *encoding*

Kolom unsur memiliki 8 macam jenis antara lain unsur Analisis dan Perancangan Sistem Informasi, Implementasi Sistem Informasi, Implementasi Teknologi Informasi, Operasi Teknologi Informasi, Pendidikan, Pendukung Kegiatan Pranata Komputer, Pengembangan Profesi, Penyusunan Kebijakan Sistem Informasi. Proses *encoding* kolom unsur diperlukan guna mengetahui unsur mana yang paling banyak diambil dan berpengaruh terhadap lama kenaikan jabatan pranata komputer.

Proses *encoding* menggunakan *Label Encoder* untuk mentransformasi label kata menjadi bentuk numerik. Kemudian dilanjut menggunakan *One Hot Encoding*, yang merupakan proses pengubahan data kategorikal *integer* menjadi *boolean* (*true* atau *false*) dimana setiap unsur yang sudah dilakukan label *encoder* akan di *expand* menjadi kolom atau parameter baru.

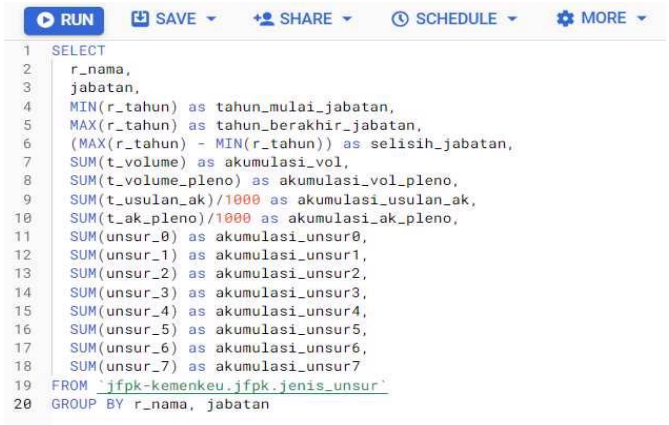
```
# Proses encoding menggunakan Label Encoder

from sklearn.preprocessing import LabelEncoder
# creating initial dataframe
unsur = proses_encode
jenis_unsur = pd.DataFrame(unsur, columns=['unsur'])
# creating instance of labelencoder
labelencoder = LabelEncoder()
# Assigning numerical values and storing in another column
jenis_unsur['jenis_unsur'] = labelencoder.fit_transform(jenis_unsur['unsur'])
jenis_unsur
```

	unsur	jenis_unsur
0	PENDUKUNG KEGIATAN PRANATA KOMPUTER	5
1	PENDIDIKAN	4
2	PENDIDIKAN	4
3	PENDIDIKAN	4
4	PENDIDIKAN	4
...	...	...
35457	PENDIDIKAN	4

Gbr. 4 Hasil *encoding* kolom unsur

8) Melakukan *grouping*

Kemudian, dilakukan grouping berdasarkan jabatan pada Google Big Query seperti gambar dibawah ini :



```
RUN  SAVE  SHARE  SCHEDULE  MORE
SELECT
  r_nama,
  jabatan,
  MIN(r_tahun) as tahun_mulai_jabatan,
  MAX(r_tahun) as tahun_berakhir_jabatan,
  (MAX(r_tahun) - MIN(r_tahun)) as selisih_jabatan,
  SUM(t_volume) as akumulasi_vol,
  SUM(t_volume_pleno) as akumulasi_vol_pleno,
  SUM(t_usulan_ak)/1000 as akumulasi_usulan_ak,
  SUM(t_ak_pleno)/1000 as akumulasi_ak_pleno,
  SUM(unsur_0) as akumulasi_unsur0,
  SUM(unsur_1) as akumulasi_unsur1,
  SUM(unsur_2) as akumulasi_unsur2,
  SUM(unsur_3) as akumulasi_unsur3,
  SUM(unsur_4) as akumulasi_unsur4,
  SUM(unsur_5) as akumulasi_unsur5,
  SUM(unsur_6) as akumulasi_unsur6,
  SUM(unsur_7) as akumulasi_unsur7
FROM `jfpk-kemenkeu.jfpk.jenis_unsur`
GROUP BY r_nama, jabatan
```

Gbr. 5 Proses *grouping*

Setelah dilakukan proses diatas, dihasilkan dataset baru dengan jumlah kolom sebanyak 18 dan jumlah baris sebanyak 269 serta ditentukan variabel dependen atau data kelas/target serta variabel independen untuk pemodelan klasifikasi. Berikut adalah variabel dependen atau data kelas/target :

TABEL III
VARIABEL DEPENDEN

Simbol	**Variabel**	**Tipe Data**	**Nilai Data**	**Keterangan**
y	Kelas Lama Kenaikan Jabatan	Kategorik	Label A/0	Label A untuk pegawai dengan lama kenaikan jabatan 0 – 2 tahun.
			Label B/1	Label B untuk pegawai dengan lama kenaikan jabatan 3 – 5 tahun
			Label C/2	Label C untuk pegawai dengan lama kenaikan jabatan diatas 5 tahun

Berikut adalah variabel independen yang digunakan untuk *Modelling* :

TABEL IV
VARIABEL INDEPENDEN

X	**Fitur**	**Tipe Data**	**Nilai Data**	**Keterangan**
x_1	Volume	Numerik	Bilangan Riil Positif	Jumlah item kegiatan yang diusulkan oleh pranata komputer
x_2	Volume Pleno	Numerik	Bilangan Riil Positif	Jumlah item kegiatan yang dinilai oleh tim Penilai
x_3	Usulan AK	Numerik	Bilangan Riil Positif	Angka kredit yang diusulkan oleh pranata komputer
x_4	AK Pleno	Numerik	Bilangan Riil Positif	Angka kredit yang dinilai oleh tim Penilai
x_5	Tahun Mulai Jabatan	Kategorik	0 = 2014, 1 = 2015, 2 = 2016, 3 = 2017, 4 = 2018, 5 = 2019, 6 = 2020, 7 = 2021	Tahun terlama pada pegawai setelah di grouping (diambil dari kolom r_tahun)
x_6	Tahun Berakhir Jabatan	Kategorik	0 = 2014, 1 = 2015, 2 = 2016, 3 = 2017, 4 = 2018, 5 = 2019, 6 = 2020, 7 = 2021	Tahun terbaru pada pegawai setelah di grouping (diambil dari kolom r_tahun)
x_7	Jabatan	Kategorik	0 = Calon JF Terampil, 1= Terampil, 2 = Mahir, 3 = Penyelia, 4 = Calon JF Ahli, 5 = Ahli Pertama, 6 = Ahli Muda, 7 = Ahli Madya,	Jenis Jabatan saat ini
x_8	Akumu-lasi Unsur 0	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 0 (Analisis dan Peran-cangan Sistem Informasi) pada

				pegawai setelah di grouping
x_9	Akumulasi Unsur 1	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 1 (Implementasi Sistem Informasi) pada pegawai setelah di grouping
x_{10}	Akumulasi Unsur 2	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 2 (Implementasi Teknologi Informasi) pada pegawai setelah di grouping
x_{11}	Akumulasi Unsur 3	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 3 (Operasi Teknologi Informasi) pada pegawai setelah di grouping
x_{12}	Akumulasi Unsur 4	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 4 (Pendidikan) pada pegawai setelah di grouping
x_{13}	Akumulasi Unsur 5	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 5 (Pendukung Kegiatan Pranata Komputer) pada pegawai setelah di grouping
x_{14}	Akumulasi Unsur 6	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 6 (Pengembangan Profesi) pada pegawai setelah di grouping
x_{15}	Akumulasi Unsur 7	Numerik	Bilangan Riil Positif	Jumlah akumulasi unsur 7 (Penyusunan Kebijakan Sistem Informasi) pada pegawai setelah di grouping

D. *Modelling*

1) *Melakukan SetUp Environment*

Dilakukan beberapa *setting* sebelum menggunakan model algoritma klasifikasi diantaranya kolom ‘Label’ menjadi kolom target, tipe kolom target bersifat *multiclass* karena terdapat 3 label, mengubah tipe data kolom ‘akumulasi_unsur6’ dan ‘akumulasi_unsur7’ yang awalnya masuk tipe data kategorikal menjadi numerikal, menghapus atau mengabaikan kolom ‘r_nama’ dan ‘selisih_jabatan’ karena tidak bisa digunakan sebagai variabel independen, mengatur *fold* = 4, mengelompokkan kolom berdasarkan tipe kolom numerik dan kategorik, kolom yang termasuk kategorik antara lain ‘Jabatan’, ‘tahun_mulai_jabatan’,‘tahun_berakhir_jabatan’ sedangkan untuk kolom numerik antara lain ‘Akumulasi_vol’,‘akumulasi_vol_pleno’, ‘Akumulasi_ak’, ‘akumulasi_ak_pleno’,‘Akumulasi_unsur0’,‘Akumulasi_unsur1’,‘Akumulasi_unsur2’,‘Akumulasi_unsur3’,‘Akumulasi_unsur4’, ‘Akumulasi_unsur5’, ‘Akumulasi_unsur6’, ‘Akumulasi_unsur7. Untuk *fold generator* sebagai strategi dalam *cross validation* yang digunakan adalah *stratifiedkfold*, selanjutnya untuk komposisi data *training* dan data *testing* adalah sebagai berikut :

TABEL V
PEMBAGIAN DATA *TRAINING* DAN DATA *TESTING*

Keterangan	**Data Training**	**Data Testing**	**Total**
Presentase	70%	30%	100%
Jumlah	188	81	269

2) Menggunakan Model Algoritma Klasifikasi

Setelah dilakukan *SetUp*, model algoritma klasifikasi *Linear Discriminant Analysis* digunakan dengan memanggil fungsi *create_model()* Kemudian disimpan dalam variabel 'model1_lda'. Berikut ini adalah hasil nilai evaluasi matrik *Accuracy, AUC, Recall, Precision, F1 Score, Kappa* dan *MCC* dari model LDA pada masing masing *fold.* Nilai berikut ialah nilai dari hasil dataset *training.*

```
[16]: model1_lda = create_model('lda')
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9787	0.9894	0.9915	0.9814	0.9793	0.9299	0.9322
1	0.9574	0.9963	0.9048	0.9595	0.9544	0.8362	0.8482
2	0.8936	0.9867	0.8246	0.9103	0.8924	0.6564	0.6636
3	0.9149	0.9767	0.7293	0.9160	0.9118	0.7219	0.7241
Mean	0.9362	0.9873	0.8625	0.9418	0.9345	0.7861	0.7920
Std	0.0336	0.0070	0.0969	0.0297	0.0343	0.1050	0.1048

Gbr. 6 Hasil Evaluasi Metrik Model

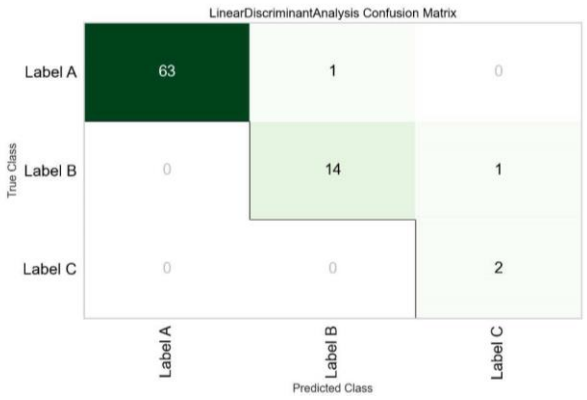
E. *Evaluation*

Jika pada tahap sebelumnya model dinilai pada dataset *training*, dalam tahap ini model akan dianalisis pada dataset *testing*.

1) Membuat plot model

Dalam membuat plot model, dapat dilakukan dengan 2 cara, yakni menggunakan fungsi *evaluate_models()* dan *plot_models()*. Apabila menggunakan fungsi *evaluate_models()* maka variabel yang sudah menyimpan model terbaik yaitu variabel 'model1_lda' harus dipanggil. Hasil dari fungsi *evaluate_models()* akan menampilkan semua plot sehingga *user* hanya perlu memilih tipe dari plot yang diinginkan, tetapi *output* ini hanya bisa ditampilkan ketika sedang menjalankan *Jupyter-Lab* saja. Sedangkan *plot_models()* dapay ditampilkan kapan saja meskipun tidak sedang menjalankan *Jupyter-Lab* Berikut ini adalah hasil dari *plot_models()*

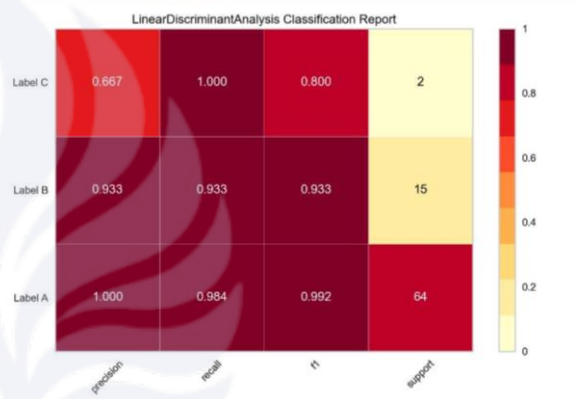
```
[19]: plot_model(model1_lda,'confusion_matrix')
```



Gbr. 7 Hasil *Confusion Matrix* Model LDA

Gambar diatas merupakan hasil dari plot *confusion_matrix*. Plot diatas menggambarkan bagaimana masing masing kelas prediksi atau *predicted class* dibandingkan dengan kelas aktualnya atau *true class.* Dari total dataset *testing* dalam data kenaikan jabatan yang berjumlah 81 data. Terdapat 64 data yang berlabel A, 15 data yang berlabel B dan 2 data yang berlabel C. Tetapi, dari hasil plot *confusion matrix*, terdapat 63 data yang diprediksi benar masuk label A dan 1 data yang diprediksi salah, karena di prediksi masuk ke label B. Untuk 15 data label B, terdapat 14 data diprediksi benar masuk label B dan 1 data yang diprediksi salah, karena di prediksi masuk ke label C. Untuk 2 data label C, keduanya diprediksi benar masuk label C.

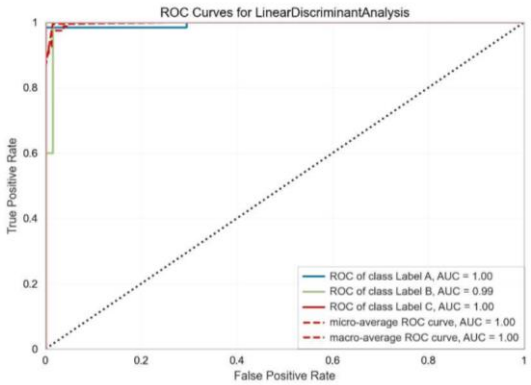
```
[23]: plot_model(model1_lda, plot = 'class_report')
```



Gbr. 8 Hasil Plot *Class Report* Model LDA

Gambar diatas merupakan hasil dari plot *class_report.* Plot diatas menampilkan nilai dari evaluasi metrik *precision, recall, f1 score* serta menampilkan *support* dari masing masing label/kelas. Dalam *class_report* semakin baik nilainya maka akan semakin gelap warna merah yang ditampilkan serta nilai mendekati angka 1.0 Dari hasil plot, dapat diketahui nilai *f1 score* pada masing masing kelas memiliki nilai diatas 0,8 yang bermakna model ini menghasilkan skor yang baik. *Support* pada plot tidak berpengaruh apa apa terhadap model, hanya menampilkan jumlah data yang sebenarnya.

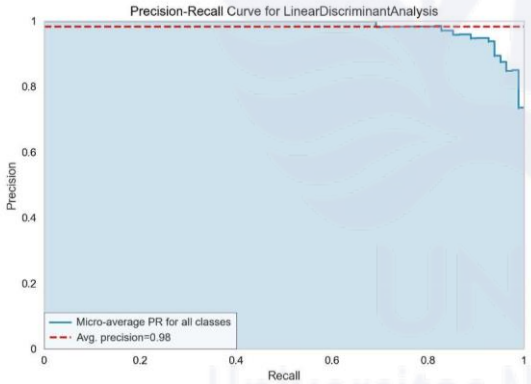
```
[20]: plot_model(model1_lda, plot = 'auc')
```



Gbr. 9 Hasil Plot AUC Model LDA

Gambar diatas merupakan hasil dari plot *auc* atau *area under the curve*. Pada plot *auc*, sumbu y akan menampilkan tingkat positif sebenarnya atau *true positive rate* dan sumbu x akan menampilkan tingkat positif palsu atau *false positive rate*. Titik ideal dari grafik tersebut adalah di sudut kiri atas ketika *false positive rate* bernilai 0 dan *true positive rate* bernilai 1. Dihasilkan nilai ROC pada setiap kelas/label, rata rata micro serta rata rata macro mendekati nilai 1.0 sehingga model dikatakan baik.

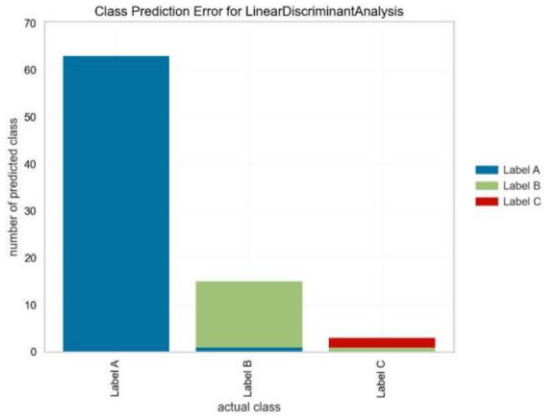
```
[21]: plot_model(model1_lda, plot = 'pr')
```



Gbr. 10 Hasil Plot *Precision Recall* Model LDA

Gambar diatas merupakan hasil dari plot *pr* atau *precision recall curve*. Plot diatas menampilkan hasil dari nilai *precision* pada sumbu y dan nilai *recall* pada sumbu x. Nilai antara *precision* dan *recall* antara 0 hingga 1. Model yang baik adalah model yang dapat memaksimalkan nilai hingga mendekati 1.0. Pada model yang sudah dibuat mendapatkan nilai rata rata sebesar 0.98

```
[24]: plot_model(model1_lda, plot = 'error')
```



Gbr. 11 Hasil Plot *Class Prediction Error* Model LDA

Gambar diatas merupakan hasil dari plot *error* atau *class prediction error*. Hampir sama seperti plot *class_report*, plot ini menampilkan jumlah *support* atau jumlah data untuk setiap kelas dalam bentuk diagram batang bertumpuk. Dapat terlihat bahwa terdapat data yang seharusnya label A tetapi masuk ke label B kemudian terdapat data yang seharusnya label B tetapi masuk ke label C.

2) Melakukan Prediksi pada *Test Set*

Pada tahap ini, dilanjutkan dengan melakukan prediksi terhadap data *testing* yang berjumlah 30% dari dataset keseluruhan. Nilai yang muncul pada saat tahapan *create_model()* merupakan nilai dari hasil menggunakan data *training*. Dalam tahapan ini menggunakan fungsi *predict_model()* untuk memprediksi. Berikut ini adalah hasil prediksi pada dataset kenaikan jabatan :

```
predict_model(model1_lda)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Linear Discriminant Analysis	0.9753	0.9952	0.9726	0.9794	0.9765	0.9295	0.9302

Gbr. 12 Hasil Evaluasi Metrik pada Data *Testing*

TABEL VI
HASIL SCORE TRAINING DAN SCORE TESTING

Model	**Metrik evaluasi**	**Score Training**	**Score Testing**
Linear Discriminant Analysis – fold 4	*F1 Score*	0,9345	0,9765

Dari hasil evaluasi metrik pada prediksi diatas, dapat dilihat jika nilai *f1 score* antara data *training* dan data *testing* tidak terlalu jauh, selisih 0,042. Hal ini menunjukkan jika model tersebut sudah baik karena

digunakan pada dataset *training* dan *testing* menghasilkan nilai yang tinggi serta tidak terlalu berbeda jauh sehingga tidak *overfitting* maupun *underfitting.*

Fungsi *predict_model()* juga akan menampilkan *prediction_label* (pada gambar, akan ditampilkan dalam 'Label') dan *prediction_score* (pada gambar, akan ditampilkan dalam '*Score'*) seperti pada gambar berikut :

tahun_berakhir_jabatan_2018	tahun_berakhir_jabatan_2019	tahun_berakhir_jabatan_2020	tahun_berakhir_jabatan_2021	Label	Label	Score
0.0	0.0	0.0	0.0	Label A	Label A	1.0000
0.0	0.0	1.0	0.0	Label B	Label B	0.9998
0.0	0.0	0.0	0.0	Label A	Label A	0.9998
0.0	0.0	0.0	0.0	Label A	Label A	1.0000
0.0	0.0	0.0	0.0	Label A	Label A	1.0000
...	...	...	...	...	...	...
0.0	0.0	0.0	0.0	Label A	Label A	1.0000
0.0	0.0	0.0	0.0	Label A	Label A	0.9998
1.0	0.0	0.0	0.0	Label B	Label B	0.9405
0.0	0.0	0.0	0.0	Label A	Label A	0.9989
0.0	0.0	0.0	0.0	Label A	Label A	1.0000

Gbr. 13 *Preview* Hasil Prediksi Dataset

Dari jumlah keseluruhan data *testing* dalam dataset kenaikan jabatan, fungsi ini hanya menampilkan 5 data paling atas dan paling bawah sebagai *preview.* Hasilnya adalah banyak data yang diprediksi masuk ke label yang benar, serta *score* yang dihasilkan juga di atas 0,9. Selain itu, fungsi ini juga dapat bekerja untuk memprediksi label pada dataset yang belum pernah terlihat (*unseen* dataset) maksudnya adalah dataset yang bukan merupakan *training* maupun *testing. Unseen* dataset didapat dari data awal kemudian dilakukan *drop* atau menghapus kolom 'label'. Sehingga kita bisa melihat dataset awal beserta hasil labelling dan juga skor.

```
[28]: # copy data_akhir dan drop kolom 'Label'

      data_baru = data_akhir.copy()
      data_baru.drop('Label', axis=1, inplace=True)
      data_baru.head()
```

[28]:	mulasi_ak_pleno	akumulasi_unsur0	akumulasi_unsur1	akumulasi_unsur2	akumulasi_unsur3	akumulasi_unsur4	akumulasi_unsur5	akumulasi_unsur6	akumulasi_unsur7
	8.447	13	0	0	0	2	5	0	0
	31.984	3	1	0	0	5	6	3	0
	36.225	79	317	0	0	11	20	0	0
	53.587	13	186	0	0	7	84	0	0
	100.000	0	0	0	0	1	1	0	0

Gbr. 14 *Drop* kolom Label

Gambar diatas merupakan dataset yang sudah dilakukan *drop* pada kolom 'label', pada kolom paling kanan adalah 'akumulasi_unsur7' bukan kolom 'label'. Data tersebut disimpan dalam variabel baru bernama 'data_baru'. Kemudian dilakukan prediksi dengan memanggil variabel model dan juga variabel dataset yang baru.

```
# predict model pada data_baru / unseen dataset

prediction_baru = predict_model(model1_lda, data = data_baru)
prediction_baru
```

akumulasi_unsur0	akumulasi_unsur1	akumulasi_unsur2	akumulasi_unsur3	akumulasi_unsur4	akumulasi_unsur5	akumulasi_unsur6	akumulasi_unsur7	Label	Score
13	0	0	0	2	5	0	0	Label B	1.0000
3	1	0	0	5	6	3	0	Label A	1.0000
79	317	0	0	11	20	0	0	Label B	1.0000
13	186	0	0	7	84	0	0	Label B	1.0000
0	0	0	0	1	1	0	0	Label A	1.0000

Gbr. 15 Hasil prediksi *Unseen* Dataset Jabatan

F. *Deployment*

Deployment adalah tahap penyimpanan file model terbaik yang sudah dilakukan analisis dan evaluasi pada tahapan sebelumnya. Pada tahapan ini menggunakan fungsi save_model(). Fungsi ini menyimpan pipa transformasi atau transformation pipeline dan objek model terlatih (train model) kedalam suatu file berformat pickle (.pkl) Kemudian untuk memuat file model yang sudah disimpan di masa mendatang untuk digunakan pada dataset baru, menggunakan fungsi load_model() Apabila model ingin dimuat untuk memprediksi dataset lainnya, hanya menjalankan fungsi predict_model() dengan menambahkan variabel yang menyimpan load_model nya dan datasetnya.

```
[17]: import pycaret
      import pandas as pd

[18]: data = pd.read_csv('[PANGKAT]_Percobaan.csv')
      data

[18]:
```

	r_nama	r_pangkat	tahun_mulai_pangkat	tahun_berakhir_pangkat	selisih_pangkat	akumulasi_vol	akumulasi_vol_pleno	akumulasi_ak	akumulasi_ak_pleno
	Putri Ariyanto	S.E.	Penata Muda Tingkat I	2016	2021	5	195	113	27.029	21.518

Gbr. 16 Percobaan dengan Data Baru

```
[19]: # Load the model
      from pycaret.classification import load_model
      loaded_model = load_model('PANGKAT8_model1_lda')
      # generate predictions / inference
      from pycaret.classification import predict_model
      pred = predict_model(loaded_model, data=data) # new data
      pred

      Transformation Pipeline and Model Successfully Loaded

[19]:
```

o	akumulasi_unsur0	akumulasi_unsur1	akumulasi_unsur2	akumulasi_unsur3	akumulasi_unsur4	akumulasi_unsur5	akumulasi_unsur6	akumulasi_unsur7	Label	Score
	8	2	9	0	0	6	7	1	0	Label B	1.0

Gbr. 17 Hasil Prediksi Data Baru

IV. PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian prediksi kenaikan jabatan pranata komputer pada Kementerian X dengan menggunakan model algoritma klasifikasi *Linear Discriminant Analysis* (LDA) menggunakan metode Cross Industry Standard Process for Data Mining (CRISP-DM) dan tools JupyterLab yang telah dilakukan, didapatkan kesimpulan sebagai berikut :

1. Proses memprediksi lama kenaikan jabatan pranata komputer pada Kementerian X menggunakan metode CRISP-DM pada tahapan *Business Understanding*, *Data Understanding*, *Data Preparation* / *Data Preprocessing* dan *Modelling*. Pada tahapan *Business Understanding*, *Data Understanding*, *Data Preparation* / *Data Preprocessing* menghasilkan dataset kenaikan jabatan sejumlah 269 baris 18 kolom dari dataset awal yang berjumlah 35.463 baris dan 22 kolom. Pada Tahapan *Modelling* 18 kolom dimasukkan dengan mengeluarkan 2 kolom serta 1 kolom sebagai target sehingga hanya menggunakan 15 kolom sebagai variabel independen.

2. Model Linear Discriminant Analysis *(LDA)* sangat baik untuk digunakan dalam proses prediksi kenaikan jabatan pranata komputer di Kementerian X ditinjau dari *F1 Score* yang dihasilkan dan hasil plot model yang baik. Nilai F1 Score training adalah 0,9345 sedangkan nilai F1 Score testing adalah 0,9765. Dikarenakan antara hasil training dan testing tidak memiliki *gap* yang terlalu besar sehingga model dikatakan baik, tidak *overfitting* maupun *underfitting*.

B. Saran

Saran yang dapat diberikan sebagai bahan pertimbangan untuk perbaikan dan pengembangan penelitian topik pemilihan model klasifikasi adalah sebagai berikut :

1. Melakukan eksplorasi lebih lanjut terhadap parameter yang ada pada bagian *SetUp Environment* agar menghasilkan nilai *f1 score* yang lebih maksimal.
2. Menggunakan metode metode untuk melakukan penanganan terhadap permasalahan data tidak seimbang, bisa menggunakan *Undersampling* maupun *Oversampling*.
3. Disarankan untuk melakukan percobaan dengan lebih banyak variasi *fold* guna mengetahui apakah pada *fold* yang berbeda-beda, model tersebut memiliki konsistensi yang baik dalam menghasilkan nilai evaluasi metrik yang tinggi serta dapat menemukan *fold* berapa yang menghasilkan nilai evaluasi metrik tertinggi.

REFERENSI

[1] Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Nomor 17 Tahun 2021 tentang Penyetaraan Jabatan Administrasi ke dalam Jabatan Fungsional.

[2] Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Nomor 1 Tahun 2023 Pasal 1 Ayat 22 tentang Jabatan Fungsional, menjelaskan mengenai Angka Kredit.

[3] Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi Republik Indonesia Nomor 32 Tahun 2020 tentang Jabatan Fungsional Pranata Komputer.

[4] Mauritsius, Tuga dan Binsar, Faisal. (2020). *Cross Industry Standard Process for Data Mining*. [Online], (https://mmsi.binus .ac.id/2020/09/18/cross-industry-standard-process-for-data-mining-crisp-dm/), tanggal akses 17 Januari 2023

[5] Kade, Ida. (2022). *CRIPS-DM sebagai salah satu standard untuk menghasilkan Data Driven Decision Making yang Berkualitas.* [Online]. (https://www.djkn.kemenkeu.go.id/artikel/baca/15134/CRISP-DM-Sebagai-Salah-Satu-Standard-untuk-Menghasilkan-Data-Driven-Decision-Making-yang-Berkualitas.html), tanggal akses 26 Juni 2023