

**IMPLEMENTASI METODE SCRUM DALAM
PENGEMBANGAN PERANGKAT LUNAK
DELIV**

SKRIPSI

**Oleh:
NAUFALDI RAFIF SATRIYA
NIM. 14650068**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

HALAMAN JUDUL

**IMPLEMENTASI METODE SCRUM DALAM PENGEMBANGAN
PERANGKAT LUNAK DELIV**

SKRIPSI

Diajukan kepada:

**Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Oleh:

**NAUFALDI RAFIF SATRIYA
NIM. 14650068**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

HALAMAN PERSETUJUAN

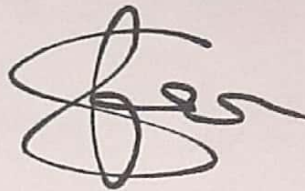
**IMPLEMENTASI METODE SCRUM DALAM
PENGEMBANGAN PERANGKAT LUNAK
DELIV**

SKRIPSI

Oleh:
NAUFALDI RAFIF SATRIYA
NIM. 14650068

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : Juni 2021

Pembimbing I



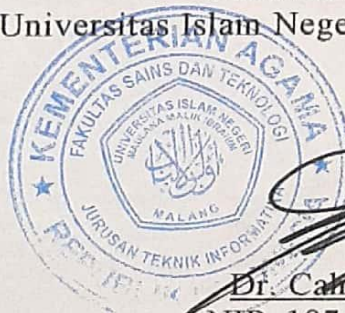
Dr. Ir. M. Amin Hariyadi, M.T
NIP. 19670118 200501 1 001


Pembimbing II



Puspa Miladin N. S A B, M.Kom
NIP. 19930828 201903 2 018

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Cahyo Crysdiyan M. CS
NIP. 19740424 200901 1 008

HALAMAN PENGESAHAN

**IMPLEMENTASI METODE SCRUM DALAM
PENGEMBANGAN PERANGKAT LUNAK
DELIV**

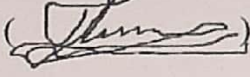

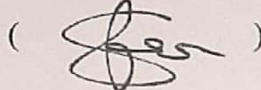
SKRIPSI

**OLEH:
NAUFALDI RAFIF SATRIYA
NIM. 14650068**

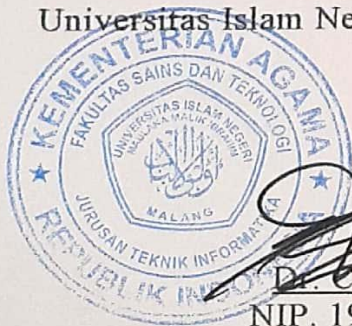
Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal Juni 2021


Susunan Dewan Penguji

Tanda Tangan

Penguji Utama	: <u>Supriyono, M. Kom</u> NIP. 198410102019031 012	()
Ketua Penguji	: <u>A'la Syauqi, M.Kom</u> NIP. 197712012008011007	()
Sekretaris Penguji	: <u>Dr. Ir. M. Amin Hariyadi, M.T</u> NIP. 19670118 200501 1 001	()
Anggota Penguji	: <u>Puspa Miladin N. S A B, M.Kom</u> NIP. 19930828 201903 2 018	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




D. Cahyo Crys dian M. CS
NIP. 19740424 200901 1 008

MOTTO

وَمَا اللَّذَّةُ إِلَّا بَعْدَ التَّعَبِ

(Tak ada kenikmatan kecuali setelah susah payah)

HALAMAN PERSEMBAHAN

الحمد لله رب العالمين

Puji syukur kehadirat Allah SWT, shalawat serta salam kepada Rasulullah SAW. Penulis persembahkan skripsi ini kepada:

Kedua orang tua penulis yang penulis cintai segenap hati, Bapak Ekbal Santoso dan Bu Tri Pujiasih serta keluarga di rumah yang selalu memberikan dukungan moral maupun fisik, doa dan bimbingan dalam penelitian skripsi ini.

Kedua adek-adek penulis yang sangat penulis cintai dan banggakan, Nadhifa M. Feponsa dan Qatrunnada Rafifa Z yang telah mendukung berupa doa dan membantu terselesaikannya penelitian skripsi ini.

Dosen pembimbing penulis Bapak Dr. M. Amin Hariyadi, M.T dan Bu Puspa Miladin, M.Kom yang telah dengan sabar membimbing penelitian skripsi ini dan selalu memberikan masukan kritikan serta saran dan semangat untuk menjalani setiap tahapan skripsi.

Seluruh dosen Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang telah membimbing dan memberikan ilmunya yang bermanfaat.

Seluruh Keluarga Teknik Informatika, terutama teman-teman Teknik Informatika angkatan 2014 yang telah memberikan semangat dan motivasi.

Komunitas IxDA Malang yang memberikan saran dan masukan terkait metode dalam pengujian dalam penulisan skripsi.

Tim Deliv Malang yang telah membantu dalam pembuatan dan proses penelitian skripsi sehingga aplikasi bisa selesai sesuai dengan penelitian yang telah dibuat.

Tim Esteh Malang dan Dipa Agency yang mendukung penyelesaian penelitian dengan memberikan dukungan dan material dalam proses pengerjakan skripsi.

Tim Batavia Hackathon Malang yang telah menemani dalam proses penelitian skripsi dan mendukung penyelesaian penelitian skripsi.

HALAMAN PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Naufaldi Rafif S

NIM : 14650068

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Implementasi Metode Scrum dalam Pengembangan
Perangkat Lunak Deliv

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, Juni 2021
Yang Membuat Pernyataan,



Naufaldi Rafif Satriya
NIM. 14650068

KATA PENGANTAR

Puji syukur bagi Allah SWT yang telah mencurahkan nikmat serta kekuatan sehingga penulis dapat menyelesaikan skripsi dengan baik. Tidak lupa juga shalawat serta salam kepada nabi Muhammad SAW yang telah membimbing umatnya ke jalan yang benar.

Dalam mengerjakan skripsi ini, tidak luput dari bantuan, bimbingan, serta dukungan berbagai pihak baik secara moril maupun materil. Oleh karenanya, izinkan penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Abdul Haris, M.Ag, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Ibu Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi.
3. Bapak Dr. Cahyo Crysdian, selaku Ketua Jurusan Teknik Informatika.
4. Bapak Dr. M. Amin Hariyadi, M.T selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing serta memberi masukan kepada penulis dalam mengerjakan skripsi ini.
5. Ibu Puspa Miladin N. S A B, M.Kom selaku dosen pembimbing II yang juga senantiasa memberikan masukan dan petunjuk tentang penulisan skripsi ini.
6. Ibu Hani Nurhayati, M.T selaku dosen wali yang selalu memberi motivasi serta nasihat selama penulis menuntut ilmu di Universitas Islam Negeri Maulana Malik Ibrahim Malang.
7. Seluruh dosen Teknik Informatika yang telah mencurahkan ilmunya kepada penulis selama kuliah.
8. Bapak, Ibu, serta keluarga besar penulis yang selalu meberikan doa serta nasihat kepada penulis selama kuliah.
9. Teman-teman Teknik Informatika 2014, khususnya Galang Luhur Pekerti, Ulung Muhammad Bestari, Bill Tanthowi Jauhari, Irvan Ariyanto, Barly Vallendito, yang telah membantu penulis dalam menyelesaikan skripsi.
10. Teman-teman IxDA Malang, yang telah membantu dalam memberikan pengetahuan dan keilmuan terkait *User Experience*
11. Teman-teman Tim Pengembang Deliv yang telah membantu proses pengembangan perangkat lunak Deliv.

Dalam Penulisan skripsi ini, masih terdapat kekurangan. Namun penulis berharap, skripsi ini dapat memberikan manfaat kepada pembaca dan kepada penulis. *Aamiin* .

Malang, 12 Juni 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
MOTTO	iv
HALAMAN PERSEMBAHAN	v
HALAMAN PERNYATAAN KEASLIAN TULISAN	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xi
ABSTRAK	xii
ABSTRACT.....	xiii
المخلص	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat.....	4
BAB 2 STUDI PUSTAKA	5
2.1 Penelitian Terkait	5
2.2 Landasan Teori	7
2.2.1 Metodologi Agile	7
2.2.2 Metode Scrum	8
2.2.3 Tahapan Scrum.....	9
2.2.4 Artefak Scrum	11
2.2.5 Tim Scrum.....	13
2.2.6 Pilar Scrum.....	14
2.2.7 MVP (Minimum Viable Product)	14
2.2.8 User Experience Questionnaire (UEQ)	15
BAB 3 DESAIN DAN IMPLEMENTASI	17
3.1 Analisa Masalah	17
3.2 Prosedur Penelitian.....	17
3.3 Pengumpulan Data	19
3.4 Metode Scrum	19
3.4.1 Inception.....	20
3.4.2 User Story	21
3.4.3 Product Backlog Item.....	28
3.5 Manajemen Komunikasi Tim	29
3.6 Identifikasi Resiko.....	30
3.7 Analisa Kebutuhan Perangkat	32
3.8 Desain Sistem	33
3.8.1 Use Case Diagram.....	33
3.8.2 Activity Diagram.....	34
3.9 Sprint Planning	50

3.10	Sprint	55
3.11	Daily Scrum.....	55
3.12	Sprint Review	55
3.13	Sprint Retropective.....	56
3.14	Pengujian	56
3.14.1	Pengalaman Pengguna	56
BAB 4 UJI COBA DAN PEMBAHASAN		58
4.1	Uji Coba	58
4.1.1	Data Uji Coba.....	58
4.2	Hasil Uji Coba	59
4.2.1	Inception.....	59
4.2.2	Sprint 1	61
4.2.3	Sprint 2.....	65
4.2.4	Sprint 3	71
4.2.5	Sprint 4.....	75
4.2.6	Sprint 5.....	78
4.2.7	Sprint 6.....	82
4.3	Pengujian	87
4.3.1	Pengujian UEQ pada Aplikasi Android Deliv	87
4.3.2	Tanggapan Responden UEQ pada Aplikasi Android Deliv.....	92
4.4	Pembahasan Hasil Uji Coba dan Pengujian	93
BAB 5 PENUTUP		97
5.1	Kesimpulan.....	97
5.2	Saran	98
DAFTAR PUSTAKA		99
LAMPIRAN.....		101

DAFTAR GAMBAR

Gambar 2.1 Kuisisioner User Experience Questionnaire (UEQ)	16
Gambar 3.1 Prosedur Penelitian.....	18
Gambar 3.2 Acara-acara di Scrum	20
Gambar 3.3 Use Case Diagram	34
Gambar 3.4 Activity diagram konsumen mendaftar di Deliv	35
Gambar 3.5 Activity diagram Konsumen Login ke Aplikasi Deliv.....	36
Gambar 3.6 Activity diagram konsumen memesan Deride	37
Gambar 3.7 Activity diagram konsumen memesan layanan Deexpress	39
Gambar 3.8 Activity diagram konsumen melihat Riwayat Order.....	40
Gambar 3.9 Activity diagram konsumen melihat Profile	41
Gambar 3.10 Activity diagram konsumen merubah Profile	42
Gambar 3.11 Activity diagram konsumen melihat Privacy Policy.....	43
Gambar 3.12 Activity diagram konsumen melihat Term of Service	44
Gambar 3.13 Activity diagram konsumen merubah bahasa	45
Gambar 3.14 Activity diagram konsumen logout dari aplikasi Deliv	46
Gambar 3.15 Activity diagram driver login ke aplikasi driver Deliv	47
Gambar 3.16 Activity diagram driver menerima order ke aplikasi driver Deliv ..	48
Gambar 3.17 Activity Diagram driver logout dari aplikasi driver Deliv	49
Gambar 4.1 Screen Android Login dan Register	63
Gambar 4.2 Screen Android Home, dan Pencarian Lokasi Tujuan Konsumen....	67
Gambar 4.3 Screen Android Lokasi Tujuan dan Pemesanan Driver Konsumen..	68
Gambar 4.4 Screen Android Home Aplikasi Driver	72
Gambar 4.5 Screen Android Home Aplikasi Driver	72
Gambar 4.6 Screen Android De Express dan Lokasi Penjemputan pada Konsumen	76
Gambar 4.7 Screen Android Memilih Lokasi Pengirim dan Identitas Barang Konsumen	80
Gambar 4.8 Screen Android Melihat Review dan Rating Driver	81
Gambar 4.9 Screen Android Menu, Logout Konsumen	84
Gambar 4.10 Screen Kebijakan Privasi dan Ketentuan Layanan	85
Gambar 4.11 Aplikasi Deliv telah diupload di Playstore.....	86
Gambar 4.12 Hasil Kuisisioner UEQ	88
Gambar 4.13 Hasil Rata-rata Penilaian Setiap Responden	89
Gambar 4.14 Diagram Benchmark Deliv.....	92

DAFTAR TABEL

Tabel 3.1 Para Stakeholder yang Terlibat dalam Scrum.....	20
Tabel 3.2 User Story Login Konsumen.....	21
Tabel 3.3 User Story Pendaftaran Konsumen.....	22
Tabel 3.4 User Story Pemesanan De Ride	22
Tabel 3.5 User Story Pemesanan De Express	23
Tabel 3.6 User Story Melihat History Detail Orders	24
Tabel 3.7 User Story Melihat Profile	24
Tabel 3.8 User Story Edit Profile Deliv	25
Tabel 3.9 User Story Melihat Term of Service Deliv	25
Tabel 3.10 User Story Melihat Privacy Policy Deliv.....	25
Tabel 3.11 User Story Merubah bahasa aplikasi.....	26
Tabel 3.12 User Story Konsumen LogOut.....	26
Tabel 3.13 User Story Driver Login	27
Tabel 3.14 User Story Menerima Order.....	27
Tabel 3.15 User Story Menerima Order.....	28
Tabel 3.16 User Story Driver Logout	28
Tabel 3.17 Product Backlog Item.....	29
Tabel 3.18 Kebutuhan Perangkat Lunak.....	32
Tabel 3.19 Minimum Viable Product	51
Tabel 3.20 Task Sprint Pertama	51
Tabel 3.21 Task Sprint Kedua.....	52
Tabel 3.22 Task Sprint Ketiga	53
Tabel 3.23 Task Sprint Keempat.....	53
Tabel 3.24 Task Sprint Kelima	54
Tabel 3.25 Task Sprint Keenam.....	54
Tabel 3.26 Pertanyaan Kuesioner kepada Konsumen.....	57

ABSTRAK

Satriya, Naufaldi Rafif. 2021. **Implementasi Metode Scrum dalam Pengembangan Perangkat Lunak Deliv**. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) Dr. M. Amin Hariyadi, M.T.,
(II) Puspa Miladin N. S A B, M.Kom.

Kata Kunci: Pengembangan Perangkat Lunak, Pengujian *User Experience Quissioner*, Scrum

Proses pengembangan perangkat lunak yang saat ini berjalan dengan menggunakan waterfall atau tidak ada metode proses manajemen pengembangan perangkat lunak sama sekali. Hal ini membuat pengerjaan sebuah perangkat lunak selalu diluar jadwal dan sering sekali aplikasi yang telah selesai memiliki banyak kekutu. Hal ini pula, membuat pengembang menjadi frustasi. Oleh karena itu diperlukan sebuah proses pengembangan perangkat lunak yang bisa membantu tim pengembang, *stakeholder* dan proyek manajemen untuk bisa berkomunikasi dan berkolaborasi agar menghasilkan aplikasi yang lebih efisien dan efektif. Maka diperlukan dengan menggunakan metode Scrum, sebuah metode pengembangan perangkat lunak yang bersifat *agile*. Pada metode pengembangan perangkat lunak dengan Scrum, dibuatlah sebuah *Minimum Viable Product* sebagai produk awal dan proses pengerjaan 6 sprint atau sekitar 3 bulan. Hasil penelitian, akan diuji menggunakan *User Experience Quissioner* dengan menggunakan responden yang ada dengan membandingkan melalui standar UEQ didapatkan hasil setiap grup penilaian UEQ mendapatkan nilai rata-rata 1,145 untuk grup Stimulasi, nilai rata-rata 1,185 untuk grup Ketepatan, nilai rata-rata 1,207 untuk Efisiensi, nilai rata-rata 1,630 untuk grup Kejelasan, nilai rata-rata 1,145 untuk grup Daya Tarik dan nilai rata-rata 0,217 untuk Kebaruan dengan hasil *Above Average* sebagai bukti bahwa menggunakan metode Scrum dapat meningkatkan efisiensi dan efektifitas dalam proses pengembangan perangkat lunak

ABSTRACT

Satriya, Naufaldi Rafif. 2021.. **Implementation Scrum Method for Developing Deliv Application**. Department of Informatics Engineering, Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University of Malang.

Supervisor: (I) Dr. M. Amin Hariyadi, M.T.,
(II) Puspa Miladin N. S A B, M.Kom

Keywords: Developing Application, Scrum, User Experience Quisioner

Software development process is currently running using the Waterfall, or there is no software development management process method at all. This makes the work of software always off schedule and often the completed application has many bugs. Its makes developers frustrated. Therefore, we need a software development process that can help the development Team, Stakeholders, and Project Management to be able to communicate and collaborate for more efficient and effective applications. It is necessary to use the Scrum method, an agile software development method. In the software development method with Scrum, a Minimum Viable Product is made as to the initial product and the process takes 6 sprints or about 3 months. The results of the study will be tested using a User Experience Questionnaire using existing respondents by comparing through the UEQ standard, the results of each UEQ assessment group get an average value of 1.145 for the Stimulation group, an average value of 1.185 for the Accuracy group, an average value of 1.207 for the Stimulation group. Efficiency, the average value is 1.630 for the Clarity group, the average value is 1.145 for the Attractiveness group and the average value is 0.217 for Novelty with Above Average results as evidence that using the Scrum method can increase efficiency and effectiveness in the software development process.

الملخص

ساتريا ونوفالدي رفيف .. 2021. تطبيق طريقة سكرم لتطوير تطبيق Deliv. قسم هندسة المعلوماتية ، كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانج. المشرف (I) :محمد امين هاربيادي الماجستير . (II) بوسبا ميلاندين ، م.كوم.

الكلمات المفتاحية :سكرم ، تطبيق ، تجربة المستخدم

عملية تطوير البرامج التي يتم تشغيلها حاليًا باستخدام طريقة عملية إدارة تطوير البرامج أو عدم وجودها على الإطلاق. هذا يجعل عمل البرنامج دائمًا خارج الجدول الزمني وغالبًا ما يحتوي التطبيق المكتمل على العديد من الأخطاء. هذا ، أيضًا ، يجعل المطورين محبطين. لذلك نحن بحاجة إلى عملية تطوير برمجيات يمكن أن تساعد فريق التطوير وأصحاب المصلحة وإدارة المشروع ليكونوا قادرين على التواصل والتعاون لإنجاز تطبيقات أكثر كفاءة وفعالية. من الضروري استخدام طريقة سكروم ، وهي طريقة تطوير برمجيات رشيقة. في طريقة تطوير البرامج مع Scrum ، يتم عمل منتج قابل للتطبيق على الأقل بالنسبة للمنتج الأولي وتستغرق العملية 6 أسابيع أو حوالي 3 أشهر. سيتم اختبار نتائج الدراسة باستخدام استبيان تجربة المستخدم باستخدام المستجيبين الحاليين من خلال المقارنة من خلال معيار UEQ ، تحصل نتائج كل مجموعة تقييم UEQ على متوسط قيمة 1.145 لمجموعة التحفيز ، بمتوسط قيمة 1.185 لمجموعة الدقة ، بمتوسط قيمة 1.207 لمجموعة التحفيز. الكفاءة ، متوسط القيمة 1.630 لمجموعة الوضوح ، ومتوسط القيمة 1.145 لمجموعة الجاذبية ومتوسط القيمة 0.217 للنتائج المبتكرة مع أعلى من المتوسط كدليل على أن استخدام طريقة سكروم يمكن أن يزيد الكفاءة والفعالية في عملية تطوير البرمجيات

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Kemacetan dan transportasi umum yang kurang dapat diandalkan adalah masalah yang selalu ditemui di sejumlah kota besar di Indonesia. Berdasarkan riset terbaru Inrix, tingkat kemacetan di dunia mengalami peningkatan, termasuk kota-kota besar di Indonesia (Cookson & Pishue, 2017). Inrix mengumpulkan data dari 1.360 kota di 38 negara mencakup lebih dari 250.000 kilometer persegi jalan dan berfokus pada kemacetan di sepanjang hari dan minggu. Traffic Scorecard 2017 mengadopsi metodologi yang sama dengan 2016 dengan menyediakan wawasan tentang skala dan dampak kemacetan pada waktu yang berbeda. Hasilnya, Indonesia mendapatkan peringkat kedua dalam negara yang memiliki tingkat kemacetan cukup tinggi dengan menghabiskan waktu sia-sia di jalan sampai 47 jam dalam satu tahun.

Penyebabnya salah satunya adalah meningkatnya jumlah kendaraan bermotor di Indonesia. Pada tahun 2015 mencapai 121,39 juta unit. Sedangkan pada tahun 2016 menurut Badan Pusat Statistik meningkat menjadi 129,281 juta unit. Mengalami peningkatan sebesar 8 juta kendaraan. Peningkatan ini didominasi oleh kendaraan sepeda motor sebanyak 105,150 juta unit kendaraan sepeda motor.

Meningkatnya jumlah kendaraan sepeda motor dan meningkatnya tingkat kemacetan di Indonesia membuat munculnya jasa transportasi online selain angkutan umum seperti ojek online akibat transportasi umum yang tidak bisa

memenuhi kebutuhan masyarakat dan kualitas transportasi umum yang menurun. Sehingga memunculkan ojek online pada tahun 2015 dan hadirnya GO-JEK dengan layanan ojek online berbasis Android dan iOS. Selang beberapa bulan kemudian muncul pula Grabbike dan layanan ojek online lainnya yang dengan cepat menjamur.

Tak terkecuali dengan Deliv yang hadir untuk meramaikan persaingan ojek online / aplikasi *on-demand* dengan menghadirkan layanan pemesanan ojek online dengan menghadirkan Android apps dan berpusat di wilayah Bojonegoro. Deliv merupakan aplikasi yang memungkinkan user untuk memesan ojek secara online dengan mengkalkulasikan biaya perjalanan dan pengemudi akan menjemput penumpang dimanapun.

Sebagai salah satu startup yang baru muncul dan menghadirkan layanannya di Bojonegoro , dimana sudah menjadi lokasi munculnya GrabBike dan GO-JEK yang terlebih dahulu meluncurkan layanan ojek online di Bojonegoro, mengharuskan Deliv untuk mempercepat pengembangan perangkat lunak dan meningkatkan sistem Deliv secara teratur.

Hasilnya, pada Oktober 2018 Deliv meluncurkan layanannya pertama kalinya di Bojonegoro dengan menghadirkan De Ride, De Food dan De Express. Ketiga layanan ini langsung diluncurkan dengan menggunakan prinsip MVP (Minimum Viable Product).

Seiring berjalannya waktu, dengan meningkatnya jumlah driver dan pengguna, maka diperlukannya pengembangan perangkat lunak untuk mempercepat dan memperlancar perangkat lunak Deliv agar mendapatkan pengalaman yang baik dari pengguna.

Maka dari itu penulis mengajukan skripsi yang berjudul “Implementasi Scrum dalam Pengembangan Perangkat Lunak Deliv”. Alasan penulis menggunakan Scrum karena sesuai dengan kondisi perangkat lunak saat ini, dimana perlunya banyak sekali perubahan yang akan terjadi dan perlunya interaksi antara Tim Pengembangan dengan *Product Owner* untuk mendapatkan sebuah keselarasan antara tujuan bisnis yang diinginkan *Product Owner* dengan tim pengembang dalam mengembangkan perangkat lunak Deliv.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan pada sub bagian sebelumnya, identifikasi masalah yang dapat dirumuskan sebagai berikut :

1. Seberapa efisien penerapan scrum dalam pengembangan aplikasi Deliv ?
2. Apakah menggunakan penerapan scrum dalam pengembangan aplikasi Deliv mampu mencapai target sesuai dengan kesepakatan ?
3. Bagaimana hasil aplikasi Android Deliv yang telah dibuat dengan metode scrum dengan menggunakan *User Experience Questionnaire* (UEQ)?

1.3 Batasan Masalah

Pada penelitian ini terdapat batasan permasalahan yaitu data yang dikelola dari data Deliv.

1. Pada penelitian ini terdapat batasan permasalahan yaitu data yang dikelola adalah data yang dimiliki oleh Deliv.
2. Pembuatan aplikasi Deliv hanya sampai pada tahap MVP (Minimum Viable Product).

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah dan batasan masalah maka tujuan penelitian yang ingin dicapai dalam penelitian ini adalah

1. Untuk mengetahui seberapa efisien penggunaan scrum dalam pengembangan aplikasi Deliv
2. Untuk mengetahui seberapa efektif penggunaan scrum dalam pengembangan perangkat lunak Deliv

1.5 Manfaat

Berdasarkan tujuan penelitian, didapatkan beberapa manfaat yang diperoleh melalui penelitian ini sebagai berikut :

1. Manfaat dari sisi Perusahaan

Mempercepat proses pengembangan perangkat lunak Deliv sehingga waktu, *cost* dan kualitas dari perangkat lunak yang dikembangkan sesuai dengan visi dan misi perusahaan.

2. Manfaat dari sisi Aplikasi

Untuk memudahkan Tim dalam berkoordinasi dengan *Product Owner* sehingga bisa memberikan perangkat lunak yang sesuai dengan apa yang diinginkan oleh *Product Owner*.

BAB 2

STUDI PUSTAKA

2.1 Penelitian Terkait

Pada bab ini akan dibahas mengenai beberapa penelitian terdahulu yang mendukung penelitian menggunakan metode Scrum, antara lain sebagai berikut :

Adi (2015) mengimplementasikan Scrum di proyek keuangan Arium Core yang melibatkan 6 orang personil. Penelitian ini untuk mengetahui sejauh mana dampak yang didapatkan dengan mengimplementasikan Scrum ke dalam pengembangan perangkat lunak. Penelitian ini menunjukkan bahwa dengan menggunakan Scrum akan didapatkan kualitas dan resiko proyek akan terlihat lebih cepat dari sebelumnya. Kecepatan dalam mengetahui *progress* serta dengan menggunakan Scrum akan didapatkan *feedback* dari *Product Owner* dan juga pelanggan diakhir *Sprint*.

Azdy & SN, (2012) mengatasi kekurangan yang mungkin muncul dalam pengembangan secara terdistribusi serta dampak yang bisa dihasilkan jika menggunakan metode Scrum sebagai salah satu varian dari metode *agile*. Menggunakan Scrum untuk pengembangan *software* terdistribusi yaitu dimana pengembang *software* dilakukan oleh beberapa kelompok pengembang yang terdapat di lokasi yang berbeda. Dalam penelitian tersebut, peneliti menggunakan metode Scrum karena Scrum merupakan kerangka kerja dimana pihak-pihak dapat mencari jalan keluar dari permasalahan yang kompleks dan pada saat bersamaan membuat produk bernilai tinggi. Dari penelitian yang dilakukan tersebut, diperoleh

hasil bahwa Scrum mampu menjawab permasalahan ketika pengembangan *software* secara terdistribusi dengan meningkatkan kecepatan waktu pengembangan *software*, memungkinkan adanya transfer ilmu pengetahuan dengan antar anggota tim, dan menciptakan komunikasi yang stabil dan konsisten bagi seluruh pihak yang terkait.

Utomo & Prayitno (2015) mengembangkan sistem informasi manajemen proyek menggunakan metode Scrum di perusahaan tersebut Hasilnya dengan menggunakan JIRA dan metode Scrum dalam memonitoring semua pengerjaan perangkat lunak yang sedang berjalan sehingga manajemen perusahaan dapat berjalan lebih efisien dan cepat dengan studi kasus di PT. FLASHiZ Indonesia. Menggunakan JIRA dan Scrum semua proses dalam pengerjaan perangkat lunak bisa terdata dan terdokumentasi dengan baik dan perusahaan bisa mengawasi jalannya proses pengerjaan perangkat lunak yang sedang berlangsung mulai dari tahap negosiasi, pengerjaan perangkat lunak oleh para pengembang, mengawasi kinerja *Quality Assurance*, mendata masalah yang ada hingga tahap selesainya suatu proyek.

Andry dkk, (2019) menggunakan Scrum dalam pembuatan sebuah perangkat lunak untuk rapat secara daring. Penelitian ini menunjukkan perangkat lunak yang dibuat dapat diselesaikan dengan cepat karena semua kebutuhan dalam pembuatan perangkat lunak untuk rapat secara daring dapat terdefinisikan dengan baik mulai dari keinginan konsumen, kesalahan dalam perangkat lunak dan pembaruan perangkat lunak. Scrum juga memberikan *meeting pattern* sehingga memberikan dukungan terhadap perangkat lunak untuk rapat secara daring yang akan dibangun.

2.2 Landasan Teori

2.2.1 Metodologi Agile

Metodologi Agile adalah salah satu metode pengembangan perangkat lunak. Metodologi ini sangat cocok untuk organisasi yang menghasilkan produk diwaktu yang singkat dan kompleks. Agile sendiri memiliki pengertian cepat, ringan, bebas bergerak dan waspada dikembangkan oleh Kent Beck dan 16 teman kerjanya dengan memulai Metode Pengembangan Perangkat Lunak Agile(Schwaber & Sutherland, 2017) . Dalam metodologi Agile terdapat tiga poin penting, yaitu mengidentifikasi masalah dan/atau peluang, bergerak menindaklanjutinya dan melakukan keduanya secara *iterative* tanpa henti dan dengan periode yang singkat.

Metodologi ini memiliki pedoman yang dinamakan Agile Manifesto yang terdiri dari beberapa poin yaitu :

- a) Individu dan interaksi lebih dari proses dan sarana perangkat lunak
- b) Perangkat lunak yang bekerja lebih dari dokumentasi yang menyeluruh
- c) Kolaborasi dengan klien lebih dari negosiasi kontrak
- d) Tanggap terhadap perubahan lebih dari mengikuti rencana

Terlihat dari manifesto di atas, bisa terlihat bahwa Agile berfokus kepada interaksi semua pihak yang terlibat dari pada proses dan alat yang digunakan dalam pengembangan perangkat lunak itu sendiri. Pengembangan perangkat lunak lebih penting daripada dokumentasi yang ada. Kolaborasi tim pengembang tidak hanya sekedar berasal dari negosiasi kontrak antara tim pengembang dan juga klien, melainkan lebih dari itu. Sehingga interaksi tidak hanya sebatas negosiasi kontrak.

Agile memiliki kepekaan tinggi terhadap perubahan. Sehingga dalam metodologi Agile merespon semua perubahan yang ada daripada mengikuti rencana yang telah ada. (Kniberg & Bima, 2007) Metodologi Agile memiliki beberapa model yang dikenal cukup luas yaitu :

1. Agile Modelling
2. Extreme Programming (XP)
3. Scrum

Perbedaan pada setiap metode terletak pada praktek atau penerapan yang dilakukan dalam pengembangan perangkat lunak.

2.2.2 Metode Scrum

Metode Kerangka Kerja Scrum adalah jenis baru dari metodologi Agile yang diharapkan dapat meningkatkan kecepatan, fleksibilitas dalam pengembangan perangkat lunak. (Schwaber & Sutherland, 2017) Scrum merupakan metode Agile yang cukup populer dimana produktifitas menjadi lebih tinggi. Prinsip kerja *Scrum* adalah bekerja secara iteratif dan bertahap hingga mencapai waktu yang telah ditentukan sehingga produk perangkat lunak yang dikembangkan dapat memenuhi kebutuhan yang diinginkan oleh konsumen. Metode Scrum dalam pengembangan perangkat lunak terbagi menjadi beberapa bagian *iteration* dimana setiap *iteration* disebut dengan *Sprint*.

Metode Scrum dimulai dengan mengumpulkan semua kebutuhan pengguna , namun tidak menutup kemungkinan bahwa semua kebutuhan harus berasal dari pengguna disaat awal-awal *Sprint*. Pengguna dapat mengubah kebutuhan kapan pun ketika sedang dalam pengembangan. Mereka bisa menambahkan fitur baru, menghapus fitur baru , atau memperbarui fitur baru.

2.2.3 Tahapan Scrum

Tahapan Scrum dibuat untuk menciptakan kesinambungan dan mengurangi tahapan lain yang tidak tercantum di dalam Scrum. Tidak adanya pelaksanaan salah satu tahapan ini akan mengurangi transparansi dan menghilangkan kesempatan untuk meninjau dan membuat perubahan.

1. *Sprint*

Sebuah fase pengembangan perangkat lunak yang memiliki batasan waktu maksimal satu bulan dengan durasi yang konsisten sepanjang proses pengembangan produk. *Sprint* baru, langsung dimulai ketika *Sprint* sebelumnya selesai. *Sprint* memuat dan terdiri dari *Sprint Planning*, *Daily Scrum*, *Sprint Review* dan *Sprint Retrospective* (Dingsøyr dkk, 2012)

2. *Sprint Planning*

Pekerjaan yang dilakukan di dalam *Sprint* direncanakan pada *Sprint Planning*. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota Tim Scrum. *Sprint Planning* dibatasi maksimum delapan jam untuk *Sprint* yang berdurasi satu bulan.

3. *Daily Scrum*

Daily Scrum adalah kegiatan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam kedepan. Hal ini dilakukan dengan meninjau pekerjaan semenjak tahapan *Daily Scrum* terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan *Daily Scrum* berikutnya.

4. *Sprint Review*

Sprint Review diadakan di setiap akhir *Sprint* untuk meninjau Inkremen dan merubah *Product Backlog* bila diperlukan. Pada saat *Sprint Review*, Tim Scrum dan stakeholder berkolaborasi untuk membahas apa yang telah dikerjakan dalam *Sprint* yang baru usai. Berdasarkan hasil tersebut dan semua perubahan *Product Backlog* pada saat *Sprint*, para hadirin berkolaborasi menentukan apa yang dapat dikerjakan di *Sprint* berikutnya, untuk mengoptimalisasi nilai produk. Pertemuan ini bersifat informal, bukan merupakan *status meeting*, dan presentasi dari Inkremen diharapkan dapat mengumpulkan masukan dan menumbuhkan semangat kolaborasi.

5. *Sprint Retrospective*

Sprint Retrospective adalah sebuah kesempatan bagi Tim Scrum untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya. *Sprint Retrospective* dilangsungkan setelah *Sprint Review* selesai dan sebelum *Sprint Planning* berikutnya. Ini adalah tahapan dengan batasan waktu

maksimum selama tiga jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan bahwa tahapan ini dilaksanakan dan setiap hadirin memahami tujuannya. *Scrum Master* mengedukasi Tim Scrum untuk melaksanakannya dalam batasan waktu yang telah ditentukan. *Scrum Master* berpartisipasi sebagai rekan yang bertanggungjawab terhadap setiap proses yang terjadi ketika Scrum berlangsung.

2.2.4 Artefak Scrum

Artefak Scrum merepresentasikan pekerjaan atau nilai, bertujuan untuk memberikan ruang transparansi, dan kesempatan-kesempatan untuk meninjau ulang dan adaptasi akan perubahan yang telah akan terjadi. (Kenett dkk, 2016) Artefak yang didefinisikan oleh Scrum secara khusus dirancang untuk meningkatkan transparansi dari setiap informasi yang ada antara tim pengembang, *product owner* dan Scrum master, dengan begitu semua pihak dapat memiliki pemahaman yang sama terhadap artefak.

1. *Product Backlog*

Product Backlog adalah daftar terurut semua fitur yang telah diketahui saat ini yang harus ada di dalam produk. *Product Backlog* adalah satu-satunya daftar sumber kebutuhan untuk setiap perubahan yang terjadi didalam produk. Salah satu cara mendapatkan *Product Backlog* berasal dari *User Story* yang telah dibuat sebelumnya. *Product Owner* bertanggung-jawab terhadap *Product Backlog*, termasuk isinya, ketersediaannya, dan urutannya. *Product Backlog* sendiri merupakan

sentral dari Scrum.

2. *Sprint Backlog*

Sprint Backlog adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan di *Sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Inkremen selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang “Selesai”.

3. Inkremen

Inkremen adalah gabungan dari semua item *Product Backlog* yang diselesaikan pada *Sprint* berjalan dan nilai-nilai dari Inkremen *Sprint-Sprint* sebelumnya. Pada akhir *Sprint*, inkremen terbaru harus “Selesai”, yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi “Selesai” yang dibuat oleh Tim Scrum. Terlepas apakah *Product Owner* akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh

4. Definisi Selesai

Ketika sebuah item *Product Backlog* atau Inkremen dikatakan “Selesai”, setiap pihak harus mengerti dengan apa yang dimaksud dengan “Selesai”. Walaupun definisi ini berbeda-beda antar tim Scrum, sesama anggota tim harus memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Ini adalah definisi selesai untuk Tim Scrum dan ini

digunakan untuk memeriksa apakah pekerjaan untuk mengembangkan Inkremen dianggap selesai.

2.2.5 Tim Scrum

Tim Scrum tidak mencakup peran tradisional seperti programmer, desainer, tester atau analisa sistem. Tetapi para *stackholder* yang bekerja sama dalam proyek untuk menyelesaikan pekerjaan mereka secara kolektif dan berkomitmen untuk menyelesaikannya. Terdiri atas *Product Owner*, Tim Pengembang dan *Scrum Master*.

1. *Scrum Master*

Fasilitator yang melayani tim tetapi bukanlah project manager. Memastikan fase metodologi Scrum dilakukan dengan baik, melindungi tim dan memastikan mereka tidak terlalu berkomitmen dengan yang dikerjakan karena perubahan dapat terjadi disetiap *Sprint*.

2. *Product Owner*

Pihak yang bertanggung jawab terhadap suksesnya pengembangan produk dan hasil kerja tim pengembang, biasanya adalah representasi dari konsumen. *Product Owner* satu-satunya orang yang bertanggung jawab mengelola *Product Backlog*.

3. Tim Pengembang

Sekelompok orang yang bertanggung jawab untuk menghasilkan product perangkat lunak. Dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang ada di dalam tim akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan.

2.2.6 Pilar Scrum

1. Transparansi

Semua variabel yang perlu diketahui dibuat transparansi agar semua orang yang peduli dengan pengembangan perangkat lunak dapat melihatnya. Transparansi mengharuskan aspek-aspek tersebut didefinisikan dengan standar yang sama, sehingga semua orang yang terlibat memiliki pemahaman yang sama mengenai apa yang sedang ditinjau.

2. Inspeksi

Semua variabel yang dibuat transparan tersebut ditinjau setiap waktunya, peninjauan seandainya tidak dilakukan terlampau sering karena dapat menyebabkan terhambatnya pekerjaan. Paling bermanfaat jika dilakukan secara rutin oleh peninjau yang kompeten pada saat pekerjaan berjalan.

3. Adaptasi

Bila variabel yang telah ditinjau ternyata akan menyebabkan hasil akhir yang tidak sesuai dengan harapan, maka perencanaan awal harus diubah.

2.2.7 MVP (Minimum Viable Product)

MVP adalah teknik pengembangan dimana produk atau situs web baru dikembangkan dengan fitur yang cukup untuk memuaskan pengguna awal. Fitur produk yang lengkap hanya dirancang dan dikembangkan setelah mempertimbangkan umpan balik dari pengguna mengenai MVP

yang telah diberikan (Lenarduzzi & Taibi, 2016). Konsep ini telah dipopulerkan oleh Eric Ries, seorang konsultan dan penulis tentang startup.

MVP yang layak hanya memiliki fitur-fitur inti yang cukup untuk memberikan produk kepada konsumen dan tidak lebih. Pengembang menerapkan produk ke pelanggan / konsumen untuk mendapatkan umpan balik dan mampu memahami visi produk dari purwarupa atau informasi pemasaran. Strategi ini untuk menghindari pembuatan produk yang sesuai dengan yang diinginkan oleh pelanggan dan berusaha memaksimalkan informasi yang didapatkan oleh pelanggan.

2.2.8 User Experience Questionnaire (UEQ)

Metode *User Experience Questionnaire* adalah sebuah metode untuk menilai pengalaman pengguna / *User Experience* dari sebuah produk.(Schrepp dkk, 2017) Metode ini mampu melakukan pengujian dengan cepat dan tepat pengalaman pengguna pada sebuah produk dengan memberikan kebebasan kepada pengguna untuk memberitahukan pengalaman, perilaku dan perasaan saat menggunakan produk (Schrepp dkk, 2017). Evaluasi dengan menggunakan *User Experience Questionnaire* (UEQ) yang terdiri dari 26 item akan memiliki 2 buah penilai kata yang saling bertolak belakang dengan 7 poin skala likert dengan 3 poin bernilai positif, 1 poin bernilai netral dan 3 poin bernilai negatif. Isi dari kuisisioner

dari *User Experience Questionnaire* (UEQ) terdapat pada Gambar 2.1 :

	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

Gambar 2.1 Kuisiomer *User Experience Questionnaire* (UEQ)

BAB 3

DESAIN DAN IMPLEMENTASI

3.1 Analisa Masalah

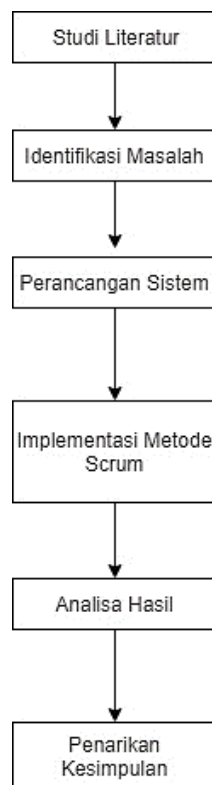
Sesuai dengan hasil penelitian , terlihat pada proses pembuatan perangkat lunak Deliv mengalami kemunduran dari jadwal yang sebelumnya sudah dibuat antara Deliv dengan para pengembang Deliv. Manajemen yang saat ini digunakan dalam pengembangan Deliv masih bersifat *waterfall* . Penyebab kemunduran proses pembuatan perangkat lunak ini disebabkan karena beberapa hal diantaranya tim pengembang dan *produk owner* hanya mengadakan pertemuan ketika satu screen situs web selesai dikerjakan atau ketika *produk owner* menginginkan untuk bertemu dengan tim pengembang sehingga tidak ada jadwal yang pasti pertemuan antara tim pengembang dan klien membahas jalannya pengerjaan perangkat lunak tersebut. Masalah lain juga karena keterlambatan tim pengembang dalam menyelesaikan *deadline* yang diberikan *client* kepada tim pengembang, adanya *miss* komunikasi antara tim pengembang, project manager dan klien.

3.2 Prosedur Penelitian

Metodologi penelitian adalah cara yang digunakan dalam memperoleh suatu data menjadi informasi yang akurat sesuai dengan permasalahan yang akan diteliti. Metodologi juga merupakan analisis teoritis mengenai suatu cara atau metode. Metodologi penelitian digunakan sebagai pedoman dalam pelaksanaan penelitian

agar hasil penelitian tidak menyimpang dari tujuan yang telah dilakukan sebelumnya.

Berdasarkan gambar 3.1, terdapat beberapa tahapan dalam penelitian yang peneliti lakukan untuk menguji metode yang digunakan. Penelitian dimulai dengan melakukan studi literature mengenai metode scrum. Tahap selanjutnya, yaitu mengidentifikasi masalah dengan menentukan pertanyaan-pertanyaan penelitian dalam penelitian ini, serta dilanjutkan dengan perancangan sistem yang akan dibangun. Selanjutnya adalah implementasi metode scrum didalam sistem yang akan dibuat. Tahap selanjutnya adalah melakukan analisa dari hasil penerapan metode scrum di dalam sistem pengembangan perangkat lunak Deliv. Tahap terakhir adalah dengan penarikan kesimpulan dari hasil yang telah didapatkan.



Gambar 3.1 Prosedur Penelitian

3.3 Pengumpulan Data

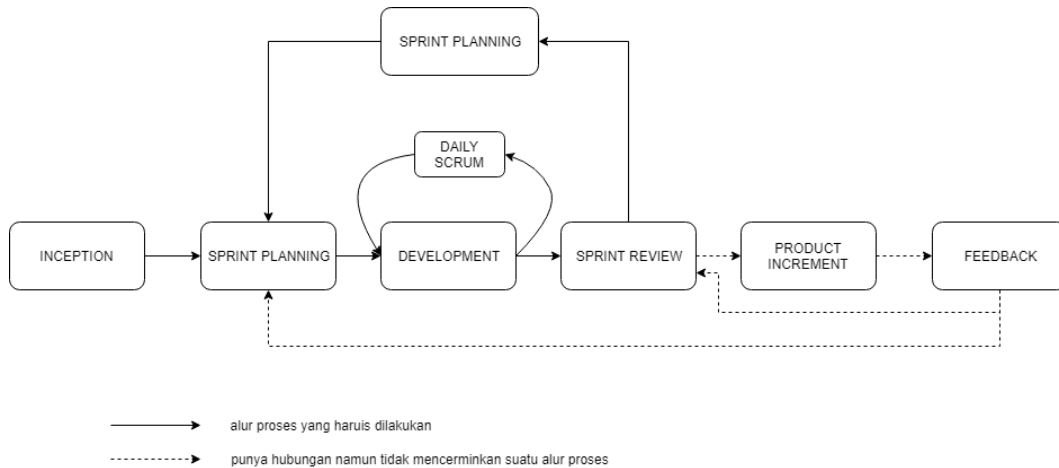
Data yang dipergunakan dalam penelitian ini diperoleh secara langsung dari Deliv Malang dan tim pengembang yakni dengan melakukan diskusi dan wawancara langsung. Data yang diperoleh antara lain proses bisnis Deliv, pengembangan perangkat lunak deliv, dan data konsumen Deliv.

3.4 Metode Scrum

Dalam penelitian pembangunan perangkat lunak Deliv dengan menggunakan metode scrum . Metode *scrum* adalah sebuah kerangka kerja untuk pengembangan dan mengelola produk kompleks. Dalam pengembangan sebuah perangkat lunak dengan menggunakan scrum yang termasuk dalam metode agile, yang memiliki *agile manifesto* yaitu :

- a) Individu dan interaksi lebih dari proses dan sarana perangkat lunak
- b) Perangkat lunak yang bekerja lebih dari dokumentasi yang menyeluruh
- c) Kolaborasi dengan klien lebih dari negosiasi kontrak
- d) Tanggap terhadap perubahan lebih dari mengikuti rencana

Terdapat beberapa tahapan-tahapan di Scrum dalam pengembangan perangkat lunak tahapan-tahapan tersebut dapat dilihat pada Gambar 3.2 :



Gambar 3.2 Tahapan di Scrum

Aturan main dari Scrum mengikat cara, peran dan artefak serta menggambarkan hubungan dan interaksi antara satu komponen dengan yang lainnya.

3.4.1 Inception

Tahapan yang dilaksanakan sebelum *Sprint* yang diikuti oleh seluruh anggota tim scrum untuk mendefinisikan mengenai semua kebutuhan ketika diadakannya scrum (Rola & Kuchta, 2015). Diskusi dari para anggota tim scrum pun sebagai berikut :

Tabel 3.1 Para *Stakeholder* yang Terlibat dalam Scrum

No	Nama Anggota	Peran	Kategori
1	Heli Suharjono	<i>Product Owner</i>	
2	Rofiqo Fauzan Firdaus	Tim Pengembang	Android Dev
3	Adwi Arifin	Tim Pengembang	Back End Dev
4	Reksa Rangga W	Tim Pengembang	Back End Dev
5.	Rahmat Arief Hidayat	Tim Pengembang	Desainer
6.	Naufaldi Rafif S	Scrum Master	

3.4.2 User Story

Langkah pertama di dalam inception adalah dengan Membuat *User Story* dari Deliv. Beberapa kolom didalam *User Story* sebagai berikut :

- a. *Actor* adalah pelaku yang melakukan *Story* tersebut
- b. *Objective* adalah tujuan dari *Story* tersebut dibuat
- c. *Pre-Condition* adalah keadaan sebelum *Story* tersebut akan dilaksanakan
- d. *Main Flow* adalah tahapan-tahapan didalam *Story* yang akan dilakukan oleh *Actor* untuk mencapai *Objective*.
- e. *Alternative Flow* adalah jika *Actor* tidak memenuhi *Pre-Condition* maka akan mengikuti alur tersebut
- f. *Post Condition* adalah kondisi setelah *Objective* selesai dilakukan.

1. *Login* Konsumen

Tabel 3.2 *User Story Login* Konsumen

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen dapat melakukan <i>login</i>
3	<i>Pre-Condition</i>	Konsumen sudah registrasi sebelumnya
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih login dengan menggunakan nomor handphone / email 2. Konsumen melakukan auth nomor hp / email 3. Konsumen dialihkan ke screen awal aplikasi android Deliv
5	<i>Alternatives Flow</i>	Konsumen yang belum daftar maka akan diarahkan ke screen pendaftaran
6	<i>Post Condition</i>	Konsumen masuk ke screen awal dengan kondisi sudah <i>login</i>

2. Konsumen – Pendaftarann

Tabel 3.3 *User Story Pendaftaran Konsumen*

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen dapat melakukan pendaftaran
3	<i>Pre-Condition</i>	Konsumen belum pernah mendaftar sebelumnya
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu Register pada screen login 2. Konsumen memilih register dengan menggunakan nomor handphone / email 3. Konsumen dialihkan ke antarmuka awal Deliv
5	<i>Alternatives Flow</i>	Konsumen yang sudah pernah daftar maka akan diarahkan ke screen login
6	<i>Post Condition</i>	Konsumen masuk ke antarmuka home di aplikasi android Deliv

3. Konsumen – Pemesanan De Ride

Tabel 3.4 *User Story Pemesanan De Ride*

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen dapat melakukan pemesanan De Ride
3	<i>Pre-Condition</i>	Konsumen sudah mendaftar di Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih lokasi yang akan dituju 2. Konsumen menkonfirmasi lokasi yang akan dituju 3. Konsumen akan memilih menggunakan sepeda motor 4. Konsumen memeriksa lokasi penjemputan dan lokasi tujuan 5. Konsumen mendapatkan pilihan untuk pembayaran menggunakan Cash
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Muncul notifikasi bahwa pengguna mendapatkan <i>driver</i>

4. Konsumen – Pemesanan De Express

Tabel 3.5 *User Story* Pemesanan De Express

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen dapat melakukan pemesanan De Express
3	<i>Pre-Condition</i>	Konsumen sudah mendaftar di Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih lokasi pengambilan barang 2. Konsumen memilih lokasi pengiriman barang 3. Konsumen memilih jenis kendaraan pengambilan 4. Konsumen mengkonfirmasi biaya dan lokasi pengiriman barang 5. Konsumen menuliskan detail pesanan barang 6. Konsumen melakukan orderan
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Muncul notifikasi bahwa pengguna mendapatkan <i>driver</i>

5. Konsumen – Melihat History Orders

Tabel 3.6 *User Story* Melihat History Detail Orders

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen melihat history detail orders
3	<i>Pre-Condition</i>	Konsumen mengakses Home
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu History pada menubar 2. Konsumen diarahkan ke screen History 3. Konsumen mendapatkan informasi order yang sedang aktif. 4. Konsumen bisa berpindah menu dengan mengarahkan ke Orders History 5. Konsumen diarahkan pada tampilan Orders History
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Konsumen dapat mengetahui dan melihat Orders History selama menggunakan aplikasi Deliv

6. Konsumen – Melihat Profile

Tabel 3.7 *User Story* Melihat Profile

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen melihat akun profile Deliv
3	<i>Pre-Condition</i>	Konsumen mengakses Home
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu Myaccount 2. Konsumen diarahkan ke screen Myaccount dan dapat melihat profile akun Deliv beserta menu untuk merubah bahasa, melihat Term of Service, Privacy Policy dan Bantuan
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Konsumen dapat melihat seluruh informasi mengenai profilnya di Myaccount

7. Konsumen – Edit Profile Deliv

Tabel 3.8 *User Story* Edit Profile Deliv

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen dapat merubah profile pada Deliv
3	<i>Pre-Condition</i>	Konsumen berada pada menu Myaccount
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu Account 2. Memilih icon menu edit profile 3. Merubah profile 4. Menekan tombol save
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Profile berubah sesuai dengan data yang telah disimpan

8. Konsumen – Melihat Term of Service Deliv

Tabel 3.9 *User Story* Melihat Term of Service Deliv

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen melihat screen Term of Service
3	<i>Pre-Condition</i>	Konsumen mengakses home Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu <i>Account</i> pada navigasi 2. Konsumen memilih menu Term of Service 3. Konsumen diarahkan pada screen Term of Service
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Konsumen dapat melihat screen Term of Service

9. Konsumen – Melihat Privacy Policy Deliv

Tabel 3.10 *User Story* Melihat *Privacy Policy* Deliv

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen melihat screen <i>Privacy Policy</i>
3	<i>Pre-Condition</i>	Konsumen mengakses home Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 4. Konsumen memilih menu <i>Account</i> pada navigasi 5. Konsumen memilih menu <i>Privacy Policy</i> 6. Konsumen diarahkan pada <i>Privacy Policy</i>

No.	Alur	Keterangan
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Konsumen Konsumen dapat melihat screen Privacy Policy

10. Konsumen – Merubah bahasa aplikasi

Tabel 3.11 *User Story* Merubah bahasa aplikasi

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen merubah bahasa aplikasi
3	<i>Pre-Condition</i>	Konsumen mengakses home Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu <i>Account</i> 2. Konsumen memilih menu Change Language 3. Muncul daftar bahasa yang tersedia 4. Konsumen memilih bahasa
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Perubahan bahasa pada aplikasi Deliv

11. Konsumen – *LogOut*

Tabel 3.12 *User Story* Konsumen *LogOut*

No.	Alur	Keterangan
1	<i>Actor</i>	Konsumen
2	<i>Objective</i>	Konsumen memilih fitur <i>logout</i> untuk keluar dari akun Deliv
3	<i>Pre-Condition</i>	Konsumen sudah <i>login</i> sebelumnya
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konsumen memilih menu <i>Account</i> 2. Konsumen memilih menu <i>Log out</i> 3. Muncul pop up untuk konfirmasi 4. Konsumen memilih tombol “yes” 5. Konsumen keluar dari aplikasi Deliv
5	<i>Alternatives Flow</i>	Konsumen memilih button “tidak” akan tetap pada aplikasi Deliv
6	<i>Post Condition</i>	Konsumen <i>logout</i> dari aplikasi Deliv

12. Driver – Login Driver

Tabel 3.13 User Story Driver Login

No.	Alur	Keterangan
1	<i>Actor</i>	Driver
2	<i>Objective</i>	Driver dapat login ke aplikasi driver Deliv
3	<i>Pre-Condition</i>	Driver sudah terdaftar sebelumnya
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Driver membuka aplikasi driver Deliv 2. Driver login menggunakan akunnya 3. Driver masuk kedalam screen driver Deliv
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Driver berada pada screen home aplikasi driver Deliv

13. Driver – Menerima Order Driver

Tabel 3.14 *User Story* Menerima Order

No.	Alur	Keterangan
1	<i>Actor</i>	Driver
2	<i>Objective</i>	Driver dapat menerima orderan melalui aplikasi android driver
3	<i>Pre-Condition</i>	Driver sudah login aplikasi driver Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Driver membuka aplikasi driver Deliv 2. Driver berada pada <i>screen home</i> 3. Driver mengaktifkan penerimaan orderan 4. Ketika order masuk , driver menerima orderan
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Driver berada pada screen home aplikasi driver Deliv

14. Driver – Menolak Order Driver

Tabel 3.15 *User Story* Menerima Order

No.	Alur	Keterangan
1	<i>Actor</i>	Driver
2	<i>Objective</i>	Driver dapat menerima orderan melalui aplikasi android driver
3	<i>Pre-Condition</i>	Driver sudah login aplikasi driver Deliv
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 5. Driver membuka aplikasi driver Deliv 6. Driver berada pada <i>screen home</i> 7. Driver mengaktifkan penerimaan orderan 8. Ketika order masuk , driver menolak orderan
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Driver berada pada screen home aplikasi driver Deliv

15. Driver – Logout Driver

Tabel 3.16 *User Story Driver Logout*

No.	Alur	Keterangan
1	<i>Actor</i>	Driver
2	<i>Objective</i>	Driver dapat <i>logout</i> dari <i>aplikasi driver</i> Deliv
3	<i>Pre-Condition</i>	Driver berada pada <i>screen home</i>
4	<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Driver membuka aplikasi driver Deliv 2. Driver memilih menu Akunku 3. Driver memilih tombol icon keluar 4. Driver konfirmasi untuk keluar dari <i>aplikasi</i>
5	<i>Alternatives Flow</i>	-
6	<i>Post Condition</i>	Driver keluar dari <i>aplikasi driver</i> Deliv

3.4.3 Product Backlog Item

Berdasarkan User Story yang telah disusun, Tim Pengembang, Scrum Master dan Product Owner Menyusun Product Backlog Item yang dijadikan landasan dalam proses Sprint selama 6 Sprint. Hasil Product Backlog Item sebagai berikut :

Tabel 3.17 *Product Backlog Item*

ID	Nama	Kepentingan	Perkiraan Waktu	Catatan
1	Fitur <i>Login</i> Konsumen	5	3 Hari	
2	Fitur <i>Register</i> Konsumen	3	3 Hari	
3.	Fitur memilih lokasi pengiriman deexpress	9	5 Hari	-
6.	Fitur memilih lokasi penjemputan deride	10	5 Hari	
7.	Fitur memilih lokasi tujuan deride	20	4 Hari	
8.	Fitur memilih roda dua	5	5 Hari	-
9.	Fitur merubah profile	16	3 Hari	
14.	Screen Term of Service	37	1 Hari	
15.	Screen Privacy Policy	25	1 Hari	
16.	Fitur driver login	4	2 Hari	
17.	Fitur driver menerima order	21	5 Hari	
18.	Fitur driver menolak order	22	5 Hari	

Pada tabel *Product Backlog Item* terdapat 15 poin yang dituliskan oleh *Product Owner* dan didiskusikan dengan tim pengembang. Dalam penulisan tabel *Product Backlog Item* terdapat penjelasan mengenai setiap item seperti kepentingan, catatan dan perkiraan waktu. Kepentingan merupakan nilai yang diberikan oleh *product owner*. Semakin tinggi angka tersebut maka semakin penting fitur tersebut. Jarak nilai kepentingan adalah dari skala 1 – 100. Sedangkan untuk kolom waktu, berisikan perkiraan tim pengembangan untuk bisa mengerjakan fitur tersebut. Kolom catatan berisikan informasi lain yang mungkin diperlukan dalam pengembangan fitur tersebut.

3.5 Manajemen Komunikasi Tim

Dalam pengembangan perangkat lunak, scrum sangat mementingkan komunikasi tim. Sesuai dengan salah satu Agile Manifesto yaitu “Individu dan

interaksi lebih dari proses dan sarana perangkat lunak” ,dengan begitu para tim pengembang dan *Product Owner* menyepakati media komunikasi secara daring guna memudahkan komunikasi antara para tim pengembang itu sendiri dan juga product owner.

a) Whatsapp

Para tim pengembang, scrum master dan *Product Owner* sepakat untuk menggunakan perangkat lunak berkirim pesan menggunakan Telegram untuk membuat grup dimana para *stakeholder* bisa saling berinteraksi dan berkomunikasi.

Pemilihan whatsapp dalam pengembangan perangkat lunak disebabkan whatsapp bisa dengan mudah berkirim file , gambar dan history chat bisa ditelusuri karena tersimpan di *server whatsapp*.

b) Trello

Trello merupakan sebuah situs yang digunakan untuk memudahkan komunikasi dalam memberikan report mengenai status / keadaan dalam pengembangan perangkat lunak sehingga para tim pengembang , scrum master dan *Product Owner* bisa saling terbuka dalam pengerjaan aplikasi Android Deliv.

3.6 Identifikasi Resiko

Dalam inception juga diperlukan persiapan untuk menangani resiko yang akan terjadi didalam pembuatan aplikasi android Deliv sehingga para tim pengembang berdiskusi bersama-sama untuk mengetahui resiko yang akan dihadapi.

a. Resiko Bisnis

Salah satu resiko yang dihadapi ketika pengembangan aplikasi yang memberikan dampak pada bisnis adalah alur proses bisnis dan kesesuaian fitur aplikasi terhadap bisnis Deliv yang sedang berjalan. Meliputi alur bisnis ojek online, antar makanan / barnag yang terjadi di Deliv, konsumen yang akan menjadi target dari aplikasi android Deliv. Jika pengerjaan aplikasi Deliv terlambat / tidak sesuai dengan jadwal yang direncanakan, maka *Product Owner* akan mendapatkan kerugian karena bisnis tidak bisa beroperasi.

b. Resiko Teknikal

Resiko yang dihadapi para tim pengembang adalah perubahan teknologi yang terus menerus terjadi. Sehingga para pengembang harus menyesuaikan kebutuhan perangkat lunak yang akan digunakan dan memastikan bahwa perangkat lunak yang digunakan dalam pengembangan aplikasi Deliv adalah versi yang stabil.

c. Resiko Operational

Resiko operational adalah resiko yang dihadapi oleh para konsumen dari Deliv. Resiko ini meliputi :

- 1) Kesesuaian desain aplikasi untuk kenyamanan konsumen dalam menggunakan aplikasi di Deliv

d. Resiko Proses

Resiko proses adalah resiko yang dihadapi tim pengembang, scrum master dan *Product Owner* dalam perjalanannya pengembangan perangkat lunak mulai dari komunikasi, *bug* dalam perangkat lunak, perubahan yang diinginkan oleh *Product Owner*.

3.7 Analisa Kebutuhan Perangkat

Tahap pengumpulan kebutuhan perangkat adalah tahap dimana para tim pengembang berdiskusi bersama-sama mendefinisikan format perangkat, mengidentifikasi semua kebutuhan perangkat, teknologi yang akan dipakai dan garis besar sistem yang akan dibuat.

a. Mendefinisikan Perangkat Lunak

Dalam tahap ini para tim pengembang bersama sama mendefinisikan perangkat lunak, identifikasi perangkat lunak ,teknologi yang dipakai dalam pengembangan aplikasi Android Deliv . Berikut kebutuhan perangkat lunak pada penelitian :

Tabel 3.18 Kebutuhan Perangkat Lunak

No	Kebutuhan Perangkat Lunak
1.	Apache Web Server
2.	MySQL
3.	PHP
4.	Visual Studio Code
5.	Sketch
6.	Android Studio
7.	Node JS
8.	Firebase

b. Sumber Daya

Sumber daya yang dibutuhkan dalam pengembangan aplikasi Android Deliv adalah para tim pengembang yang terdiri dari programmer *Backend* dan *Android Developer* yang bertugas mengembangkan Aplikasi Android Deliv. Sedangkan akan ada *Product Owner* sebagai seseorang pemilik dari perangkat lunak tersebut.

c. Waktu

Ditinjau dari waktu, waktu yang dibutuhkan para tim pengembang

disesuaikan dengan *Sprint* yang telah disepakati, dimana setiap *Sprint* akan berlangsung selama 2 minggu. Rencana akan ada 6 *Sprint* untuk bisa menyelesaikan Aplikasi Android Deliv sesuai dengan konsep MVP.

d. Biaya

Ditinjau dari segi biaya , tim pengembang Deliv dibangun dengan menggunakan perangkat lunak yang bersifat terbuka / aplikasi gratis , sedangkan untuk tahap produksi maka para tim pengembang menyewa hosting dan membeli domain untuk aplikasi Android Deliv

3.8 Desain Sistem

3.8.1 Use Case Diagram

User Case Diagram menggambarkan tentang aktifitas yang dilakukan oleh setiap pengguna pada sistem yang sedang dibangun. Dalam pembangunan sistem ini, ada 2 aktor yang berperan yaitu konsumen dan driver Deliv. *User case diagram* digambarkan secara keseluruhan fungsional yang berada dalam sistem yang sedang dibangun. Berikut adalah *use case diagram* pada aplikasi Android Deliv pada gambar :



Gambar 3.3 Use Case Diagram

Pada gambar 3.3 ada dua aktor yang berperan didalam sistem Deliv yaitu Konsumen dan Driver Deliv. Masing-masing aktor memiliki peranan yang berbeda didalam sistem. Konsumen bisa melakukan aktifitas yaitu memesan layanan Deride, memesan layanan Deexpress, melihat screen Term of Service, melihat screen Privacy Policy dan merubah serta melihat profile Deliv. Sedangkan untuk driver Deliv bisa melakukan beberapa aktifitas yang berbeda yaitu menerima pesanan deride / menolak pesanan deride, dan menerima / menolak pesanan deexpress Deliv.

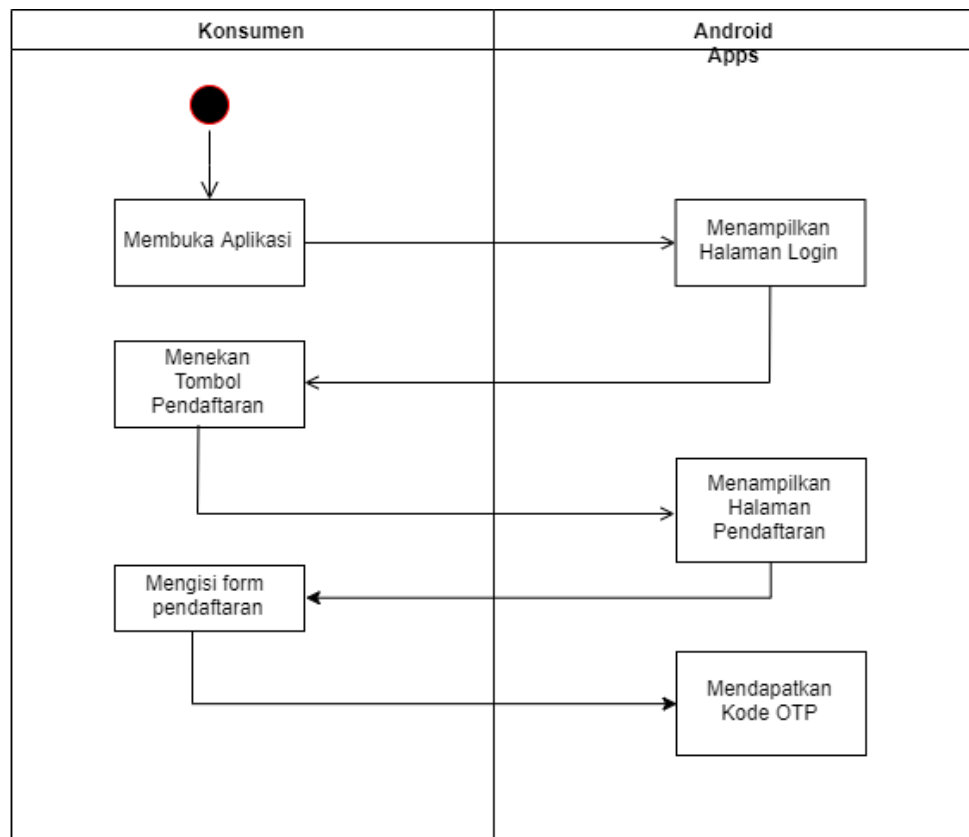
3.8.2 Activity Diagram

Activity diagram merupakan alur di dalam sebuah perangkat lunak. Pada *activity diagram* aplikasi Android Deliv dan aplikasi Android Driver Deliv menjelaskan proses yang berjalan di dalam perangkat lunak oleh pengguna dari awal menjalankan perangkat lunak hingga menutup perangkat lunak. Pada *activity diagram* aplikasi Android Deliv, terdapat 2 jenis pengguna, yaitu konsumen dan driver Deliv. Setiap pengguna memiliki

peranan , yaitu :

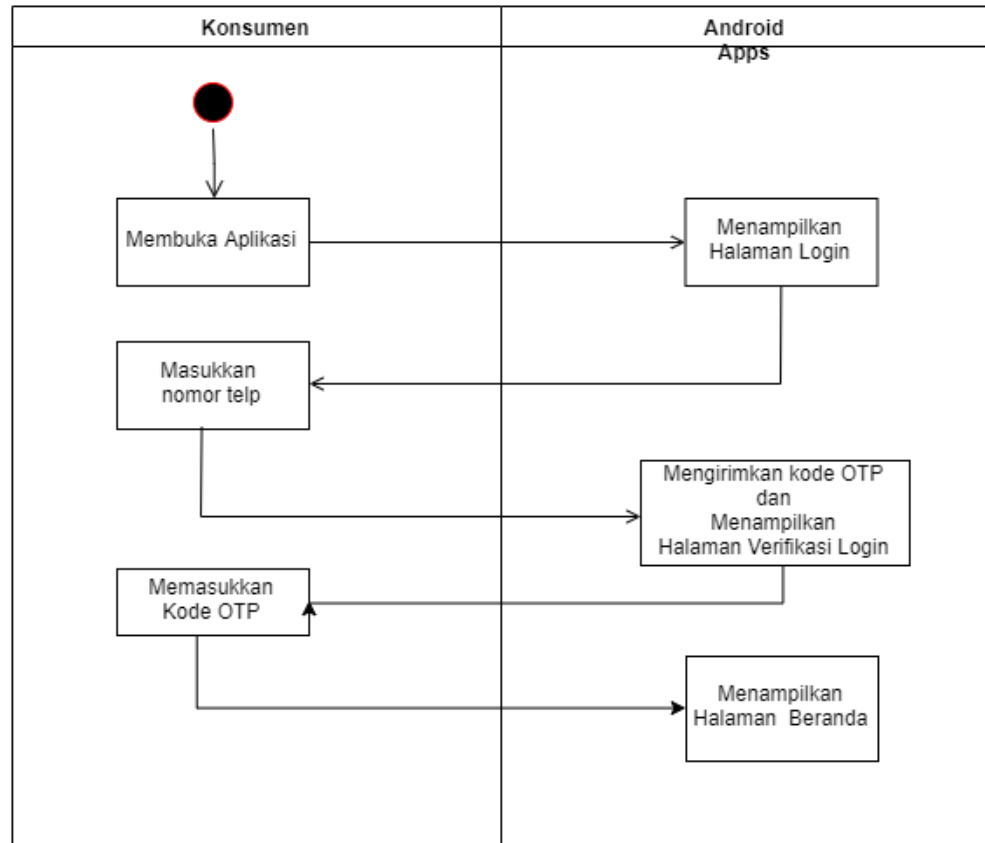
1) Konsumen

Konsumen dapat melakukan beberapa aktifitas di dalam aplikasi Android Deliv seperti mendaftarkan diri, masuk ke aplikasi, memesan deride, memesan deexpress, dan merubah screen profile.



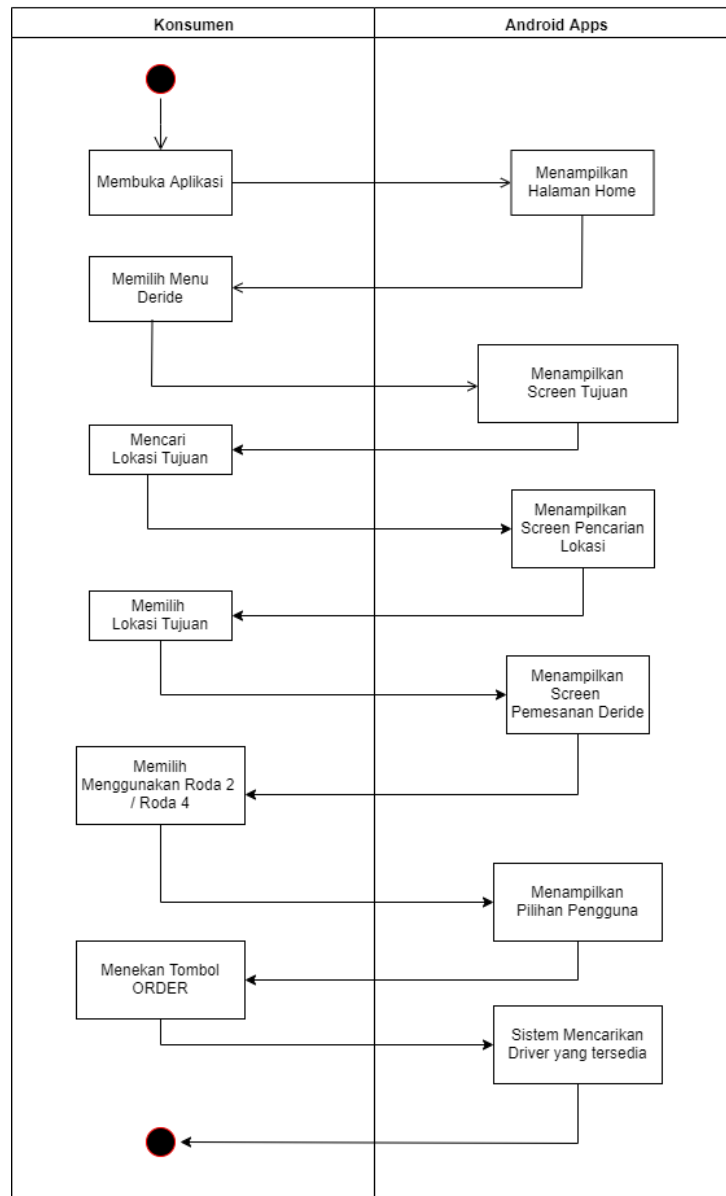
Gambar 3.4 *Activity diagram* konsumen mendaftar di Deliv

Pada rancangan *activity diagram* gambar 3.4 memperlihatkan tahapan konsumen dalam mendaftarkan diri pada perangkat lunak Deliv.



Gambar 3.5 *Activity diagram* Konsumen Login ke Aplikasi Deliv

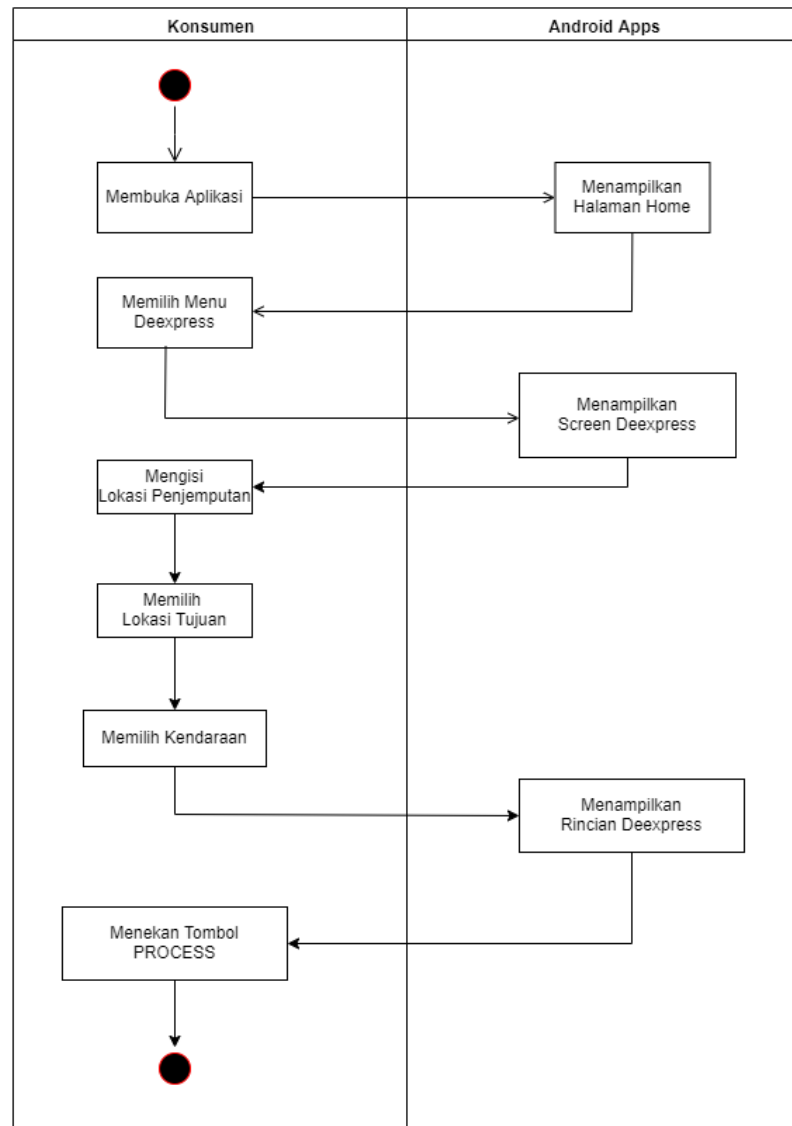
Pada rancangan *Activity Diagram* Konsumen mendaftarkan diri di *Aplikasi Deliv* adalah dengan membuka terlebih dahulu *Aplikasi Deliv* dan konsumen berada pada *screen Login*. Konsumen karena tidak memiliki akun, akan menekan tombol *Register* dan diarahkan pada *screen Register*. Pada *screen Register*, Pengguna akan mengisi form pendaftaran dengan menggunakan data asli. Setelah mendaftar, dan menekan tombol *Sign Up* pengguna akan mendapatkan kode OTP untuk verifikasi / login.



Gambar 3.6 *Activity diagram* konsumen memesan Deride

Pada rancangan *activity diagram* konsumen masuk ke *aplikasi* Deliv dengan memasukkan nomor telepon konsumen. Setelah konsumen memasukkan nomor teleponnya, mereka akan mendapatkan kode OTP (One Time Password) dan konsumen akan diarahkan ke *screen verification* OTP untuk memasukkan kode OTP yang telah didapat. Apabila kode benar, maka akan berpindah ke *screen home aplikasi* Deliv.

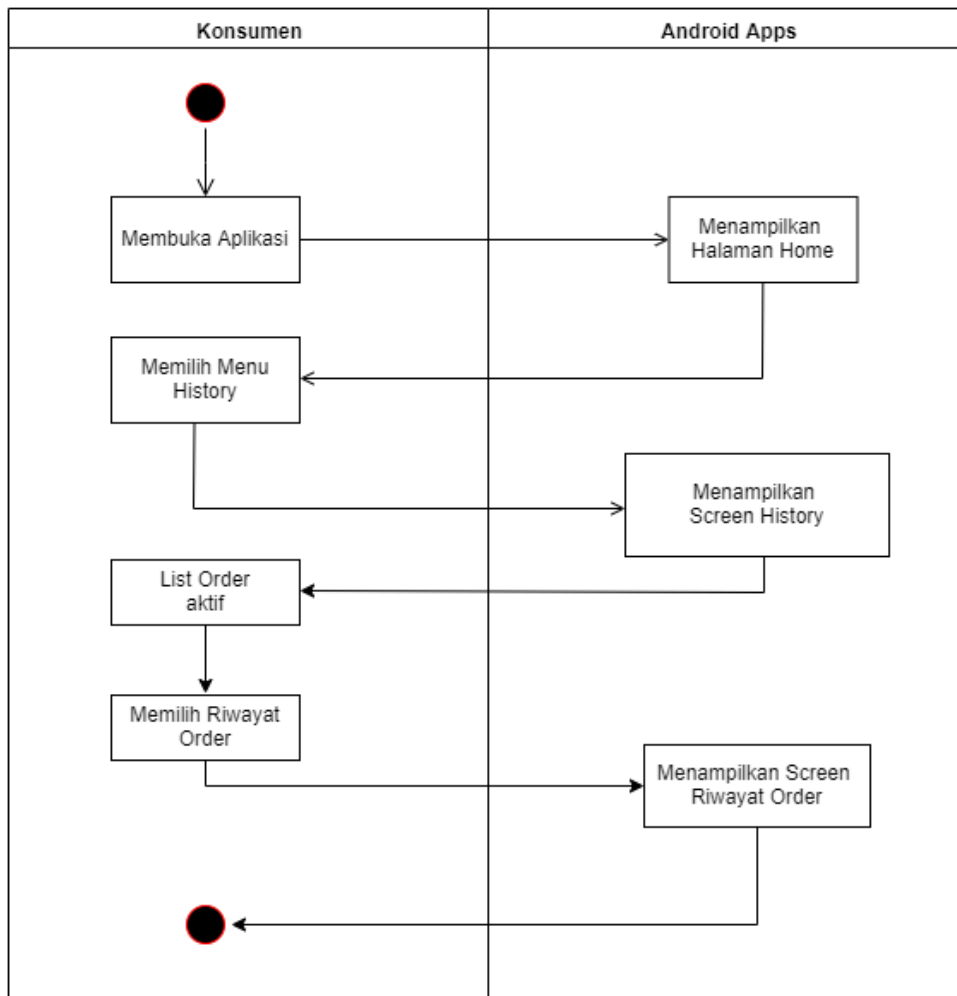
Pada rancangan *activity diagram* konsumen memesan Deride pada gambar 3.6, tahapan yang akan dilalui oleh konsumen pertama kali adalah dengan membuka *aplikasi* Deliv terlebih dahulu. Lalu Aplikasi akan menampilkan screen home pada aplikasi Deliv. Pengguna selanjutnya memilih menu Deride untuk bisa memesan layanan Deride. Ketika sudah memilih menu Deride, pengguna akan diarahkan ke screen Maps yang berisikan tujuan yang akan dituju pengguna. Pengguna tinggal mengisi lokasi tujuan Deride yang selanjutnya aplikasi akan mencari lokasi tujuan. Bila lokasi tujuan telah ditentukan maka aplikasi akan menampilkan screen pemesanan Deride. Pengguna bisa memilih menggunakan Roda 2 atau Roda 4. Bila sudah memilih, maka aplikasi menampilkan pilihan pengguna dan tinggal menekan tombol ORDER dan aplikasi akan mencari driver Deliv yang tersedia.



Gambar 3.7 *Activity diagram* konsumen memesan layanan Deexpress

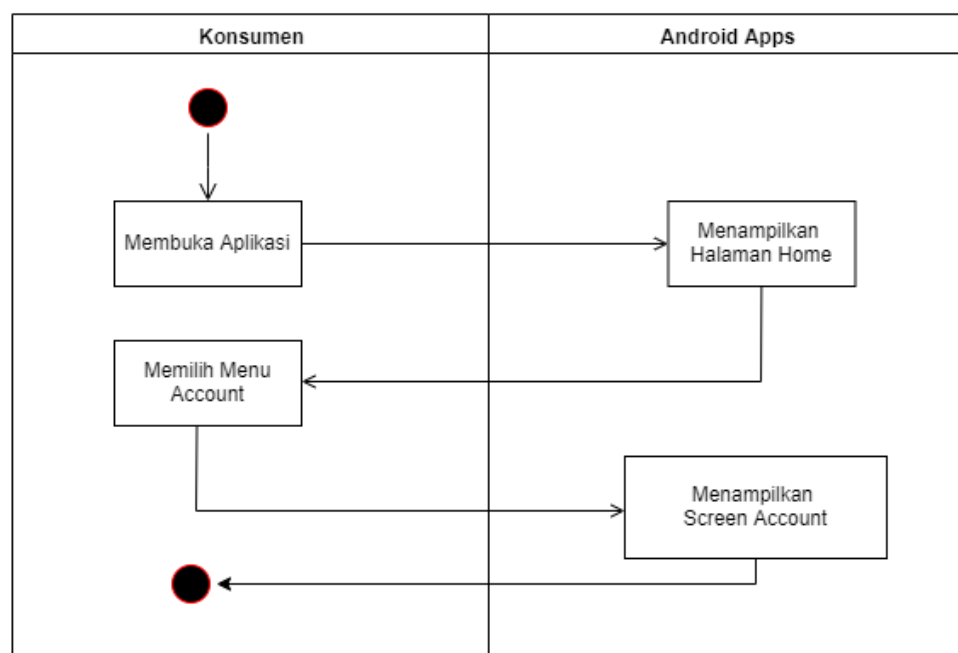
Pada rancangan *activity diagram* konsumen memesan Deride pada gambar 3.7 , tahapan yang akan dilalui konsumen pertama kali adalah dengan membuka aplikasi Deliv yang nantinya akan langsung diarahkan pada *screen Home*. Selanjutnya, memilih konsumen memilih menu Deexpress dan sistem akan mengarahkan ke *screen Deexpress*. Saat berada pada *screen Deexpress* pengguna akan mengisi form untuk memasukkan

lokasi penjemputan, lokasi tujuan dan terakhir akan memilih kendaraan yang akan digunakan untuk pengiriman barang. Setelah itu, sistem akan menampilkan rincian Deexpress yang telah kita lengkapi formulirnya dan konsumen tinggal menekan tombol *process* dan tinggal menunggu driver datang.



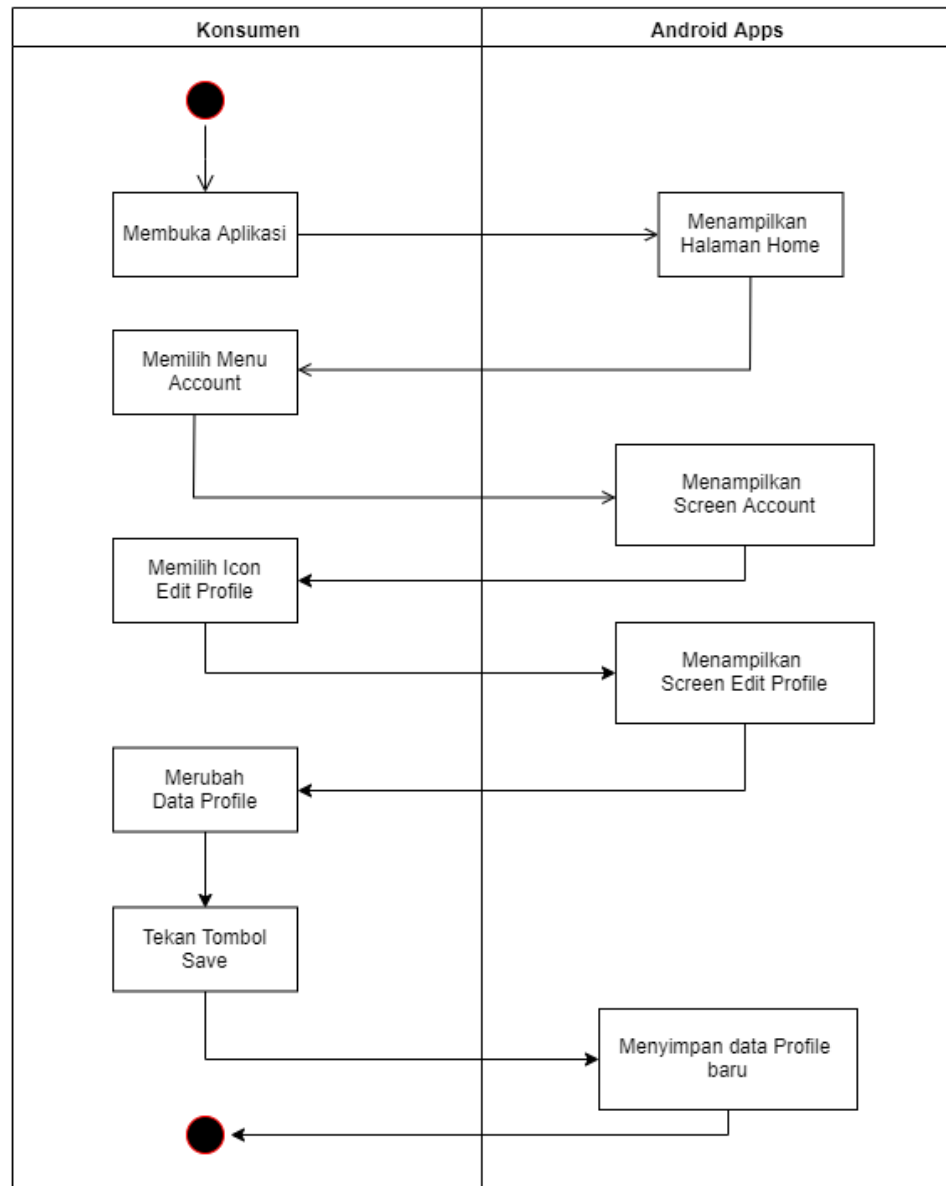
Gambar 3.8 Activity diagram konsumen melihat Riwayat Order

Pada gambar 3.8 *Activity Diagram* konsumen melihat riwayat order, dengan membuka aplikasi terlebih dahulu dan saat aplikasi sudah berada pada screen home, konsumen menekan menu History untuk bisa diarahkan pada screen History yang berisikan list orderan yang masih aktif. Selanjutnya, konsumen bisa melihat riwayat order dengan memilih tombol riwayat order dan sistem akan menampilkan screen riwayat order dari orderan yang gagal / telah diselesaikan.



Gambar 3.9 *Activity diagram* konsumen melihat Profile

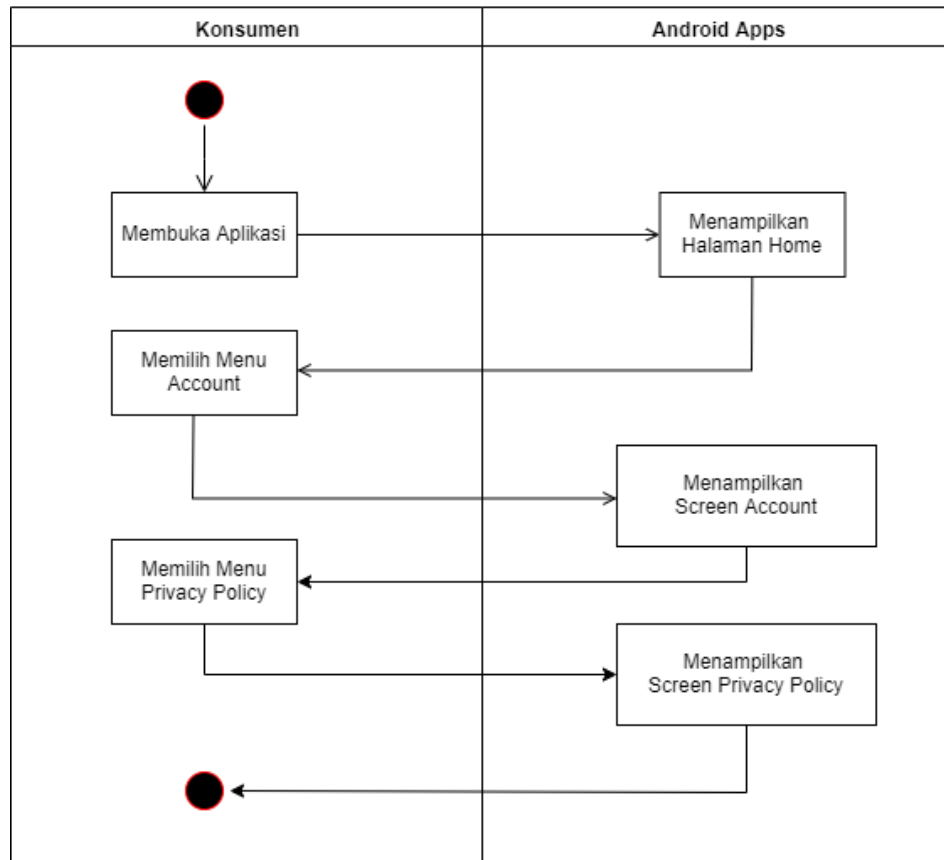
Gambar 3.9 memperlihatkan konsumen dalam melihat profile konsumen dengan langkah awal adalah membuka aplikasi Deliv dan sistem akan menampilkan *screen Home*. Saat berada pada *screen Home*, konsumen akan memilih menu Account dan diarahkan pada *screen Account* yang berisikan profil konsumen Deliv.



Gambar 3.10 *Activity diagram* konsumen merubah Profile

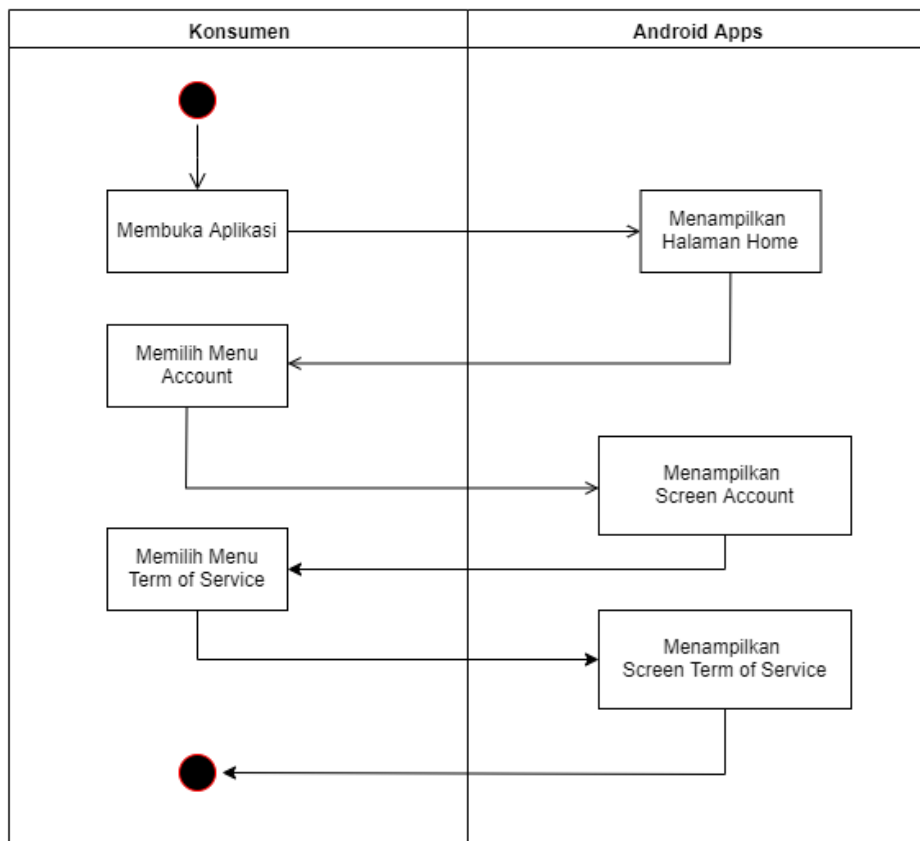
Pada gambar 3.10 *Activity diagram*, konsumen bisa pula merubah profile pada data Deliv dengan memilih menu Accoun pada *screen home*, dan saat sistem menampilkan *screen profile*, konsumen memilih icon untuk merubah profile konsumen Deliv. Selanjutnya, konsumen akan diarahkan pada halaman *edit profile* dan konsumen diharuskan mengisi form /data

yang akan dirubah. Setelah merubah data profile, maka konsumen bisa menekan tombol save dan data akan secara otomatis tersimpan.



Gambar 3.11 *Activity diagram* konsumen melihat Privacy Policy

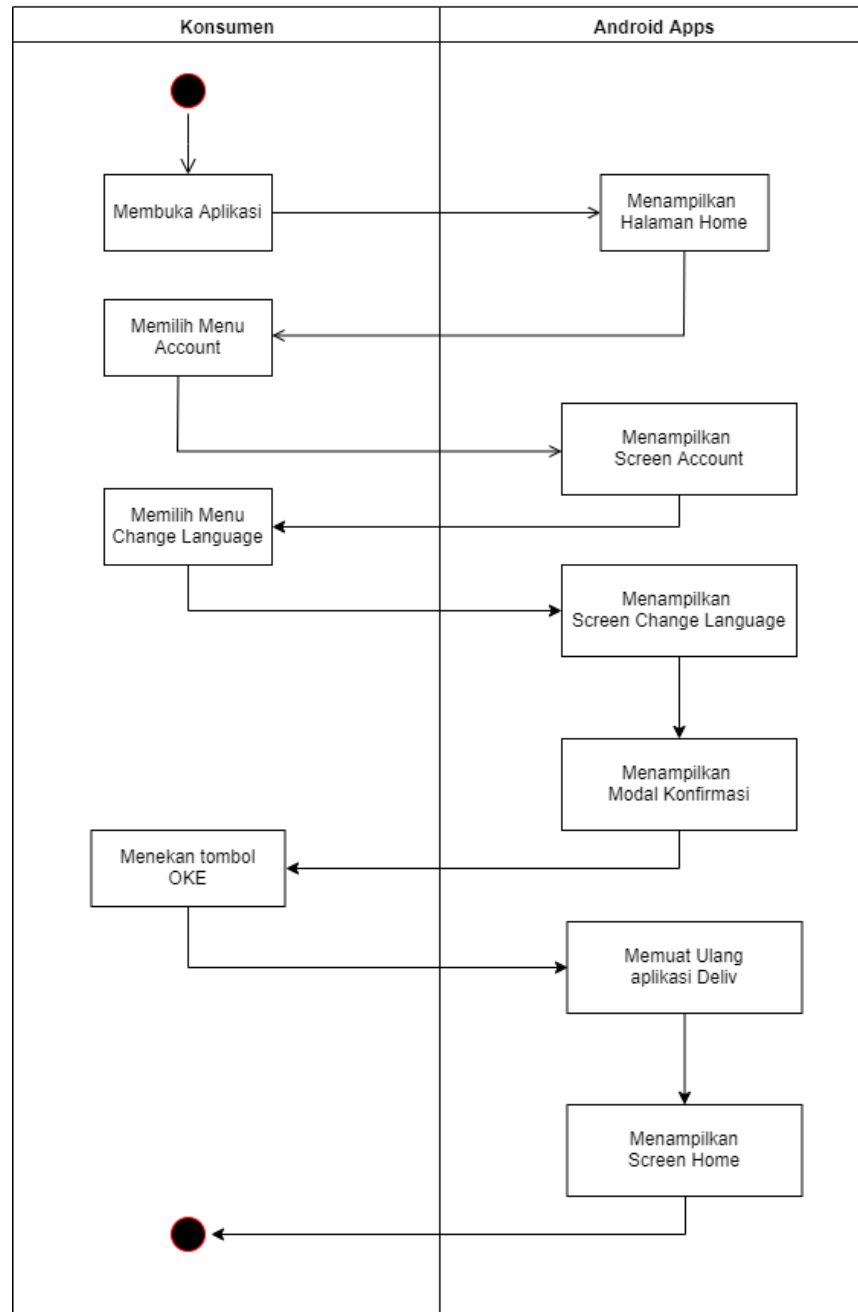
Pada gambar 3.11 Pengguna juga bisa melihat screen Privacy Policy dengan menekan menu Account saat pada *screen Home* agar diarahkan pada *screen Account*. Saat berada pada *screen Account*, konsumen memilih tombol Privacy Policy yang tersedia pada *screen* tersebut untuk selanjutnya diarahkan pada *screen Privacy Policy* yang menampilkan ketentuan privasi Deliv tersebut.



Gambar 3.12 *Activity diagram* konsumen melihat Term of Service

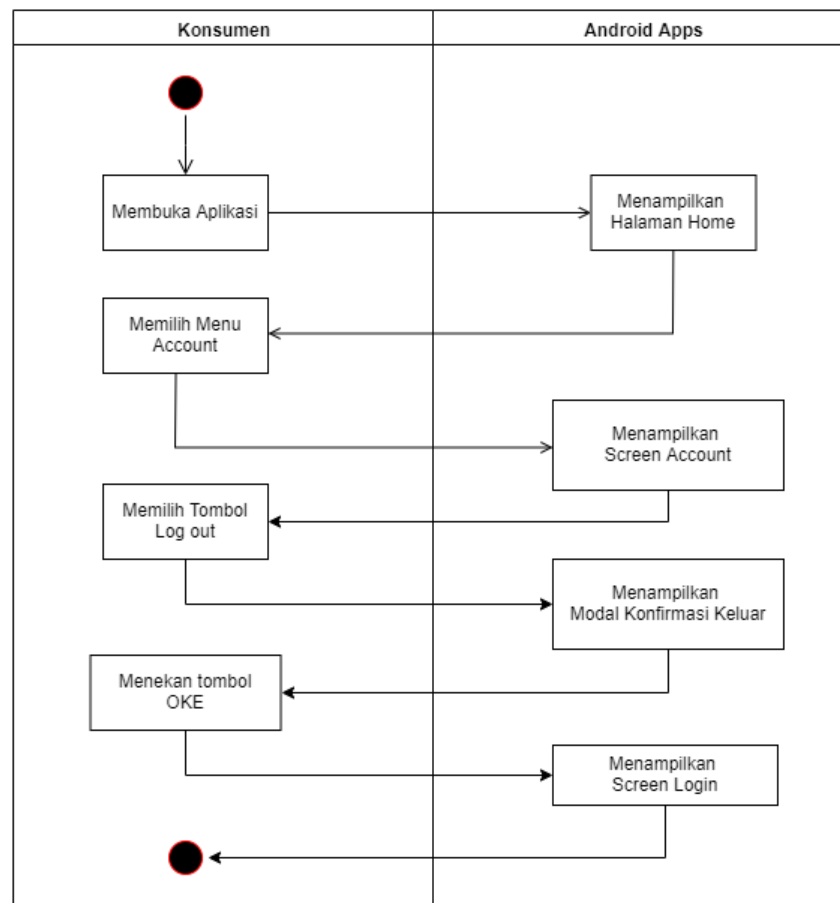
Pada gambar 3.12 Pengguna juga bisa melihat screen Term of Service dengan menekan menu Account saat pada *screen Home* agar diarahkan pada *screen Account*. Saat berada pada *screen Account*, konsumen memilih tombol Term of Service yang tersedia pada *screen* tersebut untuk

selanjutnya diarahkan pada *screen* Term of Service yang menampilkan ketentuan layanan Deliv tersebut.



Gambar 3.13 Activity diagram konsumen merubah bahasa

Pada gambar 3.13 Pada *screen Account* bisa pula merubah bahasa. Caranya adalah dengan pengguna memilih tombol *Change Language* dan akan diarahkan pada *screen* bahasa yang tersedia pada perangkat tersebut. . Selanjutnya konsumen bisa memilih menggunakan bahasa yang diinginkan dan sistem akan menampilkan modal konfirmasi untuk penggantian bahasa. Konsumen menekan tombol konfirmasi dan sistem akan memuat ulang aplikasi Deliv dan menampilkan *screen Home* yang telah berganti bahasa.

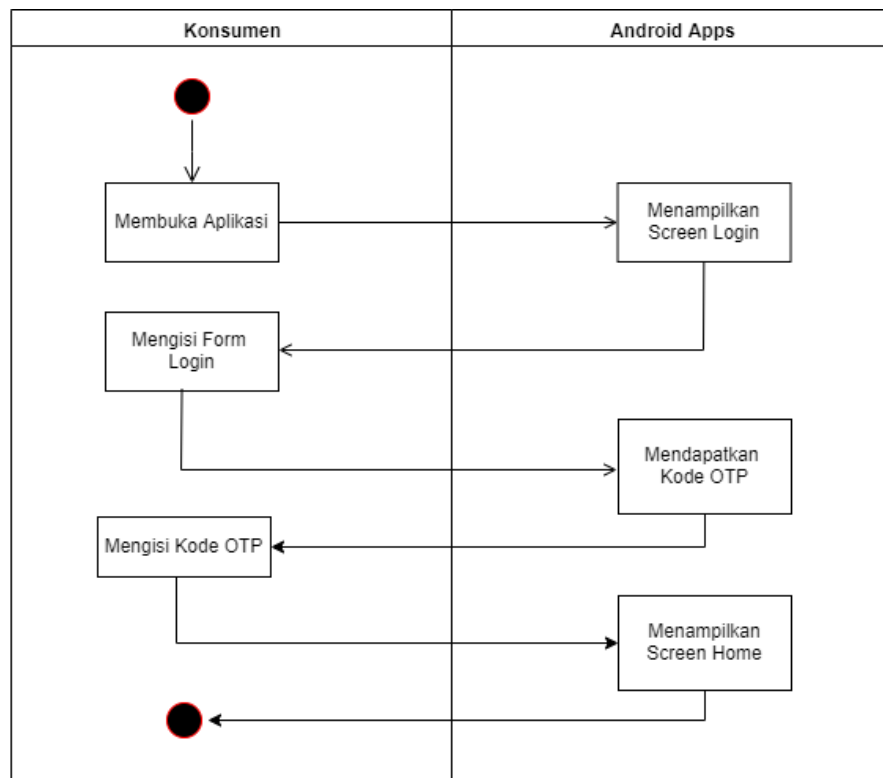


Gambar 3.14 *Activity diagram* konsumen logout dari aplikasi Deliv

Gambar 3.14 *Activity diagram* konsumen *logout* dari aplikasi Deliv menggambarkan alur konsumen untuk bisa keluar dari aplikasi Deliv saat konsumen berada pada *screen Account*, maka konsumen memilih tombol *logout* yang ada pada *screen Account* dan aplikasi akan memunculkan modal konfirmasi . Bila konsumen menekan tombol OK, maka konsumen akan keluar dari aplikasi Deliv dan diarahkan ke *screen Login* Deliv.

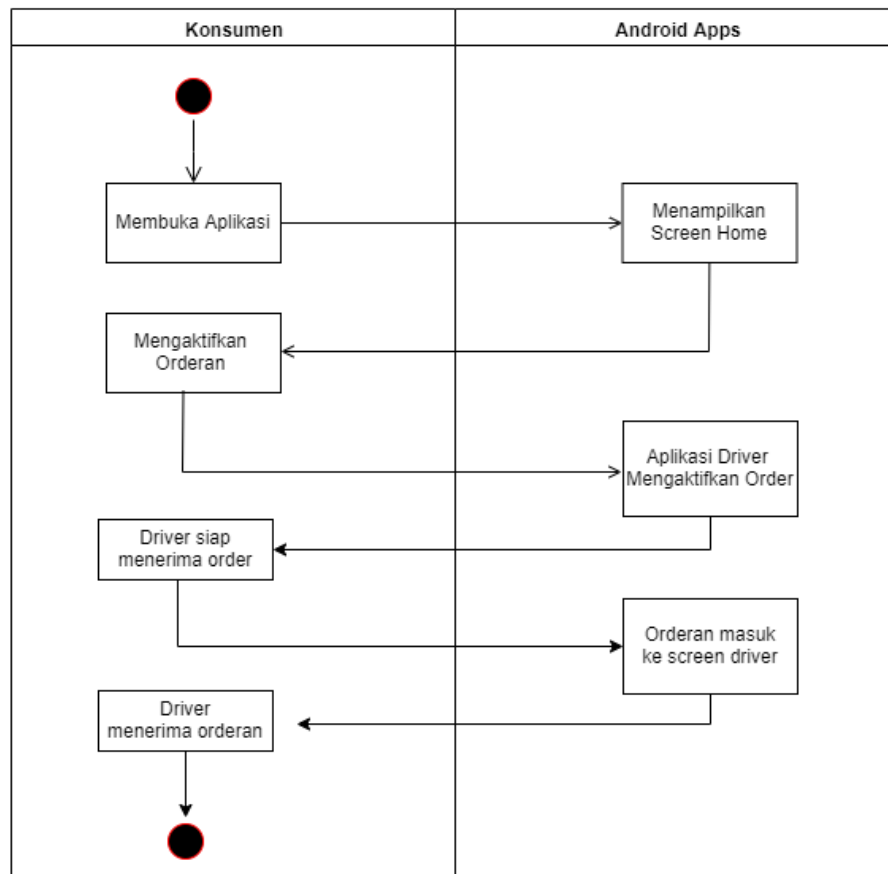
2) Driver

Driver dapat melakukan beberapa aktifitas di aplikasi driver Deliv. Berbeda dengan aplikasi Deliv, aplikasi driver Deliv dikhususkan hanya untuk driver Deliv saja. Driver Deliv bisa melakukan beberapa kegiatan seperti login, menerima orderan deride / deexpress, dan logout.



Gambar 3.15 *Activity diagram* driver login ke aplikasi driver Deliv

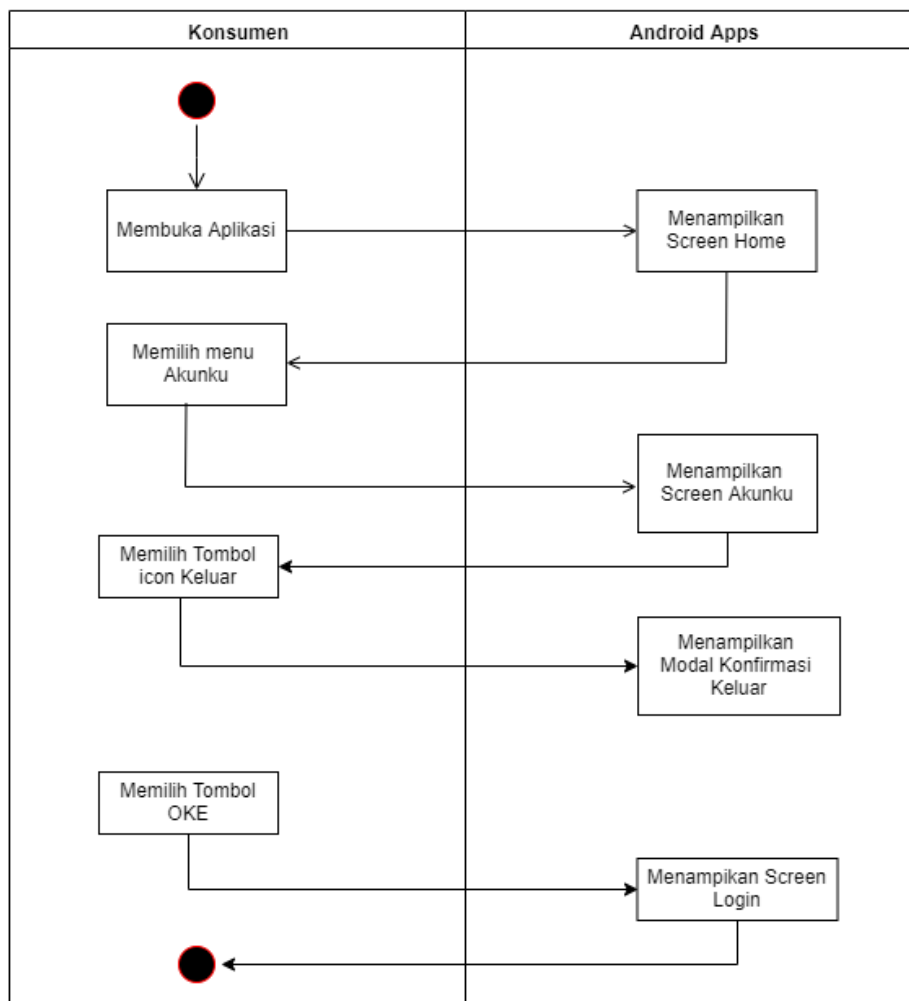
Gambar 3.14 Activity diagram driver untuk bisa login ke aplikasi driver Deliv. Pertama yang harus dilakukan driver adalah dengan cara membuka aplikasi driver Deliv. Saat membuka aplikasi driver Deliv maka akan muncul *screen login*. Driver Deliv mengisi form untuk bisa login ke aplikasi driver Deliv. Apabila data sudah benar, maka driver akan mendapatkan kode OTP yang akan digunakan untuk *login* ke aplikasi driver Deliv. Selanjutnya driver tinggal mengisi kode OTP ke dalam *screen verifikasi OTP* dan apabila kode benar maka akan diarahkan ke *screen home* aplikasi driver Deliv.



Gambar 3.16 Activity diagram driver menerima order ke aplikasi driver Deliv

Pada gambar 3.16, driver menerima order dari aplikasi driver Deliv yang

mampu untuk menerima order. Langkahnya adalah driver membuka aplikasi Deliv yang selanjutnya akan diarahkan ke halaman Screen Home. Pada screen home akan ada menu untuk mengaktifkan orderan. Setelah Order Aktif, maka driver secara otomatis akan siap menerima order. Saat orderan masuk, maka akan muncul notifikasi bahwa order telah masuk ke dalam screen driver dan driver bisa menerima orderan.



Gambar 3.17 Activity Diagram driver logout dari aplikasi driver Deliv

Driver Deliv bisa keluar dari Aplikasi driver Deliv dengan sebelumnya sudah berada pada *screen home* dan memilih menu Akunku untuk selanjutnya aplikasi

driver Deliv akan menampilkan screen akunku dan driver memilih tombol icon Keluar. Saat driver memilih tombol icon keluar, sistem akan memproses dengan menampilkan modal konfirmasi keluar. Memilih modal konfirmasi keluar, bila driver memilih tombol OKE maka driver akan logout dari aplikasi driver deliv dan sistem akan menampilkan screen login pada aplikasi driver deliv.

3.9 Sprint Planning

Tahapan dimana mendefinisikan pekerjaan yang akan dilaksanakan di dalam *Sprint*, dibuat secara kolaboratif oleh seluruh anggota Tim Scrum.

1. Waktu *Sprint*

Berdasarkan diskusi Team Pengembang dan *Product Owner*, *Sprint* akan dilaksanakan selama 12 minggu.

2. Mendefinisikan Fitur untuk Minimum Viable Product

Pada tahap ini, para tim pengembang dan *Product Owner* bersama dengan scrum master mendefinisikan mengenai sejauh mana perangkat lunak aplikasi Android akan dikembangkan untuk mencapai tahap *Minimum Viable Product*. Fitur-fitur MVP diambil dari *Product Backlog Item* (Tabel 3.14) yang diseleksi oleh tim pengembang dan *Product Owner* sehingga menghasilkan beberapa fitur yang akan dikembangkan oleh tim pengembang.

Tabel 3.19 Minimum Viable Product

No	Fitur
1.	Aplikasi Deliv
2.	Screen Login
3.	Screen Register
4.	Screen Home
5.	Screen Profile
6.	Screen Deride
7.	Screen Decar
8.	Screen Deexpress
9.	Screen History Order
10.	Screen Privacy Policy
11.	Screen Term of Service
	Aplikasi Driver Deliv
1.	Screen Login
2.	Screen Home
3.	Screen Akun
4.	Screen Privacy Policy
5.	Screen Term of Service

3. Merencanakan Prioritas *Product Backlog Item*

Tahap ini tim pengembang dan *Product Owner* bekerja sama dan berdiskusi untuk merencanakan fitur-fitur dari *Product Backlog Item* untuk diletakan di dalam *Sprint* :

a. *Sprint Ke - 1*

Tabel 3.20 *Task Sprint Pertama*

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
Fitur Register Konsumen	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Login Konsumen	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
	<i>Testing</i>	Tim Pengembang
Fitur Login Driver	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang

b. Sprint Ke - 2

Tabel 3.21 *Task Sprint* Kedua

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
Fitur Memilih Deride	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Memilih Lokasi Penjemputan Deride	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Memilih Lokasi Tujuan Deride	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang

c. Sprint Ke – 3

Tabel 3.22 *Task Sprint* Ketiga

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
Fitur Mencari Driver	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Driver Menerima Order	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang

d. Sprint Ke – 4

Tabel 3.23 *Task Sprint* Keempat

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
Fitur Driver Menolak Order	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Memilih Deexpress	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Memilih Lokasi Penjemputan Deexpress	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang

e. Sprint Ke – 5

Tabel 3.24 *Task Sprint Kelima*

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
Fitur Memilih Lokasi Pengiriman Deexpress	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Fitur Merubah Profile Konsumen	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang

f. *Sprint Ke – 6*Tabel 3.25 *Task Sprint Keenam*

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
Fitur Logout Konsumen	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Screen Logout Driver	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Screen Term of Service	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi Android</i>	<i>Android Dev</i>
	<i>Testing</i>	Tim Pengembang
Screen Privacy Policy	Desain Antar Muka	<i>Designer</i>
	Desain Basis Data	<i>Back End Dev</i>
	Pengerjaan <i>Backend</i>	<i>Back End Dev</i>
	Pengerjaan <i>Aplikasi</i>	<i>Android Dev</i>

<i>Task Product Backlog Item</i>	<i>Task</i>	<i>Pemilik Task</i>
	<i>Android</i>	
	<i>Testing</i>	Tim Pengembang

3.10 Sprint

Pada saat melakukan Sprint , Tim Pengembang dan Scrum Master akan melakukan pekerjaan sesuai dengan Product Backlog yang telah ditentukan dan berdasarkan waktu yang telah didefinisikan sebelumnya. Ketika proses pengerjaan Sprint terjadi, Tim Pengembang dan Scrum Master akan mengadakan Daily Scrum setiap harinya. Sedangkan pada akhir Sprint, akan diadakan Sprint Review dan Sprint Retrospective.

3.11 Daily Scrum

Pada Tahapan ini, setiap harinya ketika berada di dalam *Sprint*, para pengembang didampingi dengan scrum master bertemu untuk mendiskusikan beberapa hal seperti

1. Pekerjaan yang telah dilakukan tim pengembang pada hari kemarin atau yang sudah dilakukan.
2. Pekerjaan yang akan dilakukan para tim pengembang pada hari ini.
3. Permasalahan yang didapatkan para tim pengembang dalam pengembangan perangkat lunak.

Semua hasil dari daily scrum akan didiskusikan untuk memberikan transparansi antar tim pengembang.

3.12 Sprint Review

Pada tahap ini tim pengembang, *Scrum Master* serta *Product Owner* melakukan pengecekan, diskusi mengenai increment yang telah dilakukan ketika melaksanakan *Sprint*. Melihat setiap akhir *Sprint* untuk mendapatkan

umpan balik atas apa yang telah dilakukan oleh tim pengembang. *Product Owner* juga ikut dalam *Sprint Review* untuk mengetahui sejauh mana perangkat lunak telah dikembangkan dan untuk mendapatkan fitur apa saja yang telah diselesaikan tim pengembang.

3.13 Sprint Retrospective

Pada tahap ini tim pengembang bersama dengan *Scrum Master* memeriksa semua *task* yang telah dilakukan selama proses pelaksanaan *Sprint* 1 sampai *Sprint* 3. Beserta membuat planning untuk memperbaiki *Sprint* berikutnya. Mengalokasikan 1 – 3 Jam untuk mendiskusikan atau merapatkan *Sprint Retrospective*. *Scrum Master* akan *Sprint Backlog* dan merangkum *Sprint* yang baru saja dilaksanakan. Setiap *stakeholder* diberikan waktu untuk berbicara mengungkapkan pendapat mereka mengenai *Sprint* yang telah dilewati meliputi apa saja yang bisa diperbaiki, dan apa yang bisa dilakukan dengan cara yang berbeda.

3.14 Pengujian

Untuk melihat bagaimana metode Scrum dapat memberikan dampak dari segi efisien dan efektif maka dibuatkan pengujian sebagai berikut :

3.14.1 Pengalaman Pengguna

Untuk melakukan evaluasi produk yang dihasilkan dengan Scrum maka akan dilakukan *User Experience Questionnaire*. Hal ini untuk bisa mendapatkan pengalaman pengguna saat menggunakan produk yang digunakan dengan mencakup daya tarik, kejelasan, efisiensi, ketepatan, stimulasi dan kebaruan.

BAB 4

UJI COBA DAN PEMBAHASAN

4.1 Uji Coba

Pada tahapan uji coba, akan dijelaskan data apa saja yang akan digunakan dalam penelitian.

4.1.1 Data Uji Coba

Data uji coba pada penelitian ini menggunakan data yang didapat saat melakukan Scrum artefak meliputi keseluruhan kegiatan yang dilakukan saat pengembangan perangkat lunak Deliv mulai dari *Inception*, *Sprint*, *Sprint Review*, dan *Sprint Retrospektif* yang dilakukan oleh Tim Pengembang dan Scrum Master. Berikut daftar Tim Pengembang, Product Owner dan Scrum Master yang terlibat dalam Scrum artefak.

Tabel 4.1 Para *Stakeholder* yang terlibat dalam Scrum Artefak

No	Nama Anggota	Peran	Kategori
1	Heli Suharjono	<i>Product Owner</i>	
2	Rofiqo Fauzan Firdaus	Tim Pengembang	Android Dev
3	Adwi Arifin	Tim Pengembang	Back End Dev
4	Reksa Rangga W	Tim Pengembang	Back End Dev
5.	Rahmat Arief Hidayat	Tim Pengembang	Desainer
6.	Naufaldi Rafif S	Scrum Master	

4.2 Hasil Uji Coba

Pada hasil ujicoba penelitian dengan Scrum, berisikan seluruh dokumentasi kegiatan dalam Artefak Scrum selama dimulainya proses Scrum, yaitu *Inception* hingga selesainya 6 *Sprint* yang telah disepakati bersama antara Tim Pengembang, Scrum Master dan Product Owner.

4.2.1 Inception

a. Cakupan Proyek :

Berdasarkan diskusi antara Scrum Master, Tim Pengembang dan Product Owner , cakupan proyek yang akan dikerjakan selama 12 Minggu untuk mencapai produk yang bisa dikeluarkan dipasar dan sesuai dengan proses bisnis adalah dengan mencapai tahap produk yang *Minimum Viable Product (MVP)* sebagai berikut :

Tabel 4.2 *Minimum Viable Produk (MVP)*

No	Fitur
	Aplikasi Deliv
1.	Screen Login
2.	Screen Register
3.	Screen Home
4.	Screen Profile
5.	Screen Deride
6.	Screen Deexpress
7.	Screen History Order
8.	Screen Privacy Policy
9.	Screen Term of Service
	Aplikasi Driver Deliv
1.	Screen Login
2.	Screen Home
3.	Screen Akun
4.	Screen Privacy Policy
5.	Screen Term of Service

Sehingga diperoleh bahwa Aplikasi yang dibuat sesuai dengan MVP adalah 2 Aplikasi Android. Yaitu Aplikasi Android Deliv dan Aplikasi Android Driver Deliv.

b. Estimasi Jadwal Pengerjaan

Berdasarkan diskusi pula, diperoleh bahwa Proyek Aplikasi Android Deliv dan Aplikasi Android Driver Deliv diselesaikan selama Minggu dengan jadwal sebagai berikut :

Tabel 4.3 *Estimasi Jadwal Pengerjaan*

Tanggal	Tahapan	Product Backlog Item
1 Oktober 2018	Inception	Cakupan Proyek, Estimasi Harga dan Schedule, Menyiapkan lingkungan kerja
2 – 5 Oktober 2018	Sprint 1	Fitur Register Konsumen
		Fitur Login Konsumen
		Fitur Login Driver
8 – 12 Oktober 2018	Sprint 2	Fitur Memilih Deride
		Fitur Memilih Lokasi Deride
		Fitur Memilih Lokasi Tujuan Deride
15 – 19 Oktober 2018	Sprint 3	Fitur Konsumen Mencari Driver
		Fitur Driver Menerima Order
22 – 26 Oktober 2018	Sprint 4	Fitur Driver Menolak Order
		Fitur Memilih Deexpress
		Fitur Memilih Lokasi Penjemputan Deexpress
29 Oktober – 2 November 2018	Sprint 5	Fitur Memilih Lokasi Pengiriman Deexpress
		Fitur Merubah Profile Konsumen
5 – 9 November 2018	Sprint 6	Fitur Logout Konsumen
		Fitur Logout Driver
		Screen Term of Service
		Screen Privacy Policy

c. Menyiapkan Lingkungan Kerja

Untuk meningkatkan komunikasi, transparansi, dan dokumentasi. Pengerjaan proyek Aplikasi Android Deliv dan Driver Deliv menggunakan beberapa perkakas untuk menunjang kinerja.

d. Komunikasi :

Membuat dua grup untuk komunikasi. Pertama dengan Whatsapp Group yang dipergunakan untuk komunikasi Scrum Master, Tim Pengembang dan Product Owner . Group ini dibuat dengan tujuan untuk komunikasi Tim Pengembang, Scrum Master dan Product Owner sehingga terjadi komunikasi dan transparansi proyek yang sedang dibuat. Sehingga Product Owner bisa mengetahui proses setiap sprint yang ada dan memudahkan komunikasi tim pengembang dengan Product Owner . Selain membuat Whatsapp Group, tim pengembang dan scrum master membuat Slack Group sebagai sarana komunikasi khusus untuk proses pengembangan. Harapannya dengan menggunakan Slack Grup ini, proses komunikasi, transfer data dan proses pengembangan menjadi semakin mudah.

4.2.2 Sprint 1

a. Product Backlog Item

PBI disusun berdasarkan hasil Inception sebelumnya. Bila tidak ada perubahan maka Sprint Backlog akan mengacu pada Product Backlog Item.

Tabel 4.4 List Product Backlog Item

No.	Product Backlog Item
1.	Fitur Register Konsumen

No.	Product Backlog Item
2.	Fitur Login Konsumen
3.	Fitur Login Driver

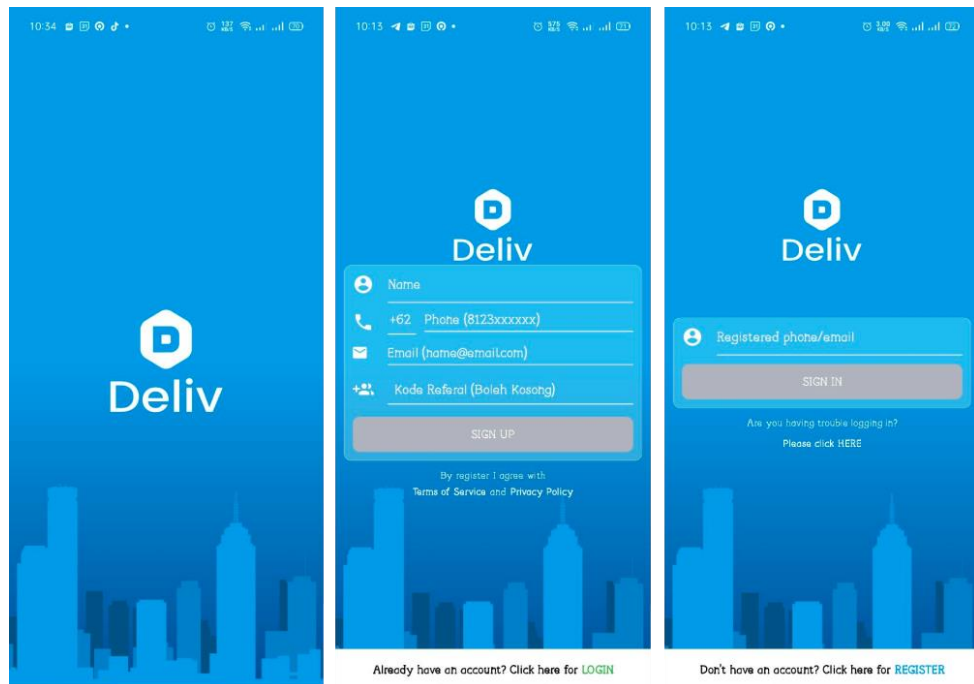
Penyusunan kembali Product Backlog Item lebih diperinci lagi berdasarkan tugas masing-masing tim pengembang untuk disesuaikan dengan kondisi pekerjaan yang sedang dikerjakan.

Tabel 4.5 Sprint Backlog Item

No.	Sprint Backlog Item
1.	Design Screen Register Konsumen
2.	API dan Database Register Konsumen
3.	Screen Android Register Konsumen
4.	Design Screen Login Konsumen
5.	API dan Database Login Konsumen
6.	Screen Android Login Konsumen
7.	Design Screen Login Driver
8.	API dan Database Login Driver
9.	Screen Android Login Driver
10.	Testing Android Konsumen
11.	Testing Android Driver

b. Hasil Sprint

Hasil sprint merupakan hasil dari proses Sprint yang telah berjalan. Hasilnya berupa produk menjadi sebuah aplikasi Android yang terdapat beberapa screen sesuai dengan Product Backlog Item yang telah didefinisikan. Beberapa screen yang telah diselesaikan pada Sprint 1 yaitu Screen Android Splashscreen, Register dan Login Konsumen



Gambar 4.1 Screen Android Login dan Register

c. Hasil Sprint Review

- 1) Scrum Master sebagai fasilitator membuka sprint review dengan salam dan sapaan kepada tim pengembang dan produk owner
- 2) Scrum Master menjelaskan sprint goal dan gambaran umum Produk Backlog Item yang dikerjakan pada sprint pertama
- 3) Demo peningkatan produk perangkat lunak dimulai. Ada dua hal yang dipresentasikan oleh tim pengembang, yaitu aplikasi android dan skema database yang sedang dibuat.
- 4) Sesi pertama anggota pengembang mendemokan aplikasi. Menceritakan aplikasi yang telah dibuat mulai dari register dan login pada aplikasi android. Pada sesi ini, pengembang juga menceritakan permasalahan permasalahan yang dihadapi saat proses pengembangan aplikasi android.

- 5) Beberapa halangan dalam pengembangan misalnya Halangan dalam komunikasi antara pengembang android dan pengembang backend / database dalam proses komunikasi. Karena pengembang android hanya seorang diri, proses pengerjaan cukup menjadi beban tersendiri namun masih bisa *dihandle* oleh pengembang android.
- 6) Product Owner juga memberikan beberapa masukan kepada tim pengembang terkait dengan aplikasi android seperti desain login dan register karena melihat dari persepektif para pengguna android generasi X
- 7) Sesi kedua, anggota pengembang mendemokan flow database. Menceritakan proses pengerjaan API dan penjelasan server yang digunakan.
- 8) Beberapa halangan juga diceritakan oleh pengembang backend seperti permasalahan dalam integrase API untuk login ke halaman akun driver dan halaman akun user. Hal ini bisa diatasi dengan menanyakan permasalahan integrase API dengan membaca dokumentasi NodeJS.
- 9) Sesi Ketiga, Product Owner menyampaikan goal apa saja yang akan dicapai pada Sprint 2 dan peyampaian beberapa bug yang didapatkan pada sprint 1. Beberapa bug yang ditemukan Product Owner misalnya ada kode verifikasi nomor telepon dan penggunaan layanan OTP yang akan digunakan guna verifikasi

d. Hasil Sprint Retropektif

- 1) Scrum master memfasilitasi tim pengembang, dan Product Owner untuk melakukan retropektif dimana proses ini scrum team menginpeksi dirinya sendiri dan membuat perencanaan peningkatan
- 2) Tim Pengembang membuat retropektif diri dan menghasilkan bahwa Komunikasi antar team untuk lebih intens. Menggunakan Slack untuk komunikasi yang lebih baik dan progress bisa dilakukan setiap sore guna mengetahui kejadian kejadian yang terjadi dihari tersebut dan apa yang terjadi esok hari

4.2.3 Sprint 2

a. Product Backlog Item

PBI pada sprint 2 disusun berdasarkan hasil Inception sebelumnya dan apabila pada sprint 1 ada penambahan, maka akan dikerjakan pada sprint 2. Bila tidak ada perubahan maka Sprint Backlog akan mengacu pada Product Backlog Item.

Tabel 4.6 *Product Backlog Item Sprint 2*

No.	Product Backlog Item
1.	Fitur Memilih Deride
2.	Fitur Memilih Lokasi Deride
3.	Fitur Memilih Lokasi Tujuan Deride
4.	Fitur Maps
5.	Fitur OTP sms

Penyusunan kembali Product Backlog Item lebih diperinci lagi berdasarkan tugas masing-masing tim pengembang untuk disesuaikan

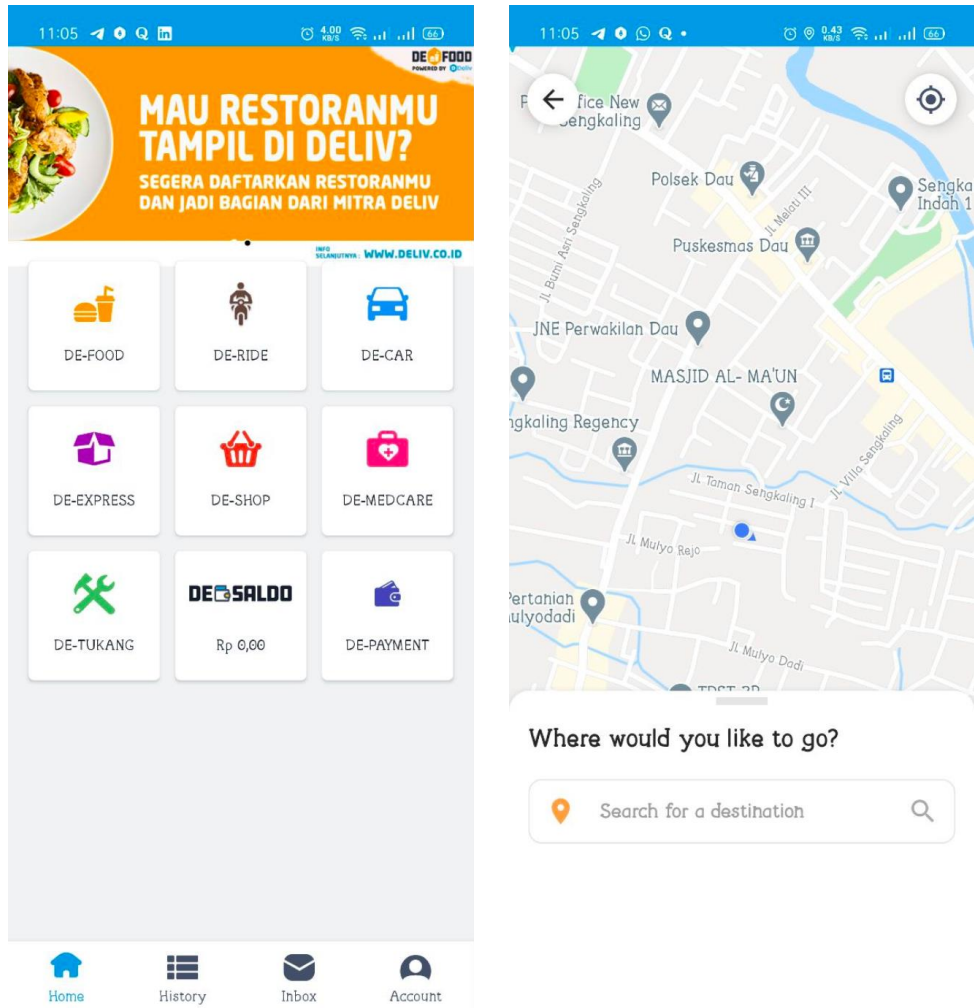
dengan kondisi pekerjaan yang sedang dikerjakan dan penyesuaian dengan Sprint 1.

Tabel 4.7 *Sprint Backlog* Item

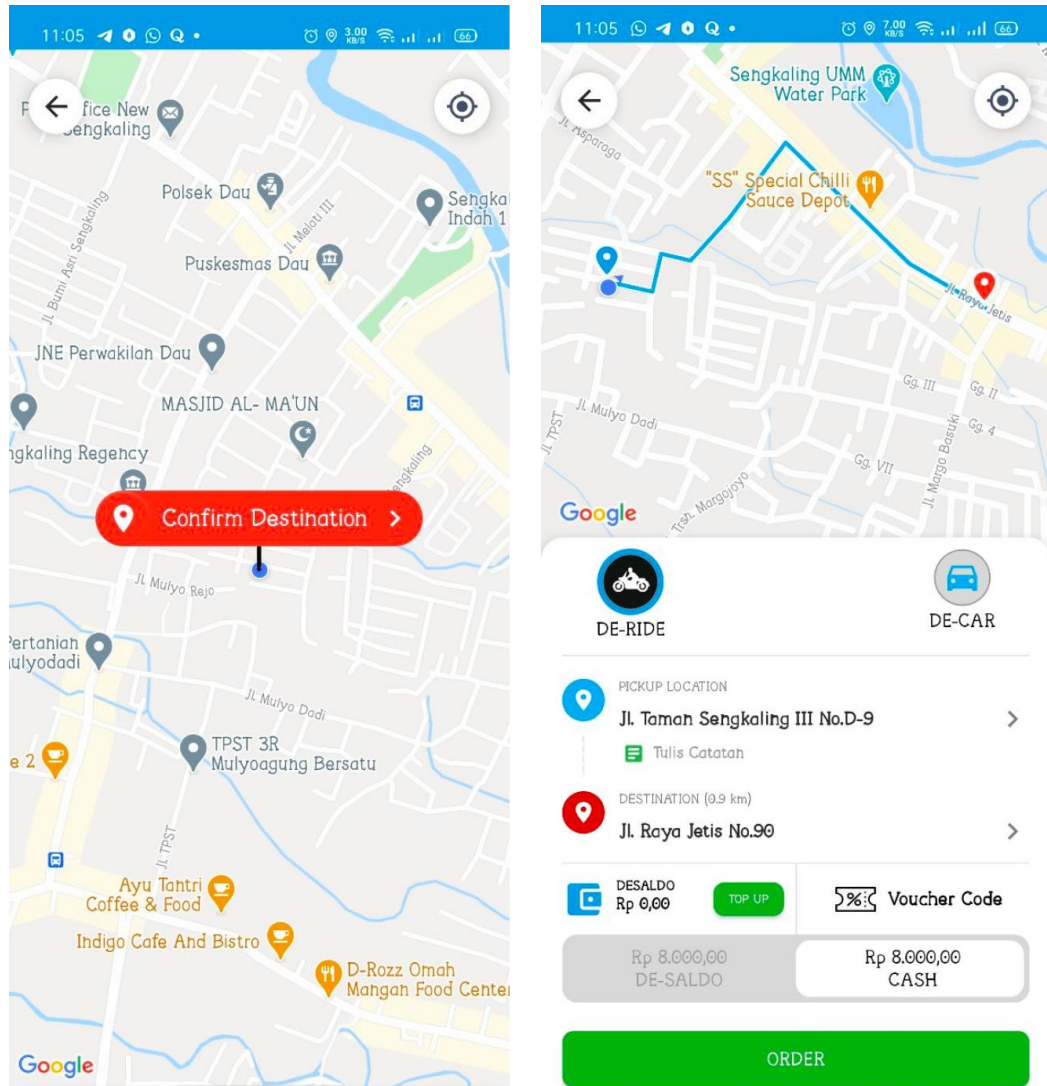
No.	Sprint Backlog Item
1.	Design Screen Pemesanan De Ride
2.	API dan Database Pemesanan De Ride
3.	Mendaftar dan mengatur Google Maps
4.	Integrasi Google Maps dan De Ride
5.	Screen Android Pemilihan Penjemputan Deride
6.	Screen Android Pemilihan Tujuan Deride
7.	Testing Android Konsumen
8.	Pemilihan dan pencarian pihak ketiga OTP
9.	Testing penggunaan OTP

b. Hasil Sprint

Hasil sprint merupakan hasil dari proses Sprint yang telah berjalan. Hasilnya berupa produk menjadi sebuah aplikasi Android yang terdapat beberapa screen sesuai dengan Product Backlog Item yang telah didefinisikan. Beberapa screen yang telah diselesaikan pada Sprint 2 yaitu Screen Android Home, Lokasi Tujuan, Pencarian Lokasi Tujuan Konsumen dan Pemesanan Driver Konsumen



Gambar 4.2 Screen Android Home, dan Pencarian Lokasi Tujuan Konsumen



Gambar 4.3 Screen Android Lokasi Tujuan dan Pemesanan Driver Konsumen

c. Hasil Sprint Review

- 1) Scrum Master sebagai fasilitator membuka sprint review dengan salam dan sapaan kepada tim pengembang dan produk owner
- 2) Scrum Master menjelaskan sprint goal dan gambaran umum Produk Backlog Item yang dikerjakan pada sprint kedua

- 3) Demo peningkatan produk perangkat lunak dimulai. Ada dua hal yang dipresentasikan oleh tim pengembang, yaitu aplikasi android Driver dan database yang telah dibuat.
- 4) Sesi pertama anggota pengembang android mendemokan aplikasi. Menceritakan aplikasi yang telah dibuat mulai Screen pemesanan De Ride, API dan database De Ride, Penggunaan OTP untuk pendaftaran pada aplikasi android. Pada sesi ini, pengembang juga menceritakan permasalahan permasalahan yang dihadapi saat proses pengembangan aplikasi android.
- 5) Beberapa halangan dalam pengembangan aplikasi Android misalnya penyiapan akun untuk pendaftaran di Google Maps memerlukan kartu kredit dan sementara menggunakan kartu kredit oleh pengembang android sendiri. Pemilihan OTP untuk pihak ketiga akhirnya jatuh pada OneSignal mengharuskan untuk mempelajari API dari OneSignal yang dokumentasinya sulit untuk dipahami.
- 6) Product Owner juga memberikan beberapa masukan kepada tim pengembang terkait dengan aplikasi android seperti desain screen android, Kode OTP yang kadang kala tidak masuk / sempat terlambat.
- 7) Sesi kedua, anggota pengembang backend mempresentasikan database dan API untuk pencarian driver, pemilihan lokasi penjemputan dan tujuan untuk fitur De Ride.
- 8) Permasalahan yang didapatkan tim pengembang adalah permasalahan dalam pencarian driver yang sulit diimplementasikan,

karena menggunakan unqi id untuk bisa menemukan driver berdasarkan jarak terdekat. Namun, berhasil diatasi dengan menggunakan range dari GPS pengguna.

- 9) Sesi Ketiga, Product Owner menyampaikan goal apa saja yang akan dicapai pada Sprint 3 dan peyampaian beberapa bug yang didapatkan pada sprint 2. Beberapa bug yang ditemukan Product Owner misalnya kode OTP saat register pada halaman register customer Deliv, kode tersebut tidak masuk. Selain hal tersebut, tidak ada permasalahan yang ditemukan.

d. Hasil Sprint Retropektif

- 1) Pembuatan akun untuk pendaftaran Google Maps dan penyiapan kartu kredit dengan bank tertentu atas nama perusahaan Deliv Indonesia
- 2) Pembuatan akun OneSignal dengan menggunakan akun pribadi. Untuk sementara waktu, hal ini dilakukan untuk akun uji coba atau akun pengembang.
- 3) Para tim pengembang masih belum terbiasa untuk menggunakan Trello sebagai laporan atas pekerjaan yang telah dilakukan sehingga masih perlu beradaptasi dan diingatkan untuk membuat laporannya.
- 4) Karena pada proses pengerjaan remote, maka pertemuan dilakukan setiap hari Senin dan Jumat untuk komunikasi lebih intens.
- 5) Apabila pengerjaan remote dirasa kurang menguntungkan, maka bisa dilakukan dengan bekerja *on site*

4.2.4 Sprint 3

a. Product Backlog Item

Product Backlog Item sprint 3 mengalami perubahan berdasarkan Sprint Review yang terjadi pada sprint 2, sehingga PBI menjadi sebagai berikut

Tabel 4.8 *Product Backlog Item Sprint 3*

No.	Product Backlog Item
1.	Fitur Konsumen Mencari Driver
2.	Fitur Driver Menerima Order
3.	Fitur Notifikasi Orderan

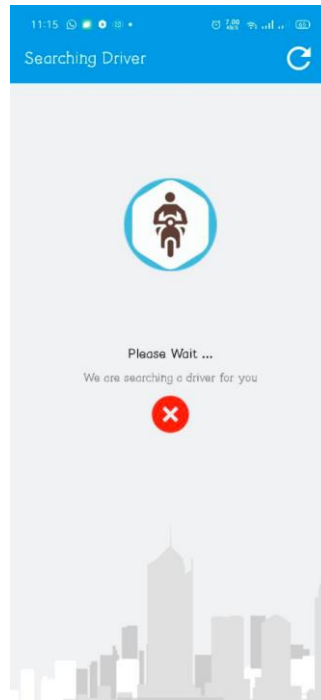
Penyusunan kembali PBI lebih diperinci lagi berdasarkan tugas masing-masing tim pengembang untuk disesuaikan dengan kondisi pekerjaan yang sedang dikerjakan.

Tabel 4.9 *Sprint Backlog Item Sprint 3*

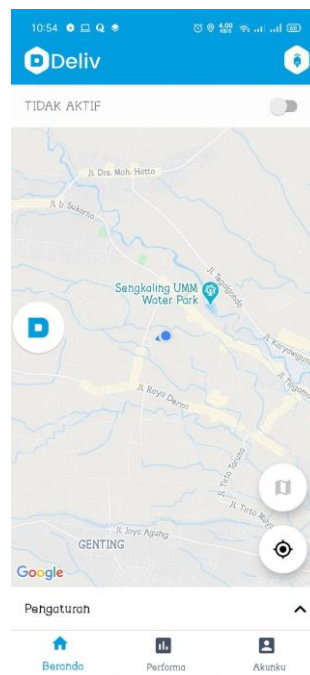
No.	Sprint Backlog Item
1.	Design Screen Konsumen Mencari Driver
2.	Design Screen Home Driver
3.	Design Screen Driver menerima order
4.	API dan Database Pencarian Driver
5.	API dan Database Driver
6.	Mendaftar dan Mengatur Push Notifikasi
7.	Integrasi Push Notifikasi

b. Hasil Sprint

Hasil sprint merupakan hasil dari proses Sprint yang telah berjalan. Hasilnya berupa produk menjadi sebuah aplikasi Android yang terdapat beberapa screen sesuai dengan Product Backlog Item yang telah didefinisikan. Beberapa screen yang telah diselesaikan pada Sprint 3 yaitu Screen Android Pencarian Driver pada Konsumen dan Screen Android Home Aplikasi Driver.



Gambar 4.4 Screen Android Home Aplikasi Driver



Gambar 4.5 Screen Android Home Aplikasi Driver

c. Hasil Sprint Review

- 1) Scrum Master sebagai fasilitator membuka sprint review dengan salam dan sapaan kepada tim pengembang dan produk owner
- 2) Scrum Master menjelaskan sprint goal dan gambaran umum Produk Backlog Item yang dikerjakan pada sprint kedua
- 3) Demo peningkatan produk perangkat lunak dimulai. Ada dua hal yang dipresentasikan oleh tim pengembang, yaitu aplikasi android Driver dan database yang telah dibuat.
- 4) Sesi pertama anggota pengembang android mendemokan aplikasi. Menceritakan aplikasi yang telah dibuat mulai dari Aplikasi Deliv yang bisa digunakan mulai dari Login, masuk ke aplikasi Deliv dan melakukan pemesanan pada fitur De Ride dan pada aplikasi Driver Deliv, bisa menerima pemesanan dan mendapatkan notifikasi pemesanan.
- 5) Pada Sprint ini, pengembang Android tidak mendapatkan halangan berarti pada proses pengembangan. Hanya permasalahan pada kode untuk pencarian Driver.
- 6) Product Owner memberikan apresiasi atas pekerjaan pengembang Android yang telah menyelesaikan tugas secara baik. Namun, saat test berlangsung Product Owner mendapatkan bug notifikasi tidak masuk ke HP driver.
- 7) Sesi kedua, para anggota pengembang backend mempresentasikan database dan API yang telah digunakan oleh pengembang Android,

yaitu proses pencarian driver, pendaftaran driver hingga loginnya serta integrasi dengan API OneSignal.

- 8) Pada Sprint kali ini, pengembang backend tidak begitu banyak mengalami hambatan yang berarti.
- 9) Sesi Ketiga, Product Owner menyampaikan goal apa saja yang akan dicapai pada Sprint 4 dan pemberian apresiasi kepada para pengembang android dan backend yang telah menyelesaikan tugas dengan baik.

d. Hasil Sprint Retropektif

- 1) Apresiasi untuk para pengembang karena sudah menyelesaikan tugas sesuai dengan waktu yang telah diharapkan
- 2) Peningkatan kinerja para pengembang dalam melaksanakan tugasnya
- 3) Penambahan fitur agar progress driver bisa terlihat pada aplikasi pengguna, merubah tampilan font pada driver agar bisa lebih terlihat, dan merubah beberapa font disesuaikan dengan pengguna generasi tua.
- 4) Mencari tahu penyebab notifikasi tidak sampai pada driver.

4.2.5 Sprint 4

a. Product Backlog Item

Product Backlog Item sprint 4 tidak mengalami perubahan, sehingga PBI menjadi sebagai berikut :

Tabel 4.10 *Product Backlog Item Sprint 4*

No.	Product Backlog Item
1.	Fitur Driver Menolak Order
2.	Fitur Memilih Deexpress
3.	Fitur Memilih Lokasi Penjemputan Deexpress
4.	Memperbaiki Fitur Notifikasi
5.	Memperbaiki Bug pada Aplikasi Deliv

Penyusunan kembali PBI lebih diperinci lagi berdasarkan tugas masing-masing tim pengembang untuk disesuaikan dengan kondisi pekerjaan yang sedang dikerjakan.

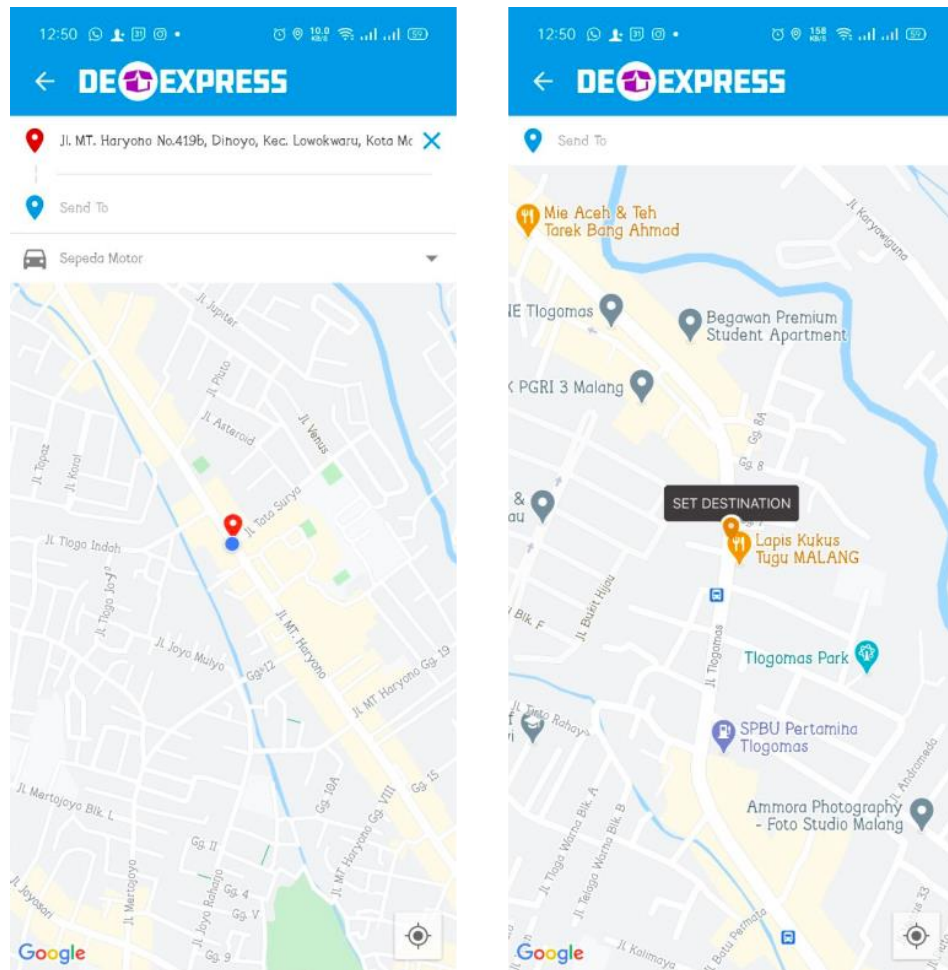
Tabel 4.11 *Sprint Backlog Item pada Sprint 4*

No.	Sprint Backlog Item
1.	Design Screen De Express
2.	Design Screen Driver menerima order
3.	Perbaiki Tampilan Font dan desain button
4.	API dan Database De Express
5.	API dan Database Driver
6.	Perbaiki Galat Push Notifikasi
7.	Integrasi Push Notifikasi

b. Hasil Sprint

Hasil sprint merupakan hasil dari proses Sprint yang telah berjalan. Hasilnya berupa produk menjadi sebuah aplikasi Android yang terdapat beberapa screen sesuai dengan Product Backlog Item yang telah didefinisikan. Beberapa screen yang telah diselesaikan pada *Sprint 4* adalah Screen Android De Express dan Lokasi Penjemputan pada

Konsumen.



Gambar 4.6 Screen Android De Express dan Lokasi Penjemputan pada Konsumen

c. Hasil Sprint Review

- 1) Scrum Master sebagai fasilitator membuka sprint review dengan salam dan sapaan kepada tim pengembang dan produk owner
- 2) Scrum Master menjelaskan sprint goal dan gambaran umum Produk Backlog Item yang dikerjakan pada sprint keempat
- 3) Demo peningkatan produk perangkat lunak dimulai. Ada dua hal yang dipresentasikan oleh tim pengembang, yaitu aplikasi android dan API yang telah dibuat.

- 4) Sesi pertama anggota pengembang mendemokan aplikasi. Mendemonstrasikan proses bagaimana konsumen bisa melakukan pendaftara, login dan melakukan pemesanan untuk bisa menggunakan fitur De Express dan Driver bisa mendapatkan notifikasi orderan De Express.
- 5) Beberapa halangan dalam pengembangan aplikasi Android masih seputar notifikasi dimana pada prosesnya, ada notifikasi yang tidak masuk kedalam smartphone driver.
- 6) Product Owner memberikan masukan kepada pengembang Android bahwa selain permasalahan notifikasi, pada proses testing pada Android versi 9 keatas dan Smartphone dengan dimensi 16:9, terdapat beberapa desain yang tidak proporsional pada Screen Login, Register, dan Screen Home.
- 7) Sesi kedua, anggota pengembang Backend mempresentasikan API De Express dan API Driver untuk bisa digunakan oleh pengembang Android agar bisa digunakan serta memperbaiki kode sebelumnya agar lebih rapi.
- 8) Beberapa halangan juga diceritakan oleh pengembang backend seperti dokumentasi aplikasi yang belum ada sehingga cukup menyusahkan dalam pengembangan aplikasi Deliv.
- 9) Sesi Ketiga, Lead Team, memberikan arahan akan adanya beberapa fitur baru terkait dengan aplikasi untuk dikerjakan pada Sprint 5 yaitu seperti Fitur Promo yang nantinya bisadigunakan pada De Ride dan De Express untuk bisa memberikan promosi kepada para

kosumen. Selain itu Lead Team juga memberikan atau merangkum semua bug yang ada.

d. Hasil Sprint Retropektif

- 1) Dokumentasi pada proses pengembangan masih terkesan kurang seperti yang dikatakan para pengembang Backend sehingga perlu adanya dokumentasi API yang ada pada Deliv
- 2) Para pengembang merasakan kurangnya leadership sehingga koordinasi masih terasa kurang
- 3) Kurangnya komunikasi masih terasa pada para pengembang sehingga memutuskan untuk setiap Senin dan Jumat untuk bekerja dari Kantor dan tidak melakukan Remote Working.

4.2.6 Sprint 5

a. Product Backlog Item

PBI pada sprint 2 disusun berdasarkan hasil Inception sebelumnya dan apabila pada sprint 4 ada penambahan, maka akan dikerjakan pada sprint 5. Bila tidak ada perubahan maka Sprint Backlog akan mengacu pada Product Backlog Item.

Tabel 4.12 *Product Backlog Item Sprint 5*

No.	Product Backlog Item
1.	Fitur Memilih Lokasi Pengiriman De Express
2.	Fitur Merubah Profile Konsumen
3.	Fitur Pengiriman De Express
4.	Fitur Pemberian Rating dan Review
5.	Fitur Cek Rating dan Review pada Driver

Penyusunan kembali Product Backlog Item lebih diperinci lagi berdasarkan tugas masing-masing tim pengembang untuk disesuaikan

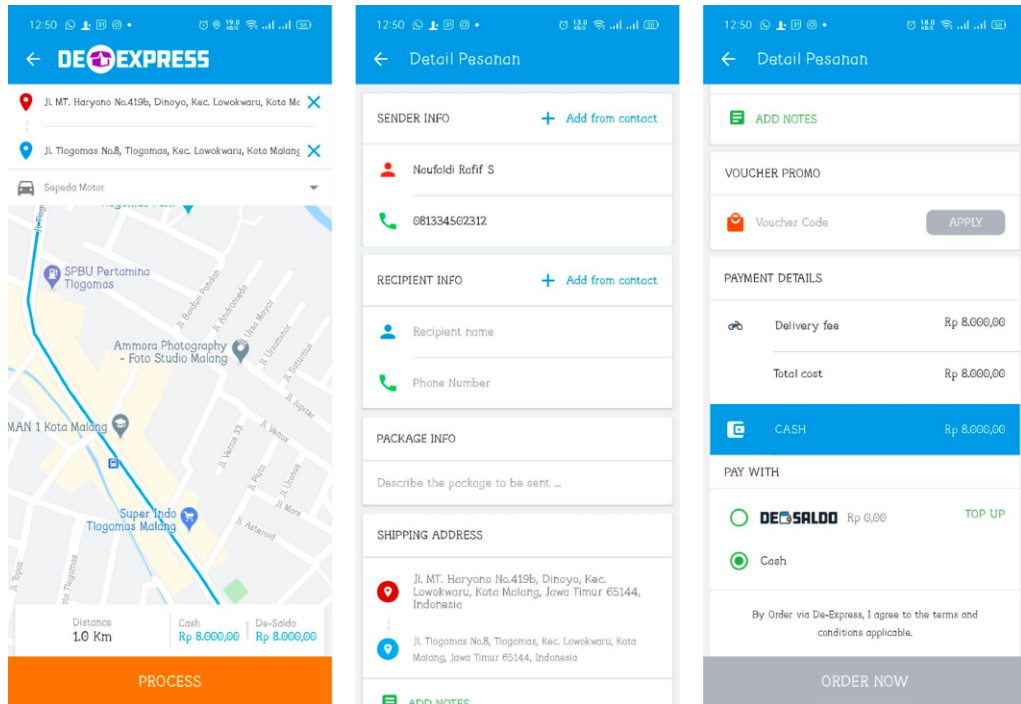
dengan kondisi pekerjaan yang sedang dikerjakan dan penyesuaian dengan Sprint 4.

Tabel 4.13 *Sprint Backlog Item pada Sprint 5*

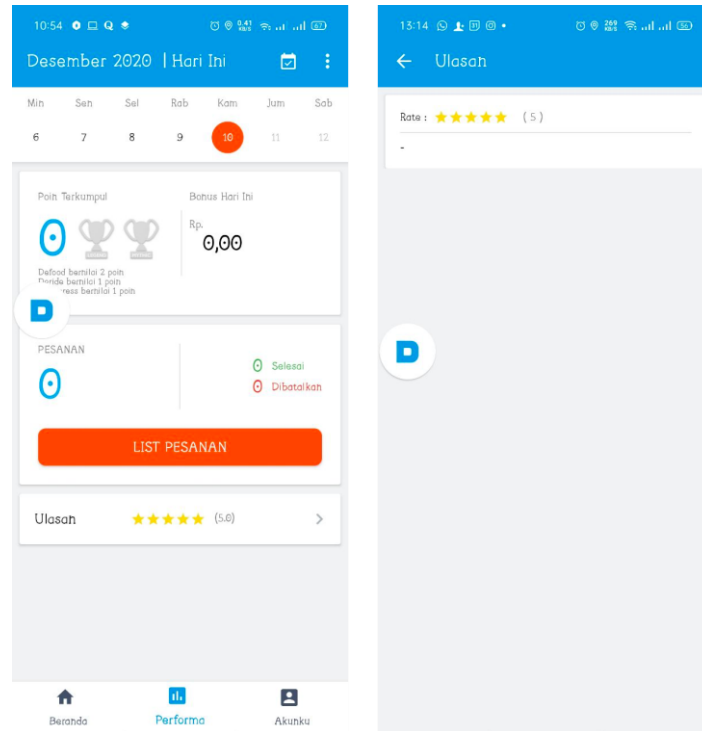
No.	Sprint Backlog Item
1.	Design Screen Lokasi Pengiriman De Express
2.	API dan Database Lokasi Pengiriman De Express
3.	Screen Android Lokasi Pengiriman De Express
4.	Testing Android Lokasi Pengiriman De Express
5.	Design Screen Pengiriman De Express
6.	API dan Database Pengiriman De Express
7.	Screen Android Pengiriman De Express
8.	Testing Android Pengiriman De Express
9.	Design Screen Merubah Profile Konsumen
10.	API dan Database Merubah Profile Konsumen
11.	Screen Android Merubah Profile Konsumen
12.	Testing Android Merubah Profile Konsumen
13.	Design Screen Pemberian Rating dan Review
14.	API dan Database Pemberian Rating dan Review
15.	Screen Android Pemberian Rating dan Review
16.	Testing Android Pemberian Rating dan Review
17.	Design Screen Cek Rating dan Review pada Driver
18.	API dan Database Cek Rating dan Review pada Driver
19.	Screen Android Cek Rating dan Review pada Driver
20.	Testing Android Cek Rating dan Review pada Driver

b. Hasil Sprint

Hasil sprint merupakan hasil dari proses Sprint yang telah berjalan. Hasilnya berupa produk menjadi sebuah aplikasi Android yang terdapat beberapa screen sesuai dengan Product Backlog Item yang telah didefinisikan. Beberapa screen yang telah diselesaikan pada *Sprint 5* yaitu Screen Android Memilih Lokasi Pengirim dan Identitas Barang Konsumen serta Screen Android Melihat Review dan Rating Driver.



Gambar 4.7 *Screen Android* Memilih Lokasi Pengirim dan Identitas Barang Konsumen



Gambar 4.8 Screen Android Melihat Review dan Rating Driver

c. Hasil Sprint Review

- 1) Scrum Master sebagai fasilitator membuka sprint review dengan salam dan sapaan kepada tim pengembang dan produk owner
- 2) Scrum Master menjelaskan sprint goal dan gambaran umum Produk Backlog Item yang dikerjakan pada sprint kelima
- 3) Demo peningkatan produk perangkat lunak dimulai. Ada dua hal yang dipresentasikan oleh tim pengembang, yaitu aplikasi android dan API yang telah dibuat.
- 4) Sesi pertama anggota pengembang mendemokan aplikasi. Mendemonstrasikan proses konsumen bisa melakukan pengiriman barang menggunakan De Express, salah satunya pemilihan lokasi pengiriman barang, konsumen mampu untuk merubah profile,

memberikan rating dan review pada driver dan driver mampu untuk mengecek rating dan review di aplikasi.

- 5) Product Owner memberikan masukan kepada pengembang Android bahwa aplikasi sudah berjalan dengan cukup baik dan mampu menyesuaikan dengan sprint goal yang dikerjakan.
- 6) Sesi kedua, anggota pengembang Backend mempresentasikan API De Express, API Konsumen dan API Driver sehingga bisa digunakan oleh pengembang Android. Sejauh ini tidak ada halangan yang terjadi saat proses pengembangan API.
- 7) Sesi Ketiga, Lead Team, memberikan apresiasi atas kinerja tim pengembang karena berhasil menyelesaikan product backlog item dengan cukup baik tanpa ada halangan sama sekali.

d. Hasil Sprint Retropektif

- 1) Tim pengembang diharapkan mampu menjaga kinerja dalam proses pengembangan aplikasi Deliv
- 2) Komunikasi antara tim pengembang agar selalu dijaga agar memudahkan dalam komunikasi antara sesama tim pengembang dan produk owner.
- 3) Product Owner , berharap scrum master untuk terus memfasilitasi tim pengembang dalam mengembangkan product Deliv.

4.2.7 Sprint 6

a. Product Backlog Item

PBI pada sprint 2 disusun berdasarkan hasil Inception sebelumnya dan apabila pada sprint 1 ada penambahan, maka akan dikerjakan pada

sprint 2. Bila tidak ada perubahan maka Sprint Backlog akan mengacu pada Product Backlog Item.

Tabel 4.14 *Product Backlog Item Sprint 6*

No.	Product Backlog Item
1.	Fitur Logout Konsumen
2.	Fitur Logout Driver
3.	Screen Term of Service
4.	Screen Privacy Policy

Penyusunan kembali Product Backlog Item lebih diperinci lagi berdasarkan tugas masing-masing tim pengembang untuk disesuaikan dengan kondisi pekerjaan yang sedang dikerjakan dan penyesuaian dengan Sprint 5.

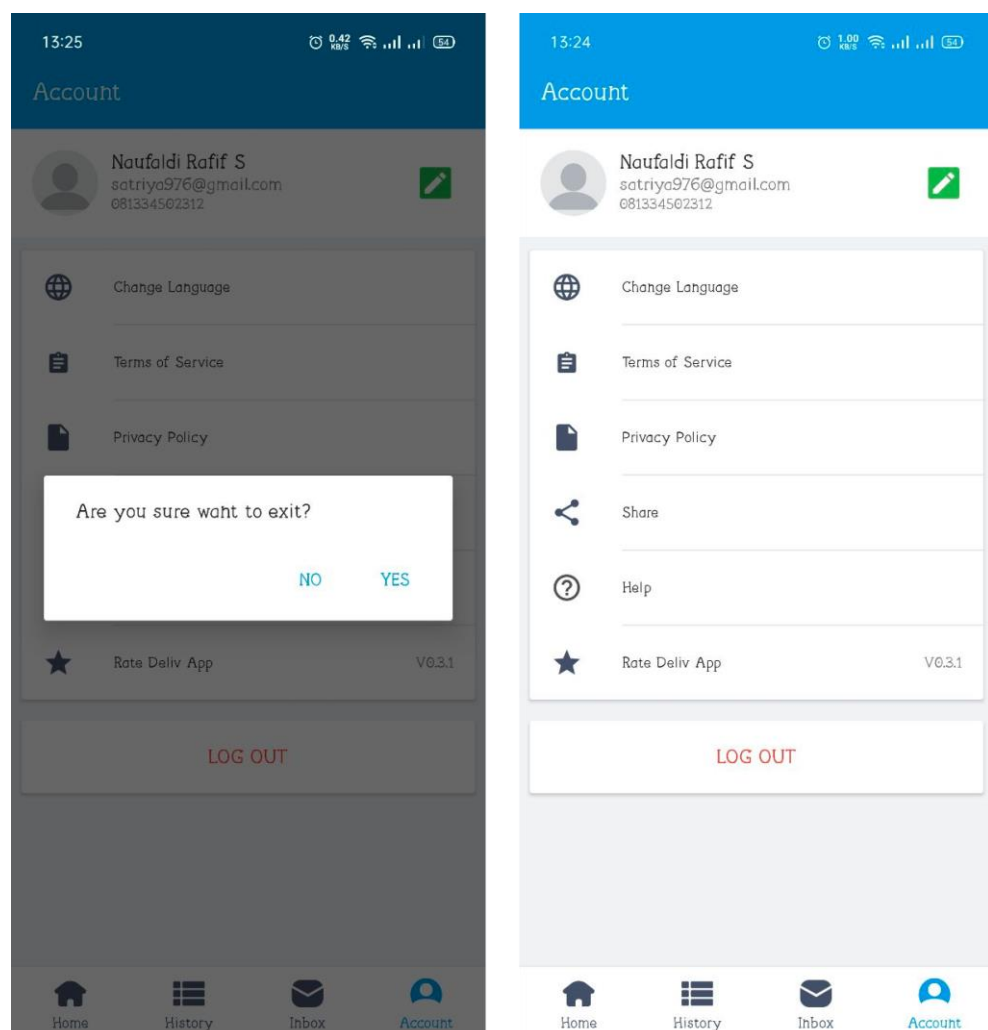
Tabel 4.15 *Sprint Backlog Item pada Sprint 6*

No.	Sprint Backlog Item
1.	Design Screen Logout Konsumen
2.	API dan Database Logout Konsumen
3.	Screen Android Logout Konsumen
4.	Testing Android Logout Konsumen
5.	Design Screen Logout Driver
6.	API dan Database Logout Driver
7.	Screen Android Logout Driver
8.	Testing Android Logout Driver
9.	Design Screen Term of Service
10.	API dan Database Term of Service
11.	Screen Android Term of Service
12.	Testing Android Term of Service
13.	Design Screen Privacy Policy
14.	API dan Database Privacy Policy
15.	Screen Android Privacy Policy
16.	Testing Android Privacy Policy

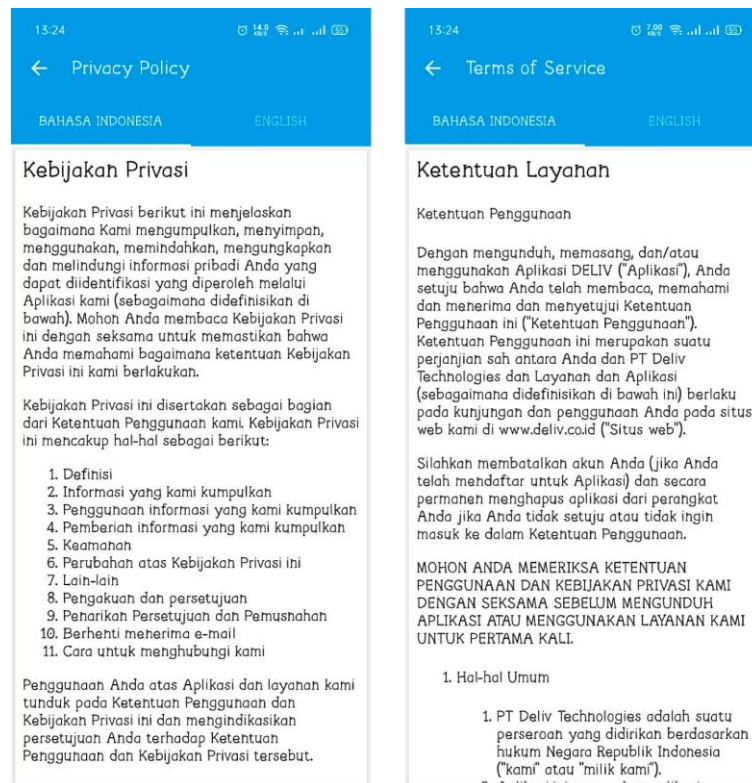
b. Hasil Sprint

Hasil sprint merupakan hasil dari proses Sprint yang telah berjalan.

Hasilnya berupa produk menjadi sebuah aplikasi Android yang terdapat beberapa screen sesuai dengan Product Backlog Item yang telah didefinisikan. Beberapa screen yang telah diselesaikan pada *Sprint65* yaitu Screen Android Menu, Logout Konsumen, Screen Menu Profile dan Fitur Logout Driver



Gambar 4.9 Screen Android Menu, Logout Konsumen

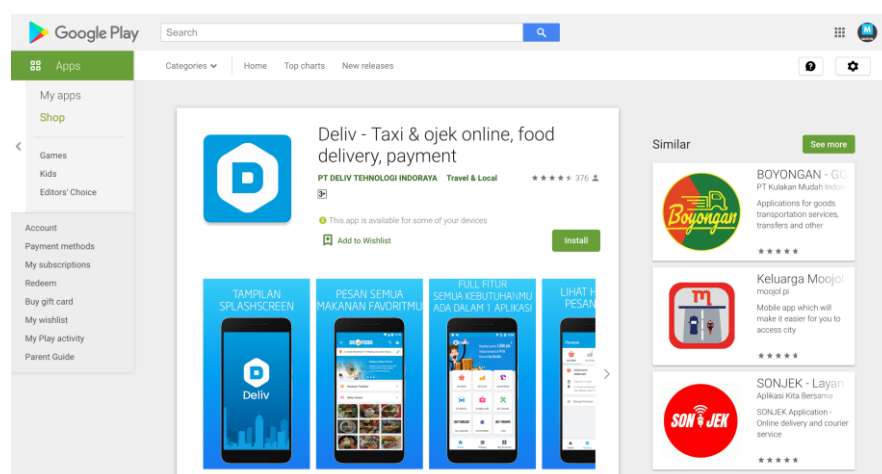


Gambar 4.10 Screen Kebijakan Privasi dan Ketentuan Layanan

c. Hasil Sprint Review

- 1) Scrum Master sebagai fasilitator membuka sprint review dengan salam dan sapaan kepada tim pengembang dan produk owner
- 2) Scrum Master menjelaskan sprint goal dan gambaran umum Produk Backlog Item yang dikerjakan pada sprint keenam dan terakhir untuk produk bisa mengeluarkan MVP.
- 3) Demo peningkatan produk perangkat lunak dimulai. Ada dua hal yang dipresentasikan oleh tim pengembang, yaitu aplikasi android dan API yang telah dibuat.

- 4) Sesi pertama anggota pengembang mendemonstrasikan proses konsumen bisa melakukan logout pada aplikasi Deliv, driver mampu melakukan logout pada aplikasi driver Deliv dan user mampu untuk melihat Term of Service serta Privacy Policy.
- 5) Product Owner memberikan masukan kepada tim pengembang bahwa aplikasi sudah berjalan dengan cukup baik dan mampu menyesuaikan dengan sprint goal yang dikerjakan.
- 6) Sesi kedua, Lead Team, memberikan apresiasi atas kinerja tim pengembang karena berhasil menyelesaikan product backlog item dengan cukup baik tanpa ada halangan sama sekali dan membuat produk bisa diselesaikan dengan baik sehingga bisa untuk diupload pada Google Play.
- 7) Sesi Terakhir, Tim Pengembang melakukan proses pendaftaran dan mengupload hasil pengembangan di Playstore dan bisa diunduh aplikasi Deliv.



Gambar 4.11 Aplikasi Deliv telah diupload di *Playstore*

d. Hasil Sprint Retropektif

- 1) Tim pengembang diharapkan mampu menjaga kinerja dalam proses pengembangan aplikasi Deliv
- 2) Komunikasi antara tim pengembang agar selalu dijaga agar memudahkan dalam komunikasi antara sesama tim pengembang dan produk owner.
- 3) Product Owner , berharap scrum master untuk terus memfasilitasi tim pengembang dalam mengembangkan product Deliv.
- 4) Product Owner, berharap tim pengembang untuk segera mengunggah aplikasi Deliv dan driver Deliv ke Play Store sehingga bisa digunakan oleh user dan driver.
- 5) Product Owner, akan memfasilitasi kebutuhan tim pengembang terkait dengan kebutuhan yang diperlukan akan berhasil melakukan upload pada Google Play Store.

4.3 Pengujian

4.3.1 Pengujian UEQ pada Aplikasi Android Deliv

Pengujian UEQ pada aplikasi Android Deliv dilakukan terhadap 26 responden dengan memberikan skenario dan metode yang sama. Hasil pengujian UEQ terhadap 26 responden yang diisi dari poin 1 hingga 7 pada gambar dibawah ini :

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	6	3	3	1	5	5	5	3	3	5	3	5	5	5	4	3	2	3	5	3	6	3	3	3	5	
5	6	2	3	1	5	5	5	2	2	5	2	6	5	3	6	2	2	3	6	1	6	2	2	1	6	
2	6	6	6	3	5	5	5	5	6	6	3	6	5	2	5	4	6	4	3	2	6	2	6	4	2	
6	5	6	6	1	4	5	5	5	6	6	5	6	5	3	5	2	4	2	6	2	6	6	5	6	4	
6	7	3	3	1	5	6	6	1	2	6	1	6	7	3	6	4	3	2	5	2	5	4	2	1	6	
5	6	4	6	5	4	4	6	5	4	6	5	6	5	4	5	6	4	5	6	5	6	5	5	6	5	
6	5	2	4	2	6	6	6	5	3	5	1	5	6	5	7	3	3	4	5	2	4	2	3	2	6	
6	6	3	2	2	5	7	5	2	3	6	2	6	5	4	5	3	3	3	6	2	6	2	3	2	5	
6	3	6	2	1	3	3	4	2	5	6	2	6	5	2	6	2	6	5	2	1	3	2	5	4	2	
5	5	4	5	4	4	4	6	3	4	5	5	6	5	4	6	5	5	4	3	5	6	5	5	5	3	
5	6	2	2	2	6	6	5	2	4	6	2	7	5	5	6	2	2	2	5	2	6	1	1	1	4	
5	4	6	3	3	3	3	5	3	5	4	3	5	4	3	4	3	4	3	4	3	5	3	4	4	3	
6	6	2	2	2	4	6	6	2	4	6	1	6	6	4	6	2	3	2	6	2	6	2	2	2	4	
4	6	5	2	2	5	5	6	2	2	6	5	6	5	4	5	5	3	5	6	2	6	2	2	2	4	
6	6	6	3	3	4	5	5	4	6	5	4	6	5	2	6	4	4	4	5	3	5	4	5	3	4	
5	6	6	2	2	5	4	5	3	6	5	3	7	5	3	5	4	4	3	6	1	6	2	3	2	3	
7	7	1	1	1	7	7	4	1	4	7	1	7	7	7	7	2	3	2	7	1	7	4	1	1	7	
5	7	2	1	1	6	6	7	2	1	6	3	7	6	7	6	1	2	2	5	1	6	6	2	1	6	
6	7	3	2	2	4	5	6	1	3	5	2	7	6	4	6	2	3	3	7	2	7	2	3	2	5	
6	7	2	1	2	6	6	6	2	2	6	2	6	6	5	6	2	2	2	5	1	5	2	5	2	6	
5	5	5	4	6	5	5	5	3	5	5	5	5	4	3	4	4	4	4	5	4	4	4	5	3	4	
5	5	3	4	3	5	5	4	4	4	5	4	5	5	4	5	3	3	3	4	3	5	4	4	3	5	
5	6	1	1	1	6	7	6	2	2	6	2	5	6	3	6	2	2	2	6	1	7	1	7	7	1	

Gambar 4.12 Hasil Kuisisioner UEQ

Hasil penilaian yang dilakukan oleh responden diolah menjadi rentan bobot -3 sampai +3 untuk jawaban negatif menuju positif. Sedangkan rentan bobot +3 sampai -3 untuk jawaban positif menuju negatif dengan bobot 0 apabila netral.

Scale means per person						
Daya tarik	Kejelasan	Efisiensi	Ketepatan	Stimulasi	Kebaruan	
1,17	1,25	1,25	1,00	1,75	1,00	
1,83	2,00	2,00	1,25	1,75	1,25	
-0,17	1,00	0,50	0,75	0,25	-2,00	
0,00	0,75	0,25	1,75	1,00	-1,25	
2,50	2,00	1,25	1,50	1,75	1,00	
-0,17	0,00	0,00	0,25	-0,25	0,25	
2,17	1,00	0,50	1,00	1,75	1,50	
1,50	2,00	2,00	1,25	1,75	0,75	
1,00	1,50	0,25	0,75	-0,25	-1,75	
0,17	0,25	0,25	0,50	-0,25	-0,25	
2,00	2,25	2,00	1,75	2,00	0,75	
0,33	0,75	0,75	0,75	-0,25	-1,25	
2,17	2,00	2,00	2,00	1,25	0,50	
0,83	2,00	2,00	0,50	1,25	0,25	
0,83	1,50	0,50	0,50	0,50	-1,50	
1,17	2,50	1,75	0,75	0,75	-1,50	
3,00	3,00	2,25	1,75	2,50	2,25	
1,83	3,00	0,75	2,50	2,25	2,50	
1,83	2,50	2,75	1,50	1,00	0,75	
1,50	2,75	1,50	2,00	2,00	1,75	
0,00	0,50	0,50	0,50	0,00	-0,75	
0,67	0,75	0,25	0,75	1,00	0,50	
0,17	2,25	2,50	2,00	2,50	0,25	

Gambar 4.13 Hasil Rata-rata Penilaian Setiap Responden

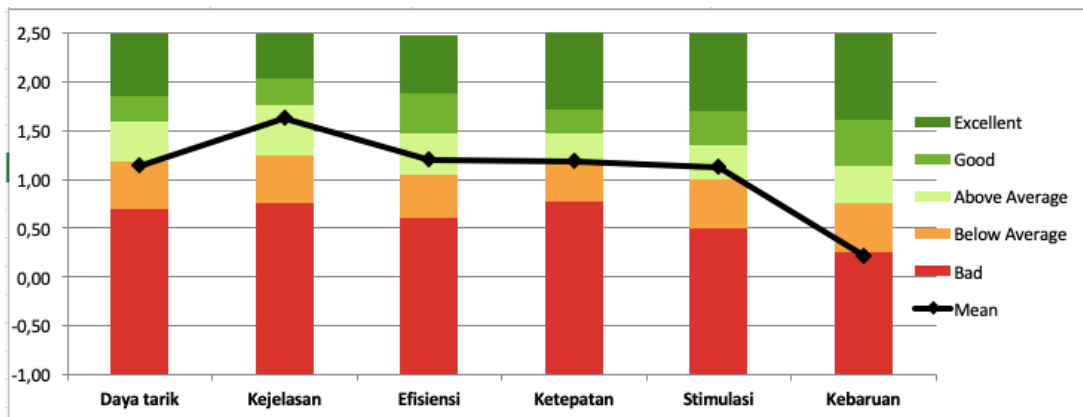
Berdasarkan Tabel pada hasil rata-rata penilaian setiap responden, didapatkan data berupa rata-rata setiap grup penilaian mendapatkan nilai diatas rata-rata kecuali kebaruan. Pada grup Stimulasi mendapatkan nilai rata-rata 1,145 dengan jumlah 4 item penilaian, untuk grup penilaian Ketepatan mendapatkan nilai rata-rata 1,185 dengan jumlah 4 item penilaian, untuk grup penilaian Efisiensi mendapatkan nilai rata-rata 1,207 dengan jumlah 4 item penilaian, untuk grup penilaian Kejelasan mendapatkan nilai rata-rata 1,630 dengan jumlah 4 item penilaian, untuk grup penilaian Daya Tarik mendapatkan nilai rata-rata 1,145 dengan jumlah 6 item penilaian dan grup penilaian terakhir, Kebaruan mendapatkan nilai rata-rata 0,217 dengan jumlah 4 item penilaian.

Tabel 4.16 Hasil Penilaian Setiap Item Deliv Apps

No.	Grup Penilaian	Item Penilaian	Rata-rata Penilaian Item	Rata-rata Penilaian Grup
1.	Daya Tarik	Menyusahkan - Menyenangkan	1,4	1,15
		Baik - Buruk	1,1	
		Tidak disukai - Menggembirakan	1,3	
		Tidak nyaman - Nyaman	1,5	
		Atraktif - Tidak atraktif	0,4	
		Ramah pengguna - Tidak ramah pengguna	1,1	
2.	Kejelasan	Tak dapat dipahami - Dapat dipahami	1,8	1,63
		Mudah dipelajari - Sulit dipelajari	1,0	
		Rumit - Sederhana	2,0	
		Jelas - Membingungkan	1,7	
3.	Efisiensi	Cepat - Lambat	1,2	1,21
		Tidak efisien - Efisien	1,1	
		Tidak praktis - Praktis	1,6	
		Terorganisasi - Berantakan	1,0	
4.	Ketepatan	Tak dapat diprediksi – dapat diprediksi	1,3	1,19
		Menghalangi - Mendukung	1,6	
		Aman - Tidak aman	1,0	

No.	Grup Penilaian	Item Penilaian	Rata-rata Penilaian Item	Rata-rata Penilaian Grup
		Memenuhi ekspektasi - Tidak memenuhi ekspektasi	0,9	
5.	Stimulasi	Bermanfaat - Kurang bermanfaat	1,8	1,13
		Membosankan - Mengasyikkan	0,9	
		Tidak menarik - Menarik	1,2	
		Memotivasi - Tidak memotivasi	0,7	
6.	Kebaruan	Kreatif - Monoton	0,4	0,2
		Berdaya cipta - Konvensional	0,3	
		Lazim - Terdepan	0,1	
		Konservatif - Inovatif	0,3	

Pada gambar 4.13 dijelaskan bahwa hasil benchmark Deliv bila dibandingkan data 20.190 responden dari 452 studi(UEQ Data Analisis Tools Versi 8) maka nilai *Above Average* dilihat dari 4 poin grafik, 1 poin di *Below Average* dan 1 poin di *Bad* dengan detail item grup daya tarik mendapatkan *Below Average*, Item grup Kejelasan, Efisiensi, Ketepatan mendapatkan *Above Average* dan Item grup Kebaruan mendapatkan *Bad*.

Gambar 4.14 Diagram *Benchmark Deliv*

4.3.2 Tanggapan Responden UEQ pada Aplikasi Android Deliv

Responden mencoba menggunakan produk Deliv, pengguna memberikan respon / tanggapan terhadap aplikasi android Deliv yang telah dicoba. Responden memiliki berbagai tanggapan terhadap aplikasi Android Deliv yang telah dikembangkan. Hasil evaluasi dengan UEQ didapatkan bahwa keseluruhan aplikasi Android Deliv. Pada Tabel 4.16 dijelaskan mengenai hasil rangkuman tanggapan responden setelah menggunakan produk Deliv

Tabel 4.17 *Tanggapan Responden pada Aplikasi Android Deliv*

No.	Tanggapan	Kode Responden
1.	Desainnya biasa saja. Mirip dengan tampilan / desain dari Gojek yang lama	R-1,R-3,R-5,R-8,R-10
2.	Aplikasi sudah berjalan dengan baik. Bisa digunakan fitur-fiturnya	R-2,R-3,R-5
3.	Menu navigasi memudahkan untuk berpindah halaman.	R-8,R-9,R-10
4.	Saat pertama kali menggunakan, Deliv memberikan kejelasan tentang fungsi aplikasi jelas.	R-4,R-5
5.	Memberikan notifikasi / pemberitahuan apabila Deliv masih belum beroperasi pada wilayah tertentu. Sehingga saat	R-8,-10

No.	Tanggapan	Kode Responden
	ingin memesan tidak menunggu lama karena ketidak tersediaan <i>service</i> di daerah tersebut.	
6.	Menambahkan keterangan saat login menggunakan nomor 0 atau +62 untuk mendapatkan kode OTP	R-8,R-7
7.	Bisa memperbarui tampilan Deliv agar sesuai dengan trend desain yang terbaru. Sehingga tampilan tidak begitu membosankan.	R-1,R-6
8.	Memberikan keterangan jenis mobilnya pada saat menggunakan De Express, seperti ukuran mobil, beban yang bisa diangkut sehingga bisa tahu berapa barang yang bisa diantar.	R-2, R-4
8.	Memberikan keterangan beban maksimal pada mobil / motor yang akan digunakan untuk mengantarkan barang.	R-3
9.	Memberikan keterangan jumlah penumpang pada mobil sehingga bisa mengetahui jumlah maksimal penumpang di Mobil.	R-10

4.4 Pembahasan Hasil Uji Coba dan Pengujian

Berdasarkan hasil uji coba yang selama 6 sprint, didapatkan bahwa Tim Pengembang, Scrum Master dan Product Owner, mampu menyelesaikan Minimum Viable Product yang telah disepakati bersama sehingga product Perangkat Lunak Deliv dalam bentuk *Mobile Apps Android* telah terselesaikan.

Hasil uji coba yang telah diuji dengan menggunakan metode *User Experience Quistioner* juga telah mendapatkan hasil benchmark bahwa Aplikasi Deliv bila dibandingkan data 20.190 responden dari 452 studi (UEQ Data Analisis Tools Versi 8) maka nilai *Above Average* dilihat dari 4 poin grafik pada gambar 4.13.

Hal ini membuktikan penyelesaian perjanjian yang dilakukan oleh Tim Scrum telah sesuai dengan Surat Ali ‘Imran ayat 76 bahwa seorang muslim diwajibkan dalam menepati janjinya.

بَلَىٰ مَنْ أَوْفَىٰ بِعَهْدِهِ وَاتَّقَىٰ فَإِنَّ اللَّهَ يُحِبُّ الْمُتَّقِينَ

Artinya : “(Bukan demikian), sebenarnya siapa yang menepati janji (yang dibuat)nya dan bertakwa, maka sesungguhnya Allah menyukai orang-orang yang bertakwa. ?”(QS. Ali ‘Imran : 76)

Dalam Quraish Shihab atau Al Mishbah dikatakan “ ... Sesungguhnya orang yang melaksanakan hak orang lain, menepatinya sesuai waktu yang mereka janjikan, dan takut kepada Allah dengan tidak mengurangnya dan menundanya, tentu akan beruntung mendapatkan kecintaan Allah, karena ia bertakwa kepada-Nya(1). (1) Ayat ini mengisyaratkan kewajiban menepati janji. Banyak ayat lain sebelum ayat ini yang juga berbicara tentang kewajiban menepati janji, seperti ayat 27 surat al-Baqarah, misalnya. Umat Islam dituduh tidak memelihara perjanjian.”

Hasil Pengujian yang mendapatkan nilai *Above Average* juga membuktikan bahwa Tim Pengembang Scrum telah melakukan yang terbaik secara tepat, efisien dan tuntas sejalan dengan Hadits yang diriwayatkan oleh Imam Thabrani :

إِنَّ اللَّهَ يُحِبُّ إِذَا عَمِلَ أَحَدُكُمْ الْعَمَلَ أَنْ يُتَقَنَّهُ (رواه الطبران)

Artinya: “Sesungguhnya Allah sangat mencintai orang yang jika melakukan sesuatu pekerjaan, dilakukan secara *Itqan* (tepat, terarah, jelas dan tuntas)”(Marhum Sayyid, 2001: 34).

Artinya, pada proses pengembangan perangkat lunak dengan metode Scrum telah menjalan nilai nilai dan amal perbuatan yang dicintai oleh Allah SWT yang mengatur segala sesua dengan baik, tepat, efisien, dan tuntas berdasarkan syariat dalam ajaran agama islam.

Imam Nawawi (2002: 25) mencantumkan sebuah Hadist yang diriwayatkan oleh Imam Muslim dari Abu Ya'la, Rasulullah SAW bersabda

إِنَّ اللَّهَ كَتَبَ الْإِحْسَانَ عَلَى كُلِّ شَيْءٍ (رواه المسلم)

Artinya : “*Sesungguhnya Allah mewajibkan untuk berlaku ihsan atas segala sesuatu*”(HR. Muslim).

Ibnul ‘Atthar Asy-Syafi’i *rahimahullah* yang makruf dengan sebutan Mukhtashar An-Nawawi—sebagaimana julukan ini disebut oleh Ibnu Katsir—menyatakan tentang hadits Arba’in nomor urut 17 ini, bahwa hadits tersebut termasuk hadits singkat namun sarat makna, juga berisi kaedah pokok dalam agama ini. Hadits tersebut berisi perintah untuk berbuat baik pada diri sendiri. (Lihat *Syarh Al-Arba’in An-Nawawiyah* karya Ibnul ‘Atthar, hlm. 112).

Kata ihsan bermakna melakukan sesuatu secara maksimal dan optimal. Oleh karena itu, dalam segala sesuatu harus dilakukan dengan secara optimal, secara baik, benar dan tuntas. Apalagi dalam proses pengembangan perangkat lunak yang menjadi hajat orang banyak dan proses perencanaan begitu lama. Membuat para *stakeholder* mengalami rasa ketakutan apabila proses pengembangan perangkat lunak menjadi gagal dan tidak sesuai dengan harapan.

Proses pengembangan perangkat lunak dengan menggunakan metode Scrum secara tidak langsung telah menunjukkan bahwa umat muslim bisa dan

mampu untuk mengembangkan perangkat lunak dengan nilai diatas rata-rata dengan cara yang efisien dan efektif sesuai dengan syariat islam.

BAB 5

PENUTUP

5.1 Kesimpulan

Hasil penelitian dari penggunaan Manajemen Pengembangan Perangkat Lunak Scrum pada Pengembangan Aplikasi Deliv dapat ditarik kesimpulan bahwa

1. Berdasarkan hasil pengujian menggunakan User Experience Questionare (UEQ) didapatkan hasil bahwa penggunaan metode pengembangan perangkat lunak Scrum pada Aplikasi Deliv mampu memberikan nilai diatas rata-rata / *Above Average* yaitu 1,207 untuk efisiensi dibandingkan data 20.190 responden dari 452 studi(UEQ Data Analisis Tools Versi 8).
2. Aplikasi Deliv mampu diselesaikan dalam kurun waktu 12 minggu atau 6 Sprint dengan menggunakan metode pengembangan perangkat lunak Scrum secara lebih efektif dengan menghasilkan *Minimum Viable Product* dibuktikan dengan nilai hasil pengujian menggunakan User Experience Questionare (UEQ) dengan hasil secara keseluruhan mendapatkan nilai *Above* yaitu 1,085 atau diatas rata-rata aplikasi lain dibandingkan data 20.190 responden dari 452 studi(UEQ Data Analisis Tools Versi 8).
3. Hasil pengujian menggunakan User Experience Questionare (UEQ) didapatkan bahwa hasil secara keseluruhan mendapatkan nilai *Above Average*, atau diatas rata-rata daripada aplikasi aplikasi lain sesuai dengan standar yang ditentukan oleh UEQ. Pada setiap grup penilaian UEQ mendapatkan nilai rata-rata 1,145 untuk grup Stimulasi, nilai rata-rata

1,185 untuk grup Ketepatan, nilai rata-rata 1,207 untuk Efisiensi, nilai rata-rata 1,630 untuk grup Kejelasan, nilai rata-rata 1,145 untuk grup Daya Tarik dan nilai rata-rata 0,217 untuk Kebaruan.

4. Pada proses wawancara dengan 10 responded, didapatkan hasil bahwa ditemukan beberapa permasalahan yaitu responded yang kesulitan untuk melakukan login pada aplikasi karena kesalahan dalam memasukkan kode nomor telepon saat akan login. Permasalahan bisa teratasi saat pengguna menggunakan kode +62 untuk mengganti angka 0. Selain itu, karena konsep desain Deliv mirip dengan competitor, keseluruhan responden berpendapat bahwa ketiadaan pembaruan pada desain sehingga terlihat membosankan.

5.2 Saran

Pada penelitian berikutnya, diharapkan proses pengerjakan dalam menggunakan metode Scrum diharapkan untuk

1. Menggunakan metode UEQ sebelum melakukan pembaruan / *update application* untuk bisa meningkatkan penampilan / keterbaruan dalam perangkat lunak Deliv agar mendapatkan nilai yang lebih baik atau diatas rata-rata.
2. Dalam proses manajemen pengembangan perangkat lunak Deliv untuk versi selanjutnya atau pembaruan perangkat lunak disarankan menggunakan metode Scrum untuk bisa tetap mempertahankan efisiensi dan efektifitas sesuai dari pengujian dengan metode UEQ.

DAFTAR PUSTAKA

- Adi, P. (2015). Scrum Method Implementation in a Software Development Project Management. *International Journal of Advanced Computer Science and Applications*, 6(9), 198–204. <https://doi.org/10.14569/ijacsa.2015.060927>
- Andry, J. F., Riwanto, R. E., Wijaya, R. L., & Adi, A. (2019). Development Point of Sales Using SCRUM Framework. *Journal of Systems Integration*, 10(1), 36–48. <https://doi.org/10.20470/jsi.v10i1.359>
- Azdy, R. A., & SN, A. (2012). Implementasi Scrum Pada Pengembangan Software Terdistribusi. *Seminar Nasional Informatika 2012*, 1(2), 32–37.
- Cookson, G., & Pishue, B. (2017). Global Traffic Scorecard. *Inrix Global Traffic Scorecard*, 1(February), 44. <https://media.bizj.us/view/img/10360454/inrix2016trafficscorecarden.pdf%0Ahttp://inrix.com/industries/>
- Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213–1221. <https://doi.org/10.1016/j.jss.2012.02.033>
- Kenett, R. S., Lenarduzzi, V., & Taibi, D. (2016). Implementing SCRUM using Business Process Management and Pattern Analysis Methodologies. *Proceedings - 42nd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2016*, 2(October 2017), 112–119. <https://doi.org/10.17708/drmj.2013.v02n02a03>
- Kniberg, H., & Bima, I. (n.d.). *Mengimplementasikan Scrum dan XP*.
- Lenarduzzi, V., & Taibi, D. (2016). MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product. *Proceedings - 42nd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2016*, 112–119. <https://doi.org/10.1109/SEAA.2016.56>
- Rola, P., & Kuchta, D. (2015). Implementing Scrum Method in International Teams—A Case Study. *Open Journal of Social Sciences*, 03(07), 300–305. <https://doi.org/10.4236/jss.2015.37043>
- Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017a). Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4), 40. <https://doi.org/10.9781/ijimai.2017.445>
- Schrepp, M., Hinderks, A., & Thomaschewski, J. (2017b). Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(6), 103. <https://doi.org/10.9781/ijimai.2017.09.001>
- Schwaber, K., & Sutherland, J. (2017). The Scrum Guide: The Definitive The Rules of the Game. *Scrum.Org and ScrumInc*, November, 19. <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

Utomo, P., & Prayitno, F. W. (2015). Perancangan Dashboard Sistem Informasi Untuk Agile Manajemen Proyek dengan Menggunakan JIRA (Studi Kasus: di PT. FLASHiZ Indonesia). *Sisfotek Global*, 5(2), 17–24. <http://journal.stmikglobal.ac.id/index.php/sisfotek/article/view/75>

LAMPIRAN

Lampiran 1

PENGISIAN KUISIONER UEQ OLEH RESPONDEN

Responses cannot be edited

Kuesioner Pengukuran User Experience Pada Aplikasi Mobile Deliv

Assalamu'alaikum wr. wb.
Perkenalkan nama saya Naufaldi Rafif S. Seorang UX Researcher dan General Member of IxDA Malang.

Pada saat ini saya sedang melakukan penelitian terkait user experience atau pengalaman pengguna pada aplikasi mobile Deliv. Deliv merupakan sebuah platform untuk pemesanan ojek online, makanan dan antar barang yang berlokasi di daerah Malang, Bojonegoro dan Tuban.

Download Apps Deliv Android : <https://bit.ly/deliv-andro>

Kuesioner ini akan memakan waktu sekitar 3-6 menit. Pilihlah jawaban yang sesuai dengan persepsi kamu, tidak ada jawaban benar ataupun salah.

Data yang diberikan akan dijaga kerahasiaannya dan hanya digunakan untuk kepentingan penelitian. Informasi pribadi, seperti nama dan nomor telepon, akan disamarkan.

Dimohon kepada responden untuk mencoba aplikasi Deliv terlebih dahulu dan memberikan penilaian dengan jujur tanpa adanya paksaan dan manipulasi. Terimakasih banyak! Semoga sukses dan sehat selalu :)

*** Required**

Email *

elfanzamuhammad@gmail.com

Nama *

Muhammad Elfanza

Atractiveness (Daya Tarik)

Skala terdiri atas 7 titik yang menggambar daya tarik aplikasi Deliv kepada pengguna

*

1 2 3 4 5 6 7

Menyusahkan (Annoying) Menyenangkan (enjoyable)

*

1 2 3 4 5 6 7

Baik (Good) Buruk (Bad)

*

1 2 3 4 5 6 7

Tidak disukai (unlikeable) Menggembirakan (pleasing)

*
1 2 3 4 5 6 7
Tidak Nyaman (unpleasant) Nyaman (pleasant)

*
1 2 3 4 5 6 7
Atraktif (attractive) Tidak Atraktif (unattractive)

*
1 2 3 4 5 6 7
Ramah Pengguna (friendly) Tidak Ramah Pengguna (unfriendly)

*
Kesan anda saat melakukan sesuatu atau task pada aplikasi Deliv
1 2 3 4 5 6 7
Terorganisasi (organized) Berantakan (cluttered)

Perpicuity (Kejelasan)

Skala terdiri atas 7 titik yang menggambar kejelasan aplikasi Deliv kepada pengguna

*
1 2 3 4 5 6 7
Tidak dapat dipahami (not understandable) Dapat dipahami (understandable)

*
1 2 3 4 5 6 7
Mudah dipelajari (easy to learn) Sulit dipelajari (difficult to learn)

*
1 2 3 4 5 6 7
Rumit (complicated) Sederhana (easy)

*
1 2 3 4 5 6 7
Jelas (clear) Mbingungkan (confusing)

Dependability (Ketepatan)

Skala terdiri atas 7 titik yang menggambar ketepatan aplikasi Deliv saat digunakan oleh pengguna

*
1 2 3 4 5 6 7
Tidak dapat diprediksi (unpredictable) Dapat diprediksi (predictable)

Efficiency (Efisiensi)

Skala terdiri atas 7 titik yang menggambar efisiensi aplikasi Deliv saat digunakan oleh pengguna

*
1 2 3 4 5 6 7
Cepat (fast) Lambat (slow)

*
1 2 3 4 5 6 7
tidak efisien (inefficient) efisien (efficient)

*
1 2 3 4 5 6 7
Tidak Praktis (impractical) Praktis (practical)

Stimulation (Stimulasi)

Skala terdiri atas 7 titik yang menggambarkan stimulasi aplikasi Deliv kepada pengguna

*
1 2 3 4 5 6 7
Bermanfaat (valuable) Kurang bermanfaat (inferior)

*
1 2 3 4 5 6 7
Membosankan (boring) Mengasyikkan (exciting)

*
1 2 3 4 5 6 7
Tidak Menarik (not interesting) Menarik (interesting)

*
1 2 3 4 5 6 7
memotivasi (motivating) tidak memotivasi (demotivating)

Novelty (Kebaruan)

Skala terdiri atas 7 titik yang menggambarkan kebaruan aplikasi Deliv kepada pengguna

*
1 2 3 4 5 6 7
Kreatif (Creative) Monoton (dull)

*
1 2 3 4 5 6 7
Berdaya Cipta (inventive) Konvensional (conventional)

*
1 2 3 4 5 6 7
Lazim (usual) Terdepan (leading edge)

*
1 2 3 4 5 6 7
Konservatif (conservative) Inovatif (innovative)

Lampiran 2

DAFTAR RESPONDEN KUISIONER UEQ

No.	Nama	Jenis kelamin	Usia
1.	Muhammad Elfanza	Laki-laki	20
2.	Nugra Agung Pramono	Laki-laki	25
3.	Oki Krisbianto	Laki-laki	31
4.	Sadam Alfian Pradana	Laki-laki	19
5.	Ahmad Riza	Laki-laki	23
6.	Putri Fahimatul Hasni	Perempuan	23
7.	Nadhif M Feponsa	Laki-laki	21
8.	Annisaa Utami	Perempuan	22
9.	Dhika Dewanti	Perempuan	24
10.	Puci	Perempuan	25
11.	Muhammad Surga Savero	Laki-laki	26
12.	Refly Ilham Syabana	Laki-laki	24
13.	Riski Kristiyan I	Perempuan	23
14.	Teguh Arief Ramadhan	Laki-laki	22
15.	Dani Prisma permana	Laki-laki	28
16.	Daniel Agung Setyawan	Laki-laki	30
17.	Ricky Tri Wibowo	Laki-laki	26
18.	Erisa Febriana	Perempuan	25
19.	Lidya	Perempuan	25
20.	Nada	Perempuan	22
21.	Galang	Laki-laki	25
22.	Anindya Rulita Apsari	Perempuan	25
23.	Abida Zhafira Inayasary	Perempuan	25
24.	Ridha Fara Andini	Perempuan	25
25.	Nuzulul	Laki-laki	24
26.	Dany Asary	Laki-laki	28

Lampiran 3

IDENTIFIKASI KEBUTUHAN PERANGKAT LUNAK DAN SOURCE CODE
PERANGKAT LUNAK DELIV

Perangkat Keras

No	Kebutuhan Perangkat Lunak	Versi
1.	Apache Web Server	2.2.15
2.	MySQL	5.7
3.	PHP	8.0.0
4.	Visual Studio Code	1.57
5.	Sketch	7.0
6.	Android Studio	4.0
7.	Node JS	15.12.0
8.	Firebase	4.6.0
9.	Google Map Service	3.45
10.	Git	2.32
11.	Java	8
12.	Kotlin	1.3.9
13.	Android	8.0

Perangkat Keras

No	Kebutuhan Perangkat Keras	Spesifikasi
1.	Virtual Private Server	RAM 4 GB, CPU 4 Core, Disk Space 80 GB
2.	Macbook Air 2017	RAM 8 GB, SSD 128GB
3.	Lenovo Idepad 330	RAM 8 GB, SSD 128GB
4.	Xiaomi Redmi Note 5	RAM 4 GB, Storage 16 GB Android 8.0
5.	Samsung Note 7	RAM 4 GB, Storage 64 GB Android 6.0

Kode Sumber *Homepage*

```

package co.id.deliv.deliv.entity.model

import android.graphics.Color
import co.id.deliv.deliv.R

data class BankAccount(var accountName: String, var bankName: String, var
accountNumber: String)

data class Inbox(var type: String)
data class Fiture(var title: String, var code: Int, var icon: Int, var color: Int)
data class History(var name: String, var nominal: String, var code: String, var
time: String,
                var address: String, var status: String, var paymentMethod:
String)

data class User(var id_user: String, var nama: String, var no_hp: String,
                var email: String, var token: String)

fun HomeMenu(balance: String, point: String): ArrayList<Fiture> {
    return ArrayList<Fiture>().apply {
        add(Fiture("DE-FOOD", 1, R.drawable.new_defood,
Color.parseColor("#FE9800")))
        add(Fiture("DE-RIDE", 2, R.mipmap.de_ride_logo,
Color.parseColor("#5d4037")))
        add(Fiture("DE-CAR", 5, R.drawable.new_derental,
Color.parseColor("#2195F0")))
        add(Fiture("DE-EXPRESS", 4, R.drawable.new_deexpress,
Color.parseColor("#A43BB5")))
        add(Fiture("DE-SHOP", 3, R.drawable.new_deshop,
Color.parseColor("#F34336")))
        add(Fiture("DE-MEDCARE", 6, R.drawable.new_demedcare,
Color.parseColor("#ff4081")))
        add(Fiture("DE-TUKANG", 7, R.drawable.new_detukang,
Color.parseColor("#5AB862")))
        add(Fiture(balance, 8, R.drawable.de_saldo, Color.parseColor("#000000")))
        add(Fiture("DE-PAYMENT", 9, R.drawable.new_depayout,
Color.parseColor("#3E4FB1")))
        //        add(Fiture(point, 9, R.drawable.de_points,
Color.parseColor("#000000")))
    }
}

```

Kode Sumber *Login*

```
package co.id.deliv.deliv.modules.Login

import android.content.Context
import android.content.Intent
import co.id.deliv.deliv.view.modules.login.LoginActivity

interface ILoginInteractor {
    fun login(username:String):Pair<Boolean, String?>
}

interface ILoginPresenter {
    fun login(username:String)
}

class LoginWireframe() {
    fun toLogin(context: Context) {
        context.startActivity(Intent(context, LoginActivity::class.java))
    }
}
```

Kode Sumber *DeExpress*

```
package co.id.deliv.deliv.modules.DeExpress

import android.content.Context
import android.content.Intent
import co.id.deliv.deliv.support.utils.Utils
import co.id.deliv.deliv.view.modules.deexpress.DeExpressView
import co.id.deliv.deliv.view.modules.deexpress.DetailProsesPemesanan
import co.id.deliv.deliv.view.modules.deexpress.ProsesSearchingDriver
import co.id.deliv.deliv.view.modules.deexpress.ProsesPengiriman

class DeExpressWireframe {

    fun toHome(context: Context) {
        val toHome = Intent(context, DeExpressView::class.java)
        toHome.flags = Intent.FLAG_ACTIVITY_NEW_TASK
        context.startActivity(toHome)
    }

    fun toProsesPemesanan(context: Context, idTransaksi: String, proses: String) {
        val to = Intent(context, ProsesPengiriman::class.java)
        to.flags = Intent.FLAG_ACTIVITY_NEW_TASK
        to.putExtra("idTransaksi", idTransaksi)
        to.putExtra("proses", proses)
        context.startActivity(to)
    }

    fun toSearchDriver(context: Context, idTransaksi: String) {
        val to = Intent(context, ProsesSearchingDriver::class.java)
        to.flags = Intent.FLAG_ACTIVITY_NEW_TASK
        to.putExtra("idTransaksi", idTransaksi)
        context.startActivity(to)
    }

    fun toLihatDetailPesanan(context: Context, json: String, action: String) {
        val intent = Intent(context, DetailProsesPemesanan::class.java)
        intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK
        intent.action = action
        intent.putExtra(Utils.JSON, json)
        context.startActivity(intent)
    }
}
```

Kode Sumber *Register*

```

package co.id.deliv.deliv.modules.Register

import co.id.deliv.deliv.BuildConfig
import co.id.deliv.deliv.support.Api.AuthAPI
import co.id.deliv.deliv.support.Api.MainApi
import co.id.deliv.deliv.support.ERROR_MESSAGE
import org.json.JSONObject

class RegisterInteractor constructor(var api:AuthAPI):IRegisterInteractor{
    override fun register(nama: String?, phone: String?, email: String?, token:
String?, referral: String?): Pair<Boolean, String?> {
        val body = JSONObject().apply {
            put("name", nama)
            put("phone", phone)
            put("email", email)
            put("gcm_id", token)
            put("referralCode", referral)
        }.toString()

        return try {
            val response = api.register(body).execute()
            when(response.isSuccessful) {
                true -> Pair(true, response.body())
                false -> {
                    val error = JSONObject(response.errorBody()?.string())
                    Pair(false, error.getString("message"))
                }
            }
        } catch (error:Exception) {
            if(BuildConfig.DEBUG) {error.printStackTrace()}
            return Pair(false, ERROR_MESSAGE)
        }
    }
}

```

Kode Sumber *Profile*

```

package co.id.deliv.deliv.modules.profil

import android.content.Context
import co.id.deliv.deliv.entity.model.DataProfile
import com.google.gson.Gson
import com.google.gson.reflect.TypeToken

class ProfilPresenter (val interactor: ProfilInteractor

```


Lampiran 4

DOKUMENTASI TIM PENGEMBANG DELIV DALAM *SPRINT PLANNING*
DAN *SPRINT REVIEW*



