BAB II

TINJAUAN PUSTAKA

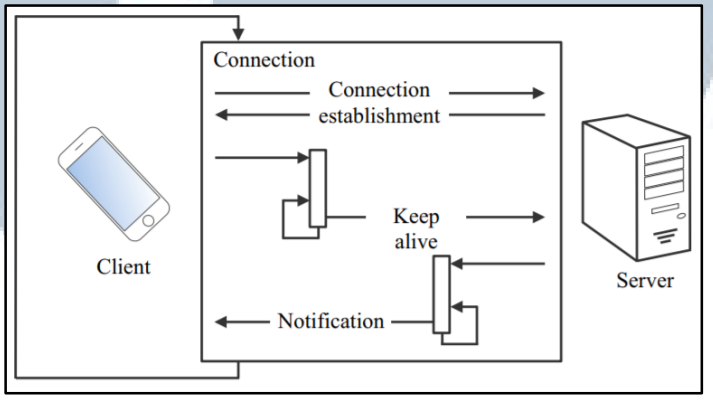
2.1 Push Notification

Push notification adalah pesan kecil dan ringkas yang digunakan oleh aplikasi *mobile* untuk memberitahu pengguna mengenai *event* terbaru (Acer dkk., 2015). Menurut Ding dkk. (2014), *push notification* merupakan fitur yang terutama dalam layanan *mobile computing* dan telah diimplementasikan secara luas pada aplikasi *mobile*. Layanan *push notification* pada *mobile* juga telah menjadi layanan yang penting untuk mengirimkan konten ke pengguna (Liu, 2011). *Push notification* memperbolehkan untuk mengirim pesan ke *end users* walaupun aplikasi sedang tidak berjalan (Liu, 2011). Hal ini menyelesaikan masalah yang dihasilkan dari *smartphone* yang tidak mendukung fitur *multi-tasking*, penghematan baterai, dan aplikasi yang belum terbuka (Liu, 2011). Dengan menggunakan *push notification*, informasi yang dibutuhkan pengguna dapat langsung dikirimkan daripada pengguna dari suatu sistem informasi harus melakukan request ke dalam sistem (Umbach, 1997).

Pan dkk. (2015) menjelaskan bahwa sejak pengembangan *Internet of Things*, semakin banyak *device* yang terkoneksi dengan *Internet*, seperti sensor, *wearable devices*, dan alat-alat rumah tangga. Notifikasi dapat di-*push* pada *device-device* seperti ini. Beberapa *push notification* juga dapat diinisiasi oleh *device-device* ini untuk memantau lingkungan/tubuh/alat atau *providing context*, seperti lokasi. Konvergensi *mobile* dengan IoT juga memberikan tantangan baru pada bagaimana sistem dapat menangani *mixed push channels* yang dirancang oleh

komunikasi M2M (*Machine to Machine)* dan interaksi manusia, serta memungkinkan interaksi yang efektif dengan melibatkan manusia dan IoT *devices*.

Metode *push* membutuhkan dua software yang saling memahami satu sama lain, yaitu *software* untuk klien yang mengerti bagaimana cara untuk melakukan *request*, menerima, menyimpan, hingga menampilkan data dan *software* untuk *server* yang mengerti *request* yang dikirimkan oleh klien, sehingga dapat *server* dapat mengirimkan datanya (Umbach, 1997). Skema cara kerja metode *push* dapat dilihat pada Gambar 2.1



Gambar 2.1 Skema Cara Kerja Metode Push
(Sumber: Burgstahler dkk., 2013)

2.2 Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) atau yang dahulu dikenal sebagai Google Cloud Messaging (GCM) adalah sebuah solusi dalam pengiriman pesan antar *platform* yang memungkinkan pengiriman dilakukan secara *reliable* dan tanpa biaya. Dengan menggunakan FCM, aplikasi klien dapat dinotifikasi ketika terdapat data baru yang siap disinkronisasi. FCM juga memungkinkan untuk mengirimkan pesan notifikasi kepada masing-masing pengguna atau pengguna pada *segment* tertentu (firebase.google.com, Tanpa Tahun).

Terdapat dua komponen utama dalam mengimplementasikan Firebase Cloud Messaging pada sebuah sistem. Aplikasi *server* yaitu, *trusted environment* atau *Notifications Console GUI* pada Firebase yang digunakan untuk mengirimkan dan menargetkan pesan notifikasi ke klien. Aplikasi klien yang berupa *iOs, Android* atau *Web* (*JavaScript*) untuk menerima pesan notifikasi.

Gambar 2.2 Arsitektur FCM
(Sumber: https://firebase.google.com/docs/cloud-messaging/)

Gambar 2.2 merupakan arsitektur dari FCM. Pengiriman pesan notifikasi dari *server* ke klien dapat dilakukan dengan dua cara, yaitu melalui *Notifications console* GUI dan *trusted environment*. Penggunaan *console* sangat berguna dalam uji coba atau untuk *highly targeted marketing* dan *user engagement*. Pengiriman pesan dengan cara ini tidak membutuhkan teknik untuk *coding*, hanya memasukkan data-data yang dibutuhkan. Pada *trusted environment*, pengiriman pesan dapat dilakukan melalui *Admin* SDK dan HTTP/XMPP. *Admin* SDK menyediakan sebuah API untuk mengirimkan pesan dari FCM kepada *end user devices*, tetapi SDK ini hanya mendukung bahasa pemrograman *Node.js*. Dalam mengirimkan pesan notifikasi dapat dilakukan *programmatically* kepada masing-masing *devices*,

grup *devices*, *topics*, dan *conditions*. HTTP/XMPP merupakan dua *server* protocol yang disediakan oleh FCM. Pengiriman melalui HTTP hanya dapat dilakukan secara *downstream*, yaitu dari *cloud-device*. Sedangkan, XMPP dapat dilakukan secara *upstream* dan *downstream*, yaitu dari *device-cloud* dan *cloud-device* (firebase.google.com, Tanpa Tahun).

Untuk dapat menerima pesan notifikasi dari Firebase, aplikasi harus menggunakan *service* yang meng-*extend* FirebaseMessagingService, dimana *service* ini digunakan untuk meng-*handle* setiap notifikasi yang didapatkan. Sedangkan, untuk dapat mengirimkan pesan notifikasi ke *device* yang diinginkan, aplikasi harus meng-*extend* FirebaseInstanceIdService agar mendapatkan *registration token* yang berguna untuk membedakan aplikasi yang ter-*install* pada setiap *device* (firebase.google.com, Tanpa Tahun).

Firebase Cloud Messaging memiliki dua buah tipe pesan notifikasi yang dapat dikirimkan kepada klien, yaitu *notification messages* dan *data messages*. Tipe pesan notifikasi yang pertama, yaitu *notification messages*, secara otomatis ditampilkan oleh FCM pada *devices* pengguna. Pada tipe pesan notifikasi ini, FCM yang akan meng-*handle* pesan notifikasi. Sedangkan, pada *data messages*, pesan notifikasi akan diproses oleh aplikasi klien. Format data yang digunakan untuk komunikasi antara klien dengan *server* adalah JavaScript Object Notation (JSON). Gambar 2.3 dan Gambar 2.4 merupakan contoh JSON yang akan dikirimkan dari *server*.

```
{
  "to" : "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
  "notification" : {
    "body" : "great match!",
    "title" : "Portugal vs. Denmark",
    "icon" : "myicon"
  }
}
```

Gambar 2.3 Contoh Notification Messages FCM
(Sumber: https://firebase.google.com/docs/cloud-messaging /concept-options)

```
{
  "to" : "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
  "data" : {
    "Nick" : "Mario",
    "body" : "great match!",
    "Room" : "PortugalVSDenmark"
  },
}
```

Gambar 2.4 Contoh *Data Messages* FCM
(Sumber: https://firebase.google.com/docs/cloud-messaging /concept-options)

2.3 Google Maps Distance Matrix API

Google Maps Distance Matrix API adalah sebuah layanan yang menyediakan jarak dan waktu perjalanan dari *origins* ke *destinations* yang diinginkan berdasarkan rute yang direkomendasikan antara titik awal dan akhir. Untuk dapat mengakses API ini, dibutuhkan HTTP untuk me-*request* URL yang diinginkan, yaitu menggunakan *origins* dan *destinations*, serta API *key*.

```
https://maps.googleapis.com/maps/api/distancematrix/json?units=imperial&origins=Washington,DC&destinations=New+York+City,NY&key=YOUR_API_KEY
```

Gambar 2.5 Contoh URL *Request* Pada *Distance Matrix* API
(Sumber: https://developers.google.com/maps/documentation/distance-matrix/start)

Gambar 2.5 merupakan contoh URL untuk mengakses API Distance Matrix dengan *origins* berada pada Washington, DC dan *destinations* berada pada New York City, NY (developers.google.com, Tanpa Tahun).

```
{
   "destination_addresses" : [ "New York, NY, USA" ],
   "origin_addresses" : [ "Washington, DC, USA" ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "225 mi",
                  "value" : 361715
               },
               "duration" : {
                  "text" : "3 hours 49 mins",
                  "value" : 13725
               },
               "status" : "OK"
            }
         ]
      }
   ],
   "status" : "OK"
}
```


Gambar 2.6 Contoh *Response* Dari *Distance Matrix* API
(Sumber: https://developers.google.com/maps/documentation/distance-matrix/start)

Gambar 2.6 merupakan *response* dari Google Maps Distance Matrix API. *Response* ini berisi jarak, durasi, dan alamat antara *origins* dan *destinations*.

2.4 Click Ratio

Untuk mendapatkan bagaimana reaksi pengguna atas suatu notifikasi yang didapat, dilakukan pengukuran untuk *click time* (Shirazi dkk., 2014). Penelitian yang dilakukan oleh Shirazi dkk. (2014) adalah mengukur jumlah klik untuk notifikasi pada setiap kategori aplikasi *mobile*.

Pada penelitian ini akan dilakukan penghitungan *click ratio* untuk notifikasi yang didapat oleh pengguna. *Click ratio* dihitung dengan membagi jumlah klik pada notifikasi dengan jumah notifikasi yang dikirimkan.

2.5 Click Time

Untuk mendapatkan selisih waktu antara notifikasi yang didapat dengan notifikasi yang diklik, digunakan *click time*. Penelitian yang dilakukan oleh Shirazi dkk. (2014) adalah mengukur *click time* per kategori aplikasi *mobile*. Hasil penelitian yang didapat adalah kategori *system* memiliki *click time* tercepat dan kategori *messenger* pada urutan kedua. Kesimpulan yang didapat mengenai *click time* adalah lima puluh persen interaksi pengguna dengan notifikasi, terjadi pada selang waktu tiga puluh detik setelah notifikasi didapatkan.

Pada penelitian ini akan diukur *click time* terhadap notifikasi yang didapat oleh pengguna, yaitu dengan menghitung selisih waktu antara notifikasi yang didapat dan notifikasi yang diklik oleh pengguna. Selanjutnya jumlah *click time* tersebut akan dibagi dengan jumlah notifikasi yang dikirimkan.

2.6 User Experience Questionnaire

Pengalaman pengguna adalah pengalaman menyeluruh yang terdiri dari seriap aspek dari interaksi pengguna terhadap suatu produk atau layanan (Park dkk., 2013). *User Experience Questionnaire* (UEQ) merupakah sebuah penilaian yang dapat dilakukan secara cepat untuk mendapatkan pengalaman pengguna dari sebuah produk interaktif (UEQ-Online, 2015). Tujuan utama dari UEQ adalah untuk mendapatkan penghitungan yang cepat dan langsung dari pengalaman pengguna terhadap suatu produk (Laugwitz dkk., 2008). Contoh UEQ dapat dilihat pada Gambar 2.7. Sugiyono (2010) mengatakan dalam bukunya bahwa menurut Roscoe, jumlah minimal ukuran sampel yang layak adalah 30 sampel. Oleh karena itu, responden yang akan mengisi kuesioner ini minimal 30 orang.

	1	2	3	4	5	6	7		
annoying	○	○	○	○	○	○	○	enjoyable	1
not understandable	○	○	○	○	○	○	○	understandable	2
creative	○	○	○	○	○	○	○	dull	3
easy to learn	○	○	○	○	○	○	○	difficult to learn	4
valuable	○	○	○	○	○	○	○	inferior	5
boring	○	○	○	○	○	○	○	exciting	6
not interesting	○	○	○	○	○	○	○	interesting	7
unpredictable	○	○	○	○	○	○	○	predictable	8
fast	○	○	○	○	○	○	○	slow	9
inventive	○	○	○	○	○	○	○	conventional	10
obstructive	○	○	○	○	○	○	○	supportive	11
good	○	○	○	○	○	○	○	bad	12
complicated	○	○	○	○	○	○	○	easy	13
unlikable	○	○	○	○	○	○	○	pleasing	14
usual	○	○	○	○	○	○	○	leading edge	15
unpleasant	○	○	○	○	○	○	○	pleasant	16
secure	○	○	○	○	○	○	○	not secure	17
motivating	○	○	○	○	○	○	○	demotivating	18
meets expectations	○	○	○	○	○	○	○	does not meet expectations	19
inefficient	○	○	○	○	○	○	○	efficient	20
clear	○	○	○	○	○	○	○	confusing	21
impractical	○	○	○	○	○	○	○	practical	22
organized	○	○	○	○	○	○	○	cluttered	23
attractive	○	○	○	○	○	○	○	unattractive	24
friendly	○	○	○	○	○	○	○	unfriendly	25
conservative	○	○	○	○	○	○	○	innovative	26

Gambar 2.7 Contoh UEQ
(Sumber: http://www.ueq-online.org/)

Pada penelitian ini akan digunakan hasil UEQ untuk didapatkan pengalaman pengguna terhadap produk ini. Pada UEQ terdapat enam skala yang dituangkan dalam 26 butir penilaian untuk didapatkan hasil UEQ, dimana keenam skala tersebut adalah *Attractiveness*, *Perspicuity*, *Efficiency*, *Dependability*, *Stimulation*, dan *Novelty* (Santoso dkk., 2016).

Attractiveness merupakan kesan umum terhadap produk, apakah pengguna menyukainya atau tidak. Terdapat enam butir untuk skala ini, yaitu *annoying* /

enjoyable, good / bad, unlikable / pleasing, unpleasant / pleasant, attractive / unattractive, friendly / unfriendly. Perspicuity merupakan kemudahan untuk mengerti bagaimana menggunakan produk. Terdapat empat butir untuk skala ini, yaitu *not understandable / understandable, easy to learn / difficult to learn, complicated / easy, clear / confusing.*

Efficiency merupakan kemudahan pengguna untuk menggunakan produk dengan cepat dan efisien. Terdapat empat butir untuk skala ini, yaitu *fast / slow, inefficient / efficient, impractical / practical, organized / cluttered. Dependability* merupakan skala untuk mengukur bagaimana perasaan pengguna terhadap interaksi dengan sistem, apakah interaksi tersebut aman dan sesuai harapan. Terdapat empat butir untuk skala ini, yaitu *unpredictable / predictable, obstructive / supportive, secure / not secure, meets expectations / does not meet expectations.*

Stimulation merupakan skala untuk mengukur tingkat ketertarikan pengguna terhadap produk, apakah pengguna menginginkan untuk menggunakan produk lebih lanjut atau tidak. Terdapat empat butir untuk skala ini, yaitu *valuable / inferior, boring / exciting, not interesting / interesting, motivating / demotivating. Novelty* merupakan pandangan pengguna mengenai tingkat kreativitas produk. Terdapat empat butir untuk skala ini, yaitu *creative / dull, inventive / conventional, usual / leading edge, conservative / innovative.*

Menurut panduan UEQ (Santoso dkk., 2016), UEQ menggunakan skala dengan jarak antara +3 untuk pernyataan positif dan -3 untuk pernyataan negatif. Setelah pengguna selesai mengisi kuesioner, akan dilakukan penghitungan untuk didapatkan nilai rata-rata. Apabila nilai tersebut diantara -0.8 dan 0.8, maka skala

tersebut dianggap netral. Sedangkan, apabila dibawah -0.8, maka dianggap jelek dan begitu pun sebaliknya, apabila diatas 0.8, maka dianggap bagus.

2.7 Hands-on Measurement

Hands-on measurement merupakan sebuah metode yang didesain untuk mengukur kegunaan aplikasi *mobile* secara kuantitas (Nayebi dkk., 2012). Dalam pengembangan sebuah aplikasi, metode ini dapat digabungkan bersama dengan *field studies*. Menurut Nayebi dkk. (2012), metode ini cocok digunakan apabila aspek-aspek yang ingin diukur pada aplikasi *mobile* telah jelas, sehingga dapat membuat peneliti untuk mendapatkan data-data untuk perbandingan menjadi lebih akurat.

2.8 Field Studies

Field studies merupakan metode yang umum digunakan untuk mengumpulkan data-data mengenai pengguna, keinginan pengguna, dan kebutuhan produk (Nayebi dkk., 2012). Pada metode ini partisipan atau pengguna dilibatkan secara langsung dengan melakukan pekerjaan-pekerjaan yang mungkin dapat terjadi pada lingkungan yang sebenarnya. Pada makalah Nayebi dkk. (2012) disebutkan bahwa beberapa penelitian lain menyarankan untuk memberikan kuesioner kepada partisipan pada akhir periode pengujian.

2.9 Alat Nolong.in

Alat Nolong.in merupakan alat yang dibuat pada penelitian Gunawan tahun 2017. Alat ini memiliki tombol yang dapat digunakan untuk memberi sinyal kepada aplikasi bahwa pengguna sedang mengalami suatu insiden. Alat ini menggunakan Arduino Pro Mini 3.3V sebagai komponen utama untuk dapat menjalankan setiap

perintah yang diberikan. Selain menggunakan Arduino, alat ini juga menggunakan Bluetooth HC-05 untuk dapat mengoneksikan alat dengan aplikasi notifikasi insiden. Pada alat ini terdapat baterai lithium 3.7V dengan kapasitas sebesar 2200 mAh. Alat Nolong.in juga memiliki *switch* untuk menyalakan dan mematikan alat supaya dapat menghemat baterai.

Gambar 2.8 Alat Nolong.in