**Personal Email Financial Analysis**

1. Introduction
   1. Purpose

This project is intended to extract email data based on the subject, originator email, body to have more specific data pattern in a report manner so that can be used onward decision tasks in regular basis

* 1. Scope

Using personal email especially in domain GMAIL this project is initiated,with limitations in categorising in extracting and analysing which is about billing statements or related expenses email.

1. Background

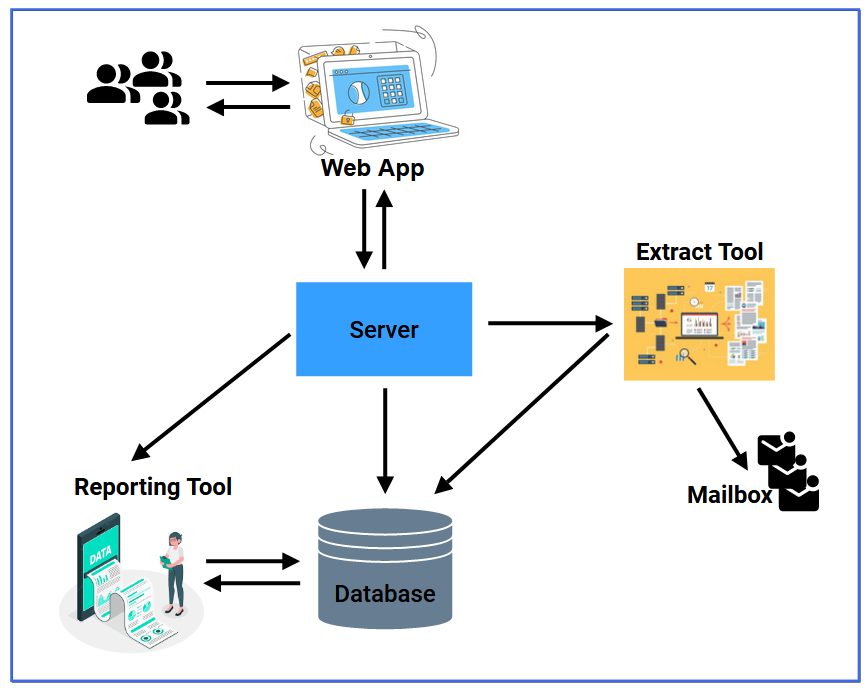
In daily operational tasks having unorganised email needs some allocated times to categorise each email manually one by one which is not efficient. Moreover it will still lead to pending and abandoned requests or solutions. To make it easier this approach will help to auto-process emails with some predefined categories.

As a part of this prototype the main goal is solely how to give the basic knowledge and understanding how the process is being chosen. Though there are many options in choosing the study case for data science exercise, still using this old and mediocre type of working in email opens up many ideas. As this project will narrow down to extract and analyse email monthly basis expenses only thus opening others to explore more in this research.

1. System Architecture

Objectively, the concept of this research is to build a prototype of a smart personalised financial report system from a mailbox. Users can analyse through a web portal to generate a financial report of total expenses reported in their personal mailbox. As seen in the diagram below, the user will provide the account name and application password generated by the domain server so that can access the tool. By having the correct credential the extraction tool will collect the clean and sufficient data to be stored in the database in a period time depending on the volume.

Next process is when the user needs to access the report, some logic from the reporting tool will display the report in the web portal.



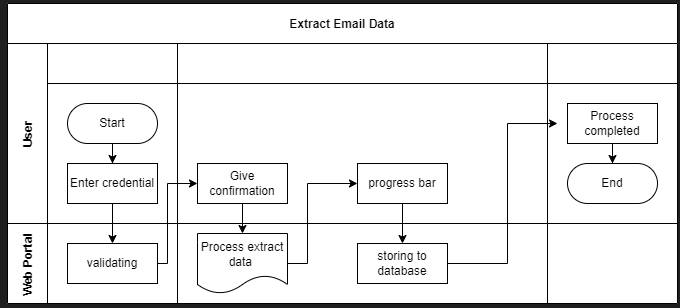
1. System Requirements
   1. Technology Requirements
      1. The project is using Python as the main code development

(TBA)

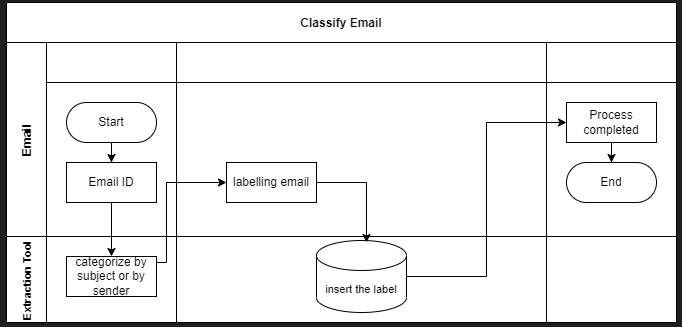
* + 1. The project using MySQL as the main database

(TBA)

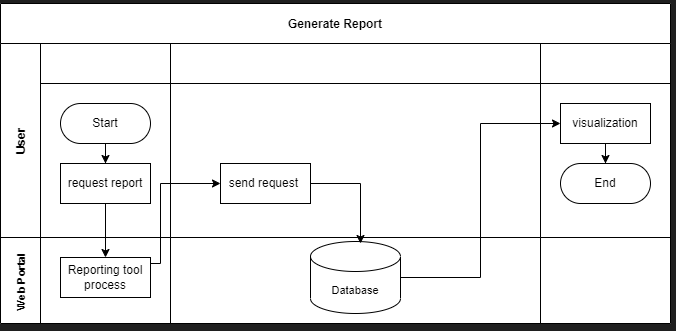
* 1. Functional Requirements
     1. The project will be able to extract the unstructured email data into database



* + 1. The project will be able to classify email into different categories



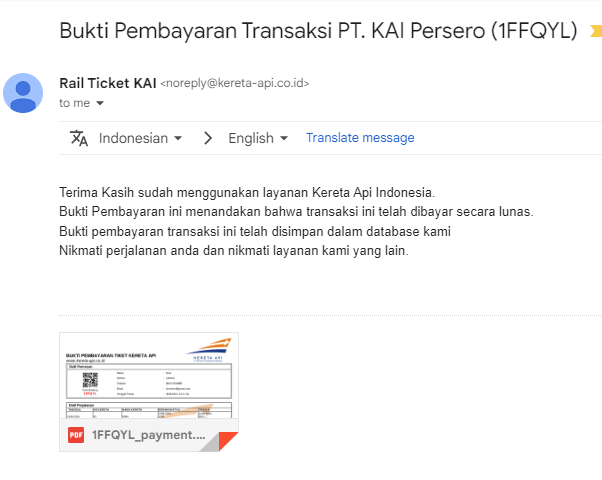
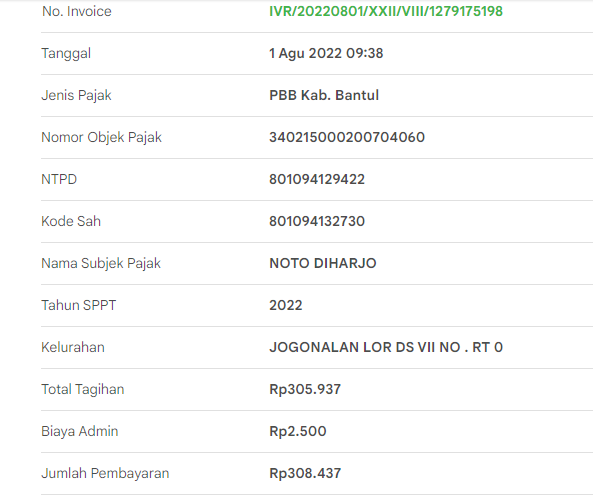
* + 1. The project will be able to count total monthly payment for each category



1. Approach
   1. Email classification

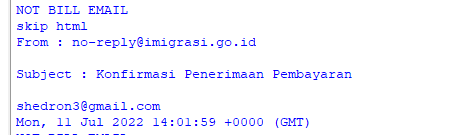
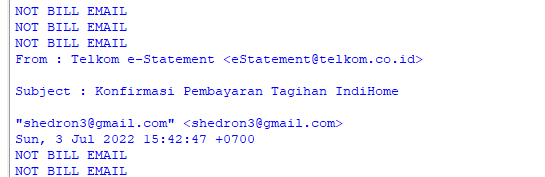
All kinds of email are being retrieved in the mailbox at once. To make it a simplified model this category is needed. Especially In the personal mailbox which is being used as the sample of this project, many types of categories exist. There are job vacancies, promotions, entertainment, hobbies and many more. Usually manual and time consuming activities are being carried out to handle this process by labelling or grouping differentiated by email id. Such logic will be implemented to separate in classifying email focused on financial related email as billing /expenses email health payment, internet payment, shopping payment and other types shown as below:





* 1. Extract unstructured email data

Using small code Python the emails are being extracted by categorising the Subject by word ‘Payment’ or ‘Billing’ or ‘Tagihan’ to get below data; originator email, subject, destination email, date and body. Those data will be used for analysis. Especially in extracting the body to find out the format in each email to be as below, only data in the body is not shown as part of collecting data. Specifically from body of each email, total paid amount and other information can be extracted



Below expected structured data

From : no-reply@imigrasi.go.id

Subject : Konfirmasi Penerimaan Pembayaran

To:shedron3@gmail.com

Date :Mon, 11 Jul 2022 14:01:59 +0000 (GMT)

Total amount paid : Rp.500.000

Category : government

From : Telkom e-Statement <eStatement@telkom.co.id>

Subject : Konfirmasi Pembayaran Tagihan IndiHome

To: "shedron3@gmail.com" <shedron3@gmail.com>

Date:Wed, 3 Aug 2022 06:51:47 +0700

Total amount paid : Rp.300.000

Category : communication

From : NOTIF UM UGM <noreply-simaster@ugm.ac.id>

Subject : [Pembayaran]

To:shedron3@gmail.com

Date:Mon, 14 Mar 2022 06:47:38 -0700 (PDT)

Total amount paid : Rp.500.000

Category : education

From : "Shopee" <info@shopee.co.id>

Subject : Pembayaranmu Berhasil Dikonfirmasi

To :<resmiatisari03@gmail.com>

Date: Wed, 22 Jul 2022 10:38:00 +0800

Total amount paid : Rp.300.000

Category : shopping

From : ZALORA <order@updates.zalora.co.id>

Subject : Pembayaran untuk order 236366957 telah kami terima

To:resmiatisari03@gmail.com

Date:Tue, 12 Jul 2022 09:01:19 +0000 (UTC)

Total amount paid : Rp.400.000

Category : shopping

1. Data Analysis Method
   1. Qualitative Analysis

This method is being used to categorise mailboxes into more specific groups based on their similar source. Below table will show results of a classification process used in this project.

| **Subject** | **Sender** | **Category** |
| --- | --- | --- |
| Konfirmasi Penerimaan Pembayaran | [no-reply@imigrasi.go.id](mailto:no-reply@imigrasi.go.id) | Government |
| [Pembayaran] | NOTIF UM UGM <noreply-simaster@ugm.ac.id> | Education |
| Konfirmasi Pembayaran Tagihan IndiHome | Telkom e-Statement <eStatement@telkom.co.id> | Communication |
| Pembayaran untuk order 236366957 telah kami terima | ZALORA <order@updates.zalora.co.id> | Shopping |

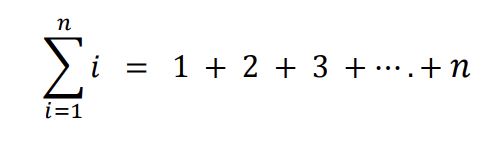
Government category is defined by sender email containing “go.id” , meanwhile Education category uses “ac.id” identifier. Other categories will be classified using master data.

* 1. Quantitative Analysis

This method is being used to count total monthly expenses based on billing notified to the mailboxes. By using the same categorization, the total amount will be calculated shown as below table:

| **To** | **Date** | **Total**  **Amount Paid** | **Category** |
| --- | --- | --- | --- |
| shedron3@gmail.com | Mon, 11 Jul 2022 14:01:59 +0000 (GMT) | Rp.500.000 | Government |
| shedron3@gmail.com | Mon, 14 Mar 2022 06:47:38 -0700 (PDT) | Rp.500.000 | Education |
| shedron3@gmail.com | Wed, 3 Aug 2022 06:51:47 +0700 | Rp.300.000 | Communication |
| resmiatisari03@gmail.com | Tue, 12 Jul 2022 09:01:19 +0000 (UTC) | Rp.400.000 | Shopping |
| resmiatisari03@gmail.com | Wed, 22 Jul 2022 10:38:00 +0800 | Rp.300.000 | Shopping |

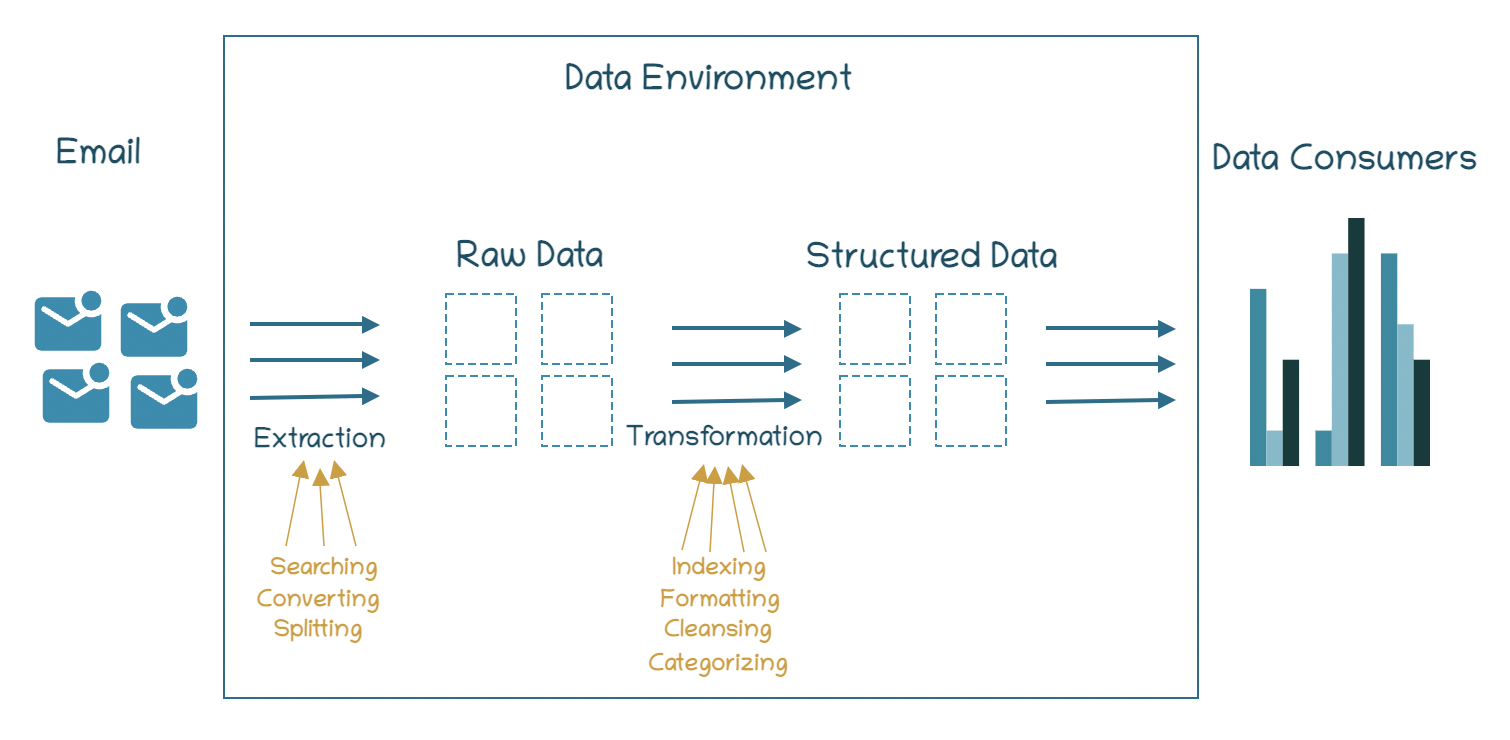
The sum of whole numbers can be represented in the following manner:



For email resmiatisari03@gmail.com, total expenses along July 2022 is

Rp.400.000 + Rp.300.000 = Rp.700.000

1. System Design



* 1. Data Extraction
     1. Search

Searching the targeted email is using Subject search (‘Pembayaran’) by intersectional of incomplete email (waiting confirmation payment) for 2 targeted mailboxes. With the current designated code the same subject cannot read the data with the same type and with different mailboxes raise an error when parsing the data. This is also a room for optimizing in searching.

* + 1. Convert

Avoiding the incompatible codec, there is limitation in searching all targeted email. Thus the next study case can be used to optimize the code for searching it. In this study, extraction is being completed for text and html based. For pdf attachment still being left for the next study case.

* + 1. Split

Email data from GMAIL will be extracted using Google Auth API. The extracted data is then being read by the program and split for every word. The amount of payment will be recognized using words: ‘Total’, ‘Jumlah’, ‘dibayar’,’dibayarkan’,’Pembayaran’ Please see below the raw data example for some payments after being splitted.

| **Payment Channel** | **Raw Data** | **Amount** |
| --- | --- | --- |
| Tokopedia - Angsuran | ['Halo', 'Angel', 'Berta,', 'Selamat,', 'pembayaran', 'Angsuran', 'Kredit', 'kamu', 'berhasil!', 'Terima', 'kasih', 'sudah', 'mempercayakan', 'Tokopedia', 'untuk', 'memenuhi', 'kebutuhanmu.', 'Berikut', 'detail', 'transaksimu:', 'No.', 'Invoice', 'IVR/20220830/XXII/VIII/1316675685', 'Tanggal', '30', 'Agu', '2022', '08:07', 'Nomor', 'Pelanggan', '030221004670', 'Nama', 'Pelanggan', 'ANGEL', 'BERTA', 'DESI', 'SURYANTI', 'Nomor', 'Referensi', '1073979175', 'Tagihan', 'ke', '9', 'Total', 'Tagihan', '48', 'Admin', 'Rp6.500', 'Jatuh', 'Tempo', '24-SEP-22', 'Total', 'Tagihan', 'Rp3.882.000', 'Total', 'Bayar', 'Rp3.888.500', 'Bayar', 'tagihan', 'rutin', 'lebih', 'mudah', 'dan', 'cepat', 'pakai', 'fitur', 'Langganan!', 'Mulai', 'Berlangganan', 'Butuh', 'bayar', 'tagihan', 'apa', 'lagi?', 'Semua', 'bisa', 'bayar', 'di', 'Tokopedia!', 'Pulsa', 'Isi', 'pulsa', 'dan', 'paket', 'data,', 'tinggal', 'klik!', 'Pascabayar', 'Bayar', 'tagihan', 'HP', 'tanpa', 'ribet', 'IPL', 'Bayar', 'Iuran', 'Pemeliharaan', 'Lingkungan', 'di', 'Tokopedia', 'E-mail', 'ini', 'dibuat', 'otomatis,', 'mohon', 'tidak', 'membalas.', 'Jika', 'butuh', 'bantuan,', 'silakan', 'hubungi', 'Tokopedia', 'Care.', 'Download', 'Aplikasi', 'Tokopedia', 'Ikuti', 'Kami', '2009-2021,', 'PT', 'Tokopedia', 'Click', 'here', 'to', 'unsubscribe'] | The application will find the amount using below syntax:  index(‘Total’)+2=Rp3.888,500 |
| Telkom - Indihome | ['Yth', 'Bapak/Ibu', 'ANGEL', 'BERTA', 'DESI', 'SURYANTI', 'Nomor', 'ID', 'Pelanggan', ':', '141129109200', 'Kami', 'sampaikan', 'terima', 'kasih', 'atas', 'pembayaran', 'tagihan', 'IndiHome', 'Bapak/Ibu', 'untuk', 'Bulan', '202209', 'sebesar', 'Rp.', '314.130', 'terbilang', 'tiga', 'ratus', 'empat', 'belas', 'ribu', 'seratus', 'tiga', 'puluh', 'rupiah.', 'Selamat', 'menggunakan', 'layanan', 'IndiHome.', 'Untuk', 'percepatan', 'penanganan', 'gangguan', 'layanan,', 'pembayaran', 'tagihan', 'segera', 'install', 'apps', 'Myindihome,', 'silahkan', 'unduh', 'apps', 'MyIndiHome', 'di', 'Play', 'Store', 'untuk', 'Android,', 'dan', 'di', 'App', 'Store', 'untuk', 'IOS.', 'Email', 'ini', 'bersifat', 'informasi', 'dan', 'tidak', 'dapat', 'di-reply,', 'bila', 'ada', 'hal', 'lain', 'yang', 'ingin', 'Bapak/Ibu', 'tanyakan', 'tentang', 'layanan', 'Telkom', 'dapat', 'menghubungi', 'kami', 'melalui', ':', 'Contact', 'Center', 'Telkom', ':', '147', 'Twitter', ':', 'twitter.com/IndiHomeCare', 'Facebook', ':', 'www.facebook.com/IndiHomeCare', 'Instagram', ':', 'www.instagram.com/indihomecare/', 'Email', ':', 'CustomerCare@telkom.co.id', 'Apps', ':', 'myIndihome', 'Demikian', 'disampaikan,', 'atas', 'perhatian', 'Bapak/Ibu', 'kami', 'ucapkan', 'terima', 'kasih.', 'Hormat', 'kami,', 'Customer', 'Care', 'PT', 'Telekomunikasi', 'Indonesia,', 'Tbk', '---------------------DISCLAIMER', ':This', 'electronic', 'mail', 'and/', 'or', 'any', 'files', 'transmitted', 'with', 'it', 'may', 'contain', 'confidential', 'or', 'copyright', 'information', 'of', 'PT', 'Telkom', 'Indonesia', '(Persero', 'Tbk).', 'and/', 'or', 'its', 'Subsidiaries.', 'If', 'you', 'are', 'not', 'an', 'intended', 'recipient,', 'you', 'must', 'not', 'keep,', 'forward,', 'copy,', 'use,', 'or', 'rely', 'on', 'this', 'electronic', 'mail,', 'and', 'any', 'such', 'action', 'is', 'unauthorized', 'and', 'prohibited.', 'If', 'you', 'have', 'received', 'this', 'electronic', 'mail', 'in', 'error,', 'please', 'reply', 'to', 'this', 'electronic', 'mail', 'to', 'notify', 'the', 'sender', 'of', 'its', 'incorrect', 'delivery,', 'and', 'then', 'delete', 'both', 'it', 'and', 'your', 'reply.', 'Finally,', 'you', 'should', 'check', 'this', 'electronic', 'mail', 'and', 'any', 'attachments', 'for', 'the', 'presence', 'of', 'viruses.', 'PT', 'Telkom', 'Indonesia', '(Persero', 'Tbk).', 'accepts', 'no', 'liability', 'for', 'any', 'damages', 'caused', 'by', 'any', 'viruses', 'transmitted', 'by', 'this', 'electronic', 'mail.'] | The application will find the amount using below syntax:  index(‘Rp.’)+1=314.130 |
| Astra Pay | ['Automatic', 'Email', 'Hai', 'Angel', 'Berta', 'Desi', 'Suryanti,', 'Terima', 'kasih!Transaksi', 'kamu', 'berhasil', 'Anda', 'telah', 'berhasil', 'melakukan', 'pembayaran', 'QR:', 'Rp', '645,599.00', 'Jika', 'kamu', 'tidak', 'melakukan', 'transaksi', 'di', 'atas,', 'segera', 'hubungi', 'Customer', 'Service', 'AstraPay', '(1500793)', 'DETAIL', 'TRANSAKSI', 'No.', 'Transaksi', ':', 'INV/QRI/001/220618/1408189YJIP', 'Status', 'Transaksi', ':', 'BERHASIL', 'Id', 'Customer', ':', '08\*\*\*\*\*\*\*\*\*\*\*\*83', 'No', 'Referensi', ':', '8210001511653', 'Id', 'Merchant', ':', 'ID2021071994093', 'KODE', 'MERCHANT', ':', '210000600', 'Nama', 'Merchant', ':', 'Astra', 'Dht', 'Yogya', 'Magelang', 'Lokasi', 'Merchant', ':', 'SLEMAN', 'Id', 'Terminal', ':', 'A01', 'JUMLAH', ':', 'Rp', '645,599.00', 'Total', 'Bayar', 'Rp', '645,599.00', 'Punya', 'Pertanyaan?', 'Email', 'customercare@astrapay.com', 'Call', 'Center', '1500-793'] | The application will find the amount using below syntax:  index('Total’)+3=645,599.00 |

* 1. Data Transformation
     1. Indexing

Indexing being used when the targeted column (amount) is identified manually in the raw data after a unique keyword. But in this case there are many keywords for different notification emails. In the future this study can be researched more..

* + 1. Formatting

Extracted email data will be formatted into the same format. For example, the amount starts with ‘Rp’ and ends with ‘.00’ but other amounts from other emails have different formats. Formatting the data will make them more structured.

* + 1. Cleansing

Keyword searching is being supported with the cleansing method since unintended email will be loaded. Sample keyword is ‘pembayaran’ there will be ‘Konfirmasi Pembayaran’ and ‘Menunggu Pembayaran’ thus this cleansing needed here to exclude ‘Menunggu Pembayaran’ because it’s not targeted email.

* + 1. Categorize
       1. Categorize by Subject

Email will be classified based on Subject containing below words:

* Health insurance
* Water, electricity bill → Public Utilities
* Postpaid, Indihome → Communication and internet
* PBB → Tax
* Mandiri Finance, credit card → Loan
* ZALORA → clothing & beauty
  + - 1. Categorize by Email domain

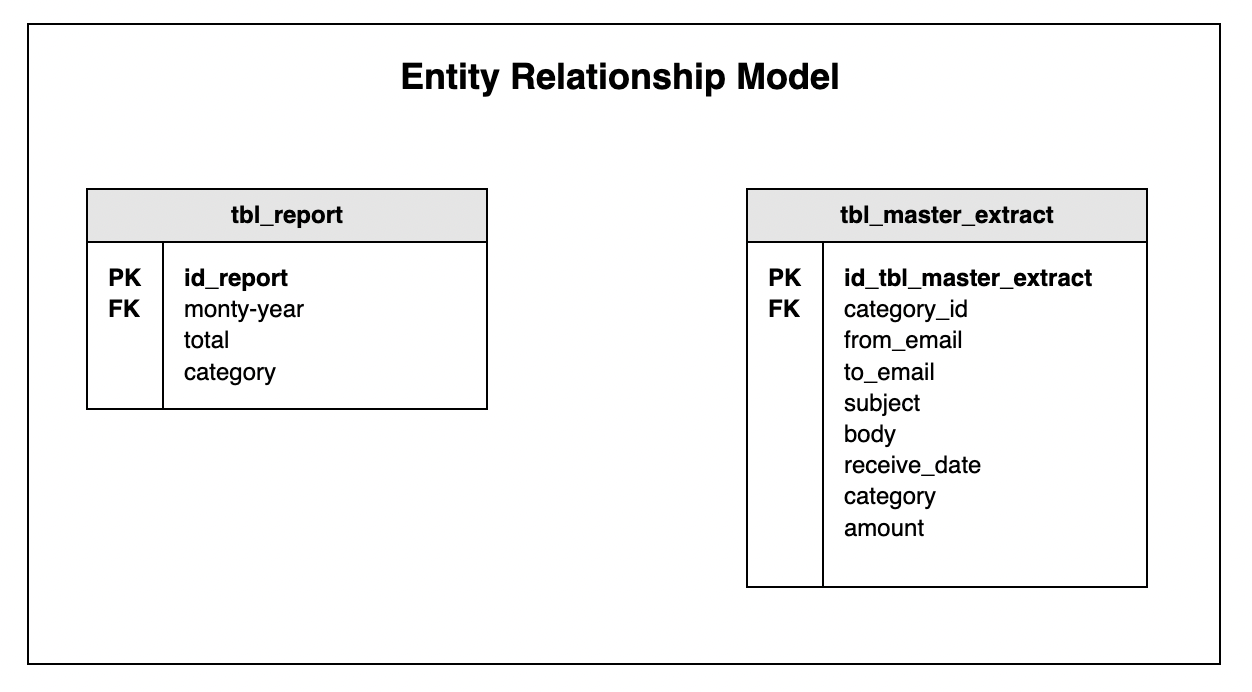
Email will be classified based on email domain containing below words:

* Domain shopee.co.id categorized as Shopping
* Domain lazada.co.id categorized as Shopping

* 1. Database Design

The application will use a MySQL database to store the structured data with amount and category.

Please see below the ER database design.



* 1. Report Design

At the end of the flow, a report will be generated using graph or table view.

1. Implementation

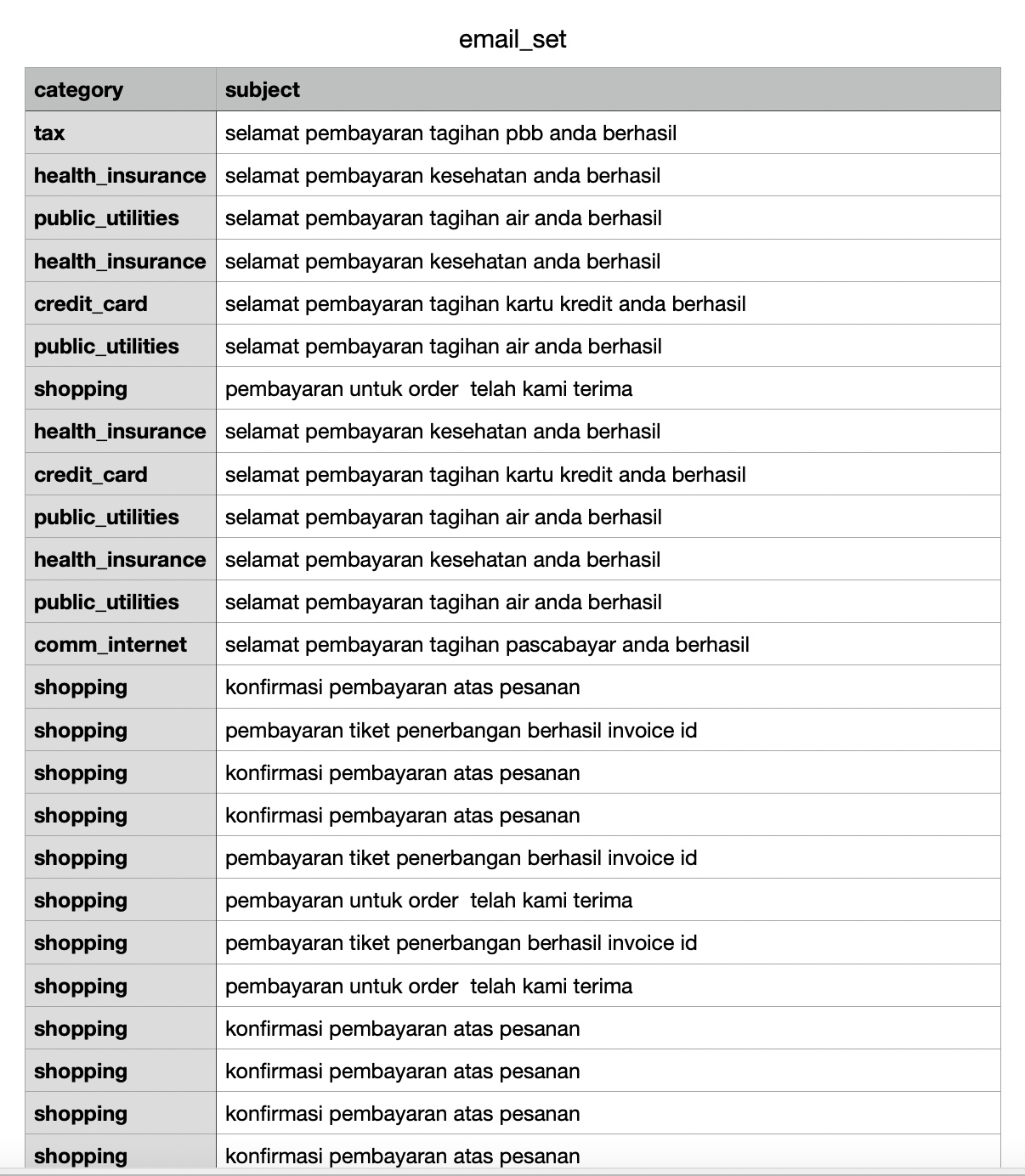
During the implementation, there are some steps which are already done from changing the unstructured data to structured data.

* 1. Data Extraction

By using the Google Auth API, the subject of email data was extracted and became raw data. The extraction process produces 189 rows of data into the csv file. Then we labeled the category manually with 7 label for categories with below details:

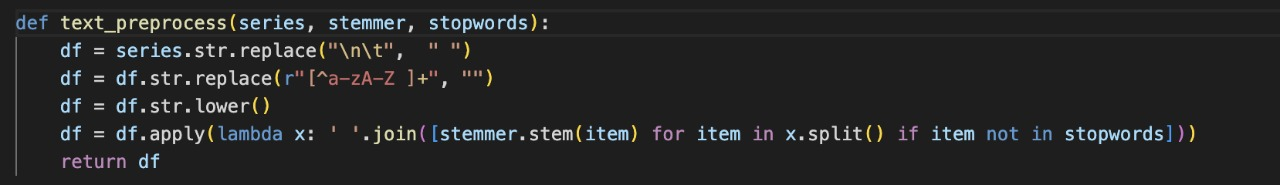
1. Tax : 2 number of rows
2. Health\_insurance : 5 number of rows
3. Public\_utilities : 4 number of rows
4. Credit\_card : 3 number of rows
5. Shopping : 112 number of rows
6. Comm\_internet : 26 number of rows
7. Loan : 37 number of rows

Please see below the dataset example from the extraction process in csv format.



* 1. Data Pre-process

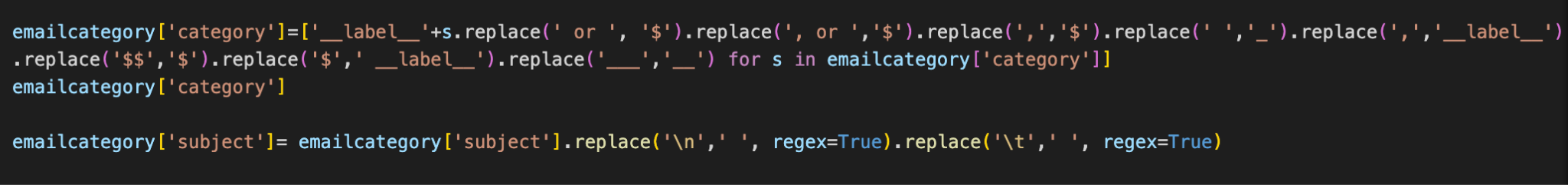
The subject data applied with a preprocessing step using Sastrawi library.



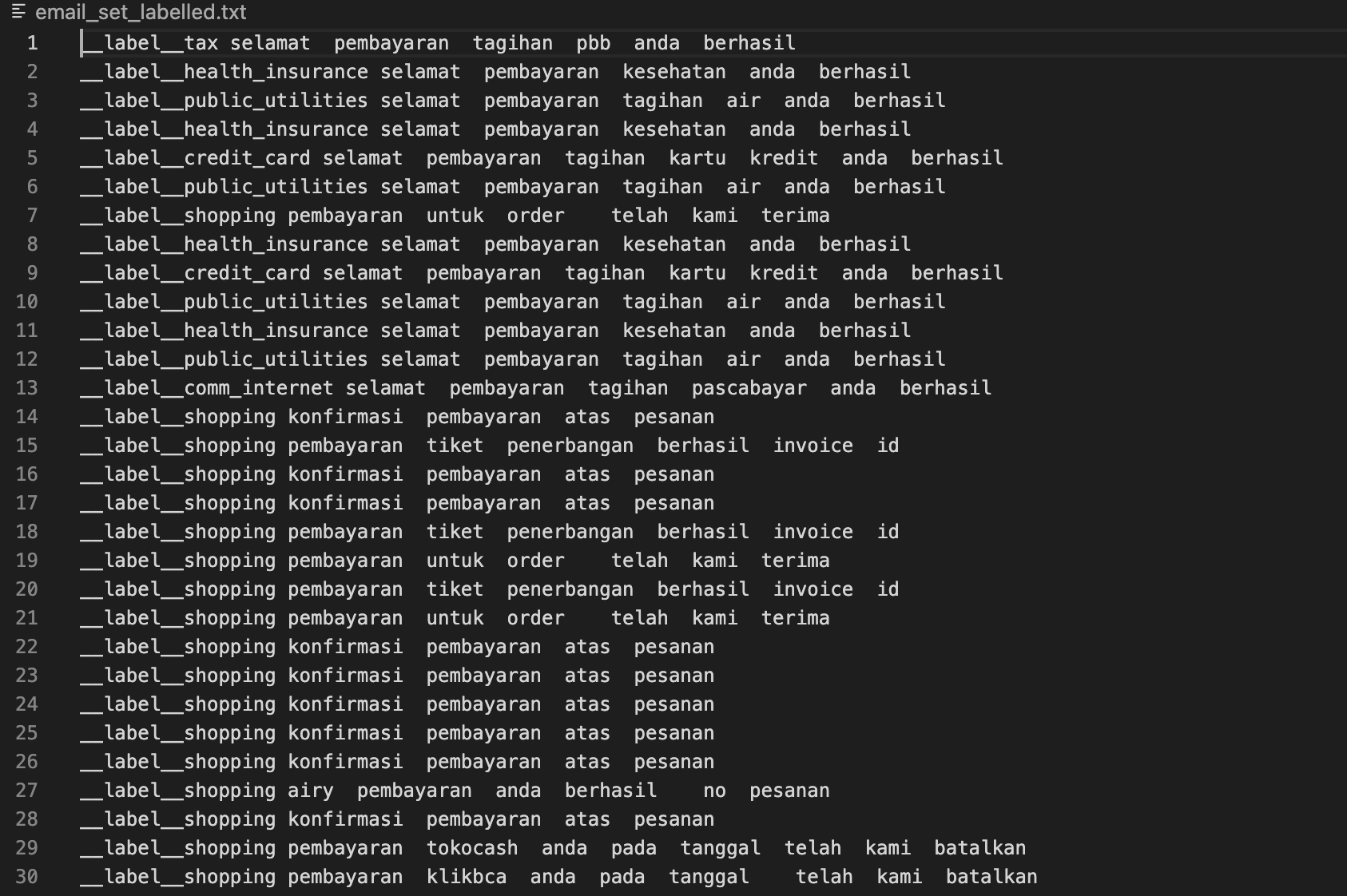
The pre-processing steps are:

* Replace the enter and tab with space
* Using the regular expression to match the uppercase and lowercase letter
* Lowering the data
* Split the sentence using ‘ ‘ into each word

The category inside the csv files were being labeled manually using below format:

****

The email set became the labeled email set as shown below:



* 1. Data Training and Set

During coding and implementation, it becomes a challenge when training a data set using a specific word. This research applies the manual labeling into the data set and then being trained by supervised learning using the fastText library. Using epoch =25, only able to distinguish 2 classifications. Request more effort to prepare the dataset. Overall performance, label shopping precision is about 99% for words containing “pembayaran” meanwhile only 3.8% precision for label comm\_internet

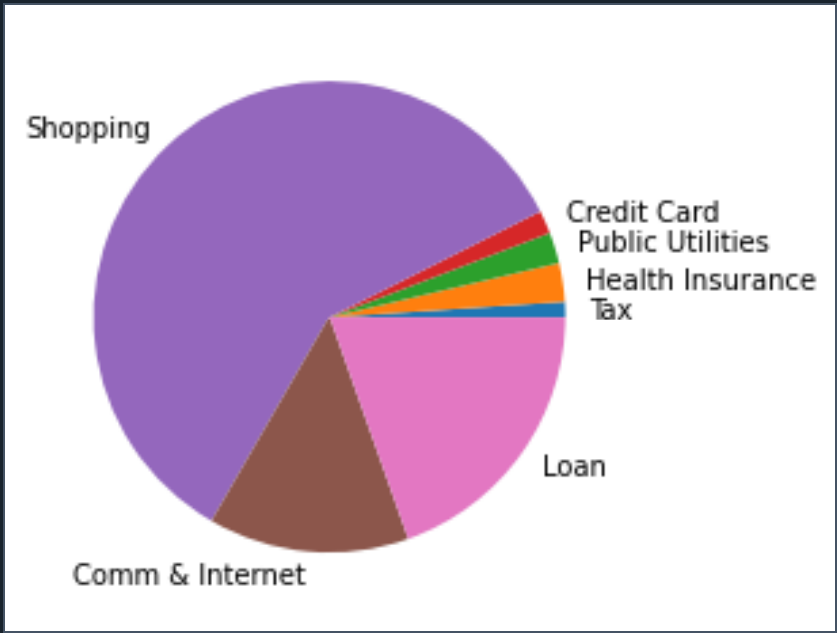
Label of Category = ['\_\_label\_\_comm\_internet', '\_\_label\_\_shopping']

Sum of Transaction = [Decimal('308437'), Decimal('26679862')]

Compare to manual labelling:

Label of Category = ['Clothing&Beauty', 'Tax', 'Public Utilities', 'Health insurance', 'Shopping', 'Communication & Internet', 'Loan']

Sum of Transaction = [Decimal('303760'), Decimal('431980'), Decimal('696000'), Decimal('762500'), Decimal('1089265'), Decimal('7382426'), Decimal('36263334')]



* 1. Data Storage

Using MySql connector the collection process will use 2 codes which are using manual labeling hardcoded and ML code using category train model. The data contains : from email, to email, subject, category and amount.

Amount is being extracted using a manual index one by one with a different format from each template email.

* 1. Data Transformation

From the raw data, the system creates a data set and saves it into a csv file. The data set contains the label category on each data. The next process will be visualizing the data into a bar chart using Matplotlib library. Please see below the sample of the report for the expenses in each category.

