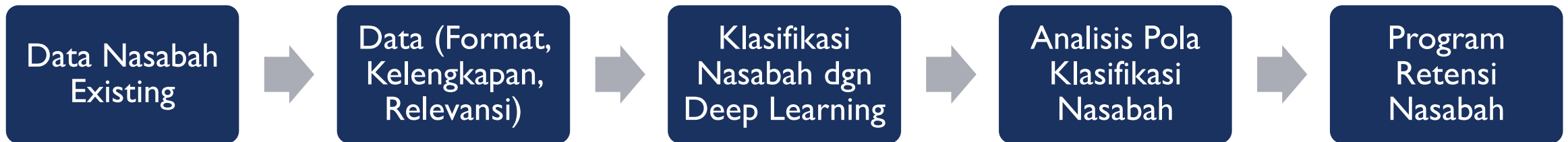

KLASIFIKASI NASABAH BANK DENGAN JARINGAN SYARAF TIRUAN (ANN) DAN K-FOLD CROSS VALIDATION

WIRA MUNGGAHA (NIM: 33218301) - [HTTPS://GITHUB.COM/IDWIRA/KLASIFIKASI-NASABAH](https://github.com/idwira/klasifikasi-nasabah)



LATAR BELAKANG

MENUJU PROGRAM RETENSI NASABAH BANK



Proyek ini adalah Klasifikasi Nasabah Bank, apakah seorang nasabah akan keluar (leave) atau tidak sebagai nasabah bank dengan berbagai parameter data yang dimiliki oleh bank, antara lain Kredit Score, Keaktifan dan Estimasi Gaji. Detail variable nya ada pada slide selanjut nya.

LATAR BELAKANG *RELATED WORKS*

*Predicting Customer Churn in
Banking Industry Using Neural
Networks*

*(Interdisciplinary Description of
Complex Systems 14(2), 116-124,
2016)*

Alisa Bilal Zorić

DATASET YANG DIGUNAKAN

[HTTPS://WWW.KAGGLE.COM/MMAJHI/CHURN-MODELLING](https://www.kaggle.com/mmajhi/churn-modelling)

Independent Variables

- Rownumber
- CustomerID
- Surname
- CreditScore
- Geography
- Gender
- Age

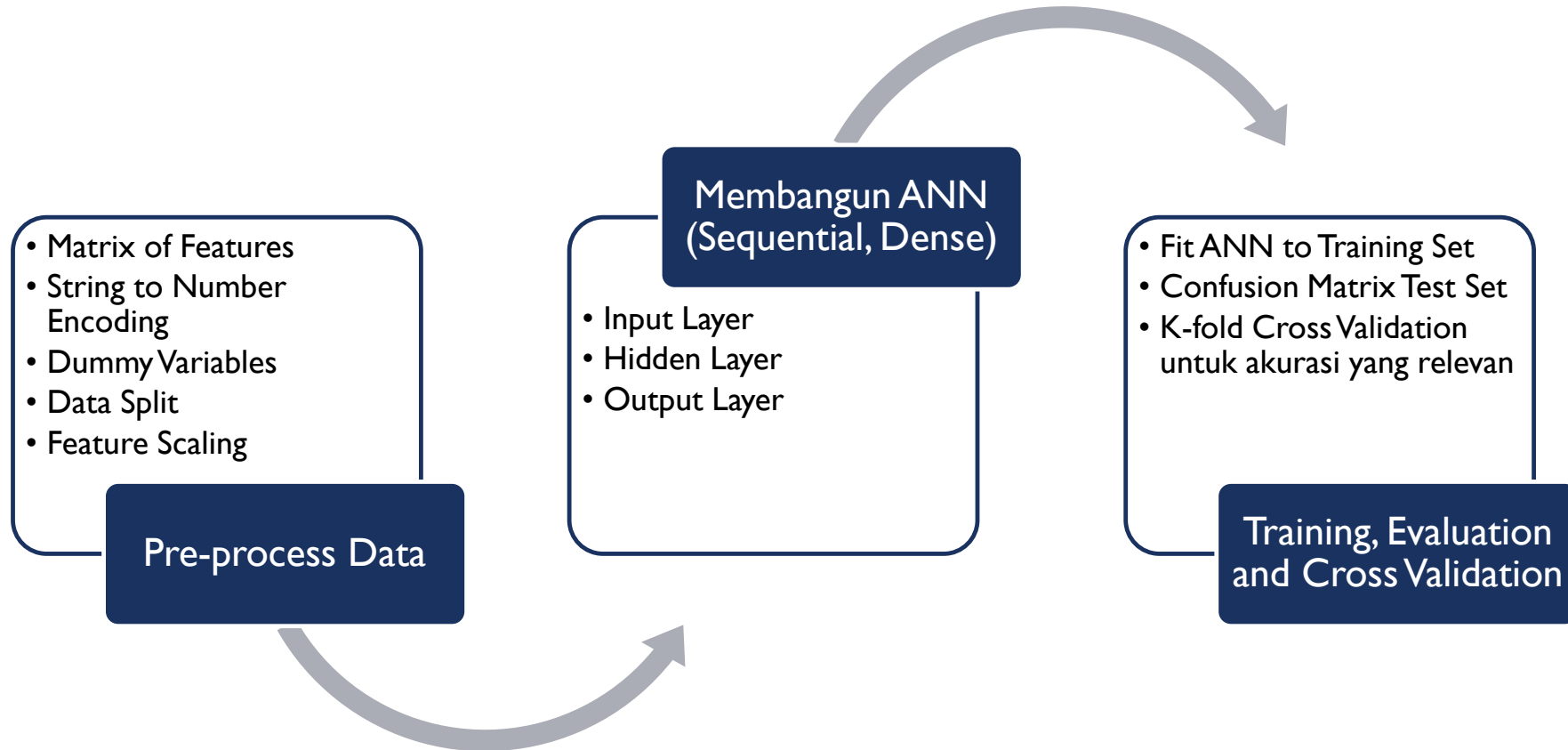
Independent Variables (cont.)

- Tenure
- Balance
- NumOfProducts
- HasCrCard
- IsActiveMember
- EstimatedSalary

Dependent Variable

- Exited (1 leave, 0 stay)

ARSITEKTUR SISTEM YANG DIBUAT



PRE-PROCESS DATASET

LANGKAH I – MEMBANGUN MATRIX OF FEATURES

Diambil indeks 3 (CreditScore) sampai dengan 12 (Estimated Salary) untuk independent variable yang relevan

dataset - DataFrame

Index	RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101349	1
1	2	15647311	Hill	608	Spain	Female	41	1	83807.9	1	0	1	112543	0
2	3	15619304	Onio	502	France	Female	42	8	159661	3	1	0	113932	1
3	4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.6	0
4	5	15737888	Mitchell	850	Spain	Female	43	2	125511	1	1	1	79084.1	0
5	6	15574012	Chu	645	Spain	Male	44	8	113756	2	1	0	149757	1
6	7	15592531	Bartlett	822	France	Male	50	7	0	2	1	1	10062.8	0
7	8	15656148	Obinna	376	Germany	Female	29	4	115047	4	1	0	119347	1
8	9	15792365	He	501	France	Male	44	4	142051	2	0	1	74940.5	0
9	10	15592389	H?	684	France	Male	27	2	134604	1	1	1	71725.7	0
10	11	15767821	Bearce	528	France	Male	31	6	102017	2	0	0	80181.1	0

PRE-PROCESS DATASET

LANGKAH 2 – ENCODE FORMAT STRING KE NUMBER

Pada dataset, data **jenis kelamin** dan **negara** masih dalam format string:

```
In [3]: X
Out[3]:
array([[619, 'France', 'Female', ..., 1, 1, 101348.88],
       [608, 'Spain', 'Female', ..., 0, 1, 112542.58],
       [502, 'France', 'Female', ..., 1, 0, 113931.57],
       ...,
       [709, 'France', 'Female', ..., 0, 1, 42085.58],
       [772, 'Germany', 'Male', ..., 1, 0, 92888.52],
       [792, 'France', 'Female', ..., 1, 0, 38190.78]], dtype=object)
```

PRE-PROCESS DATASET

LANGKAH 2 CONT.– SETELAH DI APLIKASIKAN LABEL ENCODER

Format country sudah berubah dari string ke number; 0, 1 dan 2 untuk France, Germany dan Spain:

```
In [5]: X
Out[5]:
array([[619, 0, 'Female', ..., 1, 1, 101348.88],
       [608, 2, 'Female', ..., 0, 1, 112542.58],
       [502, 0, 'Female', ..., 1, 0, 113931.57],
       ...,
       [709, 0, 'Female', ..., 0, 1, 42085.58],
       [772, 1, 'Male', ..., 1, 0, 92888.52],
       [792, 0, 'Female', ..., 1, 0, 38190.78]], dtype=object)
```


PRE-PROCESS DATASET

LANGKAH 2 CONT.– SETELAH DI APLIKASIKAN LABEL ENCODER

Format jenis kelamin sudah berubah dari string ke number; 0 untuk Female dan 1 untuk Male.
Semua variabel pada dataset sudah dalam format number

```
In [7]: X
Out[7]:
array([[619, 0, 0, ..., 1, 1, 101348.88],
       [608, 2, 0, ..., 0, 1, 112542.58],
       [502, 0, 0, ..., 1, 0, 113931.57],
       ...,
       [709, 0, 0, ..., 0, 1, 42085.58],
       [772, 1, 1, ..., 1, 0, 92888.52],
       [792, 0, 0, ..., 1, 0, 38190.78]], dtype=object)
```

PRE-PROCESS DATASET

LANGKAH 3 – MEMBUAT DUMMY VARIABLES UNTUK TIGA COUNTRY

Tiga dummy variabel untuk masing-masing country; France, Germany dan Spain:

X - NumPy array

	0	1	2	3	4	5	6	7	8	9	10	11
0	1	0	0	619	0	42	2	0	1	1	1	101349
1	0	0	1	608	0	41	1	83807.9	1	0	1	112543
2	1	0	0	502	0	42	8	159661	3	1	0	113932
3	1	0	0	699	0	39	1	0	2	0	0	93826.6
4	0	0	1	850	0	43	2	125511	1	1	1	79084.1
5	0	0	1	645	1	44	8	113756	2	1	0	149757
6	1	0	0	822	1	50	7	0	2	1	1	10062.8
7	0	1	0	376	0	29	4	115047	4	1	0	119347
8	1	0	0	501	1	44	4	142051	2	0	1	74940.5
9	1	0	0	684	1	27	2	134604	1	1	1	71725.7
10	1	0	0	528	1	31	6	102017	2	0	0	80181.1

PRE-PROCESS DATASET

LANGKAH 3 CONT. – DUA VARIABLES UTK TIGA COUNTRY

Membuang 1 dummy variable country, menghindari dummy variable trap. Pre-process data sudah selesai.

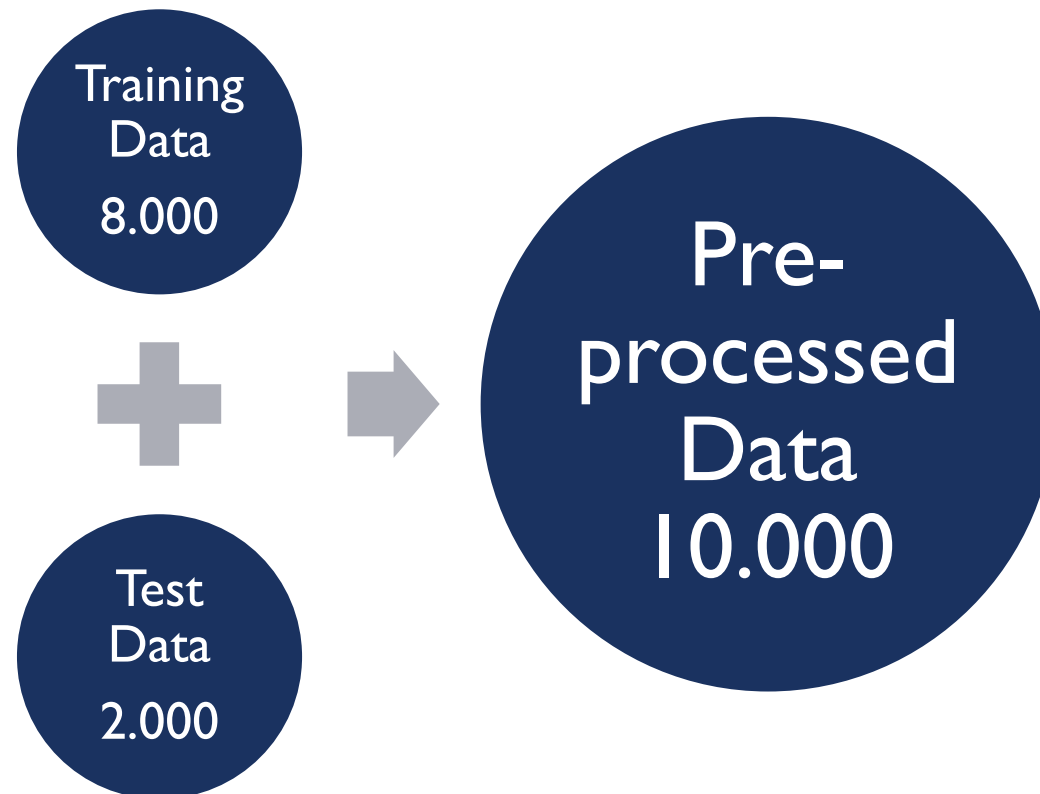
📊 X - NumPy array

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	619	0	42	2	0	1	1	1	101349
1	0	1	608	0	41	1	83807.9	1	0	1	112543
2	0	0	502	0	42	8	159661	3	1	0	113932
3	0	0	699	0	39	1	0	2	0	0	93826.6
4	0	1	850	0	43	2	125511	1	1	1	79084.1
5	0	1	645	1	44	8	113756	2	1	0	149757
6	0	0	822	1	50	7	0	2	1	1	10062.8

PRE-PROCESS DATASET

LANGKAH 4 – MEMBAGI DATA (TRAINING DAN TEST/OBSERVASI)

Membagi dataset ke dalam set training set dan set Test, train 8000 data, observasi test 2000 data



PRE-PROCESS DATASET

LANGKAH 5 – FEATURE SCALING/NORMALIZATION

SKLEARN (STANDARD SCALER)

Banyak nya komputasi, kalkulasi yang intensif

Komputasi parallel

Potensi satu independent variabel mendominasi yang lain

PRE-PROCESS DATASET

LANGKAH 5 – FEATURE SCALING/NORMALIZATON INDEPENDENT VAR.

SKLEARN (STANDARD SCALER)

X - NumPy array

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	619	0	42	2	0	1	1	1	101349
1	0	1	608	0	41	1	83807.9	1	0	1	112543
2	0	0	502	0	42	8	159661	3	1	0	113932
3	0	0	699	0	39	1	0	2	0	0	93826.6
4	0	1	850	0	43	2	125511	1	1	1	79084.1
5	0	1	645	1	44	8	113756	2	1	0	149757
6	0	0	822	1	50	7	0	2	1	1	10062.8


X_train - NumPy array

	0	1	2	3	4	5	6	7	8	9	10
0	-0.569844	1.74309	0.169582	-1.09169	-0.464608	0.00666099	-1.21572	0.809503	0.642595	-1.03227	1.10643
1	1.75487	-0.573694	-2.30456	0.916013	0.301026	-1.37744	-0.00631193	-0.921591	0.642595	0.968738	-0.748664
2	-0.569844	-0.573694	-1.1912	-1.09169	-0.943129	-1.03142	0.579935	-0.921591	0.642595	-1.03227	1.48533
3	-0.569844	1.74309	0.0355658	0.916013	0.109617	0.00666099	0.473128	-0.921591	0.642595	-1.03227	1.27653
4	-0.569844	1.74309	2.05611	-1.09169	1.73659	1.04474	0.810193	0.809503	0.642595	0.968738	0.558378
5	1.75487	-0.573694	1.29325	-1.09169	-0.177495	-1.03142	0.442535	0.809503	0.642595	-1.03227	1.63252
6	-0.569844	-0.573694	1.61283	0.916013	0.779547	-1.37744	0.304328	-0.921591	-1.55619	-1.03227	0.481496


MEMBANGUN ARTIFICIAL NEURAL NETWORK

LANGKAH 1 – 4 DARI 7


1. Inisialisasi bobot tiap node secara random ke angka kecil (mendekati 0) – Dense function



2. Input baris observasi pertama dataset ke input layer. 11 independent var. berarti 11 input node. Ditentukan 6 node di hidden layer $(11+1)/2$



3. Forward propagation dengan rectifier function utk hidden layers, dan sigmoid activation function utk output layer




4. Membandingkan hasil prediksi nya dengan hasil sebenarnya, mengukur error yang dihasilkan


MEMBANGUN ARTIFICIAL NEURAL NETWORK

LANGKAH 5 – 7 DARI 7

5. Back-propagated error langkah sebelum nya ke neural network dari kanan ke kiri, update bobot



6. Ulangi langkah 1-5, update bobot nya setiap satu batch observasi (diambil per 10 data)



7. Saat semua training set sudah melalui jaringan ANN, itu satu epoch. Run sebanyak epochs nya (di tentukan 100)

FIT ANN KE TRAINING SET

BATCH 10, EPOCHS 100

Akurasi tertinggi pada epoch 98/100 yaitu 86.36%

```
Epoch 97/100
8000/8000 [=====] - 1s 64us/step - loss: 0.3391 -
acc: 0.8606
Epoch 98/100
8000/8000 [=====] - 0s 61us/step - loss: 0.3386 -
acc: 0.8636
Epoch 99/100
8000/8000 [=====] - 0s 62us/step - loss: 0.3378 -
acc: 0.8635
Epoch 100/100
8000/8000 [=====] - 1s 65us/step - loss: 0.3379 -
acc: 0.8619
Out[8]: <keras.callbacks.History at 0x234ab3f4da0>
```

MEMBANGUN PREDIKSI DAN EVALUASI MODEL PREDIKSI HASIL TEST SET & CONFUSION MATRIX

Kecenderungan nasabah akan keluar atau tidak di dalam persen,
Karena dependent variable nya biner, maka bisa di tentukan:
Jika hasil prediksi > 0.5 maka 'True'

Dari Confusion Matrix nya, akurasi nya adalah
 $(1504 + 211) / 2000 = \mathbf{85.75\%}$

cm - NumPy array

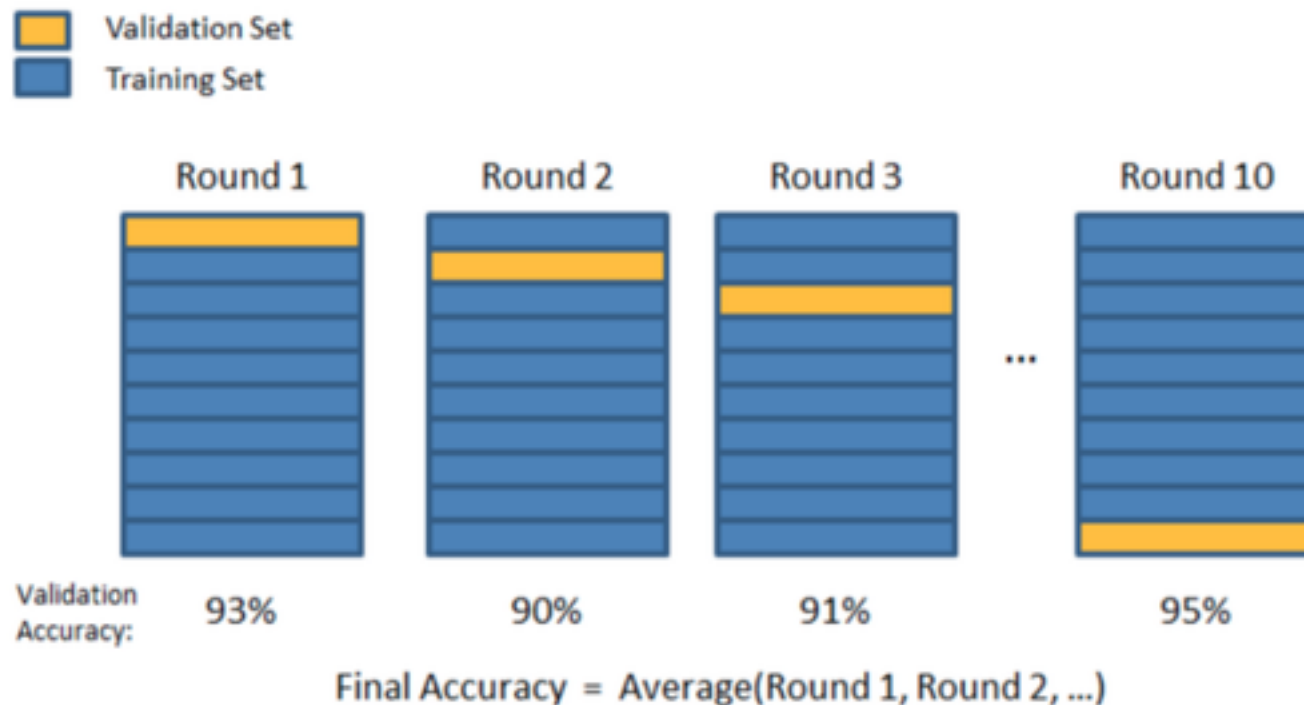
	0	1
0	1504	91
1	194	211

VALIDASI TINGKAT AKURASI

TEKNIK K-FOLD CROSS VALIDATION

Jika di ulang, maka akurasi hasil training akan bervariasi, tidak sama persis

Digunakan teknik K-fold Cross Validation untuk mendapatkan K buah kombinasi training dan test set, juga rata-rata akurasi dan standar deviasi



VALIDASI TINGKAT AKURASI

TEKNIK K-FOLD CROSS VALIDATION – MEAN DAN VARIANCE

Hasil K-fold Cross Validation

Rata-rata **84.2%**

Variance **1.17%**

mean	float64	1	0.8419999951869249
variance	float64	1	0.01167529364199761

VALIDASI TINGKAT AKURASI

TEKNIK K-FOLD CROSS VALIDATION -

Hasil K-fold Cross Validation

Rata-rata **84.2%**

Variance **1.17%**

Low Bias

Low Variance

