

# RAMALAN PENYELENGGARAAN SISTEM TRANSIT ALIRAN RINGAN MENGUNAKAN TEKNIK PEMBELAJARAN MESIN

Aini Nabila Mohd Azizi & Zalinda Othman

*Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 UKM Bangi,  
Selangor Darul Ehsan, Malaysia*

## Abstrak

Sepanjang lima puluh tahun yang lalu, sistem penyelenggaraan pengangkutan awam telah mendapat perhatian pada skala global, terutamanya dalam transit aliran ringan (LRT). Baru-baru ini, pelbagai insiden mengejut berlaku akibat prestasi LRT yang kurang memuaskan menyebabkan kekurangan LRT yang beroperasi. Maka, fokus penyelidikan adalah mencari penyelesaian kepada masalah akibat kaedah penyelenggaraan pengangkutan seperti penjadualan penyelenggaraan yang tidak efisien dan tempoh penyelenggaraan yang terlalu lama. Bagi menaiktaraf infrastruktur pengangkutan ini, kajian mencadangkan penyelesaian dengan menganalisis data dan membuat ramalan mengenai prestasi LRT dengan menggunakan teknik pembelajaran mesin. Kaedah ini dapat membantu dalam meramalkan dan merekodkan prestasi LRT dengan mudah. Kajian ini tertumpu kepada Transit Aliran Ringan (LRT) Malaysia khususnya yang beroperasi di laluan Ampang, menggunakan data sejarah yang dibekalkan oleh Prasarana Malaysia Berhad. Dua faktor yang difokuskan adalah jumlah jarak perjalanan dan jumlah masa pemampat udara digunakan. Ramalan dibuat menggunakan teknik pembelajaran mesin K-Jiran Terdekat (KNN), Pohon Keputusan dan Rangkaian Neural Buatan (ANN). Hasil akhir kajian ini adalah penghasilan sebuah model ramalan yang dapat meramal prestasi LRT. Pada akhir kajian ini, didapati model Rangkaian Neural Buatan menghasilkan ramalan yang paling tepat dengan 94% ketepatan berbanding model yang lain. Selain itu, kajian ini juga menghasilkan sebuah antara muka bagi memaparkan ramalan prestasi LRT dan mevisualisasikan graf trend data berkaitan dengan keadaan semasa LRT. Kesimpulannya, penemuan ini boleh memberi manfaat kepada pekerja dan pengguna kerana memudahkan peramalan prestasi LRT sekali gus membantu menghasilkan penjadualan penyelenggaraan yang lebih efisien.

**Kata kunci:** prestasi LRT, penyelenggaraan, teknik pembelajaran mesin

## Pengenalan

Pengangkutan kereta api bermaksud kenderaan yang berjalan di landasan. Ia merupakan salah satu mod ulang-alik penting dan pengangkutan yang baik untuk jarak jauh atau dekat. Kereta api adalah kaedah pengangkutan darat tertua di seluruh dunia yang bermula dengan membolehkan pengangkutan bot ke darat. Pada asalnya, objektif tunggal pengangkutan rel adalah penghantaran barangan perindustrian seperti arang batu dari pelabuhan dan gudang. Evolusi teknologi dan tamadun menjadikan pengangkutan rel lebih maju dari semasa ke semasa seperti perubahan penggunaan buruh manual kepada kuasa kuda, rel kayu kepada logam, dan enjin wap kepada elektrik (Lewis, 2001). Akhirnya, matlamat pengangkutan berubah dan lebih mementingkan kemudahan pergerakan manusia ke destinasi pilihan mereka menjadikan kereta api elemen penting yang diperlukan oleh industri (Spiryagin et. al. 2016). Hasilnya, perkembangan kereta api menyumbang kepada peningkatan dalam sistem transit awam, memberi pelbagai manfaat kepada orang ramai.

Di Malaysia, kereta, motosikal, bas, kereta api dan perjalanan udara komersial merupakan pengangkutan utama yang disediakan. Dengan kemajuan teknologi di Malaysia, penambahbaikan teknologi transit membolehkan perjalanan yang pantas dengan harga yang berpatutan. Penubuhan landasan atau rel pertama di Malaysia bermula kerana keperluan untuk mengangkut bijih timah dari lombong di kawasan pedalaman negeri-negeri Pantai Barat Semenanjung Malaysia ke pelabuhan pantai. Setiap sistem terdapat risiko dan penyelenggaraan yang perlu dilakukan. Dalam sektor rel, analisis risiko telah dilakukan sejak tahun 1941. Namun sehingga ketika ini, analisis risiko hanya tertumpu kepada pelanggaran yang melibatkan peranti amaran, ciri pengesanan sebagai faktor atau kriteria di persimpangan kereta api (Ismail 2016). Kurang penyelidikan yang dilaksanakan mengenai prestasi transit rel berdasarkan keadaan semasa transit rel tersebut, terutamanya di negara Asia Tenggara. Akibatnya, Malaysia tiada sumber atau proses yang boleh dirujuk untuk meramalkan kemalangan akibat prestasi transit rel. Maka, kajian ini bertujuan mengkaji dan meramal prestasi transit rel di Malaysia, berfungsi sebagai pangkalan data untuk ramalan keadaan LRT lebih awal bagi

menambah baik penyelenggaraan pengangkutan. Kajian ini perlu dilakukan bagi memudahkan pekerja mengenal pasti prestasi LRT, membantu dalam membuat keputusan bagi penyelenggaraan sesuah LRT seterusnya menyediakan LRT yang yang baik bagi kegunaan pengguna.

Peningkatan dalam jadual penyelenggaraan yang tidak konsisten telah memberi kesan negatif kepada pengguna. Ini merupakan punca utama kekurangan LRT yang beroperasi dan keadaan pengangkutan yang sentiasa kurang memuaskan. Akhirnya, ia telah mengganggu perjalanan pengguna beberapa kali sepanjang tahun lalu. Melalui kajian ini, ia didapati bahawa masalah ini berpunca akibat pekerja sukar mengenal pasti prestasi LRT berdasarkan keadaan semasa LRT. Pekerja seharusnya sentiasa peka ketika membuat pemeriksaan keselamatan agar dapat menyediakan LRT yang selamat. Namun akibat pemeriksaan yang dilakukan secara manual, proses mengenal pasti prestasi LRT terdedah kepada kesilapan manusia dan kemungkinan pekerja membenarkan LRT dalam keadaan kurang memuaskan beroperasi adalah tinggi. Selain itu, pekerja tidak mempunyai kemahiran dalam menganalisis data seperti trend kegagalan transit rel atau membuat ramalan prestasi LRT berdasarkan keadaan semasa LRT menghadkan cara meramal secara automatik.

Objektif kajian ini adalah untuk mereka bentuk dan membina model ramalan untuk meramal prestasi sebuah LRT menggunakan teknik pembelajaran mesin berdasarkan keadaan semasa LRT. Selain itu, kajian ini juga bertujuan mereka bentuk dan membangunkan sebuah antara muka menggunakan platform web ringkas bagi memudahkan penggunaan model ramalan tersebut. Kajian ini menggunakan data daripada sistem LRT PRASARANA di sepanjang laluan Ampang sahaja dan tertumpu kepada dua parameter iaitu jumlah jarak perjalanan dalam kilometer dan jumlah masa pemampat udara digunakan dalam minit sebagai faktor pemboleh ubah tidak bersandar. Kajian ini menggunakan teknik perlombongan data CRISP-DM (Cross Industry Standard Process for Data Mining) sebagai panduan membina model ramalan dan menggunakan teknik pembelajaran mesin K-Jiran Terdekat (KNN), Pohon Keputusan dan Rangkaian Neural Buatan (ANN). Melalui laporan ini, metodologi yang dilakukan akan diterangkan dengan lebih terperinci, hasil kajian serta perbandingan

model akan dilakukan, kekuatan, kekangan dan penambahbaikan yang boleh dilakukan dalam kajian ini akan dijelaskan.

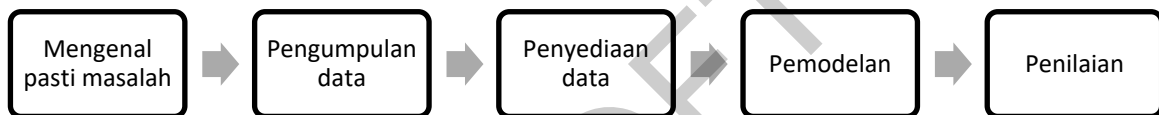
Perbandingan dengan kajian lepas dilakukan bagi mengenal pasti persamaan dan perbezaan penyelesaian yang dicadangkan dengan penyelesaian sedia ada. Ini juga adalah bagi menjelaskan kelebihan penyelesaian yang dicadangkan. Berdasarkan penyelidikan lepas, terdapat beberapa kajian yang hampei sama dengan penyelesaian yang. Penyelidikan tersebut bertajuk 'Developing Machine Learning-Based Models for Railway Inspection' oleh Chunsheng Yan dan 'Prediction of Track Deterioration Using Maintenance Data and Machine Learning Schemes' oleh Jun S. Lee. Kedua-dua kajian ini membangunkan model ramalan berkaitan dengan penyelenggaraan kereta api atau komponen berkaitan dengan transit rel. Persamaan antara kajian oleh Chunsheng Yang adalah model ramalan yang dibina menghasilkan keputusan multikelas label dan menggunakan model ramalan pohon keputusan. Manakala perbezaan adalah dalam kajian ini beberapa model lain juga dibina seperti KNN dan ANN bagi mencari model yang paling baik. Kelebihannya, kajian ini berpeluang mencari model yang lebih baik apabila membandingkan beberapa model yang berlainan.

Seterusnya adalah persamaan bagi penyelidikan yang dihasilkan oleh Jun S. Lee dengan kajian ini adalah penggunaan model ramalan yang sama iaitu ANN dan domain masalah yang sama iaitu masalah berkenaan penyelenggaraan transit rel. Manakala perbezaan mereka adalah kajian ini mempunyai pemboleh ubah sasaran berbentuk multikelas label atau klasifikasi walhal kajian Jun S. Lee berada dalam format nombor berterusan. Maka, penilaian prestasi bagi kedua-dua penyelidikan adalah berbeza kerana keputusan yang ingin dicapai berada dalam dua format yang berlainan. Kelebihan kajian ini adalah ia dapat membina beberapa model ramalan yang lain supaya berpeluang mencari model yang paling baik.

### **Metodologi Kajian**

Metodologi yang digunakan ialah kaedah perlombongan data CRISP-DM, iaitu proses standard rentas industri untuk perlombongan data. CRISP-DM terbahagi kepada lima fasa. Fasa pertama merupakan

fasa untuk mengenal pasti masalah, menentukan matlamat dan objektif sepanjang kajian dilakukan. Seterusnya, adalah pemahaman data serta pengumpulan data. Kemudian dalam fasa ketiga penyediaan data dilakukan untuk membersihkan dan menyeragamkan format data. Fasa keempat merupakan fasa pemodelan di mana model ramalan KNN, pohon keputusan dan ANN dibina. Akhirnya adalah fasa lima iaitu penilaian. Model yang telah dipilih akan dinilai dan dibandingkan antara satu sama lain untuk dianalisis. Model yang menghasilkan keputusan yang paling tepat dan konsisten adalah model ramalan yang paling sesuai untuk kajian ini. Proses ini paling sesuai digunakan bagi kajian yang melibatkan ramalan dan teknik pembelajaran mesin, sama seperti objektif yang dinyatakan.



Rajah 1: Carta aliran proses metodologi kajian

## 2.1 Mengenal pasti masalah

Penjadualan yang tidak konsisten dalam penyelenggaraan LRT telah memberikan kesan negatif kepada pengguna pengangkutan rel. Kesannya, LRT sering berhenti secara tiba-tiba semasa beroperasi dan penumpang terpaksa mencari alternatif lain untuk menuju ke destinasi masing-masing. Masalah teknikal ini berpunca daripada pekerja sukar mengenal pasti prestasi LRT. Maka, pekerja seharusnya membuat pemeriksaan setiap hari secara manual bagi memastikan setiap LRT dalam keadaan dan prestasi yang baik. Aktiviti ini memerlukan tenaga kerja yang tinggi dan masa yang panjang. Kesannya, pekerja dengan mudah dapat melakukan kesilapan dalam menentukan prestasi LRT. Selain itu, pekerja juga tidak mempunyai kemahiran untuk menganalisis data keadaan semasa LRT. Maka, dengan menggunakan model ramalan pembelajaran mesin, pekerja dengan mudah dapat meramalkan prestasi semasa LRT dan merancang jadual penyelenggaraan yang bersesuaian secara efisien.

## 2.2 Pengumpulan Data

Set data yang digunakan dalam kajian ini merupakan data keadaan semasa LRT yang telah di kumpul oleh pihak PRASARANA. Data ini adalah sah kerana PRASARANA merupakan syarikat yang mengendalikan LRT di laluan Ampang. Set data ini hadir dalam dua bentuk, tulisan tangan dan fail excel. Selain itu, data ini juga telah dikumpulkan daripada dua sumber yang berbeza kerana pihak PRASARANA tidak mempunyai sistem berpusat untuk pengumpulan data LRT. Data ini merupakan data keadaan LRT secara bulanan dari Januari 2022 hingga Disember 2022. Set data ini mempunyai lima atribut, iaitu jumlah jarak perjalanan (kilometer), jumlah masa pemampat udara digunakan di DA1 (minit), jumlah masa pemampat udara digunakan di DA2 (minit), jumlah keseluruhan masa pemampat udara digunakan di DA1 dan D2 (minit) dan purata masa pemampat udara digunakan dalam satu kilometer. Dalam kajian ini, hanya tiga atribut yang digunakan untuk meramal prestasi LRT iaitu jumlah jarak perjalanan, jumlah masa pemampat udara digunakan di DA1 (minit) dan jumlah masa pemampat udara digunakan di DA2 (minit). Rajah di bawah menunjukkan 5 baris pertama set data pada bulan Januari 2022.

	Kilometer Reading	DA1 Air Compressor Reading	DA2 Air Compressor Reading	Min per kilometer
0	5063	3823	2788	1.3057
1	3540	5381	5212	2.9924
2	7600	12773	12477	3.3224
3	8022	6530	7490	1.7477
4	7598	6196	5768	1.5746

Rajah 2: Lima rekod pertama set data mentah

## 2.3 Penyediaan Data

### 2.3.1 Penyatuan Data

Terdapat 12 fail berasingan diberikan oleh pihak PRASARANA dan penyatuan data dilakukan bagi mengumpul data daripada fail berbeza menjadi satu fail supaya mudah untuk dikendalikan. Penyatuan data tersebut dijalankan dengan mengimport fail-fail tersebut dan menggabungkan data mengikut urutan bulan. Rajah dibawah merupakan contoh data selepas disatukan ke dalam satu fail.

	A	B	C	D
1	Kilometer Reading	DA1 Air Compressor Reading	DA2 Air Compressor Reading	Min per kilometer
2	5063	3823	2788	1.3057
3	3540	5381	5212	2.9924
4	7600	12773	12477	3.3224
5	8022	6530	7490	1.7477
6	7598	6196	5768	1.5746
7	9098	5267	9167	1.5865
8	5156	4884	5459	2.006
9	7815	4253	4819	1.1608
10	9182	7403	8309	1.7112
11	9167	11517	7336	2.0566
12	5784	3010	2802	1.0048
13	4687	4268	5426	2.0683
14	4698	2638	3210	1.2448
15	8496	5111	6262	1.3386
16	7339	5247	4576	1.3385
17	1605	731	706	0.8953
18	851	1310	546	2.181
19	6464	3676	3278	1.0758
20	4358	2467	1696	0.9553

Rajah 3: Contoh rekod set data yang telah disatukan ke dalam satu fail

### 2.3.2 Pembersihan Data

Rajah 4 menunjukkan terdapat beberapa nilai yang hilang sepanjang tahun 2022. Oleh itu, pembersihan data dijalankan untuk menggantikan nilai tersebut. Nilai purata dipilih kerana ia merupakan nilai yang paling sesuai untuk menggantikan data tersebut. Hal ini kerana setiap LRT mempunyai jumlah jarak perjalanan dan penggunaan yang lebih kurang sama akibat menggunakan laluan yang sama iaitu laluan LRT Ampang serta penggunaan pemampat udara lebih kurang sama pada setiap bulan. Rajah di bawah merupakan cara penggantian data yang hilang dengan menggunakan purata bagi setiap atribut.

```
count = (df['kilometer Reading'] == 0).sum()
print('Count of zeros in kilometer reading: ', count)
count = (df['DA1 Air Compressor Reading'] == 0).sum()
print('Count of zeros in DA1 Air Compressor Reading: ', count)
count = (df['DA2 Air Compressor Reading'] == 0).sum()
print('Count of zeros in DA2 Air Compressor Reading: ', count)
count = (df['Min per kilometer'] == 0).sum()
print('Count of zeros in Min per kilometer: ', count)

Count of zeros in kilometer reading: 11
Count of zeros in DA1 Air Compressor Reading: 11
Count of zeros in DA2 Air Compressor Reading: 11
Count of zeros in Min per kilometer: 0
```

Rajah 4: Kod Python yang menunjukkan bilangan data yang hilang

```
df['kilometer Reading']=df['kilometer Reading'].replace(0,df['kilometer Reading'].mean())
df.isna().any()
```

```
kilometer Reading      False
DA1 Air Compressor Reading  False
DA2 Air Compressor Reading  False
Min per kilometer      False
dtype: bool
```

Rajah 5: Kod Python yang menyemak keberadaan data yang hilang

Setelah menggantikan data yang hilang, satu atribut baru dibina bagi mengira prestasi LRT yang dinamakan Purata Bacaan Pemampat Udara dalam Satu Kilometer (minit per kilometer). Atribut ini adalah hasil tambah bacaan masa pemampat udara digunakan dan dibahagi dengan jumlah jarak perjalanan LRT tersebut. Lebih tinggi bacaan semasa pemampat udara digunakan dengan jumlah jarak yang rendah menandakan prestasi LRT semakin menurun. Rajah di bawah merupakan contoh pembinaan atribut yang dilakukan dalam kajian ini.

```
df['Purata Bacaan Pemampat Udara per Kilometer'] = (df['DA1 Air Compressor Reading']
                                                    +df['DA2 Air Compressor Reading'])/df[' Kilometer Reading']
new_df = df.copy()
new_column = new_df.pop('Purata Bacaan Pemampat Udara per Kilometer')
new_df.insert(3, 'Purata Bacaan Pemampat Udara per Kilometer', new_column)
```

Rajah 6: Kod Python yang menunjukkan cara membina parameter baru

Satu lagi atribut bernama 'Class' dibina bagi mengelaskan purata bacaan pemampat udara dalam satu kilometer kepada tiga kategori iaitu 'Good', 'Neutral' dan 'Bad'. Bagi pembinaan atribut ini, kaedah 'binning' dilakukan dan dilabelkan berdasarkan nilai purata bacaan pemampat udara dalam satu kilometer. Atribut ini merupakan kelas prestasi LRT dimana nilai purata bacaan pemampat udara dalam satu kilometer kurang daripada 1 ialah prestasi 'Good', nilai 1 hingga 2 ialah prestasi 'Neutral' dan nilai lebih daripada 2 ialah prestasi 'Bad'. Rajah di bawah ialah contoh 'binning' yang dilakukan untuk membina atribut 'Class'. Dalam kajian ini, atribut yang digunakan bagi membuat ramalan adalah jumlah jarak perjalanan, bacaan pemampat udara digunakan di DA1 dan DA2 serta atribut 'Class'.

```
new_df.loc[new_df['Purata Bacaan Pemampat Udara per Kilometer'].between(1, 2, 'both'), 'Class'] = 'NEUTRAL'
new_df.loc[new_df['Purata Bacaan Pemampat Udara per Kilometer'].between(0, 1, 'left'), 'Class'] = 'GOOD'
new_df.loc[new_df['Purata Bacaan Pemampat Udara per Kilometer'].between(2, 10, 'right'), 'Class'] = 'BAD'
```

Rajah 7: Kod Python yang mengkategorikan prestasi LRT

### 2.3.3 Transformasi Data

Seterusnya data akan dinormalisasikan menggunakan normalisasi standard bagi membantu menyesuaikan skala untuk kesemua atribut. Standard skala digunakan dalam kajian ini kerana terdapat perbezaan skala yang ketara antara atribut yang digunakan. Dengan menukarkan data kepada skala



yang seragam bagi semua atribut, model ramalan pembelajaran mesin dapat menjadi lebih efisien dan menghasilkan ramalan yang lebih tepat. Rajah di bawah merupakan contoh normalisasi standard yang digunakan untuk data dalam kajian ini.

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Rajah 8: Kaedah penomelan data yang digunakan

## 2.4 Pemodelan

Sebelum model dibina, langkah asas yang dilakukan oleh setiap model adalah mengimport beberapa perpustakaan atau modul penting bagi memproses, membahagikan data serta membangunkan model ramalan tersebut. Rajah 9 menunjukkan contoh kod Python semasa mengimport perpustakaan dan modul yang diperlukan.

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
```

Rajah 9: Contoh perpustakaan dan modul yang diimport

Kemudian, set data dibahagikan kepada dua rangkaian utama di mana X ialah atribut input dan y ialah pemboleh ubah sasaran iaitu kelas prestasi LRT. Rajah di bawah menunjukkan kod Python yang digunakan untuk membahagikan set data kepada X dan y.

```
X = new_df.iloc[:, [0, 1, 2]].values
y = new_df.iloc[:, -1].values
```

Rajah 10: Cara membahagikan set data kepada X dan y

Selepas proses ini, setiap model mempunyai proses berlainan kerana data perlu disesuaikan mengikut model supaya ramalan yang dibuat adalah tepat.

### 2.4.1 K-Jiran Terdekat (KNN)

KNN merupakan algoritma model ramalan yang mudah digunakan dan mempunyai kecekapan yang tinggi dalam menjalankan tugas pengelasan set data, sesuai dengan data yang digunakan dalam

kajian ini. Langkah selepas pembahagian data kepada X dan y adalah proses penukaran format atribut y kepada kategori atau nombor. Bagi memudahkan pemprosesan teks, LabelEncoder digunakan untuk mengkodkan atribut label kelas prestasi LRT kepada nilai nombor. Penukaran ini adalah untuk memastikan data senang untuk difahami dan diproses oleh model ramalan KNN. Rajah di bawah ialah kod Python yang digunakan bagi penukaran tersebut.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y[:,0] = le.fit_transform(y[:,0])
```

Rajah 11: Contoh kod Python untuk menukarkan format data menggunakan

#### LabelEncoder

Set data kemudian dibahagikan kepada set latihan dan set ujian menggunakan fungsi `train_test_split` daripada `sklearn.model_selection` modul. Sengan 80% data untuk latihan dan 20% untuk ujian. Set latihan digunakan untuk melatih pengelasan KNN, manakala set ujian digunakan untuk menilai ketepatan dalam meramal prestasi pengelas. Rajah di bawah menunjukkan cara pembahagian set data.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 10)
```

Rajah 12: Kod Python membahagikan set data kepada set data latihan dan ujian

Langkah seterusnya ialah pembinaan model KNN bagi melihat ketepatan dan kesesuaian dalam penukaran nilai parameter. Dalam kajian ini, `n_neighbors = 1` digunakan, supaya setiap contoh ujian akan diklasifikasikan berdasarkan label jiran yang paling dekat dalam set latihan. Kemudian, ia dilatih pada set data latihan, melibatkan pencarian jiran terdekat untuk setiap contoh dalam set latihan dan menyimpan maklumat yang berkaitan untuk membuat ramalan menggunakan data baharu. Ramalan kemudiannya dibuat pada set ujian dan pengelasan yang dilakukan mengaitkan label kelas kepada setiap maklumat berdasarkan jiran terdekat yang ditemui semasa latihan. Rajah di bawah menunjukkan cara model ramalan dibina dan dilatih.

```

classifier = KNeighborsClassifier(n_neighbors = 1, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)

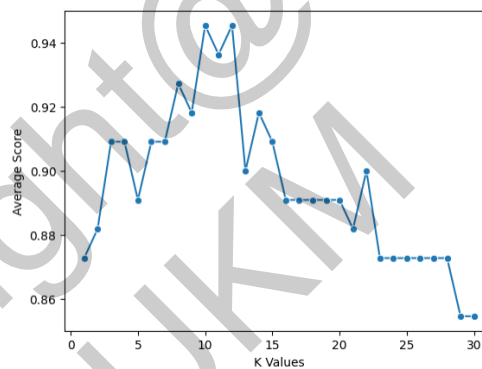
# Predict the labels for the training set
y_train_pred = classifier.predict(X_train)

# Predict the labels for the test set
y_test_pred = classifier.predict(X_test)

```

Rajah 13: Kod Python membina model ramalan KNN

Walau bagaimanapun, pendekatan ini lebih sensitif terhadap data yang bising dan boleh mengurangkan ketepatan model dalam membuat ramalan. ‘Hyperparameter tuning’ dilakukan bagi meningkatkan ketepatan model. Didapati dalam kajian ini,  $n\_neighbor = 10$  memberikan keputusan ketepatan tertinggi berbanding dengan yang lain. Rajah di bawah menunjukkan kod Python yang digunakan untuk membandingkan ketepatan model bagi setiap nilai jiran dan graf bagi melihat tren ketepatan apabila nilai jiran bertambah.



Rajah 14: Graf garis nilai ketepatan berdasarkan nilai k

#### 2.4.2 Pohon Keputusan

Pohon keputusan ialah antara contoh model ramalan pembelajaran mesin yang digunakan bagi tugas pengelasan, sesuai dengan data kajian ini. Proses seterusnya ialah mengkategorikan nilai bagi atribut input kepada 3. Ia dibahagikan berdasarkan persentil dan diberi label ‘High’, ‘Normal’ atau ‘Low’. Pengelasan mengikut kategori dilakukan kerana pohon keputusan menghasilkan ketepatan model yang lebih tinggi apabila data berada dalam bentuk kelas berbanding dengan nilai nombor yang berterusan. Rajah di bawah menunjukkan cara pengelasan dilakukan dalam kajian ini.

```

new_df[' Kilometer Reading'] = pd.qcut(new_df[' Kilometer Reading'],
                                       q=3,
                                       labels=['Low', 'Normal', 'High'])
new_df['DA1 Air Compressor Reading'] = pd.qcut(new_df['DA1 Air Compressor Reading'],
                                               q=3,
                                               labels=['Low', 'Normal', 'High'])
new_df['DA2 Air Compressor Reading'] = pd.qcut(new_df['DA2 Air Compressor Reading'],
                                               q=3,
                                               labels=['Low', 'Normal', 'High'])

```

Rajah 15: Kod Python yang membahagikan set data kepada 3 bin

```

d = {'Low': 0, 'Normal': 1, 'High': 2}
new_df[' Kilometer Reading'] = new_df[' Kilometer Reading'].map(d)
d = {'Low': 0, 'Normal': 1, 'High': 2}
new_df['DA1 Air Compressor Reading'] = new_df['DA1 Air Compressor Reading'].map(d)
d = {'Low': 0, 'Normal': 1, 'High': 2}
new_df['DA2 Air Compressor Reading'] = new_df['DA2 Air Compressor Reading'].map(d)

```

Rajah 16: Kod Python yang menukarkan label bin kepada nombor

Sama seperti dalam pembinaan model ramalan KNN, set data kemudian dibahagikan kepada set latihan dan set ujian menggunakan fungsi `train_test_split` daripada `sklearn.model_selection` modul dengan 80% data untuk latihan dan 20% untuk ujian. Rajah di bawah menunjukkan cara pembahagian set data.

```

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 101) # 70% training and 30% test

```

Rajah 17: Kod Python membahagikan set data kepada set data latihan dan ujian

Seterusnya, model pohon keputusan dibina menggunakan `DecisionTreeClassifier` di mana model ini akan menjana pohon yang paling optimum serta ketepatan yang paling tinggi. Kemudian model dilatih pada set data latihan di mana proses latihan melibatkan pencarian laluan pohon yang sesuai bagi setiap label kelas dan menyimpan maklumat yang berkaitan untuk kegunaan ramalan masa depan. Ramalan kemudiannya di buat pada set ujian dan pengelasan yang dilakukan mengaitkan label kelas kepada setiap contoh semasa melatih model serta berdasarkan maklumat yang ditemui semasa latihan. Rajah di bawah menunjukkan cara model ramalan dibina dan dilatih.

```

# Create Decision Tree classifier object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifier
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

# Predict the labels for the training set
y_train_pred = clf.predict(X_train)

# Predict the labels for the test set
y_test_pred = clf.predict(X_test)

```

Rajah 18: Kod Python membina model ramalan pohon keputusan

Setelah itu, 'parameter tuning' dilakukan untuk mencari parameter model yang menghasilkan keputusan yang paling baik. 3 kategori yang dilakukan pada permulaan pembinaan model adalah tidak sesuai dan berkemungkinan tidak dapat menangkap keseluruhan maklumat dalam data. Ini kerana berdasarkan perbezaan skala antara parameter yang besar, penambahan kategori kepada 10 adalah lebih sesuai dan sepadan dengan taburan data. Pengelasan nilai bagi atribut input diberi label seperti 'lowest', 'low-2', 'low-1', 'mid-low', 'mid', 'mid-high', 'high-1', 'high-2', 'high-3', dan 'highest. Selepas itu, proses pembahagian dan melatih model dilakukan semula menggunakan pengelasan atribut input yang baharu. Rajah di bawah menunjukkan kod Python yang digunakan semasa pengelasan data dan melatih model ramalan tersebut.

```

# Bin the feature column into 10 quantile-based bins and assign label names
bin_labels = ['lowest', 'low-2', 'low-1', 'mid-low', 'mid', 'mid-high', 'high-1', 'high-2', 'high-3', 'highest']
df['Kilometer Reading'] = pd.qcut(df['Kilometer Reading'], q=10, labels=bin_labels)
df['DA1 Air Compressor Reading'] = pd.qcut(df['DA1 Air Compressor Reading'], q=10, labels=bin_labels)
df['DA2 Air Compressor Reading'] = pd.qcut(df['DA2 Air Compressor Reading'], q=10, labels=bin_labels)

```

Rajah 19: Kod Python yang mengkategorikan data kepada 10 bahagian

```

d = {'lowest': 0, 'low-2': 1, 'low-1': 2, 'mid-low': 3, 'mid': 4, 'mid-high': 5, 'high-1': 6, 'high-2': 7, 'high-3': 8, 'highest': 9}
df['Kilometer Reading'] = df['Kilometer Reading'].map(d)
df['DA1 Air Compressor Reading'] = df['DA1 Air Compressor Reading'].map(d)
df['DA2 Air Compressor Reading'] = df['DA2 Air Compressor Reading'].map(d)

```

Rajah 20: Kod Python yang menukarkan format label kepada nombor

### 2.4.3 Rangkaian Neural Buatan (ANN)

Rangkaian Neural Buatan (ANN) merupakan salah satu model ramalan pembelajaran mesin yang sangat efisien dalam pengelasan multikelas label. Seperti dalam kajian ini, pemboleh ubah bersandar merupakan multikelas yang mempunyai 3 label iaitu 'Good', 'Neutral' dan 'Bad'. Pemboleh ubah sasaran, yang masih dalam bentuk teks perlu dikodkan kepada nilai angka

menggunakan LabelEncoder. Pemboleh ubah sasaran yang telah dikodkan kemudiannya diubah kepada format kategori menggunakan fungsi `to_categorical`. Penukaran ini dilakukan kerana model ramalan ini hanya menerima data yang berformat kategori. Selepas pemprosesan ini, data diperolehi ialah pemboleh ubah sasaran yang telah diubah menjadi `y_categorical`. Rajah di bawah menunjukkan kod Python yang digunakan untuk mengubah format `y`.

```
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
num_classes = len(label_encoder.classes_)
y_categorical = to_categorical(y_encoded, num_classes=num_classes)
le_name_mapping = dict(zip(label_encoder.classes_, label_encoder.transform(label_encoder.classes_)))
print(le_name_mapping)
{'GOOD': 0, 'NEUTRAL': 1, 'BAD': 2}
```

Rajah 21: Kod Python yang menukarkan format atribut `y` kepada nombor

Sama dengan model ramalan yang lain, proses seterusnya adalah pembahagian set data kepada set latihan dan set ujian menggunakan fungsi `train_test_split` daripada modul `sklearn.model_selection` dengan 80% data untuk latihan dan 20% untuk ujian. Rajah di bawah menunjukkan cara pembahagian set data.

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y_categorical, test_size=0.2, random_state=42)
```

Rajah 22: Kod Python membahagikan set data kepada set data latihan dan ujian

Model ANN dibina menggunakan fungsi `Sequential` daripada modul `Keras`. Dalam model ini, beberapa lapisan `Dense` digunakan, diikuti oleh lapisan keluaran `softmax`. Lapisan-lapisan ini membentuk struktur model ANN yang terdiri daripada beberapa unit atau 'neuron'. Fungsi `relu` digunakan untuk lapisan tersembunyi, manakala `softmax` digunakan pada lapisan luar untuk menghasilkan nilai kemungkinan bagi setiap kelas. Kesemua lapisan yang telah dibina, disatukan menggunakan fungsi `compile`. Dalam kod ini, pengoptimuman dilakukan menggunakan `adam` dan kehilangan (`loss`) dikira menggunakan `categorical_crossentropy`, iaitu cara pengiraan kehilangan yang sesuai untuk masalah pengelasan multikelas. Rajah di bawah menunjukkan struktur model ramalan ANN yang digunakan dalam kajian ini.

```

# Build the model
model = Sequential()
model.add(Dense(64, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dense(64, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

```

Rajah 23: Kod Python untuk membina model ramalan ANN

Model dilatih menggunakan set latihan bersama dengan perubahan parameter lain seperti jumlah epochs dan batch\_size. Epochs merujuk kepada bilangan iterasi melalui set data latihan, manakala batch\_size merujuk kepada bilangan sampel yang digunakan pada setiap pengemaskinian parameter. Setelah model dilatih, model dapat meramalkan kelas prestasi LRT untuk set ujian dan hasil ramalan ini akan memberikan nilai kemungkinan bagi setiap kelas untuk setiap sampel dalam set ujian.

```

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
# Predict on the test set
y_pred = model.predict(X_test)

```

Rajah 24: Kod Python untuk menilai model ramalan ANN

## 2.5 Penilaian

Objektif penilaian ini adalah bagi menilai model ramalan yang mempunyai ketepatan paling tinggi dalam meramal prestasi LRT. Model perlu mempunyai ketepatan meramal yang tinggi supaya tiada kesilapan dalam menentukan kelas prestasi LRT. Sebagai contoh, apabila model ramalan mengelaskan LRT itu sebagai 'Good', walhal keadaan sebenar LRT adalah 'Bad', kemungkinan LRT tersebut mengalami masalah teknikal mengejut adalah tinggi. Antara masalah yang boleh dihadapi adalah LRT berhenti beroperasi secara tiba-tiba atau pengeluaran asap daripada pemampat udara LRT. Kesimpulannya, pengujian model ramalan akan difokuskan kepada ketepatan model, keupayaan model meramal 'Positif Sebenar' serta mengenal pasti dan meramal 'Negatif Sebenar'.

Jenis pengujian yang dilakukan adalah ketepatan untuk menilai keseluruhan model membuat ramalan. Ketepatan adalah metrik utama yang digunakan semasa perbandingan di mana

lebih tinggi ketepatan, lebih bagus model ramalan tersebut. Merujuk kepada kod Python di bawah, langkah mengira ketepatan untuk set data ujian adalah dengan menggunakan alat 'accuracy\_score'. Ketepatan ini menunjukkan sejauh mana model dapat meramalkan kelas dengan betul dan antara metrik yang sering digunakan dalam menilai keberkesanan model klasifikasi.

```
# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

#### Rajah 25: Kod Python bagi mencari ketepatan model

Seterusnya adalah kepersisan iaitu alat yang dapat menguji dan menilai keupayaan model ramalan dalam meramal 'Positif Sebenar'. Metrik ini akan mengira ketepatan model dalam meramal kelas 'Good' dalam atribut prestasi LRT dan lebih tinggi kepersisan, lebih rendah kemungkinan model ramalan tersilap klasifikasi kelas 'Neutral' atau 'Bad' sebagai 'Good'. Ini adalah penting kerana silap klasifikasi prestasi 'Bad' sebagai 'Good' akan mendatangkan masalah teknikal yang tidak dijangka. Merujuk kepada kod Python di bawah, metrik ini menggunakan fungsi 'precision\_score' bagi mengira kepersisan set data ujian.

```
# Precision
precision = precision_score(y_test, y_pred, average='macro')
print("Precision:", precision)
```

#### 26: Kod Python bagi mencari kepersisan model

Selain itu, pengkhususan iaitu alat mengukur kebolehan model ramalan mengenal pasti atribut input yang akan menghasilkan keputusan 'Bad' bagi prestasi LRT. Kelebihan metrik ini adalah ia dapat mengukur keupayaan sesebuah model ramalan dalam mengenal pasti kelas yang negatif dengan betul. Merujuk kepada kod Python bawah, pengkhususan di kira dengan membahagi jumlah 'Negatif Sebenar' dengan jumlah keseluruhan kelas negatif iaitu 'Negatif Sebenar' dan 'Positif Palsu'.



```
# Extract the true negatives (TN), false positives (FP), and false negatives (FN) from the confusion matrix
TN = cm[0, 0] + cm[1, 1] # Sum of true negatives for class 2 and class 3
FP = cm[0, 2] + cm[1, 2] # Sum of false positives for class 2 and class 3
FN = cm[2, 0] + cm[2, 1] # Sum of false negatives for class 1 and class 2

# Calculate specificity
specificity = TN / (TN + FP)
print("Specificity:", specificity)
```

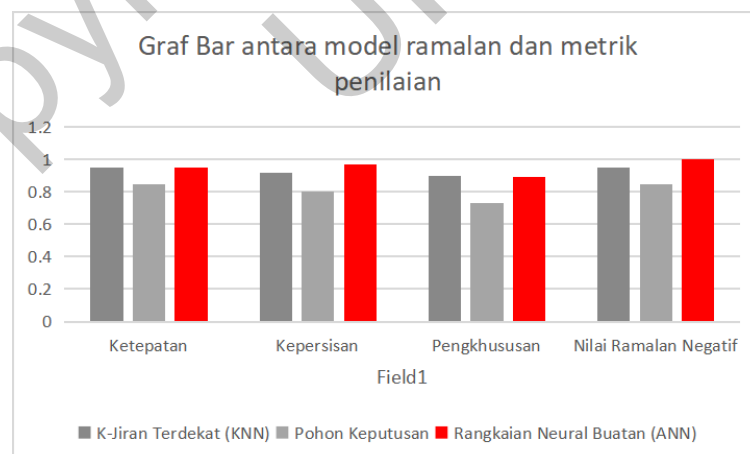
### 27: Kod Python bagi mencari pengkhususan model

Akhir sekali adalah nilai ramalan negatif di mana ia akan mengira kebolehan model untuk meramal 'Negatif Sebenar'. Metrik ini digunakan untuk mengukur sejauh mana model dapat meramalkan secara tepat kelas negatif berdasarkan populasi atau keseluruhan data. Sebagai contoh, dengan nilai ramalan negatif yang tinggi, kemungkinan meramal prestasi 'Bad' sebagai 'Good' atau 'neutral' adalah rendah. Merujuk kepada kod Python di bawah, metrik ini dikira dengan membahagi nilai 'Negatif Sebenar' dengan jumlah nilai 'Negatif Sebenar' dan 'Negatif Palsu'.

```
# Calculate negative predictive value (NPV)
NPV = TN / (TN + FN)
print("Negative Predictive Value (NPV):", NPV)
```

### Rajah 28: Kod Python bagi mencari nilai ramalan negatif model

### Keputusan dan Perbincangan



Rajah 29: Graf bar perbandingan metrik penilaian berdasarkan tiga model ramalan

	Ketepatan	Kebersihan	Pengkhususan	Nilai Ramalan Negatif
KNN	0.95	0.92	0.90	0.95
Pohon Keputusan	0.85	0.80	0.73	0.85
ANN	0.95	0.97	0.89	1.0

Rajah 30: Jadual perbandingan metrik penilaian

Daripada jadual dan rajah, perbandingan prestasi model ramalan dilakukan menggunakan empat metrik penilaian iaitu ketepatan, kebersihan, pengkhususan dan nilai ramalan negatif. Model ANN menunjukkan keseluruhan keputusan terbaik bagi keseluruhan metrik penilaian. Bagi perbandingan ketepatan, model ANN dan KNN menghasilkan ketepatan yang paling tinggi dengan nilai 0.95, manakala pohon keputusan sebanyak 0.85. Seterusnya, bagi kebersihan, model ANN mempunyai nilai yang paling tinggi dengan 0.97, diikuti dengan KNN sebanyak 0.92 dan akhir sekali adalah model pohon keputusan dengan 0.80. Selain itu, metrik penilaian pengkhususan dibandingkan dan model KNN mempunyai nilai yang paling tinggi dengan 0.90, diikuti dengan ANN sebanyak 0.89 dan pohon keputusan dengan 0.73. Akhir sekali adalah nilai ramalan negatif di mana model ANN menghasilkan nilai yang paling tinggi iaitu 1.0, model KNN dengan 0.95 dan pohon keputusan dengan 0.85. Perbandingan dibuat dengan melihat model yang menghasilkan nilai metrik yang tinggi bagi keempat-empat metrik. Ini kerana, bagi semua metrik tersebut, lagi tinggi nilai metrik, lagi bagus model dalam meramal prestasi LRT.

Kesimpulannya, jelaslah bahawa model ANN adalah model yang paling sesuai digunakan dalam kajian ini kerana model ini telah menghasilkan keputusan penilaian yang paling bagus berbanding dengan yang lain. Dengan nilai ketepatan 0.94, ia menandakan model ini mempunyai keupayaan yang tinggi dalam meramal prestasi LRT, 0.97 bagi nilai kebersihan menunjukkan model ini bagus dalam meramal prestasi 'Good', nilai pengkhususan 0.89 menandakan model berkeupayaan mengenal pasti prestasi 'Bad' dan 1.0 bagi nilai ramalan negatif adalah bukti model ramalan ini adalah yang terbaik dalam meramal 'Negatif Sebenar' iaitu prestasi 'Bad'. Model ini

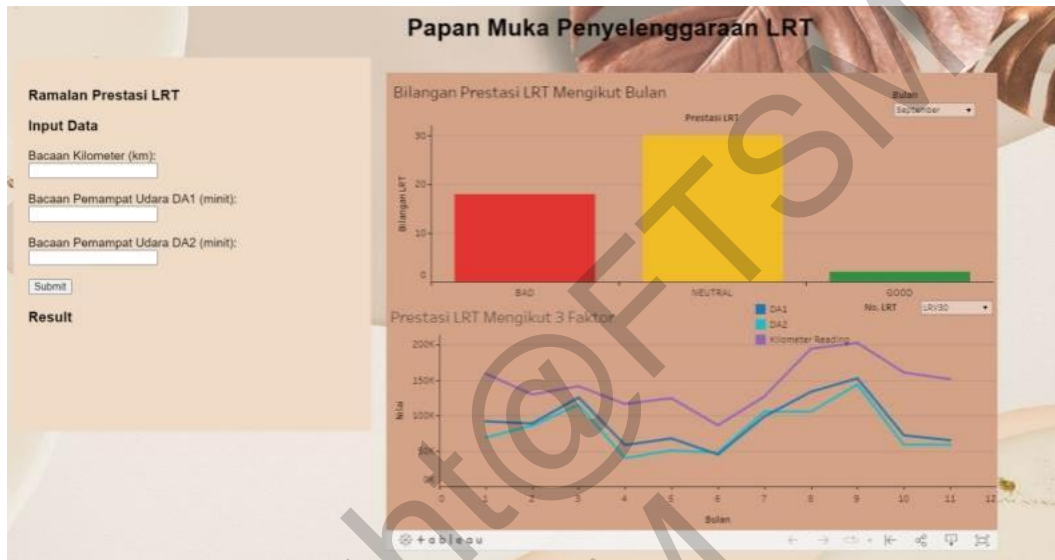
adalah paling sesuai dalam meramalkan prestasi LRT kerana keupayaan dalam meramal prestasi 'Good' dan 'Bad' adalah paling tinggi dan sesuai dengan objektif kajian.

Dibandingkan dengan kajian lepas yang dilakukan oleh Chunsheng Yan bertajuk 'Developing Machine Learning-Based Models for railway Inspection', model yang memperoleh ketepatan paling tinggi adalah model ramalan pohon keputusan. Ini kerana ciri data yang digunakan adalah dalam bentuk kategori sesuai dengan kegunaan model ramalan. Manakala kajian ini, ANN merupakan model pilihan kerana data yang digunakan berada dalam bentuk nilai berterusan. Selain itu, terdapat juga kajian lepas yang mendapat model ramalan ANN sebagai model pilihan iaitu kajian yang dilakukan oleh Jun S. Lee bertajuk 'Prediction of Track Deterioration Using Maintenance Data and Machine Learning'. Namun begitu, kajian ini berbeza dari segi format penghasilan di mana kajian Jun S. Lee menghasilkan nilai berterusan manakala kajian ini dalam format teks. Kelebihan kajian ini adalah kajian ini membina penyelesaian mudah bagi masalah harian dan boleh dimanipulasi bagi masalah yang lebih rumit di masa hadapan. Selain itu, ia dapat menambahbaik sistem penyelenggaraan LRT di Malaysia.

Selain daripada model ramalan, kajian juga menghasilkan sebuah papan pemuka yang dapat memudahkan penggunaan model ramalan tersebut. Bahasa pengaturcaraan yang digunakan dalam bahagian ini adalah bahasa HTML dan JavaScript bagi membentuk antara muka pengguna ramalan prestasi LRT. Selain itu, ia juga menggunakan tableau bagi memvisualisasi data dalam bentuk graf. Bahagian utama antara muka adalah panel ramalan prestasi LRT yang terletak di sebelah kiri. Pengguna boleh memasukkan input bagi bacaan jumlah jarak perjalanan, bacaan masa pemampat udara DA1 dan bacaan masa pemampat udara DA2. Kemudian, butang 'Submit' ditekan untuk memaparkan hasil ramalan.

Kemudian, di bahagian kanan merupakan paparan data analisis menggunakan Tableau. Di bahagian ini ia memaparkan dua graf iaitu graf bar yang menunjukkan bilangan LRT bagi setiap prestasi mengikut bulan yang dipilih. Ini adalah untuk memudahkan pengguna mengenal pasti

bilangan LRT yang memerlukan perhatian dengan segera. Selain itu, ia juga memaparkan graf garis yang menunjukkan keadaan atau kegunaan LRT dalam setahun mengikut LRT yang dipilih. Ini memberi peluang untuk memerhatikan tren prestasi sesebuah LRT dalam masa setahun berdasarkan 3 faktor. Rajah di bawah menunjukkan pengekodan dan reka bentuk keseluruhan platform web ringkas yang dibina untuk kegunaan pengguna.



Rajah 31: Papan pemuka penyelenggaraan LRT

Rajah 32: Papan pemuka bagi membuat ramalan prestasi LRT



Rajah 33: Grar bar bilangan prestasi LRT mengikut bulan



Rajah 34: Graf garis prestasi LRT mengikut tiga parameter kajian

Pelbagai kelebihan diperoleh dengan membina penyelesaian ini seperti dapat membantu memudahkan pekerja mengenal pasti prestasi LRT, membantu dalam membuat keputusan sama ada LRT tersebut memerlukan perhatian dengan segera dan memajukan sistem penyelenggaraan LRT di Malaysia. Paling ketara, kajian ini membuka peluang untuk mengimplikasikan teknik menganalisis data dalam industri sekali gus memajukan industri sistem pengangkutan. Selain itu, dengan terjaminnya prestasi LRT yang sentiasa memuaskan, pengguna akan berpuas hati dengan penggunaan LRT secara tidak langsung mempromosikan penggunaan pengangkutan awam di Malaysia. Kajian ini mempunyai kekangan data sejarah, di mana data untuk melatih model ramalan adalah terhad. Ini memberi peluang kepada penambahbaikan kajian pada masa hadapan untuk menambah data sejarah dan parameter bagi menghasilkan model yang lebih bagus.

## Kesimpulan

Melalui laporan ini, dapat disimpulkan bahawa terdapat pelbagai maklumat yang dapat diperoleh dari segi mengenal pasti masalah, pembinaan model, penilaian dan perbandingan teknik pembelajaran mesin. Hasil kajian ini mendapati bahawa model ramalan ANN adalah model yang mempunyai keupayaan yang paling tinggi dalam meramal prestasi LRT berdasarkan keadaan semasa LRT. Model ini telah menghasilkan 0.94 bagi nilai ketepatan, 0.97 bagi nilai kepersisan, 0.89 bagi pengkhususan dan 1.0 bagi nilai ramalan negatif. Ini membuktikan model ini menghasilkan keputusan penilaian yang paling bagus berbanding dengan yang lain di mana keupayaan meramal prestasi LRT berlabel 'Good' serta mengenal pasti dan meramal prestasi LRT berlabel 'Bad'. Ini adalah penting kerana kesilapan dalam meramal prestasi LRT akan mengakibatkan insiden yang tidak diinginkan dan keselamatan pengguna akan terjejas. Ini adalah sesuai dengan objektif kajian yang ingin membina model ramalan bagi membantu memudahkan proses mengenal pasti prestasi LRT.

Kajian ini telah mencapai dua objektif kajian iaitu untuk membina model ramalan prestasi LRT berdasarkan keadaan semasa LRT. Ini dapat dibuktikan dengan penghasilan model ramalan ANN yang mempunyai keupayaan meramal prestasi LRT yang tinggi. Selain itu, kajian ini juga berjaya membina satu papara atau antara muka yang dapat memudahkan penggunaan model ramalan tersebut. Seterusnya, antara muka ini juga dapat memaparkan graf bar bilangan prestasi LRT mengikut bulan bagi memudahkan pekerja merekodkan bilangan LRT yang memerlukan perhatian segera dan graf garis prestasi LRT dalam masa setahun berdasarkan 3 parameter kajian yang membantu pekerja menganalisis penggunaan LRT sepanjang tahun tersebut.

Kajian ini memberi impak yang positif dalam industri pengangkutan awam di mana ia telah menambah baik proses penyelenggaraan LRT daripada kaedah sedia ada yang hanya bergantung kepada diagnosis pakar. Kelebihan utama kajian ini adalah, ia memudahkan pekerja mengenal pasti prestasi LRT sekali gus membantu dalam menyusun semula penjadualan penyelenggaraan

mengikut keutamaan. Selain itu, kajian ini memastikan LRT yang beroperasi sentiasa dalam keadaan yang memuaskan yang terjamin daripada insiden mengejut seperti berhenti secara tiba-tiba semasa beroperasi. Terakhir adalah kajian ini memberi peluang industri mengimplimentasi teknik menganalisis data bagi memajukan sistem penyelenggaraan sedia ada sekali gus menyedarkan ramai pihak tentang kelebihan teknolgi dalam memudahkan beban kerja seseorang.

Kelemahan kajian ini adalah perbandingan teknik pembelajaran mesin yang terhad kepada tiga iaitu KNN, Pohon keputusan dan ANN mengehadkan peluang untuk mencari model ramalan yang lebih baik. Seterusnya, kekangan set data sejarah tidak mencukupi atau sedikit dapat menjejaskan keboleham model dalam membuat ramalan kerana kemungkinan model tidak dapat belajar corak dalam data adalah tinggi. Akhir sekali, data adalah terhad kepada laluan Ampang sahaja memberi impak kepada kemungkinan data yang telah diterima tidak mewakili data keseluruhan berkenaan keadaan semasa LRT dengan baik. Cadangan bagi kajian masa hadapan yang dapat dilakukan adalah menambah set data rekod keadaan semasa LRT supaya model ramalan dapat belajar corak dalam data dengan lebih terperinci. Selain itu, menambah teknik pembelajaran mesin bagu dapat membuat perbandingan yang bervariasi. Seterusnya penambahan parameter lain bagi keadaan semasa LRT supaya ramalan prestasi LRT adalah lebih tepat.

Kesimpulan keseluruhan bagi pembangunan projek ini dapat dirumuskan, setiap perancangan yang di beri telah berjaya dilaksanakan dan setiap objektif berjaya dicapai. Di akhir perlaksanaan projek ini, setiap kekuatan dan kekangan projek serta penambahbaikan di masa hadapan di senaraikan agar setiap penemuan yang di jumpai boleh dibaiki pada masa akan datang.

### **Penghargaan**

Setinggi-tinggi penghargaan saya ucapkan kepada penyelia saya iaitu Prof. Madya Dr. Zalinda Binti Othman di atas segala tunjuk ajar dan bantuan yang diberikan sepanjang proses penyiapan kajian ini. Segala jasa, ilmu dan tunjuk ajar beliau akan saya kenang dan implementasikan sepanjang hayat.

Tidak dilupakan juga ucapan terima kasih saya kepada ahli keluarga tercinta terutamanya ibu bapa saya yang sentiasa memberi semangat dan kekuatan sepanjang tempoh saya melengkapkan kajian tahun akhir ini. Tidak dilupakan juga adik saya yang banyak menyokong dan memberi nasihat supaya sentiasa fokus dalam menyiapkan kajian ini.

Akhir kata, terima kasih kepada semua yang terlibat secara langsung atau tidak langsung dalam menghasilkan kajian tahun akhir ini. Saya juga ingin memohon ribuan kemaafan sekiranya terdapat kesilapan sepanjang menjalankan kajian tahun akhir ini. Sekian, terima kasih.

### RUJUKAN

- David H. Fretwell, Morgan V. Lewis, Arjen Deij. 2001. A Framework for Defining and Assessing Occupational and Training Standards in Developing Countries. *Information Series No. 386*
- Ismail, S. 2016. Investigative Model of Rail Accident and Incident Causes Using Statistical Modelling Approach. Master Thesis. Faculty of Engineering Technology. Universiti Tun Hussein Onn Malaysia, Malaysia.
- Lee, Jun & Hwang, Sung Ho & Choi, Il & Kim, In. (2018). Prediction of Track Deterioration Using Maintenance Data and Machine Learning Schemes. *Journal of Transportation Engineering Part A: Systems. 144. 10.1061/JTEPBS.0000173.*
- Maksym Spiryagin, Peter Wolfs, Colin Cole, Valentyn Spiryagin, Yan Quan Sun, Tim McSweeney. 2016. Design and Simulation of Heavy Haul Locomotives and Trains *10(2), 324.*
- Yang, Chunsheng & Sun, Yanmin & Ladubec, Chris & Liu, Yan. 2020. Developing Machine Learning-Based Models for Railway Inspection. *Applied Sciences. 11. 13. 10.3390/app11010013.*

Aini Nabila binti Mohd Azizi (A182923)  
Prof. Madya Dr. Zalinda Binti Othman  
Fakulti Teknologi & Sains Maklumat,  
Universiti Kebangsaan Malaysia