

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 TINJAUAN PUSTAKA

Aplikasi yang memanfaatkan Apache Cordova 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.

Skripsi yang ditulis Wikjatmiko, Zhulfi Bajra *Pembelajaran Kosakata Bahasa Isyarat Menggunakan Phonegap Berbasis Android* berisi penggunaan phonegap untuk membangun aplikasi berbasis android.

Skripsi yang ditulis oleh Kresno, F. X. Dwi *Informasi Penggajian Dosen Di Stmik Akakom Berbasis Hybrid* berisi penggunaan mySQL untuk penyimpanan data menggunakan phonegap.

Skripsi yang ditulis oleh Gunawan, Arif *Manajemen Dan Forecast Keuangan Pribadi Menggunakan Phonegap* berisi pengimplementasian cordova, Ionic, Angular JS dengan media penyimpanan SQLite.

Jurnal yang ditulis oleh Putri Dwi Intan Ra'uufi & Yanuar Sinatra *Rancangan Bangun Aplikasi Delivery Order Berbasis Android Di Ageng Bakery Group Blitar* menjelaskan mengenai cara kerja aplikasi yang menggunakan teknologi Java, Apache Cordova, Android Studio.

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

Usulan yang diajukan dalam skripsi ini adalah penyimpanan data lokasi untuk media sosial pelancong dengan fitur penyimpanan data lokasi yang mengimplementasikan teknologi Apache Cordova.

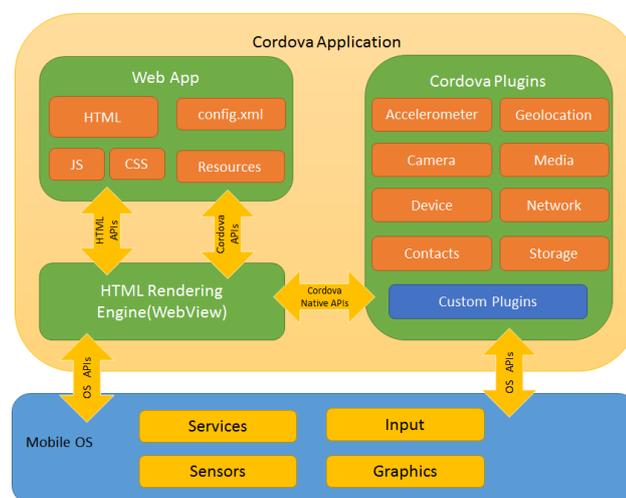
Tabel 2.1 Tinjauan Pustaka

Parameter Penulis	Objek Penelitian	Metode/ Teknologi	Database	Visualisas
Wikjatmiko , Zhulfi Bajra (2014)	<i>Bahasa Isyarat</i>	Phonegap	Sqlite	Data
Kresno, F. X. Dwi (2017)	Dosen Di Stmik Akakom	Phonegap	Mysql	Data
Gunawan, Arif (2016)	Keuangan Pribadi	Cordova Ionic Angular Js	Sqlite	Data
Putri Dwi Intan Ra'uufi & Yanuar Sinatra (2016)	Ageng Bakery Group	Java, Apache Cordova, Android Studio	Mysql	Google maps api dan Peta
Andre (2015)	Kepramukaan	JSON (Java Script Object Notation)	Mysql	Data
Anggita Dwi Setiya Putra	Data Lokasi Pelancong	Apache Cordova, Json.	MongoDB	Usulan : Membuat Aplikasi Yang Dapat Menyimpan Data Lokasi

				Dengan Memanfaatkan Apache Cordova
--	--	--	--	------------------------------------

2.2 DASAR TEORI

2.2.1 APACHE CORDOVA



Gambar 2.1 Architecture Apache Cordova.

(<https://cordova.apache.org/docs/en/latest/guide/overview/>)

Apache Cordova merupakan sebuah *framework* yang *open source* terdiri dari kumpulan perangkat API yang memungkinkan pengembang atau programmer mengakses fungsi *native* dari sebuah perangkat serta membuat aplikasi pada berbagai macam *platform* seperti Android, Blackberry, Iphone atau Windows Phone. (Putri Dwi Intan Ra'uufi & Yanuar Sinatra. 2016)

Plugin di dalam Apache Cordova adalah bagian integral dari ekosistem cordova. Mereka menyediakan antarmuka untuk Cordova dan komponen asli untuk berkomunikasi satu sama lain dan *bindings* ke API perangkat standar. Ini

memungkinkan Anda untuk mengaktifkan *script* asli dari JavaScript. Untuk menambahkan plugin ke dalam proyek dengan mengetik `$ cordova plugin add cordova-plugin-camera`.

Proyek Apache Cordova mempertahankan satu set plugin yang disebut *Core Plugins*. Plugin inti ini menyediakan aplikasi Anda untuk mengakses kemampuan perangkat seperti baterai, kamera, kontak, dll.

Penginstalan Apache Cordova dengan mengetik `C: \> npm install -g cordova`. Pada baris *script* tersebut `-g` memberitahu npm untuk menginstal cordova secara global. Kalau tidak, itu akan dipasang di subdirektori `node_modules` dari direktori kerja saat ini.

Penambahan platform pada Apache Cordova dengan mengetik `$cordova platform add android`. Baris *script* tersebut bertujuan menambahkan *platform* Android kedalam proyek. Untuk melihat *platform* yang telah ada di dalam proyek dengan mengetik `$ cordova platform ls`.

Apache Cordova Secara default membuat *script* yang menghasilkan aplikasi berbasis web yang halaman awalnya adalah file `www / index.html` yang berada di dalam proyek. Setiap inisialisasi harus ditetapkan sebagai bagian dari pengendali *event deviceready* yang didefinisikan dalam `www / js / index.js`. Untuk membangun proyek di semua jenis *platform* aplikasi dengan menjalankan `$ cordova build`.

2.2.2 CORDOVA PLUGIN CAMERA

Plugin *camera* ini mendefinisikan objek global `navigator.camera` yang menyediakan API untuk mengambil *image* dan untuk memilih *image* dari pustaka gambar sistem. Meskipun objek tersebut melekat pada navigator *global scoped*, itu tidak tersedia sampai *deviceready*. Penambahan plugin ini membutuhkan versi Apache Cordova 5.0+ untuk menambahkannya menggunakan perintah `$cordova plugin add cordova-plugin-camera`. Penggunaan pada plugin ini dengan menambahkan baris *script* berikut ke dalam proyek `camera.getPicture (successCallback, errorCallback, options)`.

Fungsi `camera.getPicture` membuka aplikasi *camera* default perangkat yang memungkinkan pengguna untuk mengambil *image* secara default - perilaku ini terjadi, ketika `Camera.sourceType` di samakan dengan *script* program `Camera.PictureSourceType.CAMERA`. Setelah pengguna mengunci *image*, aplikasi *image* akan menutup dan aplikasi akan dipulihkan.

Jika `Camera.sourceType` menggunakan `Camera.PictureSourceType.PHOTOLIBRARY` atau menggunakan yang `Camera.PictureSourceType.SAVEDPHOTOALBUM`, maka tampilan dialog yang memungkinkan pengguna untuk memilih *image* yang ada. Nilai kembalian dikirim ke fungsi callback `cameraSuccess`, dalam salah satu format berikut, tergantung pada *camera option*: *String* yang berisi *image* dengan *encode Base64* dan *String* yang mewakili lokasi *file image*. (<https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-camera>)

2.2.3 CORDOVA PLUGIN GEOLOCATION

Plugin ini memberikan informasi tentang lokasi perangkat, seperti garis lintang dan garis bujur. Sumber umum informasi lokasi termasuk Global Positioning System (GPS) dan lokasi yang disimpulkan dari sinyal jaringan seperti alamat IP, RFID, WiFi dan alamat MAC Bluetooth, dan ID sel GSM / CDMA. Tidak ada jaminan bahwa API mengembalikan lokasi aktual perangkat. API ini didasarkan pada Spesifikasi Geolokasi API W3C, dan hanya dijalankan pada perangkat yang belum memberikan implementasi.

Penambahan plugin geolocation ini membutuhkan Apache Cordova versi 5.0+ untuk menambahkannya menggunakan perintah `cordova plugin add cordova-plugin-geolocation`. Plugin Geolocation ini dapat dipakai dalam aplikasi Android, iOS dan Windows.

Plugin geolocation menggunakan koordinat berdasarkan lokasi yang tersedia dengan fungsi *call back* untuk permintaan posisi saat ini *device* berada. Propertis yang ada dalam kordinat ini adalah latitude yang menentukan titik garis lintang derajat desimal dan longitude titik garis bujur dalam derajat desimal sehingga posisi dapat akurat sesuai lokasi *device* berada saat memasukkan data. (<https://cordova.apache.org/docs/en/latest/referene/cordova-plugin-geolocation>)

2.2.4 IONIC

Ionic adalah sekumpulan layanan dan fitur canggih yang dibangun di atas *framework* unggulan yang membawa tingkat kelincahan pengembangan aplikasi yang benar-benar baru ke tim pengembang *mobile*. Ionic memiliki beberapa

layanan inti yang membantu Anda melalui siklus hidup aplikasi yang lengkap antara lain *Deploy* yaitu perbarui aplikasi Anda dari jarak jauh secara *real time* tanpa penundaan penyimpanan aplikasi, *View* yaitu Uji aplikasi Anda dengan penguji dan dapatkan masukan tanpa perangkat rumit dan penyediaan manajemen profil, *Package* yaitu membangun aplikasi di awan untuk Android dan iOS, yang terakhir *Monitoring* yaitu melacak *error runtime* dalam pembangunan kembali TypeScript asli Anda.

Penginstalan Ionic dapat dijalankan dengan menggunakan CLI terbaru dari npm, untuk melakukan penginstalan dengan menulis `npm install -g ionic` pada CLI node js. Untuk pembuatan aplikasi Ionic dengan menggunakan perintah `$ ionic start nama_aplikasi tutorial`, baris *script* tersebut bagian *start* akan memberi tahu CLI untuk membuat aplikasi baru setelah itu nama_aplikasi akan menjadi nama direktori dan nama aplikasi dari proyek Anda dan tutorial akan menjadi template awal untuk proyek Anda.

Pengujian aplikasi Ionic dengan melakukan perintah `$ ionic serve` untuk menguji menggunakan browser dan hasilnya adalah tampilan dari proyek yang sedang dibuat. Untuk *Deploy* ke apk menggunakan perintah `ionic cordova run android` dengan syarat mengaktifkan mode *debugging*, mode pengembang pada *device* yang digunakan serta telah menginstal android sdk untuk *build* ke dalam aplikasi android menggunakan versi terbaru. (<https://ionicframework.com/docs/>)

2.2.5 ANGULAR

Angular CLI adalah alat antarmuka baris perintah yang dapat membuat proyek, menambahkan *file*, dan melakukan berbagai tugas pengembangan yang sedang berlangsung seperti pengujian, penggabungan, dan pembuatan. Tujuan dalam panduan ini adalah untuk membangun dan menjalankan aplikasi Angular sederhana dalam TypeScript, menggunakan CLI saat mengikuti rekomendasi Panduan yang bermanfaat bagi setiap proyek.

Pengembangan menggunakan Angular perintah pertama yang dijalankan adalah penginsalan Angular kedalam Node.js dengan menggunakan perintah `npm install -g @angular/cli` setelah perintah tersebut dijalankan dan berhasil maka sudah dapat melakukan pembuatan proyek menggunakan Angular.

Pemanggilan data dalam Angular menggunakan HttpClient di `@angular/common/http` menawarkan API HTTP klien yang disederhanakan untuk aplikasi Angular yang bertumpu pada antarmuka XMLHttpRequest yang diekspos oleh peramban. Manfaat tambahan dari HttpClient termasuk fitur *testability*, objek permintaan dan respon yang diketik, intersepsi permintaan dan respon, apis yang dapat diamati, dan penanganan kesalahan yang efisien. Untuk dapat menggunakan HttpClient perlu di tambahkan perintah `import {HttpClientModule} from '@angular/common/http';` pada bagian `app/modul.ts` dan deklarasikan dengan perintah `HttpClientModule` di dalam `imports:[]`.

Getting JSON data di dalam Angular seperti Aplikasi sekarang sering meminta data JSON dari server. Misalnya, aplikasi mungkin memerlukan file

konfigurasi di server, `config.json`, yang menentukan URL sumber daya. Contoh program `asset/config.json` { "heroesUrl": "api/heroes", "textfile": "assets/textfile.txt"} dari baris *script* tersebut `ConfigService` memanggil file ini dengan metode `get ()` di `HttpClient`.

Service data Didalam `app/config/config.service.ts` ada *script* program `configUrl = 'assets/config.json'; getConfig() { return this.http.get(this.configUrl);}` dimana Komponen seperti `ConfigComponent`, memasukkan `ConfigService` dan memanggil metode layanan `getConfig`.

```
showConfig() {
  this.configService.getConfig()
    .subscribe((data:Config)=>
      this.config={heroesUrl:data['heroesUrl']}
    );
}
```

Gambar 2.2. Pemanggilan Config Dari Component.ts

Dalam gambar 2.2 baris *script* tersebut merupakan metode layanan mengembalikan data konfigurasi yang dapat diamati, komponen tersebut akan berlangganan ke nilai pengembalian metode. *Call back* berlangganan menyalin bidang data ke dalam objek konfigurasi komponen, yang terikat pada data dalam template komponen untuk ditampilkan. (<https://angular.io/>)

2.2.6 MONGODB(NOSQL)

RDBMS	MONGODB
Table/view	Collection
Row	Document
Index	Index
Join	Embedded Document
Foreign Key	Referenced Document
Partition	Shard

Gambar 2.3 Perbedaan MySQL Dan MongoDB.(Agung Julismam. 2015)

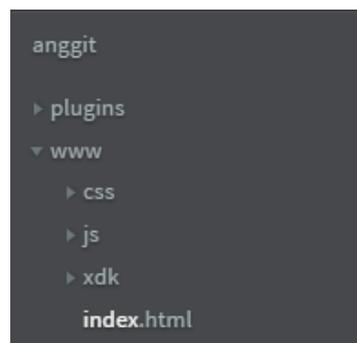
MongoDB adalah sebuah *document oriented database* yang bersifat *open source*. MongoDB merupakan salah satu database NoSQL yang memiliki sebuah konsep penyimpanan data *non-relational*. Istilah NoSQL merupakan kepanjangan dari “Not Only SQL” yaitu sistem manajemen database yang berbeda dari sistem manajemen database rasional dalam beberapa cara. Penyimpanan data tanpa perlu adanya tabel *schema* dan tidak ada bahasa sql yang terlibat dalam pemakaian database.

Dalam MongoDB tidak mengenal adanya tabel, kolom dan baris jadi tidak ada *schema* dalam MongoDB (*schema-less*). Unit paling kecil pada dari MongoDB adalah *document*, sedangkan kumpulan dari *document* adalah *collection*. Seperti halnya dalam database rasional *document*, ibarat sebuah *record* dan *collection* pada sebuah *document* dalam MongoDB dapat memiliki atribut yang berbeda-beda dengan *document* yang lainnya walaupun pada satu *collection*.

MongoDB dibuat dengan menggunakan C++, yang memiliki beberapa fitur yaitu Document-oriented storage, full index, support replication dan high availability, auto-sharding, querying, fast in-place updates, map reduce,

gridFS,commercial support. Pada MongoDB tidak membutuhkan struktur tabel jadi untuk melakukan proses *insert* maka MongoDB akan otomatis membuat struktur tabelnya. MongoDB mempunyai performa 4x lebih cepat jika dibandingkan dengan Mysql. MongoDB dikembangkan sejak Oktober 2007 oleh 10Gen dan dirilis ke public sejak Februari 2009 yang mempunyai lisensi GNU AGPL 3.0 dan Apache License untuk drivernya. (Badawi, Alwigani Rachman. 2017)

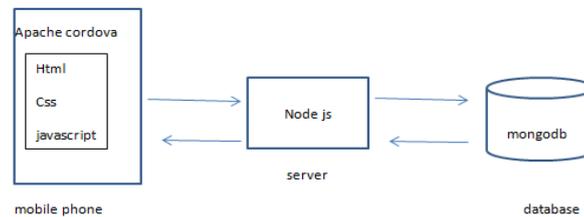
2.2.7 INTEL XDK



Gambar 2.4 Susunan Folder Intel XDK

Intel XDK adalah development kit yang dibuat oleh Intel untuk membuat aplikasi *native* untuk perangkat *mobile* menggunakan teknologi web seperti HTML5, CSS dan Javascript. Aplikasi web dikompilasi menggunakan *platform* Cordova di server online untuk membuat aplikasi *hybrid* yang *cross-platform*. Aplikasi Intel XDK sendiri adalah aplikasi *cross-platform* yang tersedia untuk sistem operasi Windows, Linux dan OSX. Dengan teknologi seperti ini, developer dapat membuat suatu aplikasi yang dapat berjalan di banyak *platform mobile* hanya dengan menggunakan satu baris *script* saja. (Selviani, Mustika. 2017)

2.2.8 NODE.JS



Gambar 2.5 Cara Kerja 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. (<https://nodejs.org/en/>)

2.2.9 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 perograman C. (<http://www.json.org/json-id.html>)

2.2.10 BSON (BINARY JSON)

BSON kependekan dari Binary JSON, adalah serialization biner dokumen JSON-encoded. Seperti JSON, BSON mendukung penyisipan *document* dan *array*

dalam *document* dan *array* lainnya. BSON juga berisi ekstensi yang memungkinkan *representation* tipe data yang bukan merupakan bagian dari spesifikasi JSON.

BSON dapat dibandingkan dengan format pertukaran biner, seperti Protocol Buffers. BSON lebih "schema-less" daripada Protocol Buffers, yang dapat memberikan keuntungan dalam fleksibilitas namun juga sedikit merugikan dalam efisiensi ruang. (<http://bsonspec.org/>)