

**PERANCANGAN APLIKASI PEMINJAMAN SEPEDA DI UNIVERSITAS
LAMPUNG BERBASIS ANDROID MENGGUNAKAN METODE *AGILE*
*SCRUM***

(Skripsi)

Oleh:

**FIONA YOVITA SYAFRI
1915061033**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

**PERANCANGAN APLIKASI PEMINJAMAN SEPEDA DI UNIVERSITAS
LAMPUNG BERBASIS ANDROID MENGGUNAKAN METODE *AGILE*
*SCRUM***

Oleh:

Fiona Yovita Syafri

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023**

ABSTRAK

PERANCANGAN APLIKASI PEMINJAMAN SEPEDA DI UNIVERSITAS LAMPUNG BERBASIS ANDROID MENGGUNAKAN METODE AGILE SCRUM

Oleh

FIONA YOVITA SYAFRI

Peminjaman sepeda merupakan salah satu fasilitas untuk menunjang mobilitas kampus untuk bepergian dari satu fakultas ke fakultas lain. Pada penelitian ini dikembangkan aplikasi peminjaman sepeda dengan adanya otentikasi pengguna menggunakan *Single Sign-On Unila*. Penelitian ini bertujuan untuk membangun sistem peminjaman sepeda *online* berbasis *android* untuk Universitas Lampung. Aplikasi ini menyediakan informasi mengenai *shelter* peminjaman, jumlah sepeda yang tersedia dan status peminjaman sepeda oleh pengguna. Metode yang digunakan dalam mengembangkan aplikasi peminjaman sepeda ini adalah metode *Agile Scrum*. Pengembangan aplikasi dilakukan selama 4 *sprint* yang berdurasi selama 40 hari dengan total 20 *backlog*. Proses pengujian aplikasi dilakukan dengan metode *black box testing* dengan total 15 skenario pengujian. Hasil pengujian *black box* menyatakan bahwa seluruh skenario menghasilkan keluaran sesuai dengan yang diharapkan. Sehingga, didapatkan kesimpulan bahwa aplikasi peminjaman sepeda yang dikembangkan sudah berjalan sesuai kebutuhan fungsional dan non-fungsional nya. Kemudian dilakukan *User Experience Questionnaire* ke *civitas* akademika Unila. Hasil UEQ menyatakan bahwa dari 6 kategori, yaitu *Attractiveness*, *Perspiciuity*, *Efficiency*, *Dependability*, *Stimulation*, dan *Novelty* mendapat nilai *Excellent* atau sangat baik dengan nilai tertinggi yaitu 2.375 pada kategori *perspiciuity*.

Kata Kunci: *Android*, Peminjaman Sepeda, *Agile Scrum*, *Black Box Testing*, *User Experience Questionnaire*

ABSTRACTION

DEVELOPMENT OF ANDROID BASED LENDING BIKE APPLICATION IN LAMPUNG UNIVERSITY USING AGILE SCRUM METHOD

By

FIONA YOVITA SYAFRI

Bike lending is one of the facilities to support mobility on campus to travel from one faculty to another. In this study, a bike lending application was developed with user authentication using Single Sign-On Unila. This study aims to build an Android-based online bike lending system for Lampung University. This application provides information about the lending shelter, the number of bicycles available and the status of bicycles borrowed by the user. The method used in developing this bike lending application is the agile scrum method. The application development was done through 4 sprints for 40 days with a total of 20 backlogs. The application testing process is done using the black box testing method with a total of 15 test scenarios. The results of black box testing state that the resulting output is in accordance with the expected output. Thus, it can be concluded that the developed bike lending application has been successful according to its functional and non-functional requirements. Then a user experience test was done by spreading the User Experience Questionnaire to the Unila academic community. The UEQ results stated that out of the 6 categories of UEQ questions, namely Attractiveness, Perspicuity, Efficiency, Dependability, Stimulation, and Novelty, received an Excellent score in all categories, with the highest score of 2,375 in the perspicuity category.

Keyword: *Android, Bike Lending, Agile Scrum, Black Box Testing, User Experience Questionnaire*

Judul Skripsi : **PERANCANGAN APLIKASI
PEMINJAMAN SEPEDA DI
UNIVERSITAS LAMPUNG BERBASIS
ANDROID MENGGUNAKAN METODE
*AGILE SCRUM***

Nama Mahasiswa : ***Fiona Yovita Syafri***

Nomor Pokok Mahasiswa : 1915061033

Program studi : Teknik Informatika

Fakultas : Teknik

MENYETUJUI

1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

Ir. Meizano Ardhi M, S.T., M.T.
NIP. 198105282012121001

Rio Ariestia P, S.Kom., M.T.I.
NIP. 198603232019031013

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

Herlinawati, S.T., M.T.
NIP. 197103141999032001

Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

MENGESAHKAN

1. Tim Penguji

Ketua : **Ir. Meizano Ardhi Muhammad, S.T., M.T.**

Sekretaris : **Rio Ariestia Pradipta, S.Kom., M.T.I.**

Penguji : **Ing. Hery Dian Septama, S.T.**

2. Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc

NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : **11 Mei 2023**

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini , menyatakan bahwa skripsi saya dengan judul “Perancangan Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android Menggunakan Metode *Agile Scrum*” dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 11 Mei 2023

Pembuat pernyataan,

Fiona Yovita Syafri

NPM. 1915061033

RIWAYAT HIDUP



Penulis dilahirkan di Tangerang, pada tanggal 23 November 2001. Penulis merupakan anak ketiga dari tiga bersaudara dari pasangan Bapak Syafri dan Ibu Afniwaty Tanjung.

Penulis menyelesaikan pendidikannya di SD Negeri Porisgaga 3 pada tahun 2013, SMP Negeri 18 Tangerang pada tahun 2016, dan SMA Negeri 7 Tangerang pada tahun 2019. Pada tahun 2019 penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan, antara lain:

1. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Komunikasi dan Informasi, Divisi Media Informasi periode 2019/2020.
2. Menjadi anggota biasa Himpunan Mahasiswa Teknik Elektro Universitas Lampung, Departemen Komunikasi dan Informasi, Divisi Media Informasi periode 2020/2021.
3. Mengikuti program Studi Independen Kampus Merdeka dari Kementerian Pendidikan dan Budaya dengan mengambil kelas Pengembang Aplikasi Android dan *Multi-Platform* di Dicoding pada tahun 2021.
4. Menjadi Juara Harapan 1 di *Startup Pitching Competition* Bandar Lampung tahun 2021.
5. Melaksanakan Kuliah Kerja Nyata pada bulan Januari sampai dengan Februari 2022 di Desa Cipicung, Kecamatan Cikedal, Kabupaten Pandeglang, Provinsi Banten.
6. Mengikuti program Magang Bersertifikat Kampus Merdeka dari Kementerian Pendidikan dan Budaya pada tahun 2022 sebagai *Front-End Mobile Developer* di PT Telkom Indonesia.

MOTTO

“Allah tidak akan pernah membebani seseorang melainkan sesuai dengan kesanggupannya.”

(Q.S. Al-Baqarah : 286)

“Maka nikmat Tuhanmu yang manakah yang kamu dustakan?”

(Q.S. Ar-Rahman)

“Untuk menulis tentang kehidupan pertama-tama kamu harus menjalaninya.”

(Ernest Hemingway)

“Kamu harus mengharapkan hal-hal besar dari dirimu sebelum kamu dapat melakukannya.”

(Michael Jordan)

PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji syukur kepada Allah SWT. atas segala Rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam kepada Nabi Muhammad SAW sebagai suri tauladan *akhlakul karimah*.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku Tercinta

Yang senantiasa selalu memberikan yang terbaik, mengorbankan begitu banyak hal dan melantunkan doa yang tak henti-hentinya untukku. Kuucapkan terima kasih sebesar-besarnya karena telah mendidik dan membesarkanku dengan kasih sayang, kecukupan, dan kebahagiaan yang akan selalu aku syukuri seumur hidupku. Semoga dengan ilmu yang kudapatkan dari hasil jerih payah kalian menyekolahkanku akan menjadi amal jariyah bagi kalian.

Diriku sendiri yang telah berjuang sampai akhir. Terimakasih telah bertahan dan maaf untuk banyaknya air mata yang harus dikeluarkan karena hidup di tanah rantau sendirian. Tidak ada orang yang akan menghargai usahamu sendiri dan bangga pada dirimu sendiri sebagaimana aku bangga padamu. Kamu hebat. Semoga kamu sukses dan bahagia selalu ya.

Seluruh Keluarga Besar Teknik Elektro 2019

Terimakasih telah menemani dan berjuang bersama di bangku perkuliahan. Begitu banyak cerita yang dilalui bersama, susah, senang, tangis yang akan selalu kukenang. Semoga kita bertemu kembali sebagai orang-orang sukses yang telah mencapai cita-cita dan kebahagiaannya masing-masing.

Almamater tercinta, Universitas Lampung dan Jurusan Teknik Elektro

SANWACANA

Puji syukur penulis panjatkan kepada kehadirat Allah SWT. yang telah memberikan rahmat dan ridho-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul “Perancangan Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android Menggunakan Metode *Agile Scrum*”. Selama masa penelitian penulis mendapatkan banyak bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Kedua orangtua tercinta dan keluarga penulis yang selalu memberikan dukungan, do’a, kasih sayang tiada akhir dan mengingatkan penulis untuk bangkit dalam menyelesaikan penelitian ini;
2. Bapak Dr. Eng. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung;
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung;
4. Bapak Mona Arif Muda, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Universitas Lampung dan telah membantu proses kelancaran pengerjaan penelitian;
5. Bapak Ir. Meizano Ardhi Muhammad, S.T., M.T. selaku Pembimbing Utama yang selalu meluangkan waktunya untuk memberikan bimbingan dan dukungan serta memudahkan penulis dalam menyelesaikan penelitian ini;
6. Bapak Rio Ariestia Pradipta, S.Kom., M.T.I., selaku Pembimbing Pendamping yang selalu memberikan dukungan serta bimbingan agar menjadi lebih baik;
7. Bapak Ing. Hery Dian Septama, S.T. selaku Penguji yang telah memberikan banyak saran dan masukan terhadap penelitian ini;

8. Bapak M. Komaruddin, S.T., M.T. selaku Pembimbing Akademik yang memberikan dukungan serta arahan agar penulis menjadi lebih baik dalam perkuliahan;
9. Mbak Rika selaku Admin Program Studi Teknik Informatika yang telah banyak membantu penulis dalam segala urusan administrasi selama perkuliahan;
10. Irdoaji, sosok nomor satu yang penulis cari ketika masa-masa sulit dan butuh dukungan selama penelitian. Terima kasih telah memberi banyak bantuan, perhatian, do'a, meluangkan waktu, mendengarkan keluh, kesah, dan tangis, serta bersabar menghadapi penulis. Semoga kita bisa seterusnya melewati banyak hal bersama-sama.
11. Alya Nurul Fakhira, Tasya Cynthia Monica L, Aurellia Salma Fertiyan, Atiqah Hanifah Shalihah, Azzahra Agitha Karina dan segenap teman-teman kelas TI A 2019 yang telah menjadi rumah bagi penulis selama masa perkuliahan. Menemani, membantu, dan memberikan pundak bagi penulis dikala susah dan sedih.
12. Teman-teman penulis di Tangerang, yang tidak mungkin penulis sebut satu persatu. Terimakasih telah menjadi pendukung nomor satu bagi penulis, selalu menemani penulis dikala kesepian dan selalu mengingatkan untuk cepat pulang ke rumah.
13. Keluarga besar Teknik Elektro Angkatan 2019 yang telah menjadi teman seperjuangan sejak mahasiswa baru. Terimakasih telah mewarnai masa perkuliahan penulis dan menulis banyak cerita bersama.

Penulis berharap agar laporan ini dapat menjadi referensi bagi pengembangan keilmuan di bidang teknik informatika. Oleh karena itu, semoga penelitian ini bermanfaat bagi yang membacanya.

Bandar Lampung, 11 Mei 2023

Penulis,

Fiona Yovita Syafri

DAFTAR ISI

Halaman

DAFTAR TABEL	vii
---------------------------	------------

DAFTAR GAMBAR.....	viii
---------------------------	-------------

I. PENDAHULUAN.....	1
----------------------------	----------

1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan	4

II. TINJAUAN PUSTAKA.....	6
----------------------------------	----------

2.1 Sepeda.....	6
2.2 Peminjaman Sepeda	7
2.3 Sistem Informasi Penyewaan	8
2.4 <i>Smartphone</i>	8
2.5 Android.....	8
2.6 Flutter.....	9
2.7 <i>Single Sign-On</i>	9
2.8 <i>Application Programming Interface (API)</i>	10
2.9 <i>Javascript Object Notation (JSON)</i>	10
2.10 <i>Firebase</i>	11
2.11 <i>Agile Scrum Method</i>	11
2.12 <i>Unified Modeling Language</i>	12

2.13	<i>Unit Testing</i>	14
2.13.1	<i>Black Box Testing</i>	14
2.13.2	<i>User Experience Questionnaire</i>	15
2.14	Trello	16
2.15	Penelitian Terkait.....	16
2.15.1	Aplikasi Android “GamaRC” Untuk Mendukung Konsep Transportasi Hijau Sepeda Kampus – Studi Kasus UGM Yogyakarta	16
2.15.2	Sistem Peminjaman Sepeda Otomatis Berbasis QR-Code..	17
2.15.3	Implementasi Aplikasi Android Untuk Sistem Penyewaan Sepeda.....	18
2.15.4	UniCycle: An Android Application of Bike Sharing System in the Digital Campus	19
2.15.5	A Study on Cost-Effective and Eco-Friendly Bicycle Sharing System for Developing Countries.....	20
2.15.6	Uji Usability Pada Penggunaan Aplikasi Gowes di Telkom University	21
2.15.7	Penerapan Agile Scrum Pada Pengembangan Aplikasi Bimbingan Daring Skripsi Mahasiswa	22
2.15.8	Application Centric Cloud-Based Notification System Using Scrum Methodology.....	22
2.15.9	Agile Software Methodology with Scrum for Developing Quality Assurance System.....	23
2.15.10	Information System for Providing Food Services Based on Mobile Application Using Flutter	24
2.16	State of The Art.....	25
III. METODOLOGI PENELITIAN		27
3.1	Waktu dan Tempat Penelitian.....	27
3.2	Jadwal Penelitian.....	27
3.3	Alat Dalam Penelitian	28
3.3.1	Alat Penelitian.....	28
3.4	Tahapan Penelitian	29
3.4.1	Tahap Perancangan Sistem.....	30
3.4.2	Tahap Pengembangan Sistem	34

3.4.3 Tahap <i>Testing</i>	36
3.4.4 Tahap Analisis	36
3.4.5 Tahap Pelaporan.....	36
IV. HASIL DAN PEMBAHASAN	37
4.1 Hasil.....	37
4.1.1 Perancangan Sistem.....	37
4.1.2 Pengembangan Sistem	52
4.1.3 Pengujian Sistem	72
4.2 Pembahasan.....	82
4.2.1 Analisa Hasil <i>Black Box Testing</i>	82
4.2.2 Analisa Hasil User Experience Questionnaire.....	83
4.2.3 Penerapan Metode <i>Agile Scrum</i>	84
V. KESIMPULAN DAN SARAN	86
5.1 Kesimpulan.....	86
5.2 Saran.....	87
DAFTAR PUSTAKA	88
LAMPIRAN	92

DAFTAR TABEL

	Halaman
Tabel 1. Simbol dan keterangan pada <i>Use Case Diagram</i>	13
Tabel 2. Jadwal Penelitian.....	27
Tabel 3. Alat yang digunakan dalam penelitian.....	28
Tabel 4. Kebutuhan fungsional aplikasi.....	39
Tabel 5. Kebutuhan non-fungsional aplikasi	40
Tabel 6. <i>List User Story</i> dan <i>Backlog</i>	49
Tabel 7. Pembagian <i>backlog</i> selama <i>sprint</i>	52
Tabel 8. <i>Backlog Sprint 2</i>	54
Tabel 9. <i>Backlog</i> pada Sprint 3	58
Tabel 10. <i>Backlog</i> pada <i>sprint 4</i>	62
Tabel 11. Hasil pengujian <i>black box</i> untuk fitur otentikasi	72
Tabel 12. Hasil pengujian daftar <i>shelter</i> dan jumlah sepeda	73
Tabel 13. Hasil Pengujian Daftar Sepeda	74
Tabel 14. Hasil pengujian <i>black box</i> fitur peminjaman sepeda	75
Tabel 15. Hasil pengujian <i>blackbox</i> pada fitur status pinjam	76
Tabel 16. Hasil pengujian fitur pengembalian sepeda	78
Tabel 17. Hasil Pengujian <i>Availability Sistem</i>	79
Tabel 18. Data pengujian UEQ	81
Tabel 19. Hasil perhitungan data	81
Tabel 20. Hasil perhitungan nilai rata-rata.....	82

DAFTAR GAMBAR

	Halaman
Gambar 1. Sepeda [6].....	6
Gambar 2. Contoh Format JSON	10
Gambar 3. Alur Kerja Agile Scrum [17].....	12
Gambar 4. <i>Workflow</i> pada penelitian [24]	19
Gambar 5. Arsitektur sistem yang dikembangkan pada penelitian [25]	20
Gambar 6. <i>Storyboard</i> pada penelitian [29]	24
Gambar 7. Tahapan pada metode <i>Agile Scrum</i>	29
Gambar 8. Format <i>User Story</i>	33
Gambar 9. Arsitektur Sistem Peminjaman Sepeda	41
Gambar 10. <i>Use Case Diagram</i> Aplikasi Peminjaman Sepeda di Universitas Lampung	42
Gambar 11. <i>Activity Diagram</i> ketika <i>user login</i>	43
Gambar 12. <i>Activity diagram</i> melihat daftar sepeda pada shelter.....	44
Gambar 13. <i>Activity Diagram</i> Peminjaman Sepeda.....	45
Gambar 14. <i>Activity Diagram</i> saat melihat status peminjaman	46
Gambar 15. <i>Activity Diagram</i> ketika pengembalian sepeda	47
Gambar 16. <i>Activity Diagram logout</i>	48
Gambar 17. <i>User story</i> dan <i>backlog</i> pada <i>board</i> Trello.....	51
Gambar 18. (a) Halaman splash, (b) Halaman Login, (c) UI Halaman utama	53
Gambar 19. (a) UI Halaman Daftar Sepeda, (b) UI Halaman Detail Sepeda, (c) Halaman Utama.....	55
Gambar 20. (a) <i>Form email</i> atau <i>password</i> kosong, (b) <i>Email</i> atau <i>password</i> salah, (c) Akun tidak terdaftar	56
Gambar 21. (a) <i>Field password</i> ketika asterisk aktif, (b) <i>Field password</i> ketika	

asterisk non aktif	57
Gambar 22. <i>Firestore Database</i> untuk aplikasi UniBike	57
Gambar 23. Daftar <i>shelter</i> peminjaman	60
Gambar 24. Daftar sepeda yang tersedia.....	60
Gambar 25. (a) Halaman Status Peminjaman, (b) Halaman Profile, (c) Halaman Riwayat Peminjaman	61
Gambar 26. Tombol Lupa Password.....	61
Gambar 27. (a) Dialog konfirmasi peminjaman, (b) Dialog peminjaman berhasil, (c) Dialog peminjaman gagal, (d) Dialog pengguna memiliki denda pinjam, (e) Dialog pengguna sedang meminjam sepeda	63
Gambar 28. <i>Code Snippet</i> Peminjaman Sepeda	64
Gambar 29. (a) Dialog pengembalian sepeda, (b) Dialog memilih <i>shelter</i> pengembalian	65
Gambar 30. (a) Dialog pengembalian berhasil, (b) Dialog pengembalian gagal...	66
Gambar 31. Dialog ketika pengguna belum memilih <i>shelter</i>	66
Gambar 32. <i>Code Snippet</i> Pengembalian Sepeda	67
Gambar 33. <i>Code Snippet</i> Denda Pinjam.....	68
Gambar 34. <i>Countdown</i> waktu peminjaman sepeda	69
Gambar 35. (a) Informasi sisa waktu peminjaman dan denda waktu peminjaman, (b) Halaman status peminjaman.....	69
Gambar 36. (a) Informasi tidak ada koneksi <i>internet</i> , (b) Dialog jam operasional peminjaman sepeda	70
Gambar 37. UEQ yang digunakan	80
Gambar 38. Grafik hasil perbandingan <i>benchmark data set</i>	83
Gambar 39. Korelasi Antar Pertanyaan	84

I. PENDAHULUAN

1.1 Latar Belakang

Universitas Lampung (Unila) dikenal dengan sebutan Kampus Hijau (*Green Campus*), dengan beberapa programnya yang sudah berjalan seperti Ruang Hijau, Lingkungan Hijau, Energi Hijau, Transportasi Hijau, Pakar Hijau, Kegiatan Hijau, Kelas Hijau dan Riset Hijau. Sejak 2020, sudah ada 3 program Transportasi Hijau, yaitu Peresmian *Car Free Day* dan Jalur Sepeda, *Shuttle Bus* dan Peluncuran Mobil Listrik Universitas Lampung [1]. Sepeda adalah transportasi yang sangat umum digunakan oleh semua orang dari berbagai kalangan usia dan saat ini bersepeda menjadi sebuah kegiatan maupun olahraga yang cukup terkenal luas di kalangan masyarakat Indonesia. Sepeda merupakan moda transportasi tanpa mesin sehingga tidak memerlukan bahan bakar dan tidak menimbulkan emisi karbon.

Kampus Universitas Lampung dengan luas 63,5 hektar dan terbagi menjadi 8 fakultas, yaitu Fakultas Ekonomi dan Bisnis, Fakultas Hukum, Fakultas Keguruan dan Ilmu Pendidikan, Fakultas Pertanian, Fakultas Teknik, Fakultas Ilmu Sosial dan Ilmu Politik, Fakultas Matematika dan Ilmu Pengetahuan Alam dan Fakultas Kedokteran [2]. Dalam hal ini untuk bepergian ke fakultas lain yang jaraknya cukup jauh, transportasi yang disediakan dan dapat digunakan oleh seluruh *civitas* akademika Unila adalah *Shuttle Bus*. Universitas Lampung sudah mulai menggunakan shuttle bus sebagai alat transportasi yang diperuntukkan untuk *civitas* Universitas Lampung dan pengunjung yang hendak berkunjung ke Universitas Lampung [3]. Tetapi dengan jumlah *Shuttle Bus* yang beroperasi, mahasiswa seringkali mendapati *Shuttle Bus* penuh sehingga tidak dapat menggunakannya dan harus menunggu *Shuttle Bus* berikutnya.

Sepeda adalah salah satu fasilitas pendukung yang sudah berjalan di beberapa kampus di Indonesia seperti Institut Pertanian Bogor [4]. Sepeda kampus tentunya akan menunjang program Transportasi Hijau di Universitas Lampung karena akan membantu mengurangi dampak polusi udara akibat kendaraan bermotor di lingkungan kampus. Peminjaman sepeda kampus ini akan memudahkan civitas akademika untuk bepergian di sekitar Universitas Lampung dan membantu mahasiswa yang sering mendapati *Shuttle Bus* penuh sehingga kesulitan mencapai fakultas yang dituju.

Sistem peminjaman sepeda di beberapa kampus masih dilakukan secara manual, dengan melihat Studi Kasus Peminjaman Sepeda di Institut Pertanian Bogor (IPB), dimana mahasiswa yang ingin meminjam sepeda harus mendatangi *shelter* peminjaman dan menyerahkan Kartu Tanda Mahasiswa (KTM) sebagai jaminan peminjaman [4]. Pencatatan data peminjaman pun masih dilakukan secara manual di setiap *shelter* peminjaman sepeda sehingga tidak ada laporan terpusat mengenai penggunaan fasilitas sepeda kampus. Oleh karena itu, penelitian ini akan mengembangkan sistem peminjaman sepeda di Universitas Lampung dan memperbaiki sistem peminjaman secara manual, yaitu dengan menggunakan aplikasi peminjaman sepeda online berbasis *android* yang mana setiap data sepeda maupun data peminjaman akan tercatat secara terpusat.

Pada Januari 2022, total populasi di Indonesia mencapai 277,7 juta jiwa dengan 370,1 juta pengguna *mobile phone*. Sehingga, dapat diartikan bahwa banyak masyarakat Indonesia yang memiliki lebih dari satu *mobile phone*. Hal tersebut menjelaskan angka pengguna *mobile phone* lebih tinggi jika dibandingkan dengan total populasi di Indonesia. Dalam kurun waktu dua tahun dari tahun 2021, pengguna *mobile phone* meningkat sebanyak 13 juta atau sekitar 3,6 persen [5]. Sehingga potensi pemanfaatan android untuk digunakan sebagai sistem peminjaman sepeda sangat tinggi.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah bagaimana membangun sistem peminjaman sepeda *online* berbasis android untuk Universitas Lampung dengan teknologi *framework Flutter* menggunakan metode *Agile Scrum*.

1.3 Tujuan Penelitian

Tujuan dari penelitian adalah membangun sistem peminjaman sepeda *online* berbasis android untuk Universitas Lampung.

1.4 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat dalam mempermudah *civitas* akademika Universitas Lampung dalam melakukan peminjaman sepeda secara *online* dengan menyediakan data ketersediaan sepeda yang dapat dipinjam.

1.5 Batasan Masalah

Batasan masalah dari penelitian ini adalah :

1. Kebutuhan sistem berdasarkan informasi yang diberikan oleh Pusat Unggulan Institusi Perguruan Tinggi *Green Technology* Universitas Lampung.
2. Sistem otentikasi disimulasikan berdasarkan *Single Sign-On* Universitas Lampung.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan pada skripsi berikut adalah:

1. BAB I: PENDAHULUAN

BAB I merupakan pendahuluan penelitian yang berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, manfaat penelitian dan sistematika penulisan.

2. BAB II: TINJAUAN PUSTAKA

BAB II ini merupakan tinjauan pustaka membahas mengenai dasar teori, beberapa istilah bertujuan sebagai sumber untuk memahami penelitian mengenai *Perancangan Aplikasi Peminjaman Sepeda Di Universitas Lampung Berbasis Android Menggunakan Metode Agile Scrum*. Bab II berisi tentang pengertian sepeda, peminjaman sepeda, sistem informasi penyewaan, *smartphone, android, flutter, single sign-on, application programming interface, Javascript object notation, firebase, agile scrum method, unit testing, black box testing, unified modeling language*, Trello dan penelitian terkait.

3. BAB III: METODOLOGI PENELITIAN

BAB III yang merupakan metodologi penelitian yang berisi tempat dan waktu penelitian, jadwal penelitian, alat dan bahan penelitian dan tahapan penelitian dengan metode *Agile Scrum*. Dimana penelitian dimulai dari requirements gathering, lalu penentuan *User Story*, penentuan *Definition of Done*, perancangan, *Sprint Planning, Sprint, Daily Stand Up Meeting, Sprint Review, Sprint Retrospective* dan *testing*.

4. BAB IV: HASIL DAN PEMBAHASAN

BAB IV berisi hasil dan pembahasan sesuai dengan tahap penelitian, yaitu: tahap perancangan sistem tahap pengembangan sistem, dan tahap pengujian.

5. BAB V: KESIMPULAN DAN SARAN

BAB V berisi kesimpulan dari penelitian yang telah dilakukan dan saran-saran untuk penelitian selanjutnya.

II. TINJAUAN PUSTAKA

2.1 Sepeda

Sepeda merupakan moda transportasi darat beroda dua, tanpa mesin yang perlu dikayuh pedalnya untuk bergerak. Awalnya tujuan diciptakan sepeda adalah untuk alat transportasi bagi manusia. Tetapi, saat ini sepeda sudah berevolusi menjadi salah satu cabang olahraga dan menjadi gaya hidup menyehatkan yang banyak disukai masyarakat [6]. Perkembangan sepeda di Indonesia mulai menjadi perhatian masyarakat dan pemerintah di tahun 2010, pada saat dibangunnya jalur khusus sepeda di kota Bandung. Sepeda juga memiliki banyak jenis dengan keunggulan nya masing-masing seperti pada gambar 1 yaitu jenis sepeda gunung yang biasanya digunakan ketika ingin bersepeda di medan yang sulit dilewati.



Gambar 1. Sepeda [6]

2.2 Peminjaman Sepeda

Peminjaman sepeda merupakan layanan dimana masyarakat dapat meminjam sepeda yang disediakan dengan adanya syarat dan ketentuan. Peminjaman sepeda adalah salah satu dari banyaknya fasilitas yang disediakan oleh beberapa kampus bagi mahasiswa maupun pengunjung [7]. Peminjaman sepeda kampus sudah banyak tersedia di beberapa universitas di Indonesia seperti di Universitas Indonesia dan Universitas Gadjah Mada. Fasilitas peminjaman sepeda mempermudah civitas akademika bepergian dari satu fakultas ke fakultas lain. Selain itu dengan ada nya peminjaman sepeda secara gratis dapat mengurangi kendaraan bermotor di sekitar kampus. Peminjaman sepeda umumnya dilakukan secara tradisional dengan cara mendatangi *shelter* peminjaman sepeda kemudian mencatat data diri atau meninggalkan kartu tanda pengenalan sebagai jaminan peminjaman [5]. Tetapi dengan kemajuan teknologi, peminjaman sepeda juga dapat dilakukan secara *online* melalui aplikasi peminjaman sepeda yang dapat diunduh di *smartphone* pengguna.

Penelitian ini mengembangkan sebuah sistem peminjaman sepeda secara *online* di Universitas Lampung menggunakan aplikasi berbasis android. Dengan tersedianya informasi sepeda yang dapat dipinjam di setiap *shelter*. Terdapat 8 *shelter* peminjaman di setiap fakultasnya, pengguna dapat melakukan pinjaman secara online sehingga tidak perlu datang ke *shelter* jika ternyata tidak ada sepeda yang dapat dipinjam. Jumlah sepeda pada yang tersedia pada *shelter* peminjaman minimal 15 sepeda. Proses pengembalian sepeda pun dilakukan secara *online* melalui aplikasi, memperbaiki sistem peminjaman sepeda konvensional dimana peminjam harus datang ke *shelter* yang sama. Pada penelitian ini pengembalian sepeda dilakukan dengan cara memilih *shelter* mana saja tempat pengguna ingin mengembalikan sepedanya.

2.3 Sistem Informasi Penyewaan

Sistem informasi merupakan sekumpulan komponen berupa manusia, prosedur kerja, informasi dan teknologi (seperti komputer). Kemudian, komponen diorganisasikan secara sistematis membentuk aliran informasi yang akan membantu dalam proses pengambilan keputusan untuk mencapai tujuan dalam sebuah organisasi [8]. Sistem informasi penyewaan menyediakan aliran informasi dalam bekerjanya sistem penyewaan secara jelas. Dalam terbentuknya sistem informasi penyewaan, diperlukan beberapa proses seperti pengumpulan data, mengelola data yang tersimpan, sampai didapatkan informasi yang mudah dimengerti oleh pengguna. Pada sistem informasi penyewaan umumnya memerlukan data diri orang yang menyewa, data mengenai barang yang disewa, dan data kapan serta dimana dilakukan penyewaan.

2.4 Smartphone

Smartphone atau telepon pintar merupakan jenis telepon genggam dengan banyak kemampuan yang tidak hanya digunakan sebagai alat berkomunikasi saja. Banyak manfaat yang diberikan oleh *smartphone*, seperti contohnya sebagai sarana berbisnis dengan bantuan *platform e-commerce* yang tersedia. Selain itu, *smartphone* memberikan kemudahan dalam melakukan *browsing*, kegiatan perbankan, membaca buku melalui *e-book*, dan lainnya. Terdapat banyak sistem operasi yang dapat dijalankan pada *smartphone*, seperti *android*, *iOS*, *RIM BlackBerry*, *Windows Mobile*, *Linux*, *Palm*, *WebOS*, dan *Symbian OS* [9].

2.5 Android

Android adalah sistem operasi dan *platform* pemrograman yang dirancang untuk perangkat seluler dan ponsel cerdas. Android dikembangkan oleh Google dan dapat

dijalankan di berbagai perangkat dari produsen yang bervariasi seperti perangkat seluler dan *tablet* [10]. Android menyediakan *platform* terbuka bagi para pengembangnya untuk dapat membuat aplikasi sendiri. Dengan sifat nya yang *open source*, menjadikan android memiliki banyak bahasa pemrograman dan *framework* seperti Java, Kotlin, Dart, React Native dan Flutter.

2.6 Flutter

Flutter merupakan *framework multi-platform* milik Google untuk membangun aplikasi android maupun *iOS* yang modern tetapi masih bersifat *native* atau asli. Flutter tidak hanya dapat membuat sebuah aplikasi untuk smartphone. Tetapi, Google juga sedang mengembangkan Flutter untuk *Website*. Flutter adalah *framework* yang bersifat *open source* sehingga tidak hanya Google yang dapat berkontribusi mengembangkannya. Terdapat komunitas pengguna Flutter yang juga berkontribusi mengembangkannya. Bahasa yang digunakan oleh Flutter adalah Dart. Pada Dart, kode dikompilasi dibuat ke dalam kode *native* sesuai dengan sistem operasi nya dan kode JavaScript yang siap digunakan untuk *Website* [11].

2.7 Single Sign-On

Single sign-on merupakan metode otentikasi yang memungkinkan pengguna untuk mengakses beberapa aplikasi maupun *platform* dengan hanya satu sistem *login* atau satu set kredensial [12]. Beberapa perusahaan atau organisasi memiliki sistem *Single Sign-On* yang memudahkan anggota nya. Salah satu contoh *single sign-on* yang sering kali digunakan adalah Google *credentials* yang diimplementasikan untuk masuk ke Google *Mail*, Google *Drive*. Bahkan, beberapa platform dan aplikasi lain banyak yang memanfaatkan Google *single sign-on* sebagai metode otentikasi nya.

2.8 Application Programming Interface (API)

Application Programming Interface merupakan sebuah teknologi untuk bertukar atau mendokumentasikan data yang tersedia untuk perangkat lunak. *API* berkarakteristik independen atau tidak memiliki keterkaitan dengan teknologi atau bahasa pemrograman yang digunakan. Dengan *API*, aplikasi lain dapat mengakses data tanpa harus mengimplementasikan objek dan prosedur yang mendasarinya [13]. *API* saat ini menjadi pusat dari banyaknya arsitektur perangkat lunak karena *API* menyediakan kecepatan dan kemudahan dalam pertukaran data sehingga memungkinkan untuk digunakan pada semua sistem perangkat lunak *modern*.

2.9 Javascript Object Notation (JSON)

Javascript Object Notation merupakan format dalam pertukaran data yang bersifat ringan serta mudah dibaca maupun ditulis oleh manusia. *JSON* merupakan format teks yang tidak tergolong dalam Bahasa pemrograman apapun tetapi mudah untuk diterjemahkan dan dibuat oleh mesin. Meskipun *JSON* tidak terikat pada bahasa pemrograman lain, tetapi penulisan *JSON* mirip dengan bahasa pemrograman keluarga C seperti C++, C#, Java dan lain nya yang membuat *JSON* sangat cocok sebagai bahasa pertukaran data [14].



```

{
  "documents": [
    {
      "name": "projects/unibike-13788/databases/(default)/documents/data_sepeda/OVORK2Hau2sL5v9Vvyja",
      "fields": {
        "foto_sepeda": {
          "stringValue": "https://firebasestorage.googleapis.com/v0/b/unibike-13788.appspot.com/o/data_sepeda%2F%OVORK2Hau2sL5v9Vvyja?alt=media&token=c49f940d-1c96-4a28-aa91-ff41e42b963c"
        },
        "jenis_sepeda": {
          "stringValue": "Sepeda Lipat"
        },
        "merk_sepeda": {
          "stringValue": "United"
        },
        "deskripsi": {
          "stringValue": "Berwarna hitam"
        },
        "status": {
          "stringValue": "Sepeda Lipat"
        },
        "tahun": {
          "stringValue": "2021"
        },
        "createTime": "2022-08-05T17:37:39.351876Z",
        "updateTime": "2022-08-05T08:07:09.950765Z"
      }
    }
  ]
}

```

Gambar 2. Contoh Format JSON

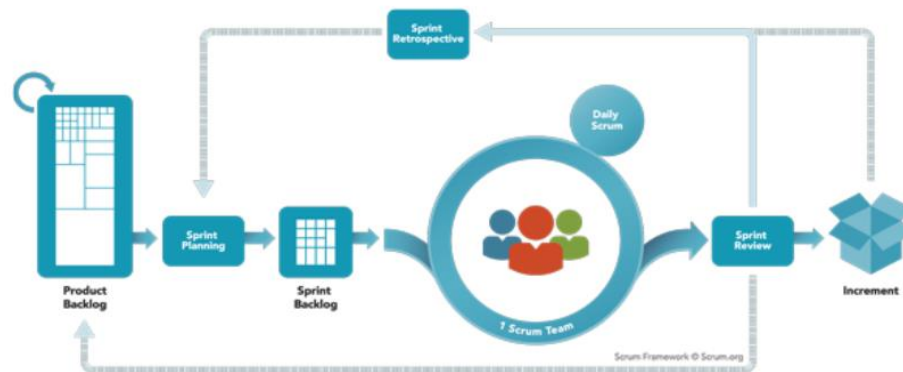
Gambar 2 menampilkan contoh format JSON dimana semua object berada di dalam kurung kurawal dan memiliki pasangan *key* dan *value*. Setiap *key* diikuti oleh titik dua dan dipisahkan oleh koma.

2.10 Firebase

Firebase adalah platform milik Google untuk mengembangkan aplikasi yang membantu developer dalam membangun dan memelihara perangkat lunak. *Firebase* menyimpan data dalam bentuk format *JavaScript Object Notation (JSON)* yang tidak memerlukan kueri untuk menambah, memperbaharui, menghapus dan memasukkan data ke dalamnya. *Firebase* tergolong sebagai sistem *backend* atau database yang mudah digunakan untuk menyimpan data. Google *Firebase* memiliki beberapa layanan diantaranya seperti *Firebase Authentication* yang menyediakan layanan otentikasi menggunakan beberapa kredensial, *Cloud Firestore Database*, *Firebase Storage*, dan lainnya [15].

2.11 Agile Scrum Method

Agile merupakan metode pengembangan perangkat lunak yang menerapkan iterasi (bertahap dan berulang). *Scrum* merupakan salah satu metode *agile*. *Agile Scrum* adalah metode kerja untuk mengembangkan dan mengelola perangkat lunak. Kerangka kerja *Scrum* terdiri dari *Scrum Team* dengan peran-peran yang berbeda, acara-acara dan aturan-aturan terkait [16]. Pada metode *scrum* terdapat beberapa tahapan yang dilakukan, yaitu *Sprint*, *Sprint Planning*, *Daily Scrum* atau *Daily Standup Meeting*, *Sprint Review* dan *Sprint Retrospective*. Masa *Sprint* pada metode *Scrum* biasa dilakukan selama satu sampai empat minggu.



Gambar 3. Alur Kerja Agile Scrum [17]

Sprint diawali dengan pendefinisian *product backlog*. *Sprint Planning* yang merupakan kegiatan perencanaan *sprint backlog* yang harus diselesaikan selama *sprint*. Metode *Agile Scrum* mengontrol kegiatan dengan adanya *daily scrum* yang merupakan kegiatan dimana setiap anggota menyampaikan hasil kerjanya selama satu hari yang lalu dan melakukan perkiraan pekerjaan yang akan dilakukan selanjutnya. *Sprint Review* dan *Sprint Retrospective* diselenggarakan pada akhir *sprint*. *Sprint Review* merupakan pendemonstrasian pekerjaan yang diselesaikan oleh tim, menjelaskan masalah yang dialami selama *sprint* dan meninjau *sprint backlog* untuk *sprint* selanjutnya. Sementara *Sprint Retrospective* adalah kegiatan untuk merefleksikan hasil kerja dan kinerja masing-masing [16].

2.12 Unified Modeling Language

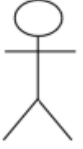
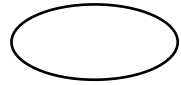

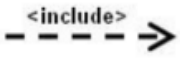
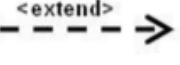
Unified Modelling Language atau UML merupakan sebuah bahasa standar yang bertujuan untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak [18]. UML dapat digunakan untuk semua metode pengembangan dan tahapan siklus hidup perangkat lunak. UML merupakan sekumpulan diagram untuk menggabungkan praktik terbaik dari perangkat lunak yang ingin dikembangkan dengan standar yang biasanya dilakukan. Dalam UML, diagram terbagi menjadi dua yaitu *Structural Diagram* dan *Behavioral Diagram* dan di dalamnya terdapat banyak

diagram seperti *class*, *object*, *component*, *use case*, *sequence*, *activity* dan lainnya.

a. *Use Case Diagram*

Use Case Diagram adalah sebuah diagram yang menggambarkan fungsional dari sebuah sistem, serta hubungan antara aktor dengan *use case* di dalam sistem tersebut. *Use case* merupakan sebuah pekerjaan tertentu yang dapat terjadi dalam sebuah sistem dan penting untuk memvisualisasikan perilaku suatu objek [18]. Dengan adanya *use case diagram*, penyajian gambaran tentang bagaimana sebuah objek dapat bekerja pada sistem perangkat lunak jadi lebih mudah. Adapun komponen pada *use case diagram* adalah sebagai berikut:

Tabel 1. Simbol dan keterangan pada *Use Case Diagram*

Nama	Simbol	Keterangan
Aktor		Merepresentasikan seseorang atau objek yang berinteraksi dengan sistem tetapi tidak memegang kendali atas <i>use case</i> .
<i>Use Case</i>		Merepresentasikan gambaran fungsional sebuah sistem.
Relasi Asosiasi		Merepresentasikan sebuah interaksi antara aktor dengan <i>use case</i> tertentu.
Relasi <i>Include</i>		Merepresentasikan relasi <i>use case</i> tambahan dimana <i>use case</i> yang ditambahkan membutuhkan <i>use case</i> ini untuk menjalankan fungsinya.
Relasi <i>Extend</i>		Merepresentasikan <i>use case</i> tambahan dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri. Arah panah mengarah pada <i>use case</i> yang ditambahkan.

Pada tabel 1 dapat dilihat terdapat 5 simbol yang digunakan dalam *use case diagram* yaitu aktor, *use case*, relasi asosiasi dan relasi *include* juga *extend*.

b. Activity Diagram

Activity Diagram merupakan diagram yang memvisualisasikan aktivitas pada sistem, mulai dari awal sistem berjalan, pilihan yang mungkin akan terjadi, dan bagaimana akhir dari sistem tersebut. *Activity diagram* adalah *state diagram* khusus, dimana banyak dari *state* nya merupakan aksi dengan munculnya sebuah transisi akibat *state* yang sebelumnya selesai [18].

2.13 Unit Testing

Unit testing merupakan sebuah metode dalam pengembangan perangkat lunak untuk memvalidasi bahwa unit dari sebuah kode yang dikembangkan berperilaku sesuai dengan yang diinginkan. Dalam *Unit testing*, kode akan diuji diharapkan untuk menghasilkan luaran yang sama setiap kali pengujian dieksekusi. *Unit testing* memerlukan potongan kode lain untuk memvalidasi perilaku dari unit tersebut. Jika hasil tidak sesuai dengan yang diinginkan, maka pengujian gagal. Pengujian unit yang hanya menguji fungsionalitas satu metode disebut dengan *System Under Test (SUT)* atau *Code Under Test (CUT)*. *Unit testing* memberikan banyak manfaat seperti *developer* dapat menemukan *bug* lebih awal. Sehingga, mempercepat pengembangan dan mencegah adanya *bug* yang sama serta membuat *developer* lebih memahami penggunaan kode nya [19].

2.13.1 Black Box Testing

Black box testing atau biasa disebut sebagai pengujian fungsional dan pengujian perilaku, memfokuskan pengujian dalam menentukan apakah program dapat

melakukan tugas berdasarkan kebutuhan nya [20]. *Black box testing* dapat menemukan kesalahan pada perilaku dalam kode di beberapa kategori seperti fungsionalitas yang salah atau hilang, tampilan *error*, kesalahan dalam struktur data yang tertera pada tampilan, dan lain-lain. Salah satu teknik pengujian pada *black box testing* adalah *Decision Tables*. Teknik *decision table* digunakan untuk menguji kombinasi perilaku yang berbeda dari beberapa input yang mirip. Tabel keputusan menampilkan ketergantungan antara masukan yang berbeda dengan keluaran nya. Hasil dari *decision table testing* tersebut kemudian disajikan dalam sebuah tabel yang mudah dimengerti yang berisikan satu set kondisi yang menjadi masukan dan satu set efek atau hasil yang merupakan keluaran nya.

2.13.2 User Experience Questionnaire

User Experience Questionnaire adalah salah satu metode pengujian menggunakan kuesioner yang berguna untuk mengetahui tingkat user experience dari produk yang dikembangkan [21]. Terdapat 6 skala penilaian dari total 26 pertanyaan yang perlu dijawab pengguna dengan 7 pilihan jawaban. Adapun 6 skala penilaian tersebut, yaitu:

1. *Attractiveness* (Daya Tarik): kesan secara keseluruhan apakah pengguna menyukai produk tersebut atau tidak?
2. *Perspicuity* (Kejelasan): seberapa mudah produk tersebut untuk dipahami dan dipelajari oleh pengguna?
3. *Efficiency* (Efisiensi): mengetahui apakah pengguna dapat menyelesaikan tugasnya dengan mudah dan tidak kesusahan atau membutuhkan bantuan?
4. *Dependability* (Ketepatan): penilaian untuk mengetahui apakah pengguna dapat mengendalikan interaksi dengan produk tersebut atau pengguna dengan mudah terarah untuk melakukan interaksi dengan produk?

5. *Stimulation* (Stimulasi): secara keseluruhan apakah produk tersebut menarik dan memotivasi untuk digunakan?
6. *Novelty* (Kebaruan): Seberapa inovatif dan kreatif dan apakah produk tersebut berhasil menarik minat pengguna?

2.14 Trello

Trello merupakan sebuah *visual tools* yang memberi kemudahan bagi anggota tim untuk mengelola proyeknya. Trello dapat mengatur alur kerja, siapa yang akan mengerjakannya, pelacakan tugas semua anggota tim untuk mengetahui sejauh mana tim bekerja, dan tenggat waktu pengerjaan. Trello berbentuk sebuah papan proyek yang dapat diakses menggunakan internet dengan mudah, fleksibel dan gratis [22].

2.15 Penelitian Terkait

2.15.1 Aplikasi Android “GamaRC” Untuk Mendukung Konsep Transportasi Hijau Sepeda Kampus – Studi Kasus UGM Yogyakarta

Penelitian yang dilakukan oleh Agung Prayogo, Arif Kusumawanto dan Rudy Hartanto pada tahun 2018 membahas tentang Aplikasi Android “GamaRC” Untuk Mendukung Konsep Transportasi Hijau Sepeda Kampus di Universitas Gadjah Mada Yogyakarta. Peminjaman sepeda pada saat itu di Universitas Gadjah Mada (UGM) masih manual dengan cara mendatangi stasiun sepeda kampus dan membawa kartu identitas mahasiswa. Kemudian, sebagai solusi dari sistem peminjaman manual, dikembangkan sebuah sistem peminjaman dalam bentuk aplikasi berbasis android dengan metode pengembangan SDLC *Waterfall*. Dimana prosedur peminjaman yang dikembangkan menjadi sebuah sistem reservasi *online*. Lalu, sistem pengembalian sepeda menggunakan *scan barcode* sebagai tanda bukti bahwa pengguna telah

mengembalikan sepeda [23].

Pada penelitian ini pengguna aplikasi harus memiliki akun terlebih dahulu. Ketika membuat akun, pengguna diminta untuk mengisi data seperti nama lengkap, email dan lainnya. Hal ini akan diubah dalam penelitian yang dilakukan yaitu *Perancangan Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android*. Pada penelitian yang dilakukan metode login menggunakan *Single Sign-On* dengan kredensial yang diberikan oleh Universitas Lampung, sehingga pengguna tidak perlu lagi melakukan pendaftaran akun baru.

2.15.2 Sistem Peminjaman Sepeda Otomatis Berbasis QR-Code

Pada penelitian yang dilakukan oleh Reinaldi Prasetya dan Heru Supriyono pada tahun 2019 membahas mengenai Sistem Peminjaman Sepeda Otomatis Berbasis *QR-Code*. Pada Universitas Muhammadiyah Surakarta, sistem peminjaman sepeda masih menerapkan sistem tradisional. Peminjam harus mengambil kunci pada petugas dan menulis data peminjam di kertas. Berdasarkan hal tersebut, kemudian dikembangkan sistem berupa aplikasi peminjaman yang digunakan oleh pengguna dan sebuah *dashboard* berbasis *website* untuk mengelola data yang digunakan oleh admin. Aplikasi yang dikembangkan menggunakan *SDLC Waterfall* ini digunakan untuk membaca *QR-code* pada sepeda dan mengharuskan pengguna untuk memiliki akun. Terdapat beberapa menu yang dapat dipilih pengguna setelah *login*, ada *menu scan barcode*, daftar sepeda, tata cara peminjaman, tentang aplikasi, *update* akun, dan *logout*. Menu *scan barcode* berfungsi jika pengguna ingin meminjam sepeda. Pengguna datang ke tempat peminjaman lalu *scan barcode* yang tersedia di kunci sepeda. Tetapi *scan barcode* ini hanya dapat dilakukan oleh pengguna yang akun nya sudah terverifikasi dan tidak memiliki denda peminjaman. Menu daftar sepeda hanya menampilkan daftar sepeda yang ada dengan status apakah sepeda tersebut tersedia atau tidak tersedia. Sementara pada *dashboard*, ada beberapa menu diantaranya daftar pengguna, daftar admin, daftar sepeda, data peminjaman dan data denda peminjam.

Pada *dashboard*, admin dapat mengubah, melihat riwayat dan menghapus data pengguna [24].

Penelitian yang dilakukan oleh Reinaldi Prasetya dan Heru Supriyono telah berhasil mengembangkan sistem peminjaman sepeda yang efektif. Namun, *QR Code* yang digunakan pada sepeda tidak berubah, sehingga setiap orang dapat melakukan pindai *QR Code* yang sama di waktu yang sama. Lalu ketika seorang pengguna memiliki denda peminjaman, maka pengguna tidak dapat meminjam sepeda sampai denda terbayarkan. Hal ini menjadi perbaikan dalam penelitian yang dilakukan. Sistem yang dikembangkan tidak berbayar, tetapi menerapkan denda jam pinjam atau pembatasan waktu pinjam.

2.15.3 Implementasi Aplikasi Android Untuk Sistem Penyewaan Sepeda

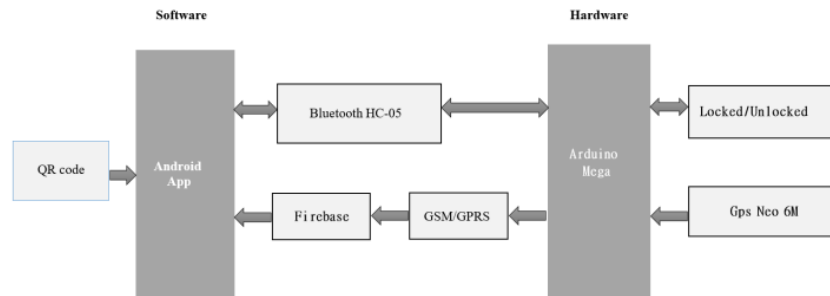
Pada penelitian yang dilakukan oleh Aprilia Christyana T.A membahas mengenai Implementasi Aplikasi Android untuk Sistem Penyewaan Sepeda. Penggunaan jasa penyewaan sepeda sudah banyak saat ini, sehingga diperlukannya suatu sistem yang dapat menunjang penyewaan sepeda. Dibutuhkan sistem yang menyediakan kemudahan, ketepatan, kecepatan dalam memberikan informasi kepada pengguna dan pihak yang membutuhkan. Dengan adanya aplikasi android, diharapkan dapat mempermudah usaha sewa sepeda dan membantu konsumen memperoleh informasi seputar hal yang diperlukan. Sistem penyewaan yang dikembangkan pada penelitian ini mengharuskan konsumen untuk datang ke tempat penyewaan sepeda terlebih dahulu. Lalu, karyawan akan membantu proses pendaftaran akun untuk konsumen. Setelah masuk ke aplikasi, barulah konsumen dapat melihat jenis sepeda beserta biaya sewanya. Penelitian ini tertuju pada sistem informasi bagi pemilik usaha penyewaan sepeda, tetapi masih kurang efisien bagi pengguna [25].

Sistem transaksi penyewaan sepeda ini akan diperbaiki dalam penelitian yang akan dilakukan yaitu *Perancangan Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android*. Dengan memberikan informasi sepeda yang tersedia tanpa

perlu mendatangi shelter peminjaman.

2.15.4 UniCycle: An Android Application of Bike Sharing System in the Digital Campus

Rozaki M.A.A, Zakaria N.H, Abdalla J.A membahas tentang pengembangan sebuah aplikasi untuk sistem *bike sharing* di kampus berbasis android. Transportasi umum yang disediakan kampus adalah bis kampus. Bis kampus memiliki kekurangan yaitu jadwal yang tidak ideal bagi mahasiswa. Sehingga, peminjaman sepeda menjadi pilihan alternatif untuk mahasiswa bepergian di dalam kampus. Aplikasi dikembangkan menggunakan *Appy Builder*, dimana aplikasi berfungsi untuk proses membuka atau mengunci sepeda yang terhubung pada Arduino Mega. Aplikasi juga digunakan untuk proses *sign in* dan *sign up*, melakukan pembayaran menggunakan *e-wallet*, dan melacak lokasi sepeda [26].



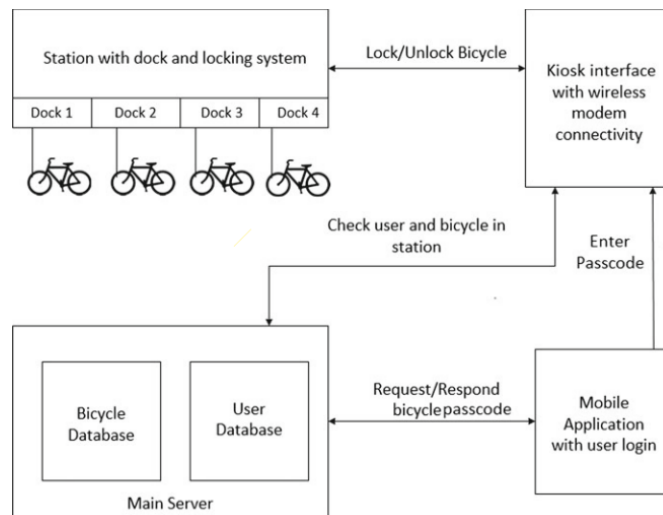
Gambar 4. *Workflow* pada penelitian [24]

Berdasarkan gambar 4, pengguna akan diminta untuk mengaktifkan *Bluetooth* agar terhubung dengan *Arduino mega*. Pada layar *smartphone* akan menampilkan *QR Code*. Ketika pengguna melakukan *scan QR code*, sepeda akan terbuka kuncinya dan mengaktifkan GPS yang datanya akan disimpan pada Firebase. Aplikasi kemudian akan mengambil data GPS dari Firebase dan menampilkan lokasi dari sepeda tersebut. Sistem *bike sharing* ini memperbolehkan pengguna menghentikan peminjaman dan memarkirkan sepeda dimana saja dan kapan saja dalam artian peminjaman sudah

selesai. Hal ini kurang efektif bagi pengguna lain karena lokasi sepeda tidak terpusat sehingga sulit untuk mahasiswa lain meminjam nya.

2.15.5 A Study on Cost-Effective and Eco-Friendly Bicycle Sharing System for Developing Countries

Pada penelitian yang dilakukan oleh Larsson Bajracharya, Tirta Mulya, Ayi Purbasari dan Mintae Hwang membahas mengenai perubahan infrastruktur transportasi. Perubahan ini mendukung lingkungan yang lebih hijau dan memperbanyak transportasi ramah lingkungan. Kemudian, diputuskan pengembangan sebuah sistem *bike sharing*. Sistem peminjaman berdasarkan registrasi *online* dan *monitoring* secara *real-time* menggunakan aplikasi pada *smartphone* [27].



Gambar 5. Arsitektur sistem yang dikembangkan pada penelitian [25]

Berdasarkan gambar 5, aplikasi yang dikembangkan akan memberikan kode sandi untuk membuka kunci sepeda yang ingin dipinjam. Sebelum meminjam, pengguna harus memiliki akun dengan *id* dan *password* yang valid. Setelah itu pengguna dapat memesan sepeda yang diinginkan terlepas dari tempat dan waktu. Peminjaman berlangsung selama 15-20 menit. Aplikasi menyediakan pembayaran *online* dan terintegrasi dengan *Google maps* untuk membantu pengguna mengetahui

letak stasiun peminjaman sepeda terdekat. Pada stasiun peminjaman sepeda akan terdapat mesin Kiosk yang akan menampilkan informasi sepeda yang tersedia [27].

Dalam penelitian ini data sepeda ditampilkan pada sebuah mesin IoT yaitu Kiosk dan penggunaan passcode yang memiliki kelemahan yaitu dapat memicu adanya penyerang ketika terjadi pertukaran data. Sehingga dikhawatirkan adanya kebocoran data.

2.15.6 Uji *Usability* Pada Penggunaan Aplikasi Gowes di Telkom University

Pada penelitian yang dilakukan oleh Asti Siti Badriah dan Puspita Kencana Sari pada tahun 2019 membahas mengenai Uji *Usability* Pada Penggunaan Aplikasi Gowes di Telkom *University*. Dengan adanya *bike sharing* yang dapat membantu mengatasi masalah transportasi saat ini, lalu diuji penggunaan dari aplikasi *mobile bike sharing* yaitu “Gowes”. Gowes merupakan aplikasi layanan berbagi sepeda berbasis android dan iOS yang dimiliki oleh PT. Surya Teknologi Perkasa (STP). Penelitian ini dilakukan dengan metode kuantitatif untuk mengukur *usability* aplikasi Gowes terhadap dua hal, yaitu gender dan jurusan di Telkom *University*. Pengujian juga dilakukan berdasarkan 7 komponen yaitu *effectiveness*, *efficiency*, *satisfaction*, *learnability*, *memorability*, *errors* dan *cognitive load* untuk mewujudkan inovasi model *bike sharing* yang sesuai dengan kecanggihan teknologi [28].

Hasil penelitian ini pada beberapa komponen seperti *effectiveness*, *efficiency*, *satisfaction* dan *learnability* dianggap baik. Pada poin *learnability*, seharusnya aplikasi memiliki kemudahan pengoperasian dan mudah dimengerti oleh pengguna saat pertama kali menggunakan aplikasi, Lalu, untuk *memorability*, aplikasi harus memiliki tingkat kemudahan bagi pengguna untuk mengingat dan dapat mengulang kembali proses pengembalian tanpa mempelajarinya kembali. Hal ini akan meningkatkan kegunaan aplikasi yang akan dikembangkan dalam penelitian *Perancangan Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android*.

2.15.7 Penerapan *Agile Scrum* Pada Pengembangan Aplikasi Bimbingan Daring Skripsi Mahasiswa

Penelitian yang dilakukan oleh Meta Amalya Dewi dan Rafi Irham pada tahun 2021 mengenai Penerapan *Agile Scrum* Pada Pengembangan Aplikasi Bimbingan Daring Skripsi Mahasiswa. Penelitian ini berhasil mengembangkan aplikasi yang menjadi solusi pelaksanaan bimbingan skripsi secara daring dikarenakan nya pandemic COVID-19. Dengan adanya pandemi Covid-19, perguruan tinggi diminta untuk menyesuaikan kegiatan perkuliahan nya termasuk bimbingan sesuai dengan protokol kesehatan. Sehingga, dikembangkan aplikasi bimbingan daring. Aplikasi dengan nama *Online Thesis Consultation Application – Tanri Abeng University (OCTA-TAU)* memiliki fitur diskusi antara dosen dengan mahasiswa, lalu mahasiswa dapat mengunggah *draft* skripsi mereka. Setelah itu, dosen dapat memperbaiki skripsi mahasiswa dengan memberikan catatan revisi [29].

Metode *Agile Scrum* pada penelitian ini dilakukan dengan 4 sprint berdurasi total 336 jam atau selama 9 minggu. *Sprint* dilakukan dengan adanya *Scrum Meeting* setiap hari, *Sprint review* dan *sprint retrospective* di akhir Sprint. Setiap *backlog* memiliki estimasi waktu yang telah ditentukan kemudian dimasukkan ke dalam *sprint backlog*. Jika *Sprint backlog* sudah selesai tetapi masih ada sisa waktu di sprint saat itu, maka sprint selanjutnya dapat dimulai dengan sprint backlog yang baru. Dengan sistem kerja seperti itu, maka proses pengembangan dapat berjalan lebih cepat. Sehingga hal ini akan dijadikan tinjauan dan diimplementasikan dalam penelitian yang dilakukan yaitu *Perancangan Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android*.

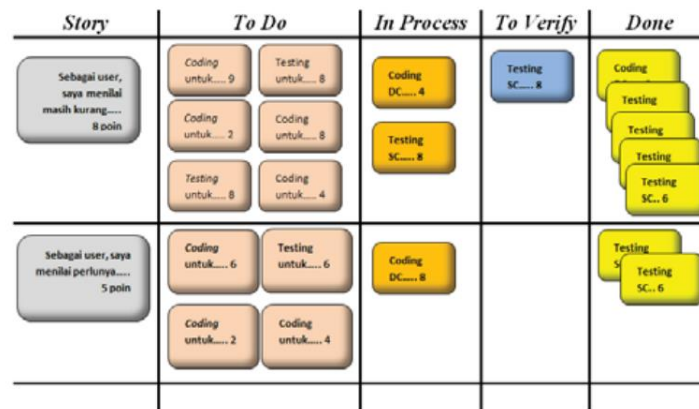
2.15.8 *Application Centric Cloud-Based Notification System Using Scrum Methodology*

Penelitian yang dilakukan oleh Iqra Urooj, Jahanzeb Jabbar, dan Naqash Azeem pada tahun 2019 membahas mengenai pengembangan aplikasi yang membantu siswa

maupun wali murid dalam melakukan pengecekan biaya akademik. Aplikasi juga melacak kegiatan non kurikuler, juga mengingatkan siswa maupun wali nya mengenai biaya dan tanggal jatuh tempo biaya akademik. Lalu, dikembangkan aplikasi berkemampuan memberikan notifikasi atau pemberitahuan. Pengembangan aplikasi menggunakan bahasa pemrograman Java. Metodologi yang dipakai adalah metodologi *Scrum* dengan beberapa tahapan yaitu membuat *product backlog*, membuat latar belakang, merancang model aplikasi, melakukan *deployment* untuk aplikasi yang telah selesai dibuat dan melakukan pengujian. Pada aplikasi notifikasi dikirimkan dengan bantuan *Firebase Cloud Messaging*. Kemudian, ada beberapa fitur lain seperti acara yang diadakan di sekolah dan penjelasan mengenai biaya akademik. Pemberian notifikasi dilakukan oleh admin melalui aplikasi berbasis web untuk siswa tertentu, kemudian notifikasi akan masuk di aplikasi dengan akun siswa tersebut [30].

2.15.9 Agile Software Methodology with Scrum for Developing Quality Assurance System

Penelitian yang dilakukan oleh Mercurius Broto Legowo, Budi Indarto, dan Deden Prayitno pada tahun 2019 membahas mengenai metodologi *Agile Scrum* dalam mengembangkan sebuah sistem *quality assurance*. Penelitian mengenai pengembangan sistem *Quality Assurance* ini dilakukan untuk mengetahui bagaimana mengelola proyek dengan kompleksitas tinggi. Dalam penelitian ini metode yang digunakan adalah penelitian tindakan dimana metode *Agile Scrum* diuji untuk mengembangkan sebuah aplikasi dengan input yang ditentukan tetapi memiliki perubahan pekerjaan yang cukup sering dilakukan. Metode *Scrum* dianggap memiliki kerangka pengerjaan proyek yang simple, dibatasi waktu, mengimplementasikan pemeriksaan dan beradaptasi untuk memperbaiki sistem kerja setiap anggota nya. Dalam penelitian ini, dilakukan tahapan *Scrum* seperti *Sprint Planning*, menentukan *Sprint Backlog*, *Daily Scrum*, *Scrum Review*, *Retrospective*, dan menampilkan backlog pada *board* [31].



Gambar 6. *Storyboard* pada penelitian [29]

Pada gambar 6 menjelaskan mengenai *storyboard* yang digunakan untuk mengelola *user story* untuk dikerjakan setiap anggota berdasarkan beberapa aspek yaitu

1. *To Do*: Berisikan semua *user story* yang belum dikerjakan, dicatat dan dikumpulkan
2. *In Process*: Berisikan semua tugas yang sedang dalam proses pengerjaan
3. *To Verify*: Berisikan semua tugas yang sudah selesai dikerjakan dan perlu di verifikasi apakah sudah sesuai dengan yang diharapkan
4. *Done*: Berisikan semua tugas dalam Sprint yang sudah selesai dan sudah terverifikasi.

Hasil penelitian menyatakan bahwa sistem *Quality Assurance* menggunakan metode *Agile Scrum* adalah yang paling tepat dalam meningkatkan efisiensi dan keefektifan pengembangan perangkat lunak yang dilakukan.

2.15.10 *Information System for Providing Food Services Based on Mobile Application Using Flutter*

Pada penelitian yang dilakukan oleh Asroni, Slamet Riyadi dan Taufiq Cahyono di tahun 2020 membahas mengenai pembuatan aplikasi *mobile* pemesanan

layanan makanan. Sebelumnya telah dibuat *website* dalam hal yang sama, yaitu pemesanan layanan makanan, tetapi dianggap kurang efisien dikarenakan harus membuka *browser* terlebih dahulu. Kemudian dikembangkan aplikasi menggunakan *framework Flutter* dengan metodologi *waterfall* sebagai solusi dari layanan pemesanan makanan melalui website tersebut. Ada beberapa fitur yang disediakan pada aplikasi seperti pengguna dapat melihat promosi makanan yang tampil di halaman utama, makanan yang paling laris, katalog, keranjang milik pengguna dan melihat informasi tentang aplikasi dan pengembangnya. Daftar makanan akan tampil pada halaman katalog, pengguna dapat melihat rincian lebih lanjut mengenai makanan yang dipilih dan memesannya. Jika pengguna melakukan pemesanan, akan ada pesan keterangan jika pesanan sedang diproses [32].

Dari penelitian ini didapatkan kesimpulan bahwa pengembangan aplikasi menggunakan *framework flutter* memiliki kelebihan jika dibandingkan dengan *framework* lain. *Flutter* menyediakan animasi yang berjalan sangat mulus dan penggunaan *widget* sebagai komponen UI yang memudahkan pengembang dalam membangun tampilan yang menarik. *Flutter* sebagai *framework multi-platform* juga memudahkan pengembang untuk mengunduh file APK untuk Android, file IPA untuk iOS dan membangun *Website* sekaligus.

2.16 State of The Art

State of the art merupakan hasil analisa yang didapatkan dari penelitian sebelumnya yang memiliki keterkaitan dengan penelitian yang akan dilakukan. Hasil analisis ini kemudian dijadikan referensi perbaikan untuk menciptakan sistem yang lebih baik dari penelitian sebelumnya. Berdasarkan penelitian [23][24], sistem peminjaman sepeda yang paling tepat untuk dikembangkan di Universitas Lampung adalah menggunakan sistem peminjaman *online* melalui aplikasi android menggunakan scan QR Code. Tetapi pada penelitian [27], sistem *code* seperti *QR Code* dan *passcode* memiliki kelemahan yaitu kerusakan data, dimana orang yang tidak

bertanggung jawab dapat mengambil dan mengubah data yang dikirim ketika sistem mengirimkan *QR Code*, *passcode* atau lokasi sesuai *GPS* ke *smartphone* pengguna. Berdasarkan penelitian [31] menyatakan bahwa *Agile Scrum* adalah metode yang paling efektif dan efisien dalam mengembangkan perangkat lunak dan lebih baik dibandingkan dengan metode tradisional seperti *SDLC Waterfall*. Hal ini dikarenakan, metode *Agile Scrum* bersifat manajerial terutama dalam prosesnya mengelola proyek dengan sumber daya manusia yang terbatas. Oleh karena itu, pada penelitian ini yang hanya dilakukan oleh satu orang *developer*, diputuskan untuk menggunakan metodologi *Agile Scrum*.

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian dilakukan pada:

1. Waktu Penelitian : Oktober 2022 sampai dengan Maret 2023
2. Tempat Penelitian : Universitas Lampung

3.2 Jadwal Penelitian

Jadwal pada penelitian ini adalah sebagai berikut:

Tabel 2. Jadwal Penelitian

No	Aktivitas	Bulan ke-					
		1	2	3	4	5	6
1	Perancangan Sistem						
2	Pengembangan Sistem (<i>Sprint 1 – Sprint 4</i>)						
3	<i>Testing</i>						
4	Analisis						
5	Pelaporan						

3.3 Alat Dalam Penelitian

3.3.1 Alat Penelitian

Alat-alat yang digunakan dalam penelitian ini adalah sebagai berikut:

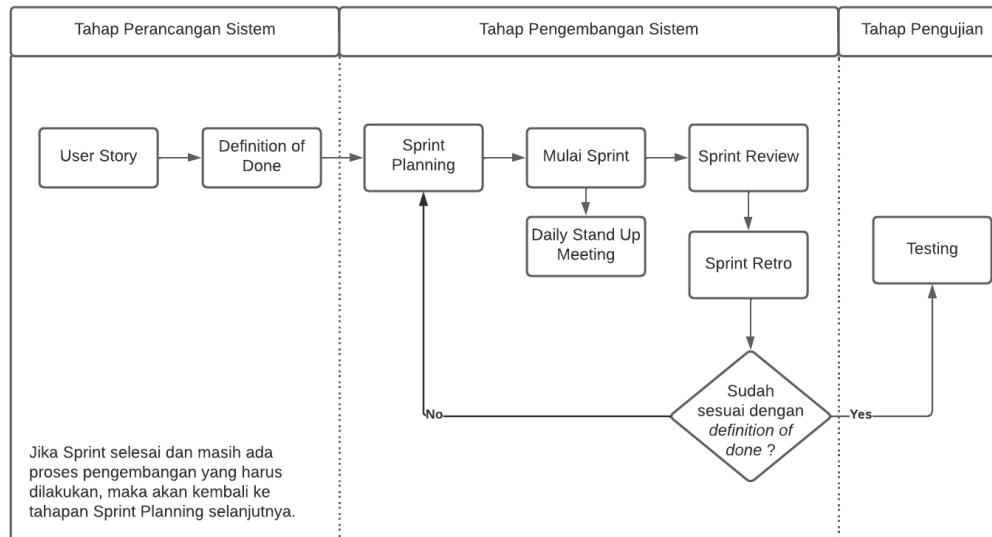
Tabel 3. Alat yang digunakan dalam penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	Laptop	Intel Core i5, RAM 8GB, dengan sistem operasi Windows 10.	Perangkat keras yang digunakan sebagai compiler dalam pemrograman android.
2	<i>Android Device</i>	Android 11.0	Perangkat keras yang digunakan untuk menginstall aplikasi yang sedang dikembangkan
3	<i>Android Emulator</i>	Android 11.0	Perangkat lunak yang digunakan untuk menjalankan aplikasi yang sedang dikembangkan.
4	<i>Visual Studio Code</i>	Versi 1.67.1	Perangkat Lunak yang digunakan untuk mengembangkan aplikasi <i>smartphone</i> berbasis Android
5	Flutter	Versi 3.3.0-0.1.pre	<i>Framework</i> pemrograman yang digunakan untuk memprogram aplikasi multiplatform
5	Figma	<i>Online Figma</i>	Perangkat lunak untuk merancang antarmuka aplikasi.
6	Trello	<i>Online Trello</i>	<i>Tools</i> untuk menjalankan <i>Scrum</i>

No	Nama Alat	Spesifikasi	Deskripsi
7	Firebase	Firebase Firestore dan Firebase <i>Storage</i>	Database tempat penyimpanan data yang akan digunakan

3.4 Tahapan Penelitian

Pada penelitian ini digunakan model pengembangan perangkat lunak *Agile Scrum*. Adapun tahapan pengembangan menggunakan metode *Agile Scrum* adalah seperti gambar di bawah ini:



Gambar 7. Tahapan pada metode *Agile Scrum*

Gambar 7 menjelaskan tahap penelitian yang terbagi menjadi 3 tahapan, yaitu: tahap perancangan sistem, tahap pengembangan sistem, dan tahap pengujian. Pada perancangan sistem dilakukan *requirement gathering* yaitu pengumpulan informasi dan kebutuhan sebelum proses pengembangan dimulai. *Requirement* yang telah didapat akan dibuat *User Story*. *User Story* inilah yang akan menjadi awal dari pengembangan aplikasi. Pada tahap pengembangan, dijalankan rangkaian kegiatan *Scrum*, seperti *Sprint Planning*, *Daily Scrum*, *Sprint Review* dan *Sprint Retrospective*.

Sprint planning diadakan setiap awal dimulai *Sprint* dengan melakukan perincian *User Story* beserta *Sprint Backlog* yang akan dikerjakan. Selama *sprint* berjalan, *Daily Scrum* dilaksanakan untuk menyampaikan progress yang telah dikerjakan. Jika satu *Sprint* selesai, maka akan diadakan *Sprint Review* dan *Sprint Retrospective*. Pada kegiatan ini ditinjau *progress Sprint* berupa *User Story* atau *backlog* yang telah berhasil diselesaikan. Sementara *Sprint Retro* sendiri menjadi wadah bagi pengembang untuk merefleksikan progress yang dikerjakan. Apabila semua *user story* telah selesai dikerjakan dan mencapai *definition of done*, maka akan dilakukan pengujian atau *testing*. Dalam penelitian ini total *sprint* yang akan dilakukan adalah 4 *Sprint*, dengan durasi satu *sprint* nya yaitu 2 minggu.

3.4.1 Tahap Perancangan Sistem

Pada perancangan sistem dilakukan tahap awal *requirements gathering* yang dibutuhkan untuk penentuan *user story* dan *backlog* dalam pengembangan sistem.

3.4.1.1 Literature Review

Pada tahap *literature review* dilakukan analisis dari beberapa penelitian mengenai sistem peminjaman sepeda di berbagai kampus. Tahap ini bertujuan untuk menjadikan peminjaman sepeda di kampus lain sebagai acuan untuk membangun sistem paling efektif yang dapat diterapkan di Universitas Lampung.

3.4.1.2 Pemodelan Sistem

Pada tahap ini dilakukan perincian hasil analisa dari *literature review* yang sudah dilakukan. Pemodelan sistem peminjaman sepeda dilakukan sebagai pondasi

dari pengembangan yang akan dilakukan. Sehingga, ada beberapa hal yang perlu ditentukan yaitu sebagai berikut:

a. Target Pengguna

Pada tahap ini dilakukan pendefinisian mengenai pengguna yang akan menggunakan aplikasi peminjaman sepeda. Penentuan target pengguna akan dilihat dari latar belakang pengembangan aplikasi. Aplikasi ini akan diimplementasikan di Universitas Lampung, sehingga terdapat 3 kategori pengguna yaitu Mahasiswa, Dosen, dan Staff.

b. Kebutuhan Fungsional Sistem

Kebutuhan Fungsional menjelaskan mengenai fungsi-fungsi ataupun fitur yang harus dipenuhi oleh sistem dan harus berjalan sesuai dengan tujuan aplikasi. Kebutuhan fungsional harus mencakup beberapa fitur seperti otentikasi pengguna, peminjaman sepeda yang terorganisir, memudahkan pengguna dalam mengetahui status peminjaman nya, serta pengembalian sepeda.

c. Kebutuhan Non-Fungsional Sistem

Kebutuhan Non-Fungsional Sistem merupakan kebutuhan tambahan untuk mendukung fungsi-fungsi utama dari sistem yang ada. Penentuan kebutuhan non-fungsional mengacu pada kebutuhan fungsional yang sudah ditetapkan. Dalam hal ini, terpenuhinya aspek *availability* aplikasi yang dapat berjalan pada *smartphone* dan aspek *security* dengan adanya otentikasi.

d. Batasan Perancangan

Batasan perancangan didefinisikan berdasarkan kebutuhan yang sudah ditentukan sebelumnya. Batasan perancangan pada penelitian ini meliputi batas pengembangan aplikasi, bahasa serta *tools* yang digunakan, dan syarat penggunaan sistem.

e. Arsitektur Sistem Peminjaman Sepeda

Arsitektur sistem menggambarkan model sistem peminjaman mulai dari *shelter*, *database*, hingga peminjaman menggunakan aplikasi *mobile*. Arsitektur sistem ini perlu digambarkan untuk mengetahui alur sistem yang akan dikembangkan dan hubungan antar objek nya.

f. Use case

Use case sistem peminjaman sepeda dibuat berdasarkan kebutuhan fungsional dan non-fungsional yang telah ditentukan sebelumnya. *Use case* yang dibuat merupakan rincian kegiatan yang dapat dilakukan oleh pengguna.

g. Activity Diagram

Activity diagram adalah sebuah gambar alir dari suatu aktivitas pada sistem. *Use case* yang sudah ditentukan kemudian dibuat proses aktivitas nya secara keseluruhan. Kemudian proses aktivitasnya digambarkan ke dalam *activity diagram*, sehingga mudah dimengerti.

3.4.1.3 Penentuan *User Story*

Tahap Penentuan *User Story* dilakukan supaya mendapatkan *Sprint Backlog* yang akan menjadi acuan *task* untuk dikerjakan di setiap *Sprint* nya. *User Story* dapat ditentukan berdasarkan kebutuhan fungsional dan non-fungsional aplikasi. Format *User Story* yang digunakan dalam penelitian ini adalah sebagai berikut:

Sebagai pengguna, saya ingin [kegiatan], agar saya [hasil yang diinginkan]

Gambar 8. Format *User Story*

Gambar 8 menampilkan format *user story* pada penelitian ini dimana kegiatan merupakan suatu aktivitas yang dilakukan oleh pengguna untuk mencapai hasil yang diinginkan. Pada sistem, semua pengguna dapat melakukan aktivitas yang sama, sehingga tidak ada perbedaan kegiatan untuk setiap pengguna.

3.4.1.4 Penentuan *Backlog*

User Story yang telah didefinisikan akan dirinci kembali ke dalam sebuah *backlog* atau *task* yang lebih kecil. *Backlog* yang sudah didapat akan dimasukkan ke dalam *board* pada *tools* Trello sesuai dengan status pengembangan nya.

3.4.1.5 Penentuan *Definition of Done*

Penentuan *Definition of Done* dilakukan untuk menentukan definisi atau kondisi dimana sebuah *backlog* dapat dikatakan selesai. Hal ini sebagai standar pengerjaan *backlog* dan transparansi antara pengembang dengan *product owner*.

3.4.2 Tahap Pengembangan Sistem

Tahap pengembangan sistem adalah tahap dimana sistem yang sudah dirancang direalisasikan dengan kode. Sehingga, menghasilkan sebuah aplikasi yang dapat digunakan oleh pengguna. Sesuai dengan metode yang digunakan dalam penelitian ini yaitu *Agile Scrum*, maka ada beberapa tahapan kecil yang akan dilakukan diantaranya adalah sebagai berikut:

3.4.2.1 *Sprint Planning*

Proses *Sprint Planning* dilakukan untuk merencanakan setiap pekerjaan atau *Sprint Backlog* yang akan dikerjakan selama masa *Sprint*. *Sprint Planning* diadakan setiap ingin memulai *sprint*. Pada kegiatan ini dibahas apa saja yang harus diselesaikan pada *Sprint* dan bagaimana *backlog* tersebut akan diselesaikan sehingga mencapai *definition of done*.

3.4.2.2 *Sprint*

Pada penelitian ini, total *Sprint* yang akan dilakukan adalah 4 *Sprint* dengan durasi satu *Sprint* selama 2 minggu. Setiap *backlog* yang akan dikerjakan akan masuk ke *board* Trello. Pada Trello, *backlog* akan disesuaikan dengan status pengerjaan nya yaitu *in progress*, *ready for test* dan *done*. Pada saat *Sprint* akan dilakukan 2 proses yaitu sebagai berikut:

a. *Daily Scrum*

Daily Scrum dilakukan setiap harinya dengan durasi waktu maksimal 15 menit.

Kegiatan ini berguna untuk menginspeksi pekerjaan yang telah dilakukan selama satu hari lalu atau semenjak *Daily Scrum* sebelumnya. Selain itu, pengembang juga menyampaikan prakiraan atau rencana pengerjaan *backlog* pada hari itu. Dengan adanya *Daily Scrum*, didapatkan catatan mengenai perkembangan pekerjaan.

b. Development

Tahap *development* merupakan tahap pengembangan sistem atau pengerjaan *backlog* dalam *Sprint* guna mencapai hasil yang diharapkan. Pada tahap ini akan dilakukan penulisan kode sesuai dengan rancangan sistem dan diharapkan mencapai *Definition of Done*.

3.4.2.3 Sprint Review

Tahap *Sprint Review* diselenggarakan guna menginspeksi *backlog*. Pada *Sprint Review* akan didemonstrasikan hasil kerja atau *backlog* yang sudah dikerjakan,. Kemudian didiskusikan masalah dan kesulitan yang dihadapi selama *sprint*. Setelahnya, didiskusikan *backlog* yang sudah direvisi, yang merupakan hasil dari *backlog* sebelumnya dan juga *backlog* yang akan diimplementasikan di *Sprint* berikutnya.

3.4.2.4 Sprint Retrospective

Tahap *Sprint Retrospective* diselenggarakan untuk mengidentifikasi jalan nya *Sprint*. Kegiatan ini membahas apa saja yang berjalan dengan baik dan membuat perencanaan untuk meningkatkan kinerja.

3.4.3 Tahap *Testing*

Tahap *testing* dilakukan menggunakan metode *Black Box Testing* untuk mengetahui apakah sistem berjalan sesuai yang diharapkan. Jika aplikasi belum mencapai yang diharapkan, maka pengembang dapat memperbaiki aplikasi tersebut. Pada penelitian ini akan dilakukan pengujian pada aplikasi yang sudah diunduh dan diuji secara langsung. Pengujian didasarkan oleh *Use Case* yang telah dibuat dan mengacu pada kebutuhan fungsional serta non fungsional. Hasil akhir dari tahap *testing* adalah aplikasi yang sistem nya sudah berjalan sesuai harapan dan siap digunakan.

3.4.4 Tahap Analisis

Tahap analisis dilakukan untuk mengetahui hasil dari pengujian pengembangan Aplikasi Peminjaman Sepeda Universitas Lampung yang didapat dari tahap *testing* sebelumnya. Tahap analisis akan dilaksanakan jika seluruh data pengujian sudah terkumpul.

3.4.5 Tahap Pelaporan

Tahap pelaporan menjadi tahap akhir dari penelitian ini, yaitu pelaporan hasil dan temuan dari penelitian Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis *Android* Menggunakan Metode *Agile Scrum*. Semua data yang telah diperoleh dan telah dianalisis akan dilakukan pengambilan kesimpulan dan saran dan digunakan sebagai skripsi pada Universitas Lampung.

IV. HASIL DAN PEMBAHASAN

4.1 Hasil

Skripsi ini menghasilkan sebuah aplikasi peminjaman sepeda berbasis android, dimana pengguna dapat meminjam dan mengembalikan sepeda secara *online*.

4.1.1 Perancangan Sistem

Pada tahap perancangan sistem, didiskusikan dengan pihak Pusat Unggulan Institusi Perguruan Tinggi *Green Technology* Universitas Lampung dan dilakukan *literature review* pada beberapa penelitian terkait peminjaman sepeda di kampus. Terdapat beberapa poin yang didapatkan dari *literature review* yaitu peminjaman sepeda di IPB, UGM dan UI dibebaskan jam peminjaman nya. Sehingga, tidak ada batas waktu peminjaman dalam sehari dalam artian sepeda menjadi tanggung jawab pengguna sampai dikembalikan kapanpun itu. Peminjaman sepeda tidak memungut biaya dan masih menggunakan kartu tanda pengenalan sebagai jaminan. Tetapi, di UI sendiri sudah diberlakukan peminjaman menggunakan aplikasi dan diperlukan login menggunakan SSO kampus. Hal ini menjadi poin penunjang untuk membangun sistem peminjaman sepeda di Universitas Lampung.

PUI-PT *Green Technology* Universitas Lampung mengutarakan bahwa

target pengguna dari aplikasi peminjaman sepeda ini adalah *civitas* akademika Universitas Lampung. Oleh karena itu, didapatkan beberapa poin sebagai dasar dalam pembuatan sistem yang efektif, yaitu aktivitas di Universitas Lampung dalam sehari berjalan selama 8 jam dan sebagian besar *civitas* akademika memiliki perangkat ponsel. Sehingga, dalam mewujudkan transportasi hijau, dikembangkanlah aplikasi *mobile* untuk peminjaman sepeda. Aplikasi peminjaman sepeda ini memerlukan verifikasi pengguna menggunakan sistem otentikasi *Single Sign-On* Unila. Pengguna diberi waktu 4 jam selama satu hari yang merupakan setengah dari total jam aktivitas di kampus. Lalu, pengguna diharuskan mengembalikan sepeda dalam waktu peminjaman 4 jam tersebut. Pengguna diperbolehkan untuk meminjam sepeda sebanyak yang dimau jika masih memiliki sisa waktu peminjaman dalam hari itu. Jika terlambat mengembalikan sepeda, maka pengguna memiliki denda waktu peminjaman. Konsekuensinya adalah pengguna tidak diperbolehkan meminjam sepeda selama satu hari setelahnya.

4.1.1.1 Pemodelan Sistem

a. Target Pengguna

Pada tahap ini dilakukan pendefinisian mengenai pengguna yang akan menggunakan aplikasi. Studi kasus pada penelitian ini dilakukan di Universitas Lampung dengan melihat 3 *user* utama sebagai target pengguna aplikasi yaitu mahasiswa Universitas Lampung, dosen Universitas Lampung dan pegawai Universitas Lampung.

a. Mahasiswa Universitas Lampung

Mahasiswa merupakan orang-orang yang akan menjadi calon sarjana dan

memiliki suatu ikatan atau keterkaitan dengan suatu perguruan tinggi, yang dididik dan juga diharapkan akan menjadi calon-calon intelektual (Knopfemacher, Suwono, 1978). Mahasiswa Universitas Lampung adalah seseorang yang sedang dalam proses menuntut ilmu di Universitas Lampung.

b. Dosen Universitas Lampung

Dosen merupakan seseorang yang mendidik mahasiswa secara profesional, melakukan penelitian dan pengabdian kepada masyarakat di berbagai program studi Universitas Lampung.

c. Pegawai Universitas Lampung

Pegawai adalah seseorang yang memiliki pekerjaan secara tetap maupun kontrak (dengan jangka waktu pekerjaan yang telah ditentukan atau disepakati dalam kontrak kerja) di Universitas Lampung. Pegawai Universitas Lampung dapat terdiri dari pegawai administrasi, pengelola akademik dan lainnya.

b. Kebutuhan Fungsional Sistem

Kebutuhan fungsional dari sistem adalah sebagai berikut:

Tabel 4. Kebutuhan fungsional aplikasi

No	ID	Fitur	Deskripsi
1.	UB-01	Daftar <i>shelter</i> peminjaman	Aplikasi menyediakan informasi daftar shelter peminjaman di Universitas Lampung
2.	UB-02	Daftar sepeda yang tersedia	Aplikasi menyediakan informasi daftar sepeda yang tersedia di <i>shelter</i> yang dipilih pengguna
3.	UB-03	Peminjaman sepeda	Transaksi peminjaman sepeda berdasarkan sepeda dan <i>shelter</i> yang dipilih pengguna.

No	ID	Fitur	Deskripsi
4.	UB-04	Pengembalian sepeda	Transaksi pengembalian sepeda dengan memilih <i>shelter</i> pengembalian.
5.	UB-05	Status peminjaman sepeda	Menyediakan informasi status peminjaman milik pengguna.

c. Kebutuhan Non Fungsional Sistem

Adapun kebutuhan non fungsional dari sistem adalah sebagai berikut:

Tabel 5. Kebutuhan non-fungsional aplikasi

No	ID	Fitur	Deskripsi
1.	UB-06	Availability	Sistem berjalan pada <i>smartphone</i> dengan sistem operasi Android, dapat berjalan jika terhubung dengan koneksi <i>internet</i> , dan berjalan sesuai dengan waktu aktivitas di Universitas Lampung.
2.	UB-08	Security	Diperlukan otentikasi menggunakan <i>Single Sign-On</i> Universitas Lampung pengguna sebelum masuk ke dalam aplikasi dan meminjam sepeda.

d. Batasan Perancangan

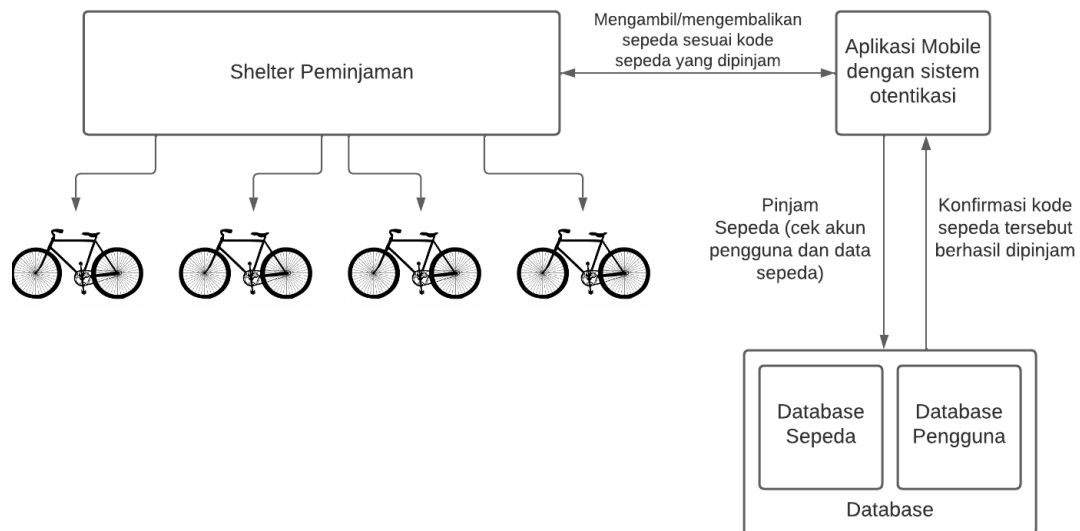
Pada penelitian ini terdapat batasan perancangan. Adapun Batasan perancangan sistem yang dikembangkan adalah sebagai berikut:

1. Aplikasi yang dikembangkan hanya dari sisi pengguna sebagai peminjam sepeda, tidak meliputi sisi pengelola sepeda.

2. Aplikasi masih bersifat lokal *server* dan belum diimplementasikan untuk digunakan oleh umum.
3. Pengembangan aplikasi menggunakan *framework Flutter*, *IDE Visual Studio Code* dan *Firebase* sebagai *database*.

e. Arsitektur Sistem Peminjaman Sepeda

Berikut ini adalah arsitektur yang dibuat untuk mengetahui hubungan setiap objek dan alur kerja dari sistem peminjaman sepeda secara garis besar nya.

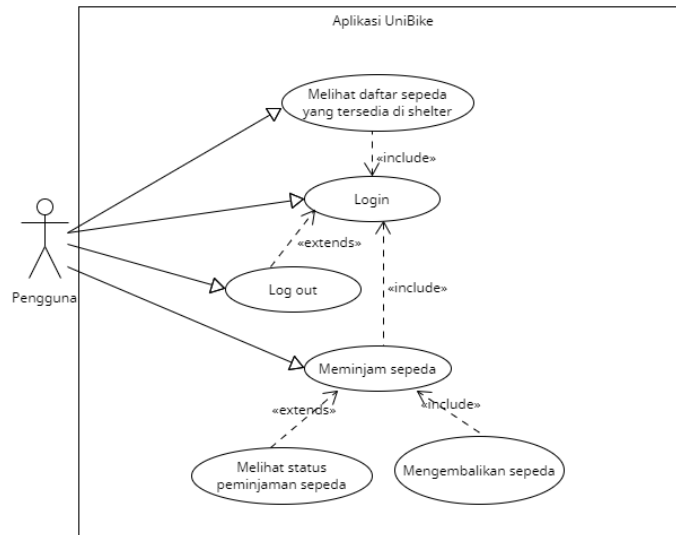


Gambar 9. Arsitektur Sistem Peminjaman Sepeda

Gambar 9 menjelaskan bahwa mengambil sepeda pada *shelter* dilakukan setelah meminjam sepeda melalui aplikasi. Dijalankan juga pengecekan akun apakah ada aspek lain seperti denda peminjaman atau sepeda telah dipinjam oleh pengguna lain,

f. Use Case

Berdasarkan *use case business* pada gambar 7, didapatkan *use case* sistem yang kemudian digambarkan ke dalam *use case diagram* seperti pada gambar berikut:



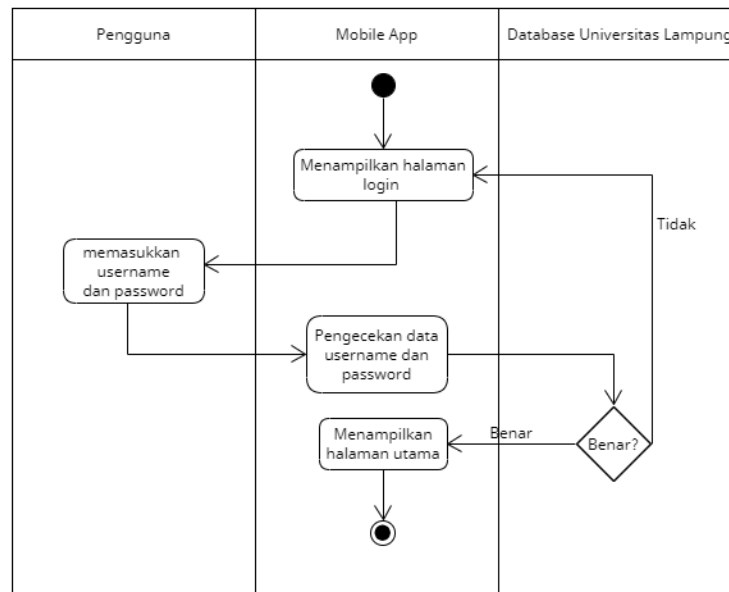
Gambar 10. *Use Case Diagram* Aplikasi Peminjaman Sepeda di Universitas Lampung

Gambar 10 menjelaskan *use case diagram* dari sistem yang dikembangkan dengan satu aktor yaitu pengguna aplikasi. Aktor pengguna pada sistem ini dapat melakukan login, melihat daftar sepeda yang tersedia di setiap *shelter* peminjaman, meminjam sepeda, melihat status peminjaman nya, dan mengembalikan sepeda.

g. Activity Diagram

Aplikasi Peminjaman Sepeda di Universitas Lampung Berbasis Android memiliki beberapa fungsi yang dapat dijelaskan melalui activity diagram.

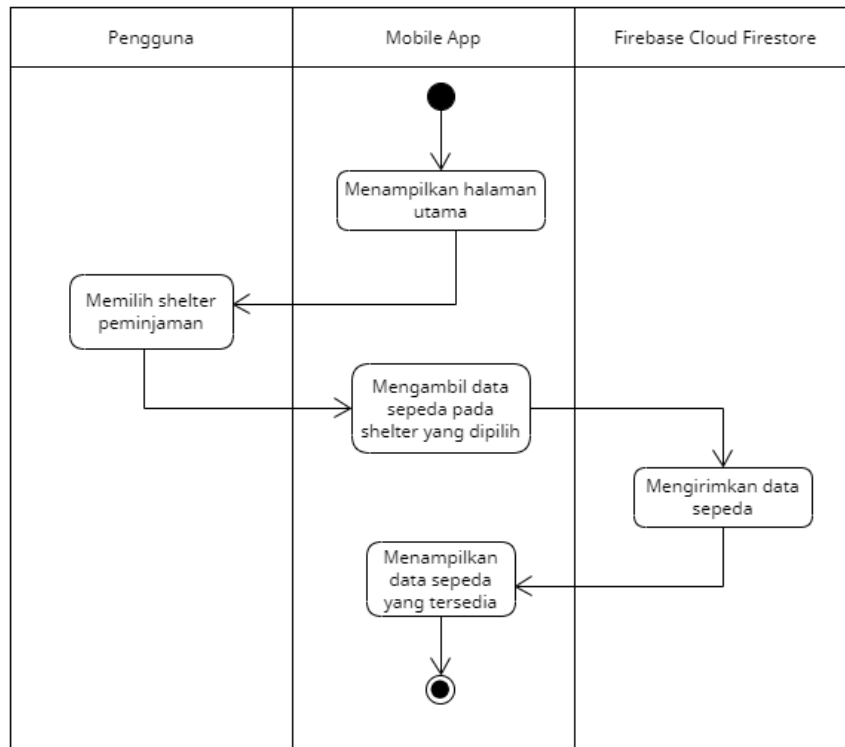
a. *Activity Diagram Login User*



Gambar 11. *Activity Diagram ketika user login*

Gambar 11 menampilkan diagram aktivitas saat proses otentikasi yang dilakukan oleh pengguna. Proses dimulai ketika aplikasi yang terunduh pada *smartphone* menampilkan halaman login. Pengguna kemudian dapat menggunakan kredensial *single sign-on* dari Universitas Lampung untuk masuk. Kemudian, sistem akan melakukan pengecekan apakah *username* dan *password* benar pada *database* Unila. Jika salah akan kembali ke halaman login, jika benar akan masuk ke halaman utama.

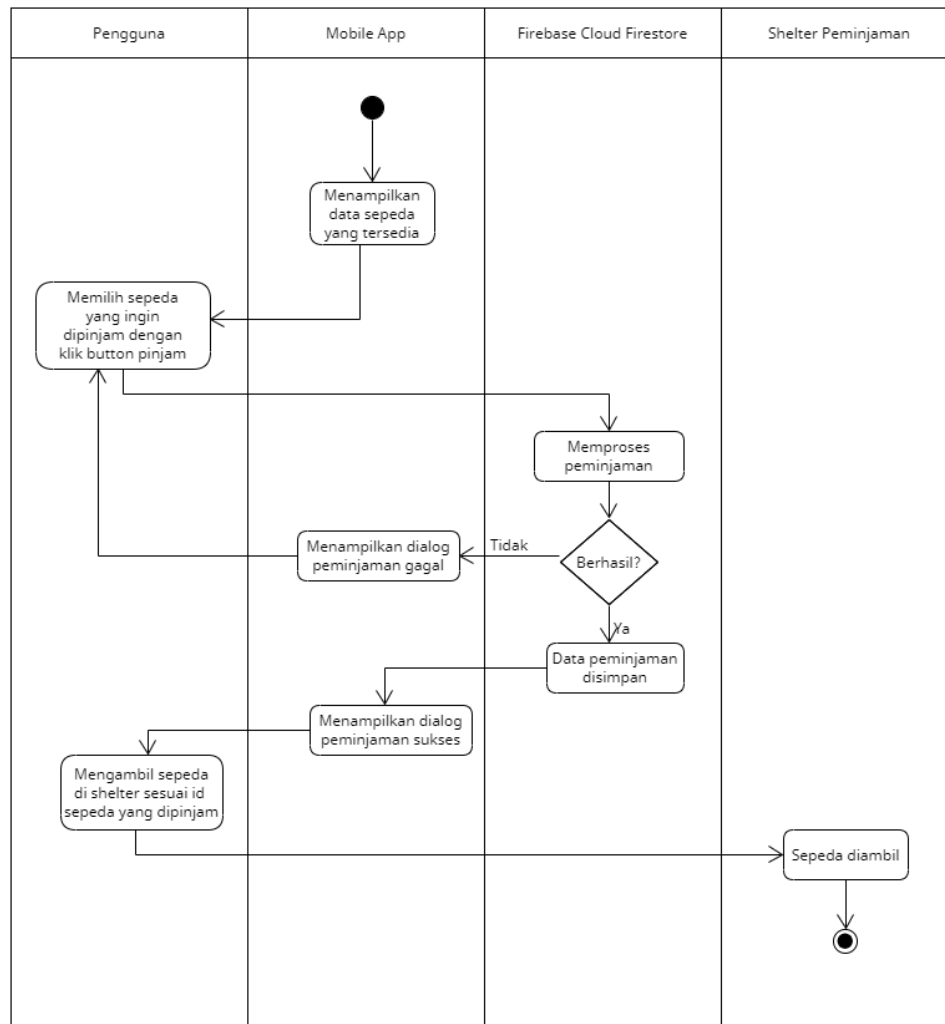
b. *Activity Diagram Melihat Daftar Sepeda*



Gambar 12. *Activity diagram* melihat daftar sepeda pada shelter

Gambar 12 menjelaskan diagram aktivitas saat pengguna ingin melihat daftar sepeda yang tersedia pada *shelter* peminjaman. Aktivitas dimulai ketika aplikasi menampilkan halaman utama ketika dibuka. Kemudian, pengguna memilih *shelter* peminjaman yang tersedia dan aplikasi akan mengambil data sepeda dari *Firebase Cloud Firestore*. *Firebase* akan mengirimkan data sepeda dalam bentuk JSON yang akan diproses pada aplikasi menjadi daftar sepeda.

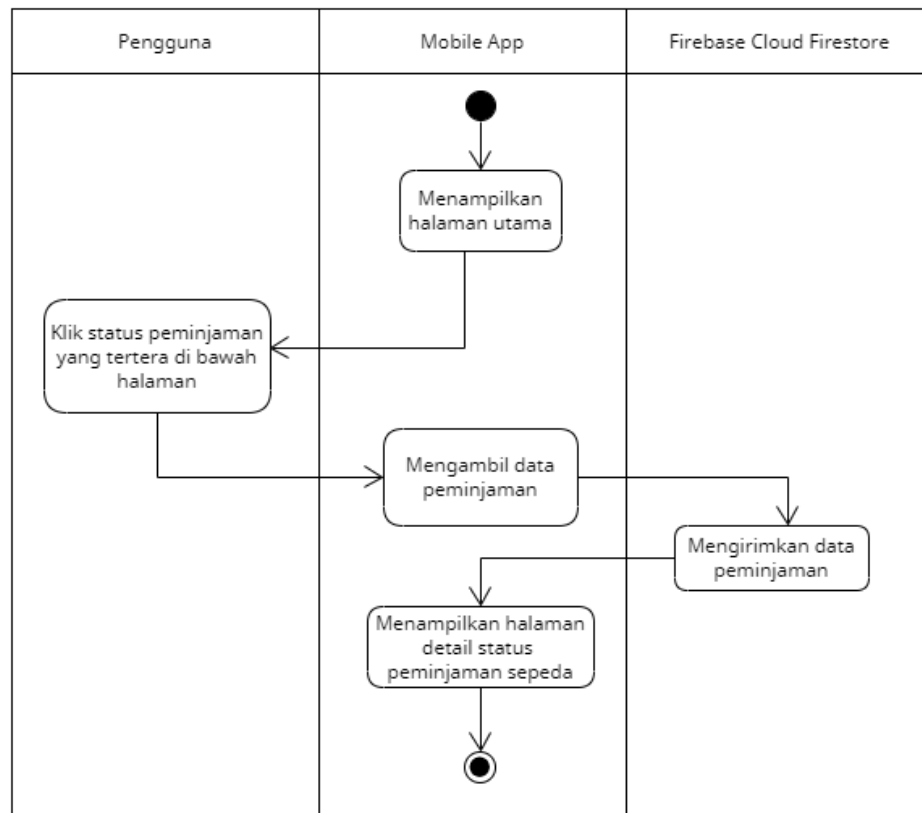
c. *Activity Diagram Meminjam Sepeda*



Gambar 13. *Activity Diagram Peminjaman Sepeda*

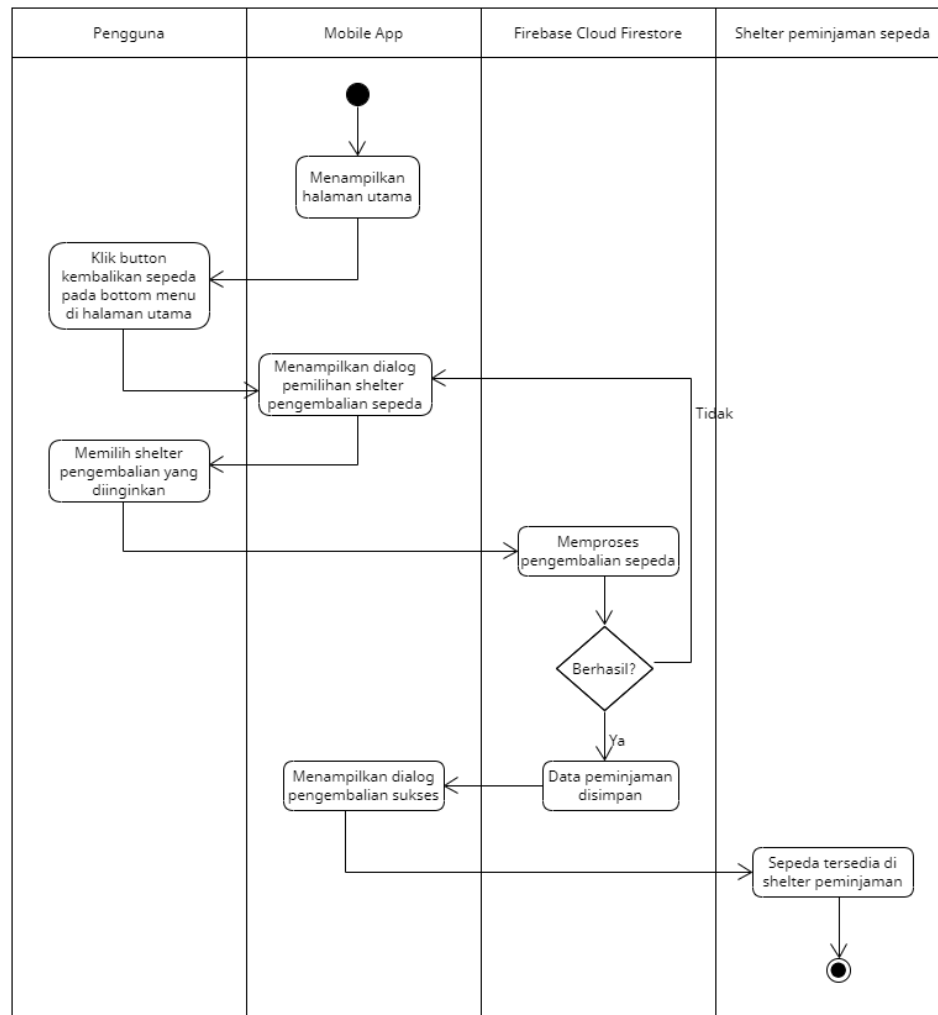
Gambar 13 merupakan diagram aktivitas meminjam sepeda melalui aplikasi UniBike dengan asumsi pengguna telah *login* dan memilih *shelter* peminjaman. Ketika aplikasi telah menampilkan daftar sepeda pada shelter yang dipilih, pengguna dapat memilih sepeda mana yang ingin dipinjam. Jika peminjaman gagal, maka akan menampilkan dialog gagal dan kembali ke pemilihan sepeda. Jika peminjaman berhasil, maka akan menampilkan dialog berhasil. Kemudian pengguna dapat mendatangi *shelter* dan meminjam sepeda sesuai dengan kode sepeda yang dipinjam pada aplikasi.

d. *Activity* Melihat status peminjaman

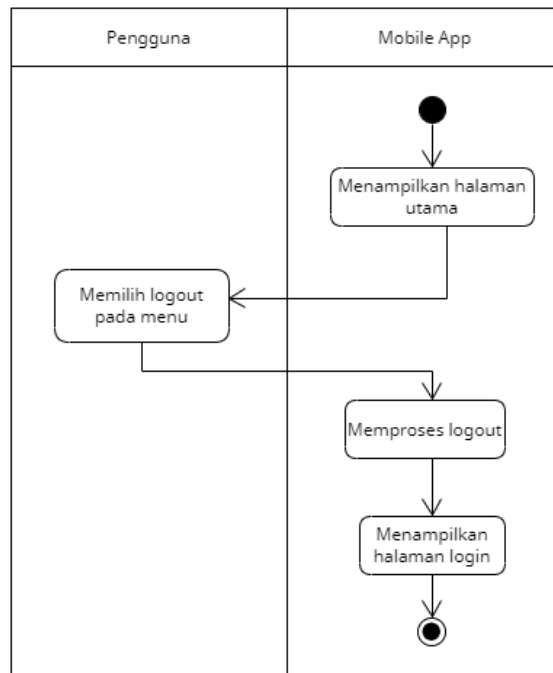


Gambar 14. *Activity Diagram* saat melihat status peminjaman

Gambar 14 memperlihatkan diagram aktivitas ketika pengguna ingin melihat status peminjaman. Jika sedang meminjam sepeda, pengguna dapat melihat status peminjaman pada *footer* di aplikasi yang jika ditekan akan berpindah ke halaman detail status peminjaman. Aplikasi akan mengambil data peminjaman dari *Firebase Cloud Firestore*. Lalu data peminjaman akan tampil pada halaman detail status peminjaman.

e. *Activity Mengembalikan Sepeda*Gambar 15. *Activity Diagram* ketika pengembalian sepeda

Gambar 15 menjelaskan diagram aktivitas ketika pengguna ingin mengembalikan sepeda. Pengguna dapat mengkonfirmasi bahwa pengguna sudah mengembalikan sepedanya melalui aplikasi. Dengan cara klik *button* kembalikan sepeda yang tampil pada halaman utama di bagian bawah. Lalu dialog pemilihan daftar *shelter* pengembalian akan muncul. Jika berhasil memilih *shelter* pengembalian, maka akan tampil dialog peminjaman selesai dan pengembalian sukses. Jika gagal, maka akan kembali ke halaman utama.

f. *Activity logout*Gambar 16. *Activity Diagram logout*

Pada gambar 16 menampilkan diagram aktivitas saat pengguna ingin *logout* dari aplikasi. Aktivitas dimulai ketika aplikasi menampilkan halaman utama. Lalu, pengguna dapat memilih menu *logout* pada hamburger menu di *appbar* aplikasi. Jika *logout* selesai diproses, aplikasi akan berganti halaman ke halaman *login*.

4.1.1.2 Penentuan *User Story*

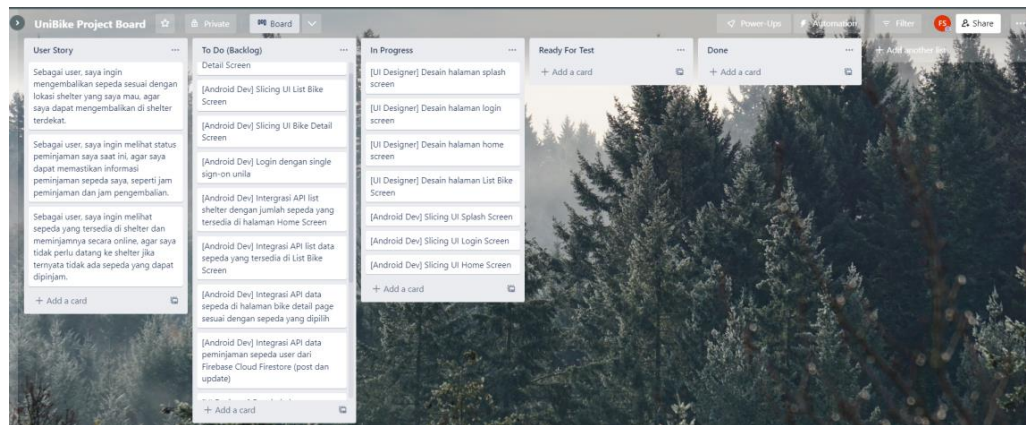
Berdasarkan tahap pemodelan sistem, didapatkan *Sprint Backlog* sebagai berikut:

Tabel 6. *List User Story dan Backlog*

<i>User Story</i>	Prioritas	Backlog	Estimasi Waktu (hari)
Sebagai <i>user</i> , saya ingin melihat sepeda yang tersedia di <i>shelter</i> dan meminjamnya secara <i>online</i> , agar saya tidak perlu datang ke <i>shelter</i> jika ternyata tidak ada sepeda yang dapat dipinjam.	Penting	1. [UI Designer] Desain halaman <i>splash screen</i>	1
		2. [UI Designer] Desain halaman <i>login</i>	1
		3. [UI Designer] Desain halaman utama	1
		4. [UI Designer] Desain halaman daftar sepeda	1
		5. [UI Designer] Desain halaman detail sepeda	1
		6. [Android Dev] <i>Slicing UI Splash Screen</i>	1
		7. [Android Dev] <i>Slicing UI Login Screen</i>	2
		8. [Android Dev] <i>Slicing UI halaman utama</i>	2
		9. [Android Dev] <i>Slicing UI halaman daftar sepeda</i>	2
		10. [Android Dev] <i>Slicing UI halaman detail sepeda</i>	1
		11. [Android Dev] <i>Login dengan single sign-on unila</i>	1
		12. [Android Dev] Integrasi API daftar <i>shelter</i> dengan jumlah sepeda yang tersedia di halaman utama	1
		13. [Android Dev] Integrasi API data sepeda yang tersedia di halaman daftar sepeda	3
		14. [Android Dev] Integrasi API data sepeda di halaman detail sepeda sesuai dengan sepeda yang dipilih	3

<i>User Story</i>	Prioritas	<i>Backlog</i>	Estimasi Waktu (hari)
		15. [<i>Android Dev</i>] Integrasi API data peminjaman sepeda <i>user</i> dari <i>Firebase Cloud Firestore</i> (<i>post</i> dan <i>update</i>)	3
Sebagai <i>user</i> , saya ingin melihat status peminjaman saya saat ini, agar saya dapat memastikan informasi peminjaman sepeda saya, seperti jam peminjaman dan jam pengembalian.	Penting	1. [<i>UI Designer</i>] Desain halaman status peminjaman 2. [<i>Android Dev</i>] <i>Slicing UI</i> halaman status peminjaman 3. [<i>Android Dev</i>] Integrasi API data peminjaman pada halaman Status Peminjaman	1 1 3
Sebagai <i>user</i> , saya ingin mengembalikan sepeda sesuai dengan lokasi <i>shelter</i> yang saya mau, agar saya dapat mengembalikan di <i>shelter</i> terdekat.	Penting	1. [<i>Android Dev</i>] <i>Delete</i> dan <i>post</i> data peminjaman di <i>Firebase Cloud Storage</i> setelah sepeda dikembalikan 2. [<i>Android Dev</i>] <i>Logout</i> dari aplikasi	3 3
Total estimasi waktu:			40

Tabel 6 menampilkan 3 *user story* yang kemudian terbagi menjadi 20 *backlog* dan total estimasi waktu yaitu 40 hari. *Backlog* tersebut kemudian dimasukkan ke dalam *board* di *Trello* selama masa sprint.



Gambar 17. *User story* dan *backlog* pada *board* Trello

Gambar 17 merupakan implementasi trello sebagai *platform* untuk menjalani *Agile Scrum* selama penelitian. *Backlog* akan dipindahkan sesuai dengan status pengerjaan nya pada masa sprint yaitu *in progress*, *ready for test* atau *done*.

4.1.1.3 Penentuan Definition of Done

Definition of Done pada pengembangan aplikasi peminjaman sepeda adalah aplikasi memenuhi kebutuhan fungsional dan non-fungsional nya. Sehingga, dapat berjalan sesuai dengan *use case* yang telah dibuat. Pada aplikasi UniBike, *definition of done* adalah aplikasi menyajikan daftar *shelter* dan daftar sepeda sehingga pengguna dapat melakukan peminjaman sepeda yang terorganisir dan mengembalikan sepeda secara online.

4.1.2 Pengembangan Sistem

4.1.2.1 Sprint

Pengembangan sistem dilaksanakan selama 8 minggu mulai dari tanggal 31 Oktober 2022 sampai dengan 23 Desember 2022. *Sprint* dilaksanakan selama 8 jam dalam satu hari yaitu pukul 08.00 sampai 17.00 (dengan jam istirahat mulai dari jam 12.00 sampai dengan jam 13.00). *Sprint* berjalan selama 5 hari kerja dalam satu minggu yaitu Senin sampai dengan Jumat. Seperti yang sudah ditentukan sebelumnya, pengembangan aplikasi akan dilaksanakan selama 4 *sprint* dengan satu *sprint* nya berdurasi dua minggu. Dengan *scrum* team yang terdiri dari, *Product Owner* yaitu bapak Ir. Meizano Ardhi Muhammad, S.T., M.T. yang mewakili *stakeholder* yaitu PUI-PT. Lalu *Scrum Master* yaitu bapak Rio Ariestia Pradipta, S.Kom., M.T.I. Adapun pembagian *backlog* yang dikerjakan selama 4 *sprint* dapat dilihat sebagai berikut:

a. *Sprint 1*

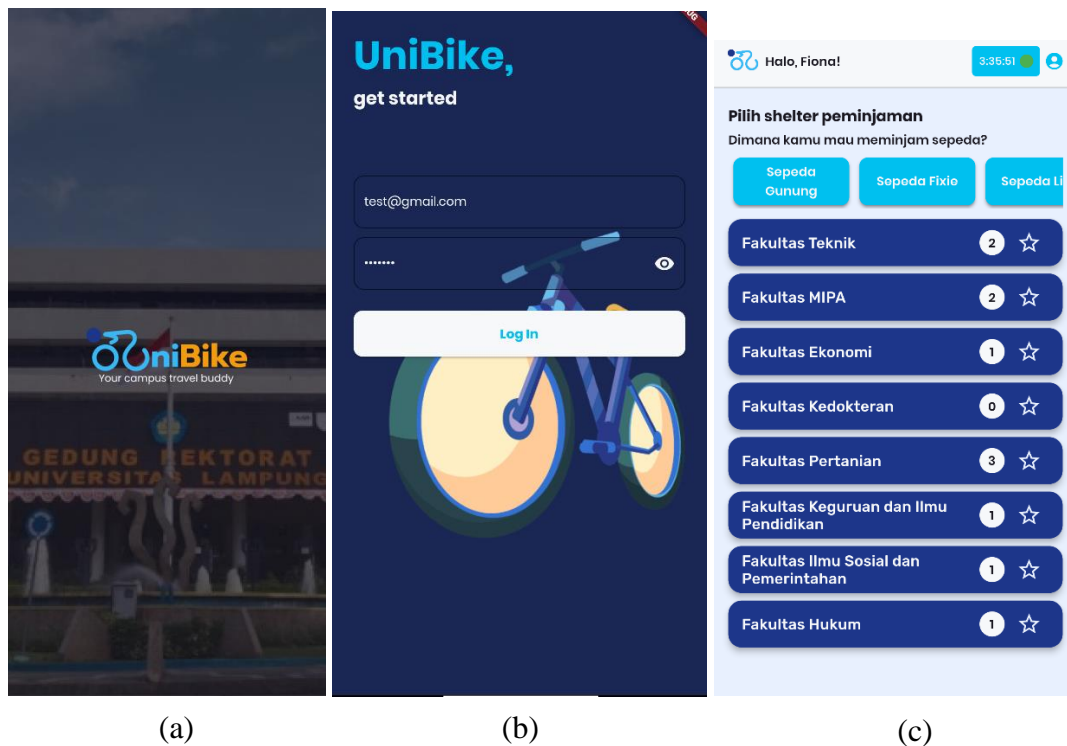
Pembagian *backlog* di setiap *sprint* diurutkan pada *sprint* pertama. *Sprint 1* terdapat 7 *backlog* yang ditentukan berdasarkan perhitungan estimasi waktu pengerjaan nya. Berikut ini adalah daftar *backlog* pada *Sprint 1*:

Tabel 7. Pembagian *backlog* selama *sprint*

<i>Backlog</i>	<i>Waktu</i>
1. [UI Designer] Desain <i>splash screen</i>	31 Oct 2022 – 11 November 2022
2. [UI Designer] Desain halaman <i>login</i>	
3. [UI Designer] Desain halaman utama	
4. [UI Designer] Desain halaman daftar sepeda	
5. [Android Dev] <i>Slicing UI Splash Screen</i>	

Backlog	Waktu
6. [Android Dev] Slicing UI halaman Login	31 Oct 2022 – 11 November 2022
7. [Android Dev] Slicing UI halaman utama	

Berdasarkan tabel 7 dapat diketahui *sprint* pertama dikerjakan *user story* peminjaman sepeda. Diawali dengan pembuatan halaman *splash*, halaman *login* dan halaman utama. Tahapan pengembangan nya dilakukan dari desain UI halaman-halaman tersebut dan kemudian direalisasikan menggunakan kode. Berikut ini adalah hasil *progress* pada *sprint* pertama:



Gambar 18. (a) Halaman *splash*, (b) Halaman *Login*, (c) UI Halaman utama

b. *Sprint 2*

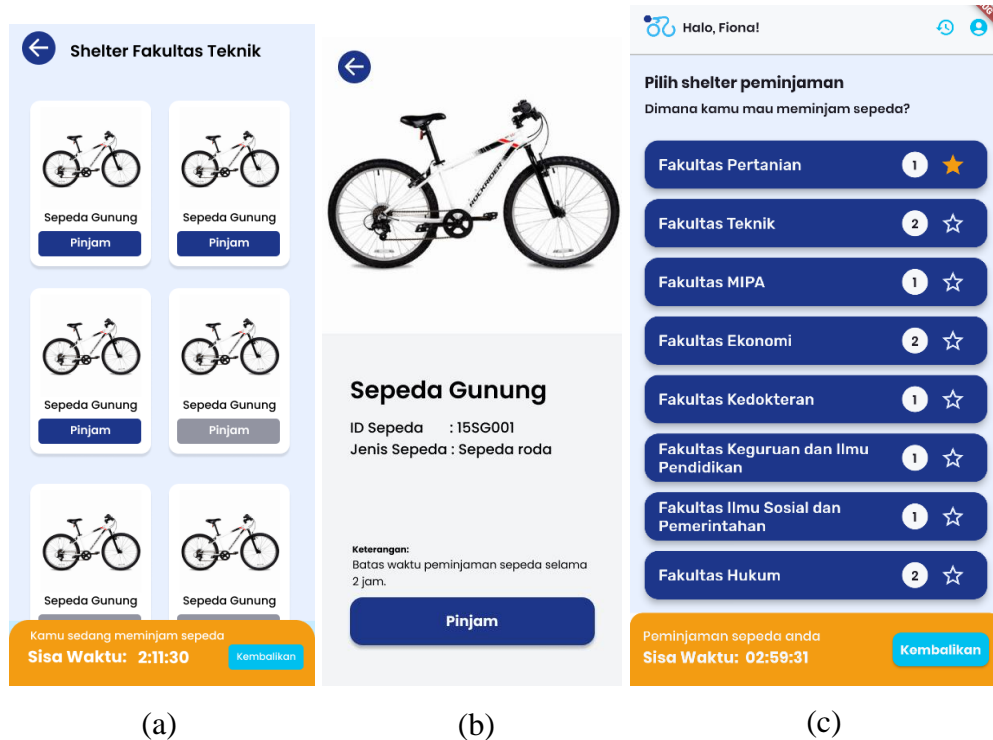
Pada *sprint* kedua juga ditetapkan 7 *backlog* berdasarkan perhitungan estimasi waktu pengerjaan nya. *Sprint* ini masih mengerjakan *user story* pertama yaitu peminjaman sepeda. Dengan *progress* dari *sprint* sebelumnya, pada *sprint* ini dilanjutkan ke halaman detail sepeda.

Tabel 8. *Backlog Sprint 2*

<i>Sprint</i>	<i>Backlog</i>	Waktu
<i>Sprint 2</i>	<ol style="list-style-type: none"> 1. [UI Designer] Desain halaman detail sepeda 2. [Android Dev] <i>Slicing UI</i> halaman daftar sepeda 3. [Android Dev] <i>Slicing UI</i> halaman detail sepeda 4. [Android Dev] <i>Login</i> dengan <i>single sign-on</i> Unila 5. [Android Dev] Integrasi API daftar <i>shelter</i> dengan jumlah sepeda yang tersedia di halaman utama 6. [Android Dev] Integrasi API data sepeda yang tersedia di halaman daftar sepeda 7. [Android Dev] Integrasi API data sepeda di halaman detail sepeda sesuai dengan sepeda yang dipilih 	14 November 2022 – 25 November 2022

Pada tabel 8 diketahui, *sprint 2* sudah memasuki tahap integrasi API untuk beberapa halaman. Sehingga data yang ditampilkan pada aplikasi yang sedang dikembangkan sudah harus berdasarkan *database*. Berikut ini adalah hasil dari *sprint* kedua:

1. UI Desain

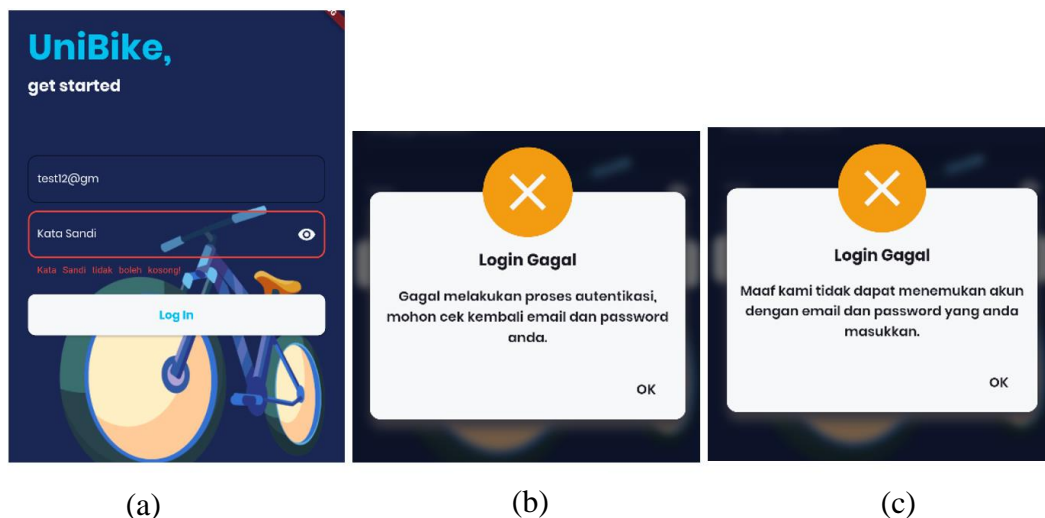


Gambar 19. (a) UI Halaman Daftar Sepeda, (b) UI Halaman Detail Sepeda, (c) Halaman Utama

Pada *sprint* ini juga terdapat penambahan fitur yang didapat Ketika *sprint review* dan *sprint planning* dilakukan yaitu penambahan filter jenis sepeda pada halaman utama. Fitur filter jenis sepeda ini dapat dilihat pada gambar 19 c. Ketika pengguna klik salah satu jenis sepeda, maka angka pada setiap *shelter* akan berubah sesuai jumlah sepeda jenis tersebut. Selain itu tampilan *appbar* di halaman utama juga berubah dimana terdapat status pengguna dengan lampu *indicator*. Status pengguna tersebut berupa sisa waktu yang dimiliki pengguna, keterangan sedang meminjam, atau memiliki denda peminjaman.

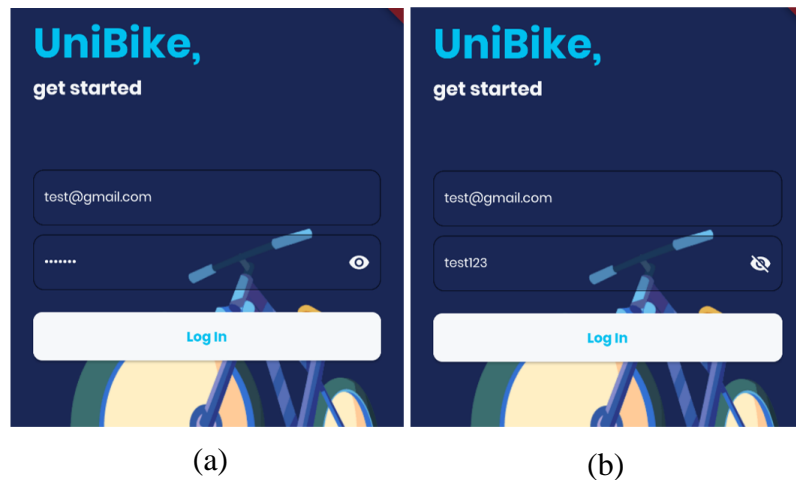
2. Implementasi *Login*

Bentuk keamanan pada aplikasi UniBike adalah otentikasi dimana pengguna perlu *login* dengan *Single Sign-On* Universitas Lampung terlebih dahulu. Sehingga, tidak sembarang orang dapat meminjam sepeda. Pengguna membutuhkan *email* dan kata sandi yang sesuai dengan *database* SSO Universitas Lampung. Terdapat beberapa *pop up dialog* yang akan muncul jika pengguna berusaha masuk dengan *email* atau *password* kosong, *email* atau *password* yang salah, dan jika akun tidak terdaftar.



Gambar 20. (a) *Form email* atau *password* kosong, (b) *Email* atau *password* salah, (c) Akun tidak terdaftar

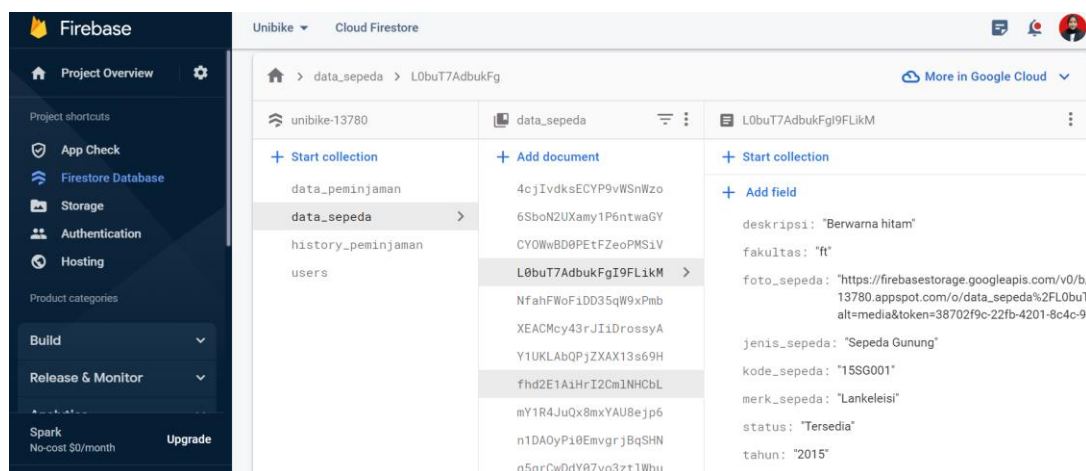
Terdapat fitur asterisk pada form *password*. Hal ini berguna untuk menyembunyikan *password* yang sedang diketik oleh pengguna dari orang-orang sekitar. Sehingga, keamanan dan privasi pengguna terjaga. Fitur dapat dinonaktifkan jika pengguna ingin memastikan bahwa *password* yang dimasukkan benar dengan cara menekan *icon* mata yang berada di *field password*.



Gambar 21. (a) *Field password* ketika asterisk aktif, (b) *Field password* ketika asterisk non aktif

3. Pembuatan *Database*

Database yang digunakan adalah *Firebase*. Ada dua produk *Firebase* yang dipakai yaitu, *Cloud Firestore Database* dan *Firebase Storage*. *Firestore database* digunakan untuk menyimpan data pengguna, data sepeda, data peminjaman, dan data riwayat peminjaman. Sementara *Firebase Storage* digunakan sebagai penyimpanan foto dari sepeda-sepeda yang akan ditampilkan pada aplikasi. Berikut ini adalah penggunaan dari *Firestore Database* untuk pengembangan aplikasi:



Gambar 22. *Firestore Database* untuk aplikasi UniBike

Pada gambar 22 dapat dilihat salah satu contoh dari data sepeda yang ada pada *Firestore Database*. Setiap data memiliki *unique ID* masing-masing, seperti pada *collection* data sepeda yang berisikan dokumen-dokumen sepeda. Pada setiap data sepeda terdapat pasangan *key/value* deskripsi sepeda, fakultas, foto sepeda, jenis sepeda, kode sepeda, merk sepeda, status dan tahun nya. Kemudian data sepeda ini digunakan dalam bentuk JSON dan ditampilkan pada halaman list sepeda dan detail sepeda. Kode sepeda ditentukan berdasarkan tahun, diikuti jenis sepeda dan nomor urut. Kode sepeda yang tertera pada gambar 22 yaitu 15SG001 didapatkan dari tahun 2015, kemudian diikuti dengan jenis sepeda SG yaitu sepeda gunung, dan nomor urut sepeda.

c. Sprint 3

Pada *sprint* ketiga hanya ada 5 *backlog* dimana tiga diantaranya adalah *backlog* hasil *spillover* dari *sprint* sebelumnya. Pada *sprint* ini didapatkan beberapa *error* yang harus diperbaiki. Sehingga, sisa waktu dari pengerjaan *backlog* akan dialihkan untuk penyelesaian *bugs*.

Tabel 9. *Backlog* pada Sprint 3

<i>Sprint</i>	<i>Backlog</i>	Waktu
<i>Sprint 3</i>	<ol style="list-style-type: none"> 1. [<i>Android Dev</i>] Integrasi API daftar <i>shelter</i> dengan jumlah sepeda yang tersedia di halaman utama 2. [<i>Android Dev</i>] Integrasi API data sepeda yang tersedia di halaman daftar sepeda 3. [<i>Android Dev</i>] Integrasi API data sepeda di halaman detail sepeda sesuai dengan sepeda yang dipilih 4. [<i>UI Designer</i>] Desain halaman status peminjaman 	28 November 2022 – 9 Desember 2022

<i>Sprint</i>	<i>Backlog</i>	<i>Waktu</i>
	5. [Android Dev] <i>Slicing UI</i> halaman status peminjaman	

Dapat dilihat pada tabel 9, selain integrasi API juga dilakukan pengerjaan halaman status peminjaman mulai dari desain dan *slicing* UI aplikasinya dengan kode. Pada *sprint* ini dilakukan juga penambahan beberapa fitur yaitu penambahan halaman riwayat peminjaman dan halaman *profile* serta menambah tombol bantuan lupa *password* di halaman *login*. Hasil dari *sprint* ketiga adalah sebagai berikut:

1. Daftar *Shelter* Peminjaman

Fitur ini memudahkan pengguna untuk meminjam sepeda di *shelter* yang diinginkan. Daftar *shelter* juga memuat informasi jumlah sepeda yang ada pada *shelter* tersebut. Selain itu, pengguna juga dapat menambah fakultas ke dalam *shelter* favorit yang akan muncul teratas di daftar *shelter* pada halaman utama. Fitur favorit ini juga mempercepat pengguna untuk menemukan fakultas tanpa perlu menggulir daftar *shelter*.



Gambar 23. Daftar *shelter* peminjaman

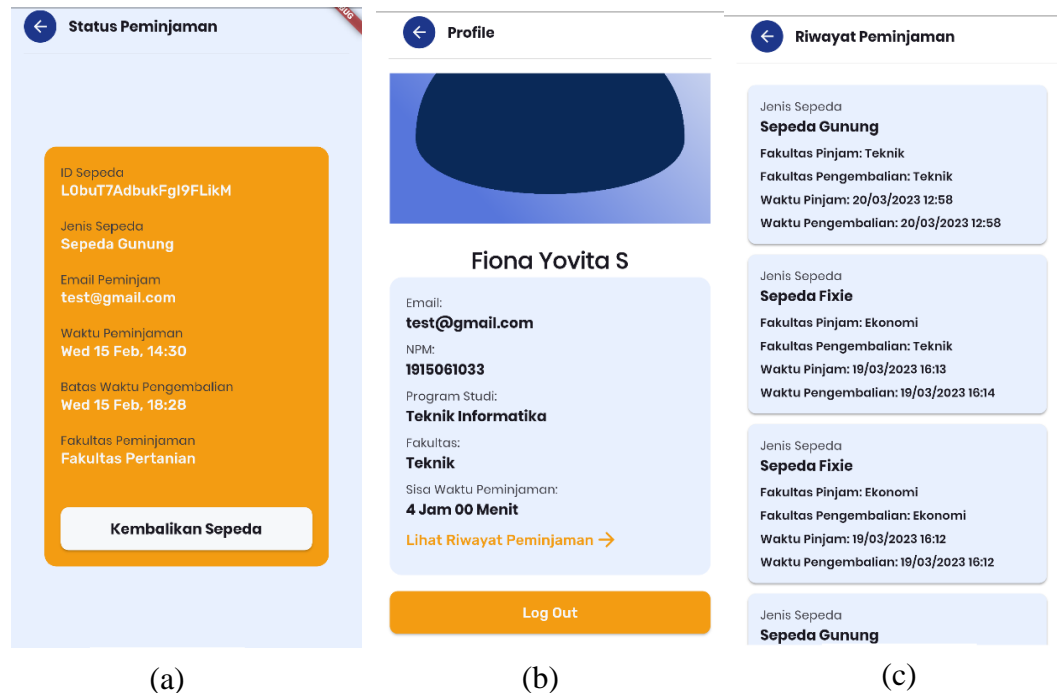
2. Daftar sepeda yang tersedia

Setelah memilih *shelter* yang diinginkan, pengguna akan masuk ke halaman daftar sepeda yang menampilkan sepeda-sepeda di shelter tersebut. Jika sepeda tersebut berstatus ‘Tersedia’ atau dapat dipinjam, maka *button* ‘Pinjam’ berwarna biru dan dapat ditekan. Jika sepeda tidak tersedia atau sedang dipinjam oleh pengguna lain, maka tombol ‘Pinjam’ akan dinonaktifkan dan berwarna abu-abu.



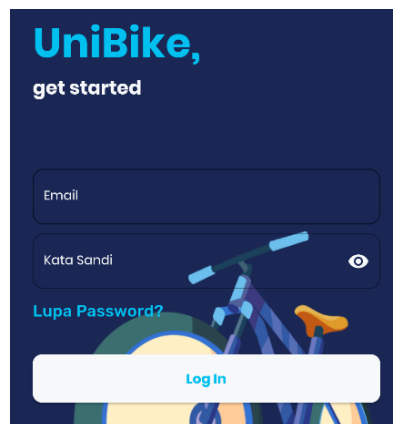
Gambar 24. Daftar sepeda yang tersedia

3. UI Desain



Gambar 25. (a) Halaman Status Peminjaman, (b) Halaman *Profile*, (c) Halaman Riwayat Peminjaman

Berdasarkan gambar 25 dapat dilihat halaman profile yang dapat diakses dengan menekan ikon profile pada *appbar* di halaman utama. Sementara itu pada halaman profile juga terdapat tombol yang mengarah pada halaman riwayat peminjaman.



Gambar 26. Tombol Lupa *Password*

d. Sprint 4

Pada *sprint* 4 hanya tersisa 3 *backlog* yang perlu dikerjakan yaitu integrasi API terkhusus untuk pengembalian sepeda dan informasi status peminjaman. Tetapi, beberapa temuan *error* dari *sprint* sebelumnya akan masuk dalam daftar perbaikan di *sprint* terakhir ini. Selain itu, beberapa perbaikan *bugs* yang telah dilakukan sebelumnya, beberapa belum berhasil diselesaikan.

Tabel 10. *Backlog* pada *sprint* 4

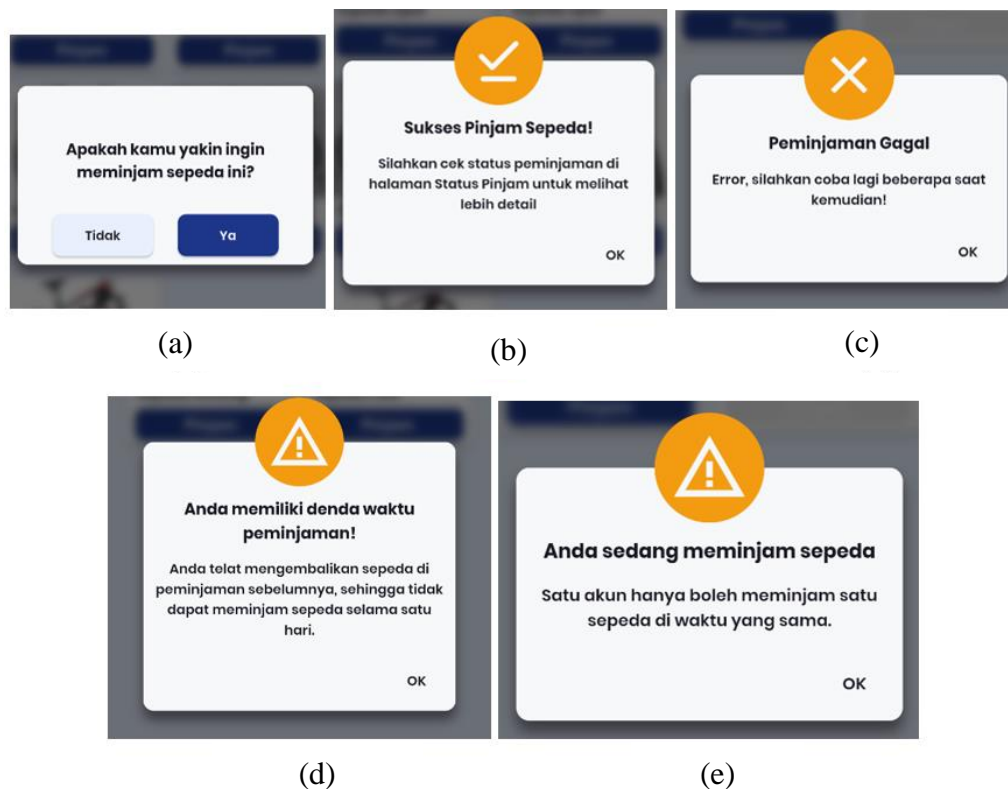
<i>Sprint</i>	<i>Backlog</i>	Waktu
<i>Sprint</i> 4	<ol style="list-style-type: none"> 1. [Android Dev] Integrasi API data peminjaman sepeda <i>user</i> dari <i>Firebase Cloud Firestore</i> (<i>post</i> dan <i>update</i>) 2. [Android Dev] Integrasi API data peminjaman pada halaman Status Peminjaman 3. [Android Dev] <i>Delete</i> dan <i>post</i> data peminjaman di <i>Firebase Cloud Storage</i> setelah sepeda dikembalikan 	12 Desember 2022 – 23 Desember 2022

Pada *sprint* ini juga dikerjakan beberapa penambahan fitur yang dibahas Ketika *sprint review* yaitu mengubah tampilan informasi status pengguna ke bagian *appbar* dan mengubah urutan daftar riwayat peminjaman dari yang terbaru ke peminjaman terlama. Adapun hasil yang didapatkan dari *sprint* terakhir ini adalah sebagai berikut:

1. Peminjaman Sepeda

Dengan adanya aplikasi UniBike, pengguna dapat melakukan peminjaman secara *online*. Sehingga, ketika datang ke *shelter* pengguna hanya perlu mengambil sepeda yang sudah dipinjam melalui aplikasi. Ketika pengguna menekan tombol

pinjam, status sepeda tersebut secara otomatis akan berubah menjadi ‘Tidak Tersedia’. Hal ini berguna agar pengguna lain tidak dapat meminjam sepeda yang sama guna menghindari adanya miskomunikasi antara pengguna. Ada beberapa dialog yang ditampilkan ketika pengguna meminjam sepeda. Ketika pengguna menekan tombol ‘Pinjam’, dialog pertama yang muncul adalah dialog konfirmasi apakah pengguna yakin ingin meminjam sepeda tersebut. Jika pengguna yakin ingin meminjam, maka selanjutnya akan tampil dialog informasi apakah peminjaman berhasil atau gagal. Selain itu juga terdapat dialog ketika pengguna memiliki denda waktu peminjaman tetapi ingin meminjam sepeda. Hal ini berguna agar pengguna tidak dapat meminjam sepeda jika memiliki denda waktu peminjaman dan membatasi peminjaman satu sepeda untuk satu akun.



Gambar 27. (a) Dialog konfirmasi peminjaman, (b) Dialog peminjaman berhasil, (c) Dialog peminjaman gagal, (d) Dialog pengguna memiliki denda pinjam, (e) Dialog pengguna sedang meminjam sepeda

Pada peminjaman sepeda dilakukan fungsi `MeminjamSepeda` dimana dilakukan *create* atau dalam *Firebase* istilahnya adalah *set* data baru untuk peminjaman sepeda. Data peminjaman sepeda berisi pasangan *key/value* dari id sepeda, jenis sepeda, email peminjam, waktu peminjaman, batas waktu pengembalian, dan fakultas pinjam. Batas waktu pengembalian ini didapatkan dari waktu peminjaman ditambah sisa jam peminjaman pengguna. Dimana jika pada hari tersebut pengguna belum meminjam sepeda sekalipun maka sisa jam nya bernilai 4 jam. Jika pengguna sebelumnya sudah meminjam sepeda dan masih ada sisa waktu, misalkan 2 jam, maka saat meminjam sepeda lagi, waktu peminjaman nya akan dipotong sesuai dengan sisa waktu yang dimiliki.

```

MeminjamSepeda() async {
  String emailUser = firebase.currentUser!.email.toString();

  final jenisSepeda = bike.fields.jenisSepeda.value;
  DateTime sisaJamDB = DateTime(today.year, today.month, today.day, today.hour, today.minute, today.second);
  DateTime formattedTime = DateTime(today.year, today.month, today.day, 0, 0, 0);
  Duration timeDifference = sisaJamDB.difference(formattedTime);

  var waktuKembali = today.add(Duration(seconds: sisaJam != "4:00:00" ? timeDifference.inSeconds : 14400));

  FirebaseFirestore.instance.collection('data_peminjaman').doc([firebase.currentUser?.uid]).set(
    {
      'id_sepeda': docId,
      'jenis_sepeda': jenisSepeda,
      'email_peminjam': emailUser,
      'waktu_pinjam': today,
      'waktu_kembali': waktuKembali,
      'fakultas': fakultas
    },
  );

  FirebaseFirestore.instance.collection('data_sepeda').doc(docId).update(
    {'status': 'Tidak Tersedia'},
  );

  FirebaseFirestore.instance.collection('users').doc(docId).update(
    {'status': 1},
  );
}

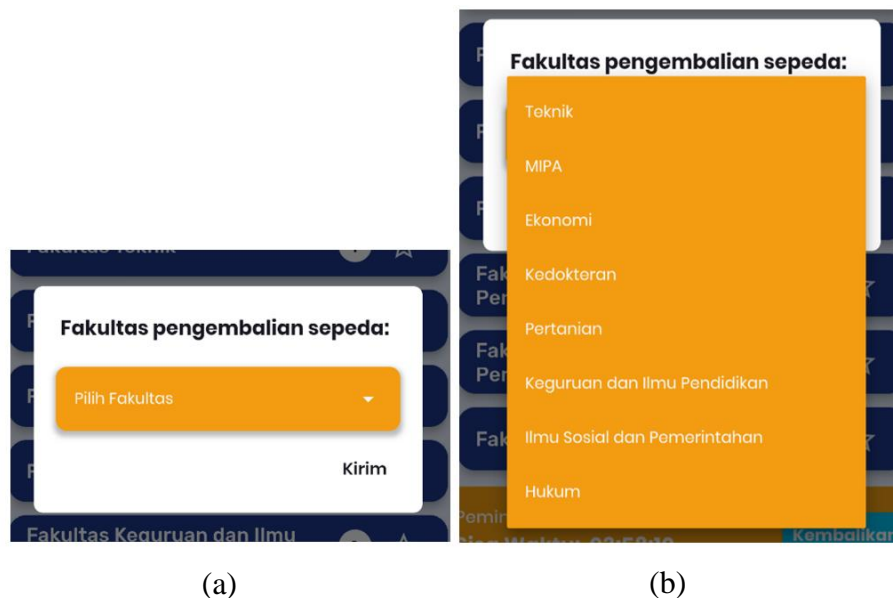
```

Gambar 28. *Code Snippet* Peminjaman Sepeda

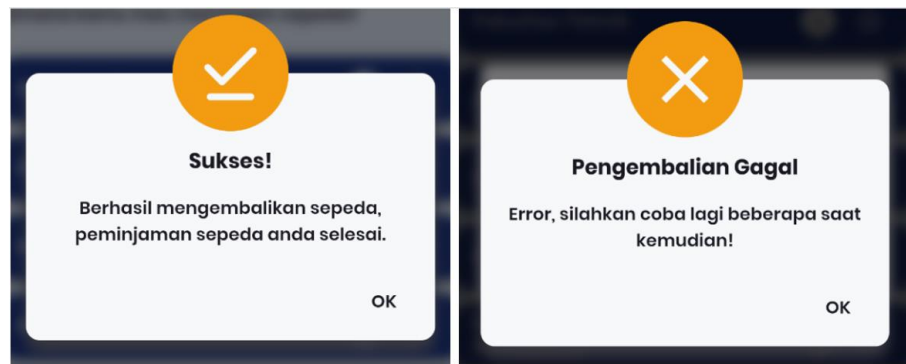
Berdasarkan gambar 28 dapat dilihat bahwa bukan hanya data peminjaman yang diproses ketika meminjam sepeda. Terdapat perubahan atau *update* data pengguna menjadi 1 atau sedang meminjam sepeda dan mengubah status pada data sepeda menjadi tidak tersedia.

2. Pengembalian Sepeda

Selain fitur peminjaman sepeda, aplikasi UniBike juga memiliki fitur pengembalian sepeda. Pengembalian sepeda ini diperlukan untuk mengetahui peminjaman selanjutnya apakah pengguna memiliki denda pinjam atau tidak. Hal ini berguna untuk membatasi peminjaman bagi pengguna yang terlambat mengembalikan sepeda. Pengembalian sepeda dapat dilakukan dengan klik tombol ‘Kembalikan’ pada *footer* di halaman utama maupun halaman daftar sepeda. Ketika ingin mengembalikan sepeda terdapat beberapa dialog yang tampil untuk melakukan konfirmasi di *shelter* mana pengguna ingin mengembalikan sepeda. Terdapat tiga dialog yang tampil ketika pengguna mengembalikan sepeda, dialog pemilihan shelter pengembalian, dialog ketika pengguna belum memilih shelter dan dialog informasi pengembalian berhasil atau gagal.



Gambar 29. (a) Dialog pengembalian sepeda, (b) Dialog memilih shelter pengembalian



(a)

(b)

Gambar 30. (a) Dialog pengembalian berhasil, (b) Dialog pengembalian gagal

Gambar 31. Dialog ketika pengguna belum memilih *shelter*

Pada pengembalian sepeda ini dilakukan fungsi mengembalikanSepeda dimana terjadi penghapusan atau *delete* data peminjaman. Setelah itu, data peminjaman akan dimasukkan ke *collection user history* atau riwayat peminjaman. Data riwayat peminjaman berisikan id sepeda, jenis sepeda, email peminjam, waktu pinjam, waktu pengembalian, fakultas tempat pengguna meminjam, dan fakultas pengembalian nya. Fungsi mengembalikanSepeda dapat dilihat pada gambar 32:

```

mengembalikanSepeda() async {
  FirebaseFirestore.instance.collection('data_peminjaman').doc(firebase.currentUser?.uid)
    .delete().catchError((error) => print("Failed to return bike: $error"));

  firestore.collection('history_peminjaman').doc(firebase.currentUser?.uid)
    .collection('user_history').doc().set(
    {
      'id_sepeda': data['id_sepeda'],
      'jenis_sepeda': data['jenis_sepeda'],
      'email_peminjam': data['email_peminjam'],
      'waktu_pinjam': data['waktu_pinjam'],
      'waktu_kembali': waktuKembali,
      'fakultas_pinjam': data['fakultas'],
      'fakultas_kembali': fakultas
    },
  );

  firestore.collection('data_sepeda').doc('${data['id_sepeda']}').update(
    {'status': 'Tersedia', 'fakultas': fakultasState},
  );

  dendaPinjam();
}

```

Gambar 32. Code Snippet Pengembalian Sepeda

Berdasarkan gambar 32 diketahui data sepeda diperbaharui ketika pengembalian. Status sepeda yang semula tidak tersedia akan berubah menjadi tersedia kembali dan fakultasnya akan diubah sesuai dengan *shelter* pengembalian. Terdapat satu fungsi lainnya yaitu `dendaPinjam` dimana data pengguna akan diubah sesuai dengan kondisi pengembalian nya. Adapun rincian dari fungsi `dendaPinjam` dapat dilihat pada gambar 33.

```

dendaPinjam() {
  var selisihJam = batasWaktu.difference(waktuKembali);

  if (selisihJam.isNegative) {
    users.doc(currentUser).update(
      {
        'status': 0,
        'sisajam': '0:00:00',
        'peminjaman_terakhir': waktuKembali,
        'denda_pinjam': selisihJam
      },
    );
  } else {
    users.doc(currentUser).update(
      {
        'status': 0,
        'sisajam': selisihJam.toString(),
        'peminjaman_terakhir': waktuKembali
      },
    );
  }
}

```

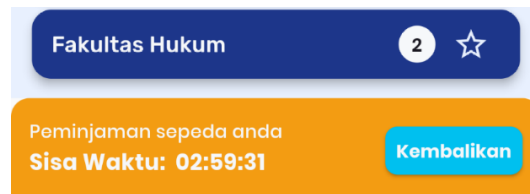
Gambar 33. *Code Snippet* Denda Pinjam

Pada gambar 33 diketahui terdapat percabangan jika waktu pengembalian negatif maka data pengguna akan diperbaharui dengan menambah denda pinjam. Lalu, akan diubah status pengguna menjadi 0 (sedang tidak meminjam sepeda), sisa jam menjadi 0 dan *update field* peminjaman terakhir. Tetapi, jika pengguna tidak terlambat mengembalikan, data pengguna diperbaharui tanpa denda pinjam. Selisih jam pada gambar 32 merupakan variabel untuk mendapatkan selisih waktu pengguna mengembalikan sepeda dengan batas waktu seharusnya pengguna mengembalikan sepeda tersebut.

3. Status Peminjaman Sepeda

Pada aplikasi yang dikembangkan pengguna dapat melihat status peminjaman sepeda nya melalui *bottom menu*. Pengguna dapat masuk ke halaman status peminjaman dengan cara klik *bottom menu*. Halaman status peminjaman menampilkan rincian peminjaman seperti nama peminjam, id sepeda, waktu peminjaman, fakultas peminjaman dan batas waktu pengembalian. Pada halaman utama juga akan menampilkan sisa waktu jika pengguna ingin meminjam sepeda lagi di hari yang sama

dan denda peminjaman ketika pengguna telat mengembalikan sepeda. Ketika sedang meminjam sepeda, terdapat fitur perhitungan mundur waktu peminjaman untuk memudahkan pengguna mengetahui berapa lama sisa waktu peminjaman nya. *Countdown* ini berada di *bottom menu* pada halaman utama dan daftar sepeda.



Gambar 34. *Countdown* waktu peminjaman sepeda

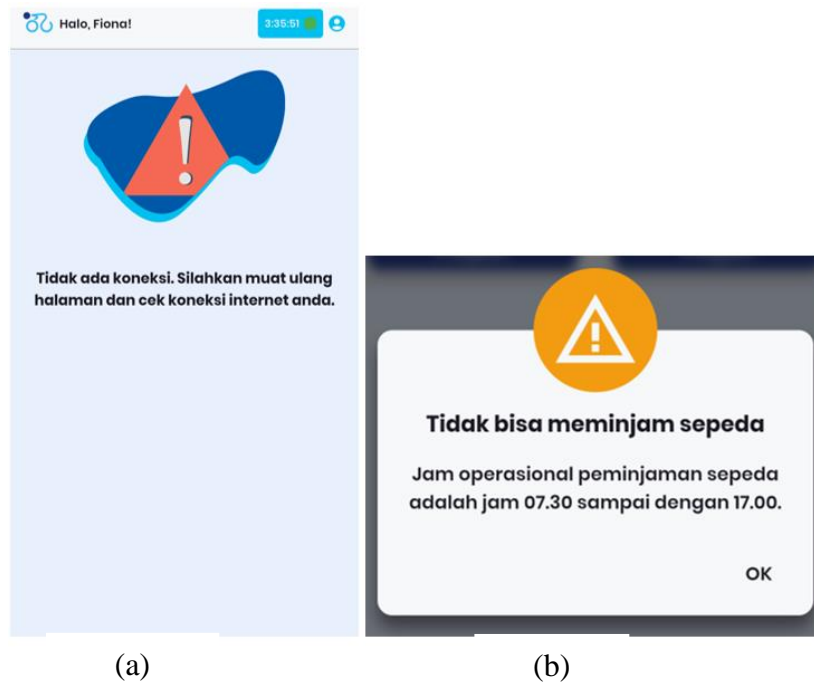


Gambar 35. (a) Informasi sisa waktu peminjaman dan denda waktu peminjaman, (b) Halaman status peminjaman

4. *Availability*

Setelah dilakukan integrasi API untuk peminjaman dan pengembalian sepeda, kemudian dipastikan *availability* sistem sudah berjalan sesuai dengan yang ditentukan.

Sistem peminjaman sepeda membutuhkan koneksi internet untuk dapat melakukan kebutuhan-kebutuhan fungsional nya seperti meminjam sepeda dan mengembalikan sepeda. Dalam hal ini, aplikasi akan menampilkan informasi bahwa tidak ada koneksi internet. Lalu terdapat dialog yang menginformasikan pengguna bahwa pengguna tidak dapat meminjam sepeda diluar jam aktivitas Universitas Lampung.



Gambar 36. (a) Informasi tidak ada koneksi *internet*, (b) Dialog jam operasional peminjaman sepeda

4.1.2.2 *Sprint Review* dan *Retrospective*

Pada tahap *sprint review* didemonstrasikan aplikasi yang telah dikembangkan selama *sprint* untuk memastikan bahwa *sprint backlog* berjalan sesuai dengan rencana. *Sprint Review* dilakukan kurang lebih selama 15 menit di hari jumat minggu terakhir *sprint*. Lalu dilanjutkan dengan *sprint retrospective* yang membahas kinerja selama *sprint*. Terdapat beberapa masukan *backlog* yang didapat dari *product owner* dan *scrum master* selama *sprint review*.

Pada *sprint* pertama seluruh *backlog* dapat diselesaikan dengan baik dan menghasilkan beberapa halaman yaitu *splash screen*, halaman *login*, halaman utama, dan halaman daftar sepeda. Kemudian pada *sprint* kedua terdapat total 7 *backlog* dimana 4 *backlog* dapat diselesaikan sesuai jadwal dan 3 *backlog* mengalami *spill-over*. *Spill over* terjadi dikarenakan terhalang oleh *limit* penggunaan *Read/Write* pada *database* yang digunakan yaitu *Firebase*. Adapun *progress* yang didapatkan pada *sprint* kedua yaitu UI halaman detail sepeda, pembuatan halaman daftar sepeda dan detail sepeda.

Pada *sprint* kedua diimplementasikan *login* dengan *Firebase Authentication* menggunakan email sebagai pengganti simulasi otentikasi menggunakan *Single Sign-On* Unila. Dari 2 *sprint* yang telah diselesaikan, terdapat beberapa *issues* atau *bugs* yang ditemukan. Sehingga, diputuskan pada *sprint* selanjutnya hanya mengerjakan 3 *backlog spill-over*, 2 *backlog* baru, dan menyelesaikan *bugs* yang ditemukan selama masa pengembangan. Selain itu terdapat fitur tambahan yang merupakan masukan dari *scrum master*, yaitu fitur filter jenis sepeda.

Lalu, pada *sprint* ketiga berhasil menyelesaikan integrasi API data sepeda di halaman utama, halaman daftar sepeda dan halaman detail sepeda. Dihasilkan juga UI halaman status peminjaman. Terdapat halaman baru yang dibuat pada *sprint* 3 yaitu halaman riwayat peminjaman dan *profile*. Dimana pada halaman *profile* terdapat tombol untuk navigasi ke halaman riwayat peminjaman.

Selanjutnya pada *sprint* terakhir, dilakukan penyelesaian *logic* untuk peminjaman sepeda, melihat status peminjaman dan mengembalikan sepeda. Integrasi API yang dilakukan meliputi *create*, *read*, *update* dan *delete* data peminjaman, data *user* juta data sepeda ketika melakukan peminjaman dan pengembalian sepeda. Dilakukan juga perubahan tampilan pada bagian *appbar* untuk menampilkan informasi status pengguna dan mengubah urutan daftar riwayat peminjaman.

4.1.3 Pengujian Sistem

4.1.3.1 *Black Box Testing*

Pada penelitian ini dilakukan pengujian unit menggunakan metode *Blackbox Testing* untuk apakah fungsional aplikasi sudah berjalan sesuai kebutuhan yang ditentukan pada bab 4.1.1.1. Pengujian ini dilakukan ke seluruh fitur seperti fitur otentikasi, fitur peminjaman sepeda, pengembalian sepeda dan logout. Teknik yang digunakan pada pengujian di penelitian ini adalah *decision tables*. Pada *decision tables* dibutuhkan *input* untuk menentukan *output* yang sesuai dengan cara menguji beberapa kombinasi *input*. Adapun fitur-fitur yang telah diuji dengan metode *blackbox* adalah sebagai berikut:

a. Otentikasi

Pengujian fitur otentikasi ini dilakukan untuk memastikan peminjaman sepeda hanya dapat dilakukan oleh pengguna yang memiliki akun *Single Sign-On* Universitas Lampung yang valid. Pengujian ini dilakukan untuk menguji kebutuhan non fungsional dengan ID UB-08 yaitu *security*.

Tabel 11. Hasil pengujian *black box* untuk fitur otentikasi

Kode	Skenario	Output yang diharapkan	Hasil
TEST. UB08.01	Melakukan <i>login</i> dengan <i>input email</i> dan <i>password</i> yang valid	Berhasil <i>login</i> dan masuk ke halaman utama	Sesuai

Kode	Skenario	Output yang diharapkan	Hasil
TEST. UB08.02	Melakukan <i>login</i> dengan <i>input email</i> dan <i>password</i> tidak valid	Menampilkan dialog gagal <i>login</i> dengan <i>keterangan email</i> atau <i>password</i> tidak valid.	Sesuai

Pada tabel 11 dapat dilihat bahwa pada fitur otentikasi dilakukan 2 skenario yang berbeda sehingga menghasilkan *output* yang berbeda, Berdasarkan tabel 8 diketahui bahwa semua kondisi pengujian menghasilkan *output* yang sesuai dengan harapan. Oleh karena itu pengujian *blackbox* pada fitur otentikasi dapat disimpulkan telah berhasil.

b. Daftar *Shelter* Peminjaman

Pengujian dilakukan untuk mengetahui apakah kebutuhan fungsional dengan ID UB-01 sudah sesuai dengan yang diharapkan.

Tabel 12. Hasil pengujian daftar *shelter* dan jumlah sepeda

Kode	Skenario	Output yang diharapkan	Hasil
TEST.UB01.01	Melihat daftar shelter peminjaman dan jumlah sepeda yang ada	Halaman utama menampilkan daftar shelter peminjaman dan jumlah sepeda pada shelter.	Sesuai

Pada tabel 12 dapat dilihat bahwa hanya terdapat satu skenario yang diuji. Dalam fitur ini, aplikasi diharuskan menampilkan daftar shelter peminjaman dan jumlah sepeda yang ada pada shelter, sehingga hanya ada satu skenario.

c. Daftar Sepeda yang Tersedia

Pengujian ini dilakukan untuk mengetahui apakah fitur daftar sepeda sudah memenuhi kebutuhan fungsional dengan ID UB-02, yaitu menampilkan daftar sepeda yang tersedia.

Tabel 13. Hasil Pengujian Daftar Sepeda

Kode	Skenario	Output yang diharapkan	Hasil
TEST.UB02.01	Memilih satu shelter peminjaman yang terdapat sepeda di shelter tersebut.	Aplikasi menampilkan daftar sepeda yang tersedia dan daftar sepeda yang tidak tersedia dengan <i>button</i> pinjam nonaktif pada halaman List Sepeda.	Sesuai
TEST.UB02.02	Memilih satu shelter peminjaman dimana tidak ada sepeda yang tersedia.	Aplikasi menampilkan informasi bahwa pada shelter tersebut tidak ada sepeda yang tersedia.	Sesuai

Berdasarkan tabel 13 diketahui hasil pengujian dengan input yang bergantung pada jumlah sepeda pada shelter tersebut. Jika pengguna memilih shelter dengan jumlah sepeda satu atau lebih, maka pada halaman aplikasi akan menampilkan daftar sepeda dengan status tersedia atau tidak tersedia. Sebaliknya, aplikasi akan menampilkan informasi tidak ada sepeda pada shelter tersebut.

d. Peminjaman Sepeda

Pengujian pada fitur peminjaman sepeda bertujuan untuk mengetahui apakah peminjaman berjalan sesuai fungsinya jika terdapat beberapa kondisi *input* yang berbeda. Berikut ini adalah hasil dari pengujian pada fitur peminjaman sepeda:

Tabel 14. Hasil pengujian *black box* fitur peminjaman sepeda

Kode	Skenario	Output yang diharapkan	Hasil
TEST.UB03.01	Meminjam sepeda yang tersedia tanpa denda pinjam dan masih memiliki waktu peminjaman.	Aplikasi menampilkan dialog peminjaman berhasil.	Sesuai
TEST.UB03.02	Meminjam sepeda ketika sedang meminjam sepeda yang lain.	Aplikasi menampilkan dialog peminjaman gagal dengan keterangan hanya boleh meminjam satu sepeda di waktu yang sama.	Sesuai
TEST.UB03.03	Meminjam sepeda ketika memiliki denda waktu peminjaman.	Aplikasi menampilkan dialog peminjaman gagal dengan keterangan pengguna tidak dapat meminjam sepeda karena memiliki denda waktu peminjaman.	Sesuai

Kode	Skenario	Output yang diharapkan	Hasil
TEST. UB03.04	Meminjam sepeda ketika pengguna sudah tidak memiliki sisa waktu peminjaman.	Aplikasi menampilkan dialog peminjaman gagal dengan keterangan pengguna sudah tidak memiliki waktu peminjaman.	Sesuai

Berdasarkan tabel 14 diketahui terdapat 4 skenario dengan masukan yang berbeda seperti denda waktu peminjaman, sisa waktu peminjaman, dan status peminjaman. Pada pengujian ini didapatkan hasil yang sesuai dengan yang diharapkan. Sehingga dapat disimpulkan bahwa fitur peminjaman sepeda yang dikembangkan berhasil.

e. Melihat Status Peminjaman Sepeda

Pengujian fitur status peminjaman dilakukan guna mengetahui apakah detail status peminjaman yang ditampilkan sesuai dengan data yang tersimpan di database.

Tabel 15. Hasil pengujian *blackbox* pada fitur status pinjam

Kode	Skenario	Output yang diharapkan	Hasil
TEST.UB04.01	Melihat status peminjaman ketika sedang meminjam sepeda.	Aplikasi menampilkan <i>bottom menu countdown</i> sisa waktu peminjaman	Sesuai

Kode	Skenario	Output yang diharapkan	Hasil
		dan halaman status peminjaman.	
TEST.UB04.02	Melihat status peminjaman ketika sedang tidak meminjam sepeda.	Aplikasi tidak menampilkan <i>bottom menu</i> peminjaman dan pengguna tidak dapat masuk ke halaman status peminjaman.	Sesuai

Pada tabel 15 diketahui terdapat 2 skenario pengujian pada fitur status peminjaman yang bergantung pada input status pengguna tersebut. Dari tabel 15 didapatkan hasil yang sesuai dengan *output* yang diharapkan. Sehingga, status pengujian nya berhasil. Oleh karena itu, dapat disimpulkan fitur status peminjaman yang dikembangkan berhasil.

f. Pengembalian Sepeda

Pengujian pada fitur pengembalian sepeda dilakukan guna mengetahui apakah data yang disimpan pada database Firebase telah diperbaharui atau dihapus. Hasil pengujian pada fitur pengembalian sepeda adalah sebagai berikut:

Tabel 16. Hasil pengujian fitur pengembalian sepeda

Kode	Skenario	<i>Output</i> yang diharapkan	Hasil
TEST.UB05.01	Mengembalikan sepeda jika sudah memilih shelter pengembalian.	Aplikasi menampilkan dialog pengembalian sepeda berhasil.	Sesuai
TEST.UB05.02	Mengembalikan sepeda jika belum memilih shelter pengembalian.	Aplikasi menampilkan dialog pengguna harus memilih shelter terlebih dahulu.	Sesuai

Pada tabel 16 diketahui terdapat 2 skenario dengan input yang berbeda berdasarkan shelter pengembalian. Dapat dilihat bahwa hasil uji terpenuhi dikarenakan sesuai dengan *output* yang diharapkan. Sehingga dapat disimpulkan bahwa fitur pengembalian sepeda yang telah dikembangkan berhasil.

g. *Availability Sistem*

Aplikasi peminjaman sepeda yang dikembangkan akan berjalan dengan baik jika ponsel terhubung ke koneksi internet yang aktif dan sistem tersedia hanya pada jam operasional UniBike. Pada pengujian ini dilakukan pengecekan kebutuhan non fungsional dengan ID UB-06. Hasil dari pengujian *availability* sistem adalah sebagai berikut:

Tabel 17. Hasil Pengujian *Availability* Sistem

Kode	Skenario	Output yang diharapkan	Hasil
TEST.UB06.01	Meminjam sepeda dengan koneksi internet yang bagus pada jam aktivitas kampus.	Aplikasi berjalan sesuai dengan fungsinya,	Sesuai
TEST.UB06.02	Meminjam sepeda tanpa koneksi internet yang bagus diluar jam aktivitas kampus.	Aplikasi menampilkan informasi tidak ada koneksi internet dan dialog bahwa peminjaman gagal karena diluar jam aktivitas kampus	Sesuai

Berdasarkan tabel 17 diketahui bahwa sistem memerlukan koneksi internet yang baik dan hanya dapat menjalankan fungsional nya pada jam aktivitas kampus. Dari hasil pengujian *availability* sistem ini dapat disimpulkan bahwa aplikasi sudah berhasil dikembangkan sesuai kebutuhan non fungsional nya.

4.1.3.2 *User Experience Questionnaire*

Berikut ini adalah isi dari kuesioner yang diisi oleh pengguna dalam UEQ ini:

Nama :

Pekerjaan :

Silahkan evaluasi produk dengan memilih satu lingkaran tiap baris item.

	1	2	3	4	5	6	7		
menyusahkan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menyenangkan	1
tak dapat dipahami	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat dipahami	2
kreatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	monoton	3
mudah dipelajari	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sulit dipelajari	4
bermanfaat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	kurang bermanfaat	5
membosankan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mengasyikkan	6
tidak menarik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menarik	7
tak dapat diprediksi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dapat diprediksi	8
cepat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	lambat	9
berdaya cipta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	konvensional	10
menghalangi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	mendukung	11
baik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	buruk	12
rumit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sederhana	13
tidak disukai	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	menggembirakan	14
lazim	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	terdepan	15
tidak nyaman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nyaman	16
aman	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak aman	17
memotivasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memotivasi	18
memenuhi ekspektasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak memenuhi ekspektasi	19
tidak efisien	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efisien	20
jelas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	membingungkan	21
tidak praktis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	praktis	22
terorganisasi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	berantakan	23
atraktif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak atraktif	24
ramah pengguna	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	tidak ramah pengguna	25
konservatif	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inovatif	26

CATATAN KRITIK DAN SARAN:

.....
.....
.....

Gambar 37. UEQ yang digunakan

Berdasarkan gambar 37 dapat dilihat 26 pertanyaan dimana skala daya tarik terdapat pada nomor 1, 12, 14, 16, 25 dan 25. Skala kejelasan ada pada pertanyaan nomor 2, 4, 13, 21. Pertanyaan efisiensi terdapat pada nomor 9, 20, 22, 23. Skala ketepatan atau *dependability* terdapat pada pertanyaan nomor 8, 11, 17, dan 19. Kemudian untuk skala stimulasi ada pada nomor 5, 6, 7, dan 18. Dan yang terakhir skala kebaruan atau *novelty* terdapat pada pertanyaan nomor 3, 10, 15, dan 26. Proses pengujian dengan UEQ ini dilakukan dengan total responden yaitu 16 orang meliputi 10 mahasiswa, 4 dosen, dan 2 staff di Universitas Lampung. Berikut ini adalah data yang diperoleh dari pelaksanaan UEQ:

Tabel 18. Data pengujian UEQ

Items																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
7	5	1	1	1	6	7	5	1	1	7	1	7	7	7	7	1	2	1	7	1	7	1	1	1	7
6	7	1	1	1	7	7	6	1	7	7	1	7	7	6	7	5	2	2	7	1	7	1	2	1	7
7	7	2	1	1	5	6	7	4	2	6	2	6	6	6	5	1	1	2	5	2	6	2	3	3	6
6	6	1	1	1	5	6	6	3	1	7	1	5	6	7	7	4	2	2	6	2	7	2	1	2	7
6	6	7	1	1	5	6	6	1	2	6	1	7	7	6	6	3	3	2	6	2	6	3	5	2	6
5	7	3	1	1	5	5	4	2	4	7	1	7	6	7	5	1	1	1	7	1	7	1	2	3	7
5	6	2	3	2	5	5	5	2	5	5	2	6	6	6	5	2	2	3	5	2	5	2	2	2	2
6	6	1	2	2	6	7	6	1	1	7	1	6	7	6	7	2	1	2	6	2	7	1	2	2	7
7	7	2	2	1	7	7	7	2	3	7	1	7	7	7	7	1	1	1	7	1	7	1	1	1	7
7	6	6	5	6	6	6	6	6	4	7	6	7	7	6	5	2	1	2	6	3	7	2	3	2	5
7	6	1	2	1	7	7	5	1	7	7	1	7	7	6	7	1	1	2	7	1	7	1	1	2	7
6	7	2	1	1	6	7	7	1	1	7	2	7	7	7	7	1	1	1	7	1	7	1	1	2	7
7	6	1	2	1	7	7	6	1	2	7	1	7	6	6	7	1	1	2	7	2	6	1	1	1	7
6	7	1	1	1	5	5	5	1	5	7	1	5	7	5	7	1	1	1	7	2	6	6	1	1	6
5	6	1	2	1	5	6	6	7	3	7	6	5	6	7	7	1	2	2	6	2	6	2	2	2	6
7	7	2	1	1	6	7	7	1	1	7	1	7	7	7	1	1	1	1	7	1	7	2	2	1	7

Dari 16 responden yang mengisi 26 pertanyaan dengan cara menghitamkan nilai dari rentang 1 sampai dengan 7, kemudian didapatkan data seperti pada tabel 18. UEQ menerapkan pengacakan pertanyaan negatif dengan positif guna menyaring jawaban responden dimana jawaban kuesioner tidak selalu berurut negatif yang bernilai 1 hingga positif bernilai 7, begitu pula sebaliknya. Sehingga didapatkan penilaian yang bervariasi.

Tabel 19. Hasil perhitungan data

Items																									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
3	1	3	3	3	2	3	1	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3
2	3	3	3	3	3	3	2	3	-3	3	3	3	3	2	3	-1	2	2	3	3	3	3	2	3	3
3	3	2	3	3	1	2	3	0	2	2	2	2	2	2	1	3	3	2	1	2	2	2	1	1	2
2	2	3	3	3	1	2	2	1	3	3	3	1	2	3	3	0	2	2	2	2	3	2	3	2	3
2	2	-3	3	3	1	2	2	3	2	2	3	3	3	2	2	1	1	2	2	2	2	1	-1	2	2
1	3	1	3	3	1	1	0	2	0	3	3	3	2	3	1	3	3	3	3	3	3	3	2	1	3
1	2	2	1	2	1	1	1	2	-1	1	2	2	2	2	1	2	2	1	1	2	1	2	2	2	-2
2	2	3	2	2	2	3	2	3	3	3	3	2	3	2	3	2	3	2	2	2	3	3	2	2	3
3	3	2	2	3	3	3	3	2	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	2	-2	-1	-2	2	2	2	-2	0	3	-2	3	3	2	1	2	3	2	2	1	3	2	1	2	1
3	2	3	2	3	3	3	1	3	-3	3	3	3	3	2	3	3	3	2	3	3	3	3	3	2	3
2	3	2	3	3	2	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	2	3
3	2	3	2	3	3	3	2	3	2	3	3	3	2	2	3	3	3	2	3	2	2	3	3	3	3
2	3	3	3	3	1	1	1	3	-1	3	3	1	3	1	3	3	3	3	3	2	2	-2	3	3	2
1	2	3	2	3	1	2	2	-3	1	3	-2	1	2	3	3	3	2	2	2	2	2	2	2	2	2
3	3	2	3	3	2	3	3	3	3	3	3	3	3	3	-3	3	3	3	3	3	3	2	2	3	3

Data yang diperoleh kemudian diubah seperti pada tabel 19 dimana nilai 1 sampai 7 dibagi menjadi dua bagian. Setengah dari penilaian tersebut bernilai positif dan setengahnya lagi adalah negatif, sehingga pada tabel 14, nilai 3 mewakili nilai paling positif dan -3 mewakili nilai paling negatif.

Tabel 20. Hasil perhitungan nilai rata-rata

UEQ Scales (Mean and Variance)		
Attractiveness	↑ 2.396	0.19
Perspicuity	↑ 2.484	0.19
Efficiency	↑ 2.297	0.47
Dependability	↑ 2.438	0.23
Stimulation	↑ 2.344	0.30
Novelty	↑ 2.188	0.53

Data dari tabel 19 kemudian dihitung nilai rata-rata untuk setiap kategori sehingga didapatkan nilai rata-rata dan variasi seperti pada tabel 20.

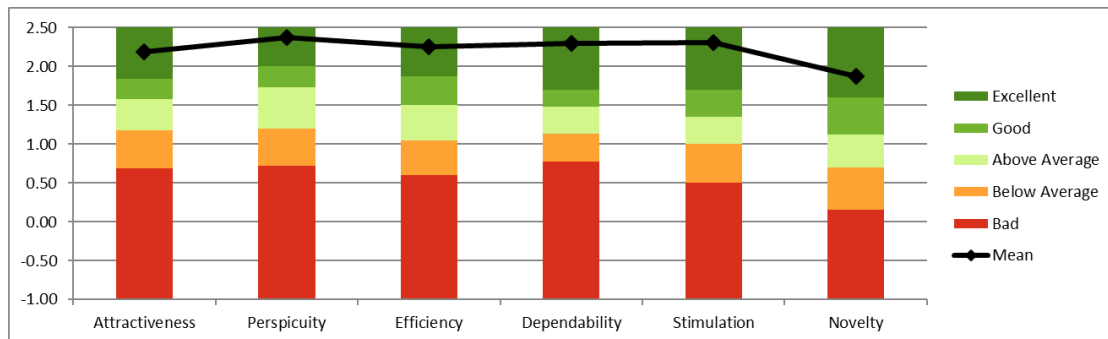
4.2 Pembahasan

4.2.1 Analisa Hasil *Black Box Testing*

Berdasarkan pengujian menggunakan metode *black box* yang telah dilakukan pada bab 4.3.1.1 menunjukkan bahwa seluruh pengujian memiliki status berhasil. Pengujian dilakukan untuk memeriksa seluruh kebutuhan fungsional dan non-fungsional yang dirinci dari *use case* sistem peminjaman sepeda. Sehingga, dapat ditarik kesimpulan bahwa aplikasi yang dikembangkan sudah berhasil berjalan sesuai dengan *use case* yang ditentukan sebelumnya.

4.2.2 Analisa Hasil *User Experience Questionnaire*

UEQ dilakukan untuk mengetahui kualitas produk yang dikembangkan dengan produk-produk lain yang sudah sukses. Dari nilai UEQ akan diketahui apakah produk yang dikembangkan bernilai sangat baik atau kurang baik dari berbagai kategori seperti orisinalitas dan daya tarik bagi pengguna.



Gambar 38. Grafik hasil perbandingan *benchmark data set*

Nilai rata-rata dari setiap kategori pertanyaan pada UEQ di tabel 17 kemudian dibandingkan dengan *benchmark data set* yang merupakan kumpulan data yang telah diisi oleh 21.175 orang dari 468 studi tentang berbagai produk. Dari gambar 38 dapat disimpulkan bahwa pada kategori *attractiveness* (daya tarik) didapatkan nilai 2,19. Kategori *perspicuity* (kejelasan) memperoleh nilai 2,38. Kategori efisiensi mendapatkan nilai 2,25. Pada kategori *dependability* (ketepatan) memperoleh nilai 2,30. Lalu kategori stimulasi mendapatkan nilai 2,31 dan yang terakhir kategori *novelty* (kebaruan) mendapat nilai 1,88. Berdasarkan UEQ yang telah dilakukan, aplikasi UniBike yang dikembangkan memperoleh nilai *excellent* atau baik sekali di semua kategori dengan nilai tertinggi pada kategori *perspicuity* atau kejelasan.

Attractiveness		Perspicuity		Efficiency		Dependability		Stimulation		Novelty	
Items	Correlation	Items	Correlation	Items	Correlation	Items	Correlation	Items	Correlation	Items	Correlation
1, 12	0.33	2, 4	0.17	9, 20	0.45	8, 11	0.18	5, 6	0.35	3, 10	0.08
1, 14	0.26	2, 13	0.15	9, 22	0.30	8, 17	0.01	5, 7	0.50	3, 15	0.49
1, 16	0.07	2, 21	0.34	9, 23	0.54	8, 19	0.25	5, 18	0.23	3, 26	0.47
1, 24	0.07	4, 13	0.29	20, 22	0.41	11, 17	0.04	6, 7	0.40	10, 15	0.18
1, 25	0.20	4, 21	0.31	20, 23	0.30	11, 19	0.62	6, 18	0.61	10, 26	0.35
12, 14	0.26	13, 21	0.62	22, 23	0.33	17, 19	0.29	7, 18	0.24	15, 26	0.38
12, 16	0.07	Average	0.31	Average	0.39	Average	0.23	Average	0.39	Average	0.33
12, 24	0.44	Alpha	0.65	Alpha	0.72	Alpha	0.55	Alpha	0.72	Alpha	0.66
12, 25	0.44	Conf. Int.	0.17	Conf. Int.	0.33	Alpha	-0.07	Conf. Int.	0.34	Conf. Int.	0.20
14, 16	0.42	Alpha (5%)	0.85	Alpha (5%)	0.88	Alpha (5%)	0.81	Alpha (5%)	0.88	Alpha (5%)	0.86
14, 24	0.10										
14, 25	0.25										
16, 24	0.54										
16, 25	0.33										
24, 25	0.59										
Average	0.29										
Alpha	0.71										
Conf. Int.	0.35										
Alpha (5%)	0.87										

Gambar 39. Korelasi Antar Pertanyaan

Berdasarkan gambar 39 didapatkan hasil *alpha* dari UEQ yang dilakukan untuk semua kategori. Hasil *alpha* ini merupakan nilai korelasi antara dua pertanyaan pada satu kategori dengan nilai yang berbeda. Nilai korelasi ini digunakan untuk melihat apakah responden benar-benar menjawab pertanyaan dengan konsisten. Perhitungan terhadap nilai korelasi ini menggunakan *Cronbachs Alpha-Coefficient*, dimana nilai *alpha* >0.7 sudah dapat dikatakan konsisten. Dari gambar 39 dilihat untuk kategori *attractiveness* memiliki nilai *alpha* 0.71, *efficiency* dengan nilai 0.72, dan *stimulation* dengan nilai 0.72. Tetapi untuk kategori *perspicuity*, *dependability* dan *novelty* nilai *alpha* masih dibawah 0.7. Sehingga disimpulkan bahwa terdapat jawaban responden yang belum konsisten pada kategori kejelasan, ketepatan dan kebaruan.

4.2.3 Penerapan Metode *Agile Scrum*

Pada penelitian ini metode pengembangan perangkat lunak yang digunakan adalah *Agile Scrum*. Selama masa pengembangan setiap backlog dipikirkan rincian tugasnya ketika sprint planning, lalu terdapat catatan progress setiap hari nya karena ada aktivitas *daily scrum* yang memudahkan pengembang untuk mengetahui kinerja nya. Ketika *sprint review*, setiap tugas dilakukan demonstrasi sehingga dapat diketahui

apakah tugas yang dikerjakan selesai dan fungsionalitas nya berjalan dengan benar. Pada saat sprint review kedua, terdapat tiga *backlog* dengan status *spill-over*, dimana kondisi dua dari *backlog* tersebut selesai tetapi fungsionalitasnya belum mencapai yang diharapkan sehingga perlu dikerjakan lagi di sprint selanjutnya. Sementara backlog yang lain nya memang belum selesai pada *sprint* tersebut dan harus dilanjutkan di *sprint* selanjutnya. Dengan metode *Agile Scrum*, pengembang mengetahui dengan jelas apakah *software* tersebut perlu perbaikan atau tidak karena di setiap akhir sprint akan ditinjau aplikasi yang telah dikembangkan. Pada penelitian ini seluruh *backlog* berhasil diselesaikan dalam masa pengembangan 4 *sprint*.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan yang diperoleh pada penelitian ini, terdapat beberapa kesimpulan sebagai berikut:

1. Berhasil dikembangkannya sebuah aplikasi Peminjaman Sepeda berbasis Android dengan nama UniBike menggunakan bahasa pemrograman *Flutter*. Aplikasi yang dikembangkan menggunakan metode *Agile Scrum* ini memiliki fitur peminjaman sepeda secara *online*, mengembalikan sepeda sesuai *shelter* yang dipilih, menampilkan daftar sepeda, dan menampilkan status peminjaman.
2. Berdasarkan hasil pengujian menggunakan metode *black box*, fitur-fitur yang dikembangkan telah berjalan sesuai dengan kebutuhan fungsional dan non-fungsional yang ditetapkan. Ada total 11 skenario yang menguji kebutuhan fungsionalnya yaitu menyediakan daftar *shelter*, daftar sepeda dan status peminjaman, serta fitur peminjaman dan pengembalian sepeda. Kemudian, ada 4 skenario untuk menguji kebutuhan non fungsional berupa *availability* dan proses otentikasi.
3. Berdasarkan hasil evaluasi *User Experience Questionnaire* dengan 16 responden, aplikasi peminjaman sepeda UniBike mendapatkan nilai *excellent* atau sangat baik pada 6 kategori yaitu *attractiveness*, *perspicuity*, *efficiency*,

dependability, *stimulation* dan *novelty*. Nilai *excellent* di rentang 1.88 – 2.38 memberikan arti bahwa sistem yang dikembangkan sudah baik dan dapat diterima oleh pengguna.

5.2 Saran

Berdasarkan penelitian yang dilakukan, terdapat beberapa saran untuk pengembangan selanjutnya yaitu sebagai berikut:

1. Penambahan *dashboard* atau laman untuk admin mengelola data sepeda, data pengguna dan data peminjaman sepeda.
2. Fitur *scan QR Code* dalam peminjaman sepeda untuk membuka kunci sepeda pada shelter peminjaman untuk menjamin keamanan sepeda.
3. Fitur *GPS (Global Positioning System)* untuk melacak keberadaan sepeda ketika dipinjam.

DAFTAR PUSTAKA

- [1] G. Unila, “Transportasi Hijau,” *greenmetricunila.ac.id*, 2022.
<https://greenmetric.unila.ac.id/category/kampus-hijau-unila/transportasi-hijau/>
(accessed Aug. 20, 2022).
- [2] U. Lampung, “Sejarah Universitas Lampung,” *unila.ac.id*, 2018.
<https://www.unila.ac.id/sejarah-universitas-lampung/> (accessed Aug. 20, 2022).
- [3] R. Unila, “Bus Kampus Unila Resmi Beroperasi,” *unila.ac.id*, 2018.
<https://www.unila.ac.id/bus-kampus-unila-resmi-beroperasi/> (accessed Aug. 20, 2022).
- [4] I. P. Bogor, “Bike Sharing,” *greencampus.ipb.ac.id*.
<https://greencampus.ipb.ac.id/moda-transportasi/bike-sharing/> (accessed Sep. 08, 2022).
- [5] S. Kemp, “DIGITAL 2022: INDONESIA,” 2022.
<https://datareportal.com/reports/digital-2022-indonesia> (accessed Sep. 01, 2022).
- [6] U. N. Malang, *Buku Ajar Olahraga Balap Sepeda*. Malang: Universitas Negeri Malang, 2019.
- [7] W. A. Hutagaol, M. B. Sanjaya, and P. A. Telnoni, “Aplikasi Peminjaman Sepeda Berbasis Web Dan Android,” in *e-Proceeding of Applied Science*, 2015, vol. 1, no. 3, pp. 1742–1751.
- [8] A. Kadir, *Pengenalan Sistem Informasi Edisi Kesatu*. Yogyakarta: ANDI Yogyakarta. doi: 10.13140/2.1.2637.6328.
- [9] I. T. M. Daeng, N. . Mewengkang, and E. R. Kalesaran, “Penggunaan

Smartphone Dalam Menunjang Aktivitas Perkuliahan Oleh Mahasiswa Fispol Unsrat Manado,” *Acta Diurna*, vol. 6, no. 1, pp. 1–15, 2017, [Online].

Available:

<https://ejournal.unsrat.ac.id/v3/index.php/actadiurnakomunikasi/article/view/15482>














- [10] G. Developer, *Android Developer Fundamental Course*, vol. 2017, no. July. Google Developer Training Team, 2016. [Online]. Available: [http://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/RSA Distribution Tariff Code Vers 6.pdf](http://www.eskom.co.za/CustomerCare/TariffsAndCharges/Documents/RSA%20Distribution%20Tariff%20Code%20Vers%206.pdf)<http://www.nersa.org.za/>
- [11] M. L. Napoli, *Beginning Flutter®*. Indianapolis: John Wiley & Sons, Inc., 2019. doi: 10.1002/9781119550860.
- [12] V. Radha and D. H. Reddy, “A Survey on Single Sign-On Techniques,” *Procedia Technol.*, vol. 4, pp. 134–139, 2012, doi: 10.1016/j.protcy.2012.05.019.
- [13] M. Meng, S. Steinhardt, and A. Schubert, “Application programming interface documentation: What do software developers want?,” *J. Tech. Writ. Commun.*, vol. 48, no. 3, pp. 295–330, 2018, doi: 10.1177/0047281617721853.
- [14] D. Crockford, “Introducing JSON,” 1999. <https://www.json.org/json-en.html> (accessed Sep. 08, 2022).
- [15] C. Khawas and P. Shah, “Application of Firebase in Android App Development-A Study,” *Int. J. Comput. Appl.*, vol. 179, no. 46, pp. 49–53, 2018, doi: 10.5120/ijca2018917200.
- [16] K. Schwaber and J. Sutherland, *Panduan Scrum*, vol. 2, no. 12. 2017. [Online]. Available: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Indonesian.pdf>
- [17] H. Gunawan, “Scrum: Metode Efektif dalam Project Management,” *Hashmicro*, 2022. <https://www.hashmicro.com/id/blog/scrum-adalah/> (accessed May 04, 2023).
- [18] S. Dharwiyanti and R. S. Wahono, *Pengantar Unified Modeling Language (UML)*. 2003. [Online]. Available: <http://www.unej.ac.id/pdf/yanti-uml.pdf>

- [19] N. E. T. Core, *Unit Testing : The Complete Guide*, vol. 7, no. 3. Bedford: Progress Software Corporation, 2021.
- [20] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012, doi: 10.5121/ijesa.2012.2204.
- [21] M. Schrepp, *User Experience Questionnaire Handbook Version 8*. 2019. [Online]. Available: www.ueq-online.org
- [22] Atlassian, “About Trello,” 2021. <https://trello.com/about> (accessed Sep. 08, 2022).
- [23] A. Prayogo *et al.*, “Aplikasi Android ‘ Gamarc ’ Untuk Mendukung Konsep Transportasi Hijau Sepeda Kampus – Studi Kasus Ugm,” in *PROSIDING SEMINAR NASIONAL GEOTIK 2019*, 2019, pp. 316–324. doi: 10.1016/j.dss.2016.09.006.
- [24] R. Prasetya and H. Supriyono, “Sistem Peminjaman Sepeda Otomatis berbasis QR Code,” *Emit. J. Tek. Elektro*, vol. 20, no. 1, pp. 26–31, 2019, doi: 10.23917/emit.v20i1.8461.
- [25] A. Christyana, T. Angel, and U. P. Raya, “Implementasi Aplikasi Android untuk Sistem Penyewaan Sepeda,” *J. Teknol. Inf. J. Keilmuan Dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, pp. 195–200, 2022.
- [26] K. N. F. Ku Azir *et al.*, “UniCycle: An Android Application of Bike Sharing System in the Digital Campus,” *J. Phys. Conf. Ser.*, vol. 1755, no. 1, 2021, doi: 10.1088/1742-6596/1755/1/012010.
- [27] L. Bajracharya, T. Mulya, A. Purbasari, and M. Hwang, “A study on cost-effective and eco-friendly bicycle sharing system for developing countries,” *Lect. Notes Electr. Eng.*, vol. 514, pp. 523–531, 2019, doi: 10.1007/978-981-13-1056-0_52.
- [28] A. S. Badriah and P. K. Sari, “Uji usability pada penggunaan Aplikasi Gowes di Telkom,” in *e-Proceeding of Management*, 2019, vol. 6, no. 2, pp. 2864–2872.
- [29] M. A. Dewi and R. Irham, “Penerapan Agile Scrum Pada Pengembangan











- Bimbingan Daring Skripsi Mahasiswa,” *J. SISKOM-KB (Sistem Komputer. dan Kecerdasan Buatan)*, vol. 4, no. 2, pp. 40–45, 2021, doi: 10.47970/siskom-kb.v4i2.195.
- [30] I. Urooj, J. Jabbar, and N. Azeem, “Application centric cloud-based notification system using scrum methodology,” in *Proceedings of the 14th IEEE Conference on Industrial Electronics and Applications, ICIEA 2019*, 2019, pp. 2517–2521. doi: 10.1109/ICIEA.2019.8833824.
 - [31] M. B. Legowo, B. Indiarto, and D. Prayitno, “Agile Software Methodology with Scrum for Developing Quality Assurance System,” in *Proceedings - 2019 2nd International Conference of Computer and Informatics Engineering: Artificial Intelligence Roles in Industrial Revolution 4.0, IC2IE 2019*, 2019, pp. 104–109. doi: 10.1109/IC2IE47452.2019.8940831.
 - [32] Asroni, S. Riyadi, and T. Cahyono, “Information System for Providing Food Services Based on Mobile Application Using Flutter Framework,” in *Proceedings of the 4th International Conference on Sustainable Innovation 2020–Technology, Engineering and Agriculture (ICoSITEA 2020)*, 2021, vol. 199, no. ICoSITEA 2020, pp. 164–169. doi: 10.2991/aer.k.210204.031.

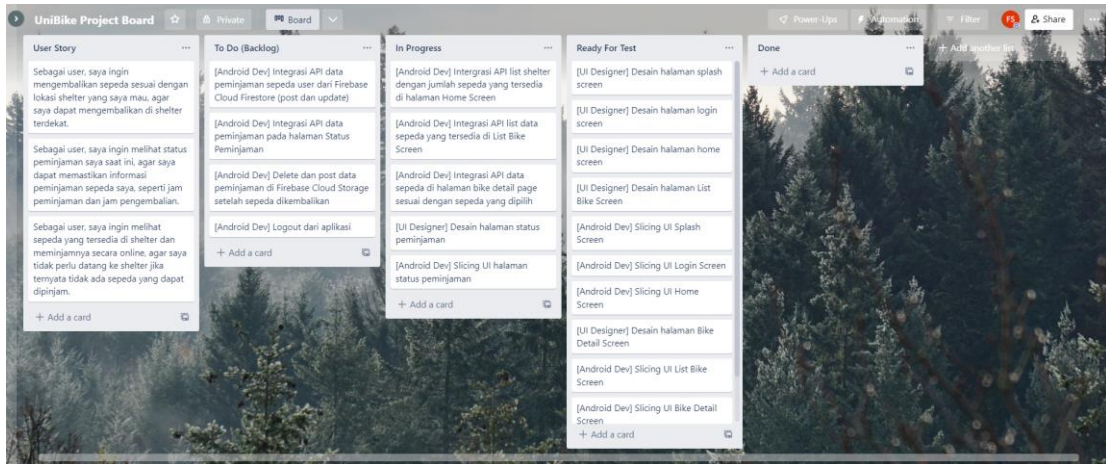
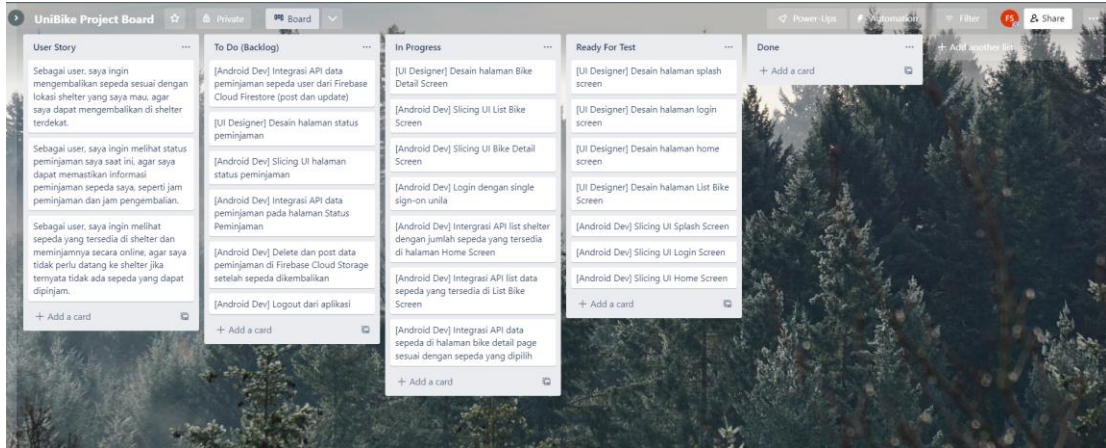
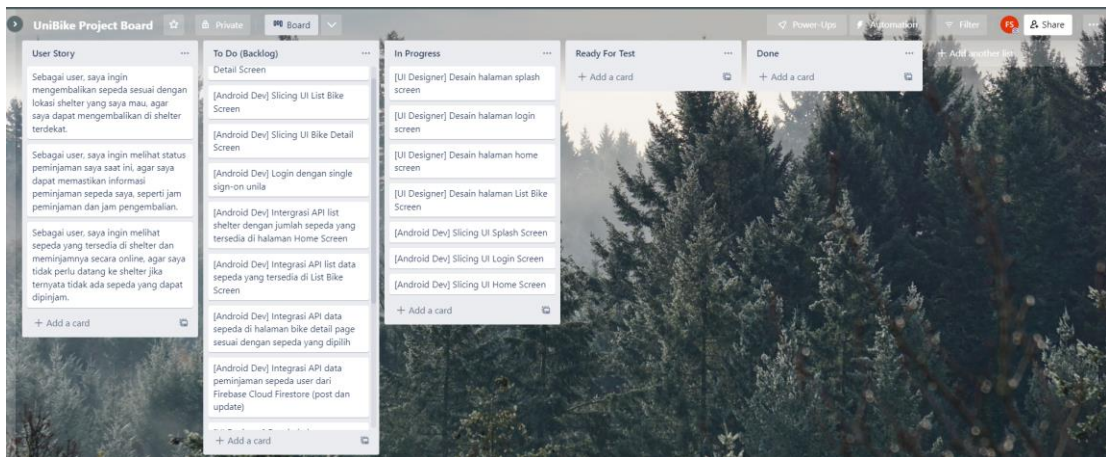
LAMPIRAN

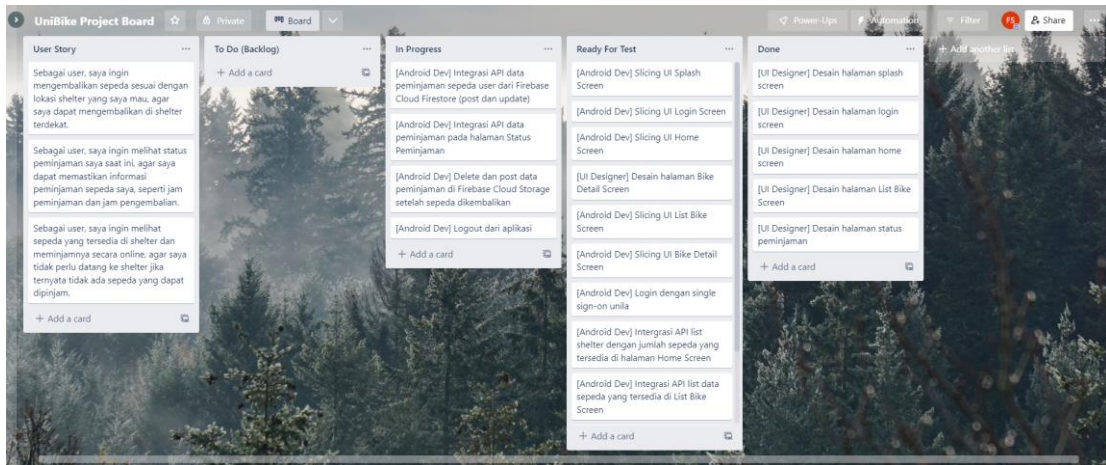
URL: https://github.com/fionayovita/unibike_progress.git

 fionayovita new feature filtering bike, and changing time status		9f74092 2 weeks ago
	android	new feature filtering bike, and changing time status
	assets	reducing file apk size
	ios	updating code new unibike
	lib	new feature filtering bike, and changing time status
	test	updating code new unibike
	web	updating code new unibike
	.gitignore	updating code new unibike
	.metadata	updating code new unibike
	README.md	updating code new unibike
	analysis_options.yaml	updating code new unibike
	pubspec.lock	updating code new unibike
	pubspec.yaml	updating code new unibike

 main ▾ [unibike_progress](#) / [lib](#) /

<div> fionayovita</div> new feature filtering bike, and changing time status		
..		
<div></div> api	reducing file apk	
<div></div> common	new feature filtering bike, and changing time status	
<div></div> function	new feature filtering bike, and changing time status	
<div></div> model	new feature filtering bike, and changing time status	
<div></div> provider	reducing file apk	
<div></div> ui	new feature filtering bike, and changing time status	
<div></div> utils	reducing file apk	
<div></div> widgets	new feature filtering bike, and changing time status	
<div></div> main.dart	new feature filtering bike, and changing time status	





Turnitin Check Skripsi_260523

ORIGINALITY REPORT

12%

SIMILARITY INDEX

11%

INTERNET SOURCES

4%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

digilib.unila.ac.id

Internet Source

1%

2

Submitted to Universitas Brawijaya

Student Paper

1%

3

www.researchgate.net

Internet Source

<1%

4

repository.ub.ac.id

Internet Source

<1%

5

123dok.com

Internet Source

<1%

6

repositori.usu.ac.id

Internet Source

<1%

7

docplayer.info

Internet Source

<1%

8

text-id.123dok.com

Internet Source

<1%

9

Reinaldi Prasetya, Heru Supriyono. "Sistem Peminjaman Sepeda Otomatis berbasis QR Code", Emitter: Jurnal Teknik Elektro, 2019

<1%