

Rancang Bangun Aplikasi Penitipan Hewan Peliharaan Berbasis Android

<http://dx.doi.org/10.28932/jutisi.v4i3.830>

Evan Tanuwijaya #1

Teknik Informatika, Universitas Ciputra
UC Town, Citraland, Surabaya

¹ etanuwijaya96@gmail.com

Abstract — A lot of people in Indonesia keeps their pets at their home. Therefore, the pets must be taken care at all time. But we as humans have many other activities besides taking care of our pets. This causes the pet not to be monitored. To solve the problem, an app is required to search for a pet daycare. This app can search for healthy, comfortable, and safe pet daycare. Applications have been widely developed in the community, especially for mobile applications. Mobile applications have many advantages and ease in accessing them. In addition, in Indonesia technology is growing very fast. Many Indonesians are also using smartphones in their daily lives. Based on the above data, a solution appears to create an app for pet daycare where pet owners can easily search for pet daycare just with their smartphones. This app uses a client-server architecture where the client is a smartphone that has an Android operating system and a server using Google Firebase. Development method used is the Systems Development Life Cycle (SDLC) method with the Waterfall model. The result of this thesis, it can be concluded that Petqper mobile application can look for healthy, convenient, and safe daycare. In addition, reliable pet daycare can use this application to offer their services and simplify the process of booking a place.

Keywords— Pet Daycare, Android Application, Firebase

I. PENDAHULUAN

Di Indonesia, survei pada World Society for the Protection of Animal (WSPA) pada tahun 2007 menunjukkan jumlah populasi hewan peliharaan dengan jenis anjing sebesar 8 juta dan populasi jenis kucing sebesar 15 juta. Perkembangan dari populasi anjing selama kurang lebih 5 tahun meningkat sebesar 22% (peringkat 9 dari 58 negara) dan pada populasi kucing bertambah sebesar 66% (peringkat 2 dari 58 negara [1]. Seiring berjalannya waktu, memiliki hewan peliharaan bukan lagi menjadi hobi memelihara saja, tetapi telah banyak berkembang sebagai hewan untuk dilombakan bahkan dipercantik layaknya manusia. Oleh sebab itu, usaha toko hewan peliharaan atau layanan terhadap hewan peliharaan mulai dipandang sebagai jenis usaha yang menjanjikan [2]. Tidak hanya terbatas pada toko-toko secara fisik, sekarang telah banyak sekali orang-

orang berjualan barang-barang hewan peliharaan di sosial media maupun toko online.

Kebutuhan akan perawatan hewan peliharaan menjadi salah satu hal yang penting sehingga pemilik harus meluangkan banyak waktu untuk mengurus hewan peliharaan mereka. Akan tetapi pemilik hewan sebagai anggota masyarakat juga memiliki banyak pekerjaan yang harus diselesaikan. Hal ini mengurangi waktu pemilik untuk merawat hewan dan menyebabkan hewan peliharaan tidak terurus dengan baik [3]. Selain bekerja, pemilik hewan yang hendak berpergian baik itu dekat ataupun jauh pasti tidak selalu membawa hewan peliharaan mereka. Hewan peliharaan yang tidak terawat secara teratur dapat mengakibatkan hewan peliharaan tersebut menjadi lemas, sakit, dan bahkan mati. Agar hewan peliharaan tersebut dapat terawat secara teratur saat pemilik nya tidak ada, pemilik biasa menitipkan kepada tempat penitipan hewan sehingga hewan peliharaan mereka dapat terjaga dengan baik.

Dengan adanya perkembangan teknologi yang pesat, peneliti dapat membuat aplikasi untuk mencari tempat penitipan hewan sebagai solusinya. Pembuatan aplikasi untuk mencari tempat penitipan hewan peliharaan berbasis Android lebih menguntungkan dari pada pembuatan aplikasi IOS karena Android sendiri memiliki banyak pengguna sehingga memiliki pasar yang sangat besar dan lebih menguntungkan [4]. Akan tetapi pembuatan aplikasi Android untuk penitipan hewan ini memiliki kelebihan dan kekurangannya masing-masing. Dengan membuat aplikasi pencarian tempat penitipan hewan berbasis Android, pembuatan aplikasi ini dapat dilakukan dengan cara *Native* yang dapat menggunakan semua fitur dari suatu *smartphone* seperti notifikasi, kalender, dan lain-lain sekaligus penggunaan memori yang lebih ringan [4].

Pada pembuatan aplikasi untuk mencari penitipan hewan yang berbasis Android ini, pemilik hewan peliharaan tidak perlu kesusahan dalam mencari tempat penitipan hewan. Dan tempat penitipan hewan dapat menggunakan aplikasi ini sebagai tempat untuk menawarkan jasa mereka pada pemilik hewan.

A. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas, maka dapat dirumuskan masalah sebagai berikut Bagaimana merancang dan membangun aplikasi untuk mencari tempat penitipan hewan berbasis Android?

B. Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan membangun aplikasi untuk tempat penitipan hewan berbasis Android.

C. Metode Penelitian

Metodologi pelaksanaan penelitian ini akan dilaksanakan berdasarkan metode *Systems Development Life Cycle* (SDLC). Model SDLC yang digunakan adalah model Waterfall dengan tahapan sebagai berikut:

1) Perencanaan Sistem (*Systems Planning*): Untuk perencanaan pembangunan aplikasi untuk mencari tempat penitipan hewan, peneliti mencari informasi mengenai pembuatan aplikasi *mobile* menggunakan Android Studio dan Firebase

2) Analisis Sistem (*Systems Analysis*): Tahap ini, peneliti mengumpulkan dan menganalisa kebutuhan sistem yang akan digunakan.

3) Perancangan Sistem (*Systems Design*): Peneliti merancang sistem untuk pembuatan aplikasi dengan menggunakan arsitektur *client-server*, dimana pada sisi *client* menggunakan smartphone yang mempunyai sistem operasi Android dan pada sisi server menggunakan Firebase.

4) Implementasi Sistem (*Systems Implementation*): Pada tahap ini, hasil perancangan sistem akan diimplementasikan ke dalam aplikasi.

5) Mencoba Sistem (*System Testing*): Pada tahap ini, pengguna akan mencoba aplikasi sehingga dapat menemukan kekurangan ataupun kesalahan yang terjadi saat pengoperasian aplikasi sehingga sistem dapat diperbaiki.

6) Pengelolaan Sistem (*System Maintenance*): Hasil dari mencoba sistem dan masukan-masukan dari pengguna akan dibuat perbaikan dan saran pengembangan aplikasi.

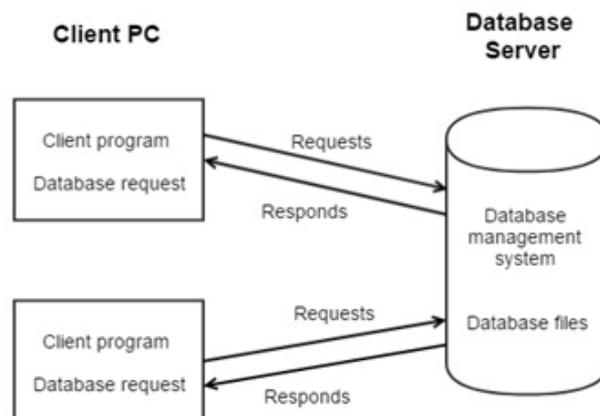
II. LANDASAN TEORI

A. Merawat Hewan Peliharaan

Dalam memelihara hewan peliharaan, seseorang harus dapat bertanggung jawab terhadap keberlangsungan hidup dan kesejahteraan (*animal welfare*) hewan tersebut. Hal yang paling utama adalah pemilik hewan harus dapat memberikan perhatian dan kasih sayang karena hewan mampu memproses kognitif dan perasaan. Pemilik hewan harus meluangkan waktunya untuk memandikan, mengajak bermain atau berjalan-jalan, dan konsultasi kesehatan. Oleh sebab itu, pemilik harus memiliki rasa tanggung jawab dengan menjaga kondisi hewan peliharaan tersebut agar tetap sehat secara fisik dan mental serta tidak kekurangan makanan, minuman, dan tempat berteduh [5].

B. Arsitektur Client-Server

Arsitektur *client-server* dapat meminimalkan waktu dalam pembuatan aplikasi dengan cara membagi fungsi pertukaran informasi kedalam dua bagian yaitu client dan server. Pada bagian client akan meminta data sementara bagaian sever akan merespon permintaan dari sisi client. Pada arsitektur *client-server*, pemrosesan data dilakukan oleh server, dan hasil dari pemrosesan tersebut akan dikirimkan pada client yang meminta, sehingga dapat mempercepat performa kerja dari sistem ini. Standar protokol yang digunakan antara client dan server untuk saling berkomunikasi adalah *Hypertext Transfer Protocol Secure* (HTTPS). Dalam penelitian ini, akan digunakan arsitektur *client-server* 2 tier (seperti Gambar 1), dimana arsitektur ini hanya terdapat basis data pada bagian server dan smartphone sebagai client [6].



Gambar 1. Arsitektur 2 tier *client-server*

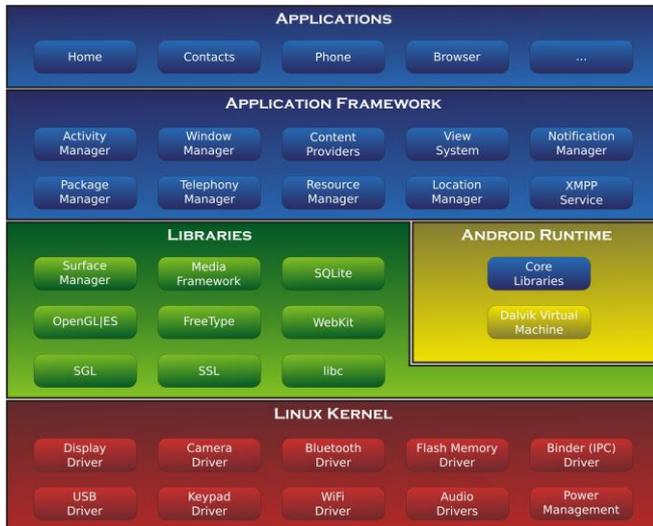
C. Aplikasi Native dan Hybrid

Dalam pengembangan aplikasi mobile saat ini, bergantung pada sistem operasi yang digunakan. Aplikasi yang dibuat spesifik untuk sebuah platform dengan *Software Development Kit* (SDK), aplikasi, dan bahasa pemrograman yang disediakan oleh vendor sistem operasi tersebut disebut *Native Application*. Pengembangan aplikasi *mobile* dengan metode *native* ini sangat baik untuk performa kerja aplikasi terhadap smartphone karena dapat sepenuhnya memanfaatkan ketersediaan sumber daya sistem mobile dengan bantuan SDK dan API yang telah disediakan. Selain pengembangan secara *native*, terdapat metode lain yaitu *hybrid* yang pengembangan aplikasinya dapat dijalankan pada semua platform. Metode *hybrid* ini dikembangkan dengan teknologi web seperti HTML5, CSS, dan JavaScript yang dijalankan pada penampung aplikasi *native*. Tetapi metode ini memiliki batasan yaitu rendahnya akses pada seluruh fitur perangkat mobile. Jika dibandingkan dengan metode *Native*, pengembangan dengan metode *hybrid* menggunakan HTML5, CSS, dan JavaScript masih bergantung pada browser [7] sehingga relatif lebih berarti

dibandingkan dengan metode native dikarenakan metode *hybrid* mirip dengan aplikasi web[8].

D. Android dan Android Studio

Sistem operasi Android mulai dipasarkan pada November 2007 yang dikenalkan oleh Google dan Open Handset Alliance (OHA). Sistem operasi Android bekerja berdasarkan sistem Linux, dimana sistem operasi Android merupakan kumpulan komponen perangkat lunak yang berbentuk stack yang dibagi kedalam 5 bagian dan 4 layer utama [9] yang digambarkan seperti Gambar 2.



Gambar 2. Arsitektur Android

Dalam penelitian ini, peneliti menggunakan Android Studio untuk membangun aplikasi. Android Studio adalah *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android. Selain merupakan kode editor dan alat pengembang aplikasi Android, Android Studio menawarkan fitur lebih banyak untuk membuat aplikasi Android, misalnya:

1. Gradle yang fleksibel
2. emulator yang cepat dan memiliki banyak fitur
3. environment untuk pengembangan semua perangkat Android
4. instant Run untuk menjalankan aplikasi tanpa perlu membuat apk baru
5. terdapat beberapa template kode dan integrasi dengan GitHub
6. memiliki alat pengujian dan kerangka kerja yang ekstensif
7. terdapat beberapa bantuan untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. dukungan C++ dan NDK
9. terdapat Google Cloud Platform, untuk mempermudah integrasi dengan Google Cloud Messaging dan App Engine [10].

E. Google Firebase

Firebase adalah sebuah *backend* basis data untuk Android, iOS, dan web aplikasi. Google menyediakan Firebase API untuk membuat basis data dan menarik data dari aplikasi secara realtime. Data yang akan disimpan pada Firebase akan ditulis dalam bentuk JSON dan dapat diakses pada semua platform [11]. Basis data pada firebase memiliki kemampuan realtime menggunakan *data synchronization*. Sehingga jika terdapat perubahan data, semua perangkat yang terhubung akan langsung menerima perubahan dalam waktu mili detik. Selain itu, penggunaan Firebase dapat langsung mengakses basis data secara langsung tanpa adanya server aplikasi [12].

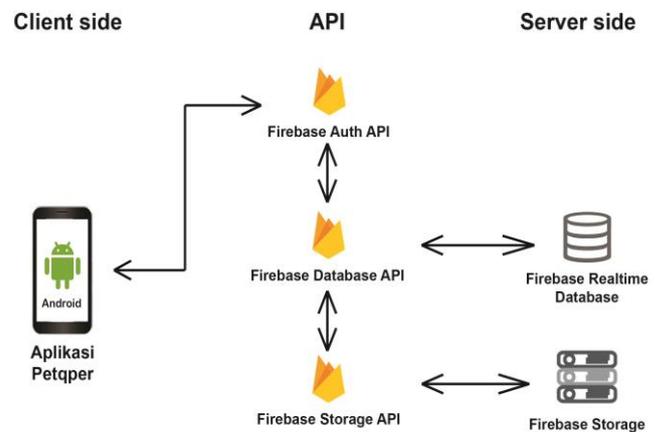
III. DESAIN SISTEM

A. Desain Arsitektur

Aplikasi Petqper memiliki desain arsitektur yaitu client server. Pada arsitektur client server ini, aplikasi yang dimiliki oleh pengguna yang disebut client dapat mengakses ke server Firebase secara bersamaan. Sistem arsitektur ini dapat dilihat dari dua sisi yaitu:

- Sisi client merupakan aplikasi yang dapat dipasang pada smartphone Android yang dapat digunakan oleh pengguna aplikasi baik itu pemilik hewan yang hendak mencari penitipan hewan ataupun bagi tempat penitipan hewan yang akan menawarkan jasanya atau baru saja membuka tempat penitipan hewan. Seluruh kegiatan yang dilakukan akan dijalankan pada sisi client ini.

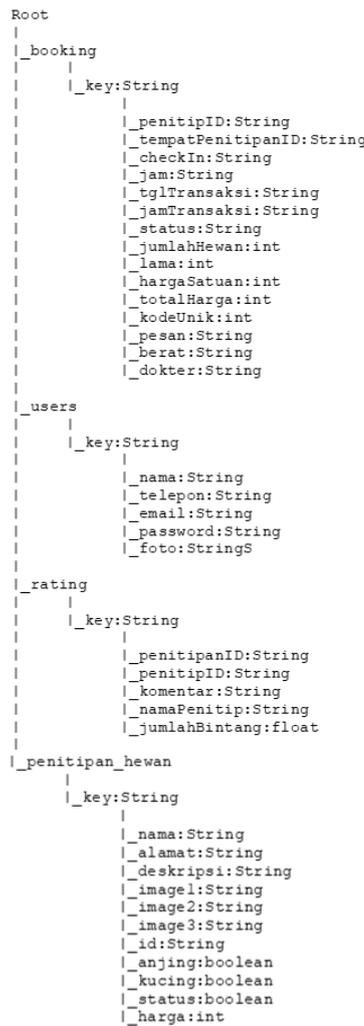
- Sisi server adalah Firebase yang dimiliki oleh Google merupakan tempat penyimpanan baik itu database dan file-file lain yang diupload dari sisi client. Sebelum sisi client mengakses sever, client akan melewati API yang telah disediakan oleh Firebase untuk melakukan interaksi baik itu menyimpan atau menarik data pada server Firebase tersebut. Gambar 3 adalah gambar desain arsitektur dari aplikasi Petqper yang dirancang.



Gambar 3 Desain Sistem Arsitektur

B. Desain Basis Data

Dalam pembuatan aplikasi Petqper, database yang digunakan adalah Firebase. Keuntungan dari Firebase sendiri adalah fitur *Realtime syncing* dimana semua data yang disimpan dapat disinkronisasikan antar pengguna secara langsung. Firebase ini menggunakan skema NoSQL untuk menyimpan segala data dari aplikasi client. Karena menggunakan NoSQL maka model database pada Firebase tidak mempunyai table-table yang memiliki relasi satu dengan yang lain layaknya SQL. Peneliti menggunakan NoSQL karena tingkat kompleksitas yang rendah untuk proyek ini. Database Firebase menggunakan model tree dimana parent memiliki Key Value yang unik sehingga performa dari database ini semakin cepat dan dapat diakses oleh aplikasi client secara realtime.

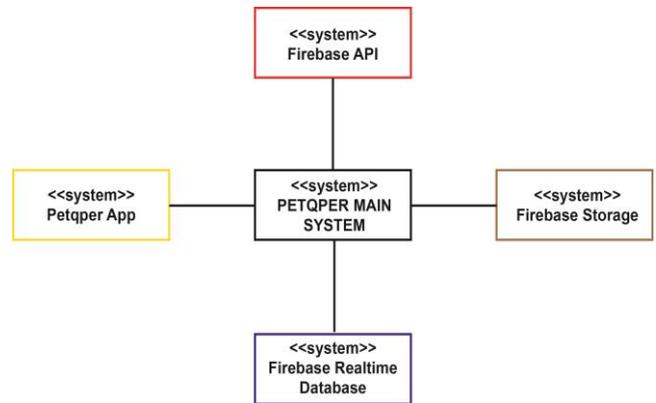


Gambar 4 Desain Database

C. Context Diagram

Aplikasi Petqper yang dibangun dengan arsitektur client server memiliki beberapa sistem yang saling

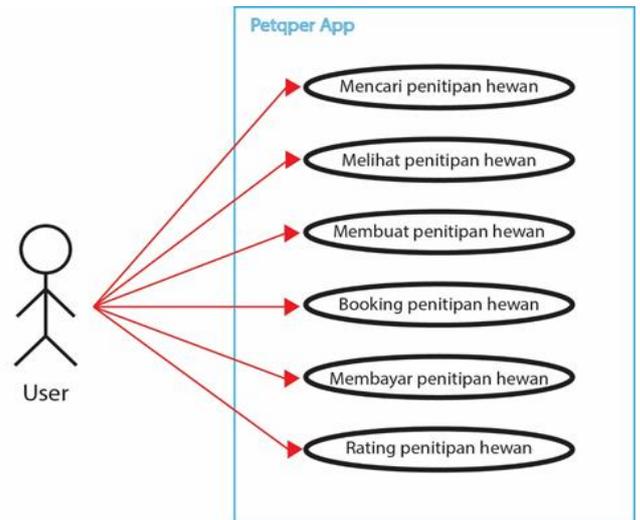
terhubung agar dapat berjalan. Dengan menggunakan *context diagram*, peneliti ingin menggambarkan besar sistem yang dirancang dimana aplikasi ini memiliki beberapa *context* dengan batasan-batasan tertentu. Petqper ini memiliki empat *context* sistem yang saling terhubung. Keempat sistem tersebut adalah aplikasi client yang dijalankan pada smartphone Android, Firebase API, Firebase Realtime Database, dan Firebase Storage. Digambarkan pada gambar 5 *context diagram* dari aplikasi Petqper dimana jika salah satu *context* tersebut dihilangkan maka aplikasi tidak dapat berjalan dengan baik.



Gambar 5 Context Diagram

D. Use Case Diagram

Dalam mengoperasikan aplikasi Petqper, ada beberapa aksi yang dapat dilakukan oleh pengguna. Pengguna aplikasi ini bertujuan untuk menitipkan hewan atau membuka penitipan hewan. Setiap pengguna tersebut dapat melakukan aksi-aksi yang digambarkan pada gambar 6.



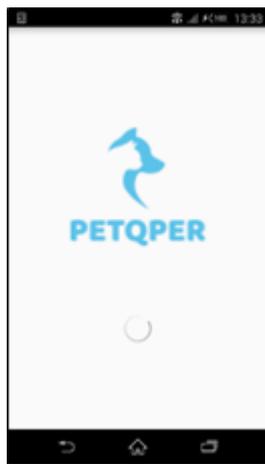
Gambar 6 Use Case Diagram

IV. IMPLEMENTASI SISTEM

A. Client-Server

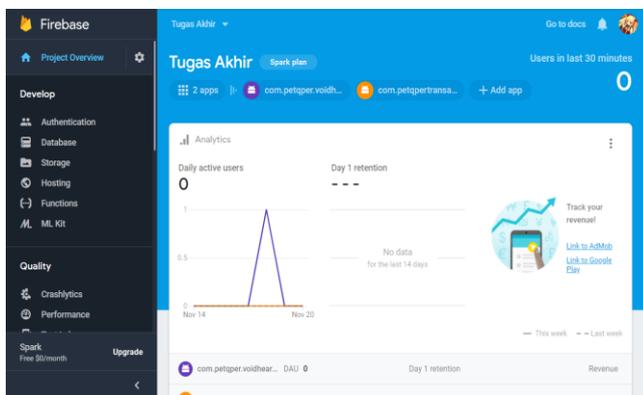
Arsitektur yang digunakan adalah arsitektur client-server. Model arsitektur ini dapat digunakan oleh banyak pengguna yang dapat mengakses server secara bersamaan dan memakai resource yang ada pada basis data maupun tempat penyimpanan pada server. Berikut ini adalah hasil penerapan arsitektur client-server.

1) Client: Penerapan pada sisi client berupa aplikasi Android yang dibuat menggunakan Android Studio. Hasil dari pembuat dari Android Studio akan dibangun menjadi aplikasi yang memiliki format APK. Pengguna aplikasi ini dapat memasang APK tersebut pada smartphone Android mereka sehingga mereka dapat langsung menggunakan aplikasi tersebut. Aplikasi tersebut dapat mengakses basis data yang telah diterapkan.



Gambar 7 Arsitektur sisi client

2) Server: Pada sisi server terdapat API yang membuat koneksi pada server. Server yang digunakan adalah Firebase yang memiliki fitur *database real time* dimana aplikasi client dapat mengakses database tersebut dan melakukan mengubah data mereka secara langsung.



Gambar 8 Arsitektur sisi server

B. Basis Data

Pada implementasi basis data di Firebase Realtime Database, digunakan JSON Tree untuk menggambarkan basis data aplikasi Petqper. Tree ini memiliki root yang merupakan pusat dan pertama kali di akses. Kemudian terdapat node-node yang ditaruh didalam root tersebut. Node tersebut merupakan child dari root tersebut. Node yang memiliki child akan disebut dengan parent. Setiap node memiliki key dan value. Key pada setiap node dapat dibuat secara otomatis ataupun ditentukan dan key tersebut memiliki sifat unik. Pada aplikasi Petqper ini, memiliki empat buah node utama yang merupakan child dari root. Empat buah node itu adalah node booking, node users, node penitipan_hewan, dan node rating.

1) *Node booking*: Pada node ini menyimpan semua booking yang dilakukan oleh pemilik hewan dengan penitipan hewan. Pada node booking terdapat child yang memiliki key berbentuk String unik untuk setiap booking yang dilakukan oleh pemilik hewan peliharaan. Dalam key tersebut terdapat value yang dijelaskan pada Tabel I.

TABEL I
KETERANGAN NODE BOOKING

No	Nama	Tipe Data	Keterangan
1	penitipID	String	ID dari pengguna yang menitipkan hewan
2	tempatPenitipanID	String	ID dari penitipan hewan
3	checkIn	String	Tanggal check in
4	jam	String	Jam mulai menitipkan
5	tglTransaksi	String	Tanggal dilakukan transaksi
6	jamTransaksi	String	Jam dilakukan transaksi
7	status	String	Status dari booking
8	jumlahHewan	Integer	Jumlah hewan yang dititipkan
9	Lama	Integer	Lama menitipkan
10	hargaSatuan	Integer	Harga perjam perhewan
11	totalHarga	Integer	Total yang harus dibayarkan
12	kodeUnik	Integer	Kode pembayaran

2) *Node users*: Node ini menyimpan seluruh data-data user. Id dari setiap user didapatkan dari Firebase

Authentication. Keterangan lengkap node dapat dilihat pada Tabel II.

TABEL II
KETERANGAN NODE USERS

No	Nama	Tipe Data	Keterangan
1	nama	String	Nama pengguna
2	telepon	String	Telepon pengguna
3	email	String	Email pengguna
4	password	String	Password pengguna
5	foto	String	Foto pengguna

3) *Node penitipan*: Node *penitipan_hewan* ini menyimpan data pengguna yang ingin membuka tempat penitipan hewan. Setiap data yang dimasukkan akan disimpan di node ini. Keterangan lengkap node ini dijelaskan pada Tabel III.

TABEL III
KETERANGAN NODE PENITIPAN_HEWAN

No	Nama	Tipe Data	Keterangan
1	nama	String	Nama penitipan hewan
2	alamat	String	Alamat penitipan hewan
3	deskripsi	String	Deskripsi penitipan hewan
4	image1	String	Gambar 1
5	image2	String	Gambar 2
6	image3	String	Gambar 3
7	id	String	Foto identitas
8	anjing	Boolean	Menerima anjing
9	kucing	Boolean	Menerima kucing
10	status	Boolean	Terbitkan penitipan hewan
11	harga	Integer	Harga perhewan perjam

4) *Node rating*: Node rating berisikan semua rating dan komentar dari pemilik hewan yang telah menitipkan hewannya pada penitipan hewan. Keterangan detail tentang isi node akan dijabarkan pada Tabel IV.

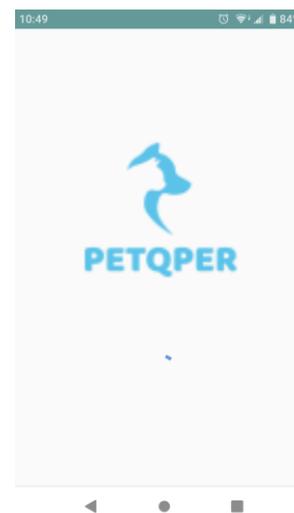
TABEL IV
KETERANGAN NODE RATING

No	Nama	Tipe Data	Keterangan
1	penitipanID	String	Penitipan ID yang di rating
2	penitipID	String	Penitip ID yang memberi rating
3	Komentar	String	Komentar yang diberikan
4	namaPenitip	String	Nama penitip
5	jumlahBinatang	String	Jumlah rating yang diberikan

C. User Interface

Dalam pembuatan tatap muka aplikasi, peneliti menggunakan library standart dari Google material desain. Untuk penggunaan warna, peneliti menggunakan warna biru muda sesuai dengan logo Petqper. Pengimplementasian tatap muka antara lain sebagai berikut.

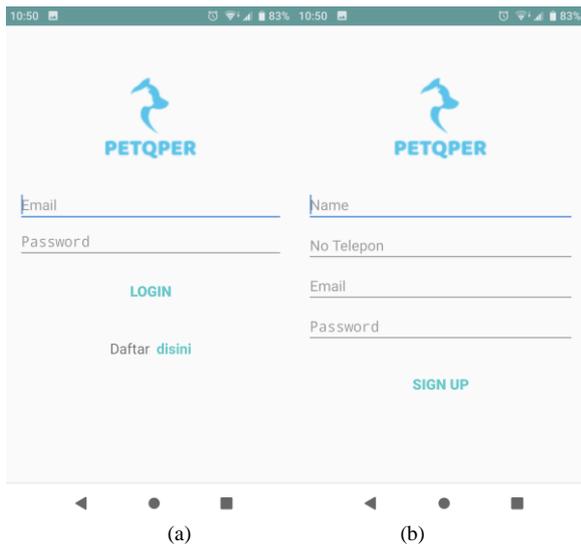
1) *Halaman Splash*: Halaman ini merupakan halaman yang akan dijalankan pertama kali saat apliksi dijalankan. Berguna untuk menginisialisasi beberapa data yang dapat disiapkan terlebih dahulu. Tampilan dapat dilihat pada gambar 9



Gambar 9 Implementasi Halaman Splash

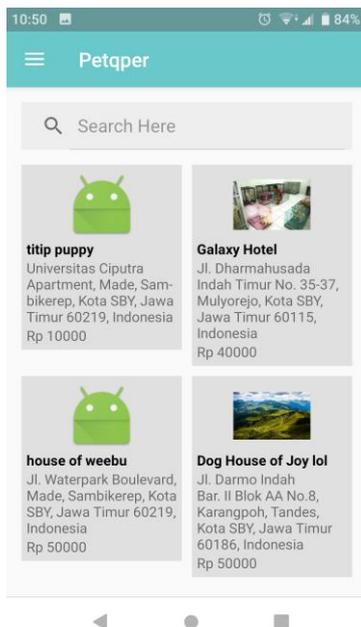
2) *Halaman Login dan Register*: Halaman Login merupakan halaman dimana pengguna memasukkan email dan password yang telah terdaftar. Jika pengguna tersebut merupakan pengguna baru, maka dapat menekan tombol daftar disini dan akan diarahkan ke

halaman registrasi untuk mengisi data nama, alamat, email, dan password. Tampilan seperti gambar 10.



Gambar 10 (a) Implementasi halaman login, (b) halaman registrasi

3) *Halaman Kumpulan Penitipan Hewan:* Pada halaman ini ditampilkan seluruh penitipan hewan. Terdapat bar pencarian, daftar penitipan hewan yang berisikan nama, alamat, no telepon, dan foto. Pada bagian action bar terdapat menu drawer untuk memindah halaman ke halaman lain. Tampilan sesuai gambar 11



Gambar 11 Implementasi Kumpulan Halaman Penitipan Hewan

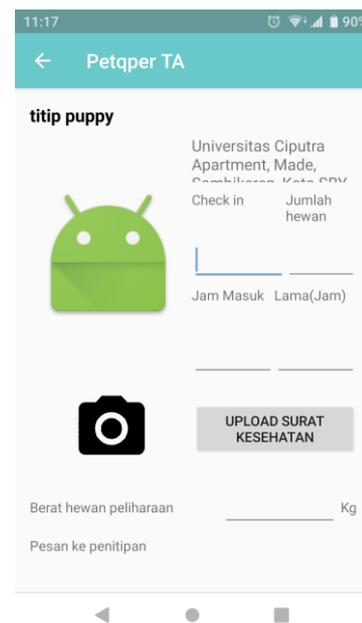
4) *Halaman Detail Penitipan Hewan:* Halaman ini menampilkan data detail mengenai tempat penitipan hewan yang dipilih. Terdapat foto, data detail, rating, komentar, dan pemilik penitipan tersebut. Terdapat tiga tombol

dibawah untuk mengirim SMS, telepon, dan booking tempat penitipan hewan tersebut. Tampilan pada gambar 12



Gambar 12 Implementasi Halaman Detail Penitipan Hewan

5) *Halaman Booking:* Pada halaman ini, pengguna mengisi tanggal *check in*, jumlah hewan, jam, dan lama menitipkan. Kemudian pada tampilan *summary* akan dihitung total yang harus dibayarkan kepada penitipan hewan. Tampilan sesuai gambar 13.



Gambar 13 Implementasi Halaman Booking

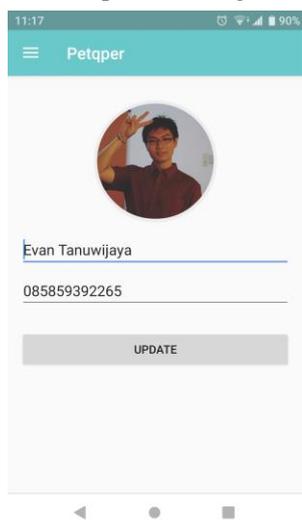
6) *Halaman Buat Penitipan Hewan:* Halaman ini untuk membuat tempat penitipan hewan. Dengan mengisi data

lengkap beserta foto-foto dan tanda pengenal. Pada bagian atas terdapat switch untuk menampilkan pos pada home atau tidak. Tampilan pada gambar 14.



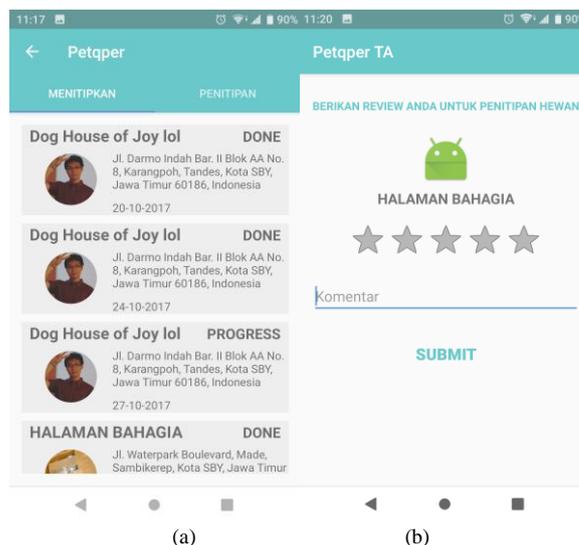
Gambar 14 Implementasi Halaman Buat Penitipan Hewan

7) *Halaman Profil:* Halaman ini ditampilkan untuk meng-update data dari pengguna seperti nama, nomor telepon, dan foto. Setelah itu tombol *update* untuk mengubah database. Tampilan sesuai gambar 15.



Gambar 15 Implementasi Halaman Profil

8) *Halaman History dan Rating:* Halaman history muncul untuk mencatat semua transaksi yang ada. Disini memiliki dua tab bagi penitipan hewan dan penitip hewan. Untuk halaman rating muncul ketika booking tersebut telah selesai dijalankan. Tampilan sesuai gambar 16.



Gambar 16 Implementasi Halaman (a) History (b) Rating

V. HASIL PENGUJIAN

Pada tahap pengujian, akan diberi indikator dan 30 orang akan mencoba aplikasi ini untuk mengetahui apakah aplikasi Petqper ini dapat menyelesaikan kondisi tertentu yang dilakukan oleh pengguna. Peneliti menggunakan metode *Usability testing* yang merupakan alat umum untuk mengevaluasi kemudahan *interface* dari aplikasi Petqper ini. Cara validasi aplikasi adalah tujuh indikator skenario akan dijalankan oleh pengguna dimana pengguna akan memikirkan cara-cara untuk menyelesaikan indikator yang ada kemudian akan dihitung jumlah klik dan waktu dalam detik. Dari validasi ini, ingin didapatkan informasi yang dibutuhkan tentang bagaimana pengguna berinteraksi terhadap antar muka yang ada pada aplikasi Petqper [13]. Pengambilan data dilakukan ini dilakukan dengan observasi kepada pengguna yang akan mencoba aplikasi.

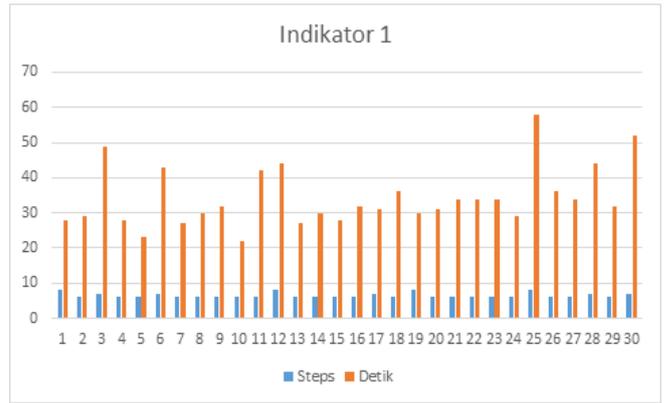
Untuk observasi ini digunakan 7 indikator yang memiliki kasus masing-masing. Tiap indikator akan menguji fitur sesuai dengan pemahaman pengguna. Indikator observasi dapat dilihat detail pada tabel V. Hasil dari observasi ini kemudian dirata-rata untuk mengetahui standar seseorang pengguna dapat menyelesaikan atau menggunakan aplikasi sehingga dapat membantu mencari penitipan hewan dengan waktu yang relatif rendah. Setelah rata-rata dari tiap indikator ditemukan, dihitung pula standar deviasi dari masing-masing indikator untuk mengetahui keberagaman nilai-nilai yang ada. Semakin rendah standart deviasi berarti semakin kecil keberagaman nilai dalam menghitung suatu rata-rata.

TABEL V.
CONTOH PERTANYAAN SKENARIO UJI VALIDASI

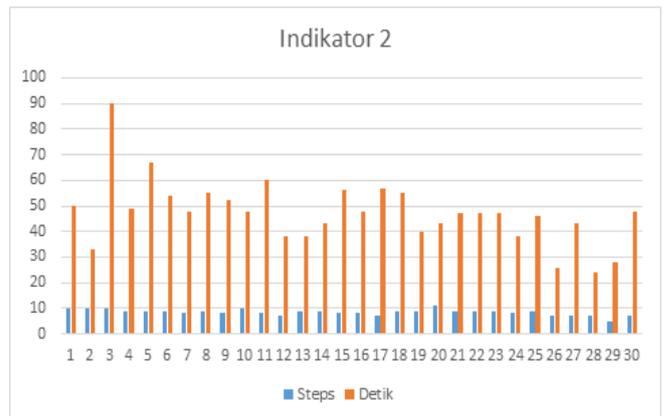
No	Idikator	Jumlah Klik	Lama (Detik)
1	Anda diminta membuat akun		
2	Anda diminta membooking tempat penitipan pada tgl dd/MM/yyyy jam JJ:mm dengan jumlah X dan lama Y tersebut		
3	Anda diminta membuat mengubah profil		
4	Anda diminta membuat tempat penitipan baru		
5	Anda diminta melihat riwayat penitipan yang telah dilakukan		
6	Anda diminta untuk logout		
7	Anda diminta untuk login dengan email yang didaftarkan		

Hasil dari observasi ini dapat lihat pada lampiran D. Untuk rata-rata dari masing-masing indikator rata-rata jumlah klik tertinggi adalah indikator nomor 4 karena indikator tersebut, pengguna diminta mengisikan data-data untuk membuat tempat penitipan hewan baru. Dan rata-rata waktu paling tinggi merupakan indikator nomor 4 yang disebabkan karena pengguna membuat penitipan hewan dan mengupload foto menuju Firebase memerlukan waktu yang relative lama berdasarkan kecepatan internet dan ukuran gambar tersebut.

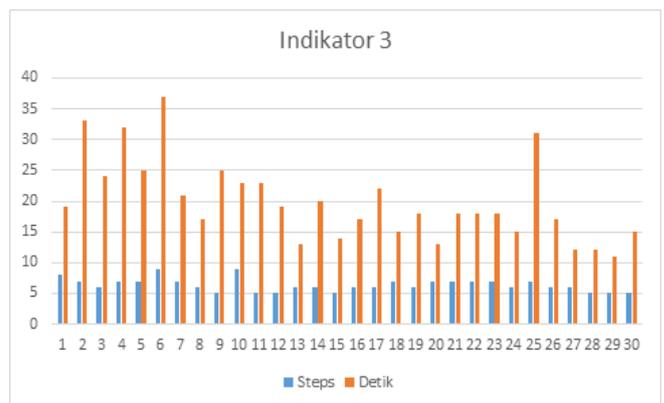
Berdasarkan hasil standar deviasi yang telah dihitung, didapatkan bahwa pada indikator nomor 4 memiliki keragaman yang tinggi. Hal ini disebabkan oleh terdapat pengguna yang memasukkan gambar dalam jumlah banyak sehingga proses upload berlangsung relatif lama. Selain nomor 4, indikator nomor 2 memiliki standar deviasi yaitu 12,71. Hal ini disebabkan oleh tampilan pada halaman booking yang tidak sesuai dengan apa yang dipikirkan oleh pengguna yang menyebabkan aksi yang dilakukan berulang-ulang. Gambar berikut merupakan grafik dari setiap indikator



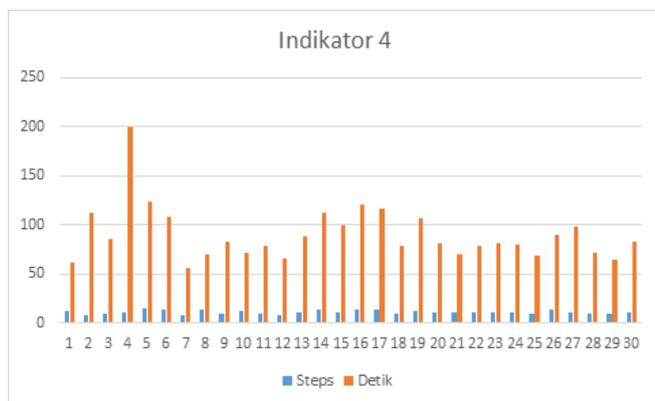
Gambar 17 Grafik Indikator 1



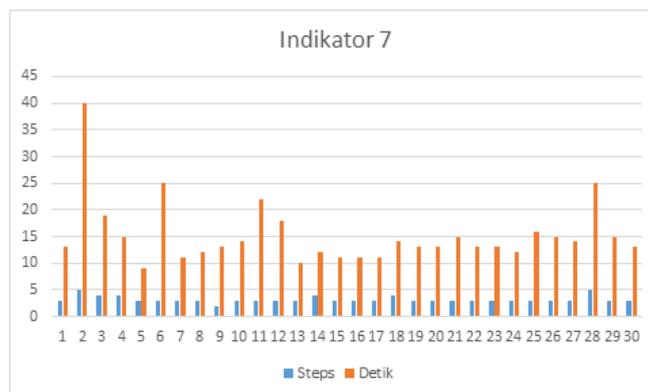
Gambar 18 Grafik Indikator 2



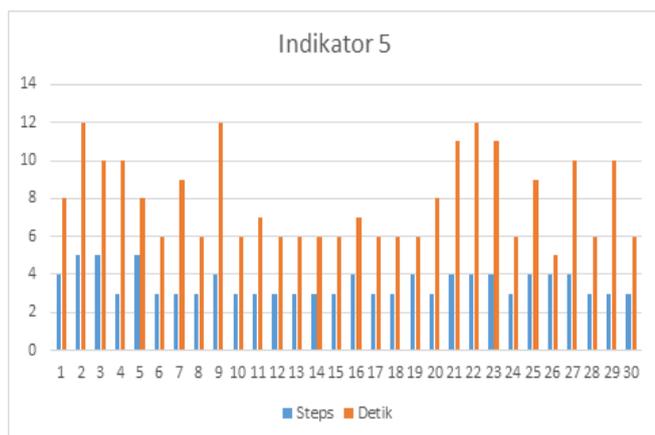
Gambar 19 Grafik Indikator 3



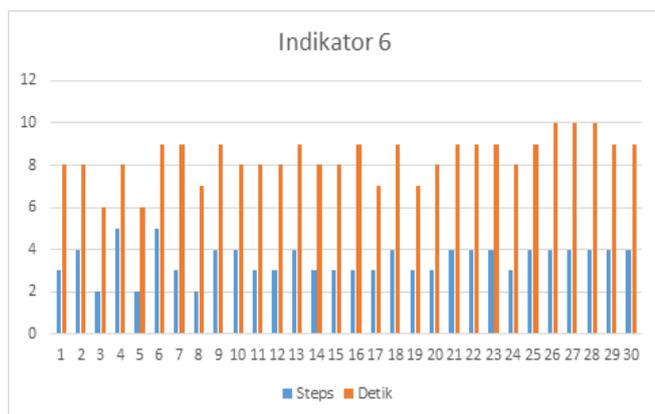
Gambar 20 Grafik Indikator 4



Gambar 23 Grafik Indikator 7



Gambar 21 Grafik Indikator 5



Gambar 22 Grafik Indikator 6

Berdasarkan hasil uji validasi, peneliti menyimpulkan beberapa hal berikut:

- Standar devisasi jumlah klik yang dilakukan oleh pengguna setiap indikator tergolong rendah dengan nilai tertinggi 1,92 yaitu pada indikator 4 akibat kurangnya koneksi internet dan besarnya data yang diupload
- Waktu yang dibutuhkan untuk menyelesaikan indikator nomor 4 tergolong lama karena harus mengisi data-data yang ada dan besar data yang diupload mempengaruhi
- Selain kecepatan internet dan besarnya data, terdapat pula tampilan-tampilan yang kurang jelas
- Kemampuan dari Android yang melambat sehingga terjadi penumpukan Google Maps saat menekan bagian alamat yang terlalu banyak

VI. KESIMPULAN

Berdasarkan hasil desain, implementasi, dan uji coba aplikasi Petqper, Kesimpulan yang didapatkan oleh peneliti adalah sebagai berikut:

- Aplikasi Petqper ini telah dirancang dan dibangun berdasarkan masalah-masalah yang dialami oleh pemilik hewan dalam mencari penitipan hewan.
- Aplikasi ini dapat digunakan untuk membantu dalam mencari tempat penitipan
- Pemilik jasa penitipan hewan dapat menggunakan aplikasi ini untuk menawarkan jasanya
- Masyarakat yang ingin mencari nafkah dapat membuat jasa penitipan hewan dengan aplikasi Petqper

TERIMA KASIH

Terima kasih kepada Teknik Informatika Universitas Ciputra yang telah membantu saya dalam melakukan penelitian ini baik dari segi bimbingan maupun pengajaran.

DAFTAR PUSTAKA

- [1] R. K. Nurlyayli and D. S. Hidayat, "Keseharian Pemilik Hewan Peliharaan yang Tinggal Terpisah dari Keluarga," *Jurnal Ilmiah Psikologi Terapan*, p. 23, 2014.
- [2] R. Adiputra, K. I. Satoto and R. Kridalukmana, "Rancang Bangun Aplikasi Layanan Perawatan dan Penitipan Hewan Berbasis SMS Gateway pada Toko Petshop "Petzone"," 2014.
- [3] A. A. g. Ekayana and G. N. E. Putra, "Rancang Bangun Sistem Pemberian Makan dan MInum Hewan Peliharaan Berbasis SMS Gateway," *Jurnal Ilmu Komputer dan Sains Terapan*, pp. 104-112, 2016.
- [4] M. E. Joorabchi, A. Mesbah and P. Kruchten, "Real Challenges in Mobile App Development," *ACM/IEEE Proceedings of the 7th International Symposium on Empirical Software Engineering and Measurement*, p. 15-24, 2013.
- [5] D. U. Rahmiati and E. S. Pribadi, "Tingkat Pendidikan dan Status Ekonomi Pemilik Hewan Kesayangan dalam Hal Pengetahuan dan Penerapan Kesejahteraan Hewan," *Jurnal Veteriner*, pp. 386-394, 2014.
- [6] H. S. Oluwatosin, "Client-Server Model," *Journal of Computer Engineering*, pp. 67-71, 2014.
- [7] D. A. Naidu and N. Hema, "A DESCRIPTIVE STUDY ON MOBILE APPLICATIONS," *International Journal of Innovative Science, Engineering & Technology*, pp. 761-763, 2015.
- [8] S.-H. Lim, "Experimental Comparison of Hybrid and Native Applications for Mobile Systems," *International Journal of Multimedia and Ubiquitous Engineering*, pp. 1-12, 2015.
- [9] A. Singh, S. Sharma and S. Singh, "Android Application Development using Android Studio and PHP Framework," *International Journal of Computer Applications*, pp. 5-8, 2016.
- [10] A. S. Google, "Mengenal Android Studio | Android Studio," Mei 2013. [Online]. Available: <https://developer.android.com/studio/intro/index.html?hl=id>. [Accessed 17 Juni 2017].
- [11] N. Singh, "Study of Google Firebase API for Android," *International Journal of Innovative Research in Computer and Communication Engineering*, pp. 16738-16743, 2016.
- [12] F. O. Google, "Firebase Realtime Database | Firebase," July 2016. [Online]. Available: <https://firebase.google.com/docs/database/>. [Accessed 17 Juni 2017].
- [13] L. Hasan and E. Abuelrub, "Usability Testing for IAJIT OpenConf Journal Management System," *Journal of Software*, vol. 8, no. 2, pp. 387-396, 2013.