

***APLIKASI REAL TIME CHAT***  
**TUGAS BESAR SISTEM TERDISTRIBUSI DAN PARALEL**

Diajukan untuk memenuhi kelulusan mata kuliah Sistem Terdistribusi dan Paralel

Tahun akademik 2018/2019



Oleh:

Kelompok 6

Aldi Nafi Almaali      (130116)074

Halim Wajdi              (1301164207)

Muhammad Diaz R      (1301164135)

Rahmatian Jayanty S    (1301164086)

IF 40-08

PROGRAM STUDI S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY BANDUNG

2018/2019

# **BAB I**

## **PENJELASAN SISTEM/APLIKASI**

### **1.1 Sistem Terdistribusi**

Sistem Terdistribusi adalah sistem yang di mana komponen perangkat keras ataupun perangkat lunak yang terletak pada jaringan komputer yang saling berkomunikasi dan berkoordinasi hanya dengan menggunakan passing messages. Messages Passing mengirimkan pesan ke suatu proses dan bergantung pada proses dan infrastruktur pendukung untuk memilih dan menjalankan actual code untuk dijalankan .

### **1.2 Client Server**

#### **a. Pengertian Client Server**

Server-client merupakan arsitektur jaringan yang memisahkan Client dari Server. Setiap hal dari perangkat lunak Client dapat mengirimkan permintaan kepada server. Tipe spesifik dari server termasuk web servers, aplikasi server, file server, terminal server, dan mail server. Walaupun tujuannya berbeda, tetapi dasar arsitektur tetap sama.

Dari definisi pada sistem client/server harus terdapat satu atau beberapa server yang menyediakan layanan dan satu atau beberapa klien yang meminta layanan tersebut (tidak peduli apakah kondisi tersebut berada pada sebuah sistem jaringan ataupun stand-alone).

- **Server**

Server merupakan komputer yang menyediakan jenis layanan tertentu untuk client dalam suatu jaringan komputer.

- **Client**

Client merupakan komputer yang mengakses layanan tertentu kepada server dengan melalui jaringan komputer.

- **Kelebihan dan Kekurangan Client-Server**

- Kelebihan model jaringan client-server:

- 1. Memberikan keamanan yang lebih baik.
    2. Lebih mudah mengatur meskipun jaringan berskala besar, karena kontrol yang terpusat.
    3. Semua data maupun fasilitas terletak pada lokasi yang sentral.

- Kekurangan model jaringan Client-Server:

- 1. Butuh spesifikasi lebih / khusus untuk digunakan pada komputer server.
    2. Butuh seorang administrator yang handal dan professional
    3. Sangat bergantung pada komputer server

#### **b. Model Client-Server**

Ada beberapa model client/server yang penting untuk diketahui. Dimulai dari arsitektur mainframe hingga arsitektur client/server.

- **Arsitektur Mainframe**

Pada arsitektur ini, terdapat sebuah komputer pusat (host) yang memiliki sumber daya yang sangat besar, baik memori, processor maupun media penyimpanan. Melalui komputer terminal, pengguna mengakses sumber daya tersebut. Komputer terminal

hanya memiliki monitor/keyboard dan tidak memiliki CPU. Semua sumber daya yang diperlukan terminal dilayani oleh komputer host. Model ini berkembang pada akhir tahun 1980-an.

- **Arsitektur File Sharing**

Pada arsitektur ini komputer server menyediakan file-file yang tersimpan di media penyimpanan server yang dapat diakses oleh pengguna. Arsitektur file sharing memiliki keterbatasan, terutama jika jumlah pengakses semakin banyak serta ukuran file yang di shaing sangat besar. Hal ini dapat mengakibatkan transfer data menjadi lambat. Model ini populer pada tahun 1990-an.

- **Arsitektur Client/Server**

Karena keterbatasan sistem file sharing, dikembangkanlah arsitektur client/server. Salah satu hasilnya yaitu berupa software database server yang menggantikan software database berbasis file server. Dikenalkan pula RDBMS (Relational Database Management System). Dengan arsitektur ini, query data ke server dapat terlayani dengan lebih cepat karena yang ditransfer bukanlah file, tetapi hanyalah hasil dari query tersebut. RPC (Remote Procedure Calls) memegang peranan penting pada arsitektur client/server.

- **Model Two-tier**

Model Two-tier terdiri dari tiga komponen yang disusun menjadi dua lapisan: Client (yang meminta service) dan server (yang menyediakan service). Tiga komponen tersebut yaitu :

1. User Interface, yaitu antar muka program aplikasi yang berhadapan dan digunakan langsung oleh user.
2. Manajemen proses.
3. Database Model ini memisahkan peranan user interface dan database dengan jelas, sehingga terbentuk dua lapisan.

- **Model Three-tier**

Pada model ini disisipkan satu layer tambahan diantara user interface tier dan database tier. Tier tersebut dinamakan middle-tier. Middle-Tier terdiri dari bussiness logic dan rules yang menjembatani query user dan database, sehingga program aplikasi tidak bisa mengquery langsung ke database server, tetapi harus memanggil prosedur-prosedur yang telah dibuat dan disimpan pada middle-tier. Dengan adanya server middle-tier ini, beban database server berkurang. Jika query semakin banyak dan/atau jumlah pengguna bertambah, maka server-server ini dapat ditambah, tanpa merubah struktur yang sudah ada. Ada berbagai macam software yang dapat digunakan sebagai server middle-tier. Contohnya MTS (Microsoft Transaction Server) dan MIDAS.

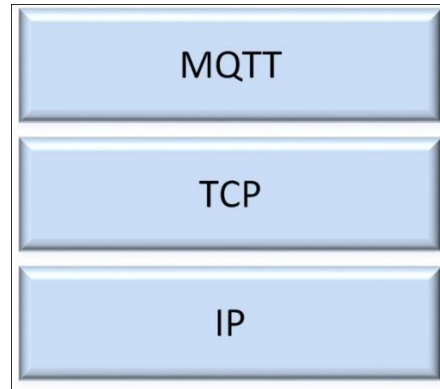
## **1.3 Protokol**

### **a. Pengertian MQTT**

MQTT (*Message Queuing Telemetry Transport*) adalah protokol pesan berbasis penerbitan standar berlangganan ISO. Ia bekerja di atas protokol TCP / IP. Ini dirancang untuk koneksi dengan lokasi terpencil di mana “jejak kode kecil” diperlukan atau

bandwidth jaringan terbatas. Pola perpesanan langganan langganan membutuhkan perantara pesan.

MQTT berjalan di atas **Transmission Control Protocol / Internet Protocol (TCP/IP)** sebagaimana digambar dalam skema berikut



MQTT lebih ringan dibandingkan dengan protokol HTTP 1.1 sehingga sangat cocok untuk digunakan pada perangkat berdaya rendah yang diharuskan mengirimkan dan menerima data dengan ukuran sekecil mungkin.

Sederhana nya *mqtt* adalah protokol untuk menyampaikan pesan dari server ke *client* maupun sebaliknya. Kenapa *mqtt* ? sebelumnya kita harus tahu dulu optimasi untuk IoT

1. Mengirim pesan secepat mungkin
2. Menimalisir *encoding* dan *decoding* data
3. Memanfaatkan *storage* sekecil mungkin

Sesuai optimasi diatas *mqtt* sudah memenuhi semuanya.

#### **b. Signal Control MQTT**

Ada 14 *signal control mqtt*, namun pada kasus ini kita akan menggunakan 3 saja:

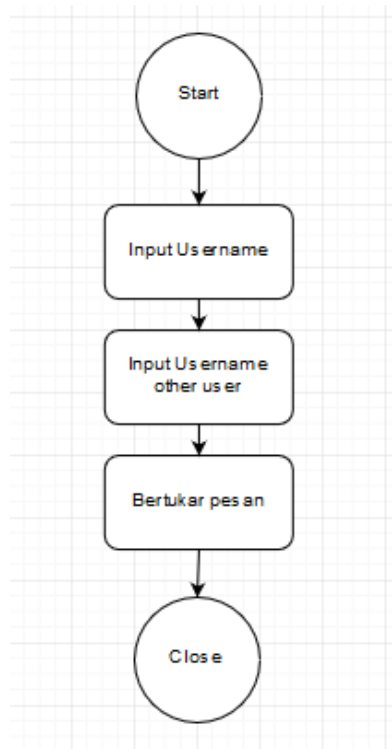
- *connect* -> connect ke server (broker)
- *subscribe* -> untuk mendapatkan pesan dengan topik tertentu
- *publish* -> mengirim pesan dengan topik tertentu

## BAB II

### RANCANGAN SISTEM

#### a. Desain

Pada aplikasi ini kami menggunakan desain aplikasi seperti yang digambarkan pada diagram flow dibawah ini

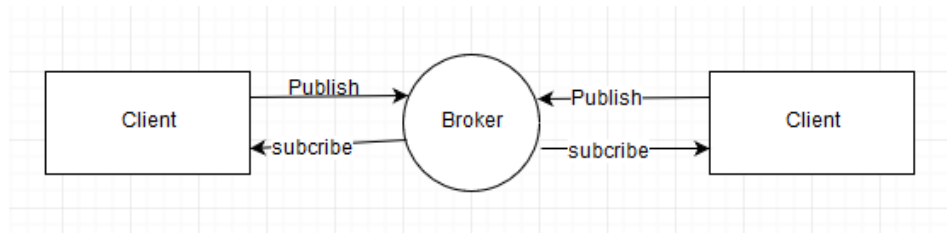


Gambar. 1 Diagram Flow Real time Chat

Pertama *user* menginputkan *username* yang berguna sebagai identitas pengguna. Setelah itu *user* menginputkan *username* dari *user* lain yang telah memiliki *username* dalam aplikasi untuk memulai pertukaran pesan. Ketika antar klien sudah terkoneksi maka *user* satu dan *user* yang lain dapat melakukan komunikasi melalui pertukaran pesan. *User* dapat mengetikkan pesan pada *text box message* yang disediakan dan dapat *submit* pesan untuk dikirim dengan menekan tombol *enter* setelah menekan tombol *enter* maka pesan akan diterima oleh *user* lain. Kemudian jika komunikasi telah selesai dilakukan maka *user* dapat menutup halaman sehingga secara otomatis koneksi terputus dan pesan yang telah terkirim tadi akan terhapus dengan sendirinya.

#### b. Arsitektur

Pada aplikasi *real time chat* ini kami menggunakan MQTT sebagai protokol yang berfungsi sebagai broker antar publisher dan subscriber. Aplikasi ini mampu bertukar pesan antara dua komputer dalam satu jaringan.

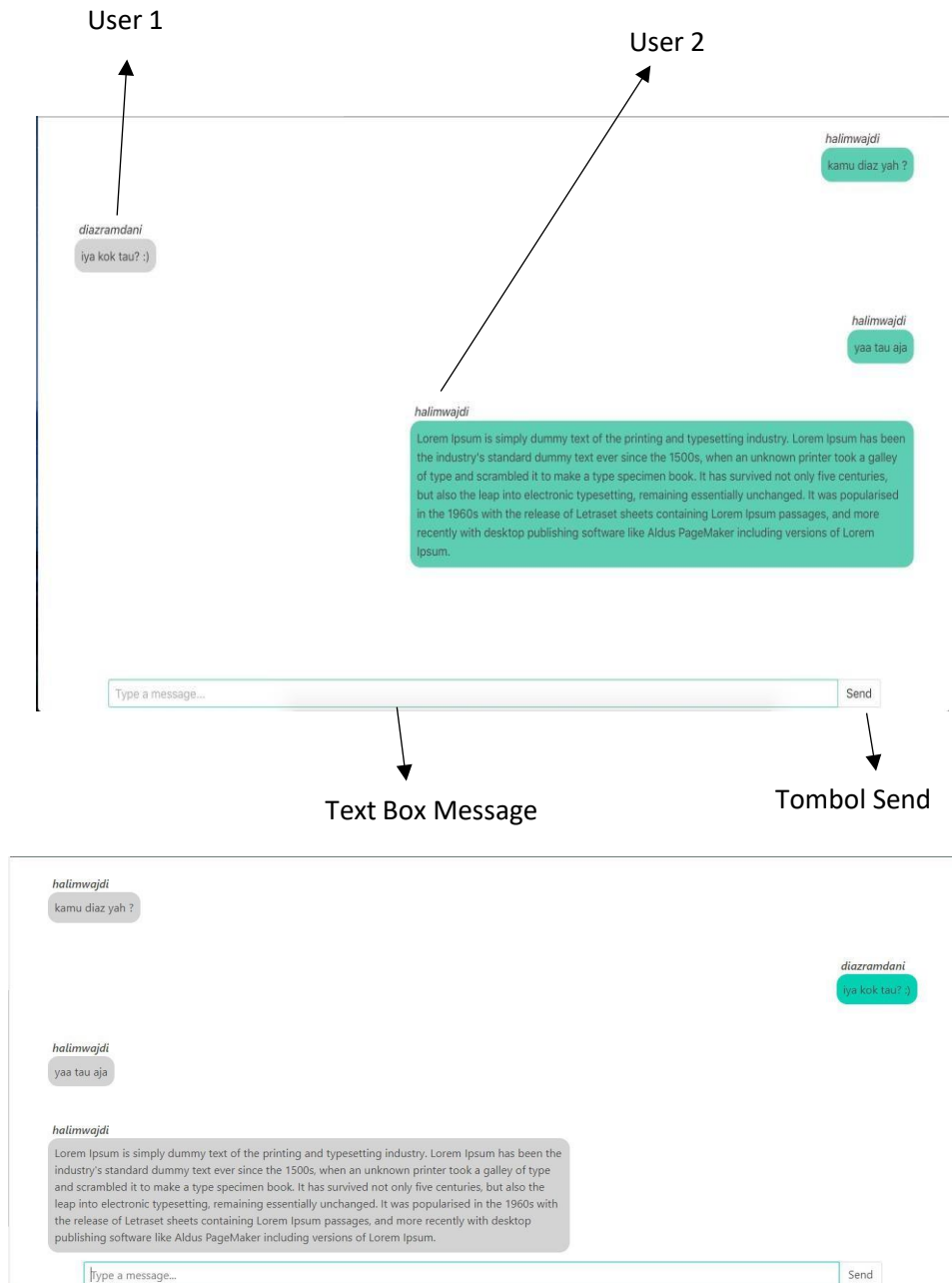


*Gambar. 2 Arsitektur MQTT pada Real Time Chat*

Kedua komputer yang berperan klient nantinya akan bertukar pesan menggunakan broker sebagai perantara. Klient akan mem-*publish* pesan ke broker dan klient lain akan men-*subscribe* pesan yang telah diterima oleh broker, begitu pula sebaliknya untuk Klient lain.

### BAB III

## SCREEN CAPTURE OUTPUT PROGRAM



Gambar diatas merupakan output tampilan aplikasi *Real Time Chat* yang kami buat. Dari gambar diatas dapat terlihat bahwa antar user 1 dan user 2 dapat saling melakukan komunikasi berupa bertukar pesan.

### PERAN SETIAP ANGGOTA

<b>Nama</b>	<b>Tugas</b>
Aldi Nafi Almaali	Membangun Aplikasi & desain
Halim Wajdi	Merancang arsitektur & desain
Muhammad Diaz	Membangun Aplikasi & Mencari solusi aplikasi hingga program terintegrasi
Rahmatian Jayanty	Menyusun Laporan