

## IMPLEMETASI TEKNIK HYBRID MOBILE APPLICATION DALAM PEMBUATAN APLIKASI MOBILE MARKETPLACE IKAN HIAS

M. Nuraminudin

Manajemen Informatika Universitas AMIKOM Yogyakarta

email : [adin.5829@amikom.ac.id](mailto:adin.5829@amikom.ac.id)<sup>1)</sup>,

### Abstraksi

Perkembangan jumlah pengguna *smartphone* di dunia, khususnya di Indonesia saat ini sangatlah pesat. Senada dengan perkembangan pengguna *smartphone*, penggemar ikan hias di Indonesia pun terus berkembang pesat, dibuktikan dengan banyaknya event ikan hias yang sering diadakan di seluruh Indonesia. Produk ikan hias yang dimiliki pengusaha di Indonesia harus dapat diakses melalui *smartphone* agar mampu mendapatkan jumlah konsumen yang besar. Salah satu cara agar konsumen dapat mengakses produk melalui *smartphone* adalah dengan melalui aplikasi yang dapat dipasang pada masing-masing *smartphone* konsumen. Saat ini terdapat dua sistem operasi *smartphone* terpopuler, yaitu Android dan IOS. Agar jumlah pengguna aplikasi maksimal, maka aplikasi harus dapat dijalankan di kedua sistem operasi tersebut. Penelitian ini akan mengimplementasikan teknik *hybrid mobile application* untuk pembuatan aplikasi *marketplace* penjualan ikan hias sehingga aplikasi dapat dibangun agar dapat dijalankan di Android serta IOS tanpa mengubah banyak *source code* di dalamnya.

### Kata Kunci :

Hybrid, Marketplace, Ionic

### Abstract

The number of smartphone users in the world, especially in Indonesia is currently growing very rapidly. In line with the development of smartphone users, ornamental fish enthusiasts in Indonesia continue to grow rapidly, as evidenced by the many ornamental fish events that are often held throughout Indonesia. Ornamental fish products owned by entrepreneurs in Indonesia must be accessible via smartphones in order to be able to get a large number of consumers. One way for consumers to access products through smartphones is through applications that can be installed on each consumer smartphone. Currently there are two of the most popular smartphone operating systems, namely Android and IOS. For the maximum number of application users, the application must be able to run on both operating systems. This research will implement a hybrid mobile application technique for making ornamental fish market marketplace applications so that the application can be built so that it can run on Android and IOS without changing many source code in it.

### Keywords :

Hybrid, Marketplace, Ionic

### Pendahuluan

Hasil survei dan prediksi yang dilakukan oleh Statista menunjukkan bahwa jumlah pengguna *smartphone* di Indonesia akan terus meningkat. Menurut Statista, terdapat 33% penduduk Indonesia yang diproyeksikan menggunakan *smartphone* pada tahun 2023. Berdasarkan data tersebut, dapat diketahui bahwa perkembangan penggunaan *smartphone* akan semakin pesat dan bisnis yang berbasis aplikasi *mobile* akan semakin banyak diminati. Terdapat 3 sistem operasi populer pada *smartphone* yang saat ini beredar di tengah masyarakat. Ketiga sistem operasi tersebut adalah Android, IOS, dan Windows. Masing-masing sistem operasi ini didukung oleh 3 perusahaan besar dunia di bidang IT, yaitu Android didukung oleh Google, IOS didukung oleh Apple, dan Windows didukung oleh Microsoft.

*Smartphone* dengan sistem operasi Android jauh mendominasi dalam hal jumlah pengguna, yaitu sebesar 82,8% dari total pengguna *smartphone*, dibandingkan dengan jumlah pengguna IOS sebesar 27,9%, dan Windows yang hanya 1,3% dari total pengguna *smartphone* [1]. Selain itu, berdasarkan pengumuman resmi dari Microsoft, dukungan terhadap Windows Phone, *smartphone* dengan sistem operasi Windows, akan dihentikan pada tanggal 10 desember 2019 dan akan dihentikan secara total pada tanggal 10 maret 2020. Dikarenakan hal tersebut, aplikasi yang dibuat untuk *smartphone* harus dapat dijalankan pada sistem operasi *smartphone* yang populer dan keberlangsungan sistemnya masih didukung kedepannya, yaitu Android dan IOS.

Sekitar 6-7 tahun lalu, untuk dapat membuat sebuah aplikasi yang dapat berjalan pada *smartphone*, satu

satunya cara adalah dengan menggunakan bahasa pemrograman dan Standar Development Kit (SDK) khusus untuk masing-masing sistem operasi *smartphone* seperti, Java untuk Android dan Objective-C atau Swift untuk membuat aplikasi pada sistem operasi IOS [2]. Namun saat ini muncul *framework-framework* baru yang memanfaatkan bahasa pemrograman *Javascript / Typescript*, HTML, dan CSS/SCSS yang memudahkan seorang atau tim *programmer* dalam membuat aplikasi yang ditujukan ke *smartphone* [3]. *Framework-framework* tersebut menggunakan suatu pendekatan pemrograman yang disebut pendekatan *hybrid application*. Melalui pendekatan ini, *programmer* dapat menghasilkan suatu aplikasi yang dapat dijalankan di berbagai sistem operasi *smartphone* dengan *source code* yang sama [2]. Contoh populer dari *framework* tersebut adalah Ionic.

Tapi hanya menggunakan Ionic saja belum cukup menjadikan aplikasi memenuhi kebutuhan fitur yang dibutuhkan pengusaha ikan hias dalam menjual ikannya. Dibutuhkan cara untuk pemrosesan data ke database yang terintegrasi dan dapat diakses kapan saja dan dimana saja. Oleh karena itu, aplikasi Ionic sebagai *client* harus digabungkan dengan *web service* sebagai *server*. Arsitektur REST adalah arsitektur untuk *web service* yang mengandalkan *Uniform Resource Identifies* (URI) untuk mendeteksi resource dan interaksi apa yang harus dilakukan *web service* terhadap permintaan yang menggunakan protokol *Hypertext Transfer Protocol* (HTTP) [4].

Penelitian ini dilakukan untuk mengetahui bagaimana implementasi teknik pemrograman *hybrid mobile application* menggunakan Angular dan Ionic serta RESTful *web service* untuk servernya dalam pembuatan aplikasi *mobile marketplace* penjualan ikan hias. Pembuatan aplikasi akan menggunakan metode *rapid prototyping* sehingga aplikasi akan dapat selesai lebih cepat dan langsung mendapat respon serta penilaian dari pengusaha ikan hias serta pakar di bidang pembuatan aplikasi.

## Tinjauan Pustaka

### Hybrid Mobile

*Hybrid mobile* adalah teknik gabungan antara aplikasi *native* dan aplikasi web. Aplikasi dengan basis *hybrid* menggunakan teknologi web yang kinerjanya bergantung pada kode HTML yang berjalan di browser sehingga memungkinkan untuk pengembangan lintas platform atau sistem operasi. Hal ini dikarenakan komponen bahasa HTML, CSS dan Javascript yang sama dapat digunakan kembali pada sistem operasi yang berbeda. Hal ini dikarenakan suatu fitur yang dinamakan *web view*. *Web view* memungkinkan aplikasi yang dibangun dengan HTML, CSS dan Javascript dijalankan

layaknya aplikasi *native* pada *smartphone* [5]. Keunggulannya adalah kode bersifat *re-usable*, artinya untuk membuat aplikasi yang dapat dijalankan di berbagai sistem operasi, cukup membuat *source code* untuk 1 sistem operasi maka *source code* tersebut dapat digunakan kembali untuk membuat aplikasi di sistem operasi yang lain.

### Ionic

Ionic adalah salah satu *framework* yang digunakan untuk membuat aplikasi berbasis *hybrid mobile* yang menggunakan bahasa pemrograman utama *typescript / javascript* serta HTML5 [5]. Ionic dapat digunakan atau digabungkan dengan *framework javascript* lain seperti Angular, React, NativeScript dan lain-lain. Saat ini Ionic yang paling terbaru adalah Ionic versi 4.

### Angular

Angular merupakan pengembangan dari AngularJS yang didukung Google. Jika AngularJS menggunakan *javascript*, maka Angular menggunakan bahasa pemrograman *typescript* sebagai basis bahasa pemrogramannya [6]. *Typescript* merupakan pengembangan dari *javascript* yang dimaksudkan agar penulisan *javascript* lebih terstruktur dan dapat diandalkan untuk membuat aplikasi / program berskala besar [7], pertama kali diperkenalkan oleh Microsoft.

### RESTful Web Service

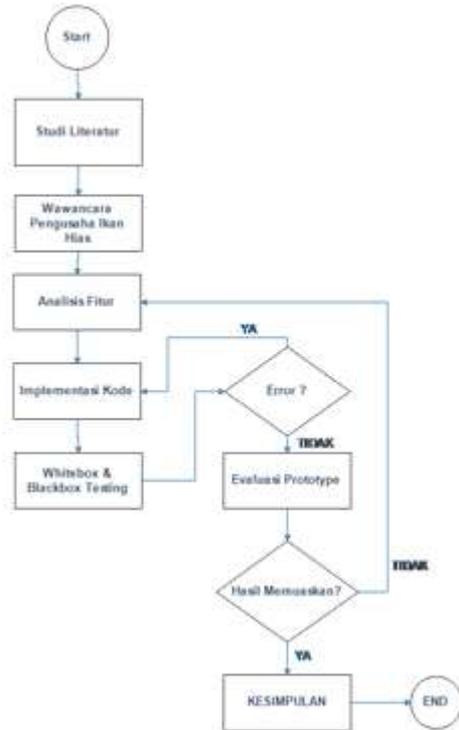
*Representational State Transfer* (REST) adalah salah satu gaya arsitektur dalam mengimplementasi *web service*. REST bukanlah sebuah standar, namun RESTful *web service* diimplementasikan menggunakan standarisasi *website*. Setiap operasi yang dilakukan pada RESTful *web service* diidentifikasi dengan alamat URL yang unik, artinya setiap operasi memiliki alamat URL yang berbeda-beda sehingga ketika server menerima permintaan melalui URL, *server* akan langsung mengetahui operasi mana yang akan dijalankan [8].

### Metode Penelitian

Langkah pertama dalam melakukan penelitian ini adalah studi literature. Pada tahap ini penulis mencari penelitian-penelitian terdahulu mengenai teknik pembuatan aplikasi *mobile*. Langkah selanjutnya adalah proses pengumpulan data terhadap proses bisnis ikan hias dengan cara melakukan wawancara dengan pengusaha ikan hias. Selanjutnya, penulis melakukan analisis awal terkait proses bisnis ikan hias sehingga menghasilkan daftar fitur yang perlu dibuat.

Lalu dengan metode *rapid prototyping* penulis akan mengimplementasikan teknik *hybrid mobile programming* dengan menggunakan *framework* Ionic versi 3 dan Angular untuk membuat *prototype* yang dapat dijalankan pada *smartphone*. Kemudian *prototype* tersebut akan diujikan dengan metode

whitebox dan blackbox testing, setelah diuji kemudian dievaluasi oleh pengusaha ikan hias dan pakar dibidang pembuatan aplikasi. Proses ini akan terus dilakukan sampai hasil evaluasi menunjukkan pengusaha ikan hias dan pakar setuju dan merasa cukup dengan semua fitur yang ada pada marketplace ikan hias. Proses penelitian dapat digambarkan menggunakan diagram alir seperti yang ditunjukkan pada Gambar 1.



Gambar 1 Diagram Alir Alur Penelitian

## Hasil dan Pembahasan

### Analisis Fitur

Selain diperoleh dari wawancara di awal penelitian hasil analisis fitur didapatkan setelah penulis melakukan wawancara secara berkala kepada pengusaha ikan hias dan juga pakar di bidang pembuatan aplikasi disetiap evaluasi prototype. Setiap kali penulis selesai menyelesaikan 1 prototype berskala kecil, penulis akan meminta *feedback* dan respon dari pengusaha ikan hias dan pakar.

Pada wawancara awal didapatkan spesifikasi kebutuhan pengusaha ikan seperti pada Tabel 1.

Tabel 1 Kebutuhan dan Fitur

Kebutuhan	Fitur
Pengusaha ikan membutuhkan wadah untuk memamerkan produk ikan hias mereka masing-masing	- Fitur <i>registrasi</i> , <i>login</i> , manajemen profil <i>user</i> - Fitur lapak <i>seller</i> dan manajemen ikan hias - Fitur pencarian ikan hias
Penggemar ikan hias dapat membeli ikan hias	- Fitur keranjang belanja - Fitur manajemen transaksi
Jaminan terhadap kepuasan dan kepercayaan pembeli	- Fitur <i>rating</i> penjual - Fitur garansi produk - Fitur <i>return</i> dan <i>refund</i> transaksi

Setelah evaluasi dilakukan beberapa kali, peneliti menemukan tambahan fitur baru untuk menyesuaikan kebutuhan dari pengusaha ikan hias dan pendapat pakar seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Kebutuhan dan Fitur Setelah Evaluasi

Kebutuhan	Fitur
Pembeli memiliki banyak alamat pengiriman	- Fitur manajemen buku alamat
Selain pembeli terhadap penjual, kepercayaan terhadap pembeli juga penting	- Fitur <i>rating</i> pembeli - Fitur <i>verified seller</i> - Fitur <i>verified buyer</i>
Media komunikasi antara pembeli dan penjual	- Fitur <i>private message</i>
Promo dan diskon sehingga menarik minat calon pembeli	- Fitur penambahan promo - Fitur diskon ikan masing-masing penjual - Fitur negosiasi harga
Penggemar ikan hias butuh wadah untuk menyimpan hasil pencarian ikan hias mereka jika suatu saat ingin kembali dan membeli ikan kesukaan mereka	- Fitur <i>shortlist</i> ikan hias favorit

### Implementasi dan Testing

Setelah mendapatkan fitur berdasarkan kebutuhan yang ada, selanjutnya melakukan implementasi kode aplikasi *mobile* dengan menggunakan Ionic versi 3 dan Angular, serta RESTful *Web Service* menggunakan *framework* Codeigniter sebagai *server*-nya. *Source code* aplikasi yang dibuat langsung ditujukan untuk 2 sistem operasi, yaitu Android dan IOS tanpa perlu dibuat *project* yang terpisah. Keunggulan dari Ionic adalah jika ada bagian kode yang perlu disesuaikan, maka dapat menggunakan modul yang telah disediakan yang disebut *platform*. Dengan *platform*, *programmer* dapat membuat kondisi percabangan untuk memilah kode mana yang dijalankan untuk tiap sistem operasi.

```

10  ngOnInit():void {
11  // this.platform.ready().then(() => {
12  //   // this.platform.is.ipad() and this.platform.is.ios() are available
13  //   // here you can do any higher level native things you might need
14  // });
15  if (this.platform.is('android')) {
16  this.statusBar.styleDefault();
17  this.statusBar.backgroundColorWithHexString('#CCCCCC');
18  } else {
19  this.statusBar.backgroundColorWithHexString('#CCCCCC');
20  this.statusBar.styleDefault();
21  this.statusBar.backgroundColorWithHexString('#CCCCCC');
22  }
23  // this.platform.registerBackButtonAction(() => {
24  //   // this one cannot be called
25  //   this.nav.root();
26  // });
27  this.platform.exitApp();
28  }
29  }
30  }
31  }
32  }
33  }
34  }
35  }
36  }
37  }
38  }
39  }
40  }
41  }
42  }
43  }
44  }
45  }
46  }
47  }
48  }
49  }
50  }
51  }
52  }
53  }
54  }
55  }
56  }
57  }
58  }
59  }
60  }
61  }
62  }
63  }
64  }
65  }
66  }
67  }
68  }
69  }
70  }
71  }
72  }
73  }
74  }
75  }
76  }
77  }
78  }
79  }
80  }
81  }
82  }
83  }
84  }
85  }
86  }
87  }
88  }
89  }
90  }
91  }
92  }
93  }
94  }
95  }
96  }
97  }
98  }
99  }
100 }
    
```

Gambar 2 Kode Inialisasi Aplikasi

Gambar 2 memperlihatkan penggunaan modul *platform* untuk mengecek *platform*/sistem operasi mana yang sedang menjalankan aplikasi, Android

atau bukan (IOS) pada baris ke-64 sampai dengan baris ke-71. Sedangkan pada Gambar 3 menunjukkan proses permintaan data ke server menggunakan modul HTTP dan fungsi *post* pada baris ke-111 sampai dengan baris ke-142. Kedua modul dan proses tersebut ditulis dalam bahasa pemrograman *typescript* menggunakan *framework* Angular dan Ionic.

```

300 cariData(buttonParam) {
301
302   if (buttonParam == 'car') {
303     this.filterData['offset'] = 0;
304     this.daftar_hasil = [];
305   }
306
307   this.presentLoading();
308
309   let url_post = this.globalVar.apiUrl+'Produk/search_data';
310
311   this.http.post(url_post, this.filterData)
312     .subscribe(
313       (val) => {
314         this.hasil = val.json();
315         this.total_data = this.hasil['total_data'];
316
317         if (this.filterData['offset'] == 0) {
318           this.daftar_hasil = this.hasil['data'];
319         } else {
320           for (let i = 0; i < this.hasil['jumlah_data']; i++) {
321             this.daftar_hasil.push(this.hasil['data'][i]);
322           }
323         }
324
325         this.filterData['offset'] = this.filterData['offset'] + this.hasil['jumlah_data'];
326
327         this.popupBort = false;
328         this.popupFilter = false;
329
330       },
331       response => {
332         this.loading.dismiss();
333       },
334       () => {
335         this.tampilkanHasil = true;
336         this.loading.dismiss();
337       }
338     );
339 }

```

Gambar 3 Kode Permintaan Data Ke Server

Gambar 4 menunjukkan kode HTML untuk menampilkan data produk populer. Semua konten yang dapat di-*scroll* pada Ionic harus berada didalam tag HTML `<ion-content></ion-content>` seperti yang terlihat pada baris ke-5. Lalu dengan menggabungkan *directive* Angular seperti *\*ngFor* untuk perulangan (baris ke-12) dan *\*ngIf* untuk percabangan (baris-ke14) *programmer* dapat secara dinamis melakukan pemrosesan data melalui HTML, bukan melalui *typescript*.

```

5 <ion-content>
6
7   <div class="itemheader">Kategori Produk</div>
8   <div class="card-wrapper slider kategori">
9
10    <ion-slides pager="false" slidesPerView="6">
11      <ion-slide class="card-image">
12        *ngFor="let item of kategori"
13        (click)="goToProdukSearch(item['id_kategori_produk'])"
14        <img
15          *ngIf="item['icon_kategori']!=null"
16          [src]="getImageKat(item['icon_kategori'])"
17
18        <p [ngClass]="getClassKat(item['icon_kategori'])"
19          {{ item['nama_kategori_produk'] }}
20        </p>
21      </ion-slide>
22      <ion-slide
23        class="card-image"
24        (click)="goToProdukSearch('')"
25        <p class="lain-lain">
26          lain<br> - <br>lain
27        </p>
28      </ion-slide>
29    </ion-slides>
30  </div>
31

```

Gambar 4 Kode HTML

Kode *web service*-nya ditunjukkan pada Gambar 5. *Web service* diimplementasi menggunakan bahasa PHP dengan *framework* Codeigniter. Pada kode ini ditunjukkan proses pengambilan variabel yang diterima melalui permintaan HTTP (baris ke-29) dan diproses untuk pengambilan data menggunakan fungsi yang terdapat pada *model* Codeigniter (baris ke-37 dan ke-39), terakhir semua data dikembalikan ke aplikasi menggunakan format JSON yang kecepatannya lebih cepat dibandingkan XML (baris ke-43).

```

28 public function search_data() {
29   $data_post = json_decode(file_get_contents("php://input"));
30   $limit = 'sema';
31   $start = $data_post->offset==null?0:$data_post->offset;
32   $keyword = $data_post->id_sub_kategori;
33
34   $orderby = $data_post->orderby;
35   $sortMethod = $data_post->sortMethod;
36
37   $hasil_pencarian = $this->Produk_model->get_search($limit, $start,$keyword);
38
39   $total_data = $this->Produk_model->count_search($keyword)->banyak;
40
41   $data_output = array('data' => $hasil_pencarian->result_array, 'jumlah_data' => $hasil_pencarian->num_rows(), 'total_data' => $total_data);
42
43   echo json_encode($data_output);
44
45 }

```

Gambar 5 Kode Web Service

Semua kode pada aplikasi dilakukan *testing* menggunakan metode *whitebox* dan *blackbox testing*. Proses *testing* ini dilakukan untuk mengecek adanya *bug* dan kesalahan logika di setiap halaman atau fitur aplikasi. Hasil implementasi kode setelah di-*build* dapat dilihat pada Gambar 6.



Gambar 6 Hasil Build Aplikasi

Setelah semua kode program berjalan lancar dan dapat dijalankan di 2 sistem operasi, barulah *prototype* dievaluasi.

**Evaluasi**

Evaluasi dilakukan setiap selesai melakukan implementasi dan *testing* kode pada *prototype*. *Prototype* akan diuji coba oleh pakar pembuatan aplikasi dan pengusaha ikan hias. Penulis akan menanyakan respon mereka mengenai *prototype* yang sudah di-*testing* dalam bentuk wawancara. Adapun yang ditanyakan dalam wawancara ditunjukkan melalui Tabel 3.

Tabel 3 Daftar Pertanyaan

No.	Pertanyaan
1	Bagaimana desain user interface aplikasi ini, apakah rapi dan enak dipandang mata?
2	Apakah nyaman dalam menggunakan aplikasi ini?
3	Bagaimana dengan kecepatan loading atau performa aplikasi?
4	Apakah fitur yang tersedia pada aplikasi ini sesuai dengan keadaan bisnis ikan hias saat ini?
5	Apakah aplikasi ini layak di- <i>publish</i> di google play store atau appstore?

Pada *prototype* versi pertama, didapatkan evaluasi bahwa aplikasi masih sederhana dan perlu banyak perbaikan *user interface* dan penambahan fitur.

*Prototype* kedua masih diperlukan penambahan fitur sesuai perkembangan kebutuhan penggemar ikan hias saat ini dan *user interface* masih perlu diperbaiki. Selanjutnya pada versi ketiga, fitur dirasa sudah cukup mewakili kebutuhan pengusaha ikan hias namun masih perlu perbaikan di sisi desain *User Experience* (UX) karena aplikasi dirasa terlalu kompleks dan berbelit-belit. Pada versi keempat, desain UX sudah dianggap cukup namun masalah lain terdapat pada kecepatan loading, dimana ketika data masih sedikit belum terlalu dirasakan efeknya namun ketika data mencapai puluhan, untuk menampilkan data hasil pencarian dirasa terlalu lama. Barulah pada *prototype* versi kelima, aplikasi dirasa sudah cukup bagus dan memuaskan serta pantas untuk di-*publish* ke *playstore* atau *appstore* guna mendapatkan pengguna lebih banyak seperti yang ditampilkan pada Gambar 7.



Gambar 7 Aplikasi publish di playstore

**Kesimpulan dan Saran**

Kesimpulan yang didapat dari penelitian ini adalah bahwa dengan menggunakan teknik *hybrid mobile application*, *programmer* dapat membuat aplikasi *marketplace* ikan hias dengan lebih efektif dan efisien baik dari segi waktu pengerjaan maupun tenaga dan biaya dikarenakan *programmer* cukup membuat 1 *source code* yang dapat di-*build* atau di-*compile* menjadi aplikasi yang dapat dijalankan di Android dan IOS. Selain itu berdasarkan hasil evaluasi, aplikasi *marketplace* diterima dan layak digunakan.

Saran untuk penelitian selanjutnya adalah untuk meneliti dan memperbaiki celah keamanan RESTful *web service* pada *server* aplikasi serta menambah kajian fitur yang dibutuhkan sesuai perkembangan teknologi dan bisnis.

## Daftar Pustaka

- [1] M. Huda, I. Lestari, and A. Trisnadoli, "Analisis Hasil Implementasi Pengembangan Aplikasi Mobile Pendekatan Hybrid Pada Aplikasi Family Tracking," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 1, pp. 17–22, May 2019.
- [2] M. Q. Huynh, P. Ghimire, and D. Truong, "Hybrid App Approach: Could It Mark the End of Native App Domination?," *Issues Informing Sci. Inf. Technol.*, vol. 14, pp. 049–065, 2017.
- [3] S. Alfeno and W. I. Tiana, "APLIKASI MOBILE COMMERCE (M-COMMERCE) BERBASIS ANDROID HYBRID," *CERITA J.*, vol. 4, no. 2, pp. 169–179, May 2018.
- [4] A. Neumann, N. Laranjeiro, and J. Bernardino, "An Analysis of Public REST Web Service APIs," *IEEE Trans. Serv. Comput.*, p. 1, Feb. 2018.
- [5] A. Ravulavaru, *Learning Ionic - Build Hybrid Mobile Applications with HTML5*. Birmingham: Packt Publishing, 2015.
- [6] S. Holmes, *Getting MEAN with Mongo, Express, Angular, and Node*. 2015.
- [7] T. C. Huber, *Getting Started with TypeScript: Includes Introduction to Angular*. Scotts Valley: CreateSpace Independent Publishing Platform, 2017.
- [8] P. J. Deitel and H. M. Deitel, *AJAX, rich Internet applications, and Web development for programmers*, vol. 46, no. 03. Boston: Pearson Education, 2008.