
Implementasi Metode Websocket Pada Aplikasi Donor Darah Real Time Gps Tracking Untuk Mengetahui Keberadaan Calon Pendonor Darah

Agustina Sri Astuti¹, Taqwan Thamrin^{2*}, Wiwin Susanty³,
Asep Trisna Setiawan⁴, Tia Tanjung⁵

^{1,2,3,4,5}Sistem Informasi, Fakultas Ilmu Komputer, Universitas Bandar Lampung
Lampung, Indonesia

¹ agustina.sri@ubl.ac.id, ² taqwan.thamrin@ubl.ac.id, ³ wiwin.susanty@ubl.ac.id, ⁴ asep.iot@ubl.ac.id,
⁵ tia.tanjung@ubl.ac.id

ABSTRACT – Blood is a body fluid that is vital to human life, which circulates in the heart and blood vessels. Heavy bleeding, either by accident or after surgery, requires a blood transfusion. The most important thing when meeting blood needs is to know the blood stock in the blood supply agency or hospital. Without realizing it, the presence of blood spread around sometimes escapes the patient when the patient is in need of a blood transfusion. On mobile devices there is a supporting feature in the form of Global Positioning System (GPS) is a computer-based information system that can store, manage and analyze and call information data that is geographically undifferentiated. By utilizing this feature, we can collect digital data of prospective blood donors by providing self-registration services through mobile android applications, as well as applying websocket to the application to display data in real time in the form of geolocation with the realtime position of the donor's existence, from there users can find out blood spread information and access it quickly when in a condition of need. From the research that has been done produces a real time blood donation gps tracking application with the name of Hemocare application.

Keywords: Blood, Blood Donors, GPS, Android, Websocket.

ABSTRAK – Darah adalah cairan tubuh yang memiliki peranan sangat penting bagi kelangsungan hidup manusia dan beredar melalui jantung serta pembuluh darah. Pendarahan yang hebat, baik karena kecelakaan atau saat setelah melakukan operasi sangat memerlukan tindakan transfusi darah. Hal yang sangat penting pada saat memenuhi kebutuhan darah adalah mengetahui stok darah yang ada pada instansi penyedia darah atau rumah sakit. Tanpa di sadari, keberadaan penyebaran darah disekeliling terkadang justru luput dari pasien ketika pasien sedang membutuhkan transfusi darah. Websocket yang didukung dengan fitur yang terpasang pada perangkat mobile berupa Global Positioning System (GPS) merupakan sistem informasi berbasis komputer yang dapat menyimpan, mengelola dan menganalisis serta memanggil data informasi yang bereferensi geografis. Dengan memanfaatkan fitur tersebut maka kita dapat mengumpulkan data digital para calon pendonor darah dengan memberikan layanan pendaftaran diri melalui mobile aplikasi android, serta menerapkan websocket pada aplikasi dan menampilkannya didalam bentuk geolokasi dengan posisi realtime keberadaan pendonor, dari sana pengguna dapat mengetahui informasi penyebaran darah dan mengaksesnya dengan cepat ketika sedang dalam kondisi membutuhkan. Dalam hal ini, websocket dapat menjadi salah satu solusi mendapatkan data penyebaran darah realtime berdasarkan teknologi GPS (Global Positioning System) yang terpasang pada perangkat smartphone. Hasil dari penelitian yang telah dilakukan berupa sebuah aplikasi donor darah real time gps tracking dengan nama aplikasi Hemocare.

Kata Kunci: Darah, Donor Darah, GPS, Android, Websocket.

1. PENDAHULUAN

Darah adalah cairan tubuh yang memiliki peranan sangat penting bagi kelangsungan hidup manusia dan beredar melalui jantung serta pembuluh darah. Darah mengangkut oksigen serta nutrisi untuk seluruh sel di dalam tubuh serta membawa hasil dari metabolisme sel. Volume darah di tubuh manusia dewasa adalah berkisar 3,6 liter (wanita) dan 4,5 liter (pria), [1]. Pendarahan hebat, baik karena kecelakaan atau saat setelah melakukan operasi sangat membutuhkan transfusi darah. Beberapa negara juga memiliki bank darah yang digunakan dalam sistem memang menjadi sarana yang sangat baik. Perangkat GPS yang tertanam pada smartphone juga

memenuhi kebutuhan transfusi darah bagi orang yang membutuhkan.

Smartphone kini sudah menjadi salah satu perangkat pendukung kegiatan bagi kehidupan banyak orang, mendapatkan informasi, membangun jadwal, mengerjakan pekerjaan dan lain-lainnya. Kecanggihan smartphone pada era ini sudah memiliki banyak fitur yang dapat mendukung segala hal terkait teknologi lainnya untuk bekerja secara bersama dalam satu buah perangkat saja, pemanfaatan smartphone untuk menjalankan sebuah dapat mendukung, karena hampir sebagian penduduk dunia kini sudah memiliki smartphone, maka smartphone



This work is licensed under a
Creative Commons Attribution 4.0 International License

DOI <http://dx.doi.org/10.36448/expert.v15i2.4438>
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 15 No. 2
Dec 31, 2025 – Hal. 232

dapat menjadi alternatif pembantu dalam mendistribusikan informasi yang dibutuhkan oleh kebanyakan orang. Dengan cara mendeteksi keberadaan menggunakan koneksi internet (Geolocation) dan dengan metode Websocket.

WebSocket merupakan standar terbaru yang digunakan untuk komunikasi *real-time* pada aplikasi web dan mobile. Protokol ini dirancang untuk diimplementasikan pada *web browser* dan *web server*, namun juga dapat digunakan oleh aplikasi klien maupun server. WebSocket menyediakan saluran komunikasi *full-duplex* melalui satu koneksi TCP.[2]. Oleh karena itu, penulis melakukan sebuah penelitian untuk membangun sebuah perangkat lunak yang terdukung dengan GPS yang tertanam didalam smartphone dan di dukung oleh metode Websocket untuk mengembangkan sistem untuk mengetahui titik-titik lokasi calon pendonor darah, dan mengangkat penelitian dengan judul “IMPLEMENTASI METODE WEBSOCKET PADA APLIKASI DONOR DARAH REALTIME GPS TRACKING UNTUK MENGETAHUI KEBERADAAN CALON PENDONOR DARAH”.

2. Metodologi

A. Studi Lapangan (Field Research)

Dalam pelaksanaan penelitian ini penulis menerapkan penelitian kualitatif deskriptif yang memiliki tujuan dalam mengetahui kenyataan sosial, dengan melihat dunia apa adanya, dengan memahami suatu permasalahan secara mendalam. Jenis penelitian kualitatif deskriptif yang diterapkan merupakan Design and Creation yang merupakan suatu jenis penelitian dalam pengembangan produk di bidang teknologi informasi serta komunikasi. Proses Design and Creation menerapkan prinsip learning by making. pemilihan penelitian ini dikarenakan konsep dari Design and Creation sesuai untuk mengelola penelitian penulis. Lokasi penelitian yang penulis gunakan adalah masyarakat Fakultas Ilmu Komputer (FIK) Universitas Bandar Lampung.

B. Sumber Data Penelitian

Sumber data dalam penelitian ini yang digunakan penulis dalam mendapatkan data guna tujuan membantu penulisan adalah sebagai berikut:

Penyebaran Kuisioner, dalam mengumpulkan informasi, penulis membuat beberapa pertanyaan dalam bentuk kuisioner penyebaran darah yang disebarluaskan untuk mendapatkan nilai yang dijadikan acuan dalam pembangunan sistem. Kuisioner disebarluaskan kepada masyarakat FIK Universitas Bandar Lampung sebagai sample pengguna. Dari pertanyaan kuisioner hasil yang di dapat dari 60 orang responden dan 10 pertanyaan di Tarik kesimpulan bahwa penyebaran calon pendonor darah tidak di ketahui secara luas, tidak mudah mendapatkan informasi mengenai calon pendonor darah dan responden memberikan pendapat bahwa di butuhkan sebuah sistem informasi mengenai penyebaran calon pendonor darah, sebuah sistem yang menampilkan para

calon pendonor darah dalam sebuah Maps digital untuk mengetahui keberadaannya secara langsung.

A. Global Positioning System

Merupakan sebuah sistem yang dapat menunjukkan titik lokasi geografis di atas permukaan bumi dalam waktu yang cepat merupakan pengertian dari Global Positioning System (GPS). GPS merupakan sistem satelit navigasi dalam menentukan titik lokasi dari penerima sinyal satelit. Sistem kordinat bujur dan lintang akan memberikan informasi titik geografis. Dalam posisi ini akan diketahui menggunakan alat penerima yang bisa memberikan informasi terhadap kordinat bujur dan lintang yang diberikan dari sinyal GPS yang diterimanya[3].

B. Metode

Websocket merupakan komunikasi realtime dengan standar baru pada aplikasi mobile dan Web. Websocket digunakan dalam perancangan server web dan browser web, juga digunakan bagi aplikasi server atau client [4]. Websocket adalah protokol yang memberikan saluran jaringan komunikasi full duplex melalui koneksi TCP tunggal [5]. WebSocket menyediakan pendekatan yang lebih efisien dalam komunikasi *full-duplex* karena tidak memiliki overhead header HTTP, khususnya untuk aplikasi yang memerlukan pembaruan *real-time* dengan cepat. HTTP pada dasarnya dirancang sebagai protokol tanpa kondisi dinamis dan tidak mendukung komunikasi *full-duplex* secara *real-time*, sehingga WebSocket dikembangkan untuk memenuhi kebutuhan tersebut.[6].

C. RabbitMQ

RabbitMQ diibaratkan seperti tukang pos dan kantor pos. RabbitMQ akan menerima pesan dari sender/publisher/pengirim dan selanjutnya akan diteruskan kembali (forward and pushing) kepada receiver/subscriber ataupun penerima dari sana dapat di simpulkan bahwa pesan sendiri memiliki sifat tidak terjadi pada waktu yang sama di saat menjalankan pesan tidak harus bergantung ataupun menunggu pada kesiapan entity lain. Sehingga di keadaan ini pengirim tidak harus menunggu pemberitahuan dari penerima tentang kesuksesan data yang telah di kirimkan[7]. Pesan pesan yang telah di kirimkan akan di tampung pada broker (tepatnya pada barisan elemen/queue) sampai akhirnya akan di lanjutkan pada tujuannya, masing-masing pihak (penerima dan pengirim) bisa melakukan aktivitas/kegiatan lain tanpa harus terganggu dengan kesanggupan maupun konfirmasi dari masing-masing pihak. RabbitMQ juga dapat berperan dalam menangani koneksi internet/jaringan internet mati ataupun tidak stabil dalam sementara waktu, dengan itu server bisa menyimpan data sementara pada RabbitMQ setelah koneksi stabil data akan kembali di teruskan kembali ke perangkat[8].

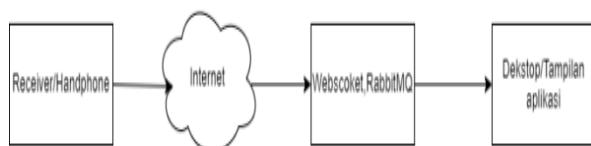
D. Mapbox API



Mapbox adalah platform pembangun yang dapat digunakan dalam menyelesaikan masalah yang berhubungan dengan data, peta, dan analisis spasial. digunakan untuk Android merupakan perangkat sumber terbuka dalam menampilkan peta ke dalam aplikasi Android[9]. Mapbox API juga bersifat open-source sehingga dengan kode terbuka dapat memungkinkan dalam mengembangkan maupun membangun sebuah aplikasi tanpa harus mengalami kendala dalam lisensi[10].

E. Implementasi Metode

Implementasi WebSocket pada aplikasi donor darah ini bekerja dengan memanfaatkan mekanisme *request* dan *response* untuk membuka status *open-connection* dengan server. Dengan demikian, server dan klien dapat saling berkomunikasi serta melakukan pertukaran data dua arah secara *real-time* ketika terdapat data baru, tanpa perlu mengulang keseluruhan protokol. Penerapan metode *Websocket* pada penelitian ini digunakan untuk melakukan *request* data lokasi *user* yang selalu di *Update* selama beberapa detik untuk mendapatkan data yang *realtime* tanpa harus ada *refresh* halaman untuk melihat perubahan data, metode *websocket* ini di implementasikan di menu *find people* atau *maps* dari aplikasi *Hemocare* yang bertujuan agar data titik lokasi dari pendonor dapat terus di perbarui di dalam *maps* yang nantinya akan di tampilkan ke pengguna lain agar memudahkan dalam pencarian lokasi titik terdekat dari pencarian donor.



Gambar 1. Alur kerja *Websocket* dan *Rabbitmq*

3. Hasil dan Pembahasan

A. Interface Program

Berikut ini merupakan beberapa tampilan halaman yang ada dalam program yang telah dibuat:

1. Halaman awal pengenalan aplikasi

Pada halaman ini akan diisikan deskripsi-deskripsi singkat mengenai keunggulan aplikasi pencari calon pendonor darah. Gambar halaman awal pengenalan aplikasi dapat dilihat pada gambar dibawah ini:

Mapbox merupakan bagian dari setiap proses dalam pembuatan peta web maupun peta perangkat mobile. Mapbox Maps Software Development Kit yang



Hemocare
Hemocare hadir dengan tujuan untuk menghubungkan seluruh pihak-pihak dari dalam satu aplikasi.



Gambar 2. Halaman Pengenalan Aplikasi

2. Halaman splashscreen

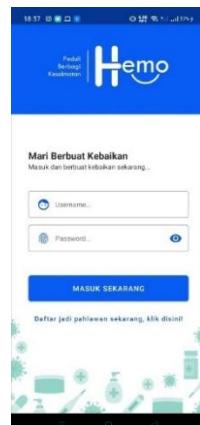
Merupakan halaman yang akan dimunculkan Ketika aplikasi pertama dibuka. Gambar halaman splashscreen aplikasi dapat dilihat pada gambar dibawah ini :



Gambar 3. Halaman Splashscreen

3. Halaman login

Halaman ini digunakan untuk masuk ke dalam sistem bagi mereka yang sudah mendaftarkan diri sebagai bagian dari sistem. Gambar halaman login dapat dilihat pada gambar dibawah ini:



Gambar 4. Halaman Login



This work is licensed under a
Creative Commons Attribution 4.0 International License

4. Halaman daftar semua pengguna

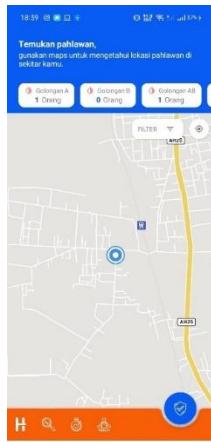
Halaman ini berisikan seluruh data pengguna yang terdaftar di dalam system dengan status setuju, status setuju didapatkan apabila pengguna telah setuju atas persyaratan yang ditetapkan oleh system. Gambar halaman daftar semua pengguna aplikasi dapat dilihat pada gambar dibawah ini:



Gambar 5. Halaman Daftar Pengguna

5. Halaman Find People atau Maps

Halaman ini berisikan halaman Maps yang menampilkan data pengguna, serta terdapat total counter data pengguna berdasarkan golongan darah. Gambar halaman find people dapat dilihat pada gambar dibawah ini:



Gambar 6. Halaman Find People atau Maps

B. Penerapan Kode Program

Penerapan fungsional untuk proses transfer data kedalam RabbitMQ di tempatkan pada 2 (dua) halaman yaitu :

Main Activity merupakan halaman inti dari system, dari berbagai macam halaman yang ada di dalam aplikasi seperti, dashboard, find people, daily activity dan juga profile merupakan halaman yang dibangun dengan menggunakan fragment activity yang di tempatkan di dalam halaman main activity. Penempatan kode pengiriman data di dalam halaman ini bertujuan agar setiap aplikasi di buka dan masuk ke berbagai macam halaman lain yaitu dashboard, find people, daily activity dan juga profile, system akan tetap mengirimkan perintah untuk pengiriman data lokasi ke dalam antrian pesan. Berikut source code dari proses pengiriman data yang ada di dalam halaman Main Activity :

```
private void amqpSender(JSONObject data) {
    try {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        factory.setHost("rmq2.vpnrik.id");
        factory.setPort(5672);
        factory.setUserName("ktr_tanggamus");
        factory.setPassword("ktr_tanggamus");
        factory.setVirtualHost("/ktr_tanggamus");

        Connection connection = factory.createConnection();
        Channel channel = connection.createChannel();

        channel.queueDeclare("QUEUE_NAME", true, false, false, null);
        channel.basicPublish("", "mq.tag", null, data.toString().getBytes());
        System.out.println(" [x] Sent '" + data.toString() + "'");

        connection.close();
    } catch (IOException | TimeoutException e) {
        throw new RuntimeException("PROBLEM", e);
    }
}
```

Gambar 7. Kode Program Pengiriman Data

Perintah diatas digunakan untuk mengirimkan data ke dalam server rabbitmq.

```
LocationCallback locationCallback = new LocationCallback() {
    @Override
    public void onLocationResult(LocationResult locationResult) {
        if (locationResult == null) {
            return;
        }
        for (Location location : locationResult.getLocations()) {
            latMe = location.getLatitude();
            lngMe = location.getLongitude();

            LATDATA = String.valueOf(latMe);
            LONGDATA = String.valueOf(lngMe);

            sendLocationFunction(latMe, lngMe);
            updateLocationFunction(latMe, lngMe);

            JSONObject json = new JSONObject();
            JSONArray locatosArray = new JSONArray();

            try {
                json.put("GUID", GUID);
                locatosArray.put(LATDATA);
                locatosArray.put(LONGDATA);
                json.put("PHONE", modelUser.getPHONE());
                json.put("FULLNAME", modelUser.getFullname());
                json.put("GENDER", modelUser.getGender());
                json.put("LOCATION", locatosArray);
                locatosArray.put(1);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
        if (modelUser.getSTATUS().equals("ya")) {
            amqpSender(json);
        }
    }
};
```

Gambar 8. Kode Program Pengiriman Lokasi

Sedangkan kode diatas digunakan untuk mendapatkan lokasi update dari system dan selanjutnya setiap lokasi berhasil di dapatkan maka akan menjalankan perintah pengiriman data ke server rabbitmq. Serta Find People / Maps Pada halaman ini, diterapkan kode program untuk menerima data dari server rabbitmq, yang nantinya untuk data lokasi yang dikirimkan akan di tampilan ke dalam tampilan maps.

```
private void getLocationFunction() {
    try {
        StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        factory.setHost("rmq2.vpnrik.id");
        factory.setPort(5672);
        factory.setUserName("ktr_tanggamus");
        factory.setPassword("ktr_tanggamus");
        factory.setVirtualHost("/ktr_tanggamus");

        Connection connection = factory.createConnection();
        Channel channel = connection.createChannel();

        channel.queueDeclare("QUEUE_NAME", true, false, false, null);
        System.out.println(" [x] Waiting for messages. To exit press CTRL+C");

        DeliverCallback deliverCallback = (consumerTag, delivery) -> {
            String message = new String(delivery.getBody(), StandardCharsets.UTF_8);
            System.out.println(" [x] Received '" + message + "'");
        };

        channel.basicConsume("QUEUE_NAME", true, deliverCallback, consumerTag -> { });
    } catch (IOException | TimeoutException e) {
        throw new RuntimeException("PROBLEM", e);
    }
}
```

Gambar 9. Kode Program Penerimaan Data

Perintah diatas digunakan untuk menerima data dari dalam server rabbitmq ke dalam sistem.





Gambar 10. Kode Program Visualisasi Data dalam Bentuk Marker

Sedangkan pada baris kode di atas digunakan untuk menampilkan data yang diterima sistem untuk visualisasian ke dalam bentuk marker di dalam halaman maps.

C. Uji Validasi Titik Lokasi

Uji validasi ini dilakukan dengan cara melihat pergerakan marker di dalam sistem saat pengguna melakukan perjalanan, saat aplikasi Hemocare aktif maka pergerakan dari pendonor dapat terdeteksi di dalam sistem aplikasi, sedangkan saat aplikasi Hemocare tidak aktif pergerakan dari pendonor tidak akan muncul di dalam sistem aplikasi. Berikut di bawah ini gambar dari uji validasi titik lokasi yang telah dilakukan:

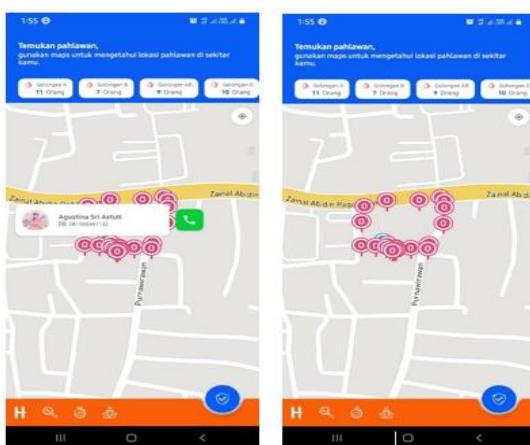
4. Kesimpulan

Dari penggeraan yang dilakukan dalam proses pembangunan sistem donor darah real time gps tracking ini. Penulis menyimpulkan beberapa hal terkait pembahasan mengenai prosedur dan metode yang berjalan didalam sistem:

1. penggunaan Mapbox sebagai media interaktif kepada pengguna dirasa cukup menjadi salah satu sarana penunjang untuk mengetahui keberadaan calon pendonor dengan akurat.
 2. Penambahan informasi terkait pendonor cukup efektif dalam mengetahui kesiapan pendonor.
 3. GPS yang terpasang pada masing masing perangkat smartphone dapat diterapkan dengan baik untuk menunjang kebutuhan data yang Update setiap saat, sehingga kondisi terkini dari keberadaan calon pendonor dapat lebih mudah ditemukan.

DAFTAR PUSTAKA

- [1] N. K. Firani, *Mengenali Sel-Sel Darah dan Kelainan Darah*. Malang: UB Press, 2018.
 - [2] B. J. Kuncoro, I. Ruslianto, S. Bahri, J. Rekayasa, and S. Komputer, “Coding : Jurnal Komputer dan Aplikasi IMPLEMENTASI HEALTH CHECKS MONITORING PADA LOAD BALANCER DI SISI WEB SERVER Coding : Jurnal Komputer dan Aplikasi Gambar 1 . Arsitektur Umum Load Balancer .,” vol. 07, no. 01, pp. 33–42, 2019.
 - [3] A. Amir and B. Isnanto, “Prototipe Aplikasi Untuk Mengetahui Tata Letak Atm Di Pangkalpinang Pada Smartphone Android,” *Jurnal Teknologi Informatika dan Komputer Atma Luhur*, vol. 1, no. September, pp. 1–6, 2014.
 - [4] A.-I. A.B., “Implementasi Teknologi Websocket dalam Pengembangan Sistem Berbagi Lokasi Berbasis Web,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 9, pp. 950–959, 2017.
 - [5] I. Fette and A. Melnikov, “Rfc 6455: The websocket protocol,” 2011, *RFC Editor*.
 - [6] V. Wang, F. Salim, and P. Moskovits, “The websocket protocol,” in *The Definitive Guide to HTML5 WebSocket*, Springer, 2013, pp. 33–60.



Gambar 11. Uji Validasi Sistem



- [7] R. Maharjan, M. S. H. Chy, M. A. Arju, and T. Cerny, “Benchmarking message queues,” in *Telecom*, MDPI, 2023, pp. 298–312.
- [8] C. F. Permatasari and H. Dhika, “Optimasi Jalur Transfer Data dari HTTP menjadi MQTT pada IoT menggunakan Cloud Services,” *JISA(Jurnal Informatika dan Sains)*, vol. 1, no. 2, pp. 67–72, 2018, doi: 10.31326/jisa.v1i2.446.
- [9] T. Lameirão, M. Melo, and F. Pinto, “Augmented reality for event promotion,” *Computers*, vol. 13, no. 12, p. 342, 2024.
- [10] A. M. Salim and A. Junaidi, “Pembuatan Aplikasi Sistem Informasi dan Pemetaan Geografis Destinasi Wisata Berbagai Kategori di Kota Malang,” *Jurnal Komputer Antartika*, vol. 1, no. 2, pp. 39–46, 2023.

