

# Pengembangan API Aplikasi Android Hi-Depok pada Proyek 1000 Aplikasi *Smart City* Kota Depok di Diskominfo Kota Depok

Toriq Wahid Syaefullah  
Jurusan Teknik Informatika dan Komputer  
Politeknik Negeri Jakarta  
Depok, Indonesia  
[toriq.syaefullah.tik14@mhs.wpnj.ac.id](mailto:toriq.syaefullah.tik14@mhs.wpnj.ac.id)

**Abstrak** – Kota Depok mengklaim sudah siap untuk menuju *smart city*. Pemerintah Kota Depok bersama Dinas Komunikasi dan Informatika (Diskominfo) berencana membuat proyek 1000 Aplikasi *Smart City* untuk mewujudkan Rencana Pembangunan Jangka Menengah Daerah (RPJMD) kota Depok tahun 2016 yang meliputi peningkatan infrastruktur, daya saing dan ketahanan ekonomi, kualitas lingkungan hidup, ekonomi kreatif dan UMKM berbasis potensi lokal, tata kota, ketahanan keluarga dan sosial kemasyarakatan, ketimpangan sosial, kemiskinan dan pengangguran, kesehatan dan kualitas SDM. *Smart city* menjadi salah satunya yang akan dibangun oleh pemkot Depok. Hi-Depok adalah aplikasi berbasis Android dan *website* yang diciptakan oleh mahasiswa Politeknik Negeri Jakarta program studi Teknik Informatika Reguler angkatan 2014, aplikasi ini mendapat apresiasi yang luar biasa sehingga dapat menjadi cikal bakal lahirnya program 1000 *Smart City* Aplikasi Kota Depok. Pengembangan lanjutan aplikasi Android Hi-Depok dengan mengubah bisnis prosesnya serta menambahkan beberapa fungsi dan modul pada *backend* aplikasi seperti API Google Maps dan JSON. Dalam pengujian aplikasi ini menggunakan metode *blackbox testing* untuk menguji fungsi pada *backend* sehingga aplikasi menjadi layak untuk digunakan oleh masyarakat Kota Depok.

**Kata kunci** – *Smart City*, API Google Maps, *Blackbox Testing*, JSON, Android.

## I. PENDAHULUAN

Negara Indonesia saat ini sedang menuju negara dengan perkembangan teknologi digital yang sangat pesat, hal ini berdampak pada hampir semua aspek kehidupan masyarakat sehingga diperlukan pelayanan informasi yang cepat, tepat dan akurat. Dengan perkembangan teknologi saat ini, *mobile phone* menjadi sarana informasi dengan kemampuan yang lebih cepat dan populer dibandingkan dengan *desktop computer*, karena menurut data *Annual Startup Report* DailySocial (2015) masyarakat Indonesia yang menggunakan *mobile phone* secara aktif telah mencapai 281,9 juta orang. Jumlah tersebut

menggambarkan bahwa setiap orang di Indonesia memegang ponsel sebanyak 1,13 unit [1].

Kota Depok tergolong sebagai salah satu kota yang masih baru di Indonesia, namun mempunyai kompleksitas yang sangat tinggi, dimulai dari kepadatan penduduk, transportasi, pendidikan, kesehatan, lingkungan dan sebagainya. Sehingga kota Depok memiliki permasalahan yang terus bertambah dalam penataan ruang yang diakibatkan oleh pembangunan kotanya sendiri. Maka dari itu, dibutuhkan manajemen kota melalui pendekatan konsep perencanaan yang berkelanjutan dengan bekerjasama dengan Organisasi Perangkat Daerah (OPD) untuk mencari jalan keluar untuk menyelesaikan masalah tersebut.

Strategi Rencana Pembangunan Jangka Menengah Daerah (RPJMD) Kota Depok tahun 2016-2021 adalah meningkatkan infrastruktur, daya saing dan ketahanan ekonomi, kualitas lingkungan hidup, ekonomi kreatif dan UMKM berbasis potensi lokal, tata kota, ketahanan keluarga dan sosial kemasyarakatan, ketimpangan sosial, kemiskinan dan pengangguran, kesehatan dan kualitas SDM. Untuk itu, Dinas Komunikasi dan Informatika (Diskominfo) sebagai salah satu Organisasi Perangkat Daerah (OPD) yang bertanggung jawab sebagai pengelola informasi dan komunikasi publik di Kota Depok mengklaim bahwa Kota Depok sudah siap menjadi *Smart City*. *Smart City* merupakan konsep yang menengahkan sebuah tatanan kota cerdas yang bisa berperan dalam memudahkan masyarakat untuk mendapatkan informasi secara cepat dan tepat dengan memanfaatkan teknologi IT.

Konsep *Smart City* sudah banyak diterapkan di kota-kota besar di Indonesia seperti Jakarta *Smart City*, Bandung *Smart City*, Surabaya *Smart City* dan lain-lain. Namun nampaknya perkembangannya masih belum membawa pemahaman yang jelas dan konsisten, karena agar kota tersebut dapat dikatakan sebagai *Smart City* dibutuhkan pemikiran, tenaga dan waktu yang tidak sebentar. Seperti salah satu program dari Dinas Informasi dan Komunikasi (Diskominfo) Kota Depok yaitu dengan membuat program 1000 Aplikasi *Smart City* yang dapat dikembangkan komunitas-komunitas yang ada di Kota Depok dan berkoordinasi dengan pihak Diskominfo Kota Depok agar aplikasi yang telah dikembangkan dapat terintegrasi, salah satu aplikasinya adalah aplikasi Android Hi-Depok.

Berdasarkan penjelasan yang sudah dipaparkan, maka dibutuhkan pengembangan lanjutan aplikasi Android Hi-Depok dengan menambahkan beberapa fungsi dan modul pada *backend* aplikasi agar bisnis proses dan tujuan program 1000 Aplikasi *Smart City* Kota Depok sebagai sarana pelayanan dan informasi publik yang dapat diakses kapan saja dan dimana saja oleh masyarakat di Kota Depok dapat tercapai.

## II. TINJAUAN PUSTAKA

### a. *Smart City*

Seorang peneliti dan profesor yang mendalami bidang kewirausahaan, dan keberlanjutan sumber daya mencetuskan diagram tentang kota pintar, yaitu Boyd Cohen membagi *smart city* menjadi 6 dimensi, yaitu: (1) *Smart economy*; (2) *Smart mobility*; (3) *Smart environment*; (4) *Smart people*; (5) *Smart living*; dan (6) *Smart governance* [2].

Berdasarkan pengertian tersebut, *smart city* merupakan konsep yang mengetengahkan sebuah tatanan kota cerdas yang bisa berperan dalam memudahkan masyarakat untuk mendapatkan informasi secara cepat dan tepat dengan memanfaatkan teknologi IT serta dapat mendorong peran aktif masyarakat sehingga berpartisipasi dalam pengelolaan kota sehingga terjadi interaksi antara warga dengan Pemerintah Daerah setempat sebagai penyedia layanan. Interaksi dua arah ini akan terus berkembang dan berproses sehingga nantinya kota akan menjadi tempat yang nyaman untuk ditinggali serta dapat merespon perubahan dan tantangan baru dengan lebih cepat. Dan juga, keenam dimensi *smart city* tersebut harus dikembangkan bersama-sama dan seimbang untuk mewujudkan suatu kota yang pintar. Seperti petunjuk Cohen, mewujudkan *smart city* harus dengan mencoba melangkah menggunakan modal yang minim dan terukur.

### b. API (*Application Programming Interface*)

JSON biasanya digunakan programmer untuk melakukan *encode* dan *decode* data antara *server* dan aplikasi menggunakan *web services*. Namun menurutnya, JSON bukanlah bagian dari JavaScript walaupun *syntax* nya hampir mirip tetapi JSON tidak semata-mata digunakan oleh JavaScript, karena JSON memiliki *parsers* dan *serializers* yang ada dalam banyak bahasa pemrograman [3].

Berdasarkan pengertian tersebut, JSON dapat diartikan sebagai fungsi yang dapat menyandi data dari *server* dan dapat dikembalikan lagi sehingga data dapat dimunculkan pada aplikasi. Biasanya fungsi JSON sudah tersedia pada *web services*.

### c. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi, karena Android menyediakan platform terbuka bagi pengembang untuk menciptakan aplikasi baru karena sistem operasi yang berbasis *open source*.

Android Studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan dan pemrograman aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 Mei 2013 pada *event* Google I/O *Conference* untuk tahun

2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android. Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT *plugin* (*Android Development Tools*) [4].

Berdasarkan pengertian tersebut, Android merupakan generasi baru platform *mobile* yang memberikan banyak fasilitas pengembangan yang *free* dan *open source* sehingga saat ini banyak dipakai oleh *programmer* untuk mengembangkan aplikasinya ke ranah *mobile*. Aplikasi android dapat dengan mudah dikembangkan menggunakan IDE Android Studio dari Google karena telah banyak plugin yang mendukung untuk pengembangan aplikasi. Umumnya Android Studio menggunakan bahasa pemrograman berbasis Java untuk pengembangannya, karena kemudahannya dan penggunaan memori yang ringan.

### d. Java

Java adalah sebuah bahasa pemrograman yang sangat terkenal, dikembangkan oleh Sun Microsystems dan mulai diperkenalkan kepada publik pada tahun 1995. Ia menambahkan bahwa seperti halnya C++, Java juga merupakan bahasa yang berorientasi objek. Dengan demikian, java memudahkan dalam pembuatan aplikasi yang berskala besar. Java menggunakan perintah-perintah yang mudah dimengerti oleh orang, karena Java mempunyai keunggulan yaitu bersifat universal. Java bisa dijumpai di berbagai macam platform seperti Linux, UNIX, Windows, Mac dan lain-lain selama sistem target memiliki JRE (*Java Runtime Environment*) [5].

Berdasarkan pengertian tersebut, dapat dikatakan bahasa Java merupakan salah satu pelopor bahasa pemrograman berorientasi objek. Bahkan telah berevolusi dari bahasa pemrograman menjadi sebuah teknologi yang dapat ditemui dengan mudah pada *mobile phone*, *computer*, hingga *server*.

### e. *Black Box Testing*

*Black box testing* atau disebut juga *behavioral testing* atau *functional testing* berfokus pada keperluan fungsional perangkat lunak. Dengan demikian, teknik pengujian *black box* memungkinkan kita untuk memperoleh serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk sebuah program [6]. Metode ini biasanya digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan interface.
3. Kesalahan dalam struktur data atau akses basis data eksternal.
4. Kesalahan lahan kinerja, inisialisasi dan kesalahan terminasi.

### f. UML (*Unified Modeling Language*)

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Menurut UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak karena hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada

kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek [7].

Berdasarkan pengertian tersebut, UML dapat diartikan sebagai standar untuk mendefinisikan kebutuhan-kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Diagram UML yang paling umum digunakan adalah *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*.

### III. HASIL DAN PEMBAHASAN

#### 1. Bisnis Proses

##### a. Sebelum aplikasi dikembangkan

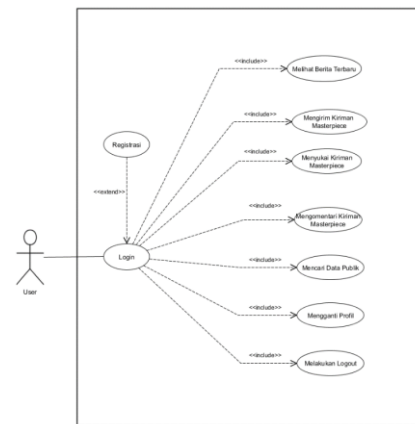
Aplikasi Hi-Depok terdiri dari beberapa modul dan sub modul yang ditujukan untuk warga Depok, bertujuan untuk memudahkan segala aktivitas masyarakat Depok. Adapun modul yang akan dibuat yaitu Sikepok, Lapok, KaDepok, Kapok, Ucok, dan Fokopok yang berfungsi sebagai wadah informasi yang dibutuhkan oleh masyarakat Depok. Aplikasi ini memiliki kegunaannya masing-masing seperti, Sikepok (Sistem Kesehatan Depok) untuk membantu aktivitas warga apabila memiliki kesulitan dalam mencari tahu permasalahan dalam bidang kesehatan, KaDepok (Kasihi Depok) untuk membantu warga Depok yang ingin berpartisipasi dalam bidang sosial, Lapok (Lapor Depok) untuk warga yang ingin memberi laporan perihal keluhan yang ada di Depok, Kapok (Kenal Depok) merupakan aplikasi yang berisi semua informasi tentang Depok, Ucok merupakan aplikasi yang berisi informasi mengenai usaha kecil dan juga menengah (UKM) di Depok, dan Fokopok (Forum Komunitas Depok) merupakan aplikasi yang berisi informasi mengenai komunitas di Depok.

##### b. Setelah aplikasi dikembangkan

Aplikasi Hi-Depok setelah dikembangkan mengalami banyak perubahan yang awalnya sebagai aplikasi pelayanan dan pelaporan masyarakat kepada pemerintah menjadi aplikasi yang lebih fokus pada informasi publik dan komunitas. Sehingga modul-modul seperti Lapok (Lapor Depok), Ucok (Usaha Kecil dan Menengah Depok), Fokopok (Forum Komunitas Depok), Sikepok (Sistem Informasi Kesehatan Depok), Kapok (Kenal Depok) dan KaDepok (Kasihi Depok) dieliminasi sehingga modul-modul sekarang yaitu *Masterpiece*, *News* dan *Event*, *Chat*, dan *Kenal Depok*. Dalam modul *Masterpiece*, pengguna dapat mengirimkan bakat atau karyanya dalam bentuk foto dan deskripsi sehingga pengguna lain dapat melihat dan memberikan respon suka dan memberikan komentar. Kiriman yang terverifikasi dan mendapat respon terpopuler akan masuk ke dalam menu Museum Depok sehingga karya-karyanya dapat dilihat oleh pengguna di seluruh Kota Depok. Modul *News Reader* dan *Event Komunitas* menampilkan berita-berita dan kegiatan-kegiatan lokal yang sedang terjadi di Kota Depok.

#### 2. Perancangan dan Analisa Sistem

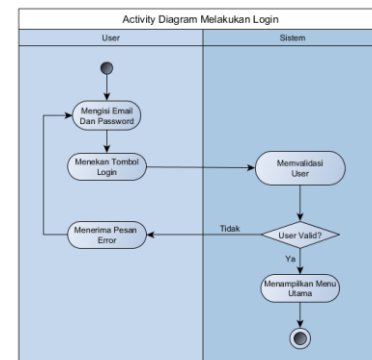
##### a. Use Case Diagram



Gambar 1 Use Case Diagram

Gambar 1 merupakan *use case diagram* dari pengembangan API aplikasi Hi-Depok Android, karena dalam aplikasi Android hanya menampilkan data-data publik saja, maka hanya terdapat satu aktor yaitu pengguna.

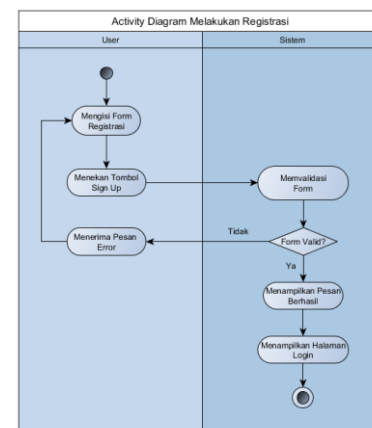
##### b. Activity Diagram Melakukan Login



Gambar 2 Activity Diagram Melakukan Login

Gambar 2 merupakan *activity diagram* dari aktivitas pengguna ketika melakukan login ke sistem, pengguna harus mengisi *form email* dan *password* yang sudah terdaftar sebelumnya. Jika data valid maka pengguna dapat masuk ke Aplikasi.

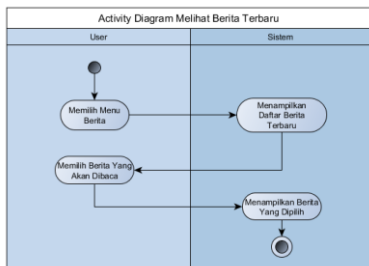
##### c. Activity Diagram Melakukan Register



Gambar 3 Activity Diagram Melakukan Register

Gambar 3 merupakan *activity diagram* dari aktivitas pengguna ketika melakukan registrasi ke sistem, pengguna harus mengisi *form* registrasi yang telah disediakan sistem dan menekan tombol *Sign Up*. Jika data pengguna yang diisi valid maka data pengguna akan terdaftar di sistem.

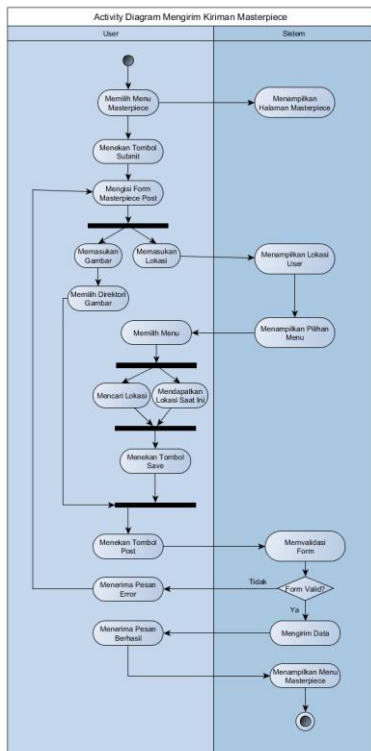
#### d. Activity Diagram Melihat News Feed



Gambar 4 Activity Diagram Melihat News Feed

Gambar 4 merupakan *activity diagram* dari halaman *news feed*. Pengguna memilih *tab News* pada aplikasi Hi-Depok Android dan sistem akan menampilkan halaman *News Feed* dan diurutkan berdasarkan paling terbaru.

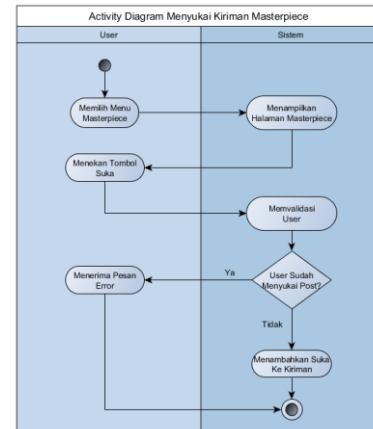
#### e. Activity Diagram Mengirim Post Masterpiece



Gambar 5 Activity Diagram Mengirim Post Masterpiece

Gambar 5 merupakan *activity diagram* dari mengirim *post masterpiece*. Pengguna memilih *tab Masterpiece* pada aplikasi Hi-Depok Android, dan menekan tombol *Submit Post*. Setelah itu pengguna mengisi *form* yang sudah disediakan seperti dapat mengunggah gambar dan lokasi terkini dari pengguna, setelah itu menekan tombol *save* untuk mengirim data ke sistem. Jika berhasil, maka kiriman pengguna akan ditampilkan di halaman *Masterpiece*.

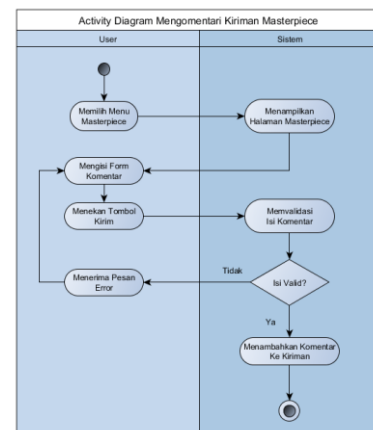
#### f. Activity Diagram Menyukai Post Masterpiece



Gambar 6 Activity Diagram Menyukai Post Masterpiece

Gambar 6 merupakan *activity diagram* dari menyukai *post masterpiece*. Pengguna memilih *tab Masterpiece* pada aplikasi Hi-Depok Android, dan menekan tombol *suka* pada *post masterpiece* dari pengguna maupun pengguna yang lain. Jika pengguna belum menekan tombol *suka*, maka jumlah *suka* pada *post* akan bertambah, namun jika pengguna sudah pernah menyukai *post* tersebut, jumlah *suka* tidak akan bertambah.

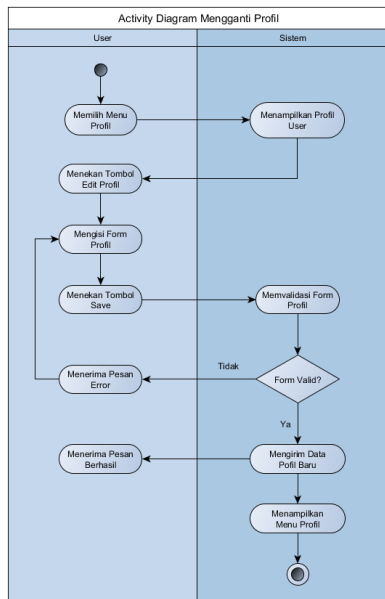
#### g. Activity Diagram Mengomentari Post Masterpiece



Gambar 7 Activity Diagram Mengomentari Post Masterpiece

Gambar 7 merupakan *activity diagram* dari mengomentari *post masterpiece*. Pengguna memilih *tab Masterpiece* pada aplikasi Hi-Depok Android, dan memilih *post masterpiece* mana yang akan dikomentari. Setelah itu pengguna mengisi *form* yang sudah disediakan untuk mengisi komentar dan menekan tombol *kirim* untuk mengirim data ke sistem. Jika berhasil, maka komentar pengguna akan ditampilkan di halaman *post Masterpiece* tersebut.

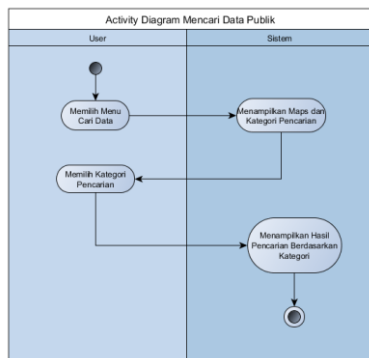
#### h. Activity Diagram Mengganti Profil



Gambar 8 Activity Diagram Mengganti Profil

Gambar 8 merupakan *activity diagram* dari mengganti profil. Pengguna memilih *tab Profile* pada aplikasi Hi-Depok Android, dan menekan tombol *edit profile*. Setelah itu pengguna mengisi *form* yang sudah disediakan untuk mengisi data mana yang akan di ubah dan menekan tombol simpan untuk mengirim data ke sistem. Jika berhasil, maka data profil pengguna akan berubah.

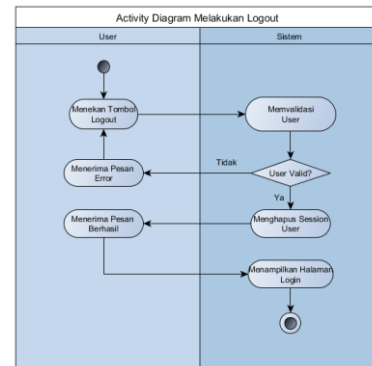
#### i. Activity Diagram Mencari Data Publik



Gambar 9 Activity Diagram Mencari Data Publik

Gambar 9 merupakan *activity diagram* dari halaman pencarian data-data publik. Pengguna dapat memilih menu Cari Data pada *tab Home* dalam aplikasi Hi-Depok Android dan sistem akan menampilkan halaman berbentuk *maps*. Pengguna dapat memilih kategori data mana yang akan ditampilkan dan sistem akan menampilkan data-data publik berdasarkan kategori yang dipilih pengguna.

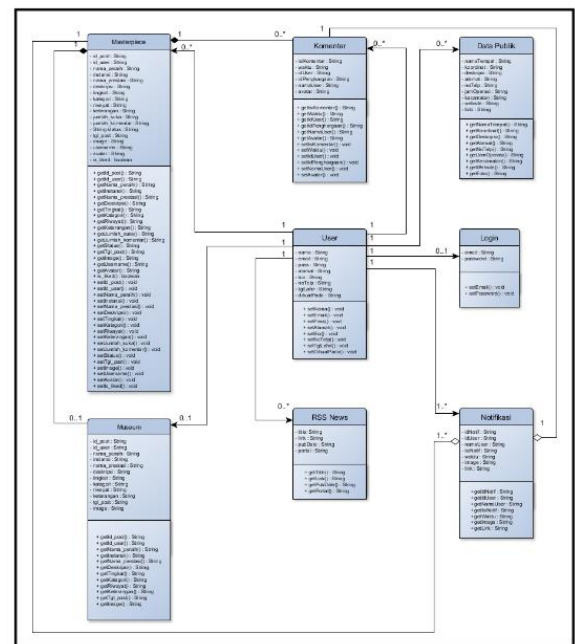
#### j. Activity Diagram Melakukan Logout



Gambar 10 Activity Diagram Melakukan Logout

Gambar 10 merupakan *activity diagram* dari melakukan *logout*. Pengguna dapat menekan tombol *logout* dimana saja dalam aplikasi Hi-Depok Android dan kemudian sistem akan memvalidasi *session* pengguna, jika berhasil sistem akan menghapus *session* yang aktif dan pengguna akan dialihkan ke halaman *login*.

#### k. Class Diagram



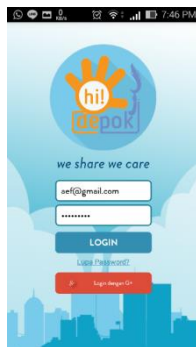
Gambar 11 Class Diagram

Gambar 11 merupakan *class diagram* pada pengembangan API aplikasi Android Hi-Depok. Pada bagian ini, *class diagram* yang dijelaskan terkait dengan hubungan antara modul-modul yang terdapat pada aplikasi Android Hi-Depok, dimana modul-modul mempunyai beberapa class yaitu *class Masterpiece*, *class Komentar*, *class Data Publik*, *class User*, *class Museum*, *class News*, *class Notifikasi*, dan *class Login*.

### 3. Implementasi

#### a. Halaman Login





Gambar 12 Implementasi Halaman *Login*

Pada halaman ini adalah halaman awal dari pengguna sebelum memasuki menu aplikasi Hi-Depok. Pengguna harus mengisi *e-mail* dan *password* yang sudah terdaftar sebelumnya lalu menekan tombol *Login*. Jika belum mendaftar, pengguna dapat menekan tombol *Login* dengan G+ untuk mendaftar melalui akun *e-mail* pengguna dan akan masuk ke halaman *Register*.

#### b. Halaman Register



Gambar 13 Implementasi Halaman *Register*

Pada halaman ini pengguna dapat melakukan registrasi untuk dapat masuk ke menu aplikasi Hi-Depok. Pengguna dapat mengisi form yang sudah disediakan oleh aplikasi seperti nama, *e-mail*, alamat, *password*, jenis kelamin, nomor telepon, tanggal lahir dan bio. Setelah itu pengguna diwajibkan untuk mencentang persetujuan untuk dan menekan tombol *Sign Up* untuk dapat melakukan registrasi. Jika data-data pengguna valid maka akan muncul *pop-up* bahwa pengguna telah sukses melakukan registrasi, dilanjutkan dengan pengguna di *redirect* ke halaman Login dan diminta untuk memasukkan *e-mail* dan *password* untuk dapat memasuki menu aplikasi.

#### c. Halaman News Feed



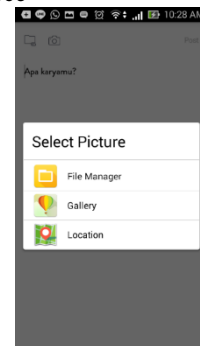
Gambar 14 Implementasi Halaman *Display News*



Gambar 15 Implementasi Halaman *Open News*

Pada halaman ini pengguna dapat melihat daftar berita-berita dari beberapa situs berita di Depok dan diurutkan berdasarkan yang terbaru, pengguna dapat menekan berita mana yang akan dibaca kemudian aplikasi akan memuat halaman berita tersebut.

#### d. Halaman Masterpiece



Gambar 16 Implementasi Halaman *Post Masterpiece*



Gambar 17 Implementasi Halaman *Display Masterpiece*



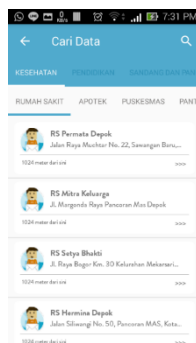
Gambar 18 Implementasi Halaman Komentar *Masterpiece*

Pada halaman ini pengguna dapat mengirim kiriman mengenai prestasi pengguna yang akan dibagikan ke pengguna lain. Selain itu pengguna juga dapat melihat kiriman pengguna yang lain serta melakukan respon dengan menekan tombol *Like* dan mengirim komentar.

#### e. Halaman Data Publik



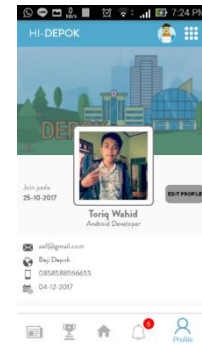
Gambar 19 Implementasi Halaman Data Publik



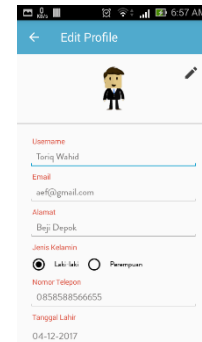
Gambar 20 Implementasi Halaman *Display* Data Publik

Pada halaman ini pengguna dapat mencari data-data publik yang terdapat di Depok seperti data Rumah Sakit, Puskesmas, Klinik, Sekolah dan sebagainya. Kemudian aplikasi akan menampilkan hasil pencarian dari pengguna berbentuk *point-point* dalam *Maps*. Selain itu, pengguna dapat mengakses *Mode List* jika pengguna ingin mencari tempat berdasarkan lokasi terdekat dari pengguna.

#### f. Halaman Profil



Gambar 21 Implementasi Halaman Profil



Gambar 22 Implementasi Halaman Ganti Profil

Pada halaman ini menampilkan tampilan profil pengguna berdasarkan data-data yang dimasukkan pada saat pendaftaran. Pengguna juga dapat melakukan perubahan data dengan menekan tombol *Edit Profile*. Kemudian pengguna akan berpindah ke halaman *Edit Profile* dan dapat mengubah data mana yang akan diubah oleh pengguna.

#### 4. Pengujian

Pengujian sistem dilakukan dengan metode *black box* untuk mengetahui apakah hasil sudah sesuai dengan kebutuhan dari *user requirement* atau belum. Target pengujian aplikasi adalah pengolahan data dan fungsi yang berjalan pada *back-end* aplikasi Hi-Depok, apakah berjalan dengan normal dan dapat terkoneksi dengan *database* pada *server*, serta dapat berjalannya fungsi *Geo-Location* untuk mengambil data lokasi terkini dari pengguna dengan menggunakan GPS pada *device* yang dipakai pengguna.

##### a. Deskripsi Pengujian

Pengujian sistem dilakukan dengan metode *black box* untuk mengetahui apakah hasil sudah sesuai dengan kebutuhan dari *user requirement* atau belum. Target pengujian aplikasi adalah pengolahan data dan fungsi yang berjalan pada *back-end* aplikasi Hi-Depok, apakah berjalan dengan normal dan dapat terkoneksi dengan *database* pada *server*, serta dapat berjalannya fungsi *Geo-Location* untuk mengambil data lokasi terkini dari pengguna dengan menggunakan GPS pada *device* yang dipakai pengguna.

##### b. Prosedur Pengujian

Pada tabel berikut ini merupakan prosedur dari pengujian yang telah dilakukan terhadap sistem aplikasi Hi-Depok dengan jenis pengujian *black box*.

Kategori Uji	Butir Uji
Registrasi <i>User</i>	Melakukan registrasi <i>user</i> dengan akun baru
<i>Sign In / Login User</i>	Melakukan <i>sign in / login user</i> untuk masuk ke aplikasi
Menu <i>News Feed</i>	Melihat daftar berita terbaru
Menu <i>Masterpiece</i>	Mengirim kiriman <i>Masterpiece</i>
	Menyukai kiriman <i>Masterpiece</i>
	Mengomentari kiriman <i>Masterpiece</i>
Menu Cari Data	Menampilkan halaman <i>maps</i> dan kategori
Menu Profil	Menampilkan data profil <i>user</i>
	Mengganti data profil <i>user</i>
<i>Sign Out / Logout</i>	Melakukan <i>sign out / logout</i> aplikasi

### c. Data Hasil Pengujian

Dari prosedur-prosedur yang telah dijalani untuk memenuhi tujuan dan target pengujian, berikut ini merupakan hasil data pengujian sistem yang dilakukan *developer Android, end-user* dan pihak terkait terhadap aplikasi Hi-Depok. Simbol ✓ menunjukkan hasil analisis sesuai dengan yang diharapkan sedangkan simbol X menunjukkan hasil analisis tidak sesuai dengan yang diharapkan.

Kasus dan Hasil Uji : Melakukan Registrasi <i>User Baru</i>				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	<i>User</i> menekan tombol “Sign Up” pada halaman <i>login</i> Hi-Depok	Menampilka n halaman registrasi <i>user</i>	Aplikasi menampilkan halaman registrasi <i>user</i>	✓
2	<i>User</i> mengisi <i>form</i> registrasi seperti <i>email</i> , nama, alamat, <i>password</i> , <i>re-password</i> , tanggal lahir, nomor	Menampilka n <i>pop-up</i> notifikasi berhasil registrasi dan kembali ke halaman <i>login</i>	Aplikasi menampilkan notifikasi berhasil registrasi dan kembali ke halaman <i>login</i>	✓

	telepon, bio dengan <i>valid</i> dan menekan tombol <i>sign up</i>			
3	<i>User</i> tidak mengisi <i>form</i> registrasi dengan lengkap (ada <i>form</i> yang kosong)	Menampil kan pesan <i>error</i>	Aplikasi menampilkan pesan “semua <i>form</i> harus diisi”	✓
Kasus dan Hasil Uji : Melakukan <i>Login User</i>				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	<i>User</i> mengisi data <i>email</i> dan <i>password</i> pada <i>form login</i> dengan data akun yang benar	Dapat masuk ke aplikasi dan menampilkan halaman utama	Aplikasi menampilkan halaman utama	✓
2	<i>User</i> mengisi data <i>email</i> dan <i>password</i> pada <i>form login</i> dengan data akun tidak benar atau tidak lengkap	Menampilka n pesan <i>error</i>	Aplikasi menampilkan pesan “ <i>error! Login credentials is wrong. Try again!</i> ”	✓
Kasus dan Hasil Uji : Melihat Daftar Berita Terbaru				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	<i>User</i> memilih tab <i>News</i> pada menu aplikasi	Menampil kan daftar berita dari berbagai portal diurutkan berdasarkan yang	Aplikasi menampilkan berita dari berbagai portal diurutkan berdasarkan	✓



		paling terbaru	yang paling terbaru	
2	User memilih berita salah satu berita pada daftar	Memuat <i>link</i> berita dan memuat halaman berita	Aplikasi memuat <i>link</i> berita dan memuat halaman berita	√
Kasus dan Hasil Uji : Mengirim Kiriman <i>Masterpiece</i>				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	User mengisi <i>form</i> dengan teks dan memilih gambar, kemudian menekan tombol <i>post</i>	Menampilk an pesan berhasil dan memuat halaman <i>Masterpiec e</i>	Aplikasi menampilkan pesan “ <i>Post Added Successfully!</i> ”	√
2	User mengoson gkan <i>form</i> , kemudian menekan tombol <i>post</i>	Menampilka n pesan <i>error</i>	Aplikasi menampilkan pesan “ <i>error!</i> ”	√
Kasus dan Hasil Uji : Menyukai Kiriman <i>Masterpiece</i>				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	User menekan <i>icon</i> suka pada kiriman <i>Masterpiec e</i>	Menampilk an pesan berhasil menyukai kiriman dan jumlah suka bertambah	Aplikasi menampilkan n “Anda menyukai kiriman ini”	√
2	User menekan lagi <i>icon</i> suka pada kiriman <i>Masterpiec e</i>	Menampilka n pesan <i>error</i> dan jumlah suka tidak bertambah	Aplikasi menampilkan pesan “Anda sudah menyukai kiriman ini!”	√
Kasus dan Hasil Uji : Mengomentari Kiriman <i>Masterpiece</i>				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian

1	User mengisi <i>form</i> komentar pada kiriman <i>Masterpie ce</i> , lalu menekan <i>icon</i> kirim	Menampilk an pesan berhasil dan menampilkan daftar komentar	Aplikasi menampilkan pesan “Berhasil Menambahk an Kiriman” dan menampilkan daftar komentar	√
2	User mengoson gkan <i>form</i> komentar pada kiriman <i>Masterpie ce</i> , lalu menekan <i>icon</i> kirim	Menampilka n pesan <i>error</i>	Aplikasi menampilkan pesan “Berhasil Menambahka n Kiriman” dan menampilkan daftar komentar	X
Kasus dan Hasil Uji : Menampilkan Halaman <i>Maps</i> dan Kategori				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	User memilih tab <i>Home</i> pada menu aplikasi	Menampilk an halaman <i>maps</i> kota Depok	Aplikasi menampilkan halaman <i>maps</i> kota Depok	√
2	User menekan <i>icon</i> kategori pada <i>maps</i>	Menampilka n <i>pop-up</i> kategori data-data publik	Aplikasi menampilkan <i>pop-up</i> kategori data-data publik	√
3	User memilih salah satu kategori data pada <i>pop-up</i>	Menampilk an marker data-data pada halaman <i>maps</i> kota Depok	Aplikasi menampilkan marker data-data pada halaman <i>maps</i> kota Depok	√
Kasus dan Hasil Uji : Mengganti Profil <i>User</i>				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	User menekan tombol <i>Edit Profile</i> pada menu profil	Menampilk an halaman <i>edit profile</i>	Aplikasi menampilkan halaman <i>edit profile</i>	√

2	User mengisi form data user seperti <i>email</i> , nama, alamat, tanggal lahir, nomor telepon, bio dengan <i>valid</i> dan menekan tombol <i>edit profile</i>	Menampilkan pesan berhasil	Aplikasi menampilkan pesan “Berhasil Mengubah Profil” dan menampilkan halaman profil	√
3	User tidak mengisi form registrasi dengan lengkap (ada form yang kosong)	Menampilkan pesan error	Aplikasi menampilkan pesan “Form harus diisi!”	√
Kasus dan Hasil Uji : Melakukan Logout Aplikasi				
No	Skenario Pengujian	Yang Diharapkan	Pengamatan	Hasil Pengujian
1	User menekan tombol <i>logout</i> dalam aplikasi	Menampilkan pesan berhasil dan kembali ke halaman <i>login</i>	Aplikasi menampilkan pesan “Berhasil Keluar” dan menampilkan halaman <i>Login</i>	√

#### d. Analisis Data / Evaluasi

Setelah dilakukan pengujian terhadap aplikasi Android Hi-Depok dapat diketahui bahwa semua fungsi dan fitur aplikasi dapat berjalan dengan baik dan mendapat respons positif dari para pengguna dari aplikasi yang menjadi *tester*. Seluruh pengolahan data pada aplikasi yang meliputi koneksi input data yang dilakukan oleh *user* pada saat melakukan registrasi, melakukan *login*, mengirim kiriman dan sebagainya dapat terkoneksi dengan *database* dengan baik dan tanpa ada kendala. Berikut di bawah ini merupakan tabel hasil analisis yang telah dilakukan, simbol √ menunjukkan hasil analisis sesuai dengan yang diharapkan sedangkan simbol X menunjukkan hasil analisis tidak sesuai dengan yang diharapkan.

1	Registrasi User	√	Sesuai dengan yang diharapkan
2	Login User	√	Sesuai dengan yang diharapkan
3	Menu News Feed	√	Sesuai dengan yang diharapkan
4	Menu Masterpiece	√	Sesuai dengan yang diharapkan
5	Menu Cari Data	√	Sesuai dengan yang diharapkan
6	Menu Profile	√	Sesuai dengan yang diharapkan
7	Logout User	√	Sesuai dengan yang diharapkan

#### IV. KESIMPULAN

Berdasarkan hasil dan pembahasan di atas maka dapat disimpulkan bahwa:

1. Bisnis proses aplikasi mengalami perubahan, terdapat beberapa penambahan modul dan fungsi seperti fungsi pada modul *Masterpiece*, fungsi *display News* dan *Event*, fungsi Cari Data dan lainnya.
2. Dengan adanya halaman *maps* dapat membantu *user* dalam mencari informasi berdasarkan koordinat tempat yang ingin dituju.
3. Setelah dilakukan testing dengan menggunakan metode *blackbox*, aplikasi dinyatakan sudah layak untuk dapat digunakan oleh *end-user* karena fungsinya sudah berjalan dengan baik.

#### DAFTAR PUSTAKA

- [1] Daily Social. 2015. *Indonesia Startup Report 2015*. Daily Social Indonesia. <https://dailysocial.id/report/post/ds-tech-startup-report-2015>. [30 November 2017].
- [2] Cohen, Boyd. 2013. *Boyd Cohen: The Smart City Wheel*. Smart City Event. <http://www.smart-circle.org/smartcity/blog/boyd-cohen-the-smart-city-wheel/>. [29 November 2017].
- [3] Zakas, N. C. 2015. *JSON, in Professional Javascript® for Web Developers*. Indianapolis: Wiley Publishing, Inc.
- [4] Android Studio. *Meet Android Studio*. (n.d.). <https://developer.android.com/studio/intro/index.html>. [29 November 2017].
- [5] Kadir, Abdul. 2014. *Buku Pertama Belajar Pemrograman Java untuk Pemula*. Yogyakarta: Mediakom.
- [6] Pressman, Roger S., and Maxim, Bruce. 2015. *Software Engineering: A Practitioner's Approach 8th Edition*. New York: McGraw-Hill Higher Education.
- [7] Rosa. A.S., and Shalahuddin, M. 2014. *Rekayasa Perangkat Lunak 2nd ed*. Bandung: Informatika.