

CHAPTER 4

ANALYSIS AND DESIGN

Analysis

Server

The server has a thread to handle clients that will be connected to the server, the goal for the application to run more responsive, and clients who have unstable / interrupted connections do not affect other clients because each client has an important element managed by the server. In this server will have a data containing answers from the pictures that will be guessed by the clients / players. The data is stored using arrays. In 1 game room there are only 3 clients that can play for it the server is designed to limit the clients connected on the server, if a room already has 3 clients connected then the new server allows the game to run in this case clicking the snippet of the pictures which has been provided. If there is a new client connected to the server, the server will create a new room for the client and wait until there are 3 clients connected to the server and the game will run. The server also has a time-out system which will kick clients who do not interact to any server within a certain period of time.

Client

Once connected to the server, the client is required to fill his identity by the server, the identity will be sent to the server and managed by the server, this element is very important because the client has multichat feature which can interact with other clients so that the client can know who wrote the message. Data pieces of the image will be stored in the client, so that the interaction in opening the image faster and prevent the occurrence of traffic / packet loss in a game that will lead to instability of the connection on the client.

Protocol

This game has a variety of rules that serve to manage the game run on the server deal with the client.

In-Game Protocol

This protocol governs the players before they can access the game. Before accessing to the game the player will be asked to enter a name, the name is the identity of the player in the game. Once the name input is successful then the player will be allowed by the server to enter into the game.

Pre-Game Protocol

This protocol is a waiting area before playing If there are no 3 players entering the game then the player can not open the box but the player can chat to another player who has entered the game.

Box Protocol

This protocol sets the box selected by the player and raises it to other players simultaneously, prevents cheating if not yet turn then the player can not select the provided box and pops up a warning, and also prevents and gives warning to the player choosing the selected box by another player. If the player is allowed by the server to open the selected box, the server will give the box a mark that already belongs to the player choosing it. This prevents the selected box from being selected so the next player has to choose another box. If the player is allowed by the server to open the selection box then automatically turn the player will change to next player's turn.

Timer Protocol

This protocol uses a timer on the server for the game to run even if there is one client whose network is disconnected to the server or one of the client does

not do movement. Apbila timer has reached the number 30 then automatically turn will be thrown to other players.

Answer Protocol

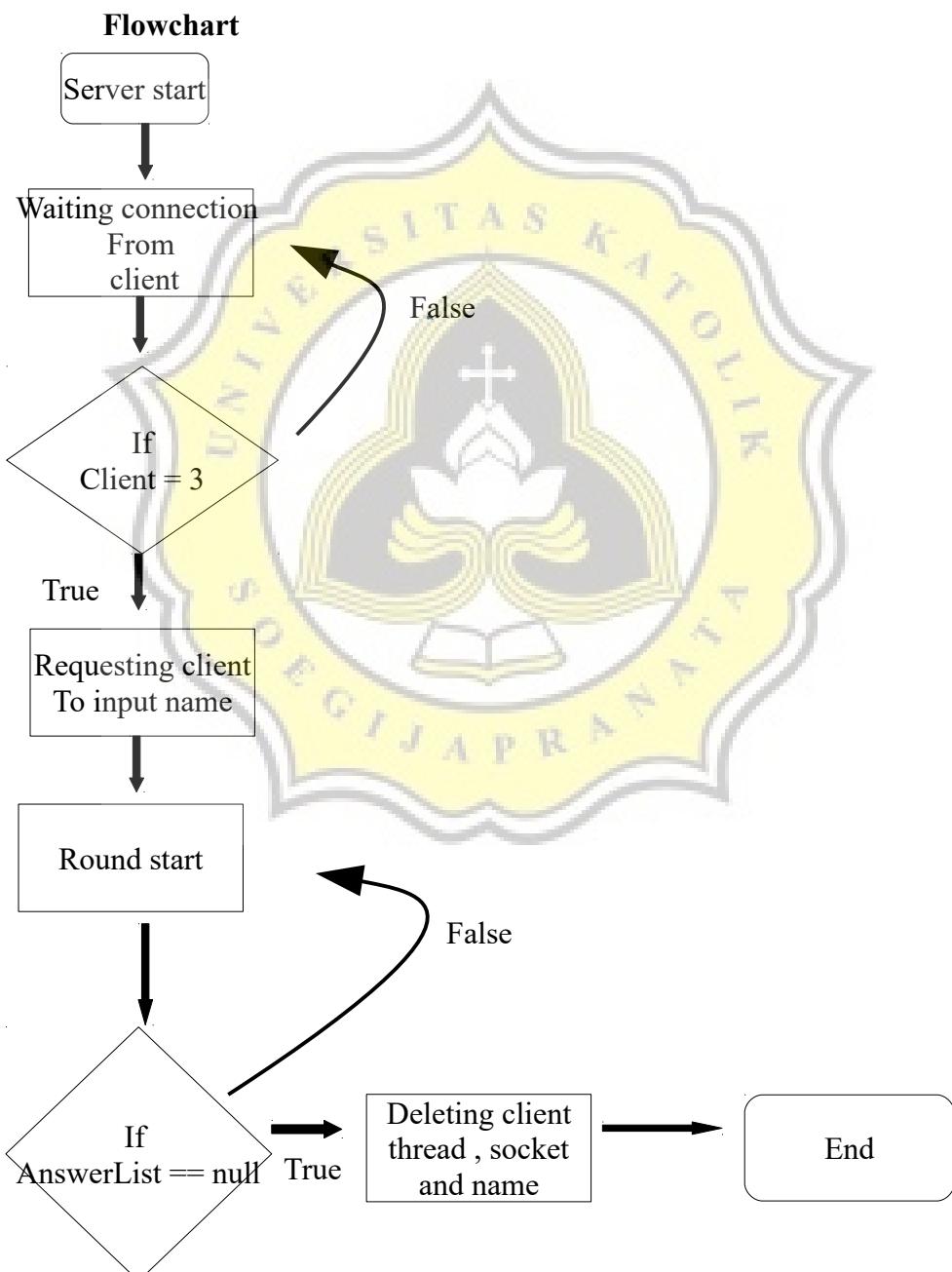
This protocol sets the answer from the players, if it is true then the server will give the correct score to the correct player in reply, the server will replace the image that will be released to the other image, and the server will return the signs of the player marked on the previous box to the original position where there is no sign on the box for the new image.

Chat Protocol

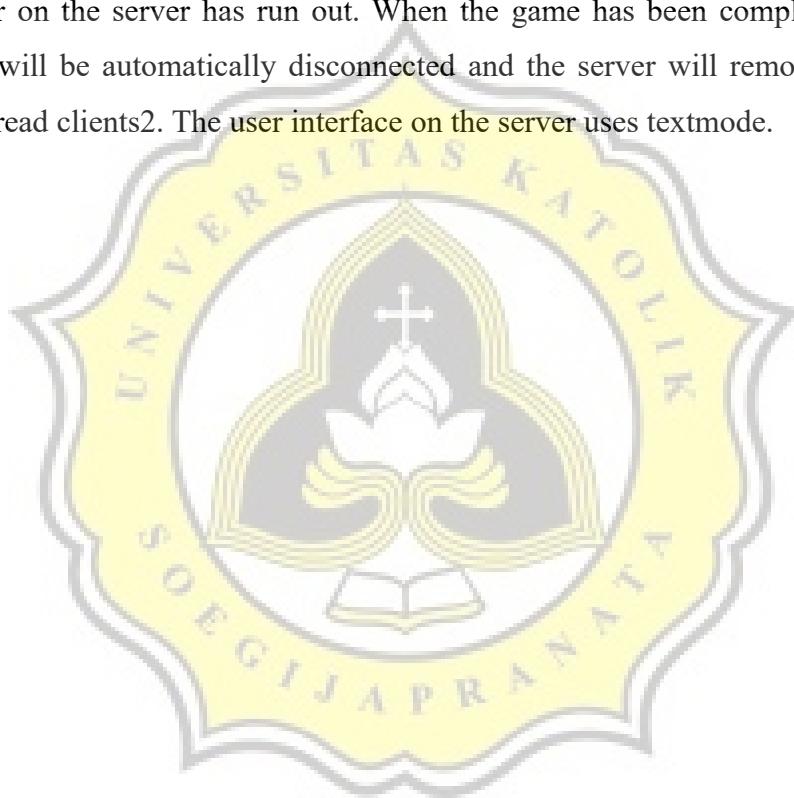
This protocol is useful for exchanging messages to other players in the game. The client will write a message and send it to the server, the server will return the message to another client.

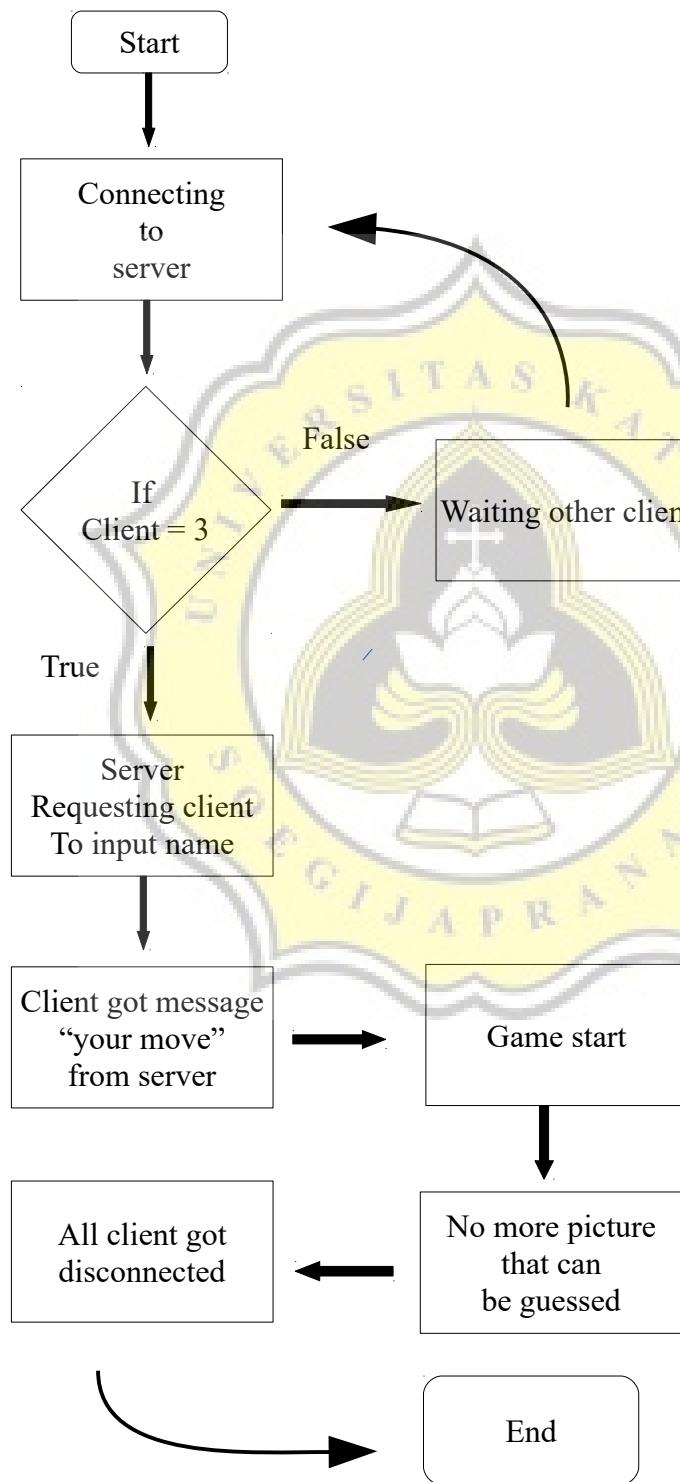
Desain

Server

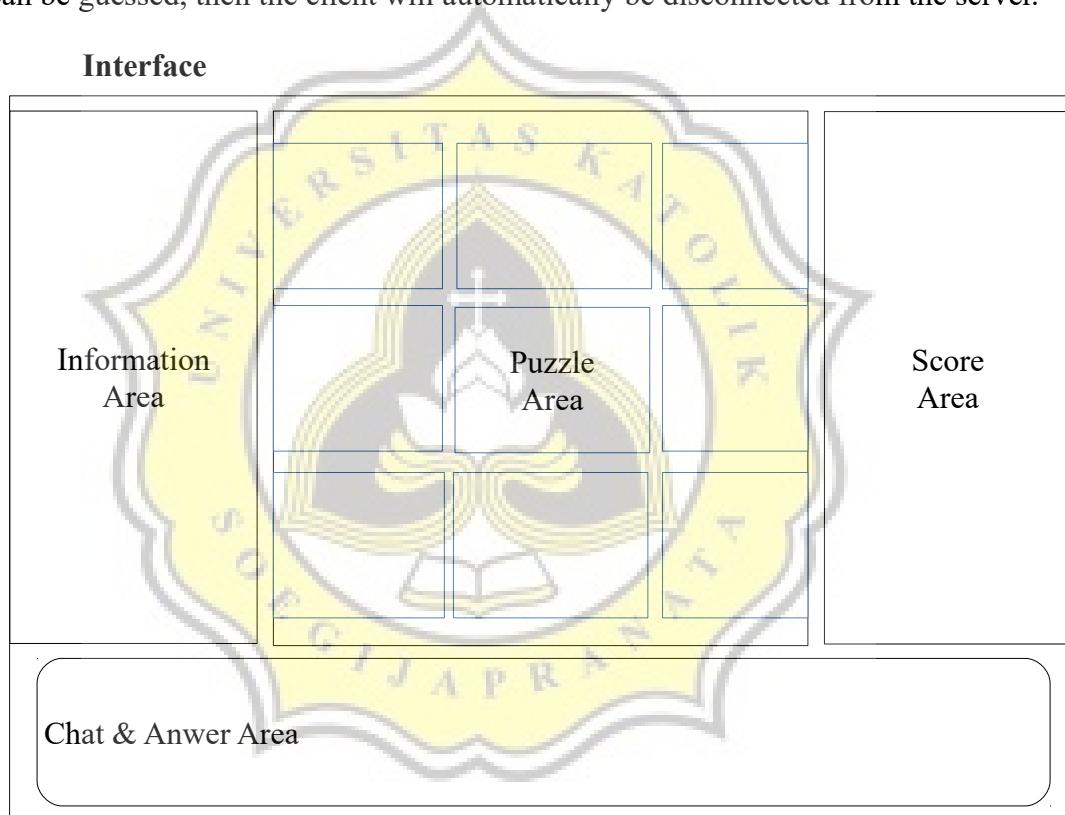


After the server is run, the server will wait for the connection from the client. The connected client will be asked to input the client's name as the client's identity. When there are 3 connected clients, the game can start with the initial player appointed by the server to start the movement first. When in the game one of the players suddenly disconnects, the server will delete all data associated with the client and close the socket and thread client. The game will take place until the answer on the server has run out. When the game has been completed then the client will be automatically disconnected and the server will remove the socket and thread clients2. The user interface on the server uses textmode.



Client**Flowchart**

Client will make connections to the server. If the server is active then the client can connect with the server. Once connected the client will be asked to enter the name as the identity of the client in the game / chat. The client will wait until the game can start. Clients who get the message "your move" from the server will start the initial movement in the game. When one of the boxes has been opened by one of its signature clients the game has started. If there is no more images that can be guessed, then the client will automatically be disconnected from the server.



Information Area = to display messages from the server

Puzzle Area = to select one of the provided boxes

Score Area = score display for each player and visible to all players

Chat & Answer Area = to exchange messages between clients, and to answer the image reply

Protocol

In-Game Protocol			
From	To	Message	Action
Server	Client	“SUBMITNAME”	The server gives the client a command to fill in the name of the place that has been provided
Client	Server	< String nama >	Client gives the name that has been inputted to the server. The server processes the name whether it is acceptable or not by checking the name that already exists or not, listed that has been created by the server
Server	Client	“NAMEACCEPTED”	The name has been confirmed by the server and can be used by the client, then the client is allowed to enter the game and can type dikolom chat that has been provided

Pre-Game Protocol			
From	To	Message	Action
Server	Client	“GAMEMESSAGE all players connected , wait for your turn”	The server will start the game if all the players are connected and have entered the game

Box Protocol			
From	To	Message	Action
Server	Client	“GAMEMESSAGE your move”	The server tells the first player to do the movement / selection of boxes that have been provided in the game
Client	Server	“BOX1”	The client notifies the server when he or she opens the number 1 box. The first server checks the location of the box whether the box has no owner or already owns the owner, and checks the player who is given the opportunity to open the box, if it

			has no owner then the server will inform the player that the box can be opened and send its location to another player to open the box
Server	Client	“VALID_MOVE_BOX 1”	The server confirms that the box can be opened and opened for the player, then the server will change the player's turn to the next player.
Server	Client	“GAMEMESSAGE Can't Move. Its Not Your Turn Yet / The Box already choosed”	The server confirmed that the box could not be opened because it was already open / not yet the player's turn to open the box
Server	Client	“OPPONENT_MOVE D”	Sends the location of the box that opens to the player 2 and opens the box that has been selected by the player given the server's opportunity
Server	Client	“OPPONENT_MOVE Z”	Sends the location of the box that is open to the player 3 and opens the box that has been selected by the player that is given the opportunity by the server

Timer Protocol			
From	To	Message	Action
Server	Client	“TIMESUP”	If the timer on the server is at number 30, then the server will automatically send a message to the client that the time has run out and the turn will be thrown to the next player. This causes the player can not select the option and must wait for the next turn.

Answer Protocol			
From	To	Message	Action
Client	Server	< String text_field >	Sending answers to the server via text_field is also used to send

			messages to other players.
Server	Client	“WINNER “	After the server checks that the client's response is correct, the server will give the player a score, and load the new image, and remove the mark on the box has been previously marked.

Chat Protocol			
From	To	Message	Action
Client	Server	< String Text_Field >	Sends messages written by the client to the server to send to other clients
Server	Client	< String “message” + name + command >	Messages that have been sent by the client will be broadcast to the entire client