

TETRIS PROGRAM

Ketidaksetaraan Wewenang dalam Rumah Tangga

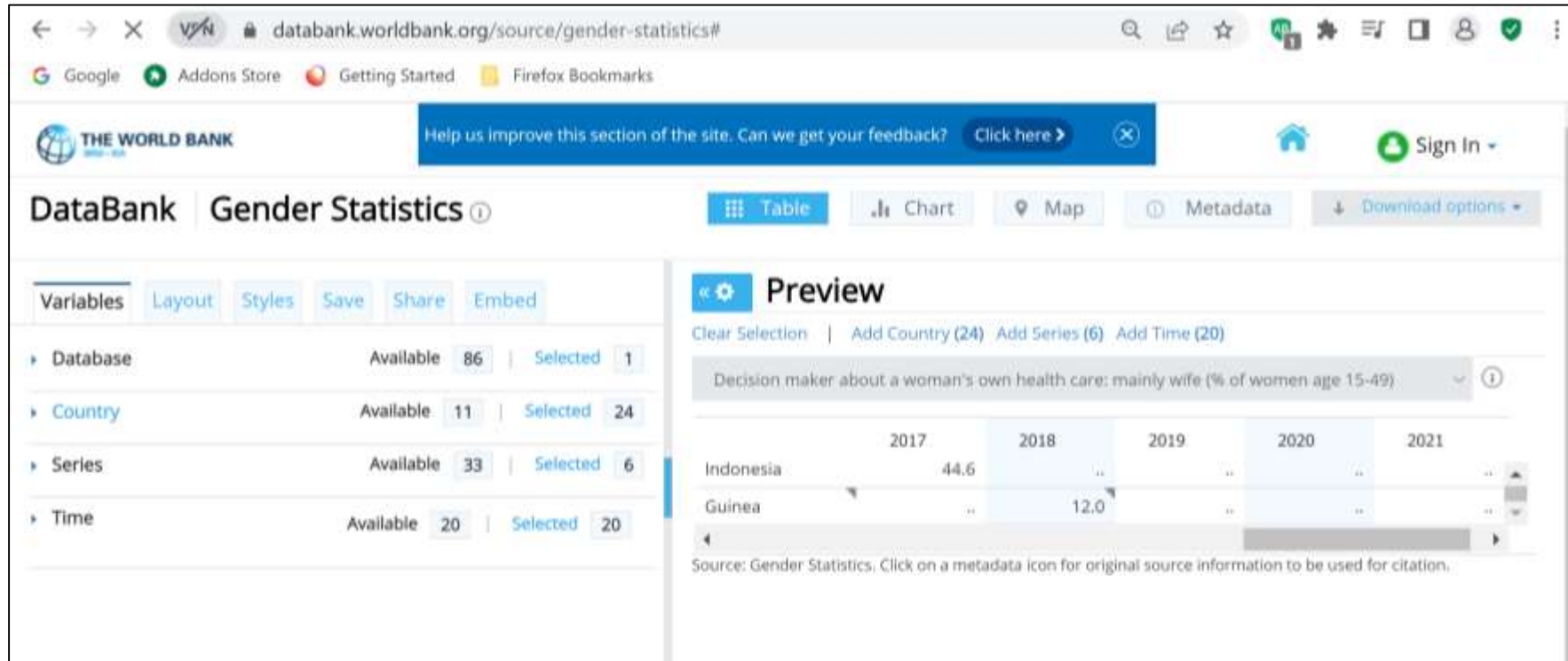
Bryan Tjandra
bryantjandra2003@gmail.com

#StackYourSkill



Step 1 - Data Collection & Data Integration

- Mengeskrak data dari world bank khususnya pada section “Norms & Decision Making” untuk periode 2002 – 2018 pada beberapa negara (termasuk Indonesia). Source: <https://databank.worldbank.org/source/gender-statistics#>



TETRIS PROGRAM

- Mengeskrak data sekunder dari world bank khususnya di Indonesia pada beberapa section seperti employment, education, health untuk menghitung korelasi dan analisa lainnya. Source: <https://databank.worldbank.org/source/gender-statistics#>

The screenshot shows the World Bank DataBank Gender Statistics interface. The sidebar on the left contains filters for Database, Country, Series, and Time. The main area displays a preview table for 'Employers, female (% of female employment) (modeled ILO estimate)' for Indonesia from 2003 to 2007.

	2003	2004	2005	2006	2007
Indonesia	1.0	1.1	1.2	1.1	1.4

Source: Gender Statistics. Click on a metadata icon for original source information to be used for citation.

TETRIS PROGRAM

- Dokumentasi mengenai series dan indicator yang digunakan terdapat pada source berikut.
<https://databank.worldbank.org/metadataglossary/all/series>

The screenshot displays the 'Metadata Glossary' page on the World Bank DataBank website. The left sidebar contains navigation links: 'DataBank Home', 'Databases', 'Create Report', 'Saved Reports', 'Saved Datasets', and 'Metadata Glossary' (which is highlighted). Below these links is a 'WHAT'S NEW' section with updates on 'Quarterly External Debt Statistics GDDS' and 'Quarterly External Debt Statistics SDDS' from July 2022, and 'Quarterly Public Sector Debt' from July 2025.

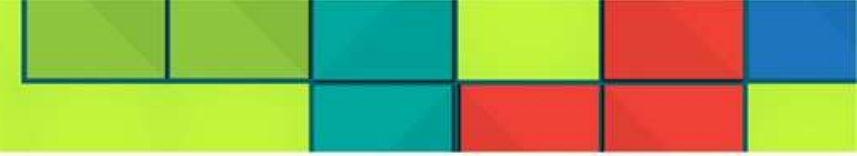
The main content area is titled 'Metadata Glossary' and features a search bar with the text 'decision maker'. Below the search bar, there are tabs for 'Select Database', 'INDICATOR', 'COUNTRY', and 'CLASSIFICATION'. A 'Remove search filter' link is present. Below the search results, there are navigation links: 'C', 'D', 'F', 'I', 'L', 'M', 'N', 'P', 'R', 'S', 'T', 'W', and 'VIEW ALL'. The results are filtered by 'D' and show 'Showing Results from 1 to 50 of 91'.

The first result is 'Decision maker about a woman's own health care: mainly husband (% of women age 15-49)'. It includes the following details:

- Dimension:** Series
- Database Name:** Gender Statistics
- Code:** SG.DMK.HLTH.HB.ZS
- Source:** Demographic and Health Surveys (DHS)
- Description:** Decision maker about women own health care: mainly husband is Percentage of currently married women aged 15-49 for whom the decision maker for their own health care is mainly the husband

A 'more' link is available at the bottom right of the description.

The second result is 'Decision maker about a woman's own health care: mainly wife (% of women age 15-49)'.



- Scope and Limitation
 - Region data yang dipakai sebagian besar Indonesia, dan juga ada beberapa negara lain yang memiliki data untuk tema tersebut
 - Range data yang dipakai berkisar dari tahun 2002 sampai dengan 2018
 - Wewenang yang dimaksudkan adalah dalam hal pengambilan keputusan dalam 3 aspek: *Visiting decision*, *Household Purchase decision*, and *Health decision*
- Data bersih, mentah, data untuk dashboard, dapat dilihat pada https://drive.google.com/drive/folders/1tQni_E-9sUHAfC8_MwhLfC7QCD8sa4F-?usp=sharing



Step 2 - Data Cleansing

- Membersihkan main data tentang *decision making*

```
] # Apply iterative imputer to impute missing value with decision making factor

imputed_countries = {}

for key,value in countries.items():
    tmp = value.iloc[:,4:].T
    imp_mean = IterativeImputer(LinearRegression(), random_state=0, max_iter=1000,
                               min_value=0, max_value=100)
    clean_tmp = pd.DataFrame(imp_mean.fit_transform(tmp),
                             columns=tmp.columns,
                             index=tmp.index)
    imputed_countries[key] = clean_tmp.T
```

```
[IterativeImputer] Early stopping criterion not reached.
[IterativeImputer] Early stopping criterion not reached.
```

```
] # Example imputed data
temp = imputed_countries[list(imputed_countries.keys())[0]]
temp.head()
```

	2003 [YR2003]	2004 [YR2004]	2005 [YR2005]	2006 [YR2006]	2007 [YR2007]	2008 [YR2008]	2009 [YR2009]	2010 [YR2010]	2011 [YR2011]
0	0.050000	0.060000	0.070000	0.080000	0.090000	0.090000	0.090000	0.130000	0.290000
1	99.689119	99.691432	99.686960	99.691262	99.696851	99.690515	99.691436	99.680781	99.690353

Steps:

- Memastikan tidak ada baris yang *all null*
- *Impute missing data* dengan metode linear regression berdasarkan aspek *decision making* masing-masing negara
- Menggabungkan *imputed data* dari data *decision making* beberapa negara yang sudah bersih menjadi satu

TETRIS PROGRAM



- Memproses data *decision making*

```
[ ] # The indicator sum must be 100, if it is not 100 , we will process it
# We group by name and code and sum it
tmp = (process_df.groupby(['Country Name','Series Code'])[process_df.columns[4:]].sum()).reset_index()
tmp.columns = ['Country Name','Series Code'] + ['Total '+str(i) for i in range(2003,2019)]
tmp = pd.concat([tmp for i in range(3)], ignore_index=True).sort_values(by=['Country Name','Series Code'])

[ ] # Standardize the indicator with sum to 100
standardize_df = pd.DataFrame(process_df.iloc[:,4:].values/tmp.iloc[:,2:].values*100,
                               columns=process_df.iloc[:,4:].columns)
standardize_df = pd.concat([imputed_decision_making[imputed_decision_making['Series Name'].str\
    .contains("Decision maker")].reset_index(drop=True)\
    .iloc[:,4:],standardize_df],axis=1)
standardize_df = pd.concat([imputed_decision_making[imputed_decision_making['Series Code']\
    .isin(["SG.VAW.REAS.Z5","SG.LAW.INDX.MR","SG.LAW.INDX.MO"])],
    standardize_df])
standardize_df = pd.concat([imputed_decision_making[imputed_decision_making['Series Name'].str\
    .contains("Women participating")],
    standardize_df]).sort_values(by=['Country Name','Series Name'])

standardize_df.head()
```

	Series Name	Series Code	Country Name	Country Code	2003 [YR2003]	2004 [YR2004]	2005 [YR2005]	2006 [YR2006]	2007 [YR2007]	2008 [YR2008]
0	Decision maker about a woman's own health	SG.DMK.HLTH.HB.ZS	Armenia	ARM	7.954604	7.624269	7.794872	7.020363	5.768802	6.37611

- Terdapat sejumlah series yang berbentuk proporsisi, sehingga jika dijumlah harus 100%
- Melakukan scaling series tersebut sehingga sesuai proporsi
- Kemudian data yang sudah di-scale digabungkan lagi menjadi satu dengan data lainnya yang tidak di-scale

TETRIS PROGRAM

- Transformasi struktur data

Mengubah year dari columns ke row, dan series dari row ke columns

```
melted_df = pd.melt(clean_decision_making, id_vars=['Series', 'Country'],
                    var_name='Year', value_name='Value')
melted_df.tail()
```

	Series	Country	Year	Value
7611	Women participating in own health care decisio...	Zimbabwe	2018	76.506525
7612	Women participating in the three decisions (ow...	Zimbabwe	2018	79.964010
7613	Women who believe a husband is justified in be...	Zimbabwe	2018	38.602836
7614	Women, Business and the Law: Marriage Indicato...	Zimbabwe	2018	80.000000
7615	Women, Business and the Law: Mobility Indicato...	Zimbabwe	2018	100.000000

```
[ ] pivoted_df = pd.pivot_table(melted_df, index=['Country', 'Year'],
                                columns='Series', values='Value')
pivoted_df.columns = ['Health decision: husband', 'Health decision: wife', 'Health decision: jointly',
                    'Visit decision: husband', 'Visit decision: wife', 'Visit decision: jointly',
                    'Purchase decision: husband', 'Purchase decision: wife', 'Purchase decision: jointly',
                    'Visit participation', 'Purchase participation',
                    'None participation', 'Health participation', 'All participation',
                    'Beating wife', 'Marriage score', 'Mobility score']
pivoted_df = pivoted_df.reset_index()
pivoted_df.head()
```

	Country	Year	Health decision: husband	Health decision: wife	Health decision: jointly	Visit decision: husband	Visit decision: wife	Visit decision: jointly
0	Armenia	2003	7.954604	33.875563	58.169833	8.184104	13.996048	81.449924
1	Armenia	2004	7.624269	35.788608	56.587123	7.921424	14.931040	83.341575
2	Armenia	2005	7.794872	34.153846	58.051282	7.959641	14.237668	83.295964
3	Armenia	2006	7.020363	28.376986	64.602652	7.107316	12.411880	85.195641
4	Armenia	2007	5.768802	21.965737	72.265461	6.024526	10.533510	86.144927

TETRIS PROGRAM



- Mentransform struktur data employment untuk pendukung dashboard

```
[ ] melted_emp_df = pd.melt(emp_df, id_vars=['Series', 'Country'],  
                             var_name='Year', value_name='Value')  
melted_emp_df.tail()
```

	Series	Country	Year	Value
1787	Self-employed, male (% of male employment) (mo...	Senegal	2018	59.84
1788	Self-employed, male (% of male employment) (mo...	Tanzania	2018	80.72
1789	Self-employed, male (% of male employment) (mo...	Uganda	2018	71.95
1790	Self-employed, male (% of male employment) (mo...	Zambia	2018	66.66
1791	Self-employed, male (% of male employment) (mo...	Zimbabwe	2018	56.98

```
[ ] pivoted_emp_df = pd.pivot_table(melted_emp_df, index=['Country', 'Year'],  
                                     columns='Series', values='Value')  
pivoted_emp_df.columns = ['Informal female', 'Informal male',  
                           'Formal female', 'Formal male']  
pivoted_emp_df = pivoted_emp_df.reset_index()  
pivoted_emp_df.head()
```

	Country	Year	Informal female	Informal male	Formal female	Formal male
0	Armenia	2003	53.880001	43.919998	46.119999	56.080002
1	Armenia	2004	53.020000	43.160000	46.980000	56.840000
2	Armenia	2005	52.200001	41.939999	47.799999	58.060001
3	Armenia	2006	50.889999	40.590000	49.110001	59.410000
4	Armenia	2007	49.169998	38.860001	50.840000	61.139999



Step 3 - Data Exploration & Data Visualisation

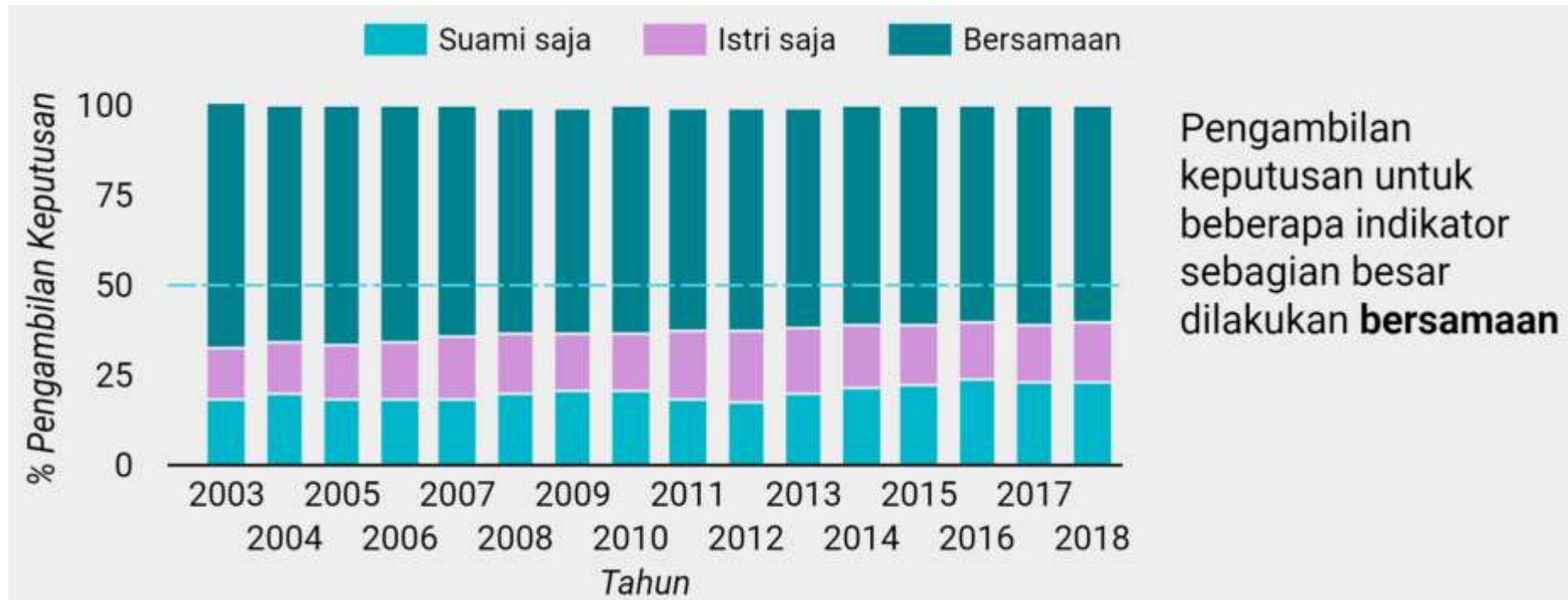
- Mencari tahu berapa banyak istri yang percaya bahwa suaminya lebih unggul.
Dari rata-rata tahun 2002-2018 menunjukkan sekitar 3 dari 10 istri masih percaya.

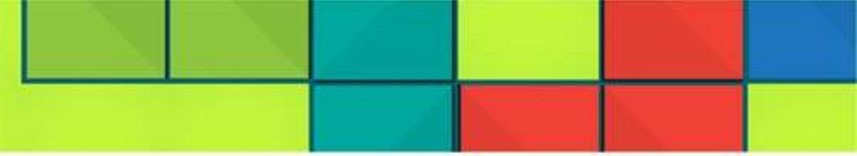


TETRIS PROGRAM

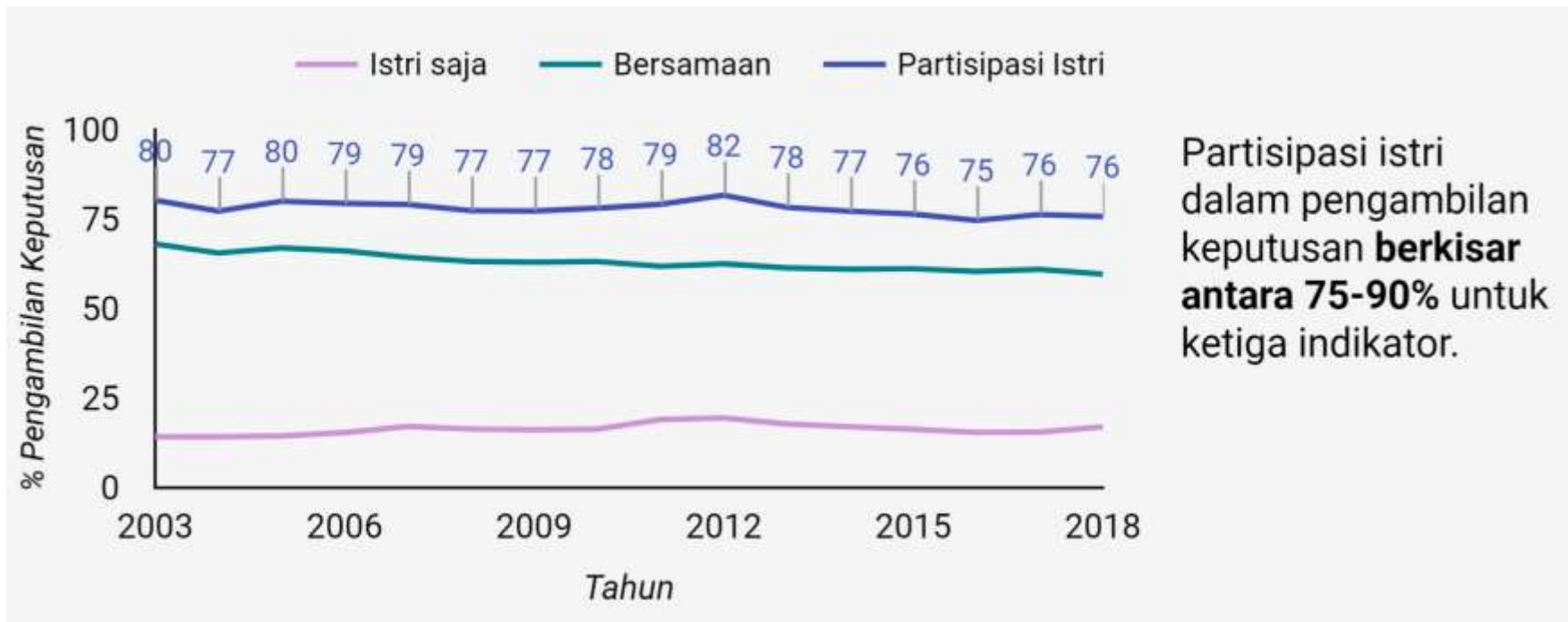


- Mencari tahu, kebanyakan keputusan rumah tangga diputuskan oleh siapa





- Mencari tahu, berapa persentase istri yang terlibat dalam pengambilan keputusan (baik sendiri / bersama)



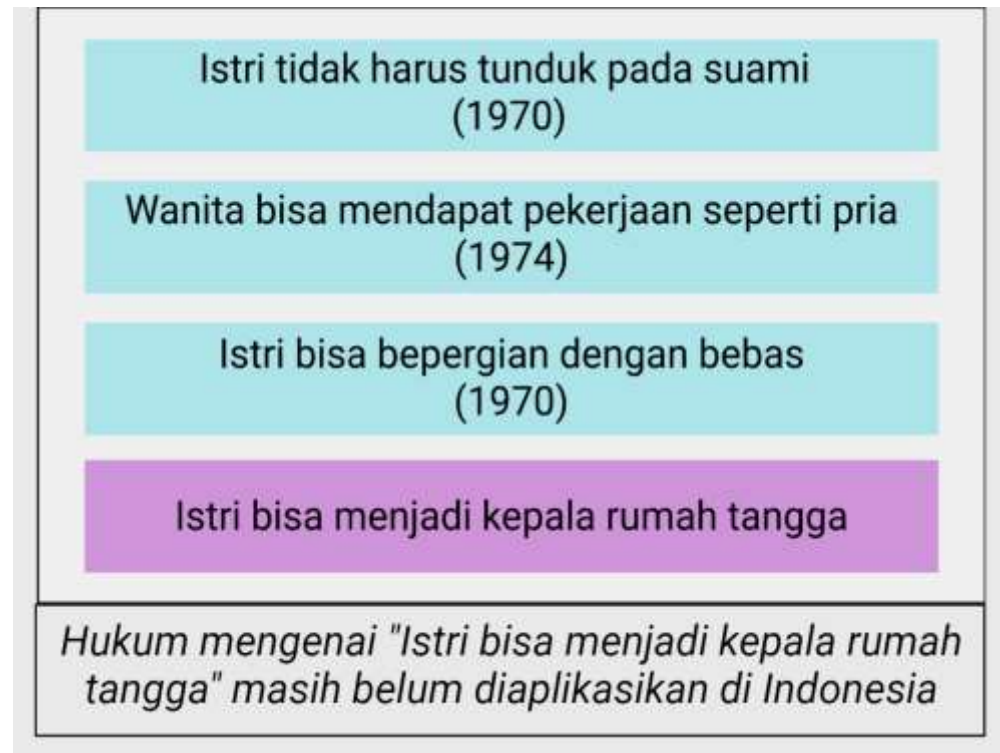


- Mencari tahu posisi negara Indonesia untuk partisipasi istri dalam mengambil keputusan pada 66 sampel negara yang diambil

Negara		Partisipasi Wanita ▾
16.	Indonesia	68,9
17.	Jordan	68,43
18.	Liberia	65,9
19.	Colombia	65,9
20.	Angola	65,4
16 - 20 / 66		< >
Indonesia menduduki posisi 16 dari 66 negara dengan partisipasi pengambilan keputusan sebanyak 68.9%		

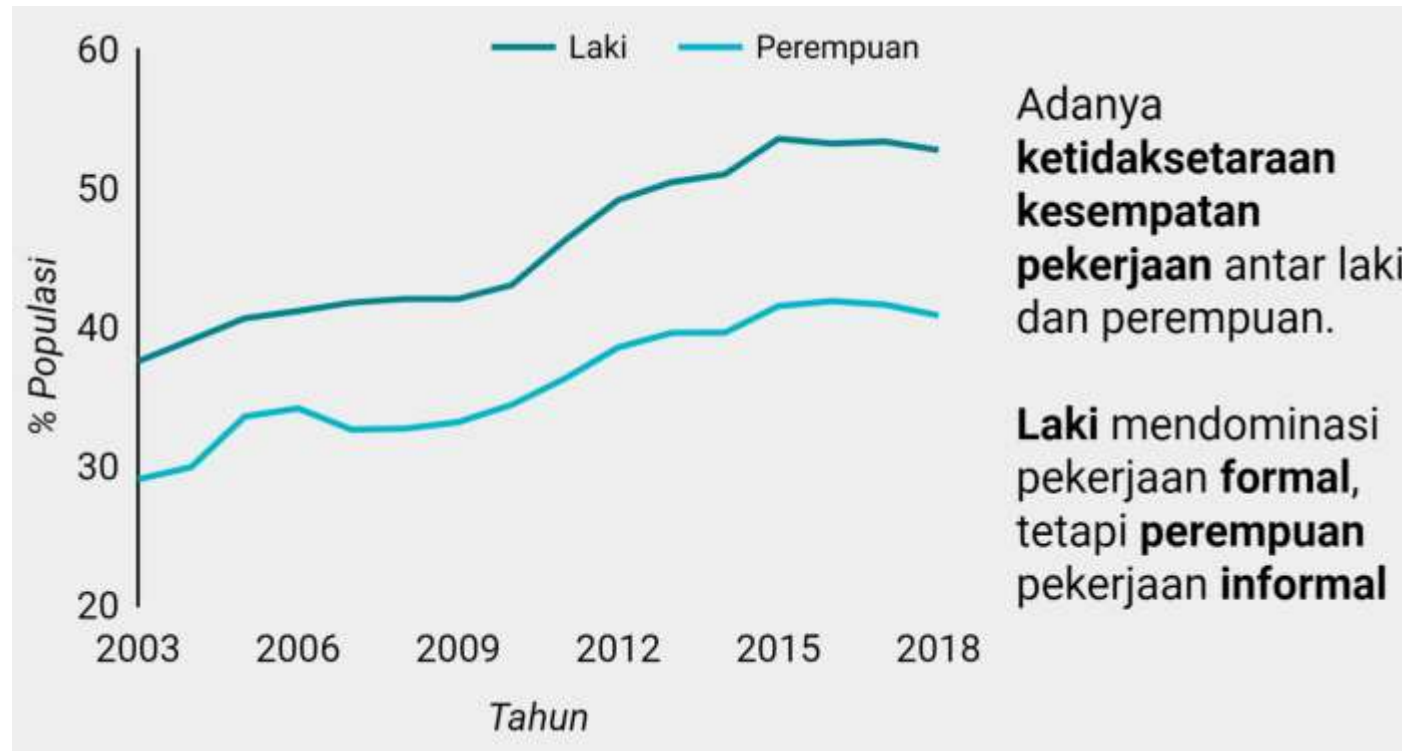


- Mencari tahu hukum tentang *equality* yang berlaku di Indonesia ataupun yang belum diterapkan

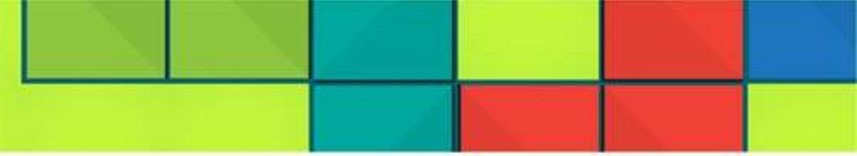




- Menganalisis perbedaan kesempatan yang diperoleh oleh laki dan wanita, salah satunya di bidang pekerjaan



TETRIS PROGRAM

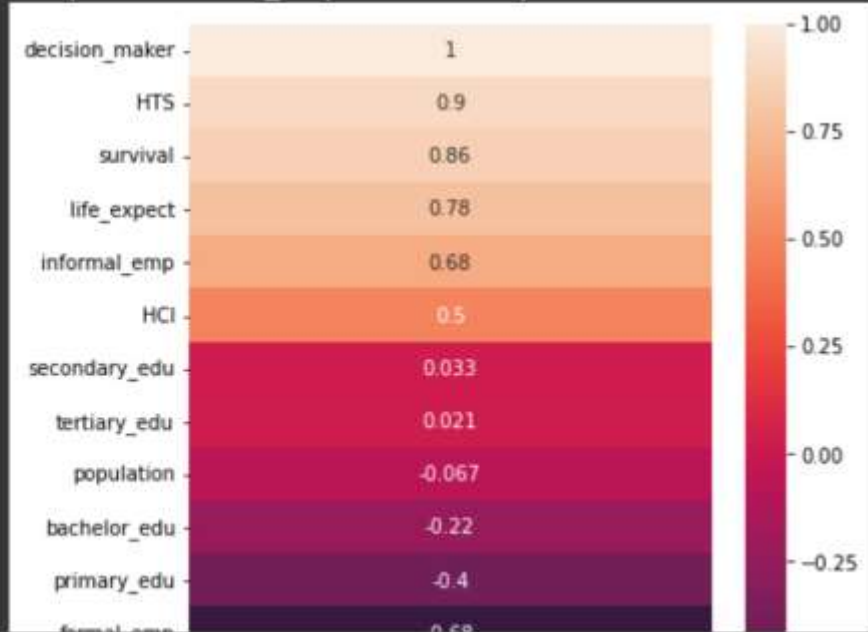


```
[ ] female = gender[gender.columns[gender.columns.str.endswith('_f')]]
male = gender[gender.columns[gender.columns.str.endswith('_m')]]
female.columns = np.arange(female.shape[1]).astype(str)
male.columns = np.arange(male.shape[1]).astype(str)

gender_df = pd.DataFrame(pd.concat([female, male], axis=0, ignore_index=True))
gender_df.columns = gender.columns[gender.columns.str.contains('_f')].str[:-2]
```

```
[ ] plt.figure(figsize=(6,8))
sns.heatmap(gender_df.corr()[['decision_maker']].sort_values(
    by='decision_maker', ascending=False), annot=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f714de71710>



- Menemukan faktor yang berkorelasi dengan partisipasi wanita dalam pengambilan keputusan

TETRIS PROGRAM

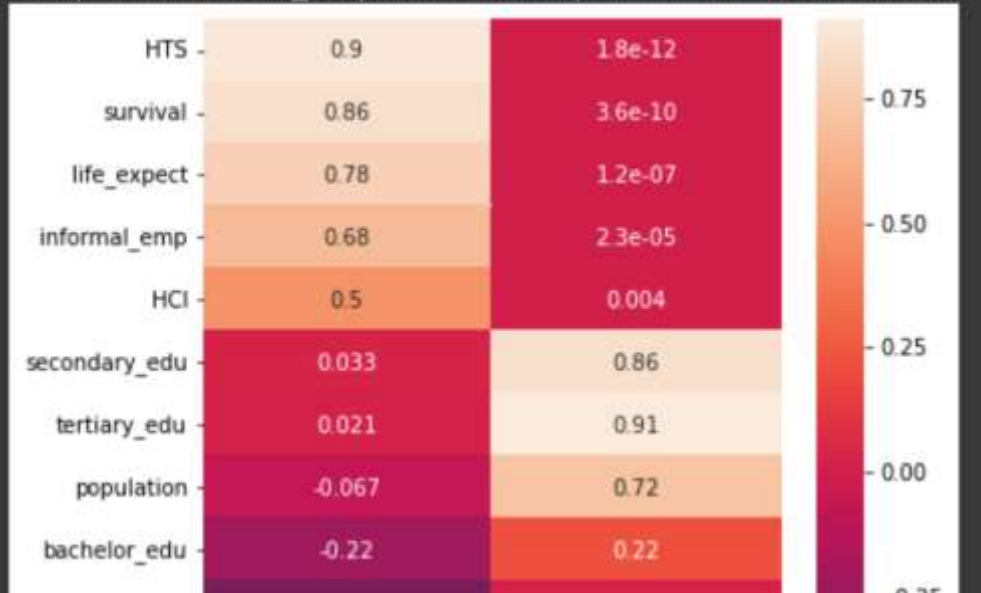
- Melakukan testing dengan p-value untuk menentukan signifikansi korelasi

```
[ ] from scipy.stats import pearsonr

new_df = {}
for i in np.setdiff1d(gender_df.columns.values, ['decision_ma
    new_df[i] = pearsonr(gender_df['decision_maker'], gender_df[
new_df = pd.DataFrame(new_df, index=['decision_maker', 'p_valu
```

```
[ ] plt.figure(figsize=(6,8))
sns.heatmap(new_df.sort_values(
    by='decision_maker', ascending=False), annot=True)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f714db7dad0>



TETRIS PROGRAM

- Melakukan prediksi time series sampai tahun 2025, untuk melihat perkiraan partisipasi wanita di masa mendatang
- Mengecek stationary untuk data partisipasi wanita, serta Autocorrelation dan partial autocorrelation

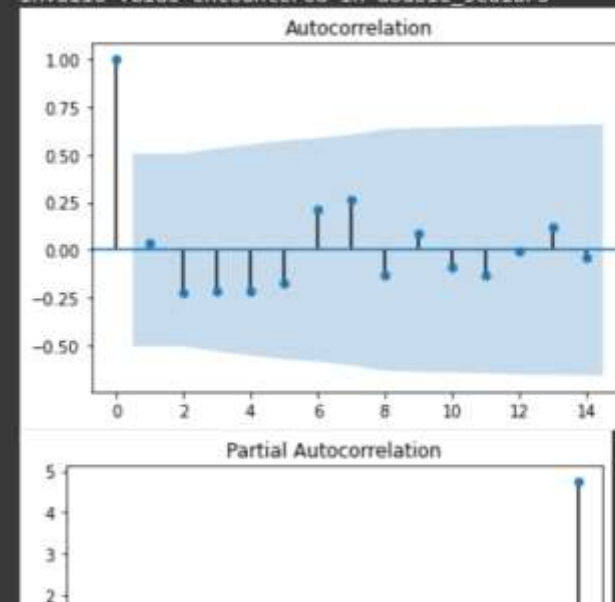
```
from statsmodels.tsa.stattools import adfuller
]
for i,j in [(indonesia_all,'All participation:'), (indonesia_none,'None participation:')]:
    result = adfuller(i.iloc[:,0].dropna())
    pvalue = result[1]
    if pvalue < 0.05:
        print(j,'Stationary')
    else:
        print(j,'Not Stationary')
```

```
All participation: Stationary
None participation: Not Stationary
```

```
] import statsmodels.api as sm
from statsmodels.graphics.tsaplots import plot_acf
from statsmodels.graphics.tsaplots import plot_pacf
```

```
plot_acf(indonesia_all.diff().dropna(), lags=indonesia_all.shape[0]-2)
plot_pacf(indonesia_all.diff().dropna(), lags=indonesia_all.shape[0])
plt.show()
```

```
invalid value encountered in sqrt
invalid value encountered in double_scalars
```



TETRIS PROGRAM

- Forecast time series menggunakan ARIMA.
- Mencari nilai AR, I , MA yang sesuai

```
[23] data_array = indonesia_none.values

d=0
for p in range(4):
    for d in range(2):
        for q in range(4):
            try:
                step = 3
                X_train, X_test = data_array[:-step], data_array[-step:]
                mod = ARIMA(X_train, order=(p,d,q))
                res = mod.fit()
                print((p,d,q),round(mean_squared_error(X_test,
                                                            res.forecast(steps=step)[0], squared=False),3))
            except:
                pass

(0, 0, 0) 0.359
(0, 0, 1) 0.279
(0, 1, 0) 0.321
(0, 1, 1) 0.262
Inverting hessian failed, no bse or cov_params available
(0, 1, 2) 1.029
(0, 1, 3) 0.917
(1, 0, 0) 0.302
(1, 0, 1) 0.253
```

TETRIS PROGRAM

Forecast time series dari tahun 2018 sampai 2025, menggunakan AR, I, dan MA yang sudah ditentukan tadi.

```
[ ] import numpy as np

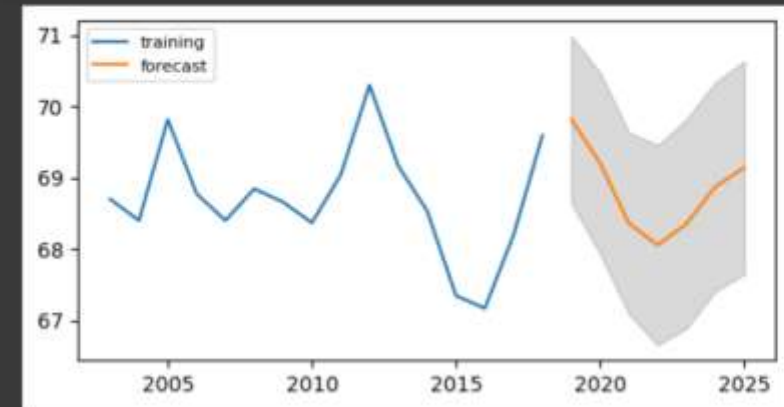
train = indonesia_all.iloc[:,0].values

model = ARIMA(train, order=(3, 0, 0))
fitted = model.fit()

# Forecast
step=7
fc, se, conf = fitted.forecast(step, alpha=0.05) # 95% conf

# Make as pandas series
idx = list(range(max(indonesia_all.index)+1, max(indonesia_all.index)+step+1))
fc_series_all = pd.Series(fc, idx)
lower_series_all = pd.Series(conf[:, 0], idx)
upper_series_all = pd.Series(conf[:, 1], idx)

# Plot
plt.figure(figsize=(6,3), dpi=100)
plt.plot(pd.Series(train, index=indonesia_all.index), label='training')
plt.plot(fc_series_all, label='forecast')
plt.fill_between(lower_series_all.index, lower_series_all, upper_series_all,
                 color='k', alpha=.15)
plt.legend(loc='upper left', fontsize=8)
plt.show()
```





Step 4 - Insight Analysis

- Sekitar 3 dari 10 wanita Indonesia masih percaya suami lebih unggul dari istrinya
- Pengambilan keputusan rumah tangga sebagian besar sudah dilakukan bersamaan
- Partisipasi istri dalam pengambilan keputusan rumah tangga cukup tinggi, berkisar antara 75-90%
- Indonesia menduduki posisi 16 dari 66 negara dengan partisipasi istri dalam pengambilan ketiga indikator keputusan sebanyak 68.9%
- Hukum mengenai "Istri bisa menjadi kepala rumah tangga" belum diaplikasikan di Indonesia.
- Masih adanya ketidaksetaraan kesempatan pekerjaan antar laki dan perempuan.
- Adanya korelasi partisipasi pengambilan keputusan dengan beberapa aspek seperti tingkat pekerjaan, usia pernikahan, pendidikan magister, *harmonized test score*, dll
- Perkiraan partisipasi istri pengambilan keputusan masih **stagnan** sampai tahun 2025

DΦLab

AYO #STACKYOURSKILL SEKARANG
dan Persiapkan Diri Menjadi Praktisi Data!

