

Pengembangan Aplikasi Pemantauan Kualitas Wi-Fi dengan *Crowdsourcing* berdasarkan Lokasi berbasis Android

I Made Setia Baruna¹, Agi Putra Kharisma², Issa Arwani³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹made.setia@gmail.com, ²agi@ub.ac.id, ³issa.arwani@ub.ac.id

Abstrak

Wi-Fi sudah menjadi pilihan utama bagi universitas sebagai perantara untuk menyediakan akses internet, namun konsekuensinya adalah penggunaan *Wi-Fi* yang meningkat tajam, namun berkurangnya kualitas dari jaringan yang diberikan. Tingginya penggunaan internet ini juga memerlukan lebih banyak alat untuk memantau kualitas dari jaringan yang ada, dan juga mengukur dan memverifikasi kualitas dari jaringan *Wi-Fi* sangatlah susah dikarenakan tidak adanya alat untuk mencakup semua variabel yang ada. Dari masalah tersebut dikembangkan sebuah aplikasi untuk memantau kualitas *Wi-Fi* yang datanya diambil secara *crowdsourcing* dimana data kualitas *Wi-Fi* tersebut dikumpulkan dari kontribusi pengguna yang melakukan tes. Data yang sudah terkumpul akan divisualisasikan ke dalam bentuk *heatmap* untuk memudahkan pengguna untuk mencari lokasi dengan kualitas *Wi-Fi* yang baik. *Heatmap* juga dapat disaring berdasarkan rentang waktu dan juga berdasarkan kecepatan *download*, *upload*, dan *ping*. Sistem yang telah dibuat, dirancang terlebih dahulu untuk fungsi-fungsi yang diperlukan, kemudian dikembangkan pada sistem Android dan diimplementasikan dengan bahasa pemrograman *Kotlin*. Aplikasi telah diuji dengan *black box testing* untuk pengujian validasi fungsionalitas dengan hasil valid untuk semua *test case*, dan mendapat skor SUS sebesar 72 yang menunjukkan nilai di atas rata-rata untuk pengujian usability.

Kata kunci: *Wi-Fi*, *Crowdsourcing*, *Heatmap*, *Kotlin*

Abstract

Wi-Fi has become the main choice for universities as media to provide internet access, but the consequence is that the use of *Wi-Fi* has increased significantly, but the quality of the network is reduced. The high use of the internet also requires more tools to monitor the quality of the existing network, and also measuring and verifying the quality of the *Wi-Fi* networks is very difficult because there is no tool to cover all variables. From this problem, an application was developed to monitor quality of *Wi-Fi*, whose data was taken by *crowdsourcing*, where *Wi-Fi* quality data was collected from contributions of users who conducted the test. The collected data will be visualized into a *heatmap* to make it easier for users to find good quality *Wi-Fi* location. *Heatmap* can also be filtered based on time range and by *download* speed, *upload* speed, and *ping*. The application was designed in advanced for the functions, and then developed for Android system implemented with *Kotlin* programming language. The application has been tested with *black box testing* for testing validation functionality with valid result for all test cases, and received SUS score of 72 which shows a value above average for usability testing.

Keywords: *Wi-Fi*, *Crowdsourcing*, *Heatmap*, *Kotlin*

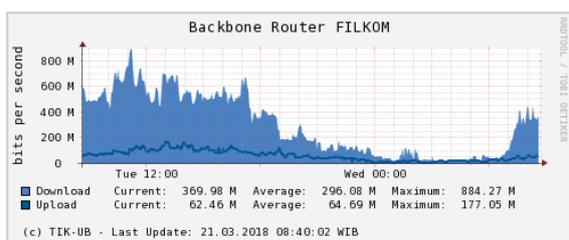
1. PENDAHULUAN

Wi-Fi sudah menjadi pilihan utama bagi universitas sebagai perantara untuk menyediakan layanan internet, karena mudahnya bagi civitas universitas untuk mengakses internet dengan laptop atau telepon genggam yang

mendukung *Wi-Fi*. Sebagai konsekuensinya, penggunaan *Wi-Fi* sangat meningkat tajam, namun berkurangnya kualitas dari jaringan yang diberikan.

Fakultas Ilmu Komputer Universitas Brawijaya (FILKOM UB) merupakan suatu fakultas di Universitas Brawijaya yang berbasis

Teknologi Informasi (FILKOM UB, 2018). FILKOM merupakan fakultas dengan penggunaan *bandwidth* internet tertinggi di Universitas Brawijaya (BITS UB, 2018). Dari Gambar 1 dapat diketahui bahwa penggunaan internet di FILKOM dengan *bandwidth* unduh mencapai 884 Mbps dan unggah mencapai 177 Mbps.



(BITS UB, 2018)

Gambar 1. Diagram data penggunaan internet FILKOM UB pada tanggal 21 Maret 2018

Tingginya penggunaan internet ini juga memerlukan lebih banyak alat untuk memantau kualitas dari jaringan yang ada. Mengukur dan memverifikasi kualitas dari *Wi-Fi* sangat susah, dikarenakan tidak ada alat yang mencakup semua variabel pengukuran. Dan juga perbedaan kualitas dari sisi pengguna sendiri dikarenakan adanya halangan fisik seperti pintu, lantai, atap, tembok, interferensi dari sinyal lain, dan kapasitas dari akses poin yang disediakan.

Civitas yang membutuhkan *Wi-Fi* sebagai alat untuk terhubung ke internet juga mengalami banyak permasalahan karena alasan yang disebutkan di atas. Susahnya memantau daerah mana yang menjadi titik-titik lemah dari *Wi-Fi* juga menjadi suatu masalah, sehingga lambatnya tindakan dari pihak Universitas itu sendiri, dikarenakan tidak bisa asal dan harus direncanakan terlebih dahulu.

Pada hasil penelitian sebelumnya oleh Bhardwaj et al. (2015), sudah menggunakan *Raspberry-PI* dan perangkat *mobile* untuk mendapatkan datanya. Namun tidak mengambil data kecepatan dari *Wi-Fi* yang terhubung. Kemudian penampilan data menggunakan web sebagai medianya.

Oleh karena permasalahan di atas, peneliti bermaksud untuk mengembangkan sebuah aplikasi yang dapat memantau kualitas dari *Wi-Fi* pada lokasi pengguna seperti sinyal, kecepatan dan *latency*, kemudian data tersebut divisualisasikan dalam bentuk *heatmap*. Data yang dikumpulkan merupakan data yang diambil secara *crowdsourcing*, dimana data tersebut merupakan kontribusi dari pengguna aplikasi

yang melakukan tes. Hasil dari penelitian ini diharapkan dapat membantu civitas universitas untuk mencari lokasi dengan kualitas *Wi-Fi* yang baik, dan juga bagi pihak Universitas untuk menggunakan data yang ada sebagai referensi untuk memperbaiki kualitas *Wi-Fi* di titik yang kurang baik.

2. LANDASAN TEORI

2.1. Pemrograman Android dengan Kotlin

Kotlin adalah bahasa pemrograman modern untuk mengembangkan aplikasi untuk sistem operasi Android. Kode Kotlin masih bisa berjalan berdampingan dengan kode Java yang sudah ada. Kotlin adalah bahasa pemrograman yang modern, *expresif*, dan terdapat fitur keamanan untuk *nullability* dan *immutability* (Android, 2018).

2.2. Wi-Fi

Wi-Fi merupakan teknologi jaringan nirkabel yang menggunakan gelombang radio berdasarkan standar IEEE 802.11. Beberapa perangkat yang bisa terhubung ke *Wi-Fi* di antara lain adalah komputer, telepon, tablet, kamera, dan printer. Perangkat yang mendukung *Wi-Fi* bisa terhubung ke internet melalui Wireless Access Point. Ada beberapa standar IEEE 802.11 antara lain 802.11b, 802.11a, 802.11g, 802.11n, dan 802.11ac, yang berkarakteristik mempunyai kecepatan lebih rendah, untuk menjangkau jarak yang lebih jauh. (Kurose, 2017).

2.3. Performa Jaringan

Ada beberapa pengukuran yang dilakukan dalam mengukur performa jaringan:

1. *Bandwidth*, merupakan berapa besar kecepatan maksimum pengiriman data yang bisa dilakukan
2. *Throughput*, merupakan kecepatan pengiriman data sebenarnya
3. *Latency*, merupakan seberapa lama waktu yang dibutuhkan untuk data dikirim dari pengirim dan diterima oleh penerima

Dari hal tersebut dibutuhkan *throughput* minimum yang diperlukan untuk mengambil konten dari internet dengan baik. Terdapat kecepatan *download* rekomendasi yang diukur dalam kB/s (*Kilobytes per second*) untuk menonton video pada youtube.com yang ditunjukkan pada Tabel 1.

Tabel 1. Kecepatan *Download* Rekomendasi

Sumber: Youtube Help, 2019

Resolusi	Kecepatan <i>Download</i>
4K	2500 kB/s
HD 1080p	625 kB/s
HD 720p	312.5 kB/s
SD 480p	137.5 kB/s
SD 360p	87,5 kB/s

Sumber: Youtube Help, 2019

Untuk kecepatan *upload* minimum yang diukur dalam kB/s (*Kilobytes per second*) untuk *streaming* dengan kualitas *HD* pada *streamspot.com* bisa dilihat pada Tabel 2.

Tabel 2. Kecepatan *Upload Minimum*

Resolusi	Kecepatan <i>Upload</i>
HD 1080p	437.5 kB/s
HD 720p	225 kB/s
SD 480p	150 kB/s

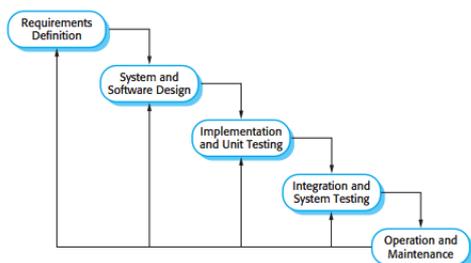
Sumber: StreamSpot, 2019

2.4. Crowdsourcing

Crowdsourcing merupakan suatu cara untuk mendapatkan suatu data, ide, atau konten, dengan meminta kontribusi dari orang banyak atau kelompok melalui media *online* atau internet, daripada menggunakan media tradisional. (MerriamWebster, 2018)

2.5. Waterfall Software Development Life Cycle

Model *waterfall* merupakan sebuah proses perancangan perangkat lunak dengan prinsip bahwa satu tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap selanjutnya, sehingga proses pengerjaannya terurut secara sekuensial, dan proses nya seperti mengalir kebawah secara terurut. (Sommerville, 2011)



Gambar 2. Tahapan SDLC model *Waterfall*

Sumber: Sommerville, 2011

2.6. Black Box Testing

Black box testing atau tes fungsional adalah testing yang tidak memedulikan bagaimana sistem bekerja, dan fokus pada hasil akhir yang dihasilkan oleh sistem (Jerry, 2003). *Black box testing* dilakukan dengan menguji semua kasus uji yang mungkin muncul, kemudian membandingkan keluaran yang muncul apakah sudah sesuai dengan hasil yang diharapkan.

2.6. System Usability Scale (SUS)

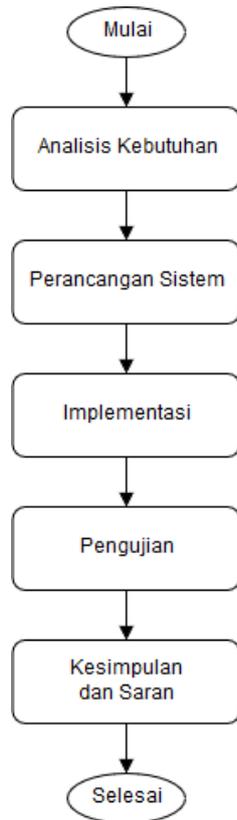
System Usability Scale (SUS) merupakan sebuah skala sederhana yang berisi 10 pernyataan untuk memberikan pandangan keseluruhan dalam penilaian usability sistem. (Brooke, 1986). SUS sudah menjadi standar industri dalam pengukuran usability yang bisa digunakan pada sampel yang kecil, dan valid dimana bisa secara efektif mengukur usability sistem.

Pengujian dilakukan kepada lima orang responden, karena dengan jumlah responden lebih dari lima hanya akan menunjukkan hasil yang sudah didapatkan dari lima responden sebelumnya, yang menunjukkan bahwa lima responden merupakan jumlah yang efektif. (NNGroup, 2000)

Dari 500 penelitian, skor rata-rata dari SUS yang didapat adalah 68. Skor di atas 68 bisa disebut sebagai di atas rata-rata, sebaliknya merupakan di bawah rata-rata. (MeasuringU, 2018).

3. METODOLOGI

Untuk mengerjakan penelitian ini, penulis memilih untuk menggunakan model Software Development Life Cycle Waterfall. Dalam model ini terdapat lima tahap yaitu dimulai dari analisis kebutuhan sampai pemberian saran untuk penelitian selanjutnya. Dalam tahapan-tahapan tersebut juga didampingi dengan studi literatur untuk mencari literatur-literatur yang dibutuhkan. Peneliti menggunakan SDLC ini karena kebutuhan sistem yang tidak berubah-ubah, sehingga bisa dikerjakan sesuai dengan tahapan yang sudah ditentukan. Alur tahapan metodologi ditunjukkan pada Gambar 3.



Gambar 3. Alur Tahapan Metodologi Penelitian

3.1. Studi Literatur

Dalam Studi literatur akan mencari literatur yang dapat membantu dalam penelitian yang berfungsi sebagai acuan dalam penelitian. Literatur yang digunakan bersumber dari buku, jurnal, halaman *website*, dokumentasi *SDK*, dokumentasi *library*, dokumentasi *framework*, dokumentasi bahasa pemrograman., serta penelitian lain yang bisa digunakan untuk mendukung penelitian ini.

3.1. Metode Pengembangan Sistem Waterfall

Tahapan-tahapan dalam *Software Development Life Cycle (SDLC) Waterfall* adalah sebagai berikut:

3.1.1 Analisis Kebutuhan

Dalam tahap analisis kebutuhan menjelaskan bagaimana akan dilakukan pencarian kebutuhan sistem untuk pengembangan aplikasi nantinya. Dan digunakan sebagai patokan untuk pengembangan aplikasi *monitoring* kualitas *Wi-Fi* dengan *crowdsourcing* berbasis Android. Analisis kebutuhan terdapat beberapa proses

meliputi:

1. **Elisitasi Kebutuhan**
 Pada proses ini elisitasi kebutuhan dilakukan dengan melakukan wawancara kepada mahasiswa Fakultas Ilmu Komputer yang akan dianalisis untuk mengetahui kebutuhan sistem.
2. **Spesifikasi Kebutuhan**
 Pada proses ini, kebutuhan yang sudah didapatkan akan diidentifikasi kembali untuk mengidentifikasi aktor dari aplikasi sehingga menghasilkan kebutuhan fungsional dan non-fungsional. Kemudian dimodelkan dalam bentuk *Usecase diagram* dan *Usecase scenario*.
3. **Manajemen Kebutuhan**
 Pada proses ini dilakukan manajemen kebutuhan dan pemberian kode pada kebutuhan yang sudah diidentifikasi .
4. **Analisis Data**
 Pada proses ini dilakukan untuk mendapatkan struktur data untuk menyimpan data yang diperlukan dalam aplikasi untuk menyimpan hasil tes yang sudah dilakukan.

3.1.2 Perancangan Sistem

Setelah tahap analisis kebutuhan sudah selesai, akan dilanjutkan dengan tahap perancangan sistem. Dalam tahap ini akan dilakukan perancangan arsitektur aplikasi, *User Interface* dan *User experience* dari aplikasi, menentukan cara untuk mengambil data kecepatan, *latency*, dan sinyal dari *Wi-Fi* yang terkoneksi dengan perangkat *Smartphone*, metode pengumpulan data ke server, dan melakukan pemodelan sistem dengan menggunakan *Unified Modeling Language (UML)*, seperti *activity diagram*.

3.1.3 Implementasi

Setelah tahap perancangan sistem selesai, akan dilanjutkan dengan tahap implementasi. Dari hasil perancangan yang sudah dibuat akan dibuat implementasi algoritme yang diperlukan. Aplikasi akan diimplementasikan dengan menggunakan bahasa pemrograman Kotlin dengan menggunakan aplikasi IDE Android Studio.

3.1.4 Pengujian

Setelah implementasi selesai, kemudian akan dilanjutkan dengan tahap pengujian untuk

mengetahui apakah aplikasi yang sudah diimplementasikan sesuai dengan kebutuhan yang sudah dianalisis pada tahap analisis kebutuhan dan perancangan sistem. Selanjutnya akan dilakukan pengujian *black box testing* untuk memvalidasi fungsi-fungsi yang sudah diimplementasikan, dan juga pengujian usabilitas untuk menguji tingkat usabilitas aplikasi.

3.1.5 Kesimpulan dan Saran

Setelah semua tahap sebelumnya dilakukan, terakhir adalah mengambil kesimpulan dari penelitian yang sudah dilakukan setelah semua tahap sebelumnya selesai. Kesimpulan ini bertujuan untuk menjawab dari rumusan masalah yang sudah dirumuskan sebelumnya. Kemudian pemberian saran untuk memberikan saran-saran sehingga penelitian selanjutnya bisa memperbaiki kekurangan dan kesalahan dari penelitian ini.

4. ANALISIS KEBUTUHAN & PERANCANGAN

4.1. Analisis Kebutuhan

4.1.1. Gambaran Umum Aplikasi

Aplikasi pemantauan kualitas *Wi-Fi* ini merupakan aplikasi yang dapat digunakan untuk melihat bagaimana kualitas *Wi-Fi* yang ada pada suatu lokasi dalam waktu tertentu. Aplikasi dapat menampilkan data kualitas *Wi-Fi* tersebut dalam bentuk *heatmap*, yang menunjukkan bagaimana baik-buruknya kualitas *Wi-Fi* dengan data yang ada.

Data kualitas *Wi-Fi* yang digunakan merupakan hasil tes dari semua pengguna aplikasi itu sendiri yang digabungkan. Pengguna dapat berkontribusi untuk melakukan suatu tes untuk mendapatkan kecepatan *download*, kecepatan *upload*, dan *ping* dari *Wi-Fi* yang terhubung, kemudian juga waktu dan lokasi dari pengguna tersebut. Data yang sudah didapat kemudian di kirimkan ke server dan di proses untuk kemudian dilihat dalam bentuk *heatmap*.

4.2. Perancangan

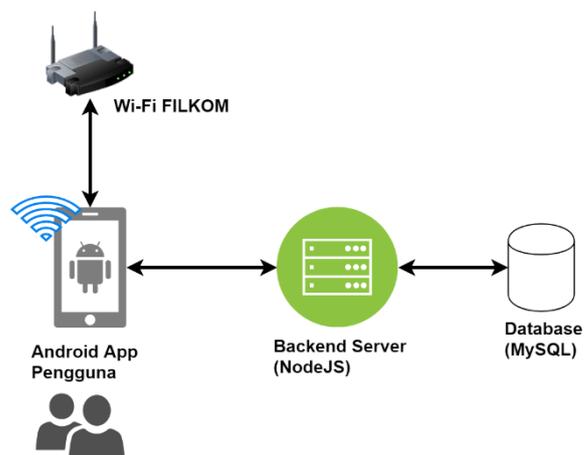
4.2.1. Perancangan Arsitektur Sistem

Dalam aplikasi ini memerlukan dua buah subsistem, yaitu aplikasi mobile Android dan aplikasi server untuk sisi *back-end*. Aplikasi mobile akan melakukan tes dan mengirimkan data ke server, kemudian untuk menampilkan

data *heatmap*, aplikasi akan mengambil data terlebih dahulu ke server, server mengolah data yang diinginkan, kemudian dikirim kembali ke aplikasi mobile untuk ditampilkan.

Pola arsitektur yang digunakan pada aplikasi ada arsitektur MVC atau *Model, View*, dan *Controller*. Model merupakan representasi dari data yang berasal dari server maupun dari perangkat itu sendiri. View merupakan bagaimana tampilan atau antarmuka dari aplikasi itu sendiri, dan terakhir merupakan *Controller* yang menjembatani pengiriman data antar *Model* dan bagaimana cara menampilkannya pada *View*. Pada *Controller* juga melakukan fungsi-fungsi lain yang diperlukan.

Kemudian pada sisi server menggunakan *Node-JS* sebagai *http* server dan *Mysql* sebagai *database*-nya. Pada *database* akan menyimpan data-data hasil tes yang sudah dikirim dari aplikasi. Yang kemudian akan dibaca kembali untuk membuat *heatmap* nya.



Gambar 5. Arsitektur Sistem

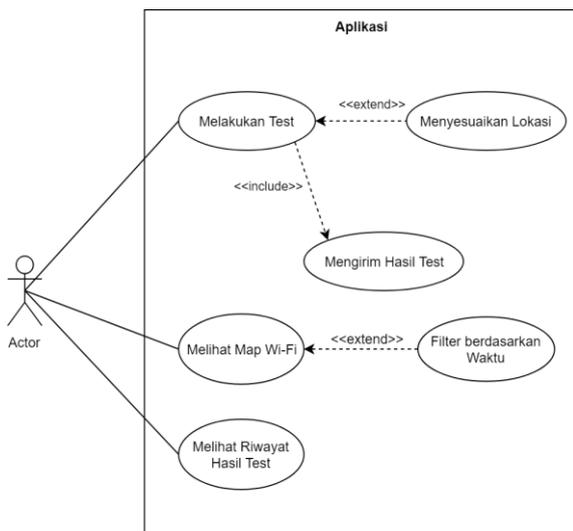
4.1.2. Analisis Use Case Diagram

Use case diagram berisi tentang apa saja yang bisa dilakukan oleh aktor pada sistem sesuai dengan kebutuhan yang sudah dianalisis. *Use case* diagram aplikasi ditunjukkan pada Gambar 4.

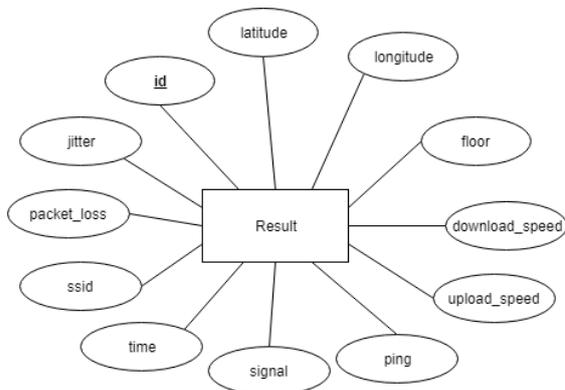
4.2.3. Perancangan Basis Data

Sebelumnya dilakukan dulu perancangan basis data untuk mengetahui bagaimana data nantinya akan disimpan pada aplikasi maupun pada server. Pada rancangan basis data pada Gambar 6, terdapat 1 buah tabel *Result* yang mempunyai kolom *latitude*, *longitude*, *floor*, *download_speed*, *upload_speed*, *ping*, *signal*, *time*, *ssid*, *packet_loss*, *jitter*, dan *id* sebagai *primary key*-nya. Tabel tersebut akan digunakan

pada aplikasi sisi Android dan sisi server.



Gambar 4. Use Case Diagram



Gambar 6. ERD Sistem

4.2.3. Perancangan Visualisasi Data

Data hasil tes yang dikumpulkan akan ditampilkan dalam bentuk *heatmap*, yang dimana map yang dibagi dalam bentuk kisi-kisi. Nilai-nilai pada suatu sel akan diambil rata-ratanya, yang kemudian ditunjukkan dengan warna hijau, kuning, dan merah sebagai penunjuk buruk, normal, dan baik. Batasan untuk warna-warna ditunjukkan pada Tabel 3.

Tabel 3. Pembagian Batasan Warna

	Hijau	Kuning	Merah
Download	>625 kb/S	137,5-625 kB/s	<137,5 kB/s
Upload	>437,5 kb/s	150-437,5 kB/s	< 150 kB/s
Ping	<100 ms	< 200 ms	> 200 ms

5. IMPLEMENTASI

5.1. Implementasi Basis Data

Dalam implementasi basis data terdapat dua sisi basis data, yaitu pada sisi Android dan pada sisi server. Pada sisi klien menggunakan *SQLite* dan pada sisi server menggunakan *MariaDB* yang sama-sama merupakan *database* berbasis SQL, sehingga implementasinya menggunakan tabel. Untuk sisi klien merupakan hasil tes yang sudah dijalankan pengguna di perangkat tersebut saja, dan di sisi server merupakan hasil tes dari semua pengguna yang berhasil dikirimkan. Untuk implementasi dari tabel yang digunakan bisa dilihat pada Gambar 7 dan 8:

test_result		
PK	id	integer(11)
	latitude	decimal(10,6)
	longitude	decimal(10,6)
	floor	integer(11)
	download_speed	unsigned int(10)
	upload_speed	unsigned int(10)
	ping	unsigned int(10)
	signal	int(10)
	ssid	varchar(255)
	packet_loss	float
	jitter	unsigned int(10)
	time	datetime
	status	tinyint(1)

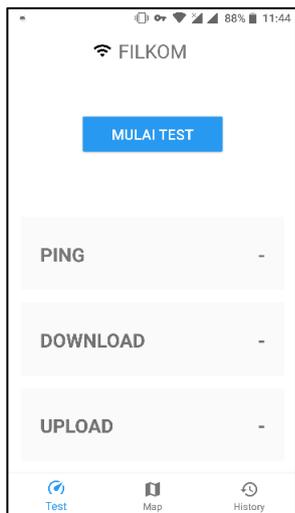
Gambar 7. ERD Implementasi Basis Data Server

test_result		
PK	id	INTEGER
	latitude	REAL
	longitude	REAL
	floor	INTEGER
	download_speed	INTEGER
	upload_speed	INTEGER
	ping	INTEGER
	signal	INTEGER
	ssid	TEXT
	packet_loss	REAL
	jitter	INTEGER
	time	TEXT
	status	INTEGER

Gambar 8. ERD Implementasi Basis Data Aplikasi

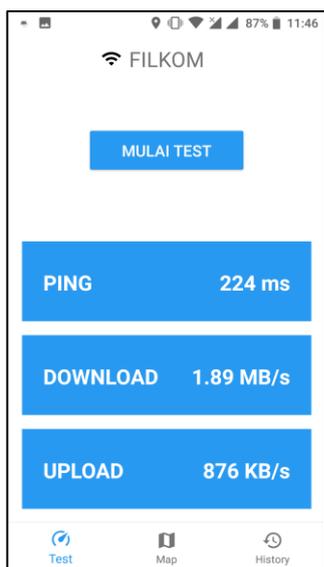
5.2. Implementasi Antarmuka

Pada Implementasi antarmuka dibuat antarmuka dari aplikasi sesuai dengan kebutuhan yang sudah dianalisis. Terdapat beberapa antarmuka, antara lain: antarmuka halaman melakukan tes *Wi-Fi*, antarmuka halaman melihat *heatmap* kualitas *Wi-Fi*, dan antarmuka halaman melihat riwayat tes.



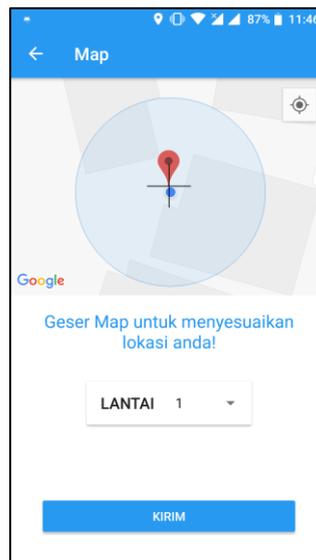
Gambar 9. Halaman Melakukan Tes Wi-Fi

Pada halaman ini terdapat tombol “Mulai Test” untuk melakukan tes *Wi-Fi*, dan juga teks *ping*, *download*, *upload* yang nantinya akan berisi nilai hasil tes. Pada bagian bawah juga terdapat menu untuk pindah ke halaman lainnya.



Gambar 10. Halaman Melakukan Tes Wi-Fi Selesai

Halaman ini merupakan halaman melakukan Test *Wi-Fi* yang sudah selesai, dimana nilai hasil test diperlihatkan.



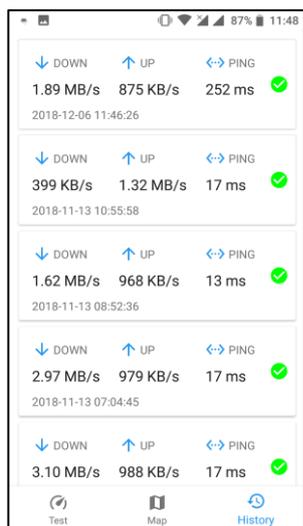
Gambar 11. Halaman Menyesuaikan Lokasi

Halaman ini merupakan halaman yang muncul setelah tes selesai untuk menyesuaikan lokasi pengguna. Pengguna akan menggeser map untuk menyesuaikan lokasinya, dan memilih lantai dari *dropdown*.



Gambar 12. Halaman Heatmap Kualitas Wi-Fi

Halaman ini merupakan halaman *heatmap* kualitas *Wi-Fi* dari hasil seluruh tes yang dilakukan. Map dapat di *zoom-in* dan *zoom-out*, dan pengguna juga dapat memilih *heatmap* berdasarkan hasil *download*, *upload*, atau *ping*. Dan juga rentang waktu yang diinginkan.

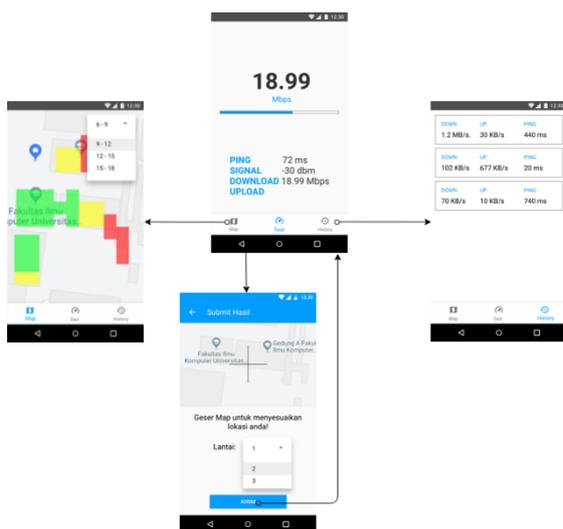


Gambar 13. Halaman Riwayat Tes

Halaman ini merupakan halaman yang menunjukkan seluruh hasil tes yang sudah dilakukan oleh pengguna.

5.2. Screen Flow

Gambar 14 menunjukkan rancangan *screen flow* dari aplikasi. Dari halaman tes, jika tombol map diklik akan menuju halaman *heatmap* kualitas *Wi-Fi*. Kemudian setelah tes selesai, akan muncul halaman submit hasil dimana user bisa menyesuaikan lokasinya, dan jika tombol kirim diklik akan kembali ke halaman tes. Terakhir jika tombol riwayat di klik, akan menuju halaman riwayat hasil tes.



Gambar 14. Screen Flow Aplikasi

6. PENGUJIAN

6.1. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan menggunakan metode *blackbox testing*. Pengujian dilakukan dengan menjalankan aplikasi sesuai dengan *task-task* yang sudah ditentukan. Setelah dilakukan pengujian, didapatkan hasil pengujian bahwa semua kebutuhan fungsional berjalan sesuai dengan ketentuan yang sudah dirancang.

6.2. Pengujian Usabilitas

Setelah dilakukan pengujian usabilitas oleh 5 orang responden, untuk *task completion rate* menunjukkan hasil 100% yang menandakan bahwa keenam tugas yang diberikan berhasil diselesaikan tanpa adanya masalah.

Dari sepuluh pernyataan yang ada, pernyataan ketiga mendapat nilai tertinggi yaitu 3.4, yang menandakan bahwa aplikasi sudah mudah digunakan. Sedangkan nilai terendah ada pada pernyataan keempat yaitu mendapat skor 2 yang menandakan bahwa aplikasi masih memerlukan panduan yang lebih jelas untuk digunakan.

Dari skor SUS yang didapatkan yaitu 72, yang menandakan nilai usabilitas di atas rata-rata dan mendapat nilai B. Hal tersebut menandakan bahwa pengguna sudah bisa menggunakan aplikasi dengan baik. Namun perlu adanya penambahan bantuan agar aplikasi bisa lebih mudah dipelajari kembali.

Hal ini berbanding lurus dari umpan balik yang diberikan oleh pengguna, yang memberikan komentar bahwa perlunya fitur bantuan dan lebih *user friendly* lagi, sehingga diperlukan penelitian lebih lanjut untuk memperbaiki *user experience* tersebut.

7. PENUTUP

Berdasarkan hasil penelitian yang sudah dilakukan, kesimpulan yang didapat adalah sebagai berikut:

1. Rancangan aplikasi pengembangan aplikasi pemantauan kualitas *Wi-Fi* terdiri dari 4 halaman yaitu: halaman untuk melakukan tes kualitas *Wi-Fi*, halaman untuk menyesuaikan lokasi, halaman untuk melihat *heatmap* kualitas *Wi-Fi*, dan halaman untuk melihat riwayat hasil tes pengguna. Tes yang dilakukan akan mengambil data kualitas *ping*, kecepatan

download, dan kecepatan *upload*, dan kemudian data tersebut dikirimkan ke server beserta lokasi dari pengguna.

2. Pengambilan data pada aplikasi pemantauan kualitas *Wi-Fi* pada data kualitas *ping*, dilakukan dengan cara melakukan tes *ping* dengan jumlah 10 kali dan diambil rata-ratanya. Untuk kecepatan *download* diambil dengan cara mengunduh data dari server, kemudian mengecek jumlah *byte* yang masuk pada waktu yang sudah berjalan. Dan untuk kecepatan *upload* diambil dengan cara mengunggah data ke server, kemudian mengecek jumlah *byte* yang sudah masuk ke server pada waktu yang sudah berjalan. Kemudian lokasi dari *smartphone* diambil, pengguna bisa menyesuaikan lokasi tersebut dan memilih lantai dimana pengguna berada, dan kemudian data hasil tes dan lokasi dikirimkan ke server.
3. Visualisasi dari data yang sudah diambil pada aplikasi pemantauan kualitas *Wi-Fi* pada data yang sudah dikumpulkan adalah dengan menggunakan *heatmap* pada map yang menunjukkan rata-rata dari kualitas *Wi-Fi* yang ada pada suatu daerah pada map tersebut. Kualitas *Wi-Fi* ditunjukkan dengan rentang warna hijau, kuning, dan merah. Data visualisasi dibagi menjadi data kualitas *download*, kualitas *upload*, dan kualitas *ping*. Dan dibagi kembali untuk memilih rentang waktu setiap 3 jam.
4. Aplikasi sudah memenuhi seluruh *requirement* melalui pengujian *blackbox*, dan mendapatkan nilai SUS skor 72 yang menunjukkan usability di atas rata-rata dan mendapat nilai B yang artinya aplikasi sudah bisa digunakan oleh pengguna dengan baik

Adapun saran untuk penelitian dan pengembangan selanjutnya adalah sebagai berikut:

1. Aplikasi dapat mengirimkan data tes yang gagal untuk mendapatkan kestabilan dari *Wi-Fi* yang terhubung
2. Visualisasi yang mengombinasikan nilai *download*, *upload*, dan *ping* untuk memperlihatkan kondisi secara keseluruhan
3. *Heatmap* dapat ditingkatkan kembali dengan tidak hanya menampilkan tiga warna, namun bisa dengan menunjukkan rentangan warna
4. Prediksi kualitas di area yang belum ada data tes-nya sama sekali

5. Pengambilan lokasi pengguna dengan cara lain yang lebih akurat, dan juga pengambilan lokasi lantai dimana pengguna berada secara otomatis
6. Penyaringan data lama agar data yang ditunjukkan lebih relevan
7. Menggunakan beberapa server sehingga tes dilakukan dengan server yang terdekat
8. Perbaikan *user experience* dengan memberikan fitur bantuan

8. DAFTAR PUSTAKA

- Android, 2018. Kotlin and Android. [online] Tersedia di: <<https://developer.android.com/kotlin/>> [Diakses 1 Agustus 2018]
- Bhardwaj, D., Kataoka, K., Kumar, V., 2014. Wi-Pi: Distributed *Wi-Fi* Performance Assessment using Raspberry Pi. Proceedings of the AINTEC 2014 on Asian Internet Engineering Conference, p.31.
- Bhardwaj, D., Kataoka, K., Vikram, V. A., & Hirani, V., 2015. Hybrid Approach to Distributed *Wi-Fi* Performance Assessment for Multi-floor Structures. 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp.1-8.
- BITS UB, 2018. Backbone Router Traffic. [image online] Tersedia di: <<http://bits.ub.ac.id/status-layanan-jaringan/>> [Diakses 21 Maret 2018]
- Brooke, J., 1986. SUS – A Quick And Dirty Usability Scale. United Kingdom.
- Kurose, R., Ross. K., 2016. Computer Networking: A Top-Down Approach. 7th Edition. New Jersey: Pearson.
- Measuring U., 2011. Measuring Usability With The System Usability Scale (SUS). [online] Tersedia di <<https://measuringu.com/sus>> [Diakses 6 Desember 2018]
- MerriamWebster, 2018. Crowdsourcing. [online] Tersedia di <<https://www.merriam-webster.com/dictionary/crowdsourcing>> [Diakses 1 Agustus 2018]
- Pressman, R. S., 2008. Software Engineering: A Practitioner's Approach. 7th Edition. New York: The McGraw-Hill Companies

Sommerville, I., 2011. Software Engineering. 9th Edition. Boston: Pearson.

StreamSpot Knowledgebase, 2019. How fast does my Internet need to be to stream?. [online] Tersedia di <[https://support.streamspot.com/hc/en-us/articles/215567677-How-fast-does-my-Internet-need-to-be-to-stream->](https://support.streamspot.com/hc/en-us/articles/215567677-How-fast-does-my-Internet-need-to-be-to-stream-) [Diakses 24 Juli 2019]

Youtube Help, 2019. System Requirements. [online] Tersedia di <[https://support.google.com/youtube/answer/78358?hl=en>](https://support.google.com/youtube/answer/78358?hl=en) [Diakses 24 Juli 2019]