CHAPTER 3
RESEARCH METHODOLOGY

This section describes the methodology that will be used to perform the exploration. The flowchart depicted in 3.1 shows the flow of the exploration that starts off with a literature review and will end in the reporting process of the results gathered.

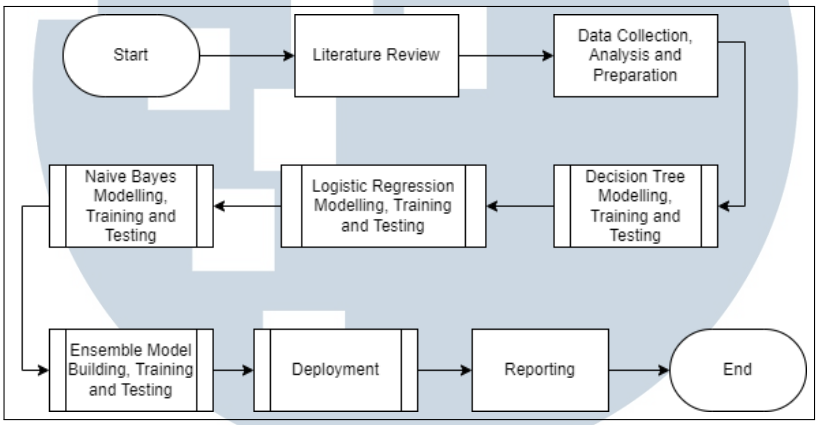

Figure 3.1. Flow of the research to be done

3.1 Literature Review

The first step into performing this study is the Literature Review. A comprehensive literature review will be done to fully understand the concepts behind implementing the proposed system. Understanding how an Ensemble Learning System works and how to implement Decision Trees, Naive Bayes, and Logistic Regression will be done. Additionally, reading into the role of permissions in an Android System will also be done. To perform this step, information will be gathered from various sources such as other previous researches and papers.

3.2 Data Collection, Analysis and Preparation

After a comprehensive review, the experimentation to obtain results for the study will be done. In order to do so, a dataset will be obtained. In this case, the data-set published by Arvind Mahindru titled *Android Permissions Dataset* will be collected [27]. This dataset holds data on 50,000 applications coming from the Android Play Store and also third parties. Each data entry holds information on

the application and the permissions required before installation, and also during run time. An example of the first few columns of the dataset is shown in Figure 3.2.

Package	Category	Total Perr	Default : Access DRM content. (S)	Default : Access Email provider data (S)	Default : Access all system downloads (S)
a.gosms.theme.sky	Personali	19	0	0	0
a1.golfshotfixex	Sports	5	0	0	0
a1.golfshotfixex	Sports	5	0	0	0
a2dp.Vol	Transport	12	0	0	0
a2z.ChirpE	Business	4	0	0	0
a3g.emyshoppinglist	Shopping	116	0	0	0
a8.kv.chilly	Cards & Ca	1	0	0	0
aa.AaCount	Sports	38	0	0	0
aa.catcheye	Business	3	0	0	0
aame.mobi.fingerfightlit	Racing	4	0	0	0
aame.mobi.helper	Tools	9	0	0	0
AB.AN	Business	0	0	0	0
ab.purple	Personali	0	0	0	0
abc.ssd.IpAddressChange	Tools	4	0	0	0

Figure 3.2. First few columns of the raw data that was gathered

Initially, the data downloaded is separated into 3 sheets, Google Play Applications, Malware Applications and Third Party Applications. These data were also not labelled. Therefore, the first step of preparing the data is to compare the Malware sheet to the Google Application and Third Party sheet. and make the labels in which is added as the last column on the dataset as seen in Figure 3.3. To label the data, if the application package is listed in the Malware sheet, it will be labelled with '1' on the TYPE column, this indicates that the application is a malware. On the other hand, if the package name is not available in the Malware sheet, it is labelled as '0' also on the TYPE column, or considered a harmless application.

BCE	BCF	BCG	BCH	BCI	BCJ
ACTION_	ACTION_	ACTION_	ACTION_	ACTION_	TYPE
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Figure 3.3. Last few columns of the labelled dataset

Once the data is all compiled into one excel sheet, it will then be loaded in to Google Colab in a form of a dataFrame. After the data is loaded into a dataFrame format, it is then split into the testing data and the training data. By using the Sklearn library, the ratio of data that will be split is 60:40, where 60% of the data will be allocated to the training dataset and the 40% will be used to test the data after the models are fitted. It can also be seen that the dataset is quite imbalanced with only 7% of the dataset being malware and the rest are benign, however no balancing processing will be done, as previous testing of balancing methods did not prove to be useful to the overall results. After this process is complete, the data is ready to be used in the model's modelling phases.

3.3 Individual Model Modelling, Training and Testing

Due to the nature of the ensemble learning system needing base learners, this is where the base learners are modelled and tuned. Finding the correct parameters and implementing the correct libraries will be conducted in this stage of the study. The main models will consist of Decision Trees, Logistic Regression Classifiers and Naive Bayes Classifiers. In this stage, the individual models will be trained using samples from the dataset and tested to ensure the best value in terms of the evaluation metrics are obtained.

A decision tree model will be modelled, trained and tested first. The model will use the SciKit library with tuned parameters. The main parameters that will be tuned to obtain the highest accuracy consists of the Random State, Max Depth, and the Criterion. The procedure in which is used to obtain these values is depicted in the flowchart in Figure 3.4. As seen below, at the end of the process, the best values for the Random State, Max Depth and Criterion will be set and ready for the aggregating process.

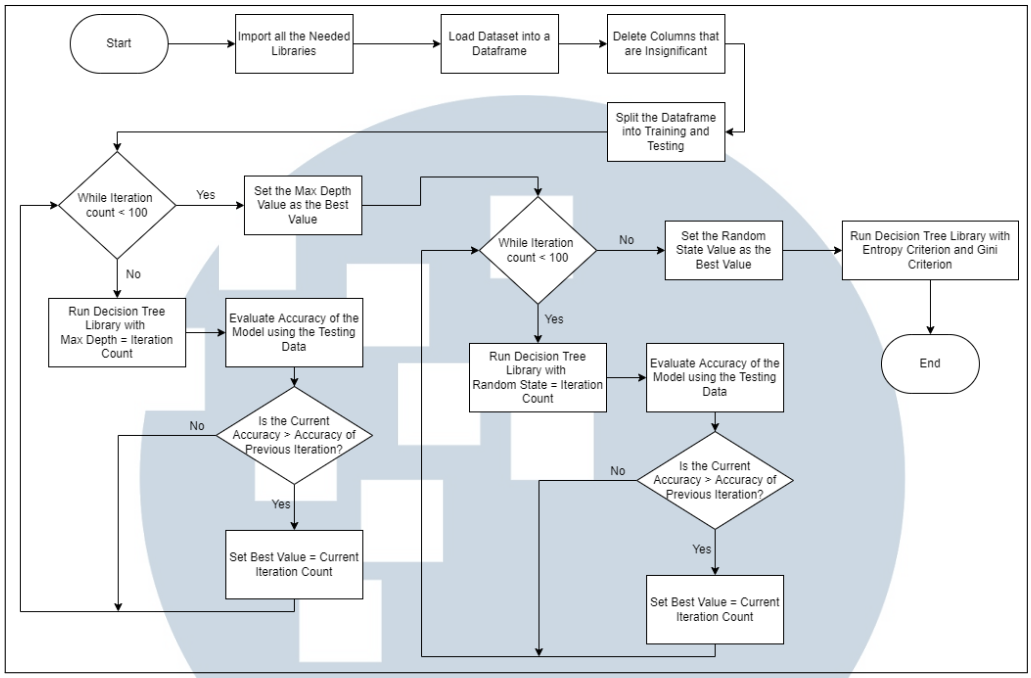

Figure 3.4. Decision Tree modelling, training and testing flowchart

The next model to be modelled, trained and tested is a Logistic Regression Classifier model, this model will be developed similarly to the decision tree model in which the parameters will be looped over and the value with the highest accuracy is the value to be used in the upcoming process. The parameters here include the solver type and the random state, this process is shown in the flowchart in Figure 3.5.

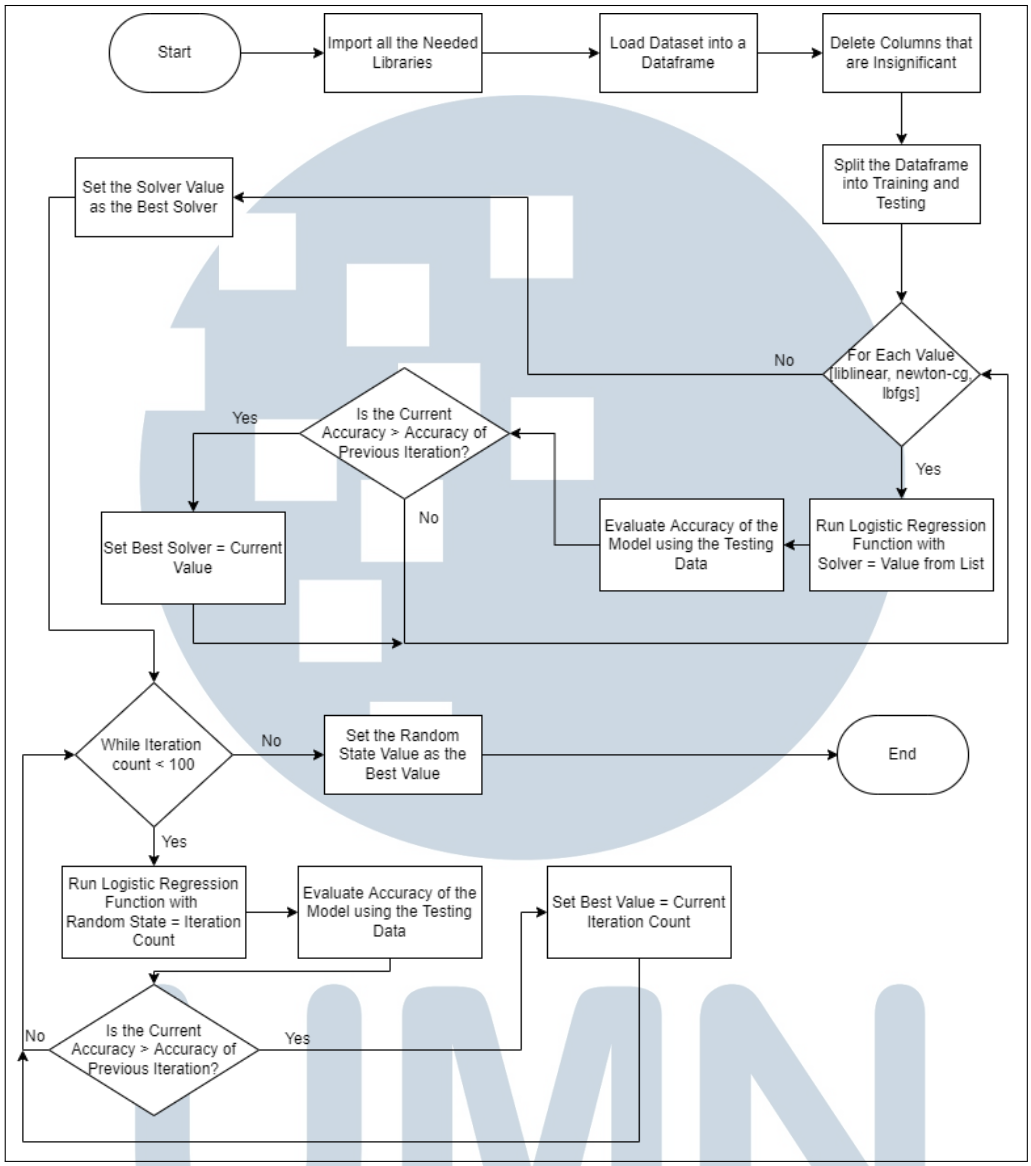

Figure 3.5. Logistic Regression modelling, training and testing flowchart

The last model is the Naive Bayes Classifier model, similarly to the previous two models, this model will go through a similar process in which the parameters are tested with different values, however, the value to be tested in the Naive Bayes Classifier includes the classifier function and the alpha value. Shown in Figure 3.6 is the flow to obtain the best parameters for the Naive Bayes model.

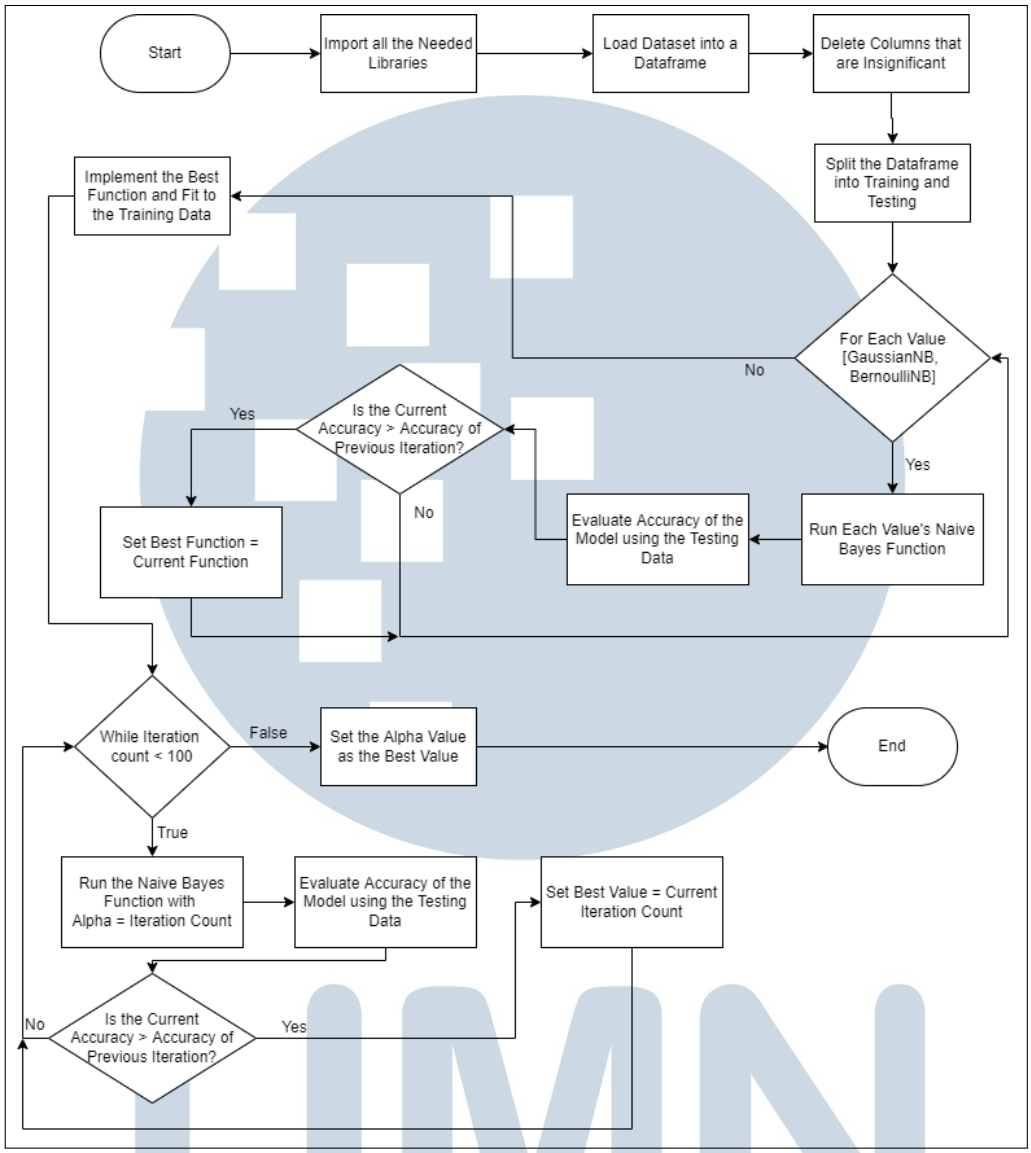

Figure 3.6. Naive Bayes modelling, training and testing flowchart

3.4 Ensemble System Building, Training and Testing

After each of the base learners are modelled, an ensemble learning system will be devised. Since the bagging, stacking ensemble technique will be used, in this stage the base learners will be arranged in parallel to produce a single output. Different combinations of the models will be put together, this combination will include the different amount of learners, and the combination of different types of learners. These combinations will be trained with the same dataset and tested to obtain the highest evaluation metric values but the final model that will be deployed

will be chosen based on the highest accuracy value.

The bagging technique will first be tested where the previously tuned models will be duplicated and the amount with the best accuracy will be noted down. The process in which this will be done is depicted in the flowchart in Figure 3.7.

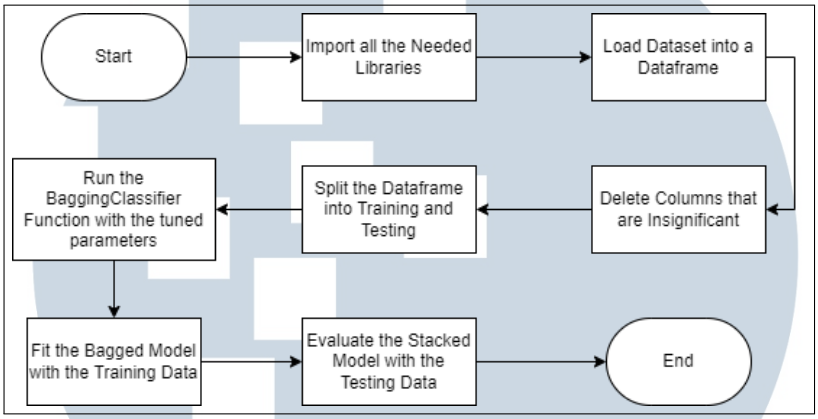

Figure 3.7. Flowchart depicting the implementation of the Bagging Technique

The other technique is the stacking method in which a heterogeneous ensemble learning model is formed. Here, the different models that were tuned in the previous stage will be combined together and then evaluated. The course that is taken to perform this is shown in the flowchart in Figure 3.8.

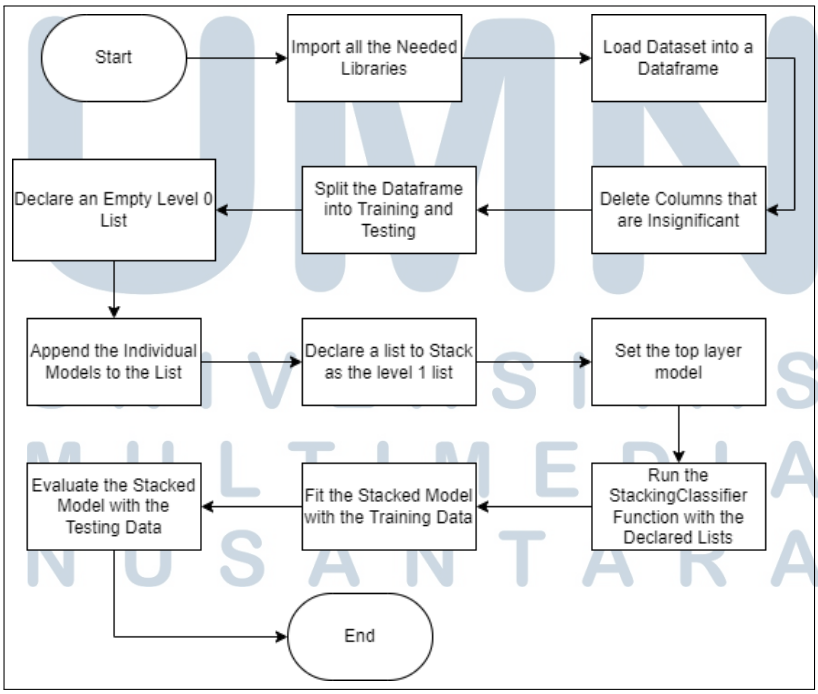

Figure 3.8. Flowchart depicting the implementation of the Stacking Technique

3.5 Deployment

Once the model is exported, a back-end system will be devised using Flask. This will be used as a means for the model to be accessed from the mobile application. In this stage too a mobile application will be developed. This application will be used by the user to input the permissions they are required to approve before a download, from there the data will be sent to the deployed Flask app in the server to be processed with the ensemble learning model that previously had the best accuracy. To further describe this process the flowchart shown in Figure 3.9 is shown.

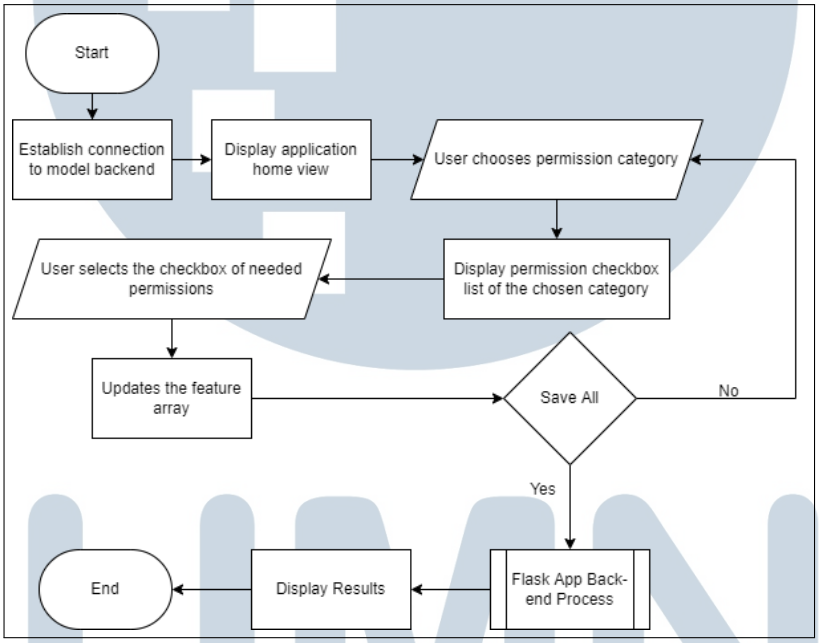

Figure 3.9. Main flow of the mobile application

The simple app will start of in a home page that shows the 13 possible categories of permissions. The interface for this will be a scroll-able layout to make sure that the buttons are visible and usable. This rough design is seen in Figure 3.10.

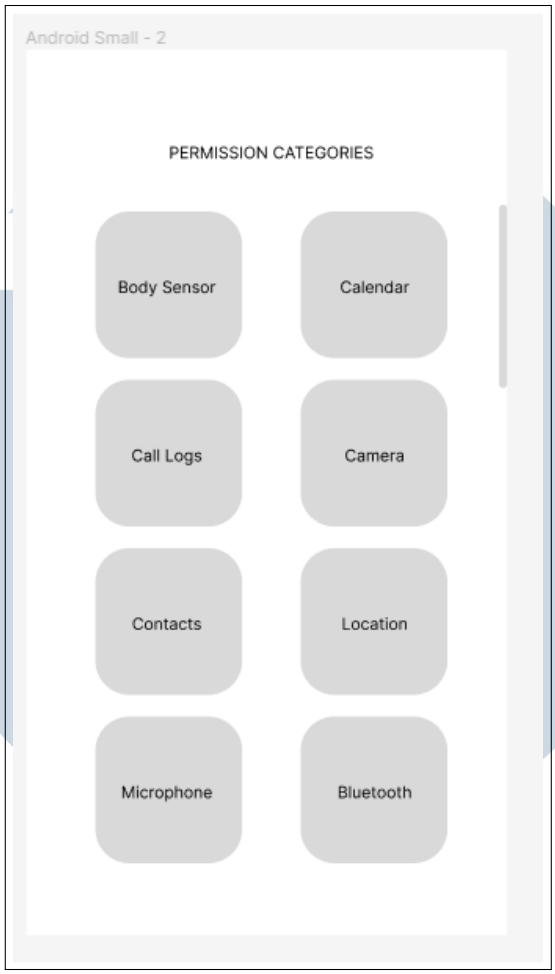

Figure 3.10. Simple home view of the mobile application

Once the user chooses a category, they will be directed onto a page with check-boxes that will list down the possible permissions that an Android device might ask when installing or running an application, as seen in Figure 3.11. This interface will be similar for all the categories, the only differentiating feature would be the permissions displayed.

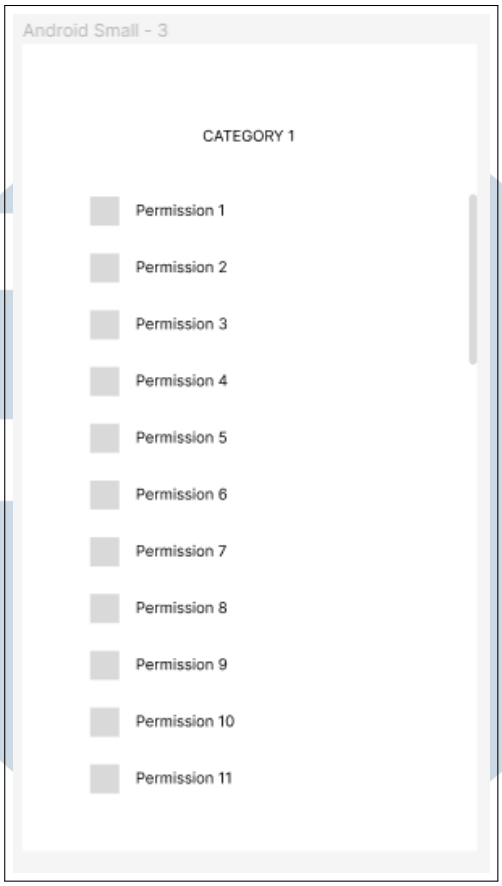

Figure 3.11. Design of per category view

Once a user uses the check-boxes and returns to the home page, the selection will be saved and added onto an array. This array will then be passed into a developed back end flask application that will run the machine learning model and produce the prediction. The flowchart to this process is depicted in Figure 3.12.

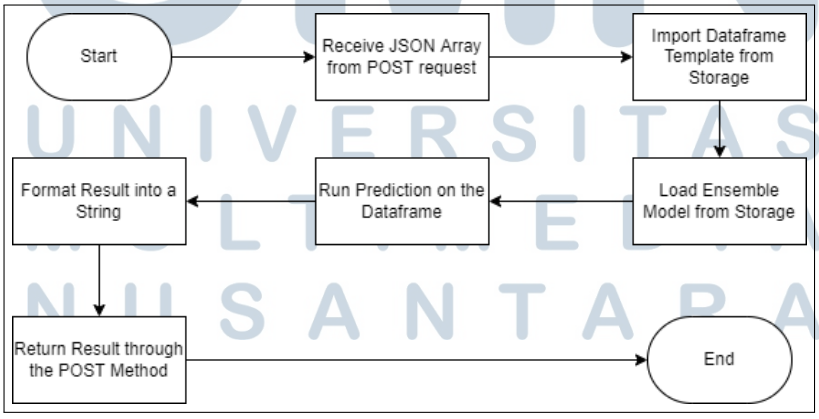

Figure 3.12. Flowchart of the Flask back-end application

Once the results are sent back to the mobile application, it will be displayed to the user mentioning if the application they wanted to check is considered 'Malware' or simply a 'Benign' application.

To fully perform the deployment stage, the system specification requirements are as follows:

3.5.1 Hardware

- RAM: 8GB
- Access to Internet
- Processor: Core i5 or equivalent

3.5.2 Software

- Google Chrome
- Google Colab
- Android Studio
- Visual Studio Code

3.5.3 Backend Requirements

- Flask v.2.1.2
- Gunicorn v.20.1.0
- Numpy v.1.22.4
- Pandas v.1.4.2
- Sckit-learn v.1.0.2

3.6 Reporting

The final stage of the study is the reporting phase, here all the activities that were previously done will be written in a report. All the observations and results will be noted down and compiled into an organized final paper. At the end of the reporting stage, suggestions will also be given for future works.