CHAPTER 3
RESEARCH METHODOLOGY

3.1. Literature Study

This is the first step in working on this project. There are at least 10 journals that have information about Support Vector Machine as a classification algorithm and canny edge detection as an image segmentation method. All journals are collected to be used as references so that researchers can conclude the topics and problems they want to study. Journals can be taken from anywhere, for this case I took journals from Google Scholar and the Unika repository.

3.2. Collecting Dataset Image

This research uses a dataset of leaf images from different plants. The dataset images can be downloaded on the Kaggle website. From the downloaded dataset, there are about 30 different plant species and each species has 34-122 images. The total image dataset is around 1835 images.

(link: https://www.kaggle.com/datasets/aminizahra/leaf-edge).

Table 3. 1 Leaf Image Dataset

No	Nama Tumbuhan	Jumlah data	No	Nama Tumbuhan	Jumlah data
1	Daun Ara Suci	63 items	16	Daun Kersen	56 items
2	Daun Bayam Hijau	122 items	17	Daun Lengkuas	50 items
3	Daun Bayam Malabar	103 items	18	Daun Malapari	61 items
4	Daun Buah Samarinda	74 items	19	Daun Mangga	62 items
5	Daun Cendana	58 items	20	Daun Melati	71 items
6	Daun Delima	79 items	21	Daun Mimba	60 items
7	Daun Ficus Auriculata	50 items	22	Daun Mint	97 items
8	Daun Jamblang	39 items	23	Daun Mondokaki	56 items
9	Daun Jambu Biji	65 items	24	Daun Nangka	56 items
10	Daun Jambu Mawar	56 items	25	Daun Oleander	62 items
11	Daun Jeruk Sitrun	57 items	26	Daun Ruku-Ruku	52 items
12	Daun Jintan	48 items	27	Daun Salam Koja	60 items
13	Daun Kelabat	36 items	28	Daun Sesawi India	34 items
14	Daun Kelor	77 items	29	Daun Sirih	48 items
15	Daun Kembang Sepatu	43 items	30	Daun Srigading	40 items
Total					**1835 Items**

3.3. Taking Input

After the dataset is obtained, then all the images are read using a method called *cv2.imshow()* from OpenCV. Since SVM only accepts inputs with the same size, then all images have to be resized to 200 * 200 pixels. The method used to resize is *cv2.resize()*. The size of 200 pixels is chosen because with 200 pixels we can still get a clear image. After the image gets the same size, the image will go through the edge detection segmentation process using the canny method. the canny method was chosen because canny is the best edge detection among other methods. After all, the canny method is a multi-stage algorithm that makes it the most precise edge detection. The method used for the segmentation process is *cv2.canny()* with a minimum and maximum threshold value of 400 to get the edge pattern of the leaf image. Then the binary results of the segmentation are labeled one by one and stored in an array. After that, the array will be saved into a ***pickle*** file with the purpose that when the program is run, there is no need to read and process the image again.

Table 3. 2 Converting name to label

Nama Tumbuhan	Labels
Daun Ara Suci	0
Daun Bayam Hijau	1
Daun Bayam Malabar	2
Daun Buah Samarinda	3
Daun Cendana	4
Daun Delima	5
Daun Ficus Auriculata	6
Daun Jamblang	7
Daun Jambu Biji	8
Daun Jambu Mawar	9
Daun Jeruk Sitrun	10
Daun Jintan	11
Daun Kelabat	12
Daun Kelor	13
Daun Kembang Sepatu	14
Daun Kersen	15
Daun Lengkuas	16
Daun Malapari	17
Daun Mangga	18
Daun Melati	19
Daun Mimba	20
Daun Mint	21
Daun Mondokaki	22
Daun Nangka	23
Daun Oleander	24
Daun Ruku-Ruku	25
Daun Salam Koja	26
Daun Sesawi India	27
Daun Sirih	28
Daun Srigading	29

3.4. Splitting Dataset

Train test split is a method that is used to separate the dataset into two parts, the part used for training data and the part for testing data with a certain ratio before it is finally applied in the modeling of algorithms. Before implementing the train test split, the previously saved data can be loaded and defined in the Feature (binary) and Label (class) parts. After the definition, implement the Feature and Label into the train/test split. For train/test split, you can use the ***scikt-learn*** library by importing the method called *train_test_split()*. For the distribution ratio of this split test train is as below.

Table 3. 3 The ratio of training and testing data

Training Data	**Testing data**
70%	30%
75%	25%
80%	20%

3.5. Model Construction, Training, and Testing

After splitting the train and test data, the data can be implemented into the algorithm model. The algorithm model in this project is Support Vector Machine. SVM is a supervised learning method that is usually used in classification. SVM also has a kernel trick to help the non-linear classification process. This trick kernel also comes with several parameters that have a great effect, which are Cost (C) and Gamma. The Cost parameter which can be called C is a parameter that works as an SVM optimization to avoid misclassification in training. In this research, we use the Radial Basis Function (RBF) kernel and the Cost parameter is set with different values, that is, C = 1, C = 10, C = 30, C = 50, and C = 100. After setting the model parameters, we must fit the model with the purpose to train the model that has been made. After the model has been trained, the model must be tested to identify and extract the accuracy of the model.

3.6. Model Evaluation

Model evaluation is a stage to measure the performance of the model created. The results of the measurements taken can be a consideration in choosing the best model. At this stage, the model will be used as a reference for testing with different images.