BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

2.1.1. Proses Distribusi Barang dengan Armada

Distribusi adalah bagian penting dalam rantai pasok, memastikan bahwa produk sampai kepada konsumen dalam kondisi optimal, secara efisien, dan tepat waktu. Dalam praktik manajemen rantai pasok, sebuah perusahaan harus membuat keputusan untuk 5 area utama yaitu produksi, persediaan, lokasi, transportasi, dan informasi. Keputusan yang diambil oleh perusahaan atas area utama akan sangat mempengaruhi kemampuan dan efektivitas rantai pasok secara keseluruhan [4]. Transportasi sebagai salah satu keputusan perusahaan menjadi aspek yang sangat penting dalam manajemen rantai pasok karena di sini representasi perusahaan berinteraksi langsung dengan konsumen akhir.

Bagi perusahaan logistik, transportasi sering kali mengalami berbagai masalah dalam praktiknya seperti keterlambatan pengantaran barang, pemilihan rute yang kurang efisien, dan kondisi kendaraan yang kurang optimal. Permasalahan tersebut menjadikan operasional perusahaan menjadi tidak efisien dan mempengaruhi kepuasan pelanggan. Permasalahan yang sama juga dialami perusahaan distributor pupuk. Pada penelitian oleh Romli dkk., perusahaan tersebut mengalami kesulitan dalam pengolahan data, pemantauan stok, dan permasalahan dalam waktu pengiriman. Untuk mengatasi masalah tersebut, Romli dkk., memberikan solusi dengan mengintegrasikan seluruh proses bisnis mulai dari pengelolaan produk, pemesanan, pengiriman, dan pelaporan hasil distribusi. Sistem ini memungkinkan agen sebagai pembeli mampu untuk melakukan pemesanan secara daring. Sementara itu, pihak distributor sebagai penjual juga mampu untuk melakukan pengelolaan data produk, permintaan pengiriman, pengiriman, dan juga dapat menghasilkan laporan secara otomatis. Dengan pengimplementasian sistem informasi dalam proses distribusi pupuk, perusahaan distributor dapat melaksanakan proses bisnisnya dengan lebih efisien yang membantu pengambilan

keputusan yang lebih baik [5].

2.1.2. *GPS Tracking*

Dalam sektor distribusi, pengelolaan armada yang baik menjadi faktor kunci perusahaan. Sistem pemantauan berbasis GPS dapat membantu perusahaan untuk melakukan optimasi rute perjalanan, mengurangi waktu tempuh, dan menekan biaya operasional perusahaan. Lebih jauh lagi, penggunaan sistem ini dapat mendukung pengambilan keputusan terkait penjadwalan perawatan kendaraan lewat pengambilan data yang dapat diolah untuk memberikan informasi terkait kondisi kendaraan yang ada [6].

Penerapan GPS juga dilakukan oleh PT Pos Indonesia sebagaimana diteliti oleh Damayanti dkk. untuk melakukan pelacakan kendaraan dan barang secara real-time. Meskipun sistem ini dapat memberikan hasil yang baik berupa capaian rata-rata skor efisiensi yang tinggi sebesar 0,953. Namun persentase rute efisien hanya sebesar 40% sedangkan 60% sisanya masih memerlukan optimasi. Temuan ini menunjukkan bahwa penerapan GPS tracker masih belum efektif karena data yang dihasilkan belum terintegrasi dengan sistem pengambilan keputusan [7].

Penerapan sistem pemantauan berbasis GPS untuk pengelolaan armada telah menunjukkan dampak positif dengan meningkatkan efisiensi operasional armada distribusi. Penelitian oleh Bilan dkk, menunjukkan bahwa sistem pemantauan berbasis GPS dapat membantu perusahaan untuk melakukan optimasi rute, mengurangi waktu tempuh, dan menekan biaya operasional perusahaan. Namun demikian, hasil penelitian oleh Damayanti dkk menunjukkan bahwa meskipun rata-rata skor efisiensi yang dihasilkan tergolong tinggi, namun presentasi rute efisien yang dihasilkan masih rendah karena data yang diperoleh belum diintegrasikan dengan sistem pengambilan keputusan. Dari kedua penelitian tersebut dapat disimpulkan bahwa integrasi antara pemantauan GPS dan sistem pengambilan keputusan menjadi faktor penting untuk meningkatkan efisiensi pengelolaan armada distribusi.

2.1.3. *QR Code Validation*

Penggunaan *Quick Response Code* (QR Code) sebagai teknik pengkodean data sendiri menjadi semakin populer seiring dengan perkembangan aplikasi *mobile*. Keunggulan yang ditawarkannya adalah dapat dipindai dari 360 derajat dan memiliki keakuratan tinggi walaupun terdapat sedikit kerusakan pada gambarnya. Berdasarkan penelitian oleh Damanik dkk. penggunaan QR Code mampu meningkatkan validitas data penerima produk serta mengurangi risiko terjadinya penyalahgunaan atau kejahatan dalam teknologi informasi [8].

Meskipun demikian, penelitian lain menunjukkan bahwa penggunaan sistem QR Code masih di bawah potensi maksimal. Hasil simulasi oleh Taufik dkk., menunjukkan bahwa sistem QR Code memiliki performa pemindaian terbaik dengan rata-rata 2 detik per item. Jika dibandingkan dengan sistem manual yang membutuhkan waktu rata-rata per item mencapai 3-5 menit, memang penggunaan sistem kode QR memberikan hasil yang sangat cepat. Namun, performa sistem QR Code masih berada di bawah potensi maksimalnya karena adanya latensi *response time* API mungkin terjadi karena beban server, *query* basis data yang belum optimal, dan kendala pada lingkungan koneksi dan *deployment* [9].

QR Code sudah terbukti menjadi solusi yang efisien dalam proses validasi data. Damanik dkk. menemukan bahwa penggunaan *QR Code* dapat meningkatkan validitas data penerima produk serta mencegah terjadinya penyalahgunaan informasi. Di sisi lain, penelitian oleh Taufik menunjukkan bahwa meskipun penggunaan *QR Code* sangat efisien dalam mengurangi waktu pemindaian rata-rata per item, kinerjanya masih belum optimal dikarenakan faktor teknis seperti beban server dan belum optimalnya *query*. Dari kedua penelitian tersebut dapat disimpulkan bahwa meskipun sistem berbasis QR Code memiliki kecepatan pemindaian yang sangat cepat, keberhasilan implementasinya juga bergantung pada faktor teknis untuk mendukung kecepatan dan keandalan proses validasi.

2.1.4. Algoritma Pemilihan Sopir Otomatis

Algoritma *Nearest Neighbor* (NN) dianggap menjadi algoritma yang sesuai karena bekerja dengan prinsip membandingkan jarak antar titik untuk menemukan

kedekatan data yang ada. Algoritma ini bekerja dengan mengambil titik terdekat dengan titik acuan yang sesuai dengan kriteria dan akan terus berulang sampai ditemukan bahwa tidak ada titik terdekat yang sesuai dengan kriteria. Dalam konteks algoritma pemilihan sopir, algoritma *Nearest Neighbor* dapat digunakan untuk membandingkan kesesuaian antara pesanan pelanggan dengan kondisi sopir di lapangan seperti jarak pelanggan dengan sopir dan stok LPG yang sedang dibawa. Penelitian oleh Lestari dkk., menunjukkan bahwa implementasi algoritma *Nearest Neighbor* dapat menekan biaya pendistribusian dengan memberikan rute yang lebih optimal dan efisien [10]. Keberhasilan penelitian tersebut membuktikan bahwa prinsip kerja algoritma *Nearest Neighbor* dapat diadaptasi untuk konteks pemilihan pemilihan sopir dalam distribusi.

2.2. Celah Penelitian

Dari tinjauan pustaka di atas, dapat diidentifikasi bahwa:

1) Pada aspek distribusi barang dengan armada, sistem informasi yang ada umumnya sudah mencakup proses bisnis hingga pelaporan, namun demikian, sebagian besar sistem belum dilengkapi dengan mekanisme penjadwalan sopir dan validasi pengantaran digital.
2) Pada aspek penggunaan teknologi GPS *tracking*, sistem pemantauan kendaraan menggunakan GPS sudah banyak digunakan untuk meningkatkan efisiensi armada, namun berfokus pada pemantauan secara real-time dan belum terintegrasi dengan mekanisme pengambilan keputusan yang dapat mendukung penjadwalan sopir otomatis.
3) Pada aspek penggunaan QR Code untuk validasi, pemanfaatan QR Code terbukti dapat meningkatkan efisiensi dan akuntabilitas data, namun umumnya sistem yang dikembangkan masih berdiri sendiri dan belum diintegrasikan dengan sistem penjadwalan sopir.
4) Pada aspek pengembangan, algoritma *Nearest Neighbor* (NN) dinilai sesuai untuk menentukan jarak terdekat karena bekerja dengan prinsip perbandingan jarak terdekat antar titik. Dalam konteks distribusi gas

LPG algoritma ini dapat dimanfaatkan sebagai sistem pendukung pengambilan keputusan untuk menentukan sopir yang terdekat dengan pelanggan, dengan mempertimbangkan berbagai kriteria.

Sebagian besar penelitian terkait sistem distribusi di Indonesia masih berpusat pada penerapan GPS *tracker* sebagai alat untuk melakukan pemantauan armada [6]. Namun sistem tersebut umumnya belum terintegrasi secara penuh dengan sistem pengambilan keputusan untuk penjadwalan sopir, sehingga efisiensi rute menjadi berkurang. Penelitian terhadap penggunaan QR Code sebagai metode validasi juga umumnya berfokus pada validasi dan optimasi kecepatan pemindaian [8] [9], tetapi belum diintegrasikan dengan sistem manajemen distribusi. Selain itu, penelitian terkait algoritma *Nearest Neighbor* masih berfokus pada optimasi rute distribusi tanpa mempertimbangkan kondisi lapangan sopir sebagai kriteria pemilihan [10]. Celah penelitian yang muncul adalah perlunya pengembangan sistem penjadwalan otomatis dan validasi pengantaran gas LPG berbasis GPS dan QR Code yang terintegrasi, dengan menggunakan algoritma *Nearest Neighbor* untuk menentukan sopir yang terdekat dengan pelanggan yang memenuhi kriteria. Sistem ini diharapkan mampu meningkatkan efisiensi perusahaan, akuntabilitas data pelanggan, serta meningkatkan keandalan proses distribusi perusahaan.

Tabel 2.1 Tabel Perbandingan Penelitian

Penulis	Sistem Distribusi barang	GPS Tracking	Validasi QR Code	Algoritma Penjadwalan
Romli dkk.	Menggunakan sistem informasi yang mencakup berbagai proses bisnis	Tidak menggunakan GPS	Tidak menggunakan QR Code	Tidak menggunakan algoritma penjadwalan
Billan	Menggunakan sistem pemantauan berbasis GPS	Sudah menggunakan GPS tapi belum memiliki sistem	Tidak menggunakan QR Code	Tidak menggunakan algoritma penjadwalan

Penulis	Sistem Distribusi barang	GPS Tracking	Validasi QR Code	Algoritma Penjadwalan
		yang membantu pengambilan keputusan		
Damayanti dkk.	Menggunakan GPS untuk melacak kendaraan dan muatan secara *real-time*	Sudah menggunakan GPS tetapi belum mendukung sistem pengambilan keputusan	Tidak menggunakan QR Code	Tidak menggunakan algoritma penjadwalan
Damanik dkk.	Membangun sistem informasi yang bisa memindai QR Code untuk validitas penerima barang	Tidak menggunakan GPS	Menggunakan QR Code untuk data penerima barang	Tidak menggunakan algoritma penjadwalan
Taufik dkk.	Membangun sistem manajemen gudang berbasis QR Code dengan integrasi fullstack JavaScript	Tidak menggunakan GPS	Menggunakan QR Code untuk validasi data barang di gudang	Tidak menggunakan algoritma penjadwalan
Lestari dkk.	Menggunakan NN untuk penentuan rute distribusi	GPS digunakan untuk menentukan matriks jarak	Tidak menggunakan QR Code	Algoritma Nearest Neighbor digunakan

Penulis	Sistem Distribusi barang	GPS Tracking	Validasi QR Code	Algoritma Penjadwalan
				untuk menentukan rute distribusi
Penulis	Mengembangkan sistem informasi distribusi LPG dengan fitur penjadwalan otomatis sopir dan validasi penerimaan digital	Menggunakan GPS untuk melacak sopir secara berkala	Menerapkan QR Code untuk melakukan validasi penerimaan LPG	Algoritma Nearest Neighbor digunakan untuk menentukan sopir terdekat yang memenuhi kriteria

2.3. Landasan Teori

2.3.1. Distribusi Barang dengan Armada

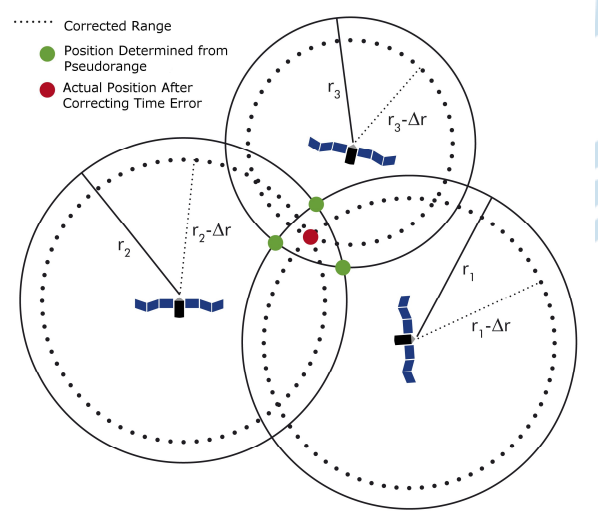
Dalam sistem rantai pasok, distributor adalah pihak yang membeli produk dalam jumlah yang besar dari produsen dan menyalurkannya kembali kepada pembeli dalam bentuk yang lebih kecil [4]. Dalam konteks pendistribusian gas LPG, distributor memiliki tanggung jawab untuk memastikan ketersediaan produk bagi masyarakat luas, mengingat gas LPG merupakan kebutuhan rumah tangga yang penting. Oleh karena itu, proses distribusi gas LPG membutuhkan perencanaan yang matang terutama pada aspek penjadwalan, rute, dan penentuan stok tiap armada. Perencanaan yang baik membantu distributor dalam memastikan seluruh pelanggan dapat dilayani dengan baik serta menumbuhkan kepercayaan pelanggan terhadap perusahaan.

Pendekatan yang dipilih oleh perusahaan distributor LPG umumnya mengadopsi konsep *Milk Run Deliveries*, yaitu metode pengiriman di mana sebuah armada akan melakukan pengiriman ke beberapa lokasi penerima yang telah

dioptimalkan rutenya. Metode ini dianggap dapat lebih menekan biaya operasional armada dibandingkan dengan metode *direct deliveries* karena mengoptimalkan sumber daya armada yang ada untuk melakukan pengiriman ke beberapa pelanggan sekaligus. Selain itu pendekatan ini juga memungkinkan adanya dukungan dari sistem penjadwalan otomatis yang dapat melakukan pembaruan rute dan jadwal kirim secara otomatis sesuai dengan lokasi armada yang terdekat dengan pelanggan.

2.3.2. *Global Positioning System*

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang bekerja dengan prinsip pengukuran jarak *receiver* dari satelit [11]. Sistem ini terdiri dari 24 satelit yang mengorbit bumi pada ketinggian 19.000 km. Satelit satelit tersebut telah disusun sedemikian rupa agar sebuah *receiver* di bumi dapat selalu menerima sinyal dari minimal 4 satelit yang ada. Setiap satelit akan secara terus menerus memancarkan sinyal radio yang berisi informasi posisi dan waktu transmisi.



Gambar 2.1 Proses Trilaterasi

GPS dapat menentukan lokasi *receiver* dengan mengalkulasikan selisih waktu antara sinyal dikirim dan diterima dengan kecepatan cahaya untuk memperoleh jarak antara satelit dan *receiver*. Dengan data jarak dari minimal 3 satelit yang ada, *receiver* dapat melakukan trilaterasi yaitu metode untuk

menghitung titik perpotongan antara 3 atau lebih lingkaran dengan radius yang diketahui untuk menemukan lokasi *receiver* di permukaan bumi. Dalam penerapannya, GPS mampu memberikan informasi lokasi dengan tingkat akurasi yang tinggi yang dapat dimanfaatkan dapat bidang navigasi.

2.3.3. *Quick Response Code*

Quick Response Code (QR Code) adalah kode matriks dua dimensi yang dibuat untuk mengatasi masalah limitasi kapasitas yang ada pada *barcode* satu dimensi [12]. Dengan menggunakan struktur geometris, QR Code dapat menampung data dalam kapasitas yang jauh lebih besar dibandingkan dengan *barcode* 1 dimensi. QR Code juga dirancang agar dapat dipindai dari berbagai sudut (omnidirectional), hal ini menjadikannya lebih mudah dipindai dengan berbagai perangkat. Keunggulan lain dari QR Code adalah adanya mekanisme *error correction*, sehingga apabila terjadi kerusakan pada saat pemindaian QR Code, pemindaian akan tetap berhasil. Dalam implementasi QR Code pada aplikasi sebaiknya mengikuti standar ISO/IEC 18004:2024 untuk memberikan pengalaman penggunaan QR Code yang maksimal kepada pengguna.

2.3.4. Algoritma *Nearest Neighbor*

Algoritma *Nearest Neighbor* adalah algoritma heuristik konstruktif yang paling natural dan banyak digunakan oleh *Vehicle Routing Problem* [13]. Alur kerja algoritma *Nearest Neighbor* dapat digambarkan sebagai berikut

1. Tentukan titik depot
2. Pilih pelanggan terdekat dari posisi saat ini yang berasal dari cluster yang belum dikunjungi
3. Jika total muatan rute saat ini + permintaan cluster ≤ kapasitas:
    1) Tambahkan pelanggan
    2) Tandai cluster pelanggan sebagai sudah dikunjungi
    3) Perbarui lokasi saat ini
    4) Kembali ke langkah 2
4. Jika total muatan rute saat ini + permintaan cluster ≥ kapasitas:

1) Tutup rute dengan kembali ke depot
2) Buat rute dengan kendaraan baru dan tetapkan depot sebagai lokasi saat ini
3) Kembali ke langkah 2

5. Ulangi langkah sampai minimal 1 pelanggan dari setiap cluster telah dikunjungi.

Dalam menghitung jarak, algoritma *Nearest Neighbor* bisa dikombinasikan dengan berbagai rumus, seperti Rumus Haversine yang digunakan untuk mengukur *great-circle distance* pada permukaan bumi berdasarkan koordinat latitude dan longitude. Rumus Haversine dinyatakan sebagai berikut:

$$d = 2Rarc\sin\left(\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)sin^2\left(\frac{\Delta\lambda}{2}\right)\lambda_1}\right)$$

Keterangan:

φ_1: latitude titik awal

φ_2: latitude titik tujuan

λ_1: longitude titik awal

λ_2: longitude titik tujuan

$\Delta\varphi$: $\varphi_2 - \varphi_1$

$\Delta\lambda$: $\lambda_2 - \lambda_1$

R: jari-jari bumi (6371 km)

Algoritma ini bersifat *greedy*, yang berarti algoritma ini mengambil sebuah keputusan berdasarkan kondisi saat itu tanpa mempertimbangkan dampak keputusan secara global. Hal ini mengakibatkan algoritma ini tidak selalu memberikan hasil yang optimal. Algoritma *Nearest Neighbor* memiliki kompleksitas algoritma $O(n^2)$ di mana n adalah jumlah total pelanggan yang membuatnya memiliki waktu eksekusi yang singkat untuk dataset kecil hingga menengah. Algoritma ini mudah diimplementasikan mengingat logika yang sederhana. Namun, karena sifat *greedy*, algoritma ini tidak selalu menghasilkan solusi yang optimal.

2.3.5. *REST API*

Representational State Transfer (REST) adalah pola arsitektural yang pada saat ini sering digunakan dalam pengembangan web modern [14]. REST API sendiri dibangun atas beberapa prinsip, prinsip pertama yaitu *client-server*, prinsip ini mementingkan pemisahan antara *client* dan server sehingga lebih modular. Prinsip kedua adalah *stateless*, prinsip ini mengharuskan setiap *request* yang ada mengandung seluruh informasi yang dibutuhkan, tidak tergantung pada *request* sebelumnya. Hal ini memberikan keuntungan berupa skalabilitas dan *maintenance* yang lebih baik. Prinsip ketiga adalah *cacheability*, yaitu hasil dari *response* dapat disimpan sementara untuk efisiensi. Prinsip keempat, *uniform interface*, yaitu setiap *client* dapat melakukan *request* dengan standar yang sama, seperti menggunakan metode HTTP. Prinsip kelima yaitu *layered system,* prinsip ini mengharuskan adanya lapisan-lapisan penyusun sistem yang disusun secara hierarkis untuk memisahkan tanggung jawab antar lapisan.

Pendekatan REST menjadi pendekatan yang populer karena sederhana, ter standarisasi, dan serbaguna untuk banyak kasus di dunia nyata. Namun, kemudahan yang ditawarkan dari pendekatan REST sendiri justru berujung pada ketergantungan akan implementasi *best practices*. Pada praktiknya, sering kali ditemukan implementasi REST API yang justru tidak mengikuti sepenuhnya prinsip dari REST API itu sendiri. Hal ini akan berujung pada performa API yang menjadi lambat dan tidak efisien. Oleh karena itu penerapan *best practices* dalam pengembangan API menjadi hal yang krusial. *Best practices* dalam pengembangan REST API mencakup implementasi peraturan penamaan *endpoint,* desain parameter *input* dan *output* yang baik, dokumentasi yang lengkap, dan selalu mempertimbangkan performa [15].

2.3.6. PostGIS

PostGIS adalah ekstensi dari Post**g**reSQL yang memungkinkannya melakukan penyimpanan, *indexing*, dan *querying* data geospasial baik dalam format 2D maupun 3D [16]. PostGIS menyediakan tipe data spasial seperti *points*, *lines*, *polygons*, dan *multi geometries* untuk komputasi data spasial. Salah satu fitur yang

ditawarkan oleh PostGIS adalah *spatial indexing*, fitur ini memungkinkan pengguna untuk melakukan pencarian lokasi berdasarkan data spasial dengan cepat. Selain itu, PostGIS juga menawarkan banyak fungsi yang dapat digunakan untuk mengolah data spasial, termasuk fungsi untuk menghitung jarak dengan ST_Distance, luas area dengan ST_Area, dan *intersection* geometri dengan ST_Intersection.

PostGIS mengimplementasikan model dari *Open Geospatial Consortium* (OGC) yaitu *Geometry* sebagai sebuah tipe data yang mengikuti standar *Simple Features Access* (SFA). Tipe data *Geometry* meliputi tipe data *atomic* seperti *Point, LineString, MultiPolygon,* dan *GeometryCollection.* PostGIS juga dapat diintegrasikan dengan berbagai *tools* lainnya seperti QGIS, GeoServer, MapServer, ArcGIS, dan Tableau, hal ini menjadikan PostGIS sebagai alat yang fleksibel untuk berbagai kebutuhan komputasi data spasial.

2.3.7. Pengembangan dengan Metode Waterfall

Metode Waterfall adalah salah satu metode *Software Development Lifecycle* (SDLC). Prinsip kerja metode ini adalah sekuensial, di mana pada pengerjaan dalam tiap tahapnya berlangsung secara terurut dari awal hingga akhir [17]. Hal ini dikarenakan setiap tahapnya memiliki ketergantungan dengan tahapan sebelumnya. Dengan sifat ketergantungan yang dimiliki atas tahap sebelumnya, metode Waterfall lebih cocok apabila diimplementasikan dalam proyek yang sudah terdokumentasi kebutuhan dan permasalahannya dengan baik [18]. Tahapan dalam pengembangan perangkat lunak menggunakan metode Waterfall adalah sebagai berikut:

1. Analisis Kebutuhan
2. Perancangan Sistem
3. Implementasi
4. Pengujian
5. Pemeliharaan

Tahapan analisis kebutuhan adalah tahapan di mana seluruh informasi meliputi permasalahan yang dialami dan kebutuhan dikumpulkan. Proses ini

mengandalkan penyampaian informasi yang baik agar seluruh kebutuhan pengguna dapat teridentifikasi secara komprehensif. Tahapan perancangan sistem merupakan langkah untuk menghasilkan rancangan sistem berdasarkan analisis dari kebutuhan yang dikumpulkan. Pada tahap ini dilakukan perancangan arsitektur sistem, pemodelan alur kerja, dan perancangan basis data. Setelah rancangan telah selesai dibuat, tahapan selanjutnya adalah implementasi. Tahap implementasi merupakan proses realisasi atas rancangan yang dibuat dalam bentuk sistem yang siap uji. Tahapan ini memerlukan kedisiplinan agar sistem yang dibuat memenuhi kebutuhan pengguna. Selanjutnya adalah tahapan pengujian, tahapan ini dilakukan untuk memastikan sistem yang dihasilkan dapat bekerja dengan baik tanpa kesalahan. Sering kali pada tahapan inilah yang menyebabkan proses pengembangan perangkat lunak menjadi memakan waktu yang lebih lama apabila terdapat kesalahan maupun perubahan kebutuhan dari pengguna. Tahapan terakhir adalah pemeliharaan, tahapan ini mencakup perbaikan dari kesalahan sistem maupun penyesuaian terhadap kebutuhan baru pengguna.

2.3.8. Push Notification

Push Notification adalah salah satu pendekatan dalam menyampaikan suatu pesan dari aplikasi kepada pengguna, tujuannya adalah untuk mendapatkan perhatian dari pengguna untuk membuka aplikasi tersebut [19]. Salah satu penyedia paling ternama dari *Push Notification* adalah Firebase Cloud Messaging (FCM). Salah satu keunggulan FCM adalah keandalannya dalam mengirimkan pesan dan pemberitahuan ke banyak aplikasi yang berbeda sistem operasinya. Penggunaan FCM sangat relevan karena menyediakan layanan push notification tanpa biaya tambahan. FCM sendiri bekerja dengan cara memberikan token/id unik khusus untuk tiap penggunanya, token ini kemudian disimpan ke basis data aplikasi. Ketika aplikasi ingin mengirimkan pesan, *server* aplikasi hanya perlu memberikan permintaan beserta dengan token dari pengguna tertentu kepada FCM. FCM kemudian bertanggung jawab untuk meneruskan pesan tersebut kepada perangkat tujuan yang diberikan oleh *server*. Dengan menggunakan mekanisme ini, FCM dapat menyebarkan pesan dengan efisien dan *real-time*.