

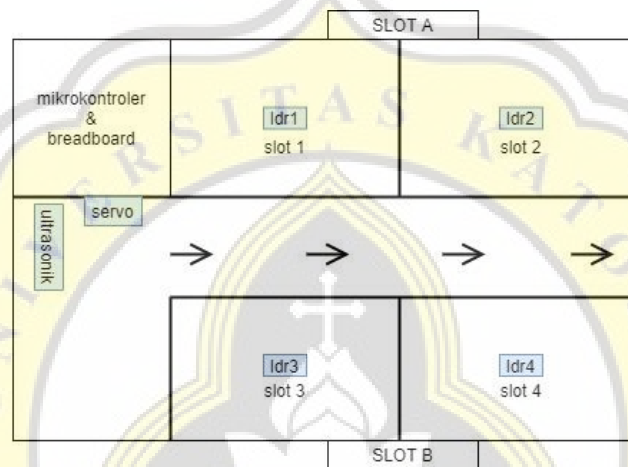
## CHAPTER 4

### ANALYSIS AND DESIGN

#### 4.1. Analysis and Design

There are several test scenarios to test this car monitoring system with several tests, such as functional and non-functional testing, such as several test plans that will be carried out to test whether the system is in accordance with the concept that has been designed or not.

##### 4.1.1. Scenario of prototype and sistem



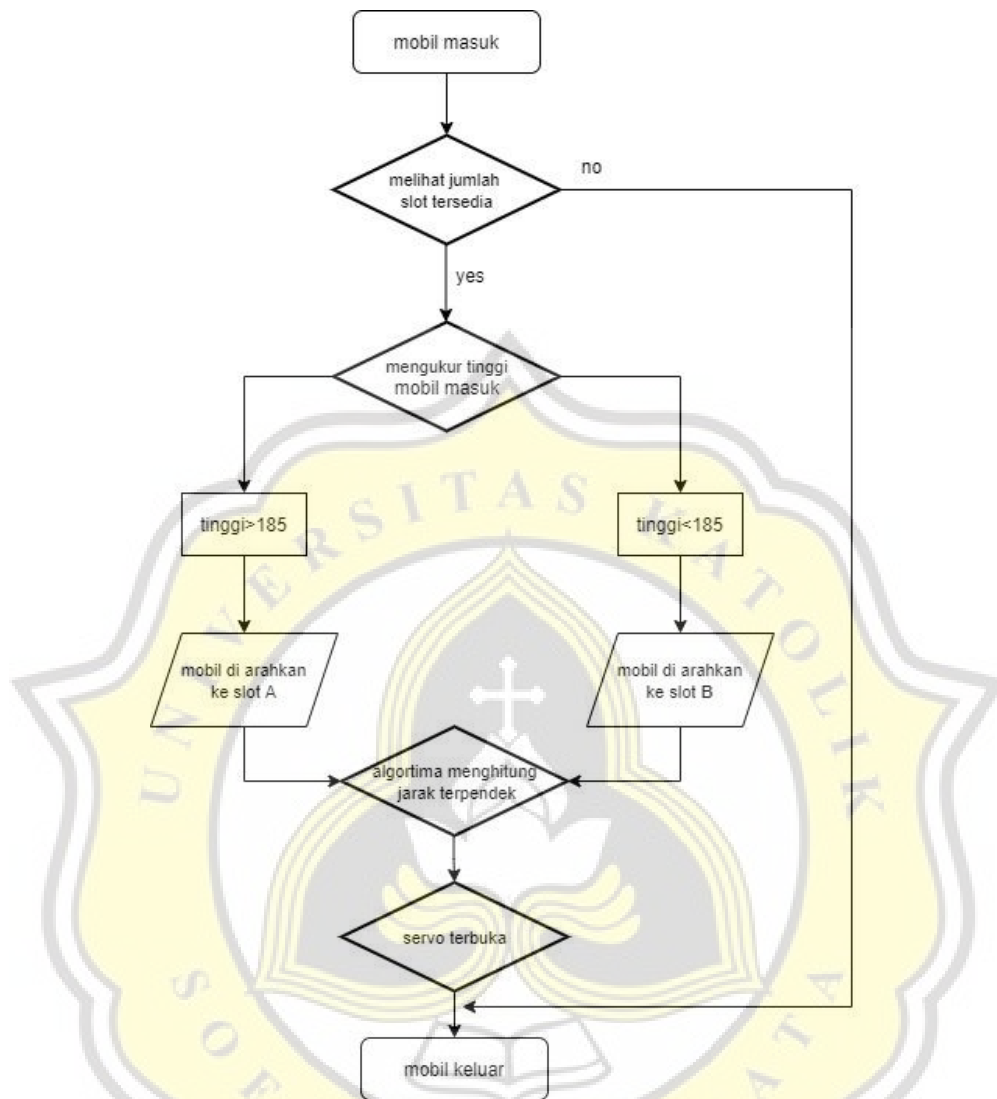
**Figure 4.1** Prototype floor plan

The picture above is a tool plan or prototype of a car parking monitoring system that has been made, it can be seen from the existing floor plan in this system using a MEGA 25660 microcontroller, using an ultrasonic sensor to measure the height of the car that will enter, using the LDR module in each existing slot. , and the use of servo motors to drive the bars. The test scenarios that will be carried out are listed in the table below:

**Table 4.1.** Scenario of testing to be carried out

No	Tests to be carried out	Result
1.	Testing of ultrasonic sensor function	Appropriate or not
2.	Servo function test	Appropriate or not
3.	Testing the function of the LDR sensor module	Appropriate or not
4.	Testing the dijkstra algorithm	Appropriate or not

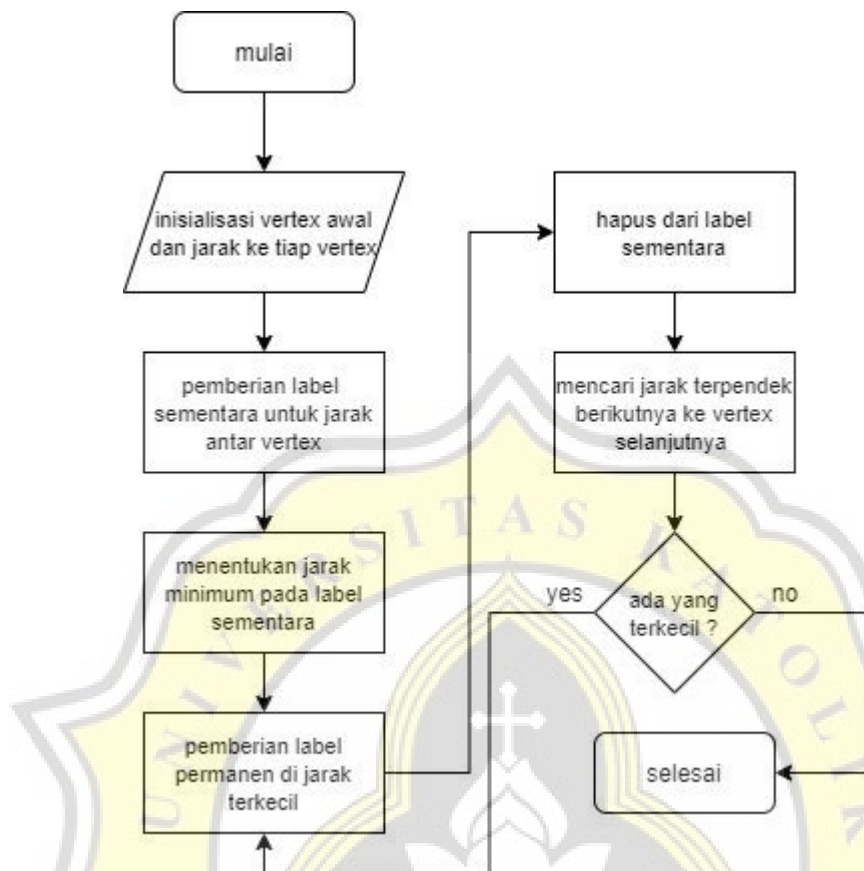
#### 4.1.2. *Scenario sistem*



**Figure 4.1** flowchart of work system logic tool(prototype)

The flowchart above is a logical description of how the system runs, starting from the car and the system checking whether the slot is still available or not if it is available then the system will measure the height of the car, if the section on calculating the height of the car is met, the car will be directed to the slot that matches the height of the car. After that entry, then the algorithm system will calculate the distance of the shortest available slot according to the intended slot, after that the servo will open and the car can enter the slot that is still available.

#### 4.1.3. Scenario algorithm



**Figure 4.1** Algorithm logic flowchart

In the logic flowchart section, the algorithm explains how the algorithm works starting from vertex initialization and initialization of weights from the starting point to each vertex, after initialization, temporary labeling is carried out for the distance between vertices, and continued by calculating or determining the minimum distance on the temporary label after being determined, labeling is carried out. permanent label at the smallest distance and removed or removed from the temporary label, then look for the next shortest distance to the next vertex of the permanent label if there is the smallest then the distance is included in the next permanent label.