# Configuration Manual

MSc Research Project
Msc in Fintech (MSCFTD1)

Fengly Anggrian
Student ID: X22183442

School of Computing
National College of Ireland

Supervisor: Brian Byrne

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Fengly Anggrian | | |
| **Student ID:** | X22183442 | | |
| **Programme:** | MSc in FinTech (MSCFTD1) | **Year:** | 2023/2024 |
| **Module:** | Research Project | | |
| **Lecturer:** | Brian Byrne | | |
| **Submission Due Date:** | 12 August 2024 | | |
| **Project Title:** | Exploring Conventional Banks and E-commerce Synergies in Jakarta, Indonesian Financial Landscape | | |
| **Word Count:** | 315 | **Page Count:** | 7 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:**

**Date:** 12 August 2024

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Fengly Anggrian
Student ID: X22183442

# 1 Section 1

This section will describe the analysis process carried out in the topic **"Exploring Conventional Banks and E-commerce Synergies in Jakarta, Indonesian Financial Landscape"** and how to implement it sequentially. Step by step will be described clearly as real evidence of the use of the system in this project.

# 2 System Configuration

**2.1 Software Specification:**
Tools used in this project:

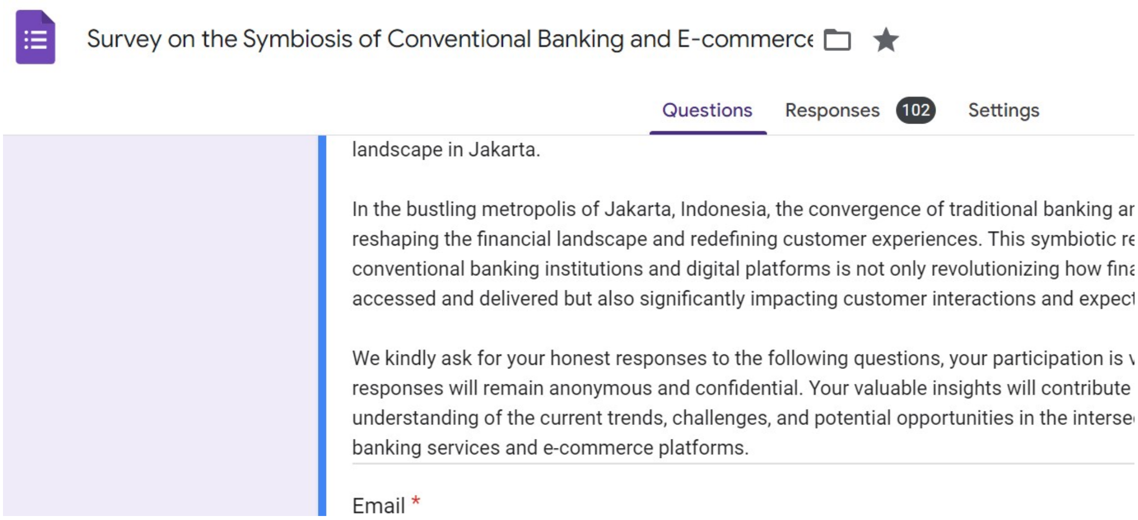
- Google form distributed to all respondents and the results processed in excel form to become a dataset.
- The dataset is then tidied up before being uploaded to Google Colab for further analysis.
- In Google Colab, all processes from data description, factor analysis to predictive modeling are executed.

**2.2 Hardware Specification:**

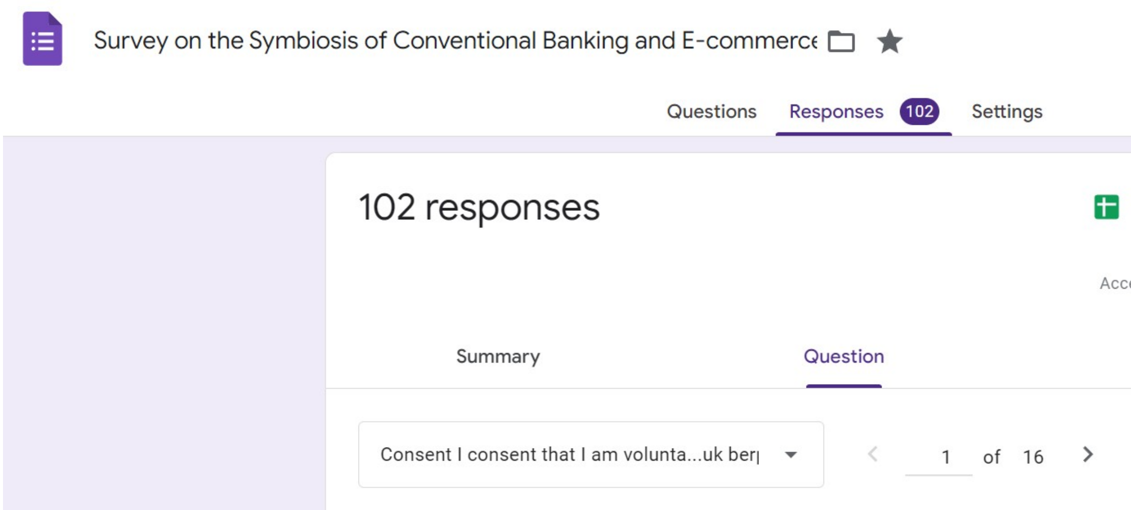
- MSI Titan 18, 512 GB SSD, 16GB RAM
- Processor: AMD Ryzen 3 4000H with Radeon Graphics

# 3 Data Generation Steps

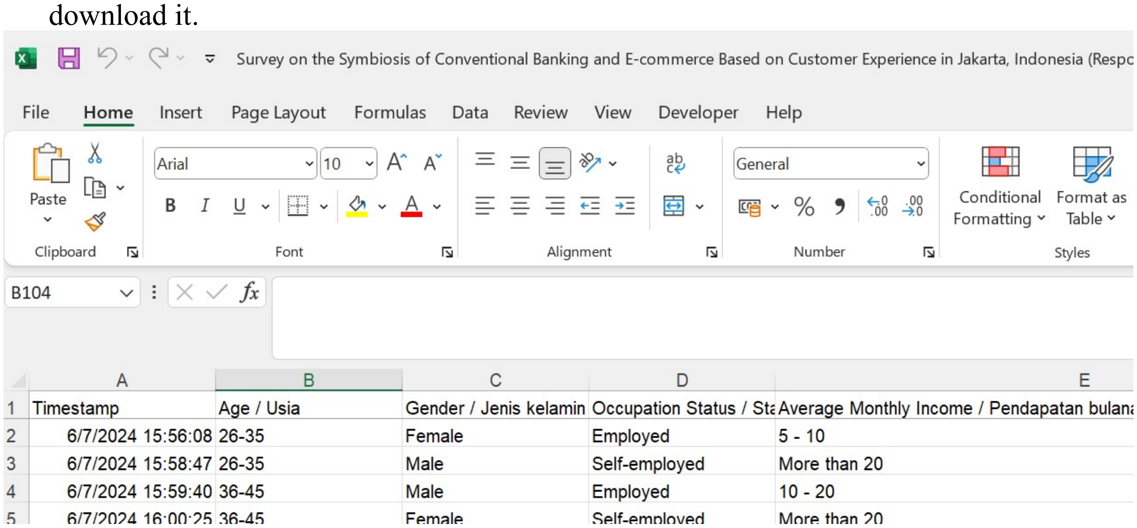
1. Open Gform in Google Chrome
2. Choose ‘Blank Form’
3. Create a survey




4. Start to make all the questions that you need.




5. After the form sent, collect the data by choose ‘responses’.
6. You can view the data in excel format by choose ‘View in Sheets’, then you can download it.




7. Open the excel data and make sure the data is tidy.

Final Code of Thesis - various latent factor.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[1] !pip install factor_analyzer semopy

    # Importing necessary libraries
    import pandas as pd
    import numpy as np
    from sklearn.preprocessing import LabelEncoder
    from google.colab import files

    # Upload the data file
```

8. Open Google Colab in Google Chrome and choose new notebook, then rename with the topic and author name.

```
[2] # Strip any leading/trailing whitespace from the col
    df.columns = df.columns.str.strip()
```

9. Display Column name to make sure there is no mistake.

```
[3] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.preprocessing import LabelEncoder
    from factor_analyzer import FactorAnalyzer
    from semopy import Model, Optimizer


    # Drop the timestamp column if it exists
    df = df.drop(columns=['Timestamp'], errors='ignore')

    # Rename the columns to Q1, Q2, ..., Q15
    question_mapping = {
        "Age / Usia": "Q1",
        "Gender / Jenis kelamin": "Q2",
        "Occupation Status / Status Pekerjaan": "Q3",
        "Average Monthly Income / Pendapatan bulanan rata-rata": "Q4",
        "How Frequently do you use traditional banking services?": "Q5",
        "How frequently do you engage in e-commerce activities?": "Q6",
        "How would you rate the convenience of traditional banking services in Indonesia? (Scale: 1-5, 1 being not convenient, 5 bei
```

10. Label the question with simple label.

```
[4] # Display the column names
    print("Column names:", df.columns.tolist())

Column names: ['Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11', 'Q12', 'Q13',
```

```
from semopy import Model
from factor_analyzer import FactorAnalyzer
from semopy import calc_stats

model_desc = '''
F1 =~ Q5 + Q7
F2 =~ Q8 + Q11 + Q12
F3 =~ Q6 + Q9 + Q10 + Q13
F4 =~ Q14 + Q15
'''

# Make Model
```

11. Propose the model for testing

```
[7] # Calculate and print fit model statistic
    stats = calc_stats(model)
    print(stats)

    # Get CFI
    cfi = stats['CFI']
    print(f'Comparative Fit Index (CFI): {cfi}')
```

```
       DoF  DoF Baseline       chi2  chi2 p-value  chi2 Baseline
Value   38            55  40.288529      0.369336     786.527733

            GFI      AGFI       NFI       TLI     RMSEA        AIC
Value  0.948777  0.925861  0.948777  0.995472  0.024419  55.210029

              BIC    LogLik
Value  128.709268  0.394986
Comparative Fit Index (CFI): Value    0.996872
Name: CFI, dtype: float64
```

```
[8] import pandas as pd
    import numpy as np

    def cronbach_alpha(df):
```

12. Execute CFA

```
[9] import pandas as pd
    from factor_analyzer import calculate_kmo, calculate_bartlett

    # Assuming df is your DataFrame and the columns Q5 to Q15 are
    question_columns = ['Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11

    # Select only the columns for the questions
    df_questions = df[question_columns]

    # Calculate KMO
    kmo_all, kmo_model = calculate_kmo(df_questions)
    print(f'KMO: {kmo_model}')
```

13. Execute EFA

```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler

# Choose relevant columns for analysis
columns_of_interest = ['Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11', 'Q12'
df = df[columns_of_interest]

# Make sure there is no mssing values
df = df.dropna()

# Data standardization
scaler = StandardScaler()
df_scaled = scaler.fit_transform(df)

# Do PCA
pca = PCA()
pca.fit(df_scaled)

# Get eigenvalues
```

14. Conduct PCA and get eigenvalue.

```python
# Getting the load factor
loadings = pca.components_.T * np.sqrt(pca.explained_variance_)

# Create a DataFrame to ease interpretation
loadings_df = pd.DataFrame(loadings, columns=[f'PC{i+1}' for i in range(len(eigen
```

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import FactorAnalysis

# Data standardization
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df)

# Conducting Factor Analysis
fa = FactorAnalysis(n_components=3)
factors = fa.fit_transform(data_scaled)

# Obtaining factor loadings
loadings = fa.components_.T

# Calculate SS Loadings
ss_loadings = np.sum(np.square(loadings), axis=0)
```

15. Get SS Loading Score

```
[19] # Ensure there are no missing values
     df.dropna(inplace=True)

     # Create factor scores based on the provided questions
     df['Conventional_Banking_Service_Quality'] = df[['Q5', '
     df['Security_of_Banking_System'] = df[['Q8', 'Q11', 'Q12
     df['Ecommerce_Effect'] = df[['Q6', 'Q9', 'Q10', 'Q13']].
     df['Readiness_to_Adopt'] = df[['Q14', 'Q15']].mean(axis=

     # Create customer experience score
     df['customer_experience'] = df[['Q9', 'Q10', 'Q14', 'Q15
```

```
[20] import statsmodels.api as sm

     # Define independent and dependent variables
     X = df[['Conventional_Banking_Service_Quality',
             'Security_of_Banking_System',
             'Ecommerce_Effect',
             'Readiness_to_Adopt']]
     y = df['customer_experience']  # The derived customer ex
```

16. Make Model for Regression and Random Forest

```
[25] !apt-get install graphviz -y
     !pip install diagrams
     !pip install python-graphviz

     from diagrams import Diagram, Cluster, Edge
     from diagrams.onprem.client import User

     graph_attr = {
         'splines': 'spline',  # Keep using splines for curved lines
         'nodesep': '0.3',      # Horizontal distance between nodes
         'ranksep': '0.5'       # Vertical distance between node ranks/rows
     }

     with Diagram("TAM Model with Relevant Factors", show=False, filename="tam_diagram_revised", direction='LR
         # External Factors
         conventional_banking_quality = User("Conv. Bank\nQuality")
         security_of_banking_system = User("Security")
         ecommerce_effect = User("E-commerce\nEase")
         readiness_to_adopt = User("Readiness")

         # TAM Constructs
         perceived_usefulness = User("Usefulness")
```

17. Visualise Tam Model

18. With Google Colab, it is possible to run a lot of analysis to get more comprehensive result, so that the result can be used to make conclusion such as the result below.

```
Regression Model
Mean Squared Error: 0.029558252962477747
R-squared: 0.9594865903451354
Feature: Conventional_Banking_Service_Quality, Impo
Feature: Security_of_Banking_System, Importance: 0.
Feature: Ecommerce_Effect, Importance: 0.3193
Feature: Readiness_to_Adopt, Importance: 0.4867

Random Forest Classifier
Accuracy: 0.6190476190476191
Classification Report:
              precision    recall  f1-score   suppo

           1       1.00      1.00      1.00
           4       0.00      0.00      0.00
           5       0.59      0.91      0.71
```