

CHAPTER V

IMPLEMENTATION AND TESTING

5.1. Implementations

The main problems in this program is how to pass the data from the client to another client through the server. The server as the center of the data exchange is designed with Socket as the communication media and Thread as the multitasking media which is needed to handle a lot of tasks simultaneously. While server runs, it opens a specific port number in order to wait and accept the requested connection from the client using the Server Socket and threads.

To start the server, specify the port number that will be used by the server socket in order to accept the client. The input format will be 'java Server portnumber'. For the example server run on the port 1234 : java Server 1234.

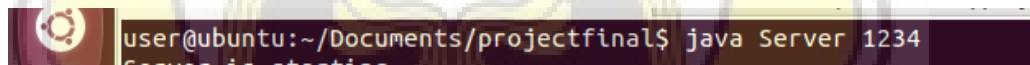
A terminal window screenshot showing a command prompt. The prompt is 'user@ubuntu:~/Documents/projectfinal\$'. The command entered is 'java Server 1234'. Below the command, the output 'Server is starting' is visible. The terminal background is dark, and the text is light-colored.

Figure 4 : Run Server with specific port number.

Client needs server IP and port number to connect to the server. The command format is 'java GUIClient serverIP portnumber' for the customer and 'java GUIcs serverIP portnumber' for the customer service. For the example, because of the client and the server on the same computer, clients use localhost as the ip number : java GUIClient localhost 1234.

```
user@ubuntu:~/Documents/projectfinal$ java GUIClient localhost 1234
```

Figure 5 : Clients connect to server

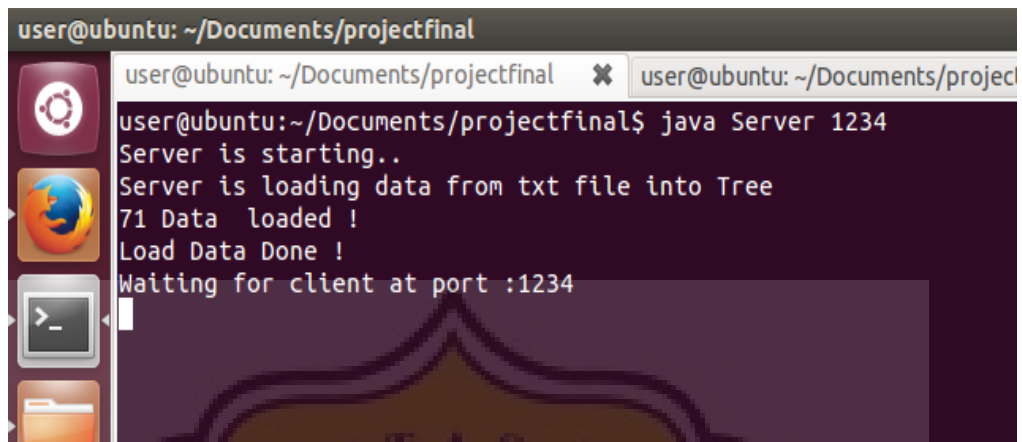
Server and Clients are using input data stream to receive data and output data stream to send data. The data that is received and sent converted to String to able to be displayed on the GUI Panel.

The Server temporarily stored all of the connected clients into the arraylist. It manage the clients based on the status that is Customer and Customer Service. Server connects the customer and the customer service and will pass the their data to the its recipient.

After the client with 'customer' status connected, Server will check the product ID from the Binary Tree. If the data with the same product ID exist, Server will send the previous data to the Customer Service.

When the session between Customer and Customer Service finished, the customer service status will become available again in order to accept another customers.

5.2. Testing

A terminal window on an Ubuntu system. The title bar shows the current directory as ~/Documents/projectfinal. The terminal output shows the execution of a Java program named 'Server' with the argument '1234'. The output consists of several lines: 'Server is starting..', 'Server is loading data from txt file into Tree', '71 Data loaded !', 'Load Data Done !', and 'Waiting for client at port :1234'. The terminal prompt is user@ubuntu:~/Documents/projectfinal\$.

```
user@ubuntu: ~/Documents/projectfinal
user@ubuntu: ~/Documents/projectfinal$ java Server 1234
Server is starting..
Server is loading data from txt file into Tree
71 Data loaded !
Load Data Done !
Waiting for client at port :1234
```

Figure 6 : Server Starts

Server starts and create the server socket using port 1234 and waiting for clients to connect. Server reads the client data from txt file and load it into the Binary Tree. After the client connected into the server, their IP and port will be displayed on the console. Every client connected to the server, Server will create a special thread to store client data such as name, status, destination, and customer data. This thread will be stored into the arraylist for the data exchange purpose.

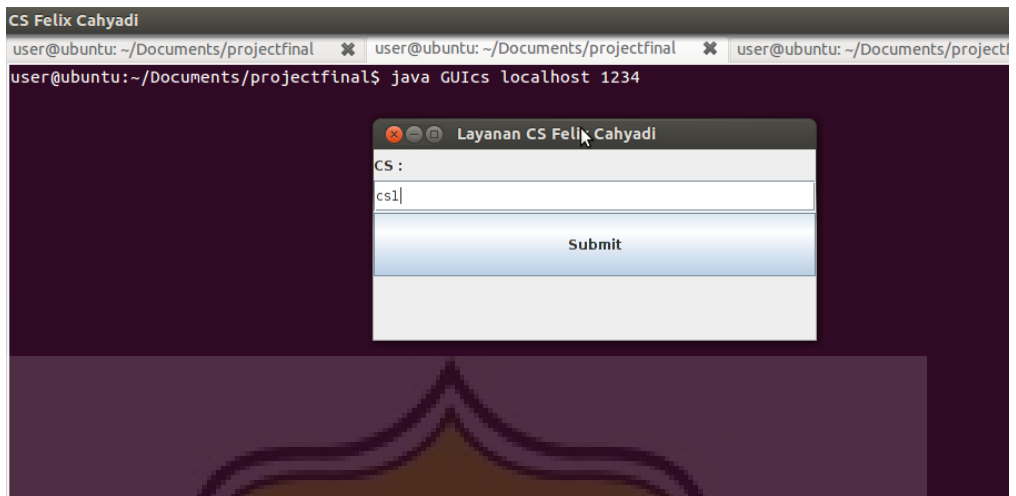


Figure 7 : GUIcs Filling the name field

When the user starts the GUIcs, they have to specify the server IP address and port number. After that, customer service should enter their desired name and click submit.

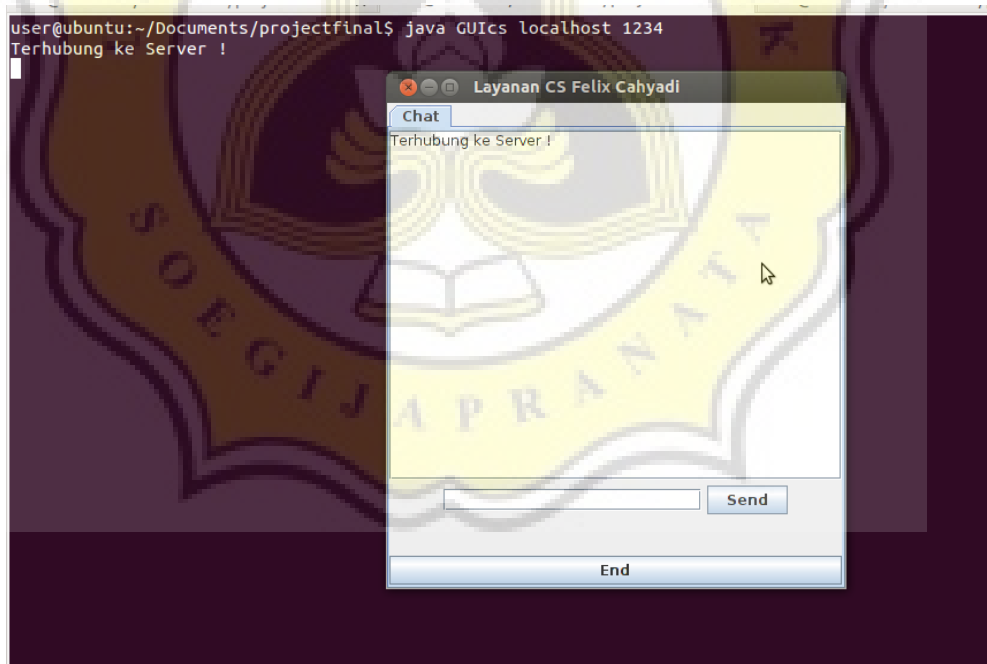


Figure 8 : Customer Service is connected to the Server

The Customer Service is connected to the Server and will received a message '*Terhubung ke Server*'.

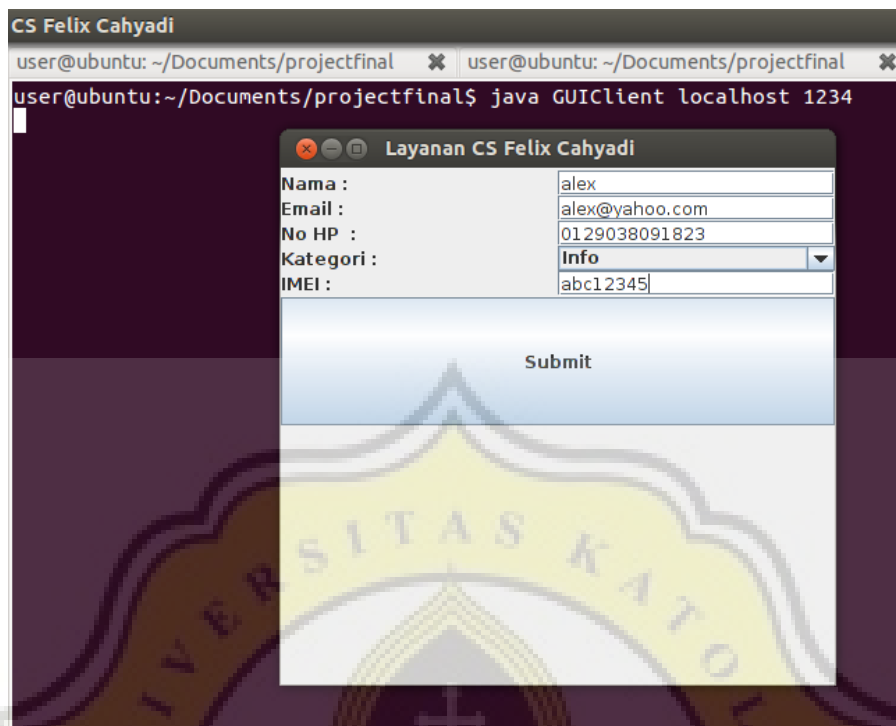


Figure 9 : GUIClient fill customer details

Same with the GUIcs, after specify the server IP Address and port number GUIClient able to connect to the Server. After that, customer should fill the customer details and click submit.

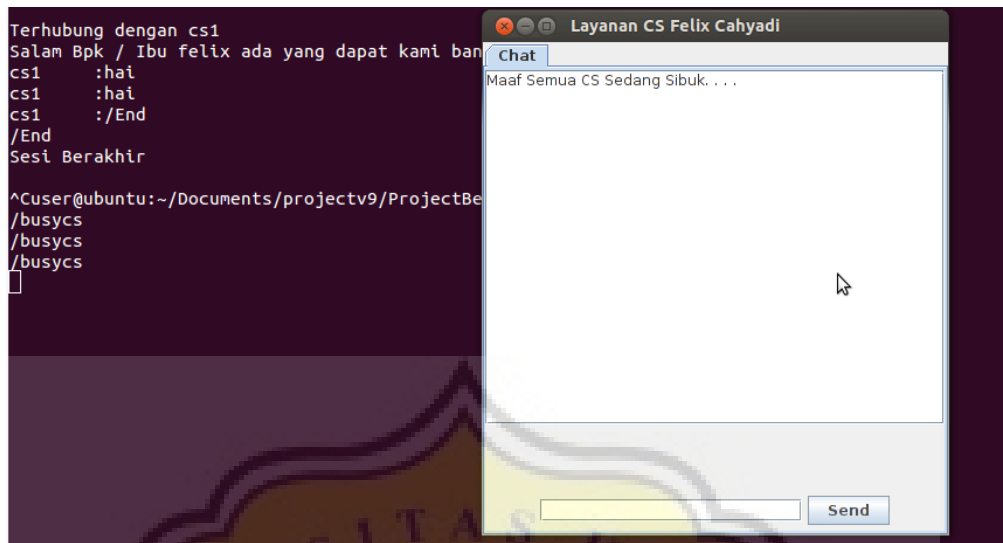


Figure 10 : Server Trying to Search Available Customer Service

When the customer is connected with the server, Server will search the available customer service from the arraylist. If available customer service is exists, server will connect them. If there is no any available customer service, server will send a special code “/busy” to the customer and it will displayed a loading screen.

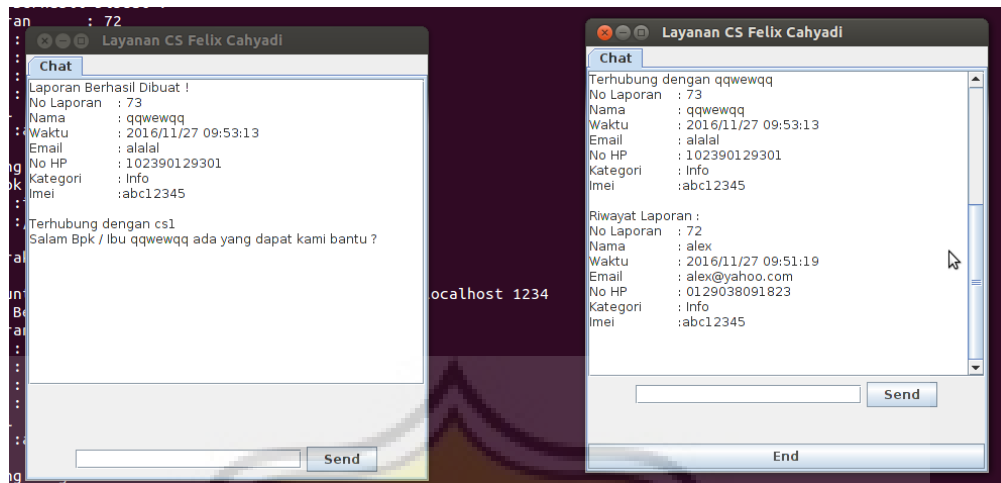


Figure 11 : Customer and Customer Service Chat Session

After the customer and customer service are connected, server will check the customer product id if it already exist from the binary tree. If it exists, server will send the previous report to the customer service. Customer and customer service can chat to each other.

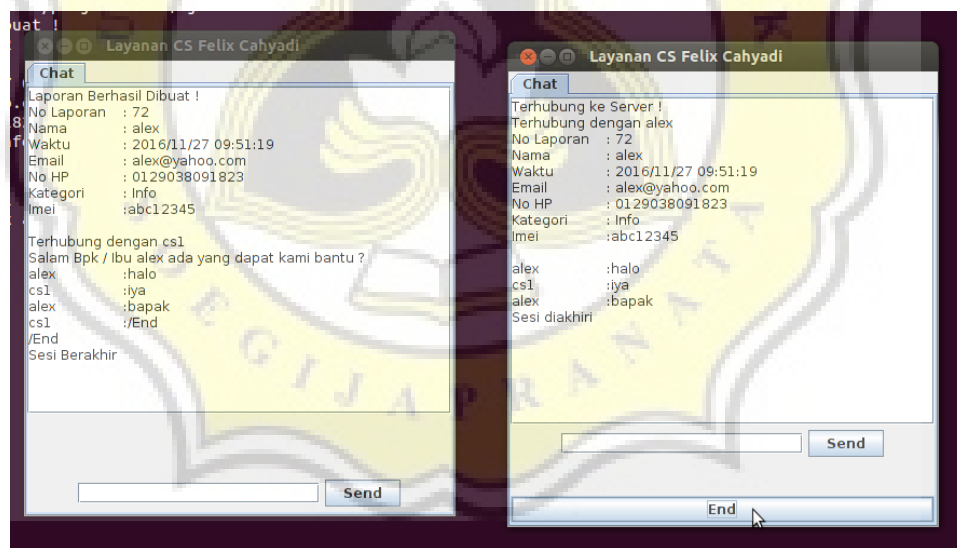


Figure 12 : Customer Service Ends The Session

After chatting session between Customer and Customer Service ends, customer service can click "End Button" in order to end the session and the Customer Service status will be revert back to available to accept the other Customers.