

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian yang dilakukan oleh Ghifari Yudha Pratama (2017) dari Program Studi Teknik Informatika, STMIK Akakom Yogyakarta. Penelitian ini mengenai konsep Ionic untuk membuat sebuah antarmuka yang mencakup *User Interface* (UI) dan *User Experience* (UX) dengan memanfaatkan elemen *color, button, cards, list, tabs*, dan *animation*.

Penelitian yang dilakukan oleh Jayanudin (2016) dari Program Studi Teknik Informatika, Universitas Pasundan Bandung. Penelitian ini mengenai pemanfaatan Ionic yang disandingkan dengan RESTful *web service* untuk membantu membuat aplikasi *Reminder* berbasis *hybrid*.

Penelitian oleh Ramly Ramadhan (2015) dari Program Studi Pendidikan Teknik Informatika Fakultas Teknik, Universitas Negeri Yogyakarta. Penelitian ini mengkaji mengenai pengembangan aplikasi Mobile School MAPS untuk pemetaan kampus. Dalam penelitian ini dibahas mengenai pengembangan dan analisis kualitas aplikasinya.

Penelitian oleh Yusuf Agung Pramono (2015) dari Program Studi Informatika Fakultas Komunikasi Dan Informatika, Universitas Muhammadiyah Surakarta. Penelitian ini mengkaji mengenai bagaimana merancang dan membuat aplikasi pemesanan produk Distro Denim pada smartphone yang berjalan dengan sistem operasi Android dengan tujuan menunjang kegiatan bisnis dan menjangkau

pelanggan secara lebih luas. Penelitian ini dikhususkan membahas tentang aplikasi mobile-nya bukan pada *website*.

Penelitian oleh Ronny (2015) dari Program Studi Teknik Informatika, Fakultas Teknologi Informasi. Penelitian ini mengkaji tentang bagaimana membangun sebuah situs web toko online dengan sistem lelang di dalamnya. Aplikasi tersebut dibangun menggunakan PHP sebagai bahasa pemrograman dan MySQL sebagai basis datanya.

Penelitian oleh Hamidatun Nikma (2014) dari Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga. Penelitian ini mengkaji tentang bagaimana membangun sistem informasi rekam medis berbasis mobile web yang dapat membantu menyelesaikan pekerjaan pegawai. Aplikasi yang dimaksudkan pada penelitian tersebut adalah sebuah web responsif Rekam Medis yang nyaman saat dibuka di perangkat desktop maupun mobile.

Tabel 2.1 Rangkuman Penelitian

No	Penulis, Tahun	Objek	Teknologi	Platform
1	Ghifari Yudha Pratama, 2017	Kedai Biasa Coffee	Ionic Framework	Mobile
2	Jayanudin, 2016	<i>Reminder</i>	Ionic Framework, RESTful <i>web service</i>	Mobile
3	Ramly Ramadhan, 2015	Lingkungan Uniiversitas Negeri Yogyakarta	Ionic Framework	Mobile
4	Yusuf Agung Pramono, 2015	Data Pakaian Distro	Android	Mobile

No	Penulis, Tahun	Objek	Teknologi	Platform
5	Ronny, 2015	PT.Growing Smart Center	PHP & MySQL	Web
6	Hamidatun Nikma, 2014	Sistem Informasi Rekam Medis	Web Mobile	Web

Dalam tinjauan pustaka yang telah dilakukan terdapat penelitian yang sudah mengimplementasikan teknologi *hybrid* dengan objek dan teknologi yang berbeda beda. Oleh karena itu sistem yang akan dibangun penulis adalah sistem multiplatform yaitu web dan aplikasi mobile. Dalam penelitian ini penulis akan mengimplementasikan Ionic untuk membangun aplikasi toko online berbasis mobile dan web.

2.2 Dasar Teori

2.2.1 Aljuna Archery

Aljuna Archery adalah usaha yang bergerak di bidang penjualan alat panahan yang didirikan oleh Danang Dwi Asmoro dan beralamatkan di Jl. Imogiri Barat No. 31, Ponggok I, Trimulyo, Jetis, Bantul, Daerah Istimewa Yogyakarta. Sebelum usaha ini didirikan Danang sering sekali dititipi oleh rekan sesama atlet Panahan untuk membelikan berbagai perlengkapan Panahan karena saat itu Danang dikenal sangat mengerti tentang impor alat-alat Panahan dengan kualitas bagus. Melihat peluang tersebut pada bulan Oktober tahun 2015 Danang berinisiatif sendiri untuk menciptakan Aljuna Archery. Dengan bermodalkan pengalaman sebagai sales dan uang dari pinjaman bank Danang mulai menyediakan stok barang dan menawarkannya ke atlet-atlet Panahan.

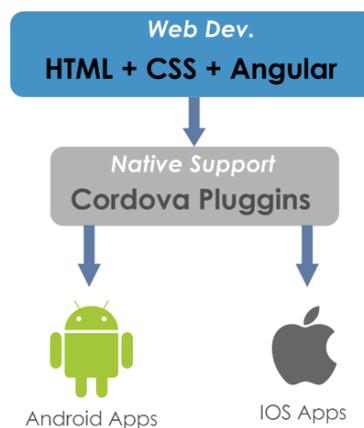
Akhir tahun 2015 semakin banyak saja kustomer yang datang hingga Danang harus dibantu oleh adiknya yaitu Bayu Tri Pamungkas. Hingga saat ini Aljuna Archery memiliki ratusan produk dan dengan perputaran uang perbulannya mencapai ratusan juta. Bukan hanya berjualan melalui toko secara offline tetapi Aljuna Archery juga memasarkan barangnya melalui Instagram dan menerima pengiriman di seluruh wilayah Indonesia. Saat ini Danang dibantu oleh adiknya dan juga 3 karyawan yang rencananya akan terus berkembang demi kemajuan Aljuna Archery.

2.2.2 Aplikasi Mobile

Menurut Pressman dan Bruce (2014:9), aplikasi mobile adalah aplikasi yang telah dirancang khusus untuk platform mobile (misalnya iOS, Android, atau windows mobile). Dalam banyak kasus, aplikasi mobile memiliki user interface dengan mekanisme interaksi unik yang disediakan oleh platform mobile, interoperabilitas dengan sumber daya berbasis web yang menyediakan akses ke beragam informasi yang relevan dengan aplikasi, dan kemampuan pemrosesan lokal untuk pengumpulan, analisis, dan format informasi dengan cara yang paling cocok untuk platform mobile. Selain itu aplikasi mobile menyediakan kemampuan penyimpanan persisten dalam platform.

2.2.3 Ionic Framework

Framework Ionic yang dikembangkan oleh Drifty Co. pada tahun 2012 merupakan sebuah framework yang mendukung pengembang web untuk mengembangkan aplikasi multiplatform atau juga dikenal sebagai aplikasi *hybrid*. Framework Ionic pada dasarnya adalah aplikasi web namun dapat dirilis ke dalam platform Android maupun iOS dengan dukungan Cordova di dalamnya. Dengan menggunakan Ionic pengembang web yang hanya memiliki kemampuan dalam bahasa pemrograman HTML5, CSS, JavaScript, MySQL sudah bisa membuat aplikasi mobile. Ionic disediakan antarmuka dan komponen yang dikhususkan untuk membuat aplikasi mobile sehingga sangat cocok digunakan untuk membangun sebuah aplikasi mobile. Walaupun pada dasarnya aplikasi yang dibuat berbasis web, namun karena Ionic berjalan di atas teknologi Angular dan disediakan juga komponen yang dikhususkan untuk membuat aplikasi mobile maka aplikasi yang dibuat memiliki performa dan antarmuka hampir seperti aplikasi *native*.



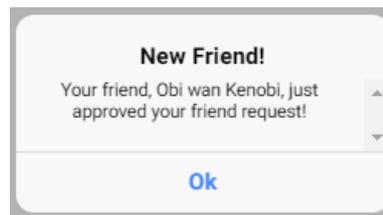
Gambar 2.1 Arsitektur Ionic

Gambar 2.1 adalah arsitektur dari Ionic, di dalamnya terdapat Apache Cordova yang berperan mengkonversi sebuah aplikasi web menjadi aplikasi *hybrid*.

1. Komponen-komponen Ionic

- Alert

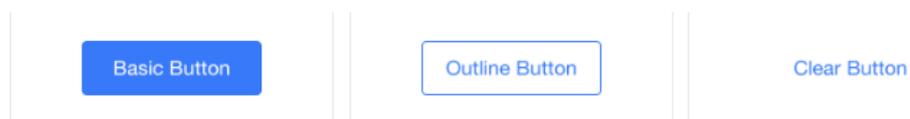
Alert adalah sebuah pesan "mengambang" dimana pesan tersebut berisi informasi penting kepada pengguna ataupun sebuah pilihan untuk melakukan tindakan tertentu. Contoh penggunaan Alert adalah untuk menampilkan peringatan apakah pengguna ingin menghapus sebuah data atau tidak.



Gambar 2.2 Alert

- Button

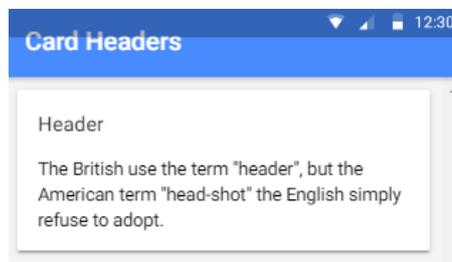
Button merupakan salah satu cara untuk berinteraksi antara pengguna dengan aplikasi. Pengguna dapat berinteraksi ataupun menavigasi aplikasi dengan cara menyentuhnya maka perintah akan dijalankan sesuai apa yang ditanamkan pada button tersebut.



Gambar 2.3 Button

- Card

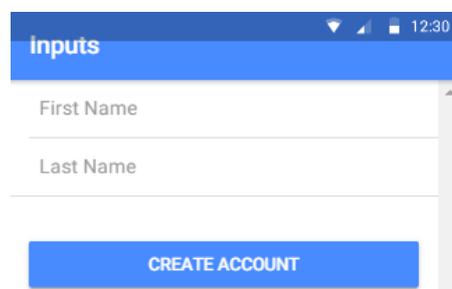
Cards adalah sebuah pola desain inti untuk sebuah konten. Penggunaan cards adalah cara yang tepat untuk menampung dan mengatur informasi. Cards memungkinkan lebih banyak kontrol, fleksibel, dan bahkan bisa dianimasikan sehingga konten yang ditampilkan akan nyaman dipandang. Cards telah menjadi pola desain pilihan bagi banyak perusahaan seperti Google, Twitter, dan Spotify.



Gambar 2.4 Card

- Inputs

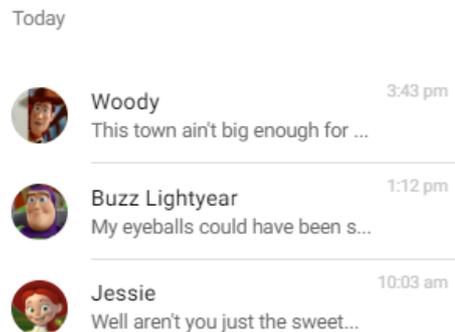
Inputs berfungsi untuk mengumpulkan dan menangani masukan pengguna secara aman. Komponen inputs biasanya digunakan untuk membuat formulir isian untuk pengguna.



Gambar 2.5 Input

- Lists

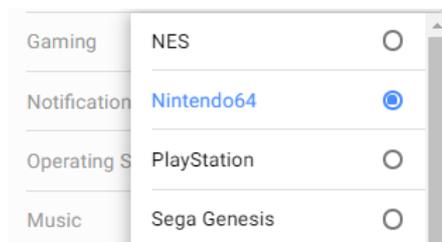
Lists digunakan untuk menampilkan baris informasi atau item-item yang di dalamnya dapat berisi, *string*, ikon, ataupun *thumbnail* yang juga dapat diberi *link* untuk menuju ke halaman tertentu.



Gambar 2.6 Lists

- Select

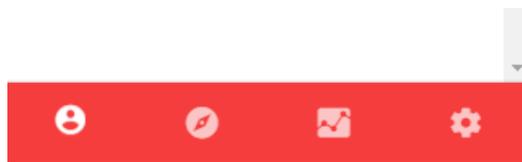
Komponen select pada ionic serupa dengan elemen HTML `<select>`. Namun, komponen select pada ionic sangat berguna untuk menyortir dan memilih opsi yang *diinginkan*. Saat pengguna menekan komponen select, sebuah dialog akan muncul dengan semua pilihan dalam daftar yang besar dan mudah dipilih.



Gambar 2.7 Select

- **Tabs**

Tabs adalah area horizontal yang berisi komponen button atau link untuk mempermudah navigasi pada aplikasi mobile. Tabs biasanya berisi button dengan konten-konten yang diutamakan pada aplikasi mobile. Tabs dapat berisi kombinasi teks dan atau ikon, serta merupakan metode navigasi populer untuk navigasi mobile.



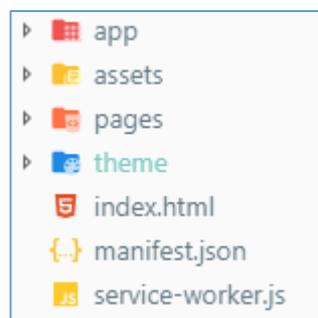
Gambar 2.8 Tabs

2. **Instalasi Ionic**

Sebelum melakukan instalasi Ionic pastikan untuk instal NodeJs terlebih dahulu karena di dalam NodeJs terdapat npm yang berguna untuk instalasi Ionic dan Cordova. Instal juga Android Studio karena dibutuhkan juga Android SDK yang sudah disematkan didalamnya yang berguna sebagai syarat proses *building* aplikasi Android. Setelah kedua *tools* tadi terinstal selanjutnya mulai proses instalasi Ionic dan Cordova dengan mengetikkan perintah “npm install -g cordova ionic” ke dalam terminal lalu tekan Enter, maka instalasi akan diproses.

3. Membuat aplikasi Ionic

Untuk memulai membuat aplikasi Ionic cukup dengan memberikan perintah “ionic start myApp blank” ke dalam terminal. Maksud dari perintah tersebut adalah memulai membuat aplikasi kosong dengan nama “myApp”. Setelah perintah dijalankan maka folder myApp akan otomatis dibuat dengan struktur folder seperti pada Gambar.



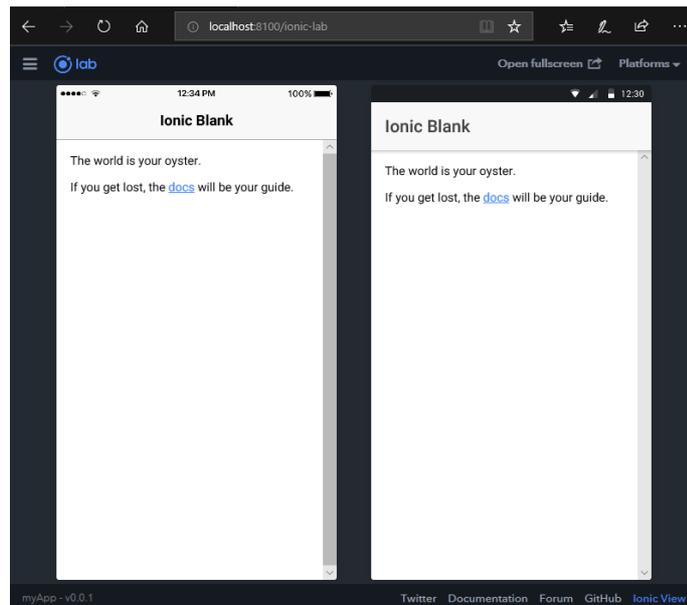
Gambar 2.9 Struktur Folder dan *File* Ionic

Gambar 2.9 menunjukkan struktur folder dan *file* dari sebuah project Ionic. Berikut fungsi-fungsinya.

- Folder app berisi pengaturan aplikasi dan pendefinisian *library* Ionic.
- Folder assets berisi *file* aset yang sewaktu waktu dapat dipanggil.
- Folder page berisi halaman-halaman aplikasi.
- Folder theme berisi pengaturan CSS pada Ionic.
- index.html adalah index dari aplikasi.
- manifest.json berisi pengaturan identitas aplikasi termasuk *icon* aplikasi.
- service-worker.js merupakan controller utama dari aplikasi agar tiap halaman dapat dipanggil dalam index.html.

4. Menjalankan aplikasi melalui Android

Untuk melakukan pengembangan aplikasi menggunakan Ionic pengguna cukup menggunakan *text editor*, namun akan sulit untuk mengidentifikasi kesalahan dan sejauh mana aplikasi tersebut berjalan jika tanpa melihat pratinjau aplikasinya. Maka dari itu Ionic menyediakan fitur untuk melihat pratinjau aplikasi melalui *browser*. Untuk menjalankan pratinjau aplikasi melalui *browser* dapat dengan memberikan perintah “ionic serve” ke dalam terminal, kemudian membuka *url* “localhost:8100/ionic-lab” ke dalam browser.



Gambar 2.10 Contoh Pratinjau Aplikasi

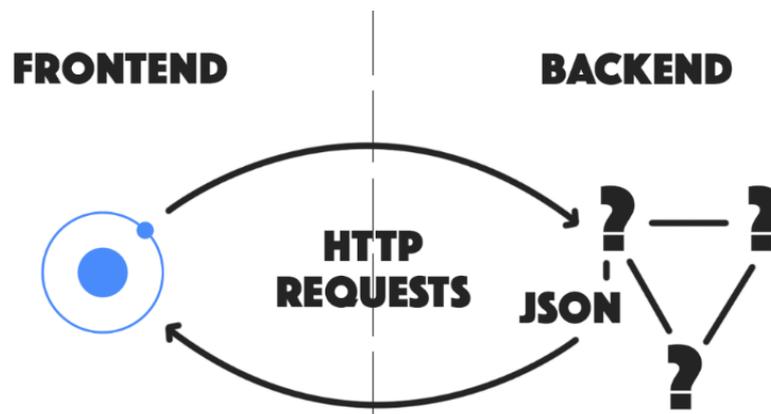
Gambar 2.10 menunjukkan contoh pratinjau aplikasi yang dibuat menggunakan Ionic. Terdapat dua buah antarmuka yaitu untuk platform iOS dan platform Android.

5. Build ke aplikasi Android

Ionic adalah framework yang segala perintahnya menggunakan CLI (*Command-Line Interface*). Jadi untuk menjadikan *project* Ionic menjadi aplikasi Android cukup dengan memberikan perintah “ionic cordova build android” maka file berekstensi “.apk” akan dibuat dan disimpan di direktori `myApp/platforms/android/app/build/outputs/apk/debug`.

6. Mengintegrasikan Backend dengan Ionic

Ionic pada dasarnya tidak terintegrasi langsung dengan *backend*, tetapi hanya berkomunikasi saja karena antara *frontend* dan *backend* berjalan secara independen. Jadi jika menggunakan *backend* PHP maka kode PHP tidak ditulis langsung dalam Ionic. Untuk mengintegrasikan keduanya perlu mengimplementasikan API.



Gambar 2.11 Integrasi Antara *Frontend* dan *Backend* pada Ionic

Seperti yang ditunjukkan pada Gambar 2.11 tidak ada integrasi langsung antara *frontend* dan *backend*. Terlihat keduanya saling berkomunikasi dengan bantuan HTTP Request dan JSON. Untuk mengintegrasikan keduanya perlu

menerapkan REST API untuk mengirim dan menerima data berupa JSON.

Berikut langkah untuk mengintegrasikan PHP (*backend*) dengan Ionic.

- Membuat file API menggunakan PHP yang menghasilkan output berupa data JSON.

```
$conn = mysqli_connect('localhost', 'username', 'password', 'db');
mysqli_set_charset($conn, 'utf8');
$query = mysqli_query($conn, 'SELECT * FROM barang');
while($row = mysqli_fetch_assoc($query)) {
    $data[] = $row;
}

$json = json_encode($data);
echo $json;
```

Gambar 2.12 Membuat API Untuk Mengambil Data Barang

Kode program berfungsi untuk mengambil data barang dari *database* dan memberikan output berupa Data JSON. Pada Gambar di atas terdapat variabel `$conn` yang merupakan perintah untuk mengkoneksikan ke dalam database db. Variabel `$query` berisi perintah untuk menyeleksi tabel barang. Dari kedua variabel tadi dieksekusi sehingga menghasilkan data yang kemudian dimasukkan ke dalam *array* `$data[]`. Selanjutnya data *array* dalam variabel `$data[]` di-*encode* ke data JSON menggunakan perintah `json_encode` sehingga hasil dari potongan kode program tersebut berupa data JSON.

- Melakukan HTTP Request pada Ionic ke URL PHP yang berisi data JSON.

```
import { Http, Headers } from '@angular/http';
import 'rxjs/add/operator/map';
import { Component } from '@angular/core';
import { NavController, ViewController } from 'ionic-angular';

@Component({
  selector: 'page-home', // merupakan halaman home dari aplikasi
  templateUrl: 'home.html' // home.html adalah antarmuka dari
})
export class HomePage {

  public items : any = [];
  constructor(public navCtrl: NavController, public http : Http,
              public navCtrl : ViewController) {
  }
  getData(){
    this.http.get('http://localhost/barang/barang.php')
      .map(res => res.json())
      .subscribe(data =>{
        this.items = data;
      })
  }
  ionViewWillEnter()
  {
    this.getData();
  }
}
```

Gambar 2.13 Mengambil Data dari API Barang

Kode program di atas merupakan contoh dari file bertipe TypeScript pada suatu halaman di proyek Ionic pada contoh ini adalah `home.ts`, berikut adalah penjelasannya.

- Kode program ini merupakan *controller* dari halaman *home* yang ditunjukkan pada variabel *selector* yaitu `page-home`.
- `home.html` merupakan *template* dari halaman *home*.
- Variabel `items` merupakan variabel bertipe *array*.
- Membuat *method* `getData()` untuk mengambil data JSON dari sebuah API yang kemudian akan disimpan pada variabel `items` yang sebelumnya sudah didefinisikan.

- Di dalam `ionViewWillEnter()` terdapat perintah untuk memanggil metode `getData()` yang sebelumnya didefinisikan. `ionViewWillEnter()` sendiri merupakan sebuah metode milik Ionic yang memiliki fungsi akan dijalankan bila halaman yang bersangkutan dibuka atau ada aksi dalam halaman yang tersebut.
- Menampilkan data yang telah dipanggil sebelumnya.

```

<ion-header>
  <ion-navbar>
    <ion-title>Home</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <ion-list>
    <ion-item ng-repeat="data in ctrl.data" *ngFor="let item of
items">
      <h2>{{ item.kode_barang }}</h2>

    </ion-item>
  </ion-list>
</ion-content>

```

Gambar 2.14 Menampilkan Data Barang

Kode program di atas adalah *file* `home.html` yang merupakan template dari halaman *home*. Pada kode `*ngFor="let item of items"` variabel `items` diinisialisasikan menjadi *item* kemudian dilakukan perulangan agar semua data dapat ditampilkan. Untuk pemanggilan datanya dapat dengan perintah `item.nama-kolom`.

2.2.4 Angular

Angular adalah sebuah platform untuk mengembangkan *frontend* dari sebuah aplikasi web yang berbasis *open source*. Platform ini dikembangkan oleh tim Google dan komunitas *open source* Angular. Platform ini

menggunakan bahasa Typescript, sebuah pengembangan dari bahasa Javascript yang merupakan bahasa utama yang digunakan untuk pengembangan web. Angular sendiri merupakan platform yang dibangun untuk meningkatkan performa dari AngularJS. Platform ini sering disebut juga dengan Angular 2 atau Angular 2+.

Platform ini seringkali digunakan untuk membangun *Single Page Application* (SPA). Penggunaan dari SPA ini berarti bahwa segala komponen yang akan digunakan dalam aplikasi web tersebut akan di-*load* dalam browser terlebih dahulu agar user tidak perlu melakukan *page load* ketika membuka page baru.

2.2.5 Apache Cordova

Apache Cordova adalah kerangka pengembangan mobile open source yang memungkinkan untuk menggunakan teknologi web standar - HTML5, CSS3, dan JavaScript untuk pengembangan aplikasi *hybrid*. Aplikasi yang dibungkus dengan cordova ditargetkan agar dapat dijalankan masing-masing platform yang diinginkan, dan bergantung pada binding API yang sesuai standar untuk mengakses kemampuan masing-masing perangkat seperti sensor, data, status jaringan, dll. Pada ionic sendiri cordova lah yang digunakan untuk membungkus projek ionic untuk dijadikan aplikasi hybrid.