BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

"Teknologi *Global positioning System (*GPS) Untuk Pelapor dan Penjemputan Sampah Berbasis Android"oleh Saeful Bahri, Satia Suhada, dan Jamal Maulana Hudin (2019). Penelitian ini mengambil teknologi yang di gunakan *GPS* yang digunakan untuk menentukan sebuah lokasi tumpukan sampah.

"Aplikasi Bengkel Online Menggunakan *Global Positioning System* (*Gps*) Berbasis Android Pada Cv. Rumah Otomotif"oleh Badri Zaki, dan Syahrizal Dwi Putra (2018). Hasil dari penelitian ini maka dengan adanya aplikasi bengkel online menggunakan *GPS* ini para pengendara bermotor dapat melakukan *booking* untuk mendapatkan jasa maupun produk, aplikasi bengkel ini tempat yang tepat untuk para pemilik bengkel mendapatkan pelanggan.

"Perancangan Aplikasi Mobile Bengkelku Sebagai Informasi Alamat Bengkel Resmi Sepeda Motor Di kota Yogyakarta Menggunakan *GPS* Berbasis Android" oleh Rizqi Fitriansyah Antasari, dan Kusrini". Penelitian ini menghasilkan aplikasi pencarian bengkel resmi sepeda motor dan menentukan jalur yang akan dilalui oleh pengguna menuju bengkel, aplikasi bengkelku dibuat melalui tahap analisis kebutuhan dan analisis kelayakan.

"Perancangan Aplikasi Untuk Mendapatkan Mekanik Sepeda Motor Menggunakan *Google Maps Api*" oleh Ratna Salkiawati, M.Hadi Prayitno, danDadi Ulul Wilhadad (2019). Penelitian ini menghasilkan aplikasi untuk mendapatkan mekanik sepeda motor.

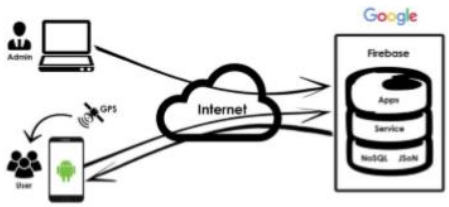
Tabel 2.1. Tinjauan Pustaka

No	Sumber	Judul	Teknologi	Hasil
1.	Saeful Bahri, Satia Suhada, dan Jamal Maulana Hudin (2019)	Teknologi *Global positioning System (*GPS) Untuk Pelapor dan Penjemputan Sampah Berbasis Android	*Global Positioning System* (GPS), Android, *Black box*	aplikasi ini memanfaatkan Black Box sebagai pengujiannnya, *Global positioning system* pada google maps
2.	Zaki Badri dan Dwi Syahrizal (2018)	Aplikasi Bengkel Online Menggunakan *Global Positioning System* (Gps) Berbasis Android Pada Cv. Rumah Otomotif	*Global Positioning System* (GPS), Android, *waterfall*	Aplikasi bengkel online
3.	.Rizqi Fitriansyah Antasari dan Kusrini	Perancangan Aplikasi Mobile Bengkelku Sebagai Informasi Alamat Bengkel Resmi Sepeda Motor Dikota Yogyakarta Menggunakan GPS Berbasis Android	*Global Positioning System* (GPS), Android, *Google Maps*	Aplikasi "Bengkelku" berbasis android dibuat melalui tahap analisis yaitu dengan menggunakan analisis kebutuhan dan analisis kelayakan.
4.	Ratna S, M Hadi Prayitno dan Dadi Ulul W	Perancangan Aplikasi Untuk Mendapatkan Mekanik Sepeda Motor Menggunakan Google Maps Api	*Google Maps Api, Location Based Service*	Aplikasi yang mempermudah para pengendara motor untuk mendapatkan mekanik terdekat.
5.	Agus Tri Wahyu	Implementasi Firebase Untuk Pemesanan Servis Sepeda Motor Berbasis Android Studi Kasus Di Kota Yogyakarta	*Google Maps Api, Firebase, Android*	Aplikasi layanan pemesanan servis motor berbasis Android

2.2. Dasar Teori

2.2.1 Firebase

Firebase yakni model layanan yang bekerja di belakang layar dan menghubungkan aplikasi *mobile* ke *cloud storage*. *Firebase Realtime Database* adalah database yang di-*host* di *cloud*. Data disimpan sebagai *JSON* dan disinkronkan secara *realtime* ke setiap klien yang terhubung. Ketika anda membuat aplikasi lintas-*platform* dengan *SDK Android, iOS,* dan *JavaScript*, semua *klien* akan berbagi sebuah *instance Realtime Database* dan menerima *update* data terbaru secara otomatis. (firebase, 2018).



Gambar 2. 1 Arsitektur Sistem Firebase

Semua data *Firebase Realtime Database* disimpan sebagai objek *JSON*. Bisa dianggap basis data sebagai *JSON* tree yang di-host di awan. Tidak seperti basis data *SQL*, tidak ada tabel atau rekaman. Ketika ditambahkan ke *JSON tree*, data akan menjadi simpul dalam struktur *JSON* yang ada. Meskipun basis data menggunakan *JSON tree,* data yang tersimpan dalam basis data bisa diwakili sebagai tipe bawaan tertentu yang sesuai dengan tipe *JSON* yang tersedia untuk membantu anda menulis lebih banyak kode yang bisa dipertahankan. Ada empat metode untuk menulis data ke *Firebase Realtime Database*:

Metode	Penggunaan umum
`setValue()`	Menulis atau mengganti data ke jalur yang didefinisikan, seperti `users/<user-id>/<username>`.
`push()`	Tambahkan ke daftar data. Setiap kali Anda memanggil `push()`, Firebase akan menghasilkan ID unik, seperti `user-posts/<user-id>/<unique-post-id>`.
`updateChildren()`	Memperbarui beberapa kunci untuk jalur yang didefinisikan tanpa mengganti semua data.
`runTransaction()`	Memperbarui data kompleks yang bisa rusak karena pembaruan bersamaan.

Gambar 2. 2 Metode Menulis Data ke *Firebase*

Untuk operasi tulis dasar, Anda bisa menggunakan *setValue*() untuk menyimpan data ke referensi yang ditetapkan, menggantikan data yang ada di jalur tersebut. Fungsi dalam pengambilan data melalui *Firebase*:

Listener	Callback kejadian	Penggunaan biasa
`ValueEventListener`	`onDataChange()`	Membaca dan mendengarkan perubahan untuk seluruh konten jalur.
`ChildEventListener`	`onChildAdded()`	Mengambil daftar item atau mendengarkan penambahan daftar item. Disarankan untuk digunakan dengan `onChildChanged()` dan `onChildRemoved()` untuk memantau perubahan daftar.
	`onChildChanged()`	Mendengarkan perubahan pada item dalam daftar. Gunakan dengan `onChildAdded()` dan`onChildRemoved()` untuk memantau perubahan daftar.
	`onChildRemoved()`	Mendengarkan item yang dibuang dari daftar. Gunakan dengan `onChildAdded()` dan`onChildChanged()` untuk memantau perubahan daftar.
	`onChildMoved()`	Gunakan dengan data diurutkan untuk mendengarkan perubahan dalam prioritas item.

Gambar 2. 3 *Callback* Kejadian dalam Pengambilan Data *Firebase*

Untuk menambahkan *listener* kejadian, gunakan metode *addValueEventListener()* atau *addListenerForSingleValueEvent().* Untuk menambahkan listener kejadian anak, gunakan metode *addChildEventListener(). Metode onDataChange()* untuk membaca cuplikan statis konten pada jalur tertentu, seperti yang telah ada pada saat kejadian. Metode ini terpicu satu kali ketika *listener* terpasang dan terpicu lagi setiap kali terjadi perubahan data, termasuk anaknya. *Callback* kejadian meneruskan cuplikan yang berisi semua data di lokasi tersebut, termasuk data anak. Jika tidak ada data, cuplikan yang dikembalikan adalah *null*. *Metode onDataChange()*

dipanggil setiap kali terjadi perubahan data pada referensi *database* yang ditetapkan, termasuk perubahan ke anaknya. (Firebase, 2018).

2.2.2 Android

Android merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah *OS* lainnya yang berkembang dewasa ini. *OS* lainnya seperti Windows Mobile, i-*Phone OS,* Symbian, dan masih banyak lagi. Akan tetapi, *OS* yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.(Hermawan, 2011 : 1).

2.2.3 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment* / IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada *IntelliJ* IDEA. Selain sebagai editor kode dan fitur *developer IntelliJ* yang andal, *Android Studio* menawarkan banyak fitur yang meningkatkan produktivitas. Anda dalam membuat aplikasi *Android*, seperti: Sistem build berbasis *Gradle* yang fleksibel, *Emulator* yang cepat dan kaya fitur, lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat *Android*, terapkan Perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi, Template kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel, *framework* dan fitur

pengujian yang lengkap, fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya, dukungan C++ dan NDK, dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi Google *Cloud Messaging* dan *App Engine.*

2.2.4 Global Positioning System (GPS)

Dalam penelitian Mulyadi.B,dkk yang berjudul "Aplikasi Sistem Pemesanan Jasa *Laundry* (*E-laundry*) Berbasis Android", GPS *(Global Positioning System)* adalah suatu sistem *navigasi* menggunakan lebih dari 24 satelit *MEO (Medium Earth Orbit* atau *Middle Earth Orbit)* yang mengelilingi bumi sehingga penerima-penerima sinyal di permukaan bumi dapat menangkap sinyalnya. *GPS* mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan arah, dan waktu satelit mengorbit pada ketinggian 12.000 *mil* di atas bumi dan mampu mengelilingi bumi dua kali dalam 24 jam.

2.2.5 Google Maps API

Google Maps API adalah suatu *library* yang berbentuk *javascript* yang berguna untuk memodifikasi peta yang ada di *Google Maps* sesuai kebutuhan. Dengan adanya Google *Maps API* ini pengguna dapat memaksimalkan kemampuan pemetaan Google untuk tujuan-tujuan dan manfaat tertentu secara spesifik. (Elian, dkk. 2012:2 dalam penelitian yang dilakukan oleh Daniel Oktodeli Sihombing yang berjudul : "Perancangan Aplikasi *Web* Untuk Pencarian Lokasi Dan Rute Rumah Sakit Berbasis *Google Maps API*").