Laporan Implementasi

Prediksi Debit Aliran menggunakan *Long Short-Term Memory* (LSTM)

Versi (Rapih) 1.0.0

Berdasarkan *Jupyter Notebook*: `github_taruma_demo_lstm_rr.ipynb`

oleh Taruma Sakti Megariansyah

22 Oktober 2019

github.com/taruma/vivaldi

1 Prediksi Debit Aliran Menggunakan *Long Short-Term Memory* (LSTM)

Jupyter Notebook (selanjutnya disebut buku) ini hanya **contoh** dan dibuat untuk **pembelajaran** mengenai *Deep Learning/Neural Networks* dan mendemonstrasikan penggunaan *Python* di bidang sumberdaya air. Buku ini masih perlu dievaluasi kembali jika digunakan untuk kepentingan riset/penelitian ataupun proyek.

Buku ini disertai catatan yang berisikan penjelasan lebih lanjut mengenai buku ini (daftar pustaka, penjelasan dataset, dll). Catatan dapat diunduh di bagian unduh buku.

1.1 Pranala buku

Buku ini bisa diunduh dengan berbagai format. Versi Google Colab akan lebih diperbarui dibandingkan versi lainnya.

- Pranala Google Colab, format Google Colab versi terakhir
- Pranala Github, format .ipynb versi 1.0.0
- Lihat melalui NBViewer, format .ipynb versi Github
- Unduh Laporan, format PDF versi 1.0.0 dengan *source code* + *outputs*
- Unduh Laporan (rapih), format PDF versi 1.0.0 hanya *outputs**
- Unduh Catatan, format PDF versi 1.0.0

Pembuatan laporan dilakukan dengan mengubah buku ke dalam bentuk LaTeX dan dilakukan perubahan sedikit, sehingga disarankan untuk mengunduh versi laporan (rapih).

1.2 Catatan

- Buku ini dikembangkan menggunakan Google Colab, sehingga penampilan terbaik dan interaktif dari buku ini diperoleh jika dibuka melalui Google Colab.
- Anda dapat mendiskusikan mengenai buku ini (atau hal lainnya seperti koreksi, kritik, saran, pertanyaan, dll) melalui isu di *repository* taruma/vivaldi atau dapat menghubungi saya melalui email *hi@taruma.info*.
- Buku ini masih perlu dievaluasi baik dari teori ataupun implementasi. Ini merupakan buku pribadi yang digunakan oleh saya sebagai latihan implementasi *Deep Learning* menggunakan *Python*. Referensi materi pembelajaran saya dapat dilihat pada catatan buku.
- **Biasakan untuk selalu memeriksa kode terlebih dahulu sebelum menjalankannya untuk masalah keamanan**.

2 Deskripsi Kasus

Bagian ini menjelaskan gambaran umum mengenai dataset, permasalahan/tujuan, dan strategi penyelesaiannya.

2.1 Dataset

Dataset merupakan data hidrologi dan klimatologi **harian** dari tanggal **1 Januari 1998** sampai **31 Desember 2008** Daerah Aliran Sungai (DAS) Bendung Baru Pamarayan. Dataset terpisah menjadi 3 kategori yaitu: data curah hujan, data klimatologi, dan data debit.

- Data curah hujan diperoleh dari 8 stasiun yaitu: `bojong_manik`, `gunung_tunggal`, `pasir_ona`, `sampang_peundeuy`, `cimarga`, `bd_pamarayan`, `ciminyak_cilaki`, `gardu_tanjak`.
- Data debit diperoleh dari 1 stasiun yaitu: `bd_pamarayan`.
- Data klimatologi diperoleh dari 1 stasiun yaitu: `geofisika_serang`.

Rincian mengenai dataset bisa dibaca di catatan buku.

2.2 Objektif

2.2.1 Tujuan

- Peneliti ingin mengetahui nilai debit berdasarkan data hidrologi dan klimatologi yang tersedia pada waktu sebelumnya.

2.2.2 Batasan Masalah

- Arsitektur (sel) *Recurrent Neural Networks* yang akan digunakan adalah *Long Short-Term Memory* (LSTM).
- Data yang hilang (`NaN`) diisi menggunakan metode interpolasi linear.
- Diasumsikan bahwa data tidak perlu diverifikasi.
- Jika data yang hilang lebih dari 1 tahun berurutan, maka paramater tersebut akan diabaikan.
- Pelatihan model (training) menggunakan data dari tahun **1998 - 2006**.
- Tidak dilakukan *feature engineering*, kolom yang bertipe ordinal atau kategori diabaikan.
- Tidak ada tahapan pemilihan model terbaik (*model selection*). Parameter akan sembarang mengikuti tulisan Kratzert et al (2018).
- Berdasarkan Kratzert et al (2018), pada buku ini mengikuti bahwa dataset hanya dibagi dua bagian yaitu *train set* dan *test set*, dimana validasi menggunakan *test set*.

2.2.3 Pertanyaan

- Berapa nilai debit pada waktu t jika telah diketahui nilai observasi pada waktu *timesteps* hari sebelumnya?

3 TAHAP 0: Pengaturan Awal dan Inisiasi

Pada tahap ini akan dilakukan pengaturan awal dan inisiasi dengan melakukan atau menjawab daftar berikut:

- Menentukan penggunaan *runtime* lokal atau *Google Colab*.
- Menentukan nama buku/proyek dan versi (digunakan jika melakukan penyimpanan).
- Memeriksa paket hidrokit.
- Menampilkan versi paket yang digunakan pada sistem.
- Impor paket utama yang akan digunakan (numpy, pandas, matplotlib).

3.1 Pengaturan Buku

```
:: INFORMASI RUNTIME
:: BUKU INI MENGGUNAKAN RUNTIME: GOOGLE COLAB

:: INFORMASI PROYEK/BUKU
:: [project_title]: 20191022_0807_taruma_demo_lstm_rr_1_0_0

:: MENGGUNAKAN TENSORFLOW 2.x (GOOGLE COLAB)
TensorFlow 2.x selected.

:: MEMERIKSA PAKET HIDROKIT
:: INSTALASI PAKET HIDROKIT
  Building wheel for hidrokit (setup.py) ... done

:: INFORMASI VERSI SISTEM
::       python version: 3.6.8
::        numpy version: 1.16.5
::       pandas version: 0.24.2
::   matplotlib version: 3.0.3
::   tensorflow version: 2.0.0
::        keras version: 2.2.4-tf
::     hidrokit version: 0.3.2

:: LOKASI PENYIMPANAN DATASET
DATASET_PATH = /content/gdrive/My Drive/Colab Notebooks/_dataset/uma_pamarayan
DROP_PATH = /content/gdrive/My Drive/Colab Notebooks/_dropbox
```

3.2 Persiapan sistem

```
:: IMPORT LIBRARY (NUMPY, PANDAS, MATPLOTLIB)
```

4 TAHAP 1: AKUISISI DATASET

Pada tahap ini, tujuan utamanya adalah membaca seluruh dataset yang dimiliki dan mengimpor dataset tersebut untuk pengolahan prapemrosesan data (*data preprocessing*).

4.1 DATA HUJAN

Pada kasus ini, terdapat 8 stasiun yang akan digunakan sehingga terdapat 8 berkas excel. Setiap berkas memiliki data curah hujan dari tahun 1998 hingga 2008 yang disimpan pada masing-masing *sheet* untuk setiap tahunnya.

Untuk memperoleh data tersebut dalam bentuk tabel (bukan dalam bentuk pivot) digunakan modul yang tersedia di `hidrokit` (hanya pada versi 0.3.x). Modul dapat diakses melalui `hidrokit.contrib.taruma.hk43` (panduan).

```
:: MEMBACA DATA HUJAN DARI [DATASET_PATH]
Found 8 file(s)
::  1  :        hujan_bojong_manik_1998_2008.xls
::  2  :        hujan_gunung_tunggal_1998_2008.xls
::  3  :        hujan_pasir_ona_1998_2008.xls
::  4  :        hujan_sampang_peundeuy_1998_2008.xls
::  5  :        hujan_cimarga_1998_2008.xls
::  6  :        hujan_bd_pamarayan_1998_2008.xls
::  7  :        hujan_ciminyak_cilaki_1998_2008.xls
::  8  :        hujan_gardu_tanjak_1998_2008.xls

:: tipe [hujan_raw] = <class 'dict'>
:: tipe [hujan_invalid] = <class 'dict'>
```

4.2 DATA DEBIT

Untuk data debit, dilakukan hal yang serupa dengan data hujan.

```
:: MEMBACA DATA DEBIT DARI [DATASET_PATH]
Found 1 file(s)
::  1  :        debit_bd_pamarayan_1998_2008.xls

:: tipe [debit_raw] = <class 'dict'>
:: tipe [debit_invalid] = <class 'dict'>
```

4.3 DATA KLIMATOLOGI

Data klimatologi diperoleh dari situs data online bmkg. Data klimatologi dari bmkg lebih mudah di impor dikarenakan data sudah tersedia dalam bentuk tabel.

Digunakan modul `hidrokit.contrib.taruma.hk73` (panduan) agar memudahkan proses impornya.

```
:: [KLIMATOLOGI_PATH] = /content/gdrive/My Drive/Colab
Notebooks/_dataset/uma_pamarayan/klimatologi_serang_1998_2008.xlsx
:: MEMBACA DATA KLIMATOLOGI DARI [KLIMATOLOGI_PATH]
:: tipe [df_klimatologi] = <class 'pandas.core.frame.DataFrame'>
```

5 TAHAP 2: PRAPEMROSESAN DATA

Pada tahap ini, dataset yang telah diimpor akan diperiksa dan dipersiapkan untuk pengolahan data di tahap selanjutnya. Berikut yang dilakukan pada tahap ini:

- Memastikan dataset berupa `pandas.DataFrame`.
- Memeriksa data yang invalid (salah input/bukan bilangan).
- Mengoreksi nilai yang invalid.
- Mengubah tipe data pada dataframe menjadi numerik.
- Memeriksa data yang hilang (`NaN`).
- Mengisi nilai hilang dengan metode interpolasi linear.
- Menyesuaikan kelengkapan dataset.

5.1 DATA HUJAN

```
:: MENGUBAH [hujan_raw] MENJADI [df_hujan] SEBAGAI DATAFRAME
:: MENAMPILKAN [df_hujan]
```

[0]:

```
            hujan_bojong_manik  ... hujan_gardu_tanjak
1998-01-01                   -  ...                   -
1998-01-02                   -  ...                   -
1998-01-03                   5  ...                   5
1998-01-04                   -  ...                   -
1998-01-05                   -  ...                   -

[5 rows x 8 columns]

:: MEMERIKSA NILAI INVALID PADA [df_hujan]
:: MENAMPILKAN NILAI INVALID
:: NILAI INVALID BERUPA = ['-', 'NaN']
```

Dari `hujan_invalid` diketahui bahwa pada data hujan memiliki data invalid berupa isian "`-`" dan "`NaN`". Isian "`-`" pada data curah hujan menandakan bahwa tidak ada hujan atau bernilai `0.`. Sedangkan data "`NaN`" menandakan bahwa data tidak terekam sama sekali sehingga tidak diketahui terjadi hujan atau tidak.

```
:: MENGOREKSI NILAI INVALID PADA [df_hujan]
:: MENGOREKSI NILAI "-" MENJADI 0.0

:: MENGUBAH TIPE DATA PADA DATAFRAME [df_hujan]
:: MENAMPILKAN INFORMASI DATAFRAME [df_hujan]:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4018 entries, 1998-01-01 to 2008-12-31
Freq: D
Data columns (total 8 columns):
hujan_bojong_manik        4017 non-null float64
hujan_gunung_tunggal      4018 non-null float64
hujan_pasir_ona           4018 non-null float64
hujan_sampang_peundeuy    4016 non-null float64
hujan_cimarga             4018 non-null float64
```

```
hujan_bd_pamarayan        4016 non-null float64
hujan_ciminyak_cilaki     4018 non-null float64
hujan_gardu_tanjak        4018 non-null float64
dtypes: float64(8)
memory usage: 282.5 KB
```

Dari informasi diatas diketahui bahwa tipe data pada dataframe telah diubah menjadi berbentuk numerik.

```
:: MENGISI NILAI HILANG MENGGUNAKAN METODE INTERPOLASI LINEAR
```

```
:: MEMERIKSA JIKA [df_hujan] MASIH MEMILIKI NILAI YANG HILANG: False
```

5.2 DATA DEBIT

Langkahnya serupa dengan data hujan.

```
:: MENGUBAH [debit_raw] MENJADI [df_debit] SEBAGAI DATAFRAME
:: MENAMPILKAN [df_debit]
```

[0]:

	debit_bd_pamarayan
1998-01-01	0
1998-01-02	0
1998-01-03	0
1998-01-04	0
1998-01-05	0

```
:: MEMERIKSA NILAI INVALID PADA [df_debit]
:: MENAMPILKAN NILAI INVALID
:: NILAI INVALID BERUPA = ['20.9.46', 'NaN', 'tad']
```

Dari `debit_invalid` diketahui bahwa `df_debit` memiliki nilai invalid berupa "`20.9.46`", "`NaN`", dan "`tad`". Nilai "`tad`" diartikan sebagai tidak ada data, sedangkan "`NaN`" adalah data yang hilang. Untuk nilai "`20.9.46`", diasumsikan terjadi kekeliruan saat memasukkan nilai, nilai tersebut dikoreksi menjadi `209.46`.

```
:: MENGOREKSI NILAI 20.9.46 MENJADI 209.46
:: MENGOREKSI NILAI tad MENJADI NaN
```

```
:: MENGUBAH TIPE DATA PADA DATAFRAME [df_debit]
:: MENAMPILKAN INFORMASI DATAFRAME [df_debit]:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4018 entries, 1998-01-01 to 2008-12-31
Freq: D
Data columns (total 1 columns):
debit_bd_pamarayan    4016 non-null float64
dtypes: float64(1)
memory usage: 62.8 KB
```

Dari informasi diatas diketahui bahwa tipe data pada dataframe telah diubah menjadi berbentuk numerik.

```
:: MENGISI NILAI HILANG MENGGUNAKAN METODE INTERPOLASI LINEAR
```

```
:: MEMERIKSA JIKA [df_debit] MASIH MEMILIKI NILAI YANG HILANG: False
```

Nilai `0.` dapat berarti terjadi kekeringan atau pengeringan (pada berkas tercantum "pengeringan" pada periode 13 Oktober 2000-31 Oktober 2000).

```
:: MEMERIKSA KEKERINGAN PADA DATA DEBIT
:: Kekeringan terjadi pada tanggal: ['01 Jan 1998-28 Feb 1998', '15 Mar 1999',
'29 Oct 1999-31 Oct 1999', '13 Oct 2001-31 Oct 2001', '24 Oct 2003-25 Oct 2003',
'15 Jun 2004', '31 Aug 2004', '16 Nov 2004-30 Nov 2004', '08 Oct 2005', '11 Oct
2005-13 Oct 2005', '02 Oct 2006-19 Oct 2006', '16 Oct 2007-18 Oct 2007', '07 Sep
2008', '17 Oct 2008-18 Oct 2008']
```

Pada awal dataset (1 Januari 1998-28 Februari 1998) selama dua bulan bernilai 0. secara berurutun. Saya mengasumsikan ini bukan terkait kekeringan/pengeringan, melainkan data pengukuran dimulai pada bulan maret tahun 1998. Sehingga, saya simpulkan bahwa dalam pemodelan **dataset akan dimulai pada tanggal 1 Maret 1998**.

5.3 DATA KLIMATOLOGI

Proses pada data klimatologi tidak jauh berbeda dengan proses data hujan/debit. Akan tetapi karena data klimatologi berasalkan dari sumber berbeda (BMKG), maka implementasinya akan berbeda dengan implementasi pada data hujan/debit.

Pada modul `hidrokit.contrib.taruma.hk73` (panduan) telah disiapkan beberapa fungsi yang memudahkan untuk memeriksa data klimatologi.

5.3.1 Persiapan

Pada tahap 1, data klimatologi telah berbentuk `DataFrame`, sehingga dapat langsung dilakukan prapemrosesan data.

```
:: MENAMPILKAN DATAFRAME [df_klimatologi]:
```

```
[0]:
```

Tanggal	Tn	Tx	Tavg	RH_avg	RR	ss	ff_x	ddd_x	ff_avg	ddd_car
1998-01-01	23.0	34.6	27.5	75	0.0	5.8	5	225	2	SW
1998-01-02	23.2	34.2	28.6	69	0.0	7.6	4	270	1	NE
1998-01-03	24.0	34.6	27.7	76	0.0	5.6	7	270	2	W
1998-01-04	23.8	34.4	28.4	70	0.0	8.0	7	225	3	SW
1998-01-05	23.5	33.4	27.7	74	1.0	3.5	6	270	2	W

Sebelum melanjutkan dalam prapemrosesan data pada data klimatologi, terdapat beberapa kolom yang dihilangkan karena batasan masalah buku ini dan mempermudah saat pemodelan. Kolom yang digunakan hanya kolom numerik yang bersifat kontinu.
Berikut kolom yang dihilangkan:

- kolom `ddd_car`, kolom ini merupakan kolom kategori yang tidak berupa angka.
- kolom `ff_x`, `ddd_x`, `ff_avg`, kolom ini (dapat) berupa kolom ordinal.

Kolom tersebut dapat diubah melalui proses *feature engineering*. Referensi lanjut bisa baca di sini dan di sini.

Selain itu, berdasarkan Megariansyah (2015) kolom RR (curah hujan) tidak dapat digunakan karena stasiun tersebut tidak termasuk pada wilayah Daerah Aliran Sungai (DAS) yang dikaji.

```
:: MEMBERSIHKAN [df_klimatologi] KE DATAFRAME [df_klimatologi_clean]
```

[0]:

Tanggal	Tn	Tx	Tavg	RH_avg	ss
1998-01-01	23.0	34.6	27.5	75	5.8
1998-01-02	23.2	34.2	28.6	69	7.6
1998-01-03	24.0	34.6	27.7	76	5.6
1998-01-04	23.8	34.4	28.4	70	8.0
1998-01-05	23.5	33.4	27.7	74	3.5

5.3.2 Prapemrosesan

Dilanjutkan dengan tahap prapemrosesan seperti memeriksa data invalid ataupun kehilangan data.

```
:: MEMERIKSA DATA YANG HILANG PADA [df_klimatologi_clean]
:: [df_klimatologi_clean] memiliki kehilangan data: True
:: Kolom yang memiliki data hilang: ['ss']
```

Diketahui bahwa pada kolom ss terdapat kehilangan data "NaN". Cek apakah kehilangan data terjadi secara beruntun.

```
:: MEMERIKSA KONDISI DATA HILANG PADA [df_klimatologi_clean].ss
:: Tanggal terjadinya kehilangan data: ['30 Apr 2003']
```

Karena kehilangan data hanya terjadi pada satu hari, maka kolom ss akan digunakan dalam pemodelan.

Berdasarkan situs BMKG, harus diperiksa juga mengenai nilai 8888 dan 9999 yang menandakan bahwa data tidak terukur dan/atau tidak ada data. Data tersebut akan dikoreksi menjadi nilai hilang (NaN) dan akan diisi menggunakan metode interpolasi.

```
:: MEMERIKSA DATA YANG TIDAK ADA/TEREKAM PADA [df_klimatologi_clean]
{'Tn': array([], dtype=int64), 'Tx': array([], dtype=int64), 'Tavg': array([],
dtype=int64), 'RH_avg': array([], dtype=int64), 'ss': array([], dtype=int64)}
```

Ternyata, pada kolom lain tidak memiliki nilai yang tidak terukur/tidak ada. Sehingga, langkah berikutnya mengisi nilai hilang menggunakan metode interpolasi linear.

```
:: MENGISI NILAI YANG HILANG MENGGUNAKAN METODE INTERPOLASI LINEAR

:: MEMERIKSA JIKA [df_klimatologi_clean] MASIH MEMILIKI NILAI YANG HILANG: False
```

5.4 PENGGABUNGAN DATASET

Ketiga data yaitu data hujan, data klimatologi, dan data debit digabungkan dalam satu DataFrame untuk pemodelan. Data gabungan sudah dipastikan tidak memiliki nilai yang invalid atau data yang hilang.

Berdasarkan prapemrosesan data debit, diketahui bahwa dua bulan pertama (Januari-Februari 1998) tidak memiliki data, maka dataset akan menggunakan periode yang dimulai dari 1 Maret 1998.

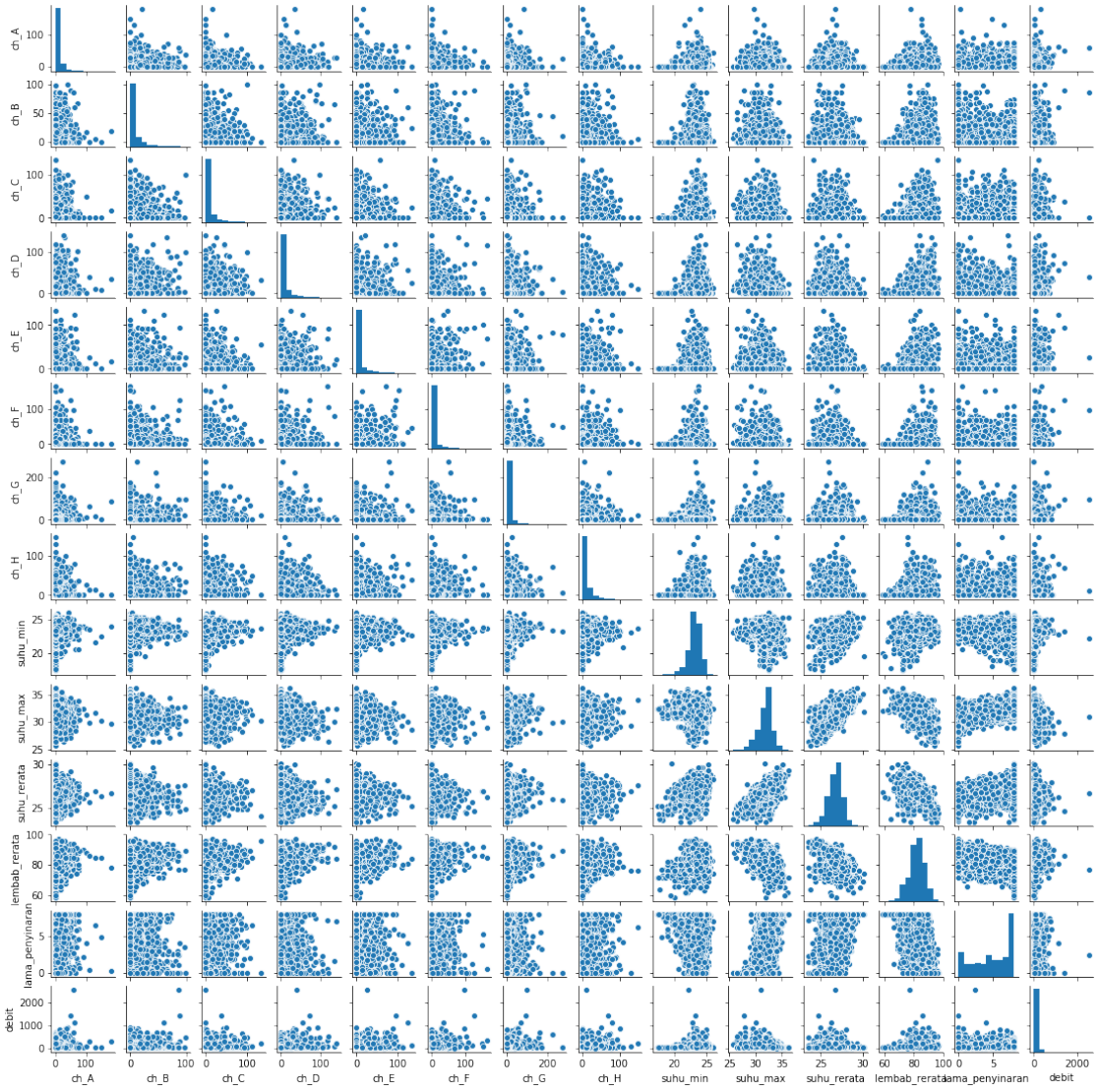
```
:: MENENTUKAN PERIODE DATASET
:: PERIODE DATASET DARI 19980301 hingga 20081231
:: DataFrame [data_hujan], [data_debit], [data_klimatologi]

:: MENGGABUNGKAN SELURUH DATA DALAM SATU DATAFRAME [dataset]

:: MENAMAI ULANG NAMA KOLOM
:: INFO [dataset]:
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3959 entries, 1998-03-01 to 2008-12-31
Freq: D
Data columns (total 14 columns):
ch_A               3959 non-null float64
ch_B               3959 non-null float64
ch_C               3959 non-null float64
ch_D               3959 non-null float64
ch_E               3959 non-null float64
ch_F               3959 non-null float64
ch_G               3959 non-null float64
ch_H               3959 non-null float64
suhu_min           3959 non-null float64
suhu_max           3959 non-null float64
suhu_rerata        3959 non-null float64
lembab_rerata      3959 non-null int64
lama_penyinaran    3959 non-null float64
debit              3959 non-null float64
dtypes: float64(13), int64(1)
memory usage: 623.9 KB

:: STATISTIK DESKRIPTIF [dataset]
```

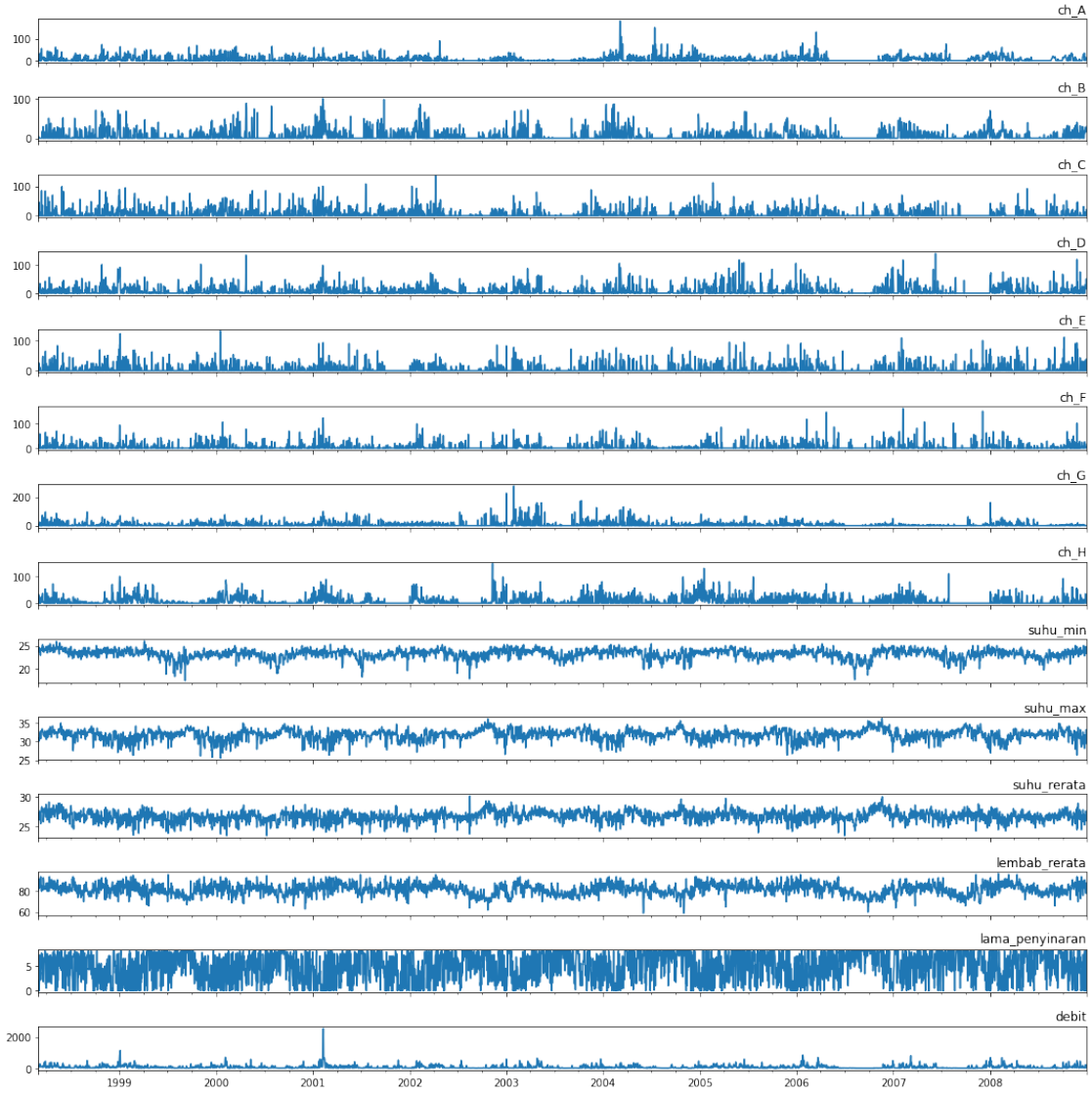
[0]:

	mean	std	min	50%	max
ch_A	6.222417	12.044965	0.0	0.00	180.00
ch_B	6.087017	12.304554	0.0	0.00	99.50
ch_C	6.249937	13.616252	0.0	0.00	135.00
ch_D	6.949987	14.510073	0.0	0.00	140.00
ch_E	5.732609	13.728183	0.0	0.00	133.00
ch_F	5.316241	13.329691	0.0	0.00	163.00
ch_G	7.955418	17.472431	0.0	0.00	275.00
ch_H	8.469437	14.768919	0.0	0.00	148.00
suhu_min	23.139227	1.026656	17.4	23.20	26.00
suhu_max	31.826421	1.376400	25.6	32.00	36.20
suhu_rerata	26.734731	0.844836	23.5	26.80	30.10
lembab_rerata	81.678707	5.295782	59.0	82.00	97.00
lama_penyinaran	4.840162	2.721498	0.0	5.20	8.00

```
debit           72.137286  105.077023   0.0  24.08  2561.58
```

5.4.1 Visualisasi





Dari grafik diatas dapat beberapa informasi baru yang diperoleh berupa:

- Terlihat nilai luaran *outlier* pada kolom debit yang terjadi di sekitar tahun 2001.
- Untuk data hujan, hanya pada stasiun G yang memiliki rentang nilai dari 0-200, sedangkan yang lain bernilai 0-100.

Nilai luaran akan disertakan dalam pemodelan. Dan pada akhir tahap ini, diasumsikan bahwa dataset sudah terverifikasi dan tervalidasi disertai selesai melewati prapemrosesan data. Sehingga, pada tahap berikutnya, dataset akan dianggap sudah memenuhi kriteria untuk pemodelan.

6 TAHAP 3: INPUT PEMODELAN

Pada tahap ini akan fokus dalam persiapan input pemodelan. Langkah yang akan dilakukan antara lain:

1. Membagi dataset menjadi dua bagian yaitu *train set* dan *test set*.
2. Normalisasi/Standarisasi nilai pada dataset.
3. Mempersiapkan *input tensor* untuk masing-masing *train set* dan *test set*.

6.1 Menentukan *train set* dan *test set*

Sudah direncanakan bahwa untuk *train set* menggunakan periode selama 8 tahun sedangkan *test set* selama 2 tahun. Sehingga diperoleh pembagian sebagai berikut:

- `train_set`, dari 1 Maret 1998 hingga 31 Desember 2006 (~8 tahun / 3228 hari).
- `test_set`, dari 1 Januari 2007 hingga 31 Desember 2008 (2 tahun / 731 hari).

```
:: Pemotongan Train set dari    : None     sampai 20061231
:: Pemotongan Test set dari     : 20070101 sampai None

:: DIMENSI TRAIN SET DAN TEST SET
:: DIMENSI [train_set]: (3228, 14)
:: DIMENSI [test_set]: (731, 14)
```

6.2 Normalisasi dataset

Agar pelatihan berlangsung secara efisien, maka seluruh dataset (input dan output) dinormalisasikan dengan cara mengurangi dengan nilai rerata dan dibagi oleh standard deviasi (LeCun et al., 2012; Minns and Hall, 1996) sebagaimana disebutkan pada makalah Kratzert et al. (2018).

Normalisasi menggunakan scikit-learn `StandardScaler` (referensi). Parameter objek `StandardScaler` hanya mengacu pada `train_set`.

6.2.1 Train set

```
:: NORMALISASI DATAFRAME [train_set] MENJADI [train_set_scale]

:: MENAMPILKAN SAMPLE DATASET [train_set_scale]
```

[0]:

	ch_F	suhu_rerata	ch_G	ch_A	ch_C
2004-12-20	0.135552	0.652575	0.114218	1.130425	-0.467389
1999-12-17	3.800097	-2.585187	0.490628	-0.472120	1.307047
1998-09-13	-0.410231	0.883843	-0.477285	-0.225575	-0.467389
1999-12-10	-0.176324	0.305672	-0.047101	-0.061211	0.952160
2003-03-31	-0.410231	0.999478	-0.477285	-0.472120	-0.467389

6.2.2 Test set

Normalisasi pada *test set* menggunakan parameter dari *train set*.

```
:: NORMALISASI DATAFRAME [test_set] MENJADI [test_set_scale]
```

```
:: MENAMPILKAN SAMPLE DATASET [test_set_scale]
```

[0]:

	ch_B	ch_F	debit	ch_C	suhu_max
2007-07-12	-0.476333	-0.410231	-0.539789	-0.467389	-0.748948
2008-06-08	-0.476333	-0.410231	-0.529627	-0.467389	-0.028437
2008-05-19	-0.081408	0.837274	-0.516454	-0.467389	0.547973
2008-08-25	-0.476333	0.759305	-0.167470	1.803890	-0.172539
2008-05-06	-0.476333	1.071181	-0.458494	1.023137	0.115666

6.3 INPUT TENSOR

Setelah melakukan proses normalisasi, maka `train_set` harus ditransformasi ke dalam bentuk tensor 3 dimensi. Dalam pemodelan RNN dimensi input berupa tensor 3 dimensi sebagai `(batch_size, timesteps, input_dim)` (referensi).

Berdasarkan Kratzert et al. (2018), ***timesteps* yang digunakan sebesar** `TIMESTEPS=365` **hari**. Nilai tersebut digunakan untuk dapat menangkap setidaknya siklus tahunan. Pada buku ini juga akan menggunakan nilai yang sama.

Proses transformasi ini akan menggunakan modul `hidrokit.contrib.taruma.hk53` (panduan).

```
:: MENENTUKAN [TIMESTEPS]
:: [TIMESTEPS] = 365 hari
```

```
:: MENENTUKAN INPUT COLUMNS DAN OUTPUT COLUMNS
:: INPUT COLUMNS = ['ch_A', 'ch_B', 'ch_C', 'ch_D', 'ch_E', 'ch_F', 'ch_G',
'ch_H', 'suhu_min', 'suhu_max', 'suhu_rerata', 'lembab_rerata',
'lama_penyinaran']
:: OUTPUT COLUMNS = ['debit']
```

6.3.1 Train Set

```
:: TRANSFORMASI TRAIN SET
:: DIMENSI [train_set_scale] = (3228, 14)
:: TRANSFORMASI [train_set_scale] MENJADI [X_train] DAN [y_train]
:: DIMENSI [X_train] = (2863, 365, 13)
:: DIMENSI [y_train] = (2863,)
```

6.3.2 Test Set

```
:: TRANSFORMASI TEST SET
:: DIMENSI [test_set_scale] = (731, 14)
:: TRANSFORMASI [test_set_scale] MENJADI [X_test] DAN [y_test]
:: DIMENSI [X_test] = (366, 365, 13)
:: DIMENSI [y_test] = (366,)
```

7 TAHAP 4: MELATIH MODEL

Pada tahap ini, akan mempersiapkan arsitektur RNN/LSTM disertai melakukan pelatihan (*training*) model.

Demi mempersingkat buku, parameter yang digunakan adalah **dua layer dan 20 sel** dengan setiap layer diberi *Dropout* layer dengan probabilitas 10% (meniru makalah Kratzert et al. (2018)).

```
:: IMPORT TENSORFLOW.KERAS LIBRARY
```

Dalam bidang hidrologi, kasus curah hujan-limpasan dapat dievaluasi dengan berbagai metrik, metrik yang biasa digunakan adalah Nash-Sutcliffe Efficiency (referensi). Karena evaluasi metrik ingin dilakukan setiap epoch, maka dari itu dibuat fungsi khusus agar disertakan saat `compile` model.

Untuk evaluasi metrik sebenarnya sudah tersedia paket HydroErr yang dibuat oleh BYU Hydroinformatics (tersedia juga paket HydroStats untuk menelaah data hidrologi). Karena objek yang diterima pada metrik `keras` harus berupa `tensorflow` harus dibuat fungsi khusus tersendiri.

```
:: MEMBUAT FUNGSI KHUSUS METRIK (NSE, NSE_MOD, R_SQUARED)
```

Pada evaluasi metrik, digunakan empat fungsi yaitu *mean absolute error* `mae`, *Nash–Sutcliffe Efficiency* `nse` , *Modified NSE* `nse_mod`, *Coefficient of Determination* `r_squared`, dan *mean squared error* `mse` sebagai *loss function*.

```
:: PEMODELAN RNN
:: SUMMARY [model]:
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 365, 20)           2720
_________________________________________________________________
dropout (Dropout)            (None, 365, 20)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 20)                3280
_________________________________________________________________
dropout_1 (Dropout)          (None, 20)                0
_________________________________________________________________
dense (Dense)                (None, 1)                 21
=================================================================
Total params: 6,021
Trainable params: 6,021
Non-trainable params: 0
_________________________________________________________________
:: TRAINING START: 20191022 08:09
```

Pada buku ini hanya akan dilakukan sebanyak `epochs=50` dengan `batch_size=30`.

```
:: TRAINING FINISH: 20191022 08:28
```

```
:: MODEL SELESAI DILATIH
:: DURASI: 0:18:57.397631
```

8 Tahap 5: EVALUASI MODEL

Tahap ini akan membahas hasil pelatihan model. Langkah yang akan dilakukan antara lain:

- Mengevaluasi metrik yang telah tercatat dalam `history`.
- Mengembalikan hasil normalisasi menjadi nilai sebenarnya.
- Mengevaluasi *train set*
- Mengevaluasi *test set*

8.1 Metrik

Terdapat 5 metrik yang telah tercatat yaitu:

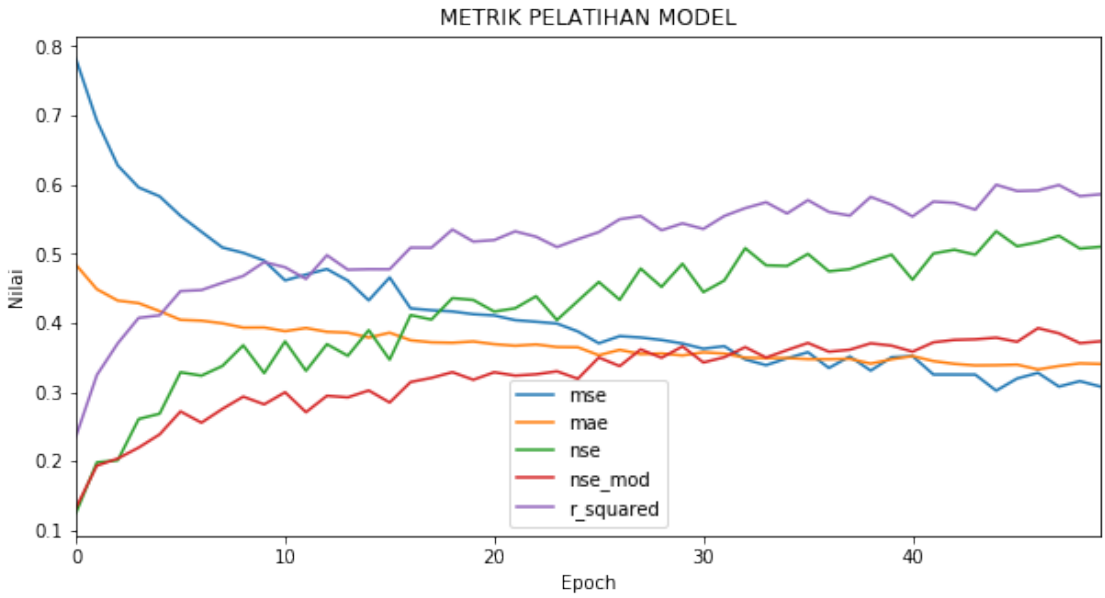
- *mean squared error* yang digunakan sebagai *loss function* `loss`: $0 \leq MSE \leq \infty$, semakin kecil semakin baik.
- *mean absolute error* `mae`: $0 \leq MAE \leq \infty$, semakin kecil semakin baik.
- *Nash-Sutcliffe Efficiency* `nse`: $-\infty \leq NSE < 1$, semakin besar semakin baik.
- *Modified NSE* `nse_mod`: $-\infty \leq NSE\_MOD < 1$, semakin besar semakin baik.
- *Coefficient of Determination* `r_squared`: $0 \leq R^2 \leq 1$, dengan nilai 1 menandakan data berkorelasi sempurna (data prediksi sama persis dengan data sebenarnya).

```
:: MENYIMPAN HISTORY METRIK DALAM BENTUK DATAFRAME
:: METRIK DISIMPAN DI [df_metric]
:: MENAMPILKAN [df_metric]
```

[0]:

	mse	mae	nse	nse_mod	r_squared
0	0.781758	0.484053	0.124985	0.132563	0.235792
1	0.692471	0.448786	0.198317	0.193541	0.324584
2	0.627708	0.432311	0.201291	0.203872	0.370504
3	0.595955	0.428561	0.261062	0.219740	0.407329
4	0.583112	0.417041	0.268653	0.238746	0.410968

```
:: GRAFIK METRIK
```



Dapat dilihat pada grafik, pada setiap epochnya model masih terus membaik, sehingga dimungkinkan untuk melanjutkan pelatihan lebih dari 50 epoch.

8.2 Object `StandardScaler` untuk kolom debit

Karena proses normalisasi data dilakukan sebelum pemisahan *train set* dan *test set*, maka harus dibuat objek `StandardScaler` baru yang mengambil atribut objek `sc` pada kolom debit (y_train/y_test).

```
:: OBJEK StandardScaler UNTUK OUTPUT [y_train] DAN [y_test]
```

8.3 Evaluasi *train set*

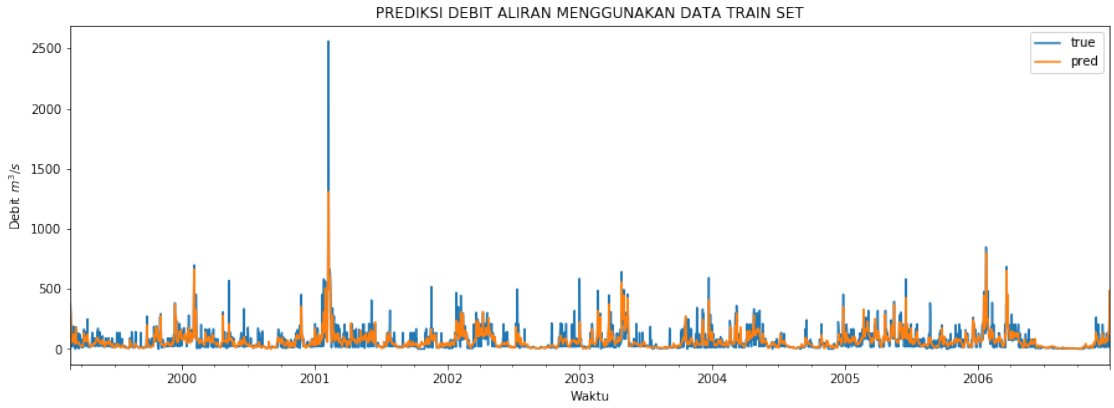
Meski mengevaluasi train set sudah tersampaikan melalui metrik diatas, saya ingin melihat bagaimana model berhasil memprediksikan data train set setiap harinya.

```
:: MENYIMPAN EVALUASI TRAIN SET [df_eval_train]
:: MENAMPILKAN [df_eval_train]
```

[0]:

	true	pred
1999-03-01	376.00	334.103973
1999-03-02	282.00	301.464172
1999-03-03	188.00	151.581390
1999-03-04	23.15	65.298645
1999-03-05	23.15	54.322853

```
:: GRAFIK DEBIT ALIRAN MENGGUNAKAN TRAIN SET
```



Sejauh ini model mampu memprediksikannya dengan cukup memuaskan. Model mampu melihat kejadian peningkatan ataupun penurunan debit. Tentunya, mengevaluasi data *train set* tidak begitu signifikan, dikarenakan model memang sudah dilatih berdasarkan data *train set*, tidak aneh jika pemodelannya memuaskan.

8.4 Evaluasi *test set*

Performa model dapat dievaluasi menggunakan data *test set*, dimana data tersebut tidak terlihat sama sekali oleh model.

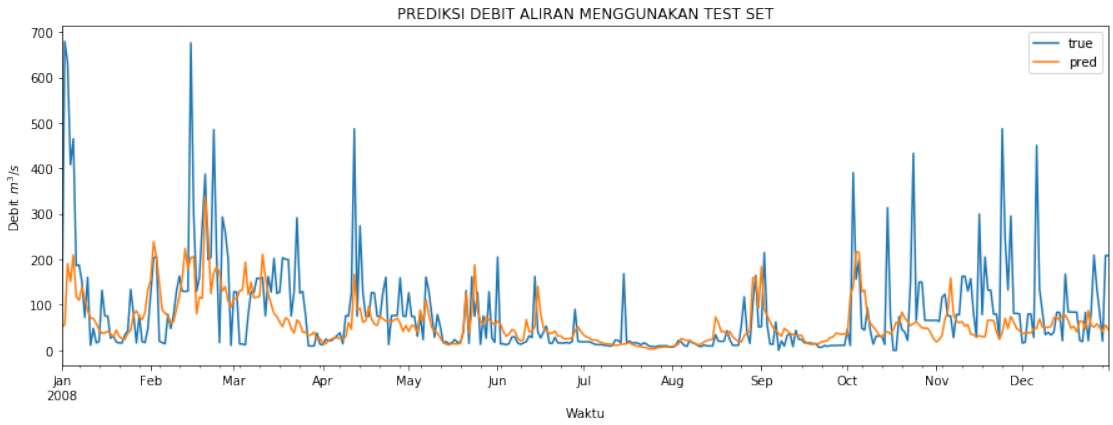
Evaluasi dilakukan dengan mengukur 5 metrik yang dilakukan serupa pada saat pelatihan (mse, mae, nse, nse_mod, dan r_squared). Perhitungan metrik akan menggunakan paket HydroErr.

```
:: MENYIMPAN EVALUASI TEST SET [df_eval_test]
:: MENAMPILKAN [df_eval_test]
```

[0]:

	true	pred
2008-01-01	76.03	50.311241
2008-01-02	679.00	56.357487
2008-01-03	632.00	189.934631
2008-01-04	408.00	151.144196
2008-01-05	465.00	209.659058

```
:: GRAFIK DEBIT ALIRAN MENGGUNAKAN TEST SET
```



Hasil prediksi debit aliran menggunakan data *test set* masih dibilang tidak tahu pasti bagus tidaknya berdasarkan visualisasi. Terlihat ada beberapa nilai yang meleset dari data observasi, akan tetapi model mampu memprediksikan kondisi kekeringan yang terjadi pada bulan juni hingga agustus.

```
:: MEMERIKSA PAKET HYDROERR
:: INSTALASI PAKET HYDROERR
  Building wheel for HydroErr (setup.py) ... done

:: MENGHITUNG METRIK DARI TEST SET

:: MENAMPILKAN METRIK TEST SET [test_metric]

[0]: mse          0.725624
     mae          0.474683
     nse          0.228164
     nse_mod      0.276037
     r_squared    0.273061
     dtype: float64


:: MEMBANDINGKAN METRIK TRAIN SET DAN TEST SET

[0]:                train      test
     mse         0.307810  0.725624
     mae         0.340651  0.474683
     nse         0.510170  0.228164
     nse_mod     0.373428  0.276037
     r_squared   0.586090  0.273061
```

Berdasarkan hasil metrik diatas, prediksi dari data *test set* tidak begitu bagus sehingga model bisa dibilang masih kurang dilatih.

9 KESIMPULAN

Jika melihat dari hasil perhitungan metrik, prediksi yang dihasilkan oleh model tergolong tidak bagus, tidak ada salah satu parameter yang dianggap memuaskan. Akan tetapi, jika dilihat dari grafik antara debit dan waktu (hidrograf), model mampu memprediksikan fluktuasi debit (kondisi kekeringan).

Dalam buku ini, hanya menampilkan bagaimana pemodelan menggunakan LSTM dilakukan saja. Masih banyak langkah yang dapat dilakukan untuk memperbaiki model sekarang seperti melakukan *parameter tuning*, *model selection*, *feature engineering*. Tentunya, data set yang digunakan harus diperiksa kembali karena dalam buku ini menggunakan asumsi sederhana bahwa data set sudah layak pakai.

10 Changelog

```
- 20191022 - 1.0.0 - Initial
```