

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Aplikasi chat messenger sebelumnya sudah pernah dibuat dengan mengangkat kasus dan penggunaan teknologi yang berbeda-beda, pada table 2.1 menunjukkan penelitian-penelitian yang sudah pernah dilakukan oleh beberapa penulis.

Pada skripsi yang ditulis oleh Andre yang berjudul Aplikasi panduan kepramukaan berbasis android menggunakan format pertukaran data JSON membahas mengenai penggunaan dari format pertukaran data JSON (Java Script Object Notation).

Pada jurnal yang ditulis oleh Gabriel L. Muller yang berjudul HTML5 WebSocket Protocol And It's Application To Distributed Computing menjelaskan mengenai cara kerja websocket yang diimplementasikan pada bahasa pemrograman HTML untuk membuat aplikasi sederhana untuk melakukan pertukaran data.

Pada skripsi yang ditulis oleh Fitri Hardianto dengan judul pengembangan Aplikasi Grup Chat di Android Menggunakan WebSocket berisi mengenai pembuatan aplikasi group chat sederhana yang hanya dapat berkirim pesan berupa teks dengan memanfaatkan websocket.

Kemudian pada skripsi yang ditulis oleh Mahardika Kurniawati yang berjudul aplikasi chat room menggunakan ajax dan php berisi mengenai

pembuatan chatroom dengan AJAX dan PHP dimana room yang dibuat dapat diberi sandi.

Pada skripsi yang ditulis oleh Ficki Nur Fadly Fauzian yang berjudul Pemanfaatan Websocket Untuk Pertukaran Data Web Secara Waktu Nyata: Studi Kasus Aplikasi Jejaring Sosial Sederhana berisi mengenai pembuatan aplikasi *chatting* dengan fitur yang menyerupai jejaring sosial facebook namun hanya dapat mengirimkan pesan teks.

Usulan yang diajukan dalam skripsi ini adalah membuat aplikasi *chat messenger* yang memiliki fitur *chat room* dan *private chat* dengan memanfaatkan teknologi websocket.

**Tabel 2.1 Tinjauan Pustaka**

Penulis / Parameter	Objek Penelitian	Metode/ Teknologi	Interface
Andre (2015)	Aplikasi panduan kepramukaan berbasis android menggunakan format pertukaran data JSON	JSON (Java Script Object Notation)	GUI
Gabriel L. Muller (2014)	HTML5 Websocket Protocol And It's Application To Distributed Computing	Websocket dan HTML	GUI
Fitri Hardianto (2015)	Aplikasi Grup Chat di Android Menggunakan Websocket	Websocket, AJAX dan PHP	GUI
Mahardika Kurniawati (2011)	Aplikasi Chat Room Menggunakan Ajax Dan Php	AJAX dan PHP	GUI
Ficki Nur Fadly Fauzian (2012)	Pemanfaatan Websocket Untuk Pertukaran Data Web Secara Waktu Nyata: Studi	Websocket, HTML, CSS	GUI

	Kasus Aplikasi Jejaring Sosial Sederhana		
I Kadek Utama Yoga	Implementasi Websocket pada aplikasi chat messenger berbasis android	Socket.io, mongoDB, Node.js	GUI

## 2.2 Dasar Teori

### 2.2.1 Chat

*Chat* adalah suatu fitur atau suatu program di Internet untuk berkomunikasi langsung sesama pengguna internet yang sedang online / yang sedang sama-sama menggunakan Internet. Komunikasi ini berupa pertukaran data baik berupa teks, file, gambar, suara maupun video (Hardianto,Fitri. 2015)

### 2.2.2 Websocket

*WebSocket* adalah standar baru untuk komunikasi *realtime* pada *Web* dan aplikasi *mobile*. *WebSocket* dirancang untuk diterapkan di *browser* web dan *server* web, tetapi dapat digunakan oleh aplikasi *client* atau *server*. *WebSocket* adalah protokol yang menyediakan saluran komunikasi *full-duplex* melalui koneksi TCP tunggal. Protokol *WebSocket* sudah di standarisasi oleh IETF sebagai RFC 6455 pada tahun 2011, dan API *WebSocket* di Web IDL sedang distandarisasi oleh W3C. (Hardianto,Fitri. 2015)

### 2.2.3 Socket.io

Socket.io adalah sebuah library yang dibuat menggunakan bahasa pemrograman JavaScript dan merupakan salah satu dari Websocket API (*Application Programming Interface*). Tidak hanya dari sisi client, library ini mendukung node.js sebagai server dengan sangat baik. Library ini memberikan performa yang sangat baik dalam transfer data secara real-time. (Julisman, Agung. 2014)

### 2.2.4 Node.js

Sebuah *platform* yang dirancang untuk pengembangan aplikasi *web* dan diimplementasikan di *server*. Node.js mampu melakukan proses baik secara *blocking (synchronous)* maupun *non-blocking (asynchronous)*. Dengan kemampuan Node.js yang dapat melakukan *non-blocking*, maka terbuka kemungkinan untuk membuat aplikasi *web instant messaging* yang ringan, *real-time* dan efisien. (nodejs.org)

### 2.2.5 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya

bahasa yang umum digunakan oleh programmer keluarga bahasa perograman C (JSON.org)

### **2.2.6 MongoDB**

MongoDB merupakan basis data noSQL (not only SQL) yang merupakan *document-oriented database* dan merupakan *open source project* yang tersedia di github (<https://github.com/mongodb/mongo>). (Julisman, Agung. 2015)

### **2.2.7 Socket Programming**

Socket programming merupakan pemrograman ke arah protocol jaringan. Dalam pemrograman socket ini ada banyak protocol jaringan yang dapat diimplementasikan seperti UDC, HTTP, TCP, serta UDP.

Ilustrasi dari socket programming dapat dicontohkan seperti dua orang yang berkomunikasi dengan telepon. Konsep telepon di mana penerima dan penghubung akan melakukan komunikasi dua arah, ketika keduanya dapat terhubung. Yang menariknya konsep ini menggunakan nomor ekstension. Sehingga ketika penghubung menelpon nomor telepon yang dituju, ada nomor ekstensi yang harus ditekan, contohnya 114 untuk layanan keuangan dan 115 untuk layanan komunikasi.

Untuk pendefinisian istilah-istilah dalam socket programming dapat dirinci sebagai berikut :

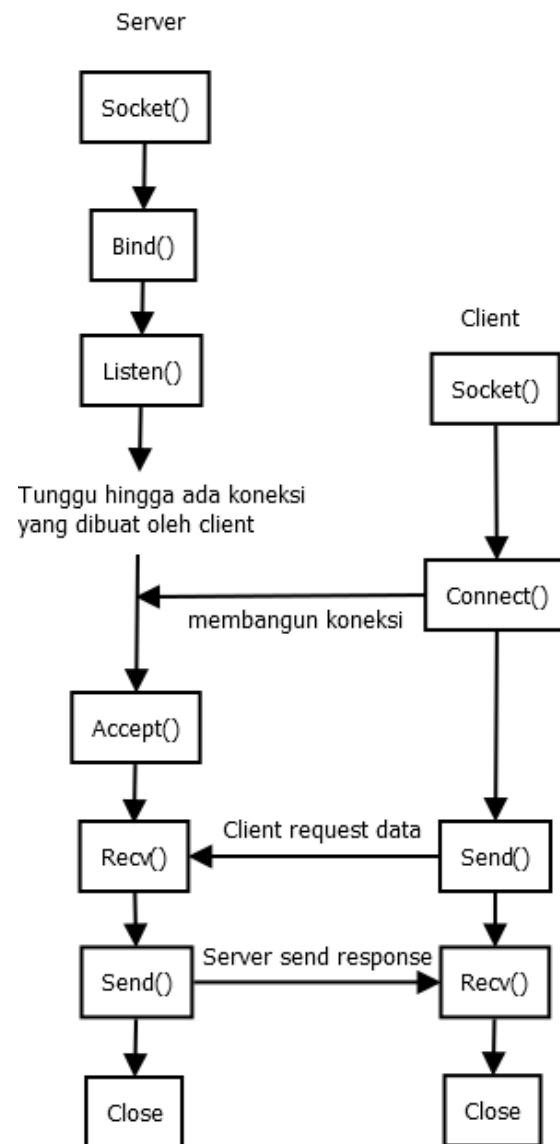
1. Server merupakan proses yang menjawab permintaan dari banyak client secara serempak. Jadi jika diilustrasikan, server ini merupakan customer service yang menerima telepon dari banyak client.
2. Client merupakan proses yang memulai suatu koneksi untuk melakukan hubungan komunikasi dengan server. Secara umum, akan ada banyak client yang mengadakan koneksi dengan suatu server dalam waktu yang sama. Jika diilustrasikan maka client merupakan orang yang menelpon customer service.
3. IP Address merupakan alamat IP yang berbeda yang dimiliki oleh masing-masing komputer untuk dapat berkomunikasi dalam suatu jaringan. Jika diilustrasikan dalam socket programming maka alamat IP merupakan nomor telepon dari suatu instansi yang besar yang memiliki banyak layanan.
4. Port Number merupakan nomor yang digunakan sebagai identifikasi atau layanan dari client yang ingin melakukan komunikasi. Jika bicara mengenai socket programming maka port number merupakan nomor ekstensi dari suatu nomor telepon. Sehingga ketika client ingin melakukan pembicaraan dengan server, client harus mengetahui IP address dan port untuk berkomunikasi. (Masya, Fajar., dan Fiade Andrew. 2015).

Analogi dari socket programming dapat digambarkan dalam beberapa fungsi.

Fungsi-fungsi tersebut diantaranya adalah :

1. Socket() digunakan untuk membangun *endpoint* dari komunikasi antara penghubung dengan customer service (client dan server).
2. Bind() digunakan untuk menentukan nomor telepon beserta nomor-nomor ekstensi yang tersedia (IP address dan port number).
3. Listen() digunakan ketika server atau customer service menunggu panggilan dari penghubung (server menunggu koneksi dari client).
4. Connect() digunakan ketika penghubung menelpon customer service dan meminta layanan (client melakukan koneksi ke server).
5. Accept() digunakan ketika customer service menerima telepon dan memberikan layanan pada penghubung (server menerima koneksi client).
6. Send() dan Recv() digunakan untuk merepresentasikan proses komunikasi antara customer service dan penghubung (client dan server).
7. Close() merupakan fungsi ketika client dan server sudah selesai berkomunikasi atau menutup telepon (client menghancurkan koneksi yang dibangun ke server).

Dan Jika digambarkan dalam bentuk diagram, maka cara kerja dari socket server dan socket client dapat dilihat pada gambar 2.1.



**Gambar 2.1** Diagram cara kerja socket server dan socket client