

**ANALISIS DAN PERANCANGAN
APLIKASI CHATBOT KOLEKTOR
 MENGGUNAKAN METODE
EXTREME PROGRAMMING DAN
 FORWARD CHAINING**



Oleh :
MUHAMAD SIDIK
92316068

**PROGRAM PASCA SARJANA
UNIVERSITAS GUNADARMA
JAKARTA
2020**

ANALISIS DAN PERANCANGAN APLIKASI CHATBOT KOLEKTOR MENGGUNAKAN METODE EXTREME PROGRAMMING DAN FORWARD CHAINING

Oleh :
**MUHAMAD SIDIK
92316068**

TESIS
**Untuk Memenuhi Salah Satu Syarat Guna
Memperoleh**
Gelar Magister Manajemen Sistem Informasi
Program Pasca Sarjana
Universitas Gunadarma

**PROGRAM PASCA SARJANA
UNIVERSITAS GUNADARMA
JAKARTA
2020**

Pernyataan Orisinalitas dan Publikasi

Saya yang bertanda tangan di bawah ini:

Nama : Muhamad Sidik
NIM : 92316068
Judul Penelitian : Analisis Dan Perancangan Aplikasi Chatbot
Kolektor Menggunakan Metode Extreme
Programming Dan Forward Chaining
Tanggal Sidang : -
Tanggal Lulus : -

Menyatakan bahwa tulisan di atas merupakan hasil karya saya sendiri dan dapat dipublikasikan sepenuhnya oleh Universitas Gunadarma. Segala kutipan dalam bentuk apapun telah mengikuti kaidah dan etika yang berlaku. Semua hak cipta dari logo serta produk yang disebut dalam buku ini adalah milik masing-masing pemegang haknya, kecuali disebutkan lain. Mengenai isi dan tulisan merupakan tanggung jawab Penulis, bukan Universitas Gunadarma.

Demikian pernyataan ini dibuat dengan sebenarnya dan dengan penuh kesadaran.

Jakarta, Februari 2020

Muhamad Sidik

Halaman Pengesahan

Judul Penelitian : Analisis Dan Perancangan Aplikasi Chatbot Kolektor Menggunakan Metode Extreme Programming Dan Forward Chaining
Nama : Muhamad Sidik
NIM : 92316068
Tanggal Lulus : -

Menyetujui,
Komisi Pembimbing

Dr. Bambang Gunawan
(Ketua)

Dr. Hustinawati SKom., MMSI
(Ketua Program Studi)

Program Pasca Sarjana Magister Teknologi dan Rekayasa

Dr. Tubagus Maulana Kusuma, SKom., MEngSc.
(Direktur)

Abstraksi

Muhamad Sidik, 92316068

ANALISIS DAN PERANCANGAN APLIKASI CHATBOT KOLEKTOR MENGGUNAKAN METODE EXTREME PROGRAMMING DAN FORWARD CHAINING.

Kata Kunci : Chatbot, line, asisten virtual, extreme programming, forward chaining, vb.net, sql server 2012

(xviii + 115 + lampiran)

Permintaan layanan informasi pada suatu bagian dalam satu perusahaan tentu menjadi sebuah kegiatan yang rutin terjadi, namun jika permintaan layanan informasi dilakukan dalam jumlah yang banyak dan waktu yang padat tentu akan menjadi sebuah masalah. Untuk menangani permasalahan menumpuknya permintaan layanan informasi tersebut banyak diterapkan pengembangan asisten virtual atau biasa disebut dengan *chatbot*. *Chatbot* merupakan sebuah aplikasi asisten virtual yang memungkinkan pemakaian untuk berinteraksi untuk mendapatkan respon tertentu secara langsung tanpa perlu menunggu operator untuk membalas pesan yang dikirim, sehingga pengembangan *chatbot* dirasa cocok untuk menangani permasalahan permintaan layanan informasi yang menumpuk. Selain itu, pengembangan *chatbot* juga berguna untuk melakukan penghematan terhadap biaya operasional yang dikeluarkan oleh perusahaan. Pada penelitian ini, dilakukan pembuatan suatu aplikasi *chatbot* dengan *platform LINE* dengan menggunakan metode *extreme programming* dan strategi *forward chaining* yang bertujuan untuk memenuhi kebutuhan informasi bagi kolektor di PT. Indomobil Finance Indonesia serta melakukan penghematan terhadap biaya operasional yang dikeluarkan oleh PT. Indomobil Finance Indonesia. Hasil dari pengembangan sistem berjalan pada layanan informasi kolektor PT. Indomobil Finance Indonesia menghasilkan sebuah aplikasi *chatbot* berbasis LINE. Hasil dari pengujian *blackbox testing* serta *user acceptance test* telah menghasilkan persentase keberhasilan aplikasi Chatbot Kolektor mencapai 95% dan persentase kegagalan 5%, hal tersebut menunjukkan bahwa proses pembuatan aplikasi *chatbot* untuk memenuhi kebutuhan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia berhasil dilakukan dan seluruh fungsional berjalan sesuai dengan yang diharapkan

Daftar Pustaka (2015 - 2020)

Abstract

Muhamad Sidik, 92316068

ANALYSIS AND DESIGN FOR CHATBOT APPLICATION USING EXTREME PROGRAMMING METHODS AND FORWARD CHAINING.

Keywords : Chatbot, line, virtual assistant, extreme programming, forward chaining, vb.net, sql server 2012

(xviii+ 115+ attachment)

Requests for information services in a section of one company would be a routine activity, but if the demand for information services is done in large quantities and time-consuming, it would be a problem. To deal with the problem of piling up requests for information services, many have applied the development of virtual assistants or commonly called chatbots. Chatbot is a virtual assistant application that allows users to interact to get certain responses directly without the need to wait for the operator to reply to messages sent, so that the chatbot development is considered suitable to handle the problems of information service requests that accumulate. In addition, the development of chatbot is also useful to make savings on operational costs incurred by the company. In this research, a chatbot application with LINE platform was made using extreme programming methods and forward chaining strategies aimed to fulfill the information needs of collectors at PT. Indomobil Finance Indonesia as well as making savings on operational costs incurred by PT. Indomobil Finance Indonesia. The results of the development of the system running on the information service collector PT. Indomobil Finance Indonesia produces a LINE-based chatbot application. The results of the blackbox testing and user acceptance tests have resulted in the success of the Chatbot Collector application reaching 95% and the failure percentage of 5%, it shows that the process of making the chatbot application to meet the information service needs of the PT. Indomobil Finance Indonesia was successfully carried out and all functional functions were as expected.

Bibliography (2015 - 2020)

Daftar Riwayat Hidup

1. Identitas Diri

Nama	:	Muhamad Sidik
NIM	:	92316068
Tempat/Tanggal Lahir	:	Bekasi, 19 November 1994
Alamat Rumah	:	Jalan Anggrek 3 rt.04 rw.015 no.15 Desa Setia Mekar, Kec. Tambun Selatan Kab. Bekasi, Jawa Barat - 17510
Email	:	m.sidik19@yahoo.com

2. Riwayat Pendidikan

Sarjana Sistem Informasi Universitas Gunadarma	2012-2016
SMKN 1 Tambelang	2009-2012
SMPN 1 Tambun Selatan	2006-2009
SDN Jatimulya 02	2000-2006

3. Pengalaman Kerja

Staff IT Development PT. Indomobil Finance Indonesia	Mey 2018-Sekarang
Freelance Project by Universitas Gunadarma	Feb 2017-Mar 2019
IOS Developer PT. IKonsultan Inovatama	Nov 2016-Feb 2017
Asisten Tetap Laboratorium Sistem Informasi Universitas Gunadarma	Jul 2014-Juni 2016
Staff Customer Relation PT. Multifiling Mitra Indonesia	May 2012-Jan 2013

Jakarta, Februari 2020

Muhamad Sidik, S. Kom

Kata Pengantar

Segala puji dan syukur penulis naikkan ke hadirat Allah SWT yang Maha Kuasa yang telah memberikan berkat, anugerah dan karunia yang melimpah, sehingga penulis dapat menyelesaikan tesis yang berjudul “ANALISIS DAN PERANCANGAN APLIKASI CHATBOT KOLEKTOR MENGGUNAKAN METODE EXTREME PROGRAMMING DAN FORWARD CHAINING”.

Pada kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Ibu Prof. Dr. E.S. Margianti, SE, MM, selaku Rektor Universitas Gunadarma yang telah memberikan kesempatan kepada penulis untuk melanjutkan pendidikan dan dalam penyusunan tesis ini.
2. Bapak Prof. Suryadi Harmanto, SSi., MMSI, Selaku Pembantu Rektor II Universitas Gunadarma.
3. Bapak Dr. Tubagus Maulana Kusuma, SKom., MEngSc., selaku Direktur Program Pasca Sarjana Universitas Gunadarma.
4. Ibu Dr. Hustinawati SKom., MMSI, selaku Ketua Prodi Program Pasca Sarjana Universitas Gunadarma.
5. Bapak Dr. Bambang Gunawan, selaku dosen pembimbing serta dosen pengajar yang telah banyak meluangkan waktu, tenaga serta pikiran dalam membimbing dan mengarahkan penulis untuk menyelesaikan penelitian tesis ini.
6. Ibu Dr. Dina Anggraini, selaku dosen pendamping yang banyak membantu dalam mengarahkan penulis untuk menyelesaikan tesis ini.
7. Ayahanda serta ibunda yang telah banyak memberikan dorongan dan semangat tak ternilai serta doa restu mereka jugalah terlaksananya penulisan tesis ini.

8. Aulia Rosiana yang telah banyak memberikan dukungan dan semangat tak terbatas bagi penulis.
9. Seluruh rekan-rekan angkatan 51 MMSI seperjuangan di Universitas Gunadarma yang telah banyak membantu penulis.
10. Semua pihak yang tidak tersebutkan yang telah membantu penyelesaian tesis ini, penulis ucapkan juga terima kasih atas segala bantuan dan sarannya.

Sebagai manusia biasa yang tak luput dari kesalahan, maka penulis meminta maaf atas segala kekurangan dan keterbatasan dalam penyusunan tesis ini.

Akhir kata, hanya kepada Tuhan jualah segalanya dikembalikan dan penulis sadari bahwa penelitian ini masih jauh dari sempurna, disebabkan karena berbagai keterbatasan yang penulis miliki. Untuk itu penulis mengharapkan kritik dan saran yang bersifat membangun untuk menjadi perbaikan di masa yang akan datang.

Semoga apa yang ada pada penelitian tesis ini dapat bermanfaat bagi kita semua. Amin.

Jakarta, Februari 2020

Penulis

Daftar Isi

Halaman Judul	i
Lembar Orisinalitas	ii
Halaman Pengesahan	iii
Abstraksi	iv
Abstract	v
Daftar Riwayat Hidup	vi
Kata Pengantar	vii
Daftar Isi	xiii
Daftar Gambar	xvi
Daftar Tabel	xvii
Daftar Lampiran	xviii
1 PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	5
1.3 Batasan Masalah	6
1.4 Tujuan Penelitian	6
1.5 Kegunaan Penelitian	7
2 TELAAH PUSTAKA	8
2.1 <i>Chatbot</i>	8
2.2 LINE (<i>Messaging LINE API</i>)	9
2.3 <i>Extreme Programming</i>	11

2.4	Visual Basic Net. 2010	12
2.5	SQL Server 2012	13
2.6	<i>Restful Web Service</i>	15
2.7	<i>Forward Chaining</i>	16
2.8	Pengujian Perangkat Lunak	17
2.8.1	<i>Black Box Testing</i>	18
2.9	Kajian Penelitian	18
3	METODE PENELITIAN	23
3.1	Analisa Masalah	23
3.2	Metode Penelitian Pembuatan Aplikasi Chatbot Kolektor . . .	24
3.3	Tahapan <i>Extreme Programming</i> Pada Pembuatan Aplikasi Cha- bot Kolektor	26
3.3.1	Tahap <i>Planning</i>	26
3.3.1.1	Pengumpulan Data	26
3.3.1.2	Analisa Sistem yang sedang berjalan	27
3.3.1.3	Analisa Arsitektur Aplikasi SMSCenter	29
3.3.1.4	Analisa arsitektur teknologi yang sedang ber- jalan	30
3.3.1.5	Analisa arsitektur konsol pada Aplikasi SM- SCenter	30
3.3.2	Tahap <i>Design</i>	32
3.3.2.1	Rancangan arsitektur Chatbot Kolektor . . .	35
3.3.2.2	Rancangan tampilan <i>chatbot</i> ketika ditam- bahkan menjadi Teman.	36
3.3.2.3	Rancangan tampilan <i>chatbot</i> ketika <i>user</i> valid melakukan pendaftaran.	37
3.3.2.4	Rancangan tampilan <i>chatbot</i> ketika <i>user</i> tidak valid melakukan pendaftaran.	38
3.3.2.5	Rancangan tampilan contoh kode <i>OTP</i> yang dikirimkan kepada <i>user</i>	39
3.3.2.6	Rancangan tampilan <i>user</i> memasukkan kode <i>OTP</i>	40
3.3.2.7	Rancangan tampilan notifikasi kode <i>OTP</i> salah. .	41
3.3.2.8	Rancangan tampilan notifikasi pendaftaran <i>user</i>	42
3.3.2.9	Rancangan tampilan menu <i>user</i>	43
3.3.2.10	Rancangan tampilan <i>help user</i>	44

3.3.2.11 Rancangan tampilan <i>user</i> jika memasukkan kata yang bukan termasuk kata kunci.	45
3.3.2.12 Rancangan tampilan respon modul BLK (<i>Blacklist</i>) bila status calon konsumen OK.	46
3.3.2.13 Rancangan tampilan respon modul BLK (<i>Blacklist</i>) bila status calon konsumen sudah pernah menjadi konsumen sebelumnya.	47
3.3.2.14 Rancangan tampilan respon modul BLK (<i>Blacklist</i>) bila status calon konsumen tidak OK.	48
3.3.2.15 Rancangan tampilan respon modul KOL (Kolektor) bila status lancar.	49
3.3.2.16 Rancangan tampilan respon modul KOL (Kolektor) bila status overdue.	50
3.3.2.17 Rancangan tampilan respon modul KOL (Kolektor) bila status ditarik.	51
3.3.2.18 Rancangan tampilan respon modul KOL (Kolektor) bila data tidak ditemukan.	52
3.3.2.19 Rancangan tampilan respon modul MSN (Mesin) bila data OK.	53
3.3.2.20 Rancangan tampilan respon modul MSN (Mesin) bila data <i>overdue</i>	54
3.3.2.21 Rancangan tampilan respon modul MSN (Mesin) bila data tidak ditemukan.	55
3.3.2.22 Rancangan tampilan respon modul RNK (Rangka) bila data OK.	56
3.3.2.23 Rancangan tampilan respon modul RNK (Rangka) bila data <i>overdue</i>	57
3.3.2.24 Rancangan tampilan respon modul RNK (Rangka) bila data tidak ditemukan.	58
3.3.3 Tahap Pembuatan <i>Coding</i>	58
3.3.4 Tahap <i>Testing</i> Aplikasi	60
3.4 Pengujian Perangkat Lunak	60

4.1.2	Tampilan <i>chatbot</i> ketika <i>user</i> valid melakukan pendaftaran.	63
4.1.3	Tampilan <i>chatbot</i> ketika <i>user</i> tidak valid melakukan pendaftaran.	64
4.1.4	Tampilan contoh kode <i>OTP</i> yang dikirimkan kepada <i>user</i>	65
4.1.5	Tampilan <i>user</i> memasukkan kode <i>OTP</i>	66
4.1.6	Tampilan notifikasi kode <i>OTP</i> salah.	67
4.1.7	Tampilan notifikasi pendaftaran <i>user</i>	68
4.1.8	Tampilan menu <i>user</i>	69
4.1.9	Tampilan <i>help user</i>	70
4.1.10	Tampilan <i>user</i> jika memasukkan kata yang bukan termasuk kata kunci.	71
4.1.11	Tampilan respon modul BLK (<i>Blacklist</i>) bila status calon konsumen OK.	72
4.1.12	Tampilan respon modul BLK (<i>Blacklist</i>) bila status calon konsumen sudah pernah menjadi konsumen sebelumnya.	73
4.1.13	Tampilan respon modul BLK (<i>Blacklist</i>) bila status calon konsumen tidak OK.	74
4.1.14	Tampilan respon modul KOL (<i>Kolektor</i>) bila status lancar.	75
4.1.15	Tampilan respon modul KOL (<i>Kolektor</i>) bila status overdue.	76
4.1.16	Tampilan respon modul KOL (<i>Kolektor</i>) bila status ditarik.	77
4.1.17	Tampilan respon modul KOL (<i>Kolektor</i>) bila data tidak ditemukan.	78
4.1.18	Tampilan respon modul MSN (<i>Mesin</i>) bila data OK.	79
4.1.19	Tampilan respon modul MSN (<i>Mesin</i>) bila data <i>overdue</i>	80
4.1.20	Tampilan respon modul MSN (<i>Mesin</i>) bila data tidak ditemukan.	81
4.1.21	Tampilan respon modul RNK (<i>Rangka</i>) bila data OK.	82
4.1.22	Tampilan respon modul RNK (<i>Rangka</i>) bila data <i>overdue</i>	83
4.1.23	Tampilan respon modul RNK (<i>Rangka</i>) bila data tidak ditemukan.	84
4.2	Implementasi <i>Coding</i> pada Aplikasi Chatbot Kolektor	85
4.2.1	Pembuatan <i>web service</i>	85
4.2.2	Pembuatan <i>Knowledge Base</i>	86

4.2.3 Konfigurasi <i>Webhook</i>	86
4.2.4 Pembuatan <i>Logic Program</i>	93
4.2.4.1 Penerapan Strategi Forward Chaining	93
4.3 Hasil Perbandingan Sistem	94
4.4 Pengujian Perangkat Lunak	98
4.4.1 <i>Blackbox Testing</i>	98
4.4.2 <i>User Acceptance Test</i>	102
5 KESIMPULAN DAN SARAN	111
5.1 Kesimpulan	111
5.2 Saran	112
Daftar Pustaka	115
Lampiran	1

Daftar Gambar

2.1	Arsitektur <i>LINE Messaging API</i>	10
2.2	Arsitektur <i>Extreme Programming</i>	12
2.3	Tampilan antarmuka Visual Studio 2010	13
2.4	Tampilan Antarmuka SQL Server 2012	15
2.5	Arsitektur <i>RESTful API</i>	16
2.6	Arsitektur <i>Forward Chaining</i>	17
2.7	Arsitektur <i>Blackbox Testing</i>	18
3.1	Tahapan Penelitian	25
3.2	Use Case Diagram aplikasi SMSCenter PT. Indomobil Finance Indonesia	28
3.3	Arsitektur Aplikasi SMSCenter	29
3.4	Arsitektur Konsol SMSCenter	31
3.5	Modem GSM untuk Konsol SMSCenter	32
3.6	Network Diagram Forward Chaining pada Aplikasi Chatbot Kolektor	34
3.7	Rancangan Arsitektur Chatbot Kolektor dengan platform LINE	35
3.8	Rancangan tampilan Chatbot Kolektor ketika ditambahkan sebagai Teman	36
3.9	Rancangan tampilan Chatbot Kolektor ketika <i>user</i> valid melakukan Pendaftaran	37
3.10	Rancangan tampilan Chatbot Kolektor ketika <i>user</i> tidak valid melakukan Pendaftaran	38
3.11	Rancangan tampilan sms berisi kode <i>OTP</i> yang dikirimkan oleh chatbot kolektor	39
3.12	Rancangan tampilan ketika <i>user</i> memasukkan kode <i>OTP</i>	40
3.13	Rancangan tampilan notifikasi <i>OTP</i> salah	41
3.14	Rancangan tampilan notifikasi pendaftaran	42
3.15	Rancangan tampilan halaman menu <i>user</i>	43
3.16	Rancangan tampilan halaman <i>help user</i>	44

3.17 Rancangan tampilan <i>user</i> ketika <i>user</i> memasukkan kata “hai”	45
3.18 Rancangan tampilan <i>user</i> ketika <i>user</i> memasukkan kata asal	45
3.19 Rancangan tampilan respon modul BLK konsumen OK	46
3.20 Rancangan tampilan respon modul BLK calon konsumen pernah menjadi konsumen	47
3.21 Rancangan tampilan respon modul BLK calon konsumen tidak OK	48
3.22 Rancangan tampilan respon modul KOL lancar	49
3.23 Rancangan tampilan respon modul KOL <i>overdue</i>	50
3.24 Rancangan tampilan respon modul KOL <i>overdue</i>	51
3.25 Rancangan tampilan respon modul KOL <i>overdue</i>	52
3.26 Rancangan tampilan respon modul MSN OK	53
3.27 Rancangan tampilan respon modul MSN <i>overdue</i>	54
3.28 Rancangan tampilan respon modul MSN tidak ditemukan	55
3.29 Rancangan tampilan respon modul RNK data OK	56
3.30 Rancangan tampilan respon modul RNK data <i>overdue</i>	57
3.31 Rancangan tampilan respon modul RNK data <i>overdue</i>	58
 4.1 Tampilan Chatbot Kolektor ketika ditambahkan sebagai Teman	62
4.2 Tampilan Chatbot Kolektor ketika <i>user</i> valid melakukan Pendaftaran	63
4.3 Tampilan Chatbot Kolektor ketika <i>user</i> tidak valid melakukan Pendaftaran	64
4.4 Tampilan sms berisi kode <i>OTP</i> yang dikirimkan oleh chatbot kolektor	65
4.5 Tampilan ketika <i>user</i> memasukkan kode <i>OTP</i>	66
4.6 Tampilan notifikasi <i>OTP</i> salah	67
4.7 Tampilan notifikasi pendaftaran berhasil	68
4.8 Tampilan halaman menu <i>user</i>	69
4.9 Tampilan halaman <i>help user</i>	70
4.10 Tampilan <i>user</i> ketika <i>user</i> memasukkan kata “hai”	71
4.11 Tampilan respon modul BLK konsumen OK	72
4.12 Tampilan respon modul BLK calon konsumen pernah menjadi konsumen	73
4.13 Tampilan respon modul BLK calon konsumen tidak OK	74
4.14 Tampilan respon modul KOL lancar	75
4.15 Tampilan respon modul KOL <i>overdue</i>	76
4.16 Tampilan respon modul KOL ditarik	77

4.17 Tampilan respon modul KOL <i>overdue</i>	78
4.18 Tampilan respon modul MSN OK	79
4.19 Tampilan respon modul MSN <i>overdue</i>	80
4.20 Tampilan respon modul MSN tidak ditemukan	81
4.21 Tampilan respon modul RNK data OK	82
4.22 Tampilan respon modul RNK data <i>overdue</i>	83
4.23 Tampilan respon modul RNK data <i>overdue</i>	84
4.24 Halaman awal LINE Developer	87
4.25 Halaman pembuatan Line Provider	87
4.26 Halaman pemilihan jenis channel	88
4.27 Halaman pembuatan channel di provider	88
4.28 Lanjutan halaman pembuatan channel di provider (1)	89
4.29 Lanjutan halaman pembuatan channel di provider (2)	89
4.30 Tampilan <i>term and condition</i> LINE	90
4.31 Tampilan <i>Setting Channel</i> LINE	90
4.32 Tampilan <i>Channel Access Token</i>	91
4.33 Tampilan setelan <i>Messaging API</i>	91
4.34 Lanjutan tampilan setelan <i>Messaging API</i>	92
4.35 Tampilan Sukses <i>Verify</i>	92

Daftar Tabel

2.1	Ringkasan Kajian Penelitian	19
3.1	Teknologi Perangkat Keras	30
3.2	Teknologi Perangkat Lunak	30
3.3	Tabel Format Kata Kunci Aplikasi Chatbot Kolektor	33
4.1	Hasil perbandingan aplikasi SMSCenter dengan Chatbot Kolektor	94
4.2	Hasil perbandingan biaya operasional SMSCenter dengan Chatbot Kolektor	96
4.3	Hasil analisa kebutuhan dan spesifikasi pengujian perangkat lunak dengan <i>blackbox testing</i>	99
4.4	Skenario Pemilihan Input Pengujian Perangkat Lunak Dengan <i>Blackbox Testing</i>	99
4.5	Skenario Pemilihan Output Pengujian Perangkat Lunak Dengan <i>Blackbox Testing</i>	100
4.6	Hasil pengujian perangkat lunak dengan menggunakan <i>Blackbox Testing</i>	101
4.7	<i>Form UAT</i> yang diberikan ke <i>user</i> untuk validasi bahwa program sudah sesuai.	103
4.8	Hasil Pengujian Kolektor pada Aplikasi Chatbot Kolektor . . .	104

Daftar Lampiran

Lampiran 1	L 1
Lampiran 2	L 4
Lampiran 3	L 7
Lampiran 4	L 17
Lampiran 5	L 22
Lampiran 6	L 27
Lampiran 7	L 33
Lampiran 8	L 39
Lampiran 9	L 42
Lampiran 10	L 50
Lampiran 11	L 59
Lampiran 12	L 69
Lampiran 13	L 73
Lampiran 14	L 83
Lampiran 15	L 87
Lampiran 16	L 88
Lampiran 17	L 89

Bab 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Pemanfaatan teknologi komunikasi dan informasi saat ini tidak dapat dilepaskan dari kehidupan sehari-hari, pada penerapan teknologi 4.0 saat ini pemanfaatan teknologi dan komunikasi diterapkan hampir diseluruh aspek kehidupan manusia, mulai dari perkantoran, industri pabrik, hingga rumah tangga. Teknologi Komunikasi dan Informasi adalah aplikasi pengetahuan dan keterampilan yang digunakan manusia dalam mengalirkan informasi atau pesan dengan tujuan untuk membantu menyelesaikan permasalahan manusia (aktivitas sosial) agar tercapai tujuan komunikasi (Setiawan, 2018). Di Indonesia sendiri, teknologi informasi berkembang cukup pesat dan berpengaruh pada banyak organisasi, menurut Liliweri, hubungan teknologi informasi dengan organisasi berfungsi mengalihkan pesan (informasi) untuk mencapai tujuan komunikasi. Teknologi Komunikasi dan Informasi harus dipahami sebagai istilah yang sangat kompleks seperti artefak, teknik dan pengetahuan yang digunakan untuk membantu memecahkan masalah manusia termasuk memecahkan informasi dan komunikasi. Secara umum dapat meliputi penggunaan komputer hingga penggunaan satelit. Oleh karena itu, frase Teknologi Komunikasi dan Informasi baru terkadang digunakan untuk menggambarkan semua teknologi yang berkaitan dengan elektronik daripada yang berarti mekanis (Setiawan, 2018).

Perkembangan teknologi informasi ini tidak hanya berpengaruh pada bidang yang berkaitan dengan komputer saja, tetapi juga berpengaruh pada bidang-bidang lain guna menunjang proses bisnis mereka agar menjadi lebih cepat, serta akurat. Salah satu bidang yang ikut terpengaruh dengan adanya perkembangan teknologi adalah *customer relation*, banyak perusahaan yang

menerapkan pembaharuan teknologi untuk mendukung aktivitas pada bagian *customer relation* mereka. Hal ini dilakukan karena *customer relation* merupakan bagian yang sangat penting bagi suatu perusahaan, dimana *customer relation* merupakan pintu gerbang pertama antara perusahaan dengan pelanggan mereka, dengan perbaikan kualitas melalui penerapan teknologi terbaru pada *customer relation*, tentu akan membawa citra yang lebih baik bagi perusahaan tersebut.

Contoh dari perkembangan teknologi yang banyak diberlakukan pada bagian *customer relation* adalah penerapan aplikasi *chatbot* atau asisten virtual. Menurut D.Suryani dan E.L. Amalia *Chatbot* merupakan aplikasi komputer yang digunakan untuk melakukan percakapan lewat tulisan. Cara kerja *chatbot* mirip seperti mesin penjawab telepon yang banyak digunakan pada kantor-kantor dengan fasilitas dial up. Pertanyaan yang mampu dijawab oleh aplikasi *chatbot* adalah pertanyaan yang bersifat umum dan sudah dipersiapkan jawabannya (Suryani and Amalia, 2017). Sedangkan menurut literasi lain disebutkan bahwa teknologi *chatbot* adalah salah satu program aplikasi yang dirancang untuk dapat berkomunikasi langsung dengan manusia melalui pesan teks, suara ataupun keduanya.(Akhsan and Faizah, 2017; Hormansyah and Utama, 2018). Dalam bidang industri dan bisnis, *chatbot* sudah mulai digunakan untuk memberikan pelayanan *customer service* dan dapat berinteraksi dengan konsumennya. *Chatbot* bertugas melayani percakapan dengan pengunjung untuk berkomunikasi seresponsif mungkin sehingga memungkinkan pengunjung mendapat pesan balasan dalam waktu yang singkat. (Benedictus et al., 2017) Manfaat dari *chatbot* adalah untuk pelayanan informasi kepada pengguna tanpa melibatkan operator sebagai penjawab, jadi pekerjaan operator sudah di alihkan ke aplikasi *chatbot* (Eli-sabet and Afrianto, 2015).

Umumnya, *chatbot* digunakan untuk membuat respon atau interaksi secara langsung kepada pengguna layanan *chatbot* tersebut. Hal ini dikarenakan keterbatasan sumber daya pada bagian *customer relation* yang biasanya tidak sebanding dengan banyaknya permintaan layanan atau pengaduan yang dihadapi oleh bagian *customer relation* itu sendiri. Bahkan dibanyak kasus, pelanggan merasa sulit sekali untuk dapat terhubung kepada layanan *customer service* suatu perusahaan karena banyaknya antrean telepon pada nomor *customer service* perusahaan tersebut. Hal ini tentu akan membawa citra buruk bagi perusahaan karena dinilai tidak mampu menghadapi keluhan serta permintaan layanan dari pelanggan mereka, bukan tidak mungkin jika

pelanggan dari perusahaan tersebut akan berpindah menggunakan produk atau jasa lain, hal ini diakibatkan keluhan atau pemintaan layanan mereka sulit ditanggapi oleh perusahaan karena buruknya pelayanan pada bagian *customer relation* mereka.

Disisi lain, *chatbot* juga digunakan sebagai bentuk perbaikan layanan yang sudah disediakan oleh suatu perusahaan untuk meningkatkan mutu layanan mereka. Hal ini karena *chatbot* akan langsung merespon permintaan yang diajukan oleh user tanpa harus menunggu operator untuk membalas pesan atau telepon yang masuk. *Chatbot* akan langsung menampilkan jawaban informasi yang dibutuhkan oleh *user* dalam waktu yang sangat singkat tanpa perlu dimonitor oleh operator. Selain itu, pembuatan *chatbot* ditujukan untuk melakukan penghematan terhadap biaya operasional yang dikeluarkan oleh sebuah perusahaan. Oleh karena itu, *chatbot* banyak digunakan sebagai solusi dari berbagai permasalahan yang dihadapi oleh perusahaan khususnya pada bagian *customer service*. Layanan *chatbot* yang saat ini sudah banyak dibuat adalah layanan *chatbot* dengan menggunakan *platform LINE*.

Berdasarkan data penelitian yang dilakukan oleh We Are Social pada tahun 2019 yang membahas tentang penggunaan media sosial di Indonesia, sosial media yang paling banyak digunakan adalah sebagai berikut : YouTube 88%, Whatsapp 83%, Facebook 81%, Instagram 80%, LINE 59%, Twitter 52%, FB Messenger 47%, BBM 38%, LinkedIn 33%, Pinterest 29%, Skype 28%, WeChat 28%, Snapchat 26%, Path 25%, Tumblr 20%, dan Reddit 16% (S.Kemp, 2019). Jika dikelompokkan berdasarkan jenisnya, maka untuk jenis *messenger* pada urutan pertama adalah Whatsapp dengan 83% kemudian disusul LINE dengan 59%. Namun WhatsApp belum mendukung pembuatan *chatbot* secara bebas, WhatsApp hanya menyediakan layanan *chatbot* berbayar melalui WhatsApp Bisnis mereka. Di lain sisi, LINE Corporation secara publik membuka *Messaging API*-nya pada September 2016, dengan *Messaging API* ini para pengembang (*developer*) dapat membangun *chatbot* dengan *platform LINE* (LINE, 2016).

Salah satu perusahaan yang membutuhkan adanya pengembangan aplikasi *chatbot* adalah PT. Indomobil Finance Indonesia. PT. Indomobil Finance Indonesia adalah perusahaan yang bergerak dalam bisnis jasa pembiayaan kendaraan bermotor dan alat berat serta pembiayaan multiguna. Dengan didukung 240 cabang di seluruh Indonesia, proses bisnis PT. Indomobil Finance Indonesia dijalankan selama lebih dari 26 tahun. Sebagai perusahaan

yang bergerak dalam bisnis jasa pembiayaan yang tersebar diseluruh Indonesia, tentu PT. Indomobil Finance Indonesia menerima banyak permintaan layanan baik dari pihak internal maupun eksternal. Salah satunya adalah permintaan informasi bagi kolektor yang ada di 240 cabang PT. Indomobil Finance Indonesia. Dengan jumlah 255 orang kolektor, tentu akan menjadi sebuah masalah jika permintaan informasi dilakukan secara serentak kepada bagian *Remedial and Collection (RC)* PT. Indomobil Finance Indonesia sebagai pemegang informasi.

Permintaan layanan informasi oleh kolektor ini bermaksud untuk menampilkan informasi yang dibutuhkan bagi kolektor untuk menunjang pekerjaan yang dilakukan. Informasi yang dapat diakses tiap kolektor tentu berbeda sesuai dengan hak akses yang diberikan oleh masing-masing cabang terhadap kolektor tersebut. Namun secara umum ada 4(empat) informasi yang dapat diakses oleh kolektor, yaitu *Blacklist* untuk menampilkan informasi apakah pelanggan atau calon pelanggan merupakan seseorang yang masuk dalam daftar hitam oleh PT. Indomobil Finance Indonesia, Kolektor untuk menampilkan informasi rinci mengenai kendaraan yang terdaftar dalam database milik PT. Indomobil Finance Indonesia berdasarkan nomor polisi kendaraan terkait, No. Mesin untuk menampilkan informasi rinci mengenai kendaraan yang terdaftar dalam database milik PT. Indomobil Finance Indonesia berdasarkan nomor mesin kendaraan terkait, dan No. Rangka untuk menampilkan informasi rinci mengenai kendaraan yang terdaftar dalam database milik PT. Indomobil Finance Indonesia berdasarkan nomor rangka kendaraan terkait.

Saat ini PT. Indomobil Finance Indonesia menyediakan layanan SMSCenter untuk melayani permintaan informasi bagi kolektor, cara kerjanya adalah dengan mengirimkan sms ke nomor yang disediakan kemudian nomor tersebut akan mengirimkan pesan balasan berupa informasi yang diminta oleh kolektor. Namun cara ini dirasa kurang efektif karena bergantung pada kartu sim yang digunakan, jika kartu sim yang digunakan mengalami gangguan, maka layanan informasi kepada kolektor ikut terputus tanpa adanya layanan lain yang serupa. Sehingga jika gangguan tersebut terjadi dalam waktu yang cukup lama, kolektor juga tidak dapat menerima informasi apapun, hal ini tentu membuat pekerjaan dari kolektor tidak dapat dilakukan secara maksimal. Selain itu, beban biaya yang dikeluarkan oleh PT. Indomobil Finance Indonesia untuk membayar tagihan biaya sms tiap bulan terbilang cukup besar karena harus menyediakan layanan bagi 240 cabang dengan waktu

layanan setiap hari tanpa ada batasan waktu.

Pada penelitian terdahulu yang telah membahas tentang pembaruan teknologi informasi untuk mengatasi masalah permintaan pelayanan informasi, seperti pada penelitian milik Ananda Dwi R, Firdha Imamah, Yusuf Mei Andre S, dan Ardiansyah tentang Aplikasi Chatbot (MILKI BOT) Yang Terintegrasi Dengan Web CMS Untuk Customer Service Pada UKM Misnu (Ananda et al., 2018), telah berhasil menyelesaikan permasalahan yang dihadapi oleh *customer service* pada UKM Misnu. Selain itu ada pula penelitian yang dilakukan oleh Junadhi, dan Mardainis tentang LINE Chatbot Informasi Cuaca Wilayah Indonesia (Junadhi and Mardainis., 2019), juga telah berhasil menyelesaikan permasalahan permintaan informasi yang diangkat pada penelitian tersebut.

Berdasarkan penelitian tersebut, maka untuk menyelesaikan permasalahan yang dialami oleh PT.Indomobil Finance Indonesia, dalam penelitian ini akan dibuat sebuah layanan *chatbot* dengan *platform* LINE untuk memenuhi permintaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia dengan metode *Extreme Programming* dan *Forward Chaining*, serta digunakan bahasa pemrograman vb.net sebagai media untuk membuat *web service*, serta memanfaatkan *Messaging API LINE* untuk menghubungkan *web service* yang dibuat dengan *Line Official Account* yang disediakan.

1.2 Identifikasi Masalah

Permintaan layanan informasi adalah suatu kegiatan rutin yang dilakukan oleh tiap kolektor PT. Indomobil Finance Indonesia yang ada di 240 cabang di Indonesia. Permintaan informasi ini berlaku sepanjang hari dan dapat diakses kapanpun oleh semua kolektor yang terdaftar dalam database PT. Indomobil Finance Indonesia.

Saat ini, PT. Indomobil Finance Indonesia menyediakan layanan SMSCenter untuk memenuhi kebutuhan informasi bagi para kolektor, namun layanan sms ini bergantung dengan kualitas kartu sim yang digunakan. Jika kartu sim mengalami gangguan, maka seluruh layanan informasi bagi kolektor juga akan terputus, hal ini tentu akan menjadi sebuah masalah serius jika gangguan terjadi dalam rentang waktu yang cukup lama atau jika gangguan terjadi dalam frekuensi yang cukup sering terjadi. Selain itu, tagihan biaya sms yang dikeluarkan oleh PT. Indomobil Finance untuk layanan sms ini terbilang cukup besar untuk dikeluarkan tiap tahun.

Untuk mengatasi masalah pada permintaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia maka dilakukan pembuatan sebuah layanan *chatbot* dengan *platform* LINE sebagai aplikasi untuk memenuhi permintaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia. Pada penelitian ini, proses pembuatan layanan *chatbot* dilakukan dengan metode *Extreme Programming* dan *Forward Chaining* serta vb.net sebagai bahasa pemrograman untuk membuat *web service*, serta memanfaatkan *messaging* LINE API sebagai penghubung antara *web service* dengan *line official account* untuk menyelesaikan permasalahan permintaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia.

1.3 Batasan Masalah

Pada penelitian ini, terdapat beberapa batasan masalah diantaranya adalah

- Proses pembuatan layanan *chatbot* dilakukan pada proses permintaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia.
- Layanan *chatbot* yang dibuat adalah layanan yang menampilkan respon yang sudah disusun dalam *web service* tanpa menggunakan AI.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membuat sebuah layanan *chatbot* untuk mengurangi biaya operasional serta memenuhi kebutuhan permintaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia dengan menggunakan *platform* LINE berdasarkan metode *extreme programming* dan *Forward Chaining* serta bahasa pemrograman vb.net sebagai media pembuatan *web service*.

1.5 Kegunaan Penelitian

Kegunaan penelitian ini adalah untuk menghasilkan suatu metode untuk memecahkan permasalahan pada aplikasi yang menyediakan layanan informasi berupa pesan singkat atau sms dengan menyediakan sebuah layanan *chatbot* dengan menggunakan *platform* LINE berdasarkan metode *extreme programming* dan *Forward Chaining* serta vb.net sebagai media pembuatan *web services*. Metode tersebut dapat menyelesaikan permasalahan berhentinya layanan karena adanya gangguan pada kartu sim, serta membengkaknya tagihan untuk membayar layanan sms.

Bab 2

TELAAH PUSTAKA

2.1 *Chatbot*

Chatbot adalah suatu sistem yang memungkinkan untuk membals pesan yang dikirim oleh pengguna secara otomatis tanpa perlu pengawasan. *Chatbot* merupakan penggabungan dua kata yaitu *chat* dan *bot*. *Chat* adalah sebuha bentuk komunikasi yang dilakukan melalui sebuah tulisan. *Bot* merupakan program yang disusun sedemikian rupa yang memiliki pengetahuan didalamnya untuk dapat memberikan respon secara otomatis. Sehingga *Chatbot* dapat diartikan sebagai sebuah program yang mampu memberikan respon secara otomatis melalui media pesan atau tulisan serta dapat memberikan informasi dengan cepat dan efisien (Elisabet and Afrianto, 2015).

Chatbot (juga dikenal sebagai *talkbot*, *chatterbot*, *bot*) juga dapat diartikan sebagai sebuah perankat lunak yang dirancang secara khusus untuk mensimulasikan percakapan intelektual dengan satu atau lebih manusia. Contoh dari penggunaan *chatbot* secara yang saat ini sudah berjalan antara lain *weather bot* untuk mengetahui cuaca terkini, *grocery bot* untuk membantu memilih dan memesan bahan makanan, *news bot* untuk memberikan informasi terbaru, *life advice bot* untuk membantu memberikan solusi atas suatu masalah yang ditentukan pada *chatbot*, *personal finance bot* untuk membantu mengatur rencana keuangan, *scheduling bot* untuk membantu menyusun jadwal kegiatan sehari-hari (Jati and Maarif, 2018).

Cara kerja *chatbot* ini adalah dengan mengirim permintaan dalam bentuk teks lalu mereka akan memberikan respon sesuai kebutuhan yang diminta oleh pengguna. Ada dua jenis *chatbot*, pertama berdasarkan pada seperangkat aturan, dan versi yang lebih canggih lainnya menggunakan *machine learning*. *Chatbot* yang berfungsi berdasarkan aturan hanya dapat membe-

rikan respon kepada perintah yang sudah ditentukan dalam tahap *coding* pembuatan *chatbot* tersebut. *Chatbot* yang menggunakan *machine learning* atau biasa dikenal sebagai *smartbot* adalah *chatbot* yang dapat menganggap permintaan dalam bahasa natural yang bebas, tidak sebatas perintah yang sudah ditentukan polanya (Wijaya et al., 2015).

2.2 LINE (*Messaging LINE API*)

Line adalah sebuah aplikasi pengirim pesan instan yang dapat digunakan pada berbagai *platform* seperti *smartphone*, tablet, atau komputer. LINE difungsikan dengan menggunakan jaringan internet sehingga pengguna LINE dapat melakukan aktivitas seperti mengirim pesan teks, mengirim gambar, video, pesan suara, dan aktivitas lainnya. LINE diklaim sebagai aplikasi pengirim pesan instan terlaris di 42 negara (Chen, 2016).

LINE *platform* akan mengirimkan informasi ke server setiap kali ada interaksi antara pengguna dengan akun yang telah dibuat. Terdapat dua jenis interaksi yang dikenali oleh *platform* Line :

1. Pengguna mengirimkan pesan.
2. Pengguna melakukan sebuah aksi atau sebuah operasi, seperti menjadikan akun bisnis sebagai teman.

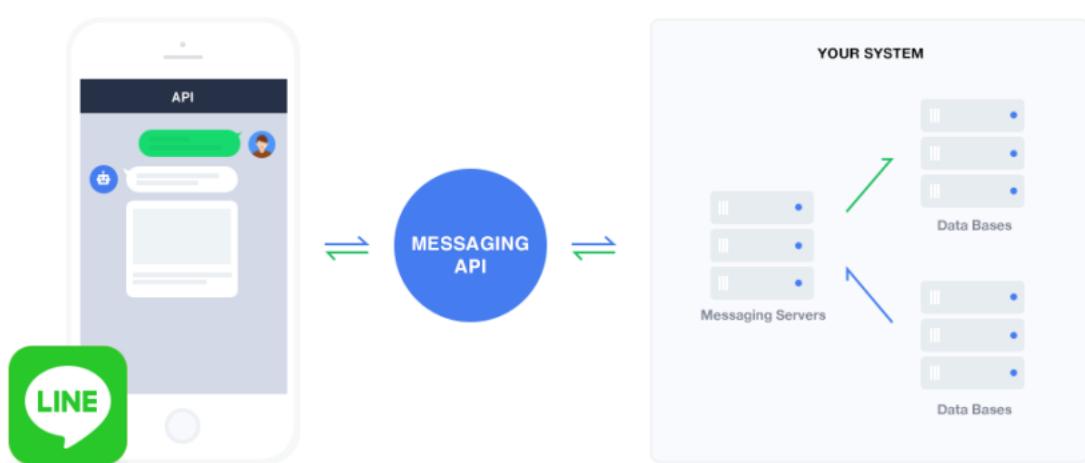
Informasi dikirim dalam format *JSON* melalui protokol *HTTPS* ke *URL* yang sudah didaftarkan sebagai *webhook*. Struktur data *JSON* yang dikirim akan bervariasi tergantung dari jenis pesan atau operasi yang akan digunakan oleh pengguna. Setiap *request* menyertakan kode *access token* yang wajib digunakan untuk memverifikasi bahwa *request* yang diterima adalah benar-benar dari *platform* LINE. Jika *access token* tidak dikenali, maka *request* dianggap tidak valid dan tidak akan mendapatkan respon dari LINE.

Ada beberapa fitur yang dapat dimanfaatkan pada *Messaging API* pada *platform* LINE, yaitu :

1. *Push Message* yang digunakan untuk Mengirim pesan langsung kepada pengguna yang sudah menambahkan akun *bot* sebagai teman.
2. *Reply Message* Membalas pesan yang dikirim pengguna ke akun *bot*. Notifikasi terkait aksi yang dilakukan oleh pengguna akan dikirimkan ke *URL webhook*.

3. *Imagemap Message* Mengirim gambar dengan teks dan tautan didalamnya. *Imagemap* dapat digunakan untuk mengirim konten seperti berita pengumuman, kupon, promo spesial, dan posting blog.
4. *Template messages* Mengirimkan pesan dengan bentuk dan format yang beragam, seperti gambar, teks, opsi pilihan dan tombol.
5. Menjalankan *Bot* di *Room* dan *Group* Berinteraksi dengan pengguna di dalam grup seperti halnya chat ke personal.

Secara umum, arsitektur dari line ditunjukan pada gambar berikut ini :



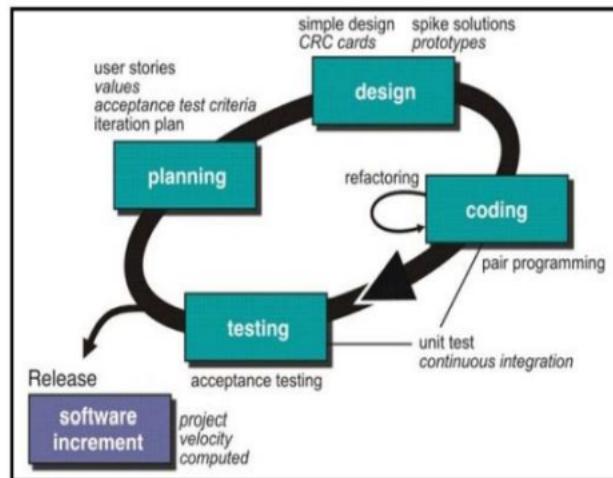
Gambar 2.1: Arsitektur *LINE Messaging API*

LINE memberikan fasilitas untuk membuat akun bisnis yang dapat memungkinkan *bot* mengirim pesan ke *customer* dan juga berkomunikasi langsung dengan pelanggan. Komunikasi tersebut dapat berbentuk chat langsung maupun dengan menggunakan fasilitas *auto reply* dan *keyword reply* yang disediakan oleh LINE. *Messaging API* dapat membuat respon yang fleksibel sehingga mampu disesuaikan dengan kebutuhan khusus yang tak tertangani fitur *auto reply* dan *keyword reply* standar. Melalui penggunaan *Messaging API*, aplikasi akan mampu bertukar informasi dengan server melalui platform LINE (Junadhi and Mardainis., 2019).

2.3 *Extreme Programming*

Metode pengembangan sistem yang digunakan adalah menggunakan metodologi *Extreme Programming*. *Extreme Programming (XP)* merupakan salah satu metode pengembangan perangkat lunak yang termasuk dalam metode pengembangan perangkat lunak *agile development*. *Extreme Programming (XP)* menggunakan pendekatan *object oriented* dan mencakup seperangkat aturan yang terjadi dalam 4 kerangka kegiatan: *planning*, *design*, *coding*, and *testing* (Carolina and Adi, 2019).

1. *Planning*, tahapan awal dari *extreme programming* adalah dengan mengumpulkan kebutuhan yang memungkinkan para anggota dari tim XP untuk dapat memahami konteks bisnis dari perangkat lunak yang akan dibuat. Hal ini juga bertujuan untuk mendapat wawasan yang luas untuk *output* apa yang diperlukan serta fitur-fitur utama dari perangkat lunak yang akan dibangun. Tahap ini akan mengarah pada pembuatan “*stories*” atau alur cerita yang menggambarkan *output* yang dibutuhkan, fitur, dan fungsi dari perangkat lunak yang akan dibuat.
2. *Design*, metode XP identik dengan aturan KIS (*KeepItSimple*). Aturan ini berjalan dengan cara konsep desain yang sederhana selalu diutamakan dibandingkan dengan representasi yang kompleks. Jika terdapat design yang sulit, XP akan menerapkan cara pembuatan *design* yang dilakukan segera, dan dibuat langsung ke tujuannya. XP juga mendukung adanya *refactoring* yang memungkinkan untuk melakukan perubahan pada kode program untuk disederhanakan tanpa merubah cara kode tersebut bekerja.
3. *Coding*, setelah “*stories*” atau alur cerita sudah dirancang dan desain awal sudah disetujui oleh user, tim tidak langsung memulai tahap *code*, tetapi sebelum memulai tahap *code*, tim terlebih dahulu merancang beberapa *unit test* yang digunakan untuk menjalankan “*stories*” dan disertakan pada *software release* saat itu. Setelah itu, pengembang fokus untuk mengimplementasikannya.
4. *Testing*, tahap testing dengan melakukan pengujian kode pada *unit test* yang telah dibuat sebelumnya. Pada metode XP dilakukan *user acceptance test (UAT)* atau biasa disebut *customer test*. Tes ini diberikan kepada *user* yang akan menggunakan fitur dan fungsi sistem yang sedang dibuat. (Carolina and Adi, 2019)



Gambar 2.2: Arsitektur *Extreme Programming*

2.4 Visual Basic Net. 2010

Microsoft Visual Studio adalah sebuah *Integrated Development Environment* yang dibuat dan dikembangkan oleh Microsoft Corporation. Microsoft Visual Studio dapat digunakan sebagai media untuk mengembangkan aplikasi dengan cara *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) maupun dengan cara *managed code* (dalam bentuk Microsoft *Intermediate Language* yang berjalan di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Windows Mobile* (yang berjalan di atas .NET Compact Framework). Visual Basic mencakup sebuah kode editor yang didukung oleh fitur *intellisense* atau yang disebut dengan *code refactoring*. *Debugger* telah terintegrasi dalam bahasa pemrograman Visual Basic ini. *Toll built in* mencakup *form designer* untuk membangun sebuah aplikasi GUI, *web designer*, *class designer* dan *database schema designer*.

Microsoft Visual Studio didukung bahasa pemrograman yang berbeda. Bahasa pemrograman yang didukung oleh Visual Basic Studio antara lain bahasa pemrograman C++, Visual Basic, dan Visual C#. Visual Studio juga dapat mendukung bahasa pemrograman lain seperti M, python dan ruby yang semuanya itu terdapat pada paket *install* yang terpisah dari visual studio. (Extise and Supriandi, 2016)



Gambar 2.3: Tampilan antarmuka Visual Studio 2010

2.5 SQL Server 2012

Microsoft SQL Server 2012 adalah sebuah media penyimpanan untuk data dan objek, termasuk didalamnya tabel, *view*, *stored procedure*, akun pengguna, dan *tools* manajemen lainnya. Database SQL Server selalu memiliki 1 esensi tunggal, yang didalamnya didukung oleh beberapa file fisik. (Novella, 2015)

Ketika aplikasi client mengirim perintah ke mesin database sebagai sebuah pernyataan T-SQL, SQL server melakukan semua kegiatan terhadap berkas, *memory*, dan pemanfaatan prosesor atas nama *user*. *User* tidak pernah mengakses *database* secara langsung, seperti aplikasi dalam aplikasi *database* desktop.

Pada arsitektur Microsoft SQL Server 2012 memiliki tiga macam bagian, yaitu :

1. *Service* yang didalamnya terdapat beberapa layanan dari SQL Server 2012 :

- SQL Server *database engine* 2012, yang bertanggung jawab untuk melaksanakan perintah yang disampaikan dalam Transact-SQL, ORT-SQL, bahasa, manajemen *database*, memori dan alokasi *disk*, dan fitur inti lainnya dari SQL Server.
- SQL Server *Agent*, yang bertanggung jawab untuk melaksanakan pekerjaan dijadwalkan, pemantauan sistem untuk kondisi waspadai didefinisikan, dan tugas-tugas administrasi lainnya.
- *Business Intelligence Components* (Komponen Intelijen Bisnis), termasuk SQL Server *Reporting Services*, SQL Server *Analysis Services*, dan SQL Server *Integration Services*.

2. *Instances*

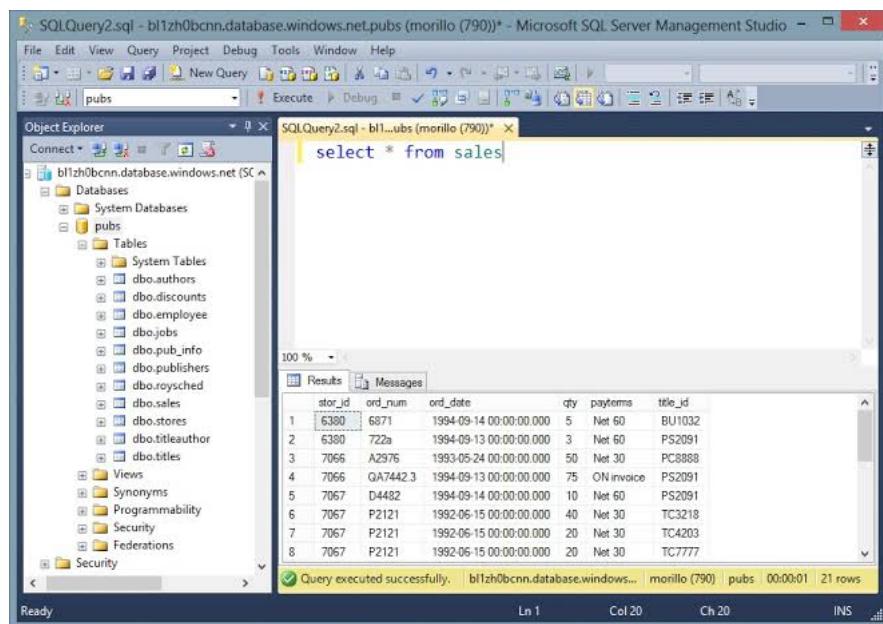
Unit dasar instalasi dan Program eksekusi untuk SQL Server adalah *Instances*. *Instances* merupakan salinan dari program SQLServer.exe, yang berjalan pada layanan Windows. Sebuah *Instances* mewakili program dan alokasi sumber daya yang mendukung sebuah salinan dari SQL Server yang sedang berjalan pada komputer, seperti *memory*, konfigurasi file, dan CPU. Setiap *Instances* terisolasi dari *instances* lain pada komputer yang sama, termasuk file *disk* yang digunakan, izin keamanan, dan sumber daya yang dialokasikan.

3. *Tools*

Bekerja dengan SQL Server Management Studio, atau yang biasa disebut dengan SSMS. SSMS adalah manajemen terpadu, yang memiliki pengembangan, dan penerapan *query* dengan banyak fitur untuk menjelajahi dan bekerja dengan *database*. Berikut fitur dari Microsoft SQL Server 2012 :

- SQL Server Pusat Instalasi, yang menyediakan kemampuan untuk menambah, menghapus, dan memodifikasi fitur program SQL Server.

- SQL Server *Configuration Manager*, yang mengatur alat untuk administrator, juga termasuk fitur yang berguna untuk mengelola software SQL Server untuk dapat diinstal pada mesin klien, seperti kemampuan untuk membuat dan mengelola alias untuk SQL Server.
- SQLCMD, adalah baris perintah yang memungkinkan *user* untuk mengirimkan perintah T-SQL sebagai alternatif menggunakan aplikasi grafis SSMS.



Gambar 2.4: Tampilan Antarmuka SQL Server 2012

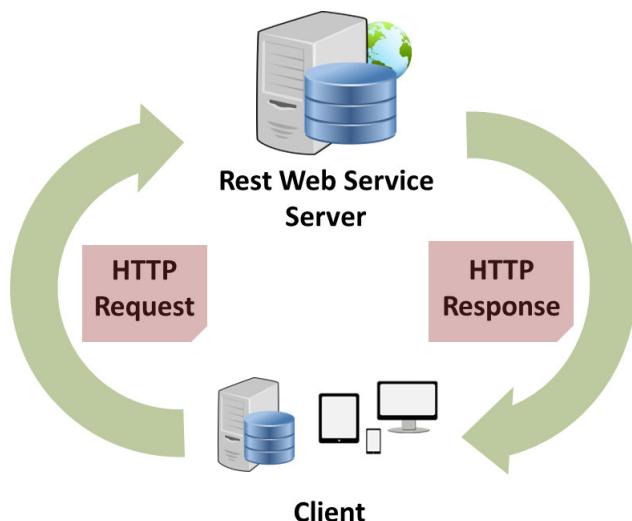
2.6 Restful Web Service

Representational State Transfer (REST) adalah sebuah bagian dari protokol HTTP (*Hypertext Transfer Protocol*) dimana REST mampu menyediakan antarmuka yang dapat membuat, mengambil, memperbarui, menghapus dan memanipulasi sumber daya dengan pertukaran representasi (Bagus et al., 2018).

Metode yang dapat dilakukan didalam sebuah *RESTful Web Service*, diantaranya:

1. *GET*, digunakan untuk mendapatkan sebuah sumber daya yang diidentifikasi dengan URI (*Uniform Resource Identifier*).

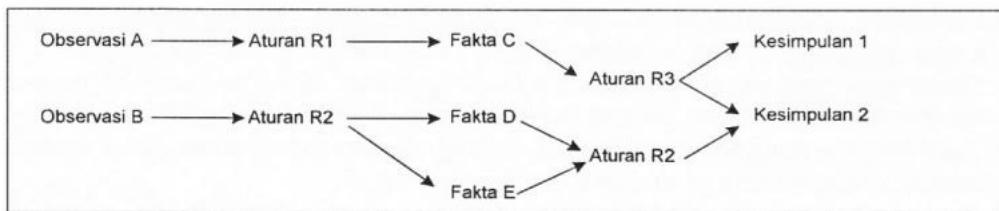
2. *POST*, digunakan untuk mengirimkan sumber daya ke server. Metode *POST* ini digunakan untuk membuat sumber daya baru.
3. *PUT*, digunakan untuk mengirimkan sumber daya ke server. Metode *POST* ini digunakan untuk memasukkan atau memperbarui sumber daya yang telah ada.
4. *DELETE*, digunakan untuk menghapus sumber daya yang diidentifikasi dengan URI



Gambar 2.5: Arsitektur *RESTful API*

2.7 *Forward Chaining*

Metode *Forward Chaining* adalah sebuah metode pencarian yang digunakan dengan cara teknik pelacakan ke depan. Metode ini dimulai dengan informasi yang ada dan digabungkan dengan rule untuk menghasilkan suatu kesimpulan atau tujuan (Lestari, 2016). Metode *Forward chaining* secara umum menghasilkan sebuah tujuan atau kesimpulan, dan merupakan suatu metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya.



Gambar 2.6: Arsitektur *Forward Chaining*

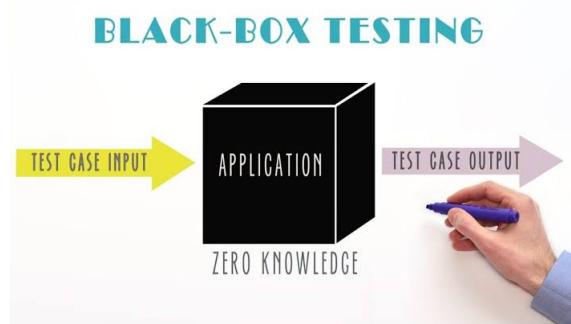
2.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak sangat dibutuhkan disetiap pengembangan atau pembuatan perangkat lunak. Hal ini bertujuan untuk memastikan sebuah perangkat lunak sudah sesuai dengan fungsional yang diharapkan. Umumnya, Pengujian perangkat lunak dilakukan setelah proses pengembangan aplikasi selesai, jika terdapat fungsional yang tidak sesuai harapan maka perangkat lunak dilakukan perbaikan. Menurut literatur Shi, pengujian atau *testing* sendiri merupakan elemen kritis untuk menjamin kualitas dari suatu perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan *software* seperti halnya analisis, desain, dan pengkodean (Sidi et al., 2015). Tahapan-tahapan yang digunakan untuk melakukan pengujian perangkat lunak adalah (Mufrizal and Indarti, 2019)

1. Perencanaan pengujian perangkat lunak yaitu melakukan perencanaan mengenai kebutuhan pengujian dari sebuah perangkat lunak yang akan diuji.
2. Analisis dan perancangan pengujian perangkat lunak yaitu menganalisa dan merancang berbagai skenario yang digunakan untuk menguji sebuah perangkat lunak.
3. Implementasi pengujian perangkat lunak yaitu menjalankan berbagai skenario pengujian pada aplikasi yang dituju.
4. Menganalisis dan mengevaluasi hasil pengujian perangkat lunak yaitu melakukan evaluasi mengenai hasil pengujian perangkat lunak yang telah selesai.

2.8.1 *Black Box Testing*

Black Box Testing (Pengujian Kotak Hitam) merupakan sebuah metode pengujian perangkat lunak yang berfokus pada kebutuhan fungsional dari aplikasi yang dikembangkan. Pengujian kotak hitam ini menguji perangkat lunak dengan menggunakan sudut pandang pengguna aplikasi. Tujuan dari pengujian kotak hitam ini adalah memastikan data diproses seperti seharusnya tanpa mengecek logika dari perangkat lunak (Vallabhaneni, 2015)



Gambar 2.7: Arsitektur *Blackbox Testing*

2.9 Kajian Penelitian

Pada Tabel 2.1 adalah beberapa penelitian terdahulu yang telah dilakukan terkait pembuatan aplikasi *chatbot*. Berikut ringkasan kajian terdahulu yang mengimplementasikan penelitian serupa.

Tabel 2.1: Ringkasan Kajian Penelitian

No	Tahun	Peneliti	Judul Penelitian	Metode	Hasil
1	2018	Ananda Dwi R, Firdha Imamah, Yusuf Mei Andre S, Ardiansyah	Aplikasi Chatbot (MILKI BOT) Yang Terintegrasi Dengan Web CMS Untuk Customer Service Pada UKM Misnu	Metode yang digunakan adalah metode <i>extreme programming</i> dan dikoneksikan dengan Web CMS milik UKM Misnu	metode yang diusulkan oleh peneliti berhasil diterapkan pada sistem <i>Customer Service</i> UKM Misnu
2	2018	Muhammad Sahid S Sadaton, Rendra Soekarta	Perancangan Sistem Informasi Pariwisata Kota Sorong Menggunakan WEB Dan LINE@ Chatbot Sebagai Media Informasi Pariwisata	Metode yang digunakan adalah <i>waterfall</i> dan <i>platform</i> yang digunakan adalah LINE@	Metode yang diusulkan oleh peneliti menunjukkan keberhasilan perancangan chatbot sebagai media penyedia informasi
3	2019	Eka Yuniar, Heri Purnomo	Implementasi Chatbot "ALITTA" Asisten Virtual Dari Ballitas Sebagai Pusat Informasi di Ballitas	Metode pengembangan yang digunakan oleh peneliti dengan <i>platform</i> berbeda serta penambahan NLP di dalamnya	Penelitian ini berhasil menunjukkan pengembangan chatbot dengan <i>platform</i> Telegram dan ditambahkan NLP untuk membuat bot yang mampu belajar sendiri.

Lanjutan Tabel 2.1: Ringkasan Kajian Penelitian

No	Tahun	Peneliti	Judul Penelitian	Metode	Hasil
4	2019	Junadhi, Mardainis	LINE Chatbot Informasi Cuaca Wilayah Indonesia	Metode yang digunakan adalah metode yang <i>waterfall</i> dengan penggunaan platform LINE yang digunakan peneliti	metode yang diusulkan oleh peneliti berhasil menampilkan informasi mengenai cuaca berdasarkan format masukan yang ditentukan
5	2019	Marline Wijaya, Junaedy, Hamdan Arfandy	Perancangan Chatbot Untuk Informasi Penerimaan Mahasiswa Baru Pada STMIK Kharisma Makassar	Metode yang digunakan adalah pemanfaatan <i>cloud service</i> sebagai jembatan antara <i>chatbot</i> dengan sumber data	Penelitian berhasil menerapkan sistem layanan chatbot yang <i>web servicenya</i> diletakkan pada <i>cloud service</i>

Pada Tahun 2018 telah dilakukan penelitian oleh Ananda Dwi R dan kawan-kawan (Ananda et al., 2018), pada penelitian tersebut pembuatan *chatbot* dilakukan menggunakan metode seperti yang peneliti gunakan yaitu metode *extreme programming*, penelitian yang dilakukan oleh Ananda Dwi R dan kawan-kawan membahas tentang bagaimana mengimplementasikan *chatbot* untuk menggantikan peran *customer service* dengan tujuan untuk memudahkan proses jual beli, selain itu pengembangan aplikasi *chatbot* pada penelitian tersebut juga bertujuan untuk menekan biaya operasional yang dikeluarkan oleh UKM Wisnu. Penelitian ini menghasilkan sebuah aplikasi

chatbot dengan memanfaatkan web *cms* yang sudah tersedia, cara kerja *chatbot* dibuat terkoneksi dengan *cms* yang sudah tersedia, sehingga *chat* yang terdapat pada *chatbot* juga akan terhubung dengan web *cms* untuk menampilkan informasi serta mengirimkan validasi kepada admin. Penelitian ini juga menjadi dasar bagi peneliti untuk melakukan pengembangan layanan *chatbot* untuk menangani permasalahan yang dihadapi oleh PT. Indomobil Finance Indonesia.

Pada tahun yang sama seperti Ananda Dwi R. dan kawan-kawan, Penelitian tentang *chatbot* juga dilakukan oleh Muhammad Sahid dan Rendra(Muhammad and Rendra, 2018), dalam penelitian yang dilakukan oleh Muhammad Sahid dan Rendra proses pembuatan *chatbot* dilakukan dengan metode waterfall sehingga waktu yang diperlukan untuk membangun *chatbot* sedikit lebih lama jika dibandingkan dengan metode yang dikerjakan oleh Ananda Dwi R. dan kawan-kawan (Ananda et al., 2018), yaitu metode *extreme programming*. Dari penelitian tersebut langkah yang diambil sebelum melakukan pembuatan *chatbot* lebih panjang dan juga memakan waktu namun hasil yang didapat cenderung lebih matang dibanding metode *extreme programming*. Hasil dari penelitian yang dilakukan oleh Muhammad Sahid adalah Chatbot yang melakukan interaksi dengan pengguna tanpa ada campur tangan manusia, berbeda dengan penelitian milik (Ananda et al., 2018) *chatbot* yang masih *semi auto* karena masih membutuhkan *input* basaran dari sisi admin untuk verifikasi pembayaran dan konfirmasi pengiriman serta penerimaan.

Di tahun berikutnya, yaitu tahun 2019 terdapat 3 penelitian lain yang membahas tentang *chatbot*. yaitu penelitian yang dilakukan oleh Junadhi dan Mardainis(Junadhi and Mardainis., 2019), Marline Wijaya dan kawan-kawan (Wijaya et al., 2019), serta Eka Yuniar dan Heri Purnomo(Yuniar and Purnomo, 2019). Penelitian yang dilakukan oleh Junadhi dan Mardainis (Junadhi and Mardainis., 2019) melanjutkan penelitian milik Muhammad Sahid dan Rendra (Muhammad and Rendra, 2018), dengan metode yang sama yaitu *waterfall*, penelitian ini berhasil membangun sebuah aplikasi *chatbot* yang berguna untuk memberikan informasi mengenai cuaca.

Sedikit berbeda dengan penelitian sebelumnya, pada penelitian Marline Wijaya dan kawan-kawan (Wijaya et al., 2019), peneliti melakukan pengembangan aplikasi *chatbot* dengan memanfaatkan layanan *cloud* sehingga resource yang digunakan dapat lebih ringan serta dapat diakses oleh semua pihak. Perbedaan yang paling mencolok adalah letak *data knowledge* pada

aplikasi *chatbot* yang dilakukan pada penelitian (Wijaya et al., 2019) berada pada *cloud service*, artinya *data knowledge* tidak memerlukan sebuah server fisik dan aplikasi *database* yang terpasang pada server fisik untuk dapat diakses dan menampung transaksi ke dalam database.

Selain itu pengembangan aplikasi *chatbot* juga dapat dilakukan pada *platform* berbeda dan dengan menambahkan pengetahuan atau *NLP* ke dalam *chatbot*, seperti contoh pada penelitian (Yuniar and Purnomo, 2019)*chatbot* dibangun dengan *platform* telegram, pada dasarnya konsep membangun aplikasi *chatbotnya* sama seperti penelitian-penelitian sebelumnya, yang membedakan pada penelitian (Yuniar and Purnomo, 2019)*chatbot* ditambahkan kecerdasan buatan dengan *platform* tambahan, sehingga membuat *chatbot* mampu belajar dari masukan yang berasal dari pengguna, sekalipun masukan tersebut belum pernah disiapkan responnya oleh pengembang aplikasi.

Dari beberapa penelitian tersebut maka pada penelitian ini dilakukan pembuatan *chatbot* dengan *platform* LINE berdasarkan metode *Extreme Programming* dan *Forward Chaining* seperti pada penelitian (Ananda et al., 2018) dan proses chatbot dipersiapkan untuk memberikan respon sesuai dengan masukan yang sudah disiapkan oleh pengembang seperti pada penelitian (Junadhi and Mardainis., 2019) dan (Muhammad and Rendra, 2018).

Bab 3

METODE PENELITIAN

3.1 Analisa Masalah

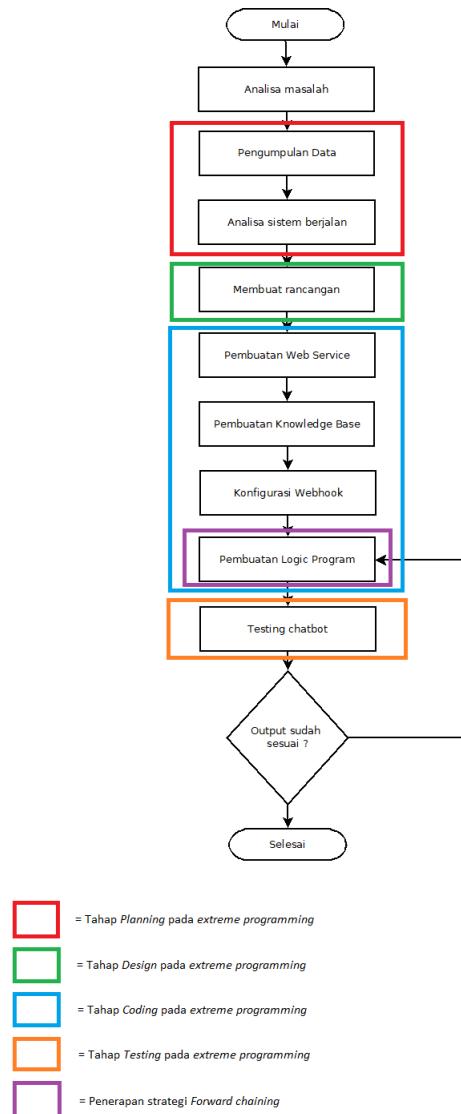
Permintaan layanan informasi kendaraan adalah kegiatan rutin yang dilakukan oleh kolektor di seluruh cabang PT. Indomobil Finance Indonesia. Banyaknya permintaan informasi dari seluruh kolektor di semua cabang PT. Indomobil Finance Indonesia tentu menjadi sebuah masalah untuk bagian *Remedial and Collection* selaku pemegang data. PT. Indomobil Finance Indonesia sudah menyediakan aplikasi layanan berbasis sms (SMSCenter) untuk memenuhi permintaan bagi para kolektor. Namun Permasalahan yang dihadapi pada aplikasi SMSCenter yang saat ini berjalan di PT. Indomobil Finance Indonesia yaitu bergantungnya sistem pada kondisi jaringan kartu sim yang digunakan, jika kartu sim mengalami gangguan baik dari sinyal maupun dari alat modem yang digunakan untuk membaca kartu sim, maka seluruh layanan juga akan terputus.

Selain masalah ketergantungan pada kondisi kartu sim, beban biaya operasional yang dikeluarkan oleh PT. Indomobil Finance Indonesia juga cukup besar karena banyaknya pesan yang dikirimkan kembali ke kolektor sebagai respon dari permintaan informasi. Untuk mengatasi masalah pada permintaan layanan informasi di PT. Indomobil Finance Indonesia, maka dilakukan pembuatan aplikasi *chatbot*. Proses pembuatan aplikasi *chatbot* berbasis LINE yang dilakukan pada penelitian ini menggunakan metode *expert programming* dan strategi *forward chaining* untuk menyelesaikan permasalahan permintaan layanan informasi di PT. Indomobil Finance.

3.2 Metode Penelitian Pembuatan Aplikasi Chatbot Kolektor

Proses pembuatan aplikasi *chatbot* pada penelitian ini dilakukan dengan menggunakan 2 tahapan yaitu metode *Extreme Programming* dan strategi *Forward Chaining*. Pada penelitian ini dilakukan 4 tahapan untuk melakukan proses pembuatan aplikasi *chatbot* sesuai dengan tahapan yang ada pada metode *Extreme Programming* (Carolina and Adi, 2019), tahapan tersebut terdiri dari *Planning*, *Design*, *Coding*, dan *Testing*. Dalam pembuatan aplikasi *chatbot* ini strategi *forward chaining* diterapkan untuk menentukan alur percakapan yang akan diterapkan pada aplikasi *chatbot*. Strategi ini menggunakan konsep “jika-maka”, dengan menentukan input apa yang akan dimasukkan oleh pengguna, maka *chatbot* akan mampu menampilkan informasi sesuai dengan apa yang diminta oleh pengguna.

Setelah menentukan tahapan pada *extreme programming* dan strategi *forward chaining*, maka dibuatlah tahapan penelitian yang mencakup kedua tahapan tersebut. Tahapan penelitian yang dilakukan pada penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1: Tahapan Penelitian

Gambar 3.1 menunjukkan detail dari masing-masing proses dalam metode *extreme programming*, yaitu garis berwarna merah untuk menunjukkan tahap *Planning*, garis berwarna hijau untuk menunjukkan tahap *Design*, garis berwarna biru untuk menunjukkan tahap *Coding*, dan garis berwarna oranye untuk menunjukkan proses *Testing*. Dari gambar 3.1 juga terlihat strategi *forward chaining* digunakan sebagai strategi pada saat proses pembuatan *logic program* yang ditandai oleh garis berwarna ungu.

Strategi *forward chaining* diterapkan pada tahap *coding* untuk menentukan *input* apa yang akan direspon benar oleh chatbot, selain itu *forward chaining* juga mampu menentukan bagaimana alur jika *input* yang diberikan oleh pengguna tidak sesuai dengan kata kunci yang sudah dibuat oleh pengembang. Untuk proses detail dari setiap tahapan, dijelaskan pada sub bab berikut.

3.3 Tahapan *Extreme Programming* Pada Pembuatan Aplikasi Chatbot Kolektor

Berdasarkan gambar 3.1, proses pembuatan *chatbot* kolektor dibuat berdasarkan 4 tahapan metode *extreme programming*, di mana tahapan tersebut dibedakan menjadi 4 tahapan sesuai dengan warna garis yang ditunjukkan pada gambar 3.1. Adapun 4 tahapan *extreme programming* yang digunakan pada penelitian ini yaitu Tahap *Planning*, Tahap *Design*, Tahap *Coding*, dan Tahap *Testing*.

3.3.1 Tahap *Planning*

Tahapan *extreme programming* dimulai dengan tahap persiapan atau biasa disebut *planning*. Pada tahap *planning* di penelitian ini, terdapat 2(dua) kegiatan yang dilakukan sebagai tahap persiapan sebelum memulai membangun aplikasi *chatbot* kolektor, pengumpulan data dan analisa sistem yang sedang berjalan.

3.3.1.1 Pengumpulan Data

Tahapan penelitian yang dimulai dengan tahapan *planning*, pada tahap *planning* ini peneliti mengumpulkan kebutuhan yang diperlukan untuk membangun aplikasi *chatbot* kolektor. Teknik pengumpulan data yang dilakukan pada penelitian ini dengan observasi. Observasi dilakukan dengan melakukan wawancara langsung pada divisi *Remedial and Collection (RC)* selaku pemegang data, selain wawancara observasi juga dilakukan dengan mempelajari aplikasi yang saat ini berjalan di PT. Indomobil Finance Indonesia, yaitu SMSCenter. Pengumpulan data ini dilakukan untuk kebutuhan perancangan dan testing pada penelitian ini. Data yang digunakan pada

penelitian ini adalah *data primer* yang terdiri dari data kolektor dan data kendaraan yang terdaftar di PT. Indomobil Finance Indonesia.

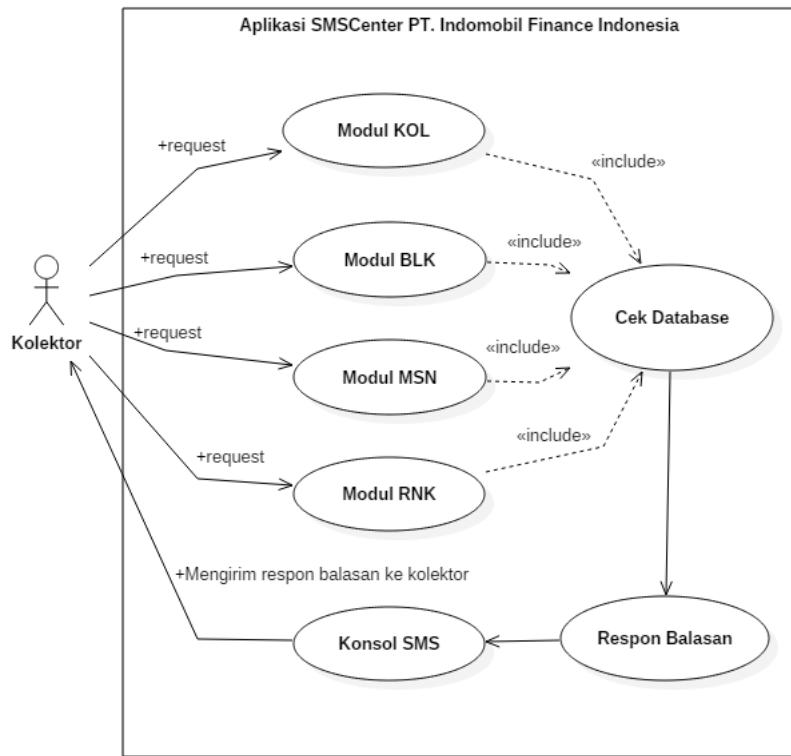
Pengumpulan data pada penelitian ini dilakukan dengan metode wawancara, wawancara dilakukan melalui *internal meeting* yang dilakukan di ruang *meeting* PT. Indomobil Finance Indonesia. *Internal Meeting* untuk persiapan pembuatan *chatbot* kolektor ini dihadiri oleh Ibu Nasya (Staff) dan Bapak Hengky (Kepala Departemen) dari divisi *Remedial and Collection* (RC) selaku pemegang data, dan Muhamad Sidik (Staff) serta Kresnoadi (Supervisor) dari divisi *Information and Technology* (IT) selaku pengembang aplikasi. Hasil dari internal meeting ini adalah kebutuhan user yakni sebuah aplikasi yang menyerupai aplikasi SMSCenter namun tidak bergantung pada *simcard*, hal ini dikarenakan seringnya terjadi masalah aplikasi SMSCenter karena adanya gangguan sinyal pada *simcard* maupun kondisi modem *hang*. Selain itu, kebutuhan ini juga bertujuan untuk melakukan penghematan pengeluaran biaya operasional tanpa mengurangi pelayanan yang ada.

Selain melalui wawancara kepada pemegang data, pengumpulan data juga dilakukan dengan melakukan observasi pada aplikasi berjalan, yaitu aplikasi SMSCenter. Tujuan dari observasi ini adalah memahami alur kerja dari aplikasi SMSCenter sebagai dasar pembuatan aplikasi *chatbot* kolektor bagi PT. Indomobil Finance Indonesia. Hasil dari observasi aplikasi SMSCenter dapat dilihat pada sub bab berikut.

3.3.1.2 Analisa Sistem yang sedang berjalan

Selain melakukan pengumpulan data, pada tahap *Planning* peneliti juga melakukan analisa terhadap aplikasi yang sudah tersedia, yaitu SMSCenter. Analisa ini dimaksudkan untuk memahami alur kerja dari permintaan layanan informasi oleh kolektor PT. Indomobil Finance Indonesia yang selama ini sudah berjalan. Selain memahami alur kerja, peneliti juga mempelajari masalah yang selama ini terjadi pada aplikasi SMSCenter sebagai pengetahuan yang solusinya akan diterapkan pada aplikasi *chatbot*, hal ini bertujuan agar permasalahan yang sama tidak terjadi pada aplikasi *chatbot* yang akan dikembangkan.

Untuk memenuhi kebutuhan permintaan layanan informasi, saat ini PT. Indomobil Finance Indonesia menggunakan sebuah aplikasi bernama SMSCenter. Gambar 3.2 merupakan Usecase Diagram cara kerja yang digunakan pada aplikasi SMSCenter PT. Indomobil Finance Indonesia.



Gambar 3.2: Use Case Diagram aplikasi SMSCenter PT. Indomobil Finance Indonesia

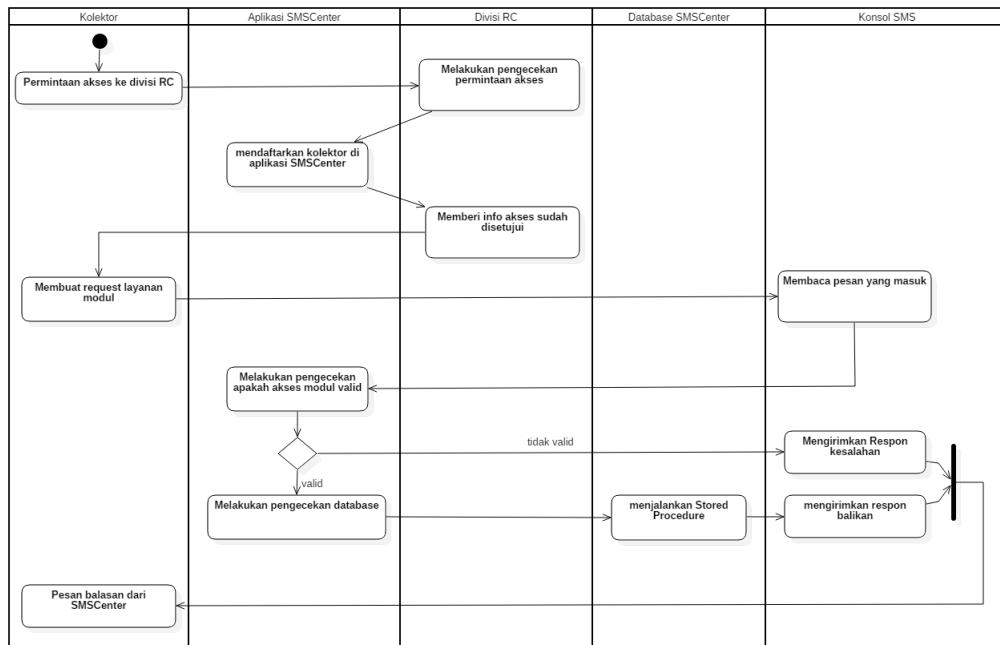
Berdasarkan gambar 3.2, arsitektur aplikasi SMSCenter yang saat ini berjalan di PT. Indomobil Finance Indonesia menunjukkan bagaimana cara kerja dari aplikasi SMSCenter tersebut. Didalam aplikasi SMSCenter, terdapat beberapa kelompok modul yang bisa diakses oleh kolektor, modul-modul ini kemudian akan melakukan pengecekan terhadap database untuk mendapatkan respon *message*. Database yang digunakan pada aplikasi SMSCenter ini adalah SQLServer, respon yang didapat dari *stored procedure* yang dijalankan oleh database kemudian akan diteruskan ke konsol sms. Konsol SMS ini yang bertugas untuk menerima pesan dari kolektor dan juga mengirimkan respon *message* yang didapat kepada kolektor dalam bentuk SMS(*short message service*).

Proses analisa aplikasi yang sedang berjalan terdiri dari 3 tahapan berikut.

- Analisa arsitektur Aplikasi SMSCenter.
- Analisa arsitektur teknologi yang digunakan.
- Analisa arsitektur Konsol SMS pada Aplikasi SMSCenter.

3.3.1.3 Analisa Arsitektur Aplikasi SMSCenter

Proses analisa arsitektur aplikasi SMSCenter dilakukan dengan mengam-barkan cara kerja aplikasi SMSCenter. cara kerja aplikasi SMSCenter ditunjukkan pada gambar 3.3.



Gambar 3.3: Arsitektur Aplikasi SMSCenter

Berdasarkan gambar 3.3, kolektor yang belum memiliki akses akan mengajukan form permintaan akses kepada divisi RC atau *Remedial and Collection*. Setelah mendapatkan form pengajuan akses, maka divisi RC akan melakukan pengecekan terlebih dahulu, jika memang data yang dikirimkan valid, maka divisi RC akan membuatkan akses bagi kolektor yang bersangkutan dan akan mengirimkan info bahwa kolektor tersebut sudah bisa melakukan request info by sms. Jika sudah memiliki akses, user mampu meminta informasi mengenai kendaraan dan calon kreditur melalui modul yang didaftarkan oleh divisi RC. Pesan yang masuk dari nomor kolektor akan dibaca oleh konsol sms untuk kemudian dicek valid atau tidaknya *request* di aplikasi SMSCenter. Jika *request* tidak valid, maka aplikasi SMSCenter akan mengirimkan balasan berupa respon kesalahan (*exception*). Jika *request* valid, maka *request* akan diteruskan ke *database* untuk kemudian diolah dalam bentuk *stored procedure* dan hasil dari *stored procedure* akan dikirimkan kembali melalui konsol sms untuk diteruskan ke kolektor.

3.3.1.4 Analisa arsitektur teknologi yang sedang berjalan

Analisa arsitektur teknologi dilakukan dengan mengidentifikasi teknologi-teknologi yang digunakan pada sistem yang sedang berjalan. Adapun analisa teknologi dilakukan pada bagian perangkat keras dan perangkat lunak yang digunakan pada sistem yang sedang berjalan. Tabel 3.1 adalah teknologi perangkat keras yang digunakan pada sistem yang sedang berjalan.

Tabel 3.1: Teknologi Perangkat Keras

No	Nama Perangkat Keras	Spesifikasi
1	Prosesor	Intel Pentium 4 6.0 GHz
2	RAM	1 GB
3	Disk	75 GB
4	Modem GSM	8 ports GSM MODEM WAVECOM
5	Sim Card	Kartu Halo

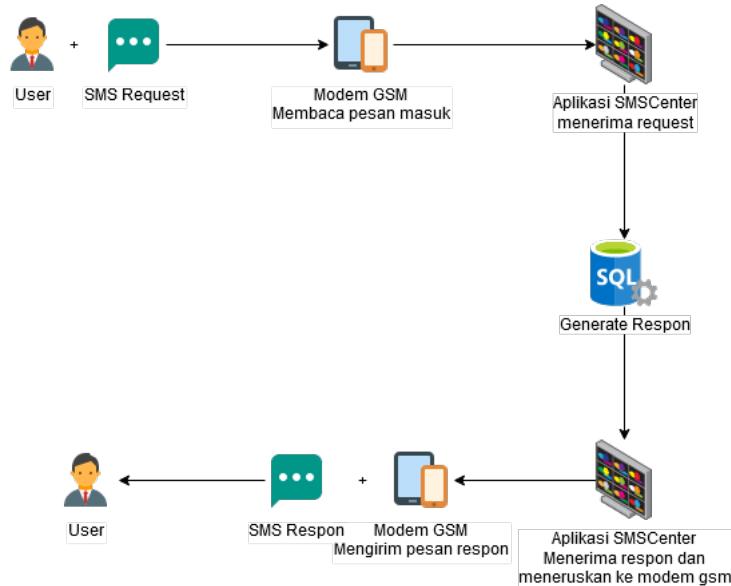
Tabel 3.2 adalah teknologi perangkat lunak yang digunakan pada sistem yang sedang berjalan.

Tabel 3.2: Teknologi Perangkat Lunak

No	Nama Perangkat Lunak	Spesifikasi
1	Bahasa pemrograman	VB.Net
2	Basis Data	SQLServer
3	OS Server	Windows Server 2003

3.3.1.5 Analisa arsitektur konsol pada Aplikasi SMSCenter

Analisa yang dilakukan pada Konsol ini berguna untuk mengetahui bagaimana flow dari pesan diterima, dibaca, dan diteruskan ke aplikasi SMSCenter. Selain itu, juga dianalisa bagaimana proses konsol ini menerima respon dari aplikasi SMSCenter untuk kemudian diteruskan ke kolektor dalam format sms.



Gambar 3.4: Arsitektur Konsol SMSCenter

Adapun tahapan-tahapan yang dilakukan berdasarkan gambar 3.4 adalah sebagai berikut.

1. Kolektor mengirimkan SMS ke nomor PT. Indomobil Finance Indonesia.
2. SMS yang dikirim user akan diterima oleh SIMCard yang terletak pada Modem WAVECOM.
3. Aplikasi Konsol SMSCenter akan membaca pesan melalui komunikasi antara aplikasi dengan Modem GSM.
4. Jika format sesuai dengan yang ditentukan, maka aplikasi akan melakukan pencarian data ke database SMSCenter.
5. Respon yang didapatkan dari database akan diteruskan kembali ke aplikasi Konsol SMSCenter.
6. Konsol SMSCenter akan mengirimkan respon yang didapatkan menuju simcard melalui modem GSM.
7. Simcard akan mengirimkan respon melalui sms kepada kolektor.



Gambar 3.5: Modem GSM untuk Konsol SMSCenter

3.3.2 Tahap *Design*

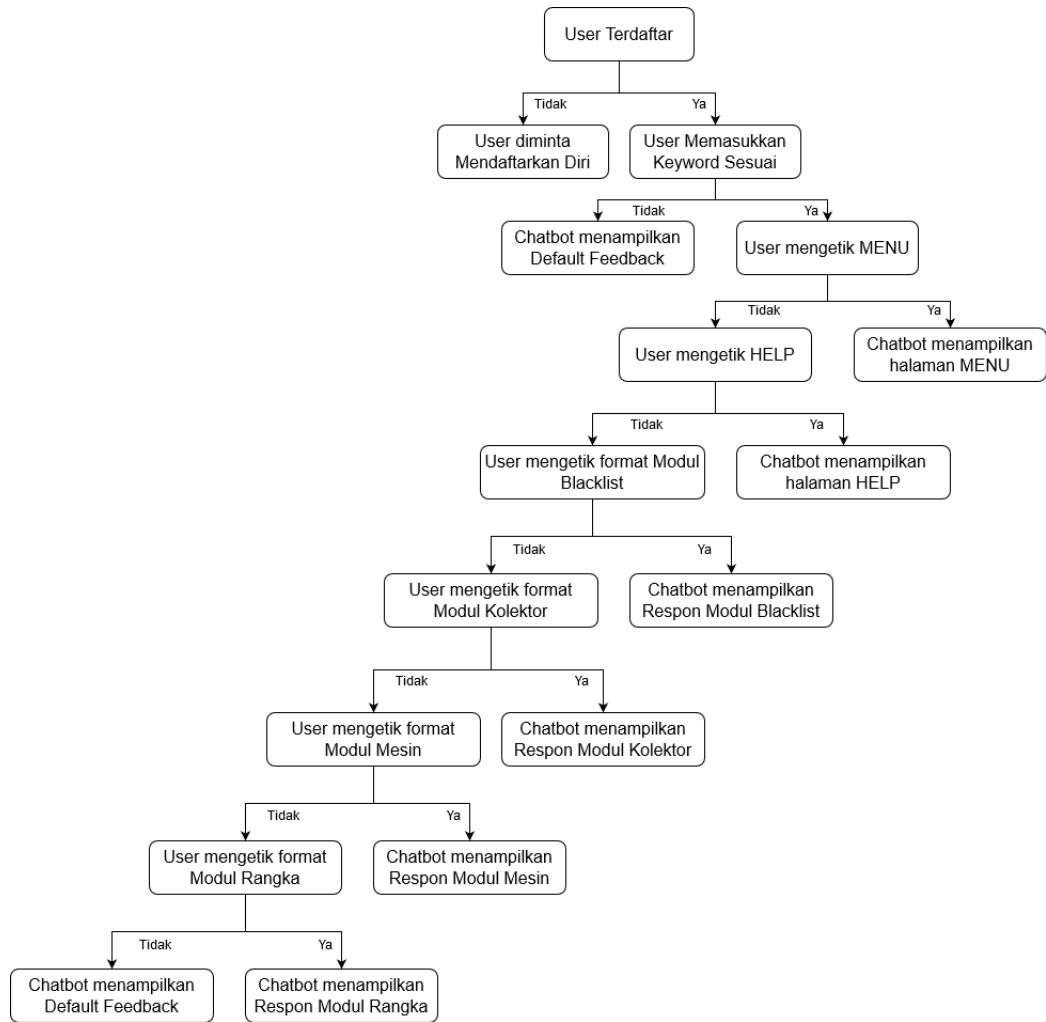
Proses *design* digunakan untuk memberikan gambaran awal tentang aplikasi *chatbot* yang akan dibuat, rancangan dibuat secara sederhana dan langsung diberikan kepada pihak pemegang data (*divisi Remedial and Collection*). Jika rancangan disetujui maka pembangunan aplikasi *chatbot* akan mengacu pada rancangan yang sudah disepakati pada tahap ini. Rancangan tampilan aplikasi *chatbot* kolektor pada penelitian ini dibangun dengan menggunakan aplikasi Balsamiq mockup. Rancangan juga dibuat untuk menentukan format *keyword* atau kata kunci yang akan memberikan respon benar jika diinput pada layar pesan Aplikasi Chatbot Kolektor. Format kata kunci pada Aplikasi Chatbot Kolektor dapat dilihat pada tabel 3.3 berikut.

Tabel 3.3: Tabel Format Kata Kunci Aplikasi Chatbot Kolektor

No	Nama Modul	Format	Contoh
1	Blacklist	BLK[spasi]JenisKonsumen [spasi]NoID	BLK K 3211234567890123
2	Kolektor	KOL[spasi]NoPolisi	KOL B1234ABC
3	Mesin	MSN[spasi]NoMesin	MSN 1234567890
4	Rangka	RNK[spasi]NoRangka	RNK 123456789012345678

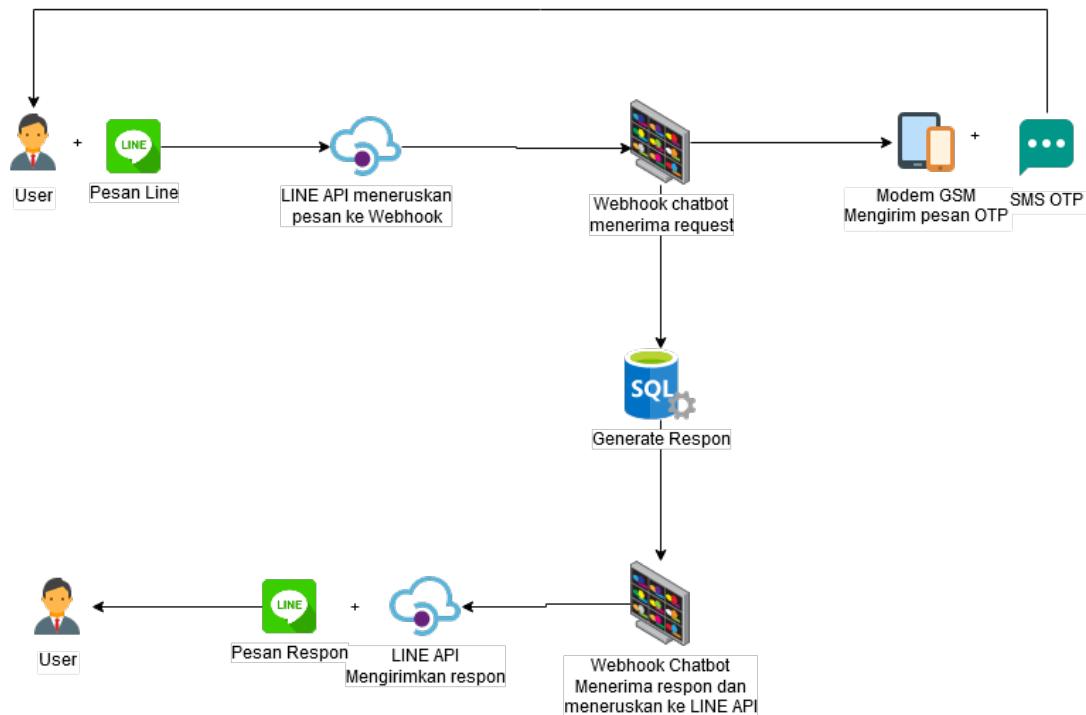
Pada tabel 3.3, Format yang ditentukan berbeda sesuai dengan masing-masing modul, format terdiri dari 3 karakter inisial masing-masing modul diikuti dengan *parameter* yang dibutuhkan untuk masing-masing modul. Untuk modul blacklist terdapat parameter jenis konsumen, jenis konsumen yang terdaftar pada modul ini adalah K untuk konsumen dengan pekerjaan sebagai karyawan, F untuk konsumen Profesional seperti Dokter dan TNI, W untuk konsumen dengan pekerjaan sebagai Wiraswasta, dan P untuk konsumen tingkat Perusahaan. Format pada Modul BLK diikuti dengan kode konsumen kemudian nomor id konsumen, terdapat modul KOL diikuti dengan nomor polisi kendaraan, modul MSN diikuti dengan nomor mesin kendaraan, dan modul RNK diikuti dengan nomor rangka kendaraan.

Pembuatan Chatbot Kolektor pada penelitian ini menerapkan strategi *forward chaining*, dengan menerapkan sistem “jika-maka” dalam strategi forward chaining, maka struktur alur aplikasi Chatbot Kolektor dapat dilihat pada gambar 3.6.



Gambar 3.6: Network Diagram Forward Chaining pada Aplikasi Chatbot Ko-lektor

3.3.2.1 Rancangan arsitektur Chatbot Kolektor



Gambar 3.7: Rancangan Arsitektur Chatbot Kolektor dengan platform LINE

Pada gambar 3.7 ditunjukkan rancangan arsitektur chatbot kolektor yang merupakan pengembangan dari arsitektur dari konsol SMSCenter. Perubahan dilakukan pada platform yang digunakan, yaitu dari sms menjadi menggunakan *LINE Messaging*. Jika dibandingkan dengan konsol SMSCenter, perbedaan terletak pada penerimaan serta pengiriman pesan dan respon. Jika pada konsol SMSCenter yang bertugas menerima dan mengirimkan pesan adalah simcard yang terletak pada GSM Modem, maka pada Chatbot Kolektor yang bertugas menerima dan mengirim pesan adalah *messaging API* yang disediakan dari platform LINE.

Messaging API LINE kemudian akan melakukan interaksi dengan *chatbot webhook* yang sudah dikonfigurasi, kemudian *chatbot webhook* akan memproses permintaan yang dikirim oleh *user* melalui *messaging API* LINE jika permintaan dari *user* sesuai dengan format yang ditentukan. Permintaan *user* akan diproses dengan menjalankan *stored procedure* yang sudah disiapkan di *database*. *Stored procedure* ini kemudian akan menghasilkan suatu respon yang akan dikirimkan kembali ke *chatbot webhook*. *chatbot webhook* kemudian mengirimkan respon tersebut kepada *messaging API* LINE untuk diteruskan dan dikirimkan kembali kepada *user*.

Adapun penggunaan *simcard* dan modem GSM pada arsitektur ini berfungsi untuk mengirimkan kode *OTP* ketika *user* pertama kali mendaftarkan diri di Aplikasi Chatbot Kolektor.

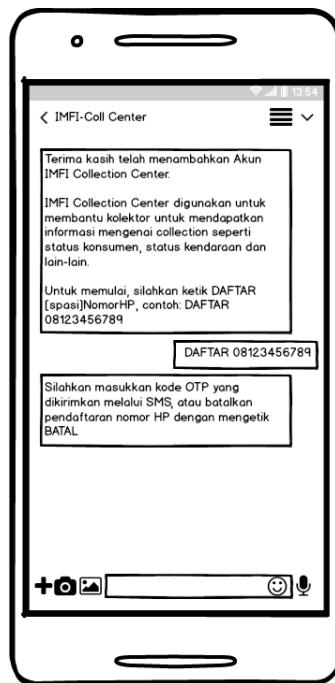
3.3.2.2 Rancangan tampilan *chatbot* ketika ditambahkan menjadi Teman.



Gambar 3.8: Rancangan tampilan Chatbot Kolektor ketika ditambahkan sebagai Teman

Tampilan pada gambar 3.8 adalah tampilan ketika akun Chatbot Kolektor ditambahkan menjadi Teman oleh *user*. Pesan otomatis akan selalu ditampilkan setiap kali akun LINE Official Chatbot Kolektor ditambahkan oleh pengguna LINE.

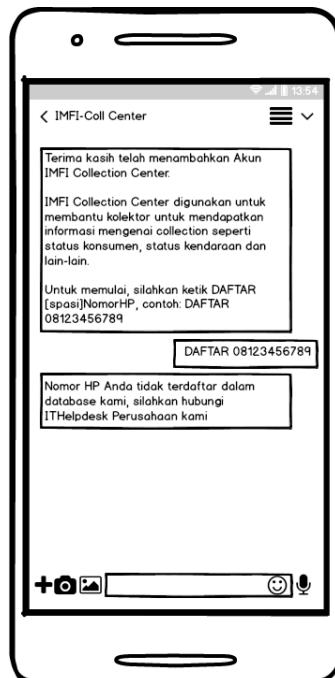
3.3.2.3 Rancangan tampilan *chatbot* ketika *user valid* melakukan pendaftaran.



Gambar 3.9: Rancangan tampilan Chatbot Kolektor ketika *user valid* melakukan Pendaftaran

Tampilan pada gambar 3.9 adalah rancangan tampilan ketika *user valid* melakukan pendaftaran. *User valid* yang dimaksud adalah user yang nomor *handphone* nya sudah terdaftar sebagai kolektor di aplikasi SMSCenter. Respon dari aksi yang dilakukan *user* pada halaman ini adalah *chatbot* akan menampilkan pesan permintaan kepada *user* untuk melakukan pengecekan sms. Pengecekan sms dilakukan untuk memeriksa kode *OTP* yang sudah dikirimkan sebagai bentuk validasi akan pendaftaran yang dilakukan.

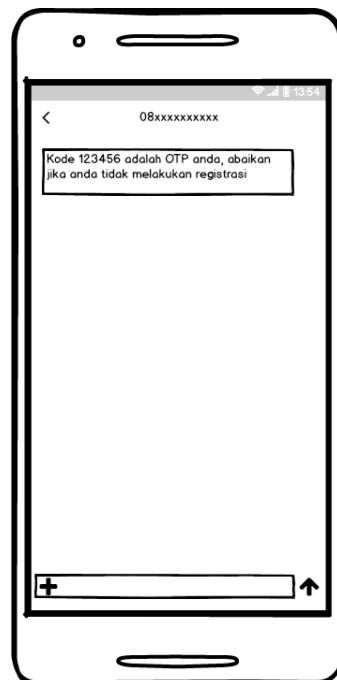
3.3.2.4 Rancangan tampilan *chatbot* ketika *user* tidak valid melakukan pendaftaran.



Gambar 3.10: Rancangan tampilan Chatbot Kolektor ketika *user* tidak valid melakukan Pendaftaran

Tampilan pada gambar 3.10 adalah rancangan tampilan ketika *user* tidak valid melakukan pendaftaran. *User* tidak valid yang dimaksud adalah user yang nomor *handphone* nya tidak terdaftar di database aplikasi SMSCenter. Respon dari aksi yang dilakukan *user* pada halaman ini adalah *chatbot* akan menampilkan pesan permintaan kepada *user* untuk menghubungi bagian ITHelpdesk dan melakukan konfirmasi untuk mendaftarkan nomor *handphone*nya terlebih dahulu.

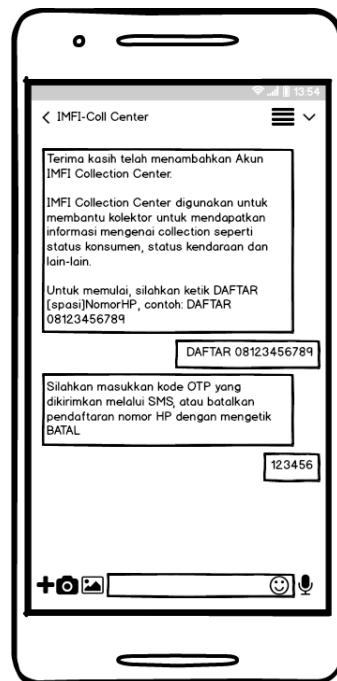
3.3.2.5 Rancangan tampilan contoh kode *OTP* yang dikirimkan kepada *user*.



Gambar 3.11: Rancangan tampilan sms berisi kode *OTP* yang dikirimkan oleh chatbot kolektor

Tampilan pada gambar 3.11 adalah rancangan tampilan isi pesan berupa kode *OTP* yang akan dikirimkan ke nomor *handphone* yang didaftarkan oleh kolektor untuk pendaftaran *Chatbot Kolektor*. *User* akan diminta untuk memasukkan kode *OTP* yang tertera pada pesan ini untuk memvalidasi jika pendaftaran benar dilakukan oleh *user* yang bersangkutan.

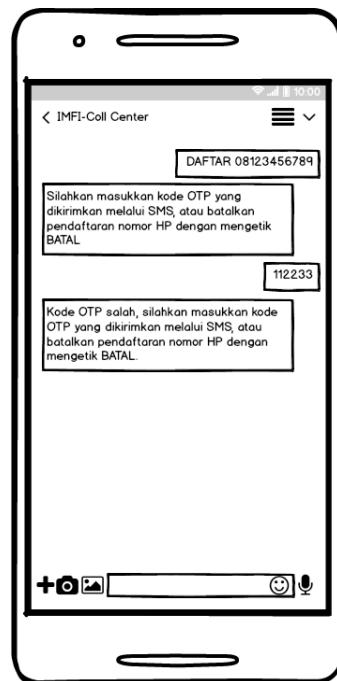
3.3.2.6 Rancangan tampilan *user* memasukkan kode *OTP*.



Gambar 3.12: Rancangan tampilan ketika *user* memasukkan kode *OTP*

Tampilan pada gambar 3.12 adalah rancangan tampilan ketika *user* memasukkan kode *OTP* yang sebelumnya sudah dikirimkan melalui *SMS* ke nomor yang didaftarkan oleh *user*.

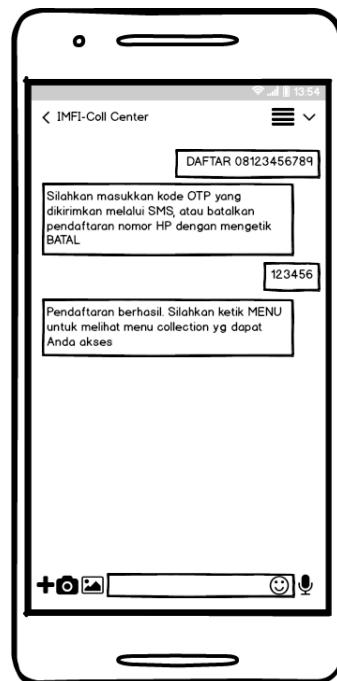
3.3.2.7 Rancangan tampilan notifikasi kode *OTP* salah.



Gambar 3.13: Rancangan tampilan notifikasi *OTP* salah

Tampilan pada gambar 3.13 adalah rancangan tampilan notifikasi jika *user* memasukkan kode *OTP* yang salah. *User* akan diminta untuk memasukkan kode *OTP* yang dikirimkan oleh chatbot kolektor melalui *SMS*.

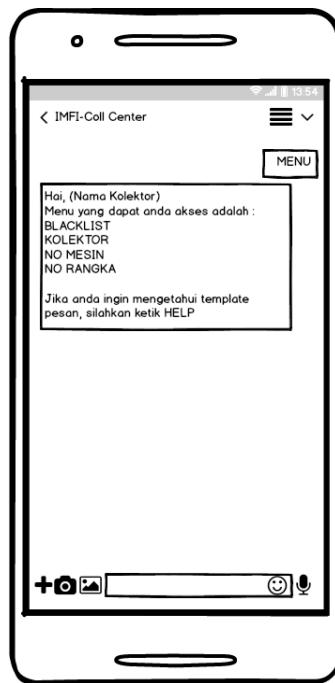
3.3.2.8 Rancangan tampilan notifikasi pendaftaran *user*.



Gambar 3.14: Rancangan tampilan notifikasi pendaftaran

Tampilan pada gambar 3.14 adalah rancangan tampilan notifikasi setelah *user* berhasil memasukkan kode *OTP* yang sesuai dengan kode *OTP* yang dikirimkan oleh chatbot kolektor.

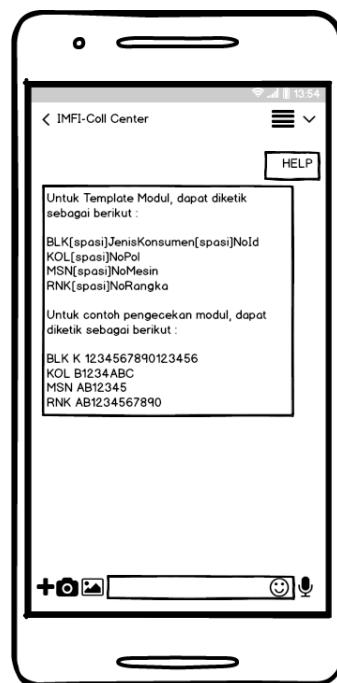
3.3.2.9 Rancangan tampilan menu *user*.



Gambar 3.15: Rancangan tampilan halaman menu *user*

Tampilan pada gambar 3.15 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* memasukkan kata MENU pada halaman pesan *chatbot*. Respon yang akan diberikan oleh *chatbot* adalah modul apa saja yang dapat diakses oleh *user*, serta memberikan suggest ketik HELP jika *user* ingin mengetahui format pesan yang ditentukan.

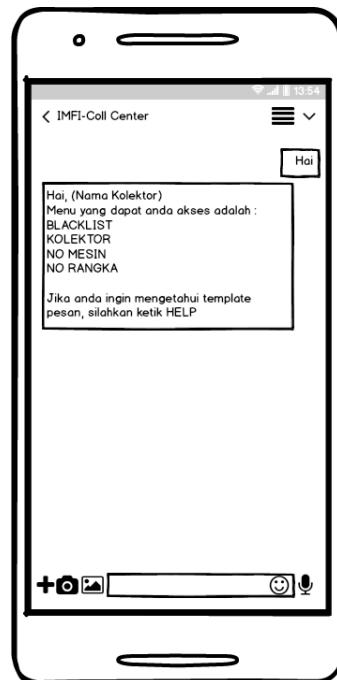
3.3.2.10 Rancangan tampilan *help user*.



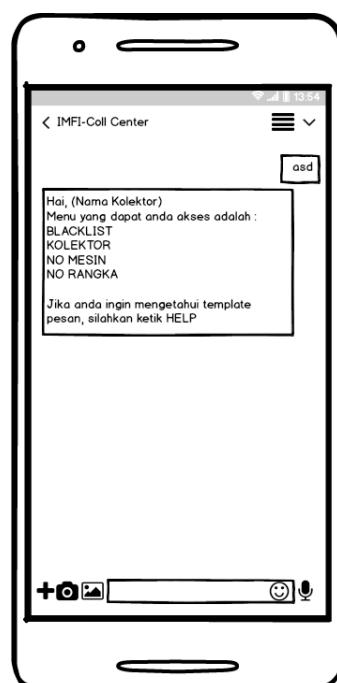
Gambar 3.16: Rancangan tampilan halaman *help user*

Tampilan pada gambar 3.16 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* memasukkan kata HELP pada halaman pesan *chatbot*. Respon yang akan diberikan oleh *chatbot* adalah *template* pesan yang dapat digunakan oleh *user* untuk meminta infomasi layanan sesuai dengan modul yang dapat diakses oleh *user*.

3.3.2.11 Rancangan tampilan *user* jika memasukkan kata yang bukan termasuk kata kunci.



Gambar 3.17: Rancangan tampilan *user* ketika *user* memasukkan kata “hai”

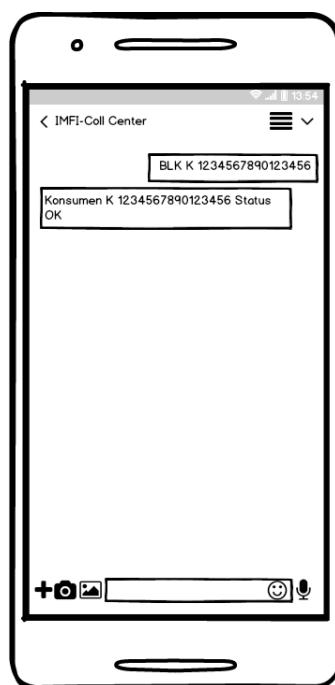


Gambar 3.18: Rancangan tampilan *user* ketika *user* memasukkan kata asal

Tampilan pada gambar 3.17 dan gambar 3.18 adalah rancangan tampilan

respon *chatbot* kolektor ketika *user* memasukkan kata yang tidak termasuk dalam kata kunci pada halaman pesan *chatbot*. Respon yang akan diberikan oleh *chatbot* adalah pesan menu yang berisikan modul apa saja yang dapat diakses oleh *user*, serta memberikan *suggest* ketik HELP jika user ingin mengetahui format pesan yang ditentukan.

3.3.2.12 Rancangan tampilan respon modul BLK (*Blacklist*) bila status calon konsumen OK.



Gambar 3.19: Rancangan tampilan respon modul BLK konsumen OK

Tampilan pada gambar 3.19 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul BLK (*Blacklist*) dengan memasukkan keyword BLK + (kode konsumen) + Nomor NIK calon konsumen. Jika NIK calon konsumen tidak ada dalam *database* konsumen dan tidak ada dalam *database* konsumen *blacklist* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon “Status OK”, yang artinya calon konsumen tersebut dapat diproses menjadi konsumen di PT. Indomobil Finance Indonesia.

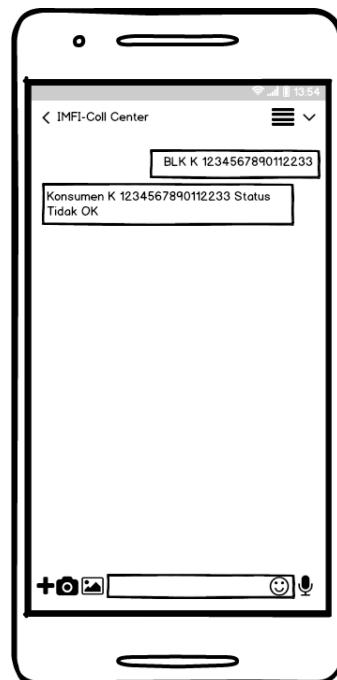
3.3.2.13 Rancangan tampilan respon modul BLK (*Blacklist*) bila status calon konsumen sudah pernah menjadi konsumen sebelumnya.



Gambar 3.20: Rancangan tampilan respon modul BLK calon konsumen pernah menjadi konsumen

Tampilan pada gambar 3.20 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul BLK (*Blacklist*) dengan memasukkan keyword BLK + (kode konsumen) + Nomor NIK calon konsumen. Jika NIK calon konsumen ada dalam *database* konsumen dan tidak ada dalam *database* konsumen *blacklist* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi mengenai konsumen tersebut, yang artinya calon konsumen tersebut dapat diproses menjadi konsumen di PT. Indomobil Finance Indonesia.

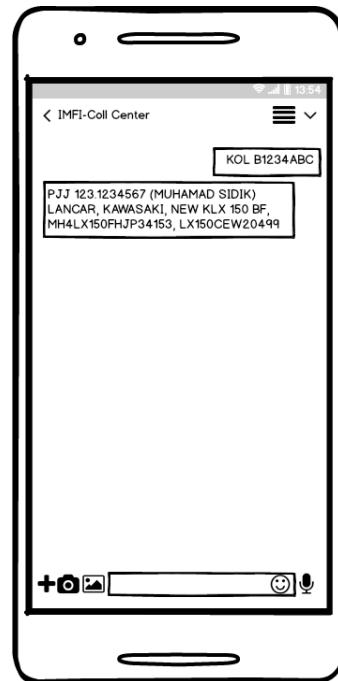
3.3.2.14 Rancangan tampilan respon modul BLK (*Blacklist*) bila status calon konsumen tidak OK.



Gambar 3.21: Rancangan tampilan respon modul BLK calon konsumen tidak OK

Tampilan pada gambar 3.21 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul BLK (*Blacklist*) dengan memasukkan keyword BLK + (kode konsumen) + Nomor NIK calon konsumen. Jika NIK calon konsumen ada dalam *database* konsumen dan ada dalam *database* konsumen *blacklist* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa calon konsumen tersebut “Tidak OK”, yang artinya calon konsumen tersebut tidak dapat diproses menjadi konsumen di PT. Indomobil Finance Indonesia.

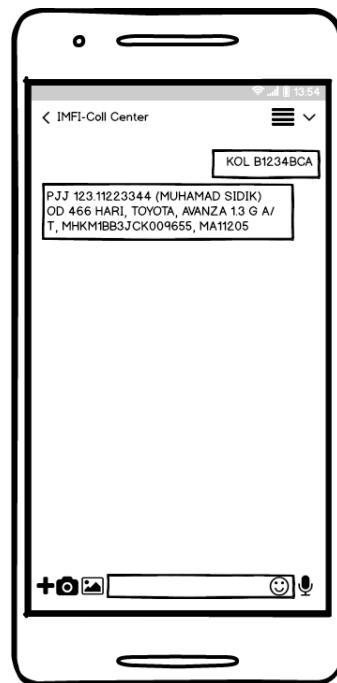
3.3.2.15 Rancangan tampilan respon modul KOL (Kolektor) bila status lancar.



Gambar 3.22: Rancangan tampilan respon modul KOL lancar

Tampilan pada gambar 3.22 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan tidak memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nomor PJJ (Perjanjian) konsumen, nama konsumen, serta detail unit yang diambil oleh konsumen.

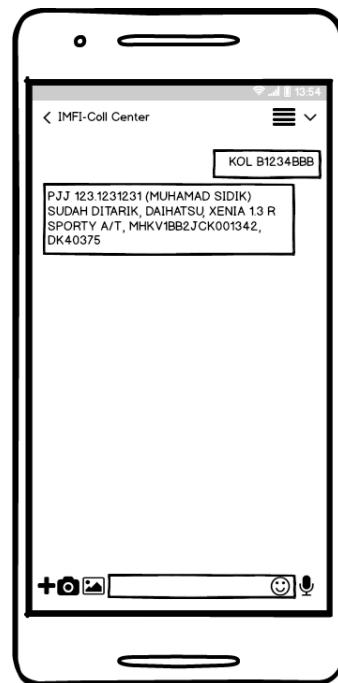
3.3.2.16 Rancangan tampilan respon modul KOL (Kolektor) bila status overdue.



Gambar 3.23: Rancangan tampilan respon modul KOL *overdue*

Tampilan pada gambar 3.23 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nomor PJJ (Perjanjian) konsumen, nama konsumen, lama tunggakan dalam satuan hari, serta detail unit yang diambil oleh konsumen.

3.3.2.17 Rancangan tampilan respon modul KOL (Kolektor) bila status ditarik.



Gambar 3.24: Rancangan tampilan respon modul KOL *overdue*

Tampilan pada gambar 3.24 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan ada dalam database PT. Indomobil Finance Indonesia dan unit sudah ditarik oleh kolektor. Maka *chatbot* akan memberikan respon informasi mengenai detail nomor PJJ (Perjanjian) konsumen, nama konsumen, status jika unit sudah ditarik, serta detail unit yang diambil oleh konsumen.

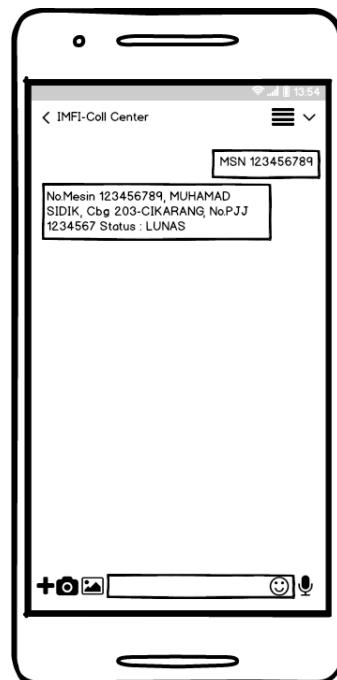
3.3.2.18 Rancangan tampilan respon modul KOL (Kolektor) bila data tidak ditemukan.



Gambar 3.25: Rancangan tampilan respon modul KOL *overdue*

Tampilan pada gambar 3.25 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan tidak ada dalam *database* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa Nomor Polisi yang dimasukkan tidak terdaftar dalam *database* PT. Indomobil Finance Indonesia (bukan unit yang diambil melalui PT. Indomobil Finance Indonesia).

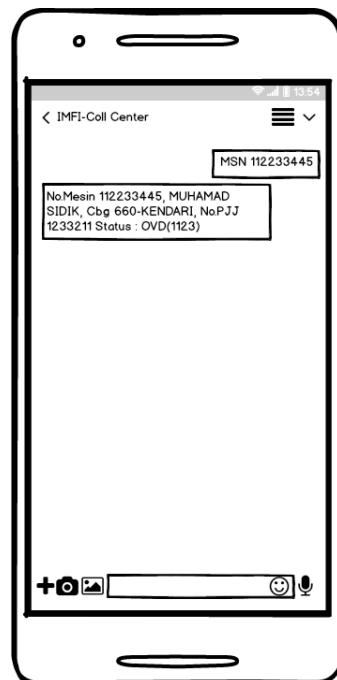
3.3.2.19 Rancangan tampilan respon modul MSN (Mesin) bila data OK.



Gambar 3.26: Rancangan tampilan respon modul MSN OK

Tampilan pada gambar 3.26 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul MSN (Mesin) dengan memasukkan keyword MSN + Nomor Mesin kendaraan. Jika No. Mesin Kendaraan yang dimasukkan ada dalam database PT. Indomobil Finance Indonesia dan tidak memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut sudah lunas.

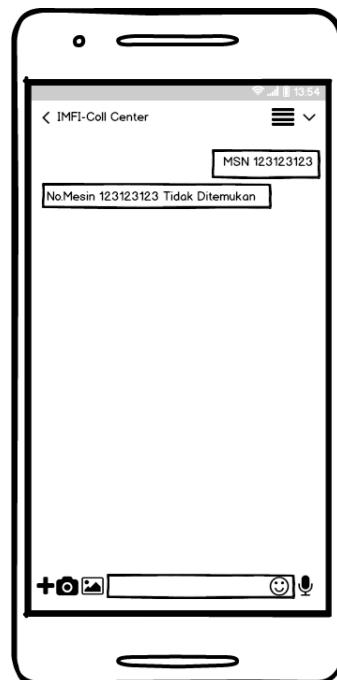
3.3.2.20 Rancangan tampilan respon modul MSN (Mesin) bila data overdue.



Gambar 3.27: Rancangan tampilan respon modul MSN *overdue*

Tampilan pada gambar 3.27 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul MSN (Mesin) dengan memasukkan keyword MSN + Nomor Mesin kendaraan. Jika No. Mesin Kendaraan yang dimasukkan ada dalam database PT. Indomobil Finance Indonesia dan memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut memiliki lama tunggakan dalam satuan hari.

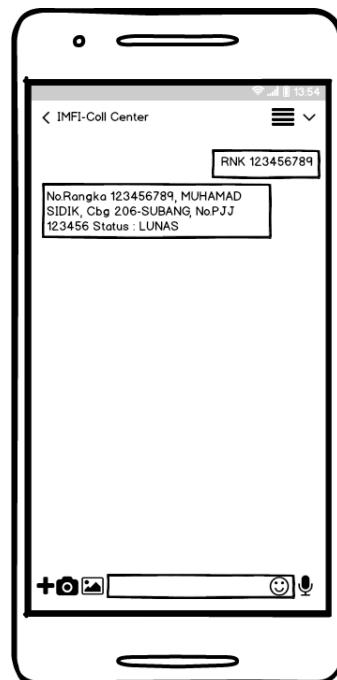
3.3.2.21 Rancangan tampilan respon modul MSN (Mesin) bila data tidak ditemukan.



Gambar 3.28: Rancangan tampilan respon modul MSN tidak ditemukan

Tampilan pada gambar 3.28 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul MSN (Mesin) dengan memasukkan keyword MSN + Nomor Mesin kendaraan. Jika No. Mesin Kendaraan yang dimasukkan tidak ada dalam *database* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa Nomor Mesin Kendaraan yang dimasukkan tidak terdaftar dalam *database* PT. Indomobil Finance Indonesia (bukan unit yang diambil melalui PT. Indomobil Finance Indonesia).

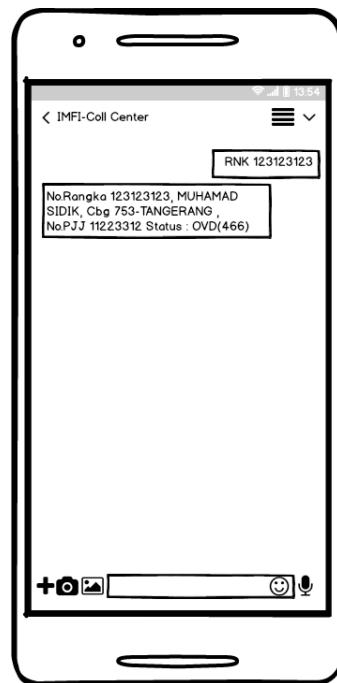
3.3.2.22 Rancangan tampilan respon modul RNK (Rangka) bila data OK.



Gambar 3.29: Rancangan tampilan respon modul RNK data OK

Tampilan pada gambar 3.29 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul RNK (Rangka) dengan memasukkan keyword RNK + Nomor Rangka kendaraan. Jika No. Rangka Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan tidak memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut sudah lunas.

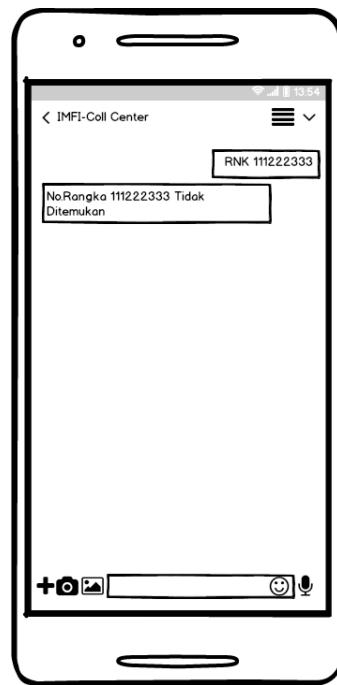
3.3.2.23 Rancangan tampilan respon modul RNK (Rangka) bila data *overdue*.



Gambar 3.30: Rancangan tampilan respon modul RNK data *overdue*

Tampilan pada gambar 3.30 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul RNK (Rangka) dengan memasukkan keyword RNK + Nomor Rangka kendaraan. Jika No. Rangka Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut memiliki lama tunggakan dalam satuan hari.

3.3.2.24 Rancangan tampilan respon modul RNK (Rangka) bila data tidak ditemukan.



Gambar 3.31: Rancangan tampilan respon modul RNK data *overdue*

Tampilan pada gambar 3.31 adalah rancangan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul RNK (Rangka) dengan memasukkan keyword RNK + Nomor Rangka kendaraan. Jika No. Rangka Kendaraan yang dimasukkan tidak ada dalam *database* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa Nomor Rangka Kendaraan yang dimasukkan tidak terdaftar dalam *database* PT. Indomobil Finance Indonesia (bukan unit yang diambil melalui PT. Indomobil Finance Indonesia).

3.3.3 Tahap Pembuatan Coding

Pada tahap *coding*, secara umum peneliti memecah kegiatan menjadi 4 bagian umum, yaitu :

1. Pembuatan *web service*, pembuatan *web service* bertujuan untuk membuat suatu wadah sebagai pintu gerbang komunikasi antara *chatbot* dengan *knowledge base* yang ada di sistem internal di PT. Indomobil Finance Indonesia, pembuatan *web service* ini menggunakan bantuan

tools visual studio 2010 dengan bahasa pemrograman visual basic Net. 2010.

2. Pembuatan *Knowledge Base*, pembuatan *knowledge base* merupakan bagian inti dari pembuatan sebuah *chatbot*, karena dengan *knowledge base* ini-lah *chatbot* akan mampu memberikan respon informasi yang dibutuhkan oleh pengguna. Pembuatan *knowledge base* pada penelitian ini memanfaatkan *database* yang sudah ada, dengan menjalankan *store procedure* yang sudah disiapkan pada *database* maka hasil dari *query store procedure* tersebut yang kemudian akan diteruskan menjadi respon dari masukkan yang *diinput* oleh pengguna.
3. Konfigurasi *Webhook*, konfigurasi webhook pada penelitian ini dilakukan untuk menghubungkan *web service* dengan layanan *API LINE*. Untuk bisa melakukan konfigurasi, maka dibutuhkan akun line developer, di-dalam akun line developer ini juga peneliti melakukan konfigurasi lain untuk *chatbot*, misal ikon *chatbot*, nama *chatbot*, dan lain-lain.
4. Pembuatan *Logic Program*, pembuatan *logic program* dilakukan pada *web service* yang sudah disiapkan sebelumnya, pada proses pembuatan *logic program* ini strategi *forward chaining* diterapkan. Dengan menerapkan strategi jika-maka yang ada pada *forward chaining*, seluruh *input* yang akan dimasukan oleh *user* untuk berinteraksi dengan *chatbot* akan ditentukan pada tahap ini, respon yang sudah disiapkan pada *knowledge base* juga akan diatur sebagai respon dalam tahap ini.

- Strategi Forward Chaining

Strategi forward chaining pada penelitian ini digunakan untuk menentukan alur atau arah interaksi yang dilakukan oleh *chatbot* dengan pengguna. Setelah *user/client* menambahkan *chatbot* kolektor sebagai teman maka bot akan menampilkan pesan pembuka. Kemudian *user* akan melakukan *input* kata kunci yang nantinya akan diproses oleh *chatbot*, apabila kata kunci sesuai maka akan langsung direspon oleh *chatbot*, apabila tidak sesuai maka *chatbot* akan menampilkan “*default feedback*” dengan chat kata-kata yang diinginkan, semisal “saya tidak mengerti maksud anda?”, “mohon masukkan kata kunci yang sesuai”, dll. Setelah ditampilkan hasil dari masukan tersebut, *user* bisa memasukkan kata kunci kembali atau mengakhiri percakapan.

3.3.4 Tahap *Testing* Aplikasi

Setelah proses pembuatan aplikasi *chatbot* selesai, maka untuk memastikan *chatbot* sudah siap digunakan dibutuhkan proses testing, proses *testing* dilakukan dengan metode *blackbox testing*. *Testing* dikerjakan secara bertahap sesuai dengan kasus yang dirasa perlu diuji, dari mulai menambahkan *chatbot* sebagai teman, melakukan *input* yang sesuai dengan kata kunci, hingga melakukan *input* yang tidak sesuai dengan kata kunci. Selain testing secara *blackbox*, untuk memastikan bahwa pengembangan aplikasi pada penelitian ini sudah sesuai dengan keinginan user, maka pada penelitian ini dilakukan *user acceptance test (UAT)*.

3.4 Pengujian Perangkat Lunak

Setelah dilakukan proses pembuatan *chatbot* kolektor, untuk memastikan *chatbot* memberikan respon sesuai dengan harapan maka diperlukan pengujian perangkat lunak. Pengujian perangkat lunak pada penelitian ini menggunakan *blackbox testing*. Pengujian dengan menggunakan *blackbox testing* ini dilakukan dengan cara memberikan *input* langsung kepada *chatbot* kolektor, baik *input* yang sesuai dengan aturan maupun *input* yang tidak sesuai dengan aturan. Testing juga dilakukan dengan mengajukan form *User Acceptance Test (UAT)* untuk diberikan kepada *user*, hal ini bertujuan untuk memastikan bahwa aplikasi yang dibangun sudah sesuai dengan yang *user* inginkan.

Bab 4

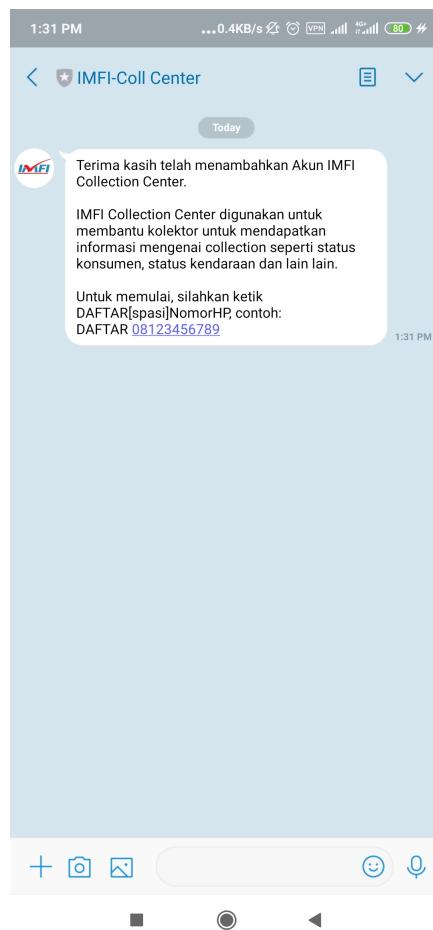
HASIL PENELITIAN DAN PEMBAHASAN

4.1 Implementasi Aplikasi Chatbot Kolektor

Setelah tahap *planning* selesai dikerjakan, maka tahap berikutnya yang dilakukan adalah tahap *design*. Tahap *design* dibuat berdasarkan hasil dari pertemuan dengan *user* pada tahap *planning*, hasil dari tahap *design* ini akan menjadi acuan untuk tahap berikutnya yaitu tahap *coding*.

Tampilan yang akan dibuat pada tahap *design* ini adalah tampilan aplikasi secara keseluruhan, dari mulai kolektor menambahkan LINE *chatbot* kolektor menjadi teman, hingga semua *input message* dari kolektor dan respon yang diberikan oleh *chatbot* kolektor. Hasil tampilan dari tahap *design* yang dilakukan pada penelitian ini adalah sebagai berikut :

4.1.1 Tampilan *chatbot* ketika ditambahkan menjadi Teman.



Gambar 4.1: Tampilan Chatbot Kolektor ketika ditambahkan sebagai Teman

Gambar 4.1 menunjukkan tampilan ketika akun Chatbot Kolektor ditambahkan menjadi Teman oleh *user*. Pesan otomatis akan selalu ditampilkan setiap kali akun LINE Official Chatbot Kolektor ditambahkan oleh pengguna LINE.

4.1.2 Tampilan *chatbot* ketika *user* valid melakukan pendaftaran.



Gambar 4.2: Tampilan Chatbot Kolektor ketika *user* valid melakukan Pendaf-
taran

Gambar 4.2 menunjukkan tampilan ketika *user* valid melakukan pendaftaran. *User* valid yang dimaksud adalah *user* yang nomor *handphone* nya sudah terdaftar sebagai kolektor di aplikasi SMSCenter. Respon dari aksi yang dilakukan *user* pada halaman ini adalah *chatbot* akan menampilkan pesan permintaan kepada *user* untuk melakukan pengecekan sms. Pengecekan sms dilakukan untuk memeriksa kode *OTP* yang sudah dikirimkan sebagai bentuk validasi akan pendaftaran yang dilakukan.

4.1.3 Tampilan *chatbot* ketika *user* tidak valid melakukan pendaftaran.



Gambar 4.3: Tampilan Chatbot Kolektor ketika *user* tidak valid melakukan Pendaftaran

Gambar 4.3 menunjukkan tampilan ketika *user* tidak valid melakukan pendaftaran. *User* tidak valid yang dimaksud adalah *user* yang nomor *handphone* nya tidak terdaftar di database aplikasi SMSCenter. Respon dari aksi yang dilakukan *user* pada halaman ini adalah *chatbot* akan menampilkan pesan permintaan kepada *user* untuk menghubungi bagian ITHelpdesk dan melakukan konfirmasi untuk mendaftarkan nomor *handphonanya* terlebih dahulu.

4.1.4 Tampilan contoh kode *OTP* yang dikirimkan kepada user.



Gambar 4.4: Tampilan sms berisi kode *OTP* yang dikirimkan oleh chatbot kolektor

Gambar 4.4 menunjukkan tampilan isi pesan berupa kode *OTP* yang akan dikirimkan ke nomor *handphone* yang didaftarkan oleh kolektor untuk pendaftaran *Chatbot Kolektor*. *User* akan diminta untuk memasukkan kode *OTP* yang tertera pada pesan ini untuk memvalidasi jika pendaftaran benar dilakukan oleh *user* yang bersangkutan.

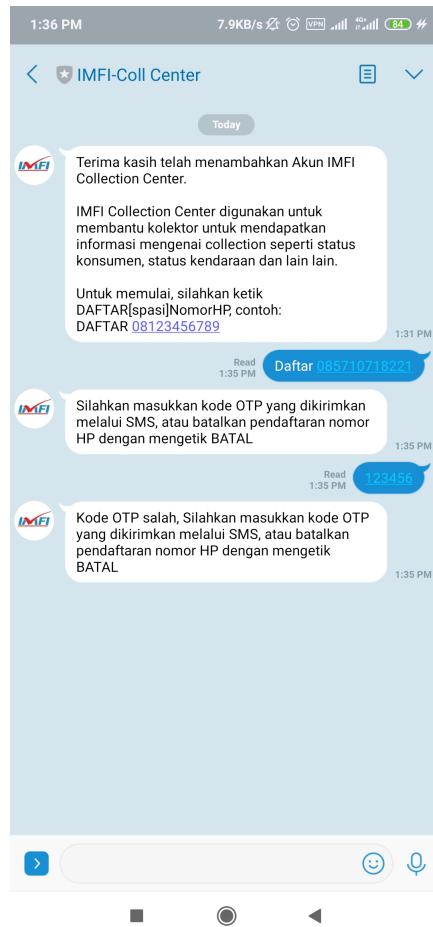
4.1.5 Tampilan *user* memasukkan kode *OTP*.



Gambar 4.5: Tampilan ketika *user* memasukkan kode *OTP*

Gambar 4.5 menunjukkan tampilan ketika *user* memasukkan kode *OTP* yang sebelumnya sudah dikirimkan melalui *SMS* ke nomor yang didaftarkan oleh *user*.

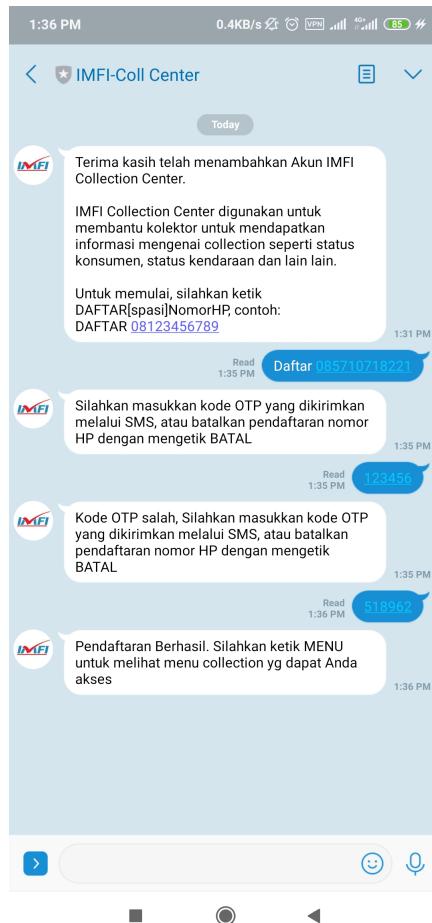
4.1.6 Tampilan notifikasi kode OTP salah.



Gambar 4.6: Tampilan notifikasi OTP salah

Gambar 4.6 menunjukkan tampilan notifikasi jika *user* memasukkan kode *OTP* yang salah. *User* akan diminta untuk memasukkan kode *OTP* yang dikirimkan oleh chatbot kolektor melalui *SMS*.

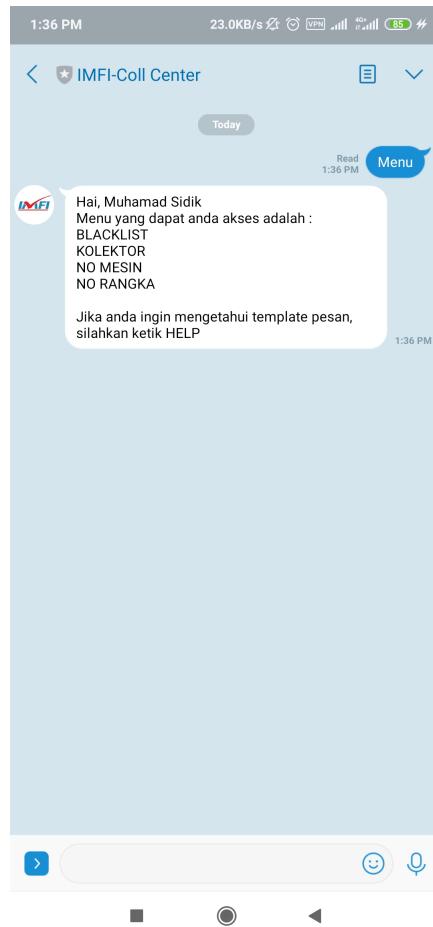
4.1.7 Tampilan notifikasi pendaftaran user.



Gambar 4.7: Tampilan notifikasi pendaftaran berhasil

Gambar 4.7 menunjukkan tampilan notifikasi setelah *user* berhasil memasukkan kode *OTP* yang sesuai dengan kode *OTP* yang dikirimkan oleh chatbot kolektor.

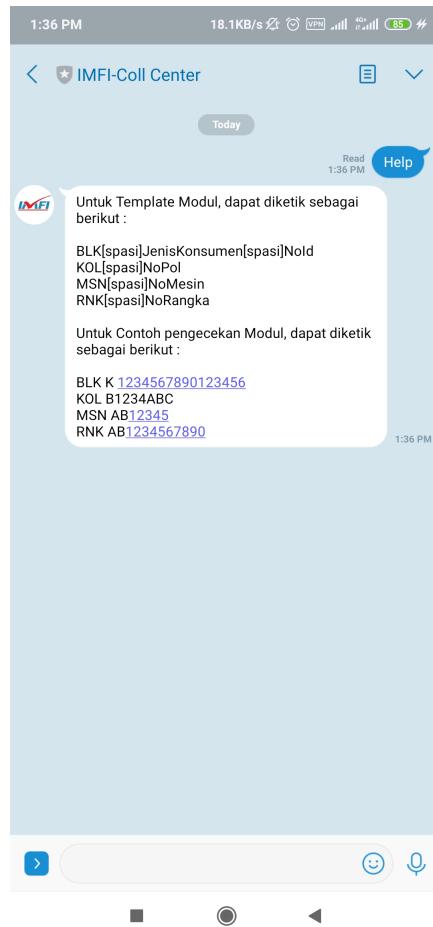
4.1.8 Tampilan menu *user*.



Gambar 4.8: Tampilan halaman menu *user*

Gambar 4.8 menunjukkan tampilan respon *chatbot* kolektor ketika *user* memasukkan kata MENU pada halaman pesan *chatbot*. Respon yang akan diberikan oleh *chatbot* adalah modul apa saja yang dapat diakses oleh *user*, serta memberikan *suggest* ketik HELP jika user ingin mengetahui format pesan yang ditentukan.

4.1.9 Tampilan *help user*.



Gambar 4.9: Tampilan halaman *help user*

Gambar 4.9 menunjukkan tampilan respon *chatbot* kolektor ketika *user* memasukkan kata HELP pada halaman pesan *chatbot*. Respon yang akan diberikan oleh *chatbot* adalah *template* pesan yang dapat digunakan oleh *user* untuk meminta infomasi layanan sesuai dengan modul yang dapat diakses oleh *user*.

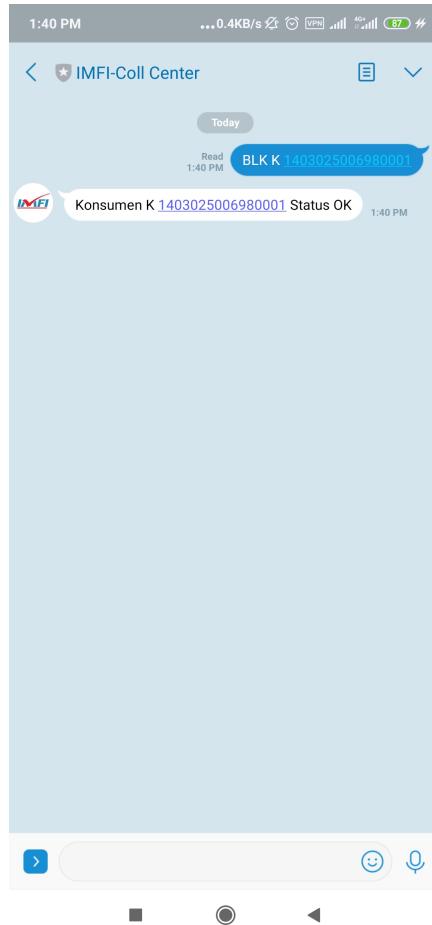
4.1.10 Tampilan *user* jika memasukkan kata yang bukan termasuk kata kunci.



Gambar 4.10: Tampilan *user* ketika *user* memasukkan kata “hai”

Gambar 4.10 menunjukkan tampilan respon *chatbot* kolektor ketika *user* memasukkan kata yang tidak termasuk dalam kata kunci pada halaman pesan *chatbot*. Respon yang akan diberikan oleh *chatbot* adalah pesan menu yang berisikan modul apa saja yang dapat diakses oleh *user*, serta memberikan *suggest* ketik HELP jika *user* ingin mengetahui format pesan yang ditentukan.

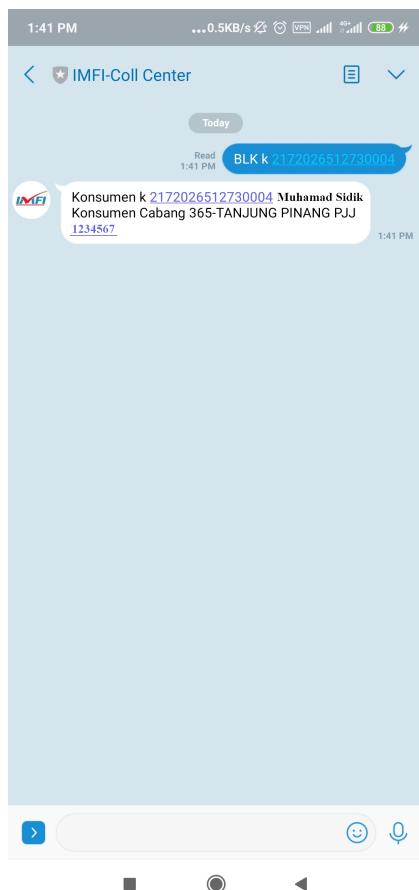
4.1.11 Tampilan respon modul BLK (*Blacklist*) bila status calon konsumen OK.



Gambar 4.11: Tampilan respon modul BLK konsumen OK

Gambar 4.11 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul BLK (*Blacklist*) dengan memasukkan keyword BLK + (kode konsumen) + Nomor NIK calon konsumen. Jika NIK calon konsumen tidak ada dalam *database* konsumen dan tidak ada dalam database konsumen *blacklist* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon “Status OK”, yang artinya calon konsumen tersebut dapat diproses menjadi konsumen di PT. Indomobil Finance Indonesia.

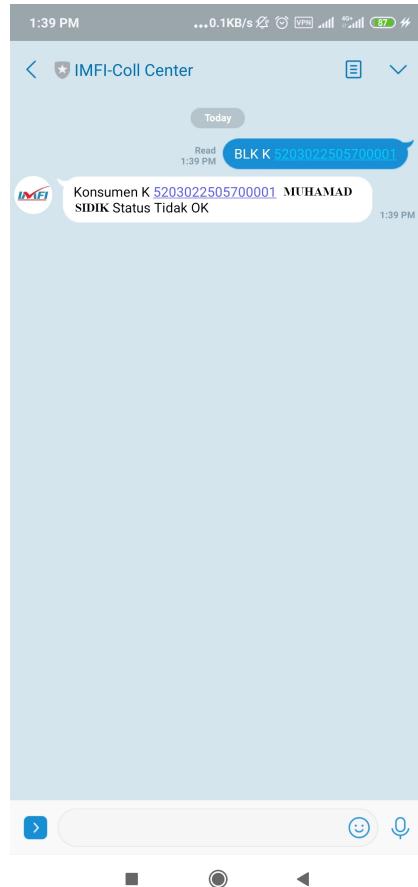
4.1.12 Tampilan respon modul BLK (*Blacklist*) bila status calon konsumen sudah pernah menjadi konsumen sebelumnya.



Gambar 4.12: Tampilan respon modul BLK calon konsumen pernah menjadi konsumen

Gambar 4.12 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul BLK (*Blacklist*) dengan memasukkan keyword BLK + (kode konsumen) + Nomor NIK calon konsumen. Jika NIK calon konsumen ada dalam *database* konsumen dan tidak ada dalam *database* konsumen *blacklist* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi mengenai konsumen tersebut, yang artinya calon konsumen tersebut dapat diproses menjadi konsumen di PT. Indomobil Finance Indonesia.

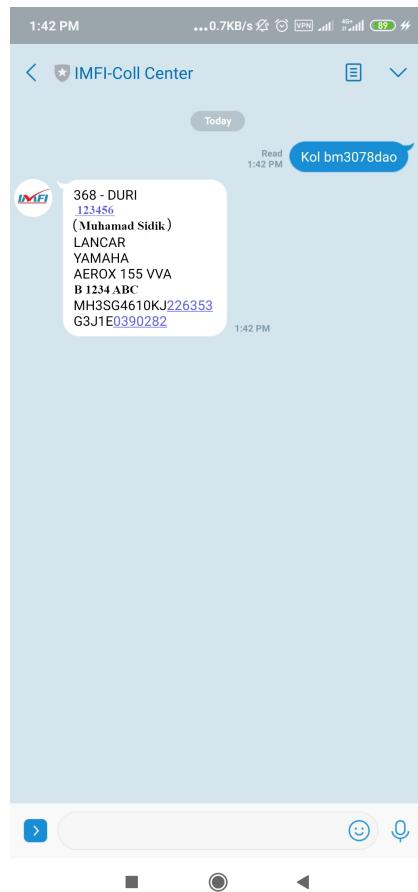
4.1.13 Tampilan respon modul BLK (*Blacklist*) bila status calon konsumen tidak OK.



Gambar 4.13: Tampilan respon modul BLK calon konsumen tidak OK

Gambar 4.13 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul BLK (*Blacklist*) dengan memasukkan keyword BLK + (kode konsumen) + Nomor NIK calon konsumen. Jika NIK calon konsumen ada dalam *database* konsumen dan ada dalam *database* konsumen *blacklist* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa calon konsumen tersebut “Tidak OK”, yang artinya calon konsumen tersebut tidak dapat diproses menjadi konsumen di PT. Indomobil Finance Indonesia.

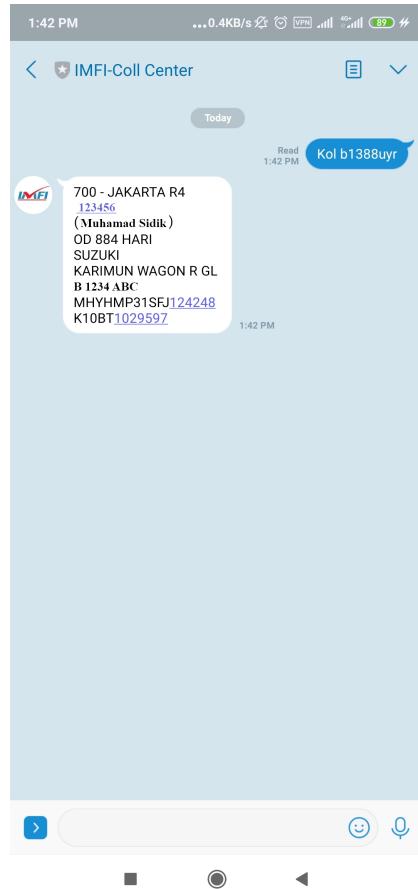
4.1.14 Tampilan respon modul KOL (Kolektor) bila status lancar.



Gambar 4.14: Tampilan respon modul KOL lancar

Gambar 4.14 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan tidak memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nomor PJJ (Perjanjian) konsumen, nama konsumen, serta detail unit yang diambil oleh konsumen.

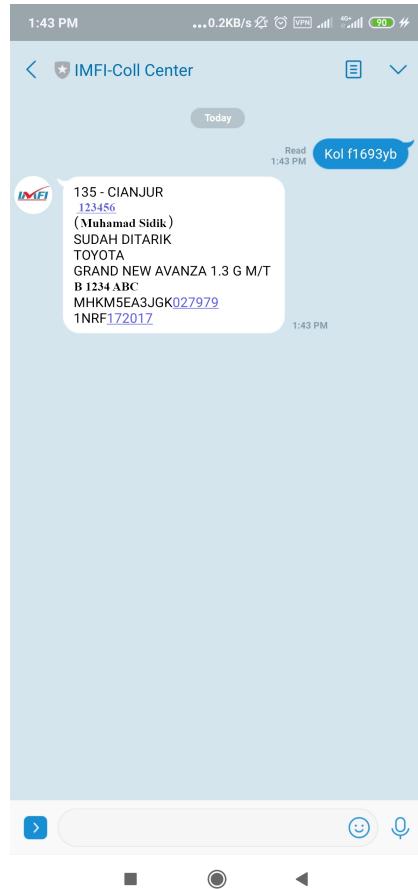
4.1.15 Tampilan respon modul KOL (Kolektor) bila status overdue.



Gambar 4.15: Tampilan respon modul KOL *overdue*

Gambar 4.15 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nomor PJJ (Perjanjian) konsumen, nama konsumen, lama tunggakan dalam satuan hari, serta detail unit yang diambil oleh konsumen.

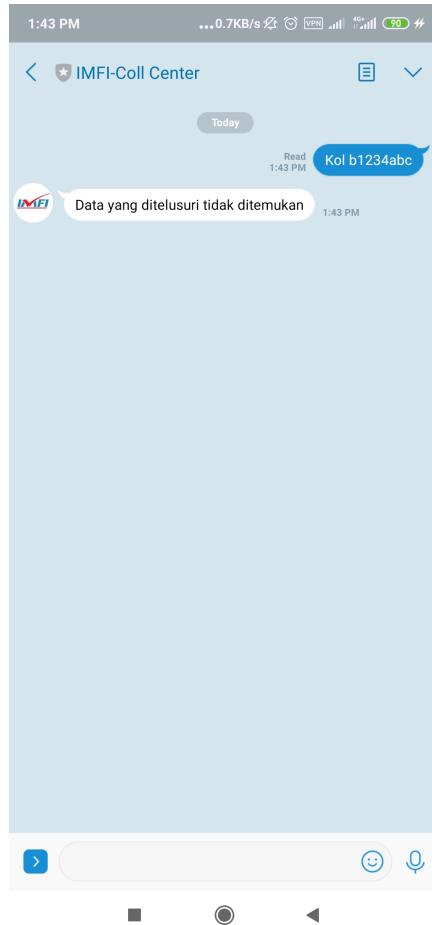
4.1.16 Tampilan respon modul KOL (Kolektor) bila status ditarik.



Gambar 4.16: Tampilan respon modul KOL ditarik

Gambar 4.16 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan unit sudah ditarik oleh kolektor. Maka *chatbot* akan memberikan respon informasi mengenai detail nomor PJJ (Perjanjian) konsumen, nama konsumen, status jika unit sudah ditarik, serta detail unit yang diambil oleh konsumen.

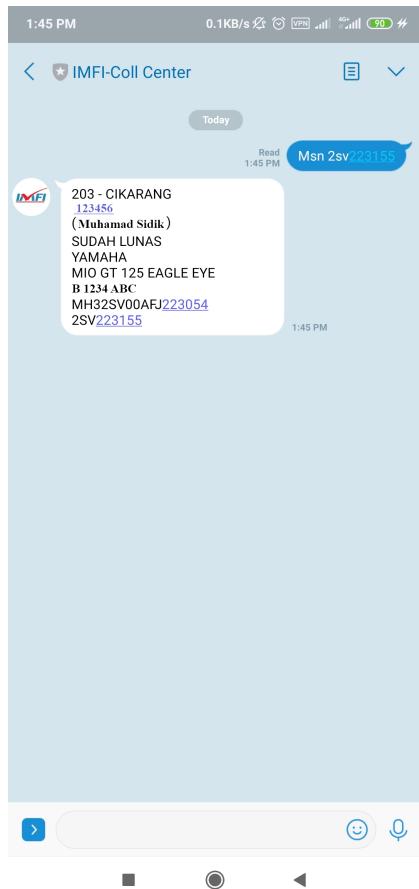
4.1.17 Tampilan respon modul KOL (Kolektor) bila data tidak ditemukan.



Gambar 4.17: Tampilan respon modul KOL *overdue*

Gambar 4.17 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul KOL (Kolektor) dengan memasukkan keyword KOL + Nomor Polisi kendaraan. Jika NoPol Kendaraan yang dimasukkan tidak ada dalam *database* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa Nomor Polisi yang dimasukkan tidak terdaftar dalam *database* PT. Indomobil Finance Indonesia (bukan unit yang diambil melalui PT. Indomobil Finance Indonesia).

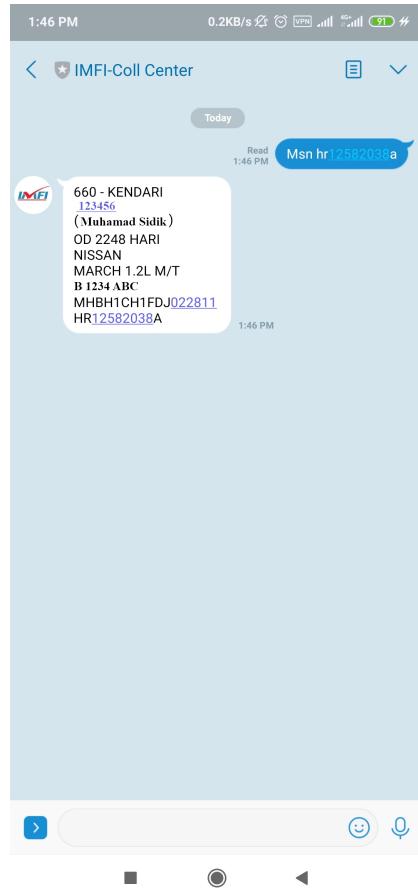
4.1.18 Tampilan respon modul MSN (Mesin) bila data OK.



Gambar 4.18: Tampilan respon modul MSN OK

Gambar 4.18 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul MSN (Mesin) dengan memasukkan keyword MSN + Nomor Mesin kendaraan. Jika No. Mesin Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan tidak memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut sudah lunas.

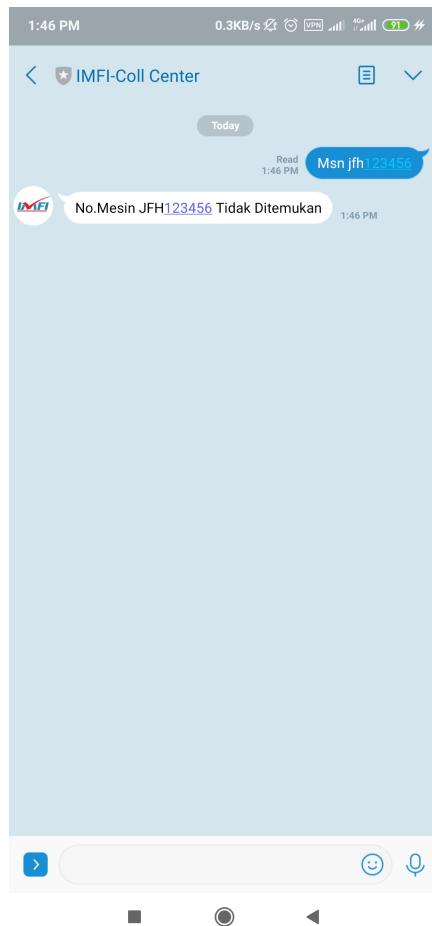
4.1.19 Tampilan respon modul MSN (Mesin) bila data overdue.



Gambar 4.19: Tampilan respon modul MSN *overdue*

Gambar 4.19 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul MSN (Mesin) dengan memasukkan keyword MSN + Nomor Mesin kendaraan. Jika No. Mesin Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut memiliki lama tunggakan dalam satuan hari.

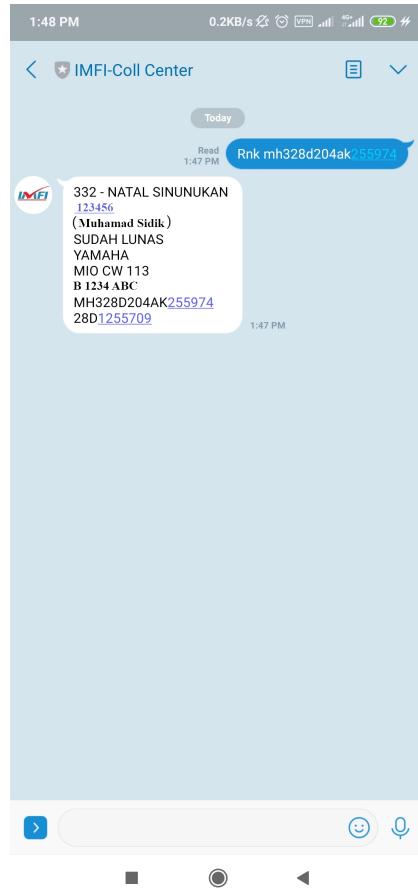
4.1.20 Tampilan respon modul MSN (Mesin) bila data tidak ditemukan.



Gambar 4.20: Tampilan respon modul MSN tidak ditemukan

Gambar 4.20 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul MSN (Mesin) dengan memasukkan keyword MSN + Nomor Mesin kendaraan. Jika No. Mesin Kendaraan yang dimasukkan tidak ada dalam *database* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa Nomor Mesin Kendaraan yang dimasukkan tidak terdaftar dalam *database* PT. Indomobil Finance Indonesia (bukan unit yang diambil melalui PT. Indomobil Finance Indonesia).

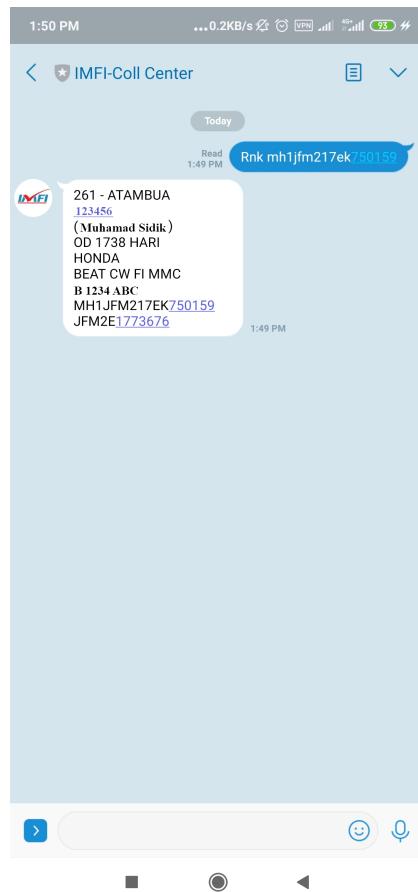
4.1.21 Tampilan respon modul RNK (Rangka) bila data OK.



Gambar 4.21: Tampilan respon modul RNK data OK

Gambar 4.21 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul RNK (Rangka) dengan memasukkan keyword RNK + Nomor Rangka kendaraan. Jika No. Rangka Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan tidak memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut sudah lunas.

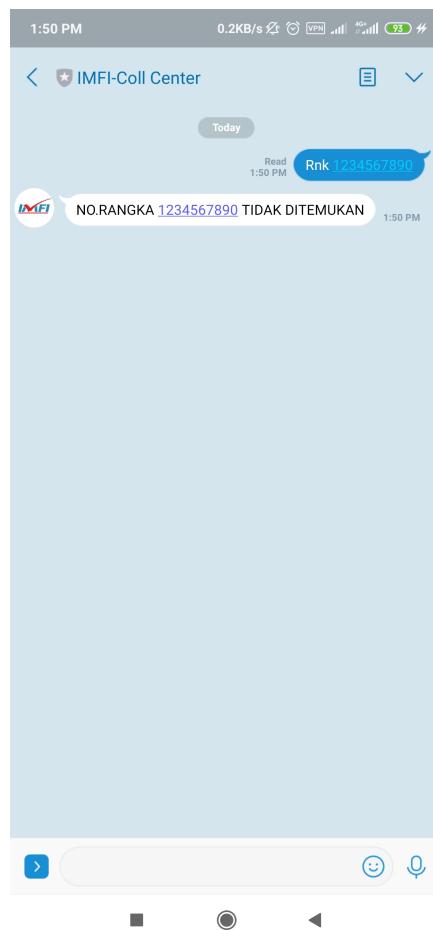
4.1.22 Tampilan respon modul RNK (Rangka) bila data *overdue*.



Gambar 4.22: Tampilan respon modul RNK data *overdue*

Gambar 4.22 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul RNK (Rangka) dengan memasukkan keyword RNK + Nomor Rangka kendaraan. Jika No. Rangka Kendaraan yang dimasukkan ada dalam *database* PT. Indomobil Finance Indonesia dan memiliki tunggakan. Maka *chatbot* akan memberikan respon informasi mengenai detail nama konsumen, cabang dimana unit tersebut diajukan, No. PJJ (Perjanjian) konsumen, serta status bahwa kendaraan tersebut memiliki lama tunggakan dalam satuan hari.

4.1.23 Tampilan respon modul RNK (Rangka) bila data tidak ditemukan.



Gambar 4.23: Tampilan respon modul RNK data *overdue*

Gambar 4.23 menunjukkan tampilan respon *chatbot* kolektor ketika *user* melakukan permintaan modul RNK (Rangka) dengan memasukkan keyword RNK + Nomor Rangka kendaraan. Jika No. Rangka Kendaraan yang dimasukkan tidak ada dalam *database* PT. Indomobil Finance Indonesia. Maka *chatbot* akan memberikan respon informasi bahwa Nomor Rangka Kendaraan yang dimasukkan tidak terdaftar dalam *database* PT. Indomobil Finance Indonesia (bukan unit yang diambil melalui PT. Indomobil Finance Indonesia).

4.2 Implementasi *Coding* pada Aplikasi Chatbot Kolektor

Setelah tahap design sudah menghasilkan tampilan aplikasi yang akan dibuat dan sudah disetujui oleh *user*, maka tahap berikutnya dari *extreme programming* ini adalah tahap *coding*. Tahap coding pada penelitian ini dibagi menjadi 4 tahapan yang secara rinci dijelaskan pada sub bab berikut.

4.2.1 Pembuatan *web service*

Web service adalah sebuah layanan yang berguna untuk mengatur interaksi antara sistem dengan platform tertentu. Pada penelitian ini dilakukan pembuatan sebuah web service yang berguna untuk mengatur interaksi antara aplikasi dan *database* yang berada pada sistem internal PT. Indomobil Finance Indonesia. Hal ini diperlukan karena PT. Indomobil Finance menggunakan jaringan tertutup sehingga hanya layanan tertentu saja yang diberikan akses yang dapat melakukan interaksi dengan aplikasi maupun database yang ada dimiliki oleh PT. Indomobil Finance Indonesia.

Web service ini yang kemudian akan menjadi jembatan antara *Messaging API* milik LINE untuk dapat berkomunikasi dengan aplikasi dan database yang ada di PT. Indomobil Finance Indonesia. Berikut adalah potongan *Pseudocode* dari *web service* yang dibuat pada penelitian ini :

```
Sub ProcessRequest(ByVal context As HttpContext) Implements IHttpHandler.
ProcessRequest
    Dim oJSONTextFormat As String = New StreamReader(context.Request.
        InputStream).ReadToEnd
    Dim oURL As String = ConfigurationManager.AppSettings
        ("URL_LINE_LOCAL")
    Dim oJSON As String = SendPOST(oURL, oJSONTextFormat)
    Dim sendLine As String = SendPOSTLine("https://api.line.me/
        v2/bot/message/reply", oJSON)

    context.Response.ContentType = "application/json"
    context.Response.Headers.Add("Authorization", "Bearer " +
        "rv0LmIA7+qAF3x3PaxtXgHm8KTruPFtVwz3p91UQpy5gEEZTC3Gh
        gp2w0bodKEgepe5k/tHAJT3XLCLKUDYFnBNkabojCruaYJfcUx/fA
        2Lym6EB9/5yZlkfH7FkXTBZeY15VON/j2ITcZKkvjBwwdB04t89/
        10/w1cDnyilFU=")
    context.Response.Write(oJSON)
End Sub
```

4.2.2 Pembuatan Knowledge Base

Tahap berikutnya adalah pembuatan *knowledge base*, *knowledge base* pada aplikasi *chatbot* berfungsi sebagai pusat informasi. *Knowledge base* adalah tempat dimana respon dari *request* yang diminta akan digenerate. Pada penelitian ini, knowledge base dibuat dengan memanfaatkan *stored procedure* yang terdapat pada SQLServer. *Stored procedure* ini bertugas untuk mencari fakta dari permintaan informasi yang masuk, *stored procedure* akan menjalankan *query* tertentu untuk dapat menghasilkan sebuah respon yang akan diteruskan ke *handler webhook* yang sudah dipersiapkan. Berikut adalah Pseudocode dari *stored procedure* yang dibuat pada penelitian ini :

```

IF dbo.isModul (@ApplicationFlag) = 0
    SET @sReplyMessage = 'Modul yang Anda minta tidak tersedia'
ELSE
    BEGIN
        IF dbo.isRegisteredModul (@ApplicationFlag, @NoHP) = 0
            SET @sReplyMessage = 'Anda tidak terdaftar untuk Modul ini'
        ELSE
            BEGIN
                DECLARE @sAfterModul           VARCHAR (480)
                SET @sAfterModul = LTRIM (SUBSTRING (@SMSMessage, CHARINDEX (' ', @SMSMessage), LEN (@SMSMessage)))
                DECLARE @sSPName               VARCHAR (50)
                DECLARE @sParameter             VARCHAR (100)

                SELECT @sSPName = SPLine, @sParameter = Parameter
                FROM SCT_Modul
                WHERE KodeModul = @ApplicationFlag

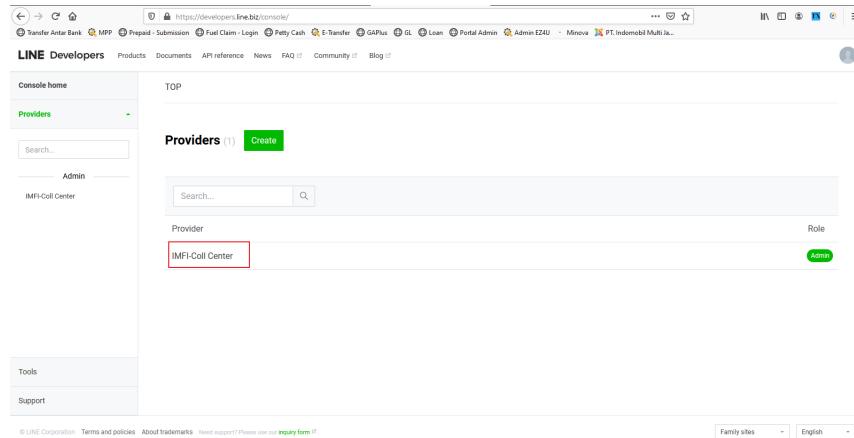
```

4.2.3 Konfigurasi Webhook

Konfigurasi *webhook* dilakukan agar LINE mampu mengenali aplikasi yang akan menjadi “*handler*” untuk menampung pesan serta menjadi pintu komunikasi antara *API* milik LINE dengan sistem internal yang dimiliki oleh PT. Indomobil Finance Indonesia. Konfigurasi *webhook* yang dilakukan pada penelitian ini yaitu :

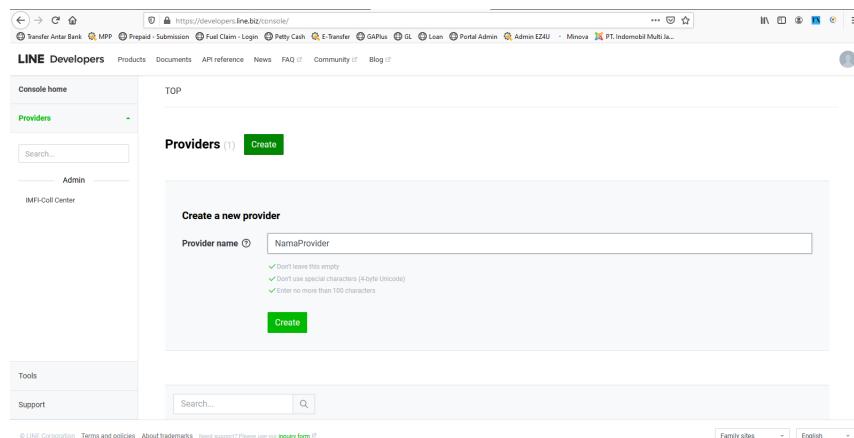
1. Tahapan pertama dari konfigurasi *webhook* adalah dengan mempersiapkan *providers* di halaman LINE Developer. Halaman awal LINE Developer ditunjukkan pada gambar 4.24. Untuk membuat *provider* baru, user perlu melakukan klik tombol “*Create*”. Jika user sudah pernah membuat provider sebelumnya, maka list provider akan ditampilkan pada halaman ini. User bisa merubah pengaturan pada provider yang

dibuat dengan melakukan klik pada list provider yang ingin diubah pengaturannya.



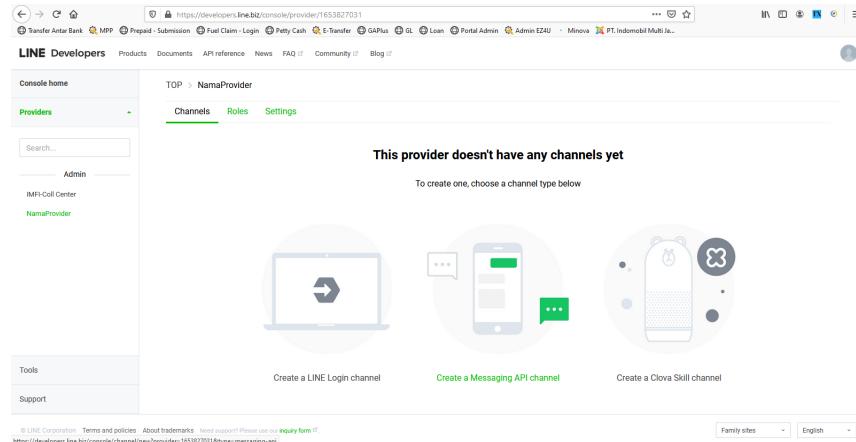
Gambar 4.24: Halaman awal LINE Developer

- Setelah melakukan klik pada tombol create, maka user akan diminta untuk mengisi nama provider seperti pada gambar 4.25.



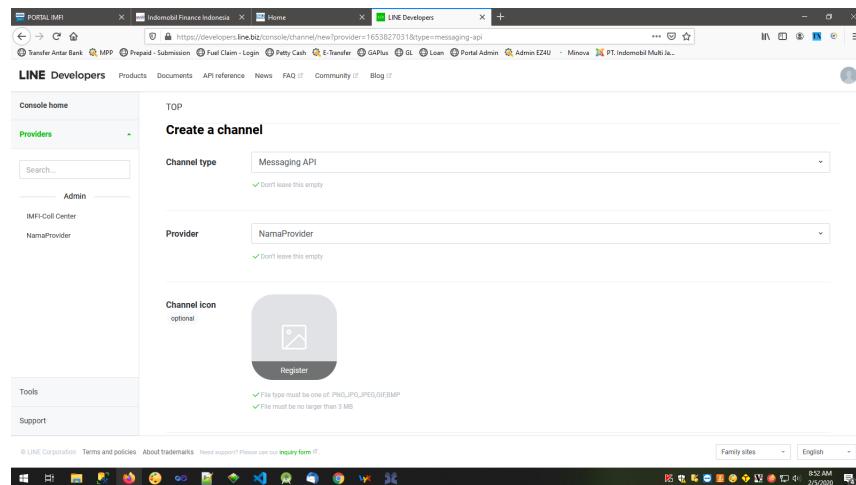
Gambar 4.25: Halaman pembuatan Line Provider

- Setelah memasukkan nama *provider*, user akan diminta untuk memilih jenis *Channel* yang akan dibuat pada *provider* ini, seperti ditunjukkan pada gambar 4.26. Pada penelitian ini *provider* yang dibuat menggunakan *Messaging API Channel* sesuai kebutuhan untuk membuat *chatbot*.

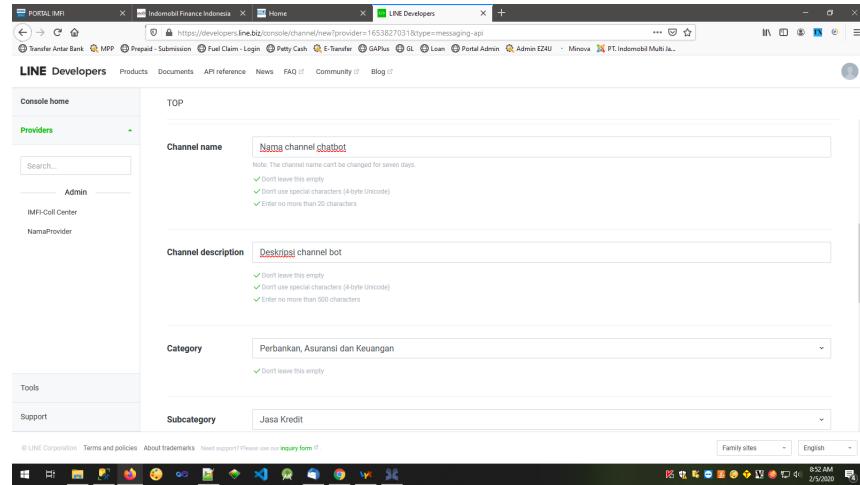


Gambar 4.26: Halaman pemilihan jenis channel

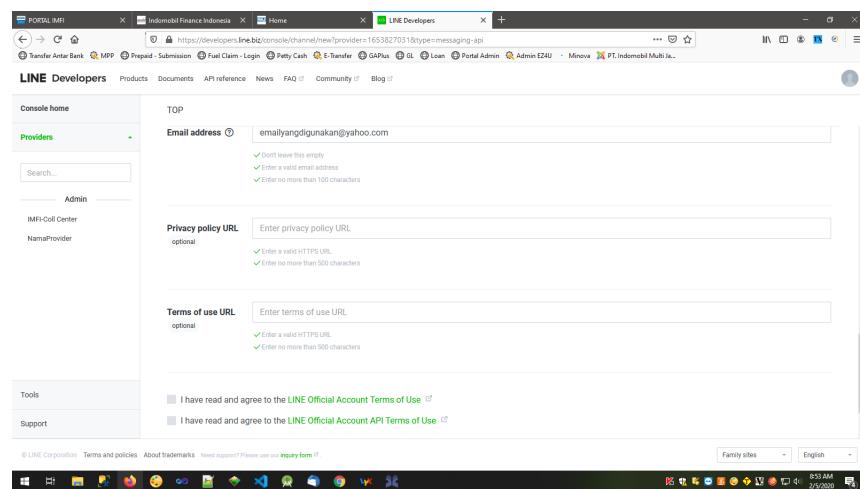
4. Jika *user* sudah memilih jenis *channel*, maka LINE Developer akan meminta *user* untuk membuat *channel* didalam *provider* yang sudah dibuat, seperti contoh pada gambar 4.27, 4.28, dan 4.29.



Gambar 4.27: Halaman pembuatan channel di provider

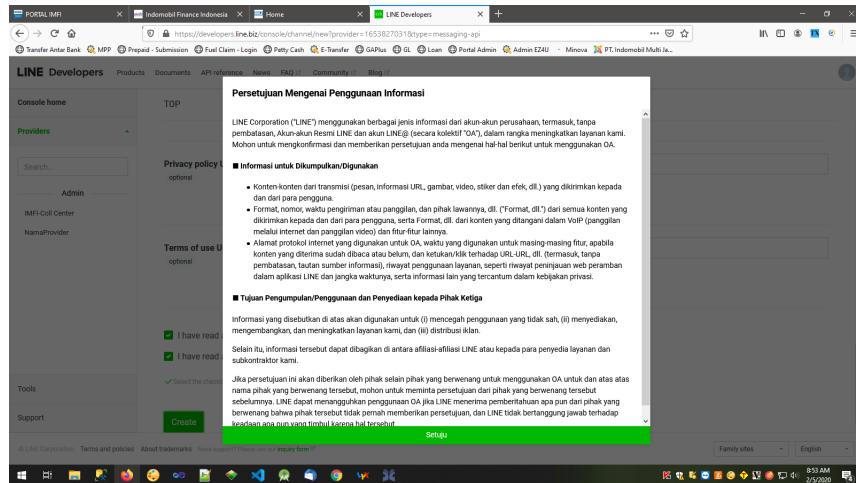


Gambar 4.28: Lanjutan halaman pembuatan channel di provider (1)



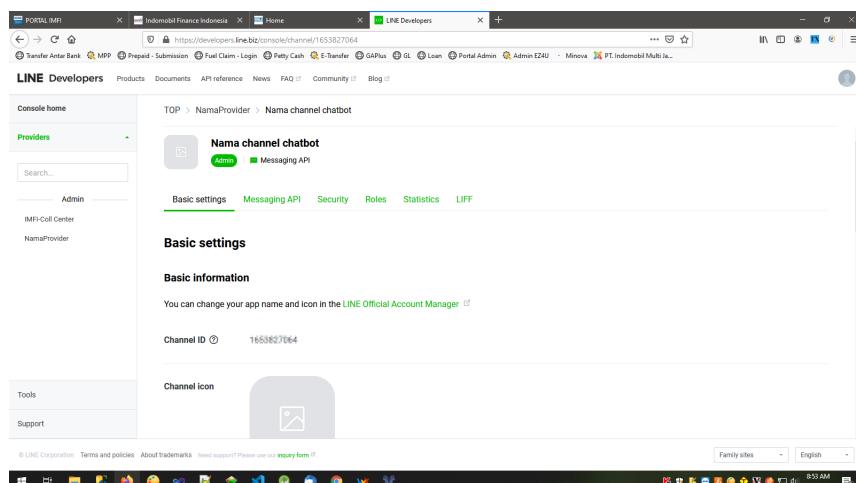
Gambar 4.29: Lanjutan halaman pembuatan channel di provider (2)

5. Setelah *user* mengisi seluruh data yang diperlukan, maka LINE akan menampilkan term and condition seperti gambar 4.30.



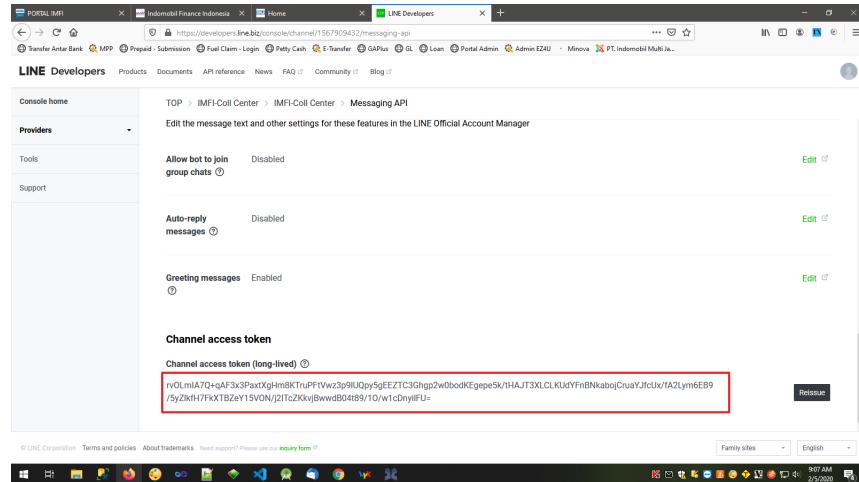
Gambar 4.30: Tampilan term and condition LINE

6. Jika sudah, maka halaman akan otomatis redirect ke halaman setting channel yang baru dibuat seperti pada gambar 4.31.



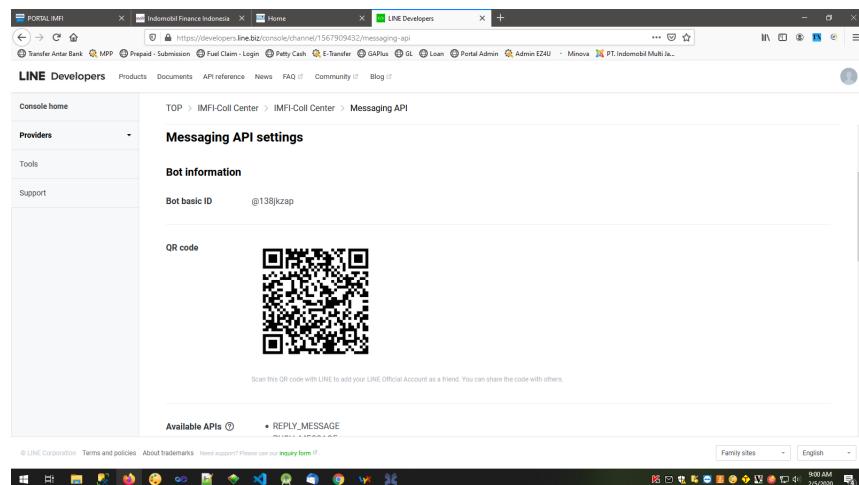
Gambar 4.31: Tampilan Setting Channel LINE

7. Setelah melakukan konfigurasi setelan pada akun *Official Account* LINE yang akan dijadikan *chatbot*, langkah selanjutnya adalah menyalin akses token *chatbot* untuk diletakkan pada web service yang sudah dibuat sebelumnya. Akses token dapat dilihat seperti pada gambar 4.32.

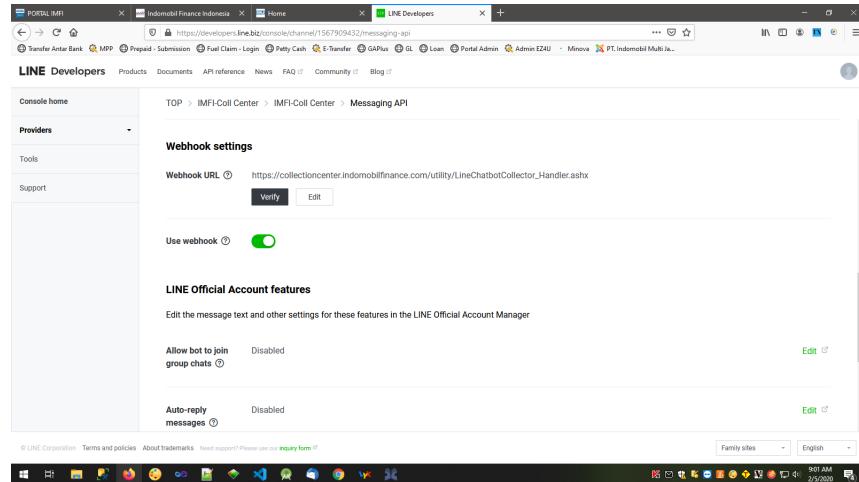


Gambar 4.32: Tampilan *Channel Access Token*

8. Untuk menghubungkan *official account* dengan *web service* yang sudah dipersiapkan, maka *user* perlu merubah setelan di bagian *Messaging API* seperti ditunjukkan pada gambar 4.33 dan 4.34. Gambar 4.33 menunjukkan *Bot ID* dan *QR Code* yang dapat digunakan untuk menambahkan *official account chatbot* sebagai teman. Sedangkan gambar 4.34 menunjukkan *URL webhook* yang ditujukan kepada alamat *url web service* yang sudah dipersiapkan.

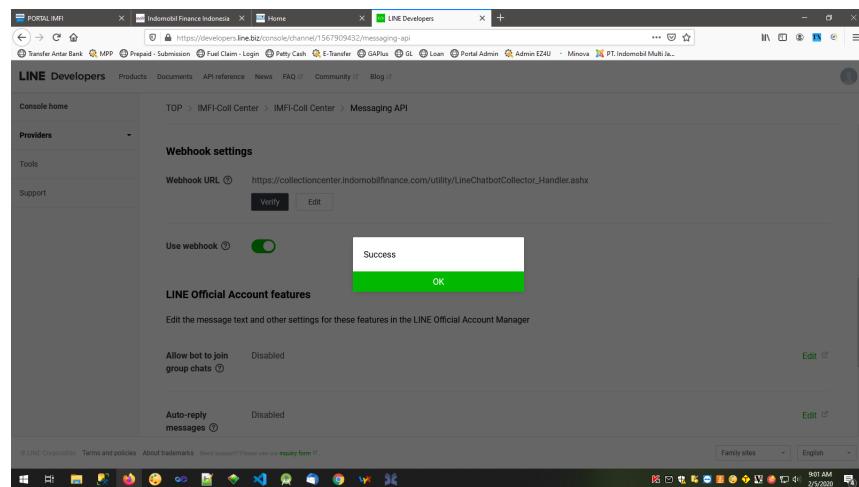


Gambar 4.33: Tampilan setelan *Messaging API*



Gambar 4.34: Lanjutan tampilan setelan *Messaging API*

- Untuk memeriksa apakah konfigurasi *webhook* dengan *web service* yang ada sudah berhasil, maka *user* perlu melakukan klik *verify*. Hasil jika konfigurasi sudah berhasil dilakukan ditunjukkan pada gambar 4.35.



Gambar 4.35: Tampilan Sukses *Verify*

Jika hasil dari aksi *verify* sudah menunjukkan hasil seperti pada gambar 4.35, maka konfigurasi *webhook* sudah berhasil dilakukan. Artinya, *Messaging API LINE* sudah terhubung dengan *web service* yang sudah dibuat sebelumnya, dan sudah dapat berkomunikasi dengan aplikasi maupun *database* yang sudah diatur dalam *web service*.

4.2.4 Pembuatan Logic Program

Pembuatan *logic program* digunakan untuk menentukan alur dari interaksi yang akan dilakukan oleh *chatbot* kolektor. Dalam proses pembuatan *logic program* ini juga strategi *forward chaining* diterapkan. Berikut adalah *pseudocode* untuk pembuatan logic program pada penelitian ini :

```

Dim oJSONTextFormat As String = New StreamReader(context.Request.InputStream)
    .ReadToEnd
Dim oJsonResult As SCT_LineMessage = ConvertToJsonFormat.ConvertToString
    (oJSONTextFormat)
Dim LineID = oJsonResult.LineID
Dim incomingmessage = oJsonResult.Message

Dim oArLineUser As ArrayList = New WSFunction().CekLineUserByID(LineID)
Dim message = ""
Dim oSCT_LineUser = New SCT_LineUser
Dim LineUserID As Int64

If (oArLineUser.Count = 0) Then
    oSCT_LineUser.LineID = LineID
    oSCT_LineUser.Status = 0
    oSCT_LineUser.DialogFlow = 1

    LineUserID = New WSFunction().CreateLineUser(oSCT_LineUser)
    oSCT_LineUser.LineUserID = LineUserID
Else
    oSCT_LineUser = CType(oArLineUser(0), SCT_LineUser)
End If

```

4.2.4.1 Penerapan Strategi Forward Chaining

Strategi forward chaining dengan menerapkan metode jika-maka yang dilakukan pada penelitian ini ditunjukkan pada *pseudocode* berikut.

```

If incomingmessage.Length > 3 Then
Dim pesanMasuk = incomingmessage.Substring(0, 4)
If pesanMasuk.ToUpper = "KOL " Or pesanMasuk.ToUpper = "BLK "
    Or pesanMasuk.ToUpper = "MSN "
    Or pesanMasuk.ToUpper = "RNK " Then
        Dim oLineNoHP As ArrayList = New LineUserMapper().RetrieveWithCondition
            ("LineID = '" + LineID + "'", "LineID desc")
        Dim dateToday As DateTime = DateTime.Today
        Dim oNewInboxLine = New SCT_InboxLine

        oNewInboxLine.SMSMessage = incomingmessage
        oNewInboxLine.NoHP = CType(oLineNoHP(0), SCT_LineUser).NoHP
        oNewInboxLine.WaktuSMS = dateToday

        If incomingmessage.Length > 3 Then
            oNewInboxLine.ApplicationFlag = incomingmessage.Substring(0, 3)
            .ToUpper
        Else
            oNewInboxLine.ApplicationFlag = "Invalid"

```

```

End If

oResult = New InboxLineMapper().Insert(oNewInboxLine)
Dim oReplyInboxLine As SCT_InboxLine = New InboxLineMapper()
    .Retrieve(oResult)
If Not IsNothing(oReplyInboxLine) Then
    message = oReplyInboxLine.ReplyMessage
Else
    message = "Reply tidak ditemukan"
End If

```

4.3 Hasil Perbandingan Sistem

Untuk mengetahui sejauh mana aplikasi yang dikembangkan mampu menyelesaikan permasalahan, maka dilakukan perbandingan antara sistem yang saat ini berjalan di PT. Indomobil Finance Indonesia, yaitu aplikasi SMSCenter dengan aplikasi Chatbot Kolektor. Berikut tabel 4.1 dan 4.2 menunjukkan hasil perbandingan aplikasi SMSCenter dengan Chatbot Kolektor.

Tabel 4.1: Hasil perbandingan aplikasi SMSCenter dengan Chatbot Kolektor

No	Permasalahan	SMSCenter	Chatbot Kolektor
1	Bergantung pada kartu SIM	Ya, secara menyeluruh	Ya, sebagian (hanya untuk mengirim SMS OTP diawal pendaftaran)
2	Bergantung pada internet	Tidak	Ya
3	Memerlukan biaya per pesan	Ya	Tidak
4	Gangguan pelayanan pada 3 bulan terakhir	Ya	Tidak
5	Maintenance Ketika gangguan	Perlu hard reset modem gsm	Hanya melakukan pengecekan apakah webhook dapat diverifikasi.

Berdasarkan pada Tabel 4.1, Aplikasi yang saat ini berjalan di PT. Indomobil Finance Indonesia yaitu aplikasi SMSCenter, mempunyai ketergantungan pada kartu sim, hal ini dikarenakan semua layanan yang berjalan pada aplikasi SMSCenter berupa pesan masuk dan pesan keluar diproses melalui pesan singkat atau *SMS* (*Short Message Service*). Hal ini membuat aplikasi SMSCenter bergantung secara menyeluruh pada kartu sim, sedangkan pada aplikasi yang dikembangkan dipenelitian ini yaitu Chatbot Kolektor,

penggunaan kartu sim hanya bergantung sebagian, yaitu ketika chatbot kolektor perlu mengirimkan pesan *OTP* untuk validasi nomor yang digunakan oleh kolektor ketika mendaftar di layanan Aplikasi Chatbot Kolektor.

Aplikasi SMSCenter tidak bergantung pada internet sama sekali, hal ini karena *service* yang berjalan sepenuhnya menggunakan komunikasi fisik dari modem gsm ke aplikasi SMSCenter. Sedangkan Chatbot Kolektor, bergantung sepenuhnya pada internet, hal ini dikarenakan pertukaran informasi yang dilakukan berjalan pada *platform messaging application* LINE yang membutuhkan internet untuk beroperasi. Selain itu, komunikasi antara Messaging API LINE dengan webhook handler di sisi internal juga membutuhkan akses internet, sehingga Chatbot Kolektor bergantung sepenuhnya pada akses internet.

Pada bagian pемbiayaan, aplikasi SMSCenter membutuhkan biaya untuk tiap pesan yang dikirimkan sebagai respon dari permintaan informasi yang dilakukan oleh kolektor, besaran biaya yang ditagihkan tergantung pada jenis kartu sim yang digunakan, pada aplikasi SMSCenter sendiri kartu sim yang digunakan adalah kartu halo dengan tarif per pesannya adalah Rp. 350,-. Sedangkan aplikasi Chatbot Kolektor tidak membutuhkan biaya untuk pertukaran pesan yang terjadi, hal ini dikarenakan aplikasi Chatbot Kolektor berjalan diatas *platform* LINE yang tidak membebankan biaya perpesanan yang dilakukan (*Free Messaging Application*) sehingga sebanyak apapun pertukaran pesan yang terjadi pada aplikasi Chatbot Kolektor tidak akan memengaruhi biaya operasional yang perlu dikeluarkan oleh PT. Indomobil Finance Indonesia.

Jika dilihat pada *track record* 3 bulan terakhir, aplikasi SMSCenter mengalami beberapa kali gangguan pelayanan, terhitung sejak Desember 2019 aplikasi SMSCenter mengalami 6 kali gangguan yang menyebabkan seluruh layanan tidak berfungsi, yaitu pada tanggal 2, 3, 4, dan 5 Januari 2020, serta tanggal 1 dan 2 Februari 2020. Gangguan ini berasal dari kondisi sinyal pada kartu sim yang tidak stabil sehingga menyebabkan modem GSM menjadi *hang* dan sistem pelayanan lumpuh hingga modem direstart secara manual. Sedangkan *track record* 3 bulan terakhir pada aplikasi Chatbot Kolektor, pelayanan berjalan penuh selama 1x24 jam perhari tanpa mengalami adanya gangguan pelayanan.

Dalam hal penanganan masalah jika terjadi gangguan, arsitektur aplikasi SMSCenter yang mengandalkan fisik menyebabkan jika sistem mengalami masalah, maka diperlukan pemeriksaan fisik terkait sistem aplikasi SMSCenter.

ter. Permasalahan yang terjadi selama ini disebabkan oleh kondisi simcard yang *hang*, ataupun modem gsm yang mengalami *hang*. Untuk bisa menyelesaikan permasalahan tersebut, maka tim *development* perlu datang ke ruang server dan mengatur ulang posisi *simcard* dan menonaktifkan modem gsm sementara untuk kemudian dinyalakan kembali. Sementara, pada aplikasi Chatbot Kolektor, jika terjadi permasalahan pada layanan, maka yang perlu dilakukan adalah melakukan pemeriksaan ketersediaan *webhook* yang *disetting* pada halaman LINE Developer untuk memastikan bahwa *Messaging API* LINE mampu berkomunikasi dengan sistem internal yang dimiliki. Jika *webhook* tidak dapat diverifikasi, maka hal yang perlu dilakukan adalah memeriksa pengaturan *API messaging* LINE pada *webhook* yang dimiliki.

Tabel 4.2: Hasil perbandingan biaya operasional SMSCenter dengan Chatbot Kolektor

No	Permasalahan	SMSCenter	Chatbot Kolektor
1	Banyaknya pesan Collector yang masuk (selama 1 tahun)	9.872 Pesan	9.872 Pesan
2	Banyaknya pesan OTP	0 Pesan	257 Pesan
3	Biaya Per Pesan Collector	Rp. 350,00	Rp. 0,00
4	Biaya Per Pesan OTP	Rp. 350,00	Rp. 350,00
5	SubTotal Biaya pesan kolektor	Rp. 3.455.200,00	Rp. 0,00
6	SubTotal Biaya pesan OTP	Rp. 0,00	Rp. 64.250,00
7	Total Biaya yang dikeluarkan	Rp. 3.455.200,00	Rp. 64.250,00

Berdasarkan data yang ditunjukkan pada tabel 4.2, terlihat hasil perbandingan pengeluaran biaya operasional antara sistem yang sedang berjalan yaitu SMSCenter, dengan sistem yang dibangun pada penelitian ini yaitu Chatbot Kolektor. Pada tabel 4.2 tersebut diambil permasalahan perhitungan dengan menggunakan data pesan yang terkirim sebagai respon permintaan layanan kolektor selama tahun 2019, yaitu 9.872 pesan seperti yang ditunjukkan pada baris 1.

Pada baris 2 tabel 4.2 ditunjukkan kebutuhan pesan yang diperlukan untuk mengirimkan pesan *OTP* (*One Time Password*). Aplikasi SMSCenter memerlukan 0 pesan karena tidak adanya pesan OTP melalui aplikasi SM-

SCenter ini, sedangkan aplikasi Chatbot Kolektor membutuhkan pesan *OTP* sebanyak 257 pesan, sesuai dengan jumlah kolektor diseluruh cabang yang terdaftar dalam *database* PT. Indomobil Finance Indonesia. Pesan *OTP* ini berlaku untuk tiap nomor baru yang didaftarkan di *database* dan mendaftar di aplikasi Chatbot Kolektor.

Baris ke 3 menunjukkan besarnya biaya per pesan yang dikeluarkan untuk membalas respon dari permintaan layanan tiap kolektor, sesuai dengan kartu sim yang digunakan oleh PT. Indomobil Finance Indonesia, beban biaya per SMS dari kartu Halo akan dikenakan RP. 350,-. Sedangkan untuk biaya per pesan aplikasi Chatbot Kolektor adalah Rp. 0,- karena pertukaran pesan yang dilakukan pada platform LINE adalah gratis. Baris ke 4 menunjukkan biaya pesan yang digunakan untuk mengirimkan pesan *OTP*, biaya yang dibebankan pada kedua aplikasi adalah sama, yaitu Rp. 350,- per pesan sesuai dengan biaya dari *simcard* Kartu Halo.

Baris ke 5 menunjukkan subtotal biaya yang dikeluarkan oleh kedua aplikasi jika terdapat 9.872 pesan sesuai dengan data pesan yang masuk selama tahun 2019, dengan biaya Rp. 350,- per pesan maka aplikasi SMSCenter mengeluarkan biaya operasional sebesar Rp. 3.455.200,00. Sedangkan dengan jumlah pesan yang sama, pada aplikasi Chatbot Kolektor biaya yang perlu dikeluarkan tetap Rp. 0,- karena tidak ada pembebanan biaya bertukar pesan dari *platform* LINE.

Baris ke 6 menunjukkan subtotal biaya yang dikeluarkan kedua aplikasi untuk pesan *OTP* yang dikirimkan, SMSCenter tidak mengirimkan pesan *OTP* sehingga beban biaya yang dikeluarkan adalah Rp. 0,-. Aplikasi Chatbot Kolektor mengirimkan 257 Pesan *OTP* ke 257 pengguna yang terdaftar pada *database*, maka beban biaya yang perlu dikeluarkan oleh aplikasi Chatbot Kolektor adalah Rp. 64.250,00. Biaya ini dikeluarkan hanya 1 kali saat pendaftaran aplikasi Chatbot Kolektor. Kedepannya, jika pengguna sudah terdaftar pada aplikasi Chatbot Kolektor, maka tidak akan dikenakan biaya apapun selamanya. Jika ada user baru yang mendaftar pada aplikasi Chatbot Kolektor, maka beban biaya yang dikeluarkan hanya untuk pesan *OTP* saja, yaitu Rp. 350,-.

Baris ke 7 menunjukkan hasil total biaya yang dikeluarkan oleh kedua aplikasi jika terdapat 9.872 pesan yang perlu dibalas dan 257 pesan *OTP* yang perlu dikirimkan, data total ini didapat dengan menjumlahkan subtotal biaya per pesan dengan subtotal biaya untuk pesan *OTP*. Dengan perhitungan tersebut, maka beban biaya yang harus dikeluarkan oleh PT. Indomobil

Finance Indonesia jika menggunakan SMSCenter adalah Rp. 3.455.200,00. Sedangkan jika menggunakan aplikasi Chatbot Kolektor biaya yang dikeluarkan adalah Rp. 64.250,00. Artinya dengan jumlah data yang sama, aplikasi Chatbot Kolektor sangat mampu untuk menghemat biaya operasional yang dikeluarkan oleh PT. Indomobil Finance Indonesia, beban biaya aplikasi chatbot kolektor hanya berjumlah 1,86% dari total biaya yang dikeluarkan oleh aplikasi SMSCenter. Biaya yang dikeluarkan oleh SMSCenter adalah biaya tentative yang nilainya akan sesuai dengan jumlah pesan yang perlu dibalas, semakin banyak pesan masuk yang perlu dibalas, maka beban biaya yang dikeluarkan juga akan semakin besar. Sedangkan biaya yang dikeluarkan oleh Chatbot Kolektor adalah biaya 1 kali pengeluaran yang hanya dikeluarkan pada saat pertama kali pengguna mendaftarkan diri pada aplikasi Chatbot Kolektor.

Dari data yang ditunjukkan pada tabel 4.1 dan tabel 4.2, maka dapat disimpulkan bahwa aplikasi Chatbot Kolektor mampu menjadi solusi untuk menghemat biaya operasional yang dikeluarkan oleh PT. Indomobil Finance Indonesia, serta menjadi solusi bagi permasalahan ketersediaan layanan informasi bagi kolektor PT. Indomobil Finance Indonesia.

4.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak pada penelitian ini menggunakan *blackbox testing*, pengujian blackbox testing dilakukan dari sisi peneliti untuk memastikan bahwa aplikasi chatbot kolektor sudah memberikan respon yang sesuai. Selain pengujian perangkat lunak, pada penelitian ini juga dilakukan dengan melaksanakan *UAT* atau *User Acceptance Test* untuk memastikan bahwa aplikasi chatbot yang dibangun sudah sesuai dengan permintaan dan kebutuhan *user*.

4.4.1 Blackbox Testing

Pengujian *blackbox testing* dilakukan pada fungsi yang terdapat di sebuah perangkat lunak. Adapun tahapan yang dilakukan pada penelitian ini untuk melakukan pengujian perangkat lunak dengan *blackbox testing* adalah

1. Analisa kebutuhan dan spesifikasi.

Perencanaan pengujian perangkat lunak dengan *blackbox testing* dilakukan dengan malkukan analisa kebutuhan dan spesifikasi pengujian yang diperlukan. Terdapat beberapa kebutuhan yang diperlukan pada penelitian ini untuk membuat *blackbox testing* yaitu aplikasi yang akan dilakukan pengujian, sistem operasi yang digunakan untuk menjalankan pengujian, dan basis data yang digunakan dalam pengujian. Tabel 4.3 menunjukkan hasil dari analisa kebutuhan dan spesifikasi yang digunakan pada pengujian perangkat lunak dengan menggunakan *blackbox testing*.

Tabel 4.3: Hasil analisa kebutuhan dan spesifikasi pengujian perangkat lunak dengan *blackbox testing*

No	Kebutuhan	Keterangan
1	Aplikasi yang dilakukan pengujian	Aplikasi yang diuji adalah aplikasi chatbot kolektor yang dibangun pada penelitian ini.
2	Sistem operasi	Windows 10 64 bit
3	Basis data	SQLServer 2012

2. Pemilihan Input.

Tahapan selanjutnya dilakukan pemilihan input untuk dilakukan pengujian. Pemilihan input ini bertujuan untuk menentukan *input* yang akan dimasukkan dalam pengujian perangkat lunak secara *blackbox testing*. Adapun pemilihan input yang dilakukan pada penelitian ini adalah input sesuai dengan keyword yang sudah ditentukan, dan input yang tidak sesuai dengan keyword yang sudah ditentukan.

Tabel 4.4: Skenario Pemilihan Input Pengujian Perangkat Lunak Dengan *Blackbox Testing*

No	Skenario Input Sesuai Keyword	Skenario Input Tidak Sesuai Keyword
1	BLK K 3216061911940019	Test
2	KOL B 4254 FUO	Hai
3	MSN 2SV223155	asdadasd
4	RNK MH328D204AK255974	KOL 123456

3. Pemilihan Output.

Proses pemilihan output digunakan untuk menentukan respon apa yang akan diberikan kepada pesan yang masuk. Pada tes yang dilakukan dipenelitian ini, output yang dipersiapkan dapat dilihat pada tabel 4.5.

Tabel 4.5: Skenario Pemilihan Output Pengujian Perangkat Lunak Dengan *Blackbox Testing*

No	Input yang diberikan	Skenario Output yang Diharapkan
1	Input sesuai dengan keyword dan data tersedia	Respon sesuai hasil yang diberikan <i>knowledge base</i>
2	Input sesuai keyword dan data tidak tersedia	Respon bahwa data tidak ditemukan
3	Input dengan keyword yang salah	Respon bahwa keyword yang diberikan salah
4	Input dengan kata random	Respon <i>default message</i> modul yang bisa diakses

4. Pengujian

Setelah dilakukan pemilihan input dan pemilihan output, maka langkah berikutnya adalah melakukan pengujian terhadap aplikasi chatbot kolektor. Hasil dari pengujian blackbox yang dilakukan dapat dilihat pada Tabel 4.6.

Tabel 4.6: Hasil pengujian perangkat lunak dengan menggunakan *Blackbox Testing*

No	Langkah Skenario	Hasil Yang Diharapkan	Hasil Aktual	Hasil Akhir
1	User input BLK K 1403025006980001 (Data Tersedia)	Chatbot merespon data blacklist	Chatbot merespon data blacklist	Pengujian Sukses
2	User input BLK K 5203022505700001 (Data Tidak Tersedia)	Chatbot merespon data tidak tersedia	Chatbot merespon data tidak tersedia	Pengujian Sukses
3	User input KOL bm3078dao (Data Tersedia)	Chatbot merespon data detail kolektor	Chatbot merespon data detail kolektor	Pengujian Sukses
4	User input KOL b1234abc (Data Tidak Tersedia)	Chatbot merespon data tidak tersedia	Chatbot merespon data tidak tersedia	Pengujian Sukses
5	User input MSN 2SV223155 (Data Tersedia)	Chatbot merespon data detail mesin	Chatbot merespon data detail mesin	Pengujian Sukses
6	User input MSN jfh123456 (Data Tidak Tersedia)	Chatbot merespon data tidak tersedia	Chatbot merespon data tidak tersedia	Pengujian Sukses
7	User input RNK mh328d204ak255974 (Data Tersedia)	Chatbot merespon data detail rangka	Chatbot merespon data detail rangka	Pengujian Sukses
8	User input RNK 123456789 (Data Tidak Tersedia)	Chatbot merespon data tidak tersedia	Chatbot merespon data tidak tersedia	Pengujian Sukses

Lanjutan tabel 4.4 : Hasil pengujian perangkat lunak dengan menggunakan *Blackbox Testing*

No	Langkah Skenario	Hasil Yang Diharapkan	Hasil Aktual	Hasil Akhir
9	User input MENU (Keyword sesuai)	Chatbot menampilkan pesan MENU	Chatbot menampilkan pesan MENU	Pengujian Sukses
10	User input HELP (Keyword sesuai)	Chatbot menampilkan pesan HELP	Chatbot menampilkan pesan HELP	Pengujian Sukses
11	User input HAI (Keyword tidak sesuai)	Chatbot menampilkan pesan MENU	Chatbot menampilkan pesan MENU	Pengujian Sukses
12	User input ASDASD (Keyword tidak sesuai)	Chatbot menampilkan pesan MENU	Chatbot menampilkan pesan MENU	Pengujian Sukses

5. Review Hasil

Berdasarkan tabel 4.6 yang berisi hasil pengujian aplikasi dengan menggunakan *blackbox testing*, dapat disimpulkan bahwa seluruh fungsi yang terdapat pada aplikasi *chatbot* kolektor telah berjalan sesuai dengan fungsinya dan respon yang diharapkan telah ditampilkan dengan sesuai.

4.4.2 *User Acceptance Test*

User Acceptance Test dilakukan untuk mendapatkan konfirmasi user bahwa aplikasi yang dibangun sudah sesuai dengan apa yang user inginkan. Pada penelitian ini, dibuat sebuah *Form UAT* untuk diberikan kepada Ibu Nasya dan Bapak Hengky dari divisi *Remedial and Collection*, untuk melakukan pemeriksaan tiap fungsi yang ada pada *chatbot* kolektor. *Form UAT* yang dibuat pada penelitian ini dapat dilihat pada tabel 4.7.

Tabel 4.7: *Form UAT* yang diberikan ke *user* untuk validasi bahwa program sudah sesuai.

USER ACCEPTANCE TEST			
Application : Line Chatbot - IMFI Collection Center			
Address : Line Account (xxxxxxxx), http://serweb03.imfi.com/smscenter			
Departemen :			
UAT Date :			
No	Test Case	Checkmark	Remark
1	Add User	-	-
	Menambahkan Friend Chatbot Line di Line user	V	V
2	Pendaftaran User	-	-
	A. User terdaftar di SMS Center	-	-
	• Melakukan pendaftaran di Chatbot Line	V	V
	B. User tidak terdaftar di SMS Center	-	-
	• Melakukan pendaftaran di Chatbot Line	V	V
3	Memasukkan Keyword module terdaftar	-	-
	A. Modul KOL	-	-
	• Melakukan pengecekan modul KOL	V	V
	B. Modul BLK	-	-
	• Melakukan pengecekan modul BLK	V	V
	C. Modul RNK	-	-
	• Melakukan pengecekan modul RNK	V	V
	D. Modul MSN	-	-
	• Melakukan pengecekan modul MSN	V	V

Form UAT pada tabel 4.7 adalah form UAT yang diisi oleh divisi *Remedial and Collection* selaku user yang mengajukan pembuatan program. Bagian *Checkmark* diisi oleh Ibu Nasya selaku Staf Divisi *Remedial and Collection* sekaligus PIC pembuatan Aplikasi Chatbot Kolektor, dan bagian *Remark* diisi oleh Bapak Hengky dari Divisi *Remedial and Collection* selaku Kepala Departemen sekaligus penanggung jawab permintaan Aplikasi Chatbot Kolektor.

Untuk pengujian terhadap kolektor, dilakukan pengujian terhadap 100 kolektor dengan mengambil pesan masuk dan respon yang diberikan oleh Chatbot Kolektor, Beberapa hasil pengujian dari 100 Kolektor ditunjukkan pada tabel 4.8 berikut.

Tabel 4.8: Hasil Pengujian Kolektor pada Aplikasi Chatbot Kolektor

No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
1	KOL B2992BKA	700 - JAKARTA R4 \n1600734 \n(RANTONI SIBARANI) \nOD 966 HARI \nSUZUKI \nNEW ERTIGA GL A/T AIRBAG \nB2992BKA \nMHYKZE81SGJ315965 \nK14BT1189803	(sudah sesuai)	Passed
2	kol b1271tyz	PJJ 702.1400694 (WIRDAWATI YANTU) OD 755 HARI, NISSAN, GRAND LIVINA 1.5 HWS AUTECH A/, J030448, 719258T	Format balasan tidak sesuai	Failed
3	MSN 515833	130 - BOGOR \n501316 \n(ASGARI) \nSUDAH LUNAS \nSUZUKI \nSHOGUN FD 125 XSD \nF 5323 GW \n515595 \n515833	(sudah sesuai)	Passed
4	BLK K64080845046 70002	Format SMS Anda Salah	Harusnya Format Pesan, bukan Format SMS	Failed

Lanjutan Tabel 4.8 : Hasil Pengujian Kolektor pada Aplikasi Chatbot
Kolektor

No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
5	BLK K 140201680480 0001	Konsumen K 1402016804800001 EKA APRELIYANTI Status Tidak OK	(sudah sesuai)	Passed
6	BLK K 327506080567 0020	Konsumen K 3275060805670020 MULYO ADI Status Tidak OK	(sudah sesuai)	Passed
7	BLK K 140201680480 0001	Konsumen K 1402016804800001 EKA APRELIYANTI Status Tidak OK	(sudah sesuai)	Passed
8	Kol h8533yi	222 - KUDUS \n1700139 \n(TEGUH SETYO BUDI CAHYONO) \nSUDAH LUNAS \nDAIHATSU \nTERIOS F700RGTX M/T \nH8533YI \nMH- KG2CJ2J8K010122 \nDAG6712	(sudah sesuai)	Passed
9	KOL AA1860ZC	224 - MAGELANG \n1400173 \n(MUHTAROM) \nSUDAH DITARIK \nSUZUKI \nST 1500 PICK UP FD \nAA1860ZC \nMHYESL415EJ321727 \nG15AID-959240	(sudah sesuai)	Passed

Lanjutan Tabel 4.8 : Hasil Pengujian Kolektor pada Aplikasi Chatbot
Kolektor

No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
10	KOL B3596BEF	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
11	MSN HR15002237D	221 - SOLO \n1300141 \n(TUGIMAN) \nOD 870 HARI \nNISSAN \nJUKE 1.5 RX RED EDITION \nAD 8475 BK \nMHB1CG1ADJ017367 \nHR15002237D	(sudah sesuai)	Passed
12	BLK K 64080845046 70002	Konsumen K 6408084504670002 Status OK	(sudah sesuai)	Passed
13	KOL B2237SFT	700 - JAKARTA R4 \n1501357 \n(BEN YOUSUF AMIR SUJUDI) \nOD 166 HARI \nDATSUN \nGO TYPE T OPTION M/T \nB2237SFT \nMHB1CG1ADJ017367 \nHR12729057T	(sudah sesuai)	Passed
14	RNK MHBG1CG 1ABJ057042	No.Rangka MHBG1CG1ABJ057042, FITRIA CHAERANI, Cbg 700-RYONO R4, No.PJJ 1100492 Status : OVD(1500)	(sudah sesuai)	Passed

Lanjutan Tabel 4.8 : Hasil Pengujian Kolektor pada Aplikasi Chatbot
Kolektor

No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
15	KOL B9712TIA	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
16	KOL H8770BW	756 - SEMARANG USED CAR \n1702780 \n(CLARA LILIS TRINGGA NINGSIH) \nOD 278 HARI \nTOYOTA \nAGYA G M/T \nH8770BW \nMH- KA4DA3JFJ061198 \n1KRA188320	(sudah sesuai)	Passed
17	MSN MHBG1CG1 ABJ057042	No.Mesin MHBG1CG1ABJ057042 Tidak Ditemukan	(sudah sesuai)	Passed
18	KOL B1634TZR	231 - YOGYAKARTA USED CAR \n1700128 \n(JAJANG SUPRIATNA) \nOD 466 HARI \nDAIHATSU \nXENIA 1.3 X M/T STD \nB1634TZR \nMH- KV1BA1JCK013350 \nMA21977	(sudah sesuai)	Passed
19	Kol p600pal	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

Lanjutan Tabel 4.8 : Hasil Pengujian Kolektor pada Aplikasi Chatbot
Kolektor

No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
20	KOL BA 4017 OS	Format SMS Anda Salah	Harusnya Format Pesan, bukan Format SMS	Failed
21	KOL B1680UH	130 - BOGOR \n1700016 \n(YERI WILSON) \nSUDAH DITARIK \nHONDA \nCR-V 2.0 A/T \nB1680UH \nMHR-RE18408J804155 \nR20A1-4907027	(sudah sesuai)	Passed
22	Kol be2105nf	Anda tidak terdaftar untuk Modul ini	(sudah sesuai)	Passed
23	KOL BG1471FD	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
24	Blk k 123456789012 3456	Konsumen k 1234567890123456 Status OK	NIK tidak ada seharusnya tidak muncul Status OK	Failed
25	Kol ag3213er	284 - PARE\n1600779\n(EDI SUHARYOKO)\nSUDAH LU- NAS\nYAMAHA\nNEW MIO M3 BLUE CORE\nAG3213ER\nMH3SE8820FJ006241\nE3R2E0201932	(sudah sesuai)	Passed

Lanjutan Tabel 4.8 : Hasil Pengujian Kolektor pada Aplikasi Chatbot
Kolektor

No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
26	KOL AA1860ZC	224 - MAGELANG \n1400173 \n(MUHTAROM) \nSUDAH DITARIK \nSUZUKI \nST 1500 PICK UP FD \nAA1860ZC \nMHYESL415EJ321727 \nG15AID-959240	(sudah sesuai)	Passed
27	MSN HR15002337D	No.Mesin HR15002337D Tidak Ditemukan	(sudah sesuai)	Passed
28	Kol h1707qv	226 - SALATIGA \n1702473 \n(TUMINEM) \nOD 217 HARI \nSUZUKI \nFUTURA PICK UP FD \nH1707QV \nMHYESL415HJ796957 \nG15AID1087501	(sudah sesuai)	Passed
29	BLK K3212151311 740003	Format SMS Anda Salah	Harusnya Format Pesan, bukan Format SMS	Failed
30	Kol b2748sfq	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

Pada tabel 4.8, hasil pengujian yang ditampilkan adalah 30 data dari 100 data yang diuji pada pengujian UAT terhadap kolektor, untuk data ke 31-100 terdapat pada lampiran. 30 data pada tabel 4.8 mewakili keberagaman respon Chatbot Kolektor yang sesuai maupun yang tidak sesuai dalam memberikan respon terhadap permintaan dari kolektor. Dari 100 data yang diuji, terdapat 5 data yang mengalami gagal uji karena respon yang diberikan tidak sesuai dengan yang diharapkan, pada data pengujian nomor 2 terjadi gagal uji karena format balasan tidak sesuai dengan yang seharusnya, sehingga ketika balasan sampai pada kolektor, pesan akan sulit dibaca. Pada data pengujian nomor 4, 20, dan 29 gagal uji terjadi karena respon balasan yang diberikan ketika user salah memasukkan format dibalas dengan kata “Format SMS salah”, seharusnya respon yang ditampilkan adalah “Format pesan salah”. Dan pada data pengujian nomor 24 terjadi gagal uji karena ketika NIK (Nomor Induk Kependudukan) yang dimasukkan adalah nomor secara acak, respon yang diberikan tetap mengatakan NIK tersebut OK, seharusnya divalidasi terlebih dahulu apakah nik yang dimasukkan sesuai dengan format nik yang sesuai. Berikut perhitungan persentase keberhasilan Aplikasi Chatbot Kolektor berdasarkan pengujian pada 100 data.

$$\text{Persentase Keberhasilan} = \frac{\text{Data Berhasil}}{\text{Jumlah Data}} \times 100\%, \text{ maka}$$

$$\text{Persentase} = \frac{(100-5)}{100} \times 100\% = \frac{95}{100} \times 100\% = 95\%$$

$$\text{Persentase Kegagalan} = \frac{\text{Data Gagal}}{\text{Jumlah Data}} \times 100\%, \text{ maka}$$

$$\text{Persentase} = \frac{5}{100} \times 100\% = 5\%$$

Dari perhitungan berdasarkan pengujian, keberhasilan Aplikasi Chatbot Kolektor dalam memberikan respon yang sesuai mencapai 95% dan persentase kesalahan yang terjadi adalah 5%. Dengan angka keberhasilan dan kegagalan tersebut, maka implementasi Aplikasi Chatbot Kolektor di PT. Indo-mobil Finance Indonesia dapat diteruskan untuk kemudian perlahan menggantikan sistem yang saat ini berjalan yaitu SMSCenter.

Bab 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan bahwa aplikasi Chatbot Kolektor sudah berhasil dibuat, pengembangan aplikasi Chatbot Kolektor pada PT. Indomobil Finance Indonesia dapat dilakukan dengan menggunakan metode *extreme programming* dan strategi *forward chaining*. Proses pengembangan aplikasi chatbot dikembangkan dari aplikasi SMSCenter yang saat ini sudah berjalan di PT. Indomobil Finance Indonesia untuk memenuhi layanan informasi bagi kolektor. Berdasarkan hasil analisa dan pengujian aplikasi dengan *blackbox testing* serta *user acceptance test*, persentase keberhasilan yang dicapai oleh aplikasi Chatbot Kolektor mencapai 95% dan persentase kegagalan aplikasi 5%, dengan angka tersebut maka aplikasi Chatbot Kolektor telah berhasil dibangun dan mampu menangani permintaan layanan informasi bagi kolektor di PT. Indomobil Finance Indonesia serta menunjukkan seluruh fungsional berjalan sesuai dengan yang diharapkan. Berdasarkan data analisa perbandingan antara aplikasi SMSCenter yang saat ini berjalan dengan aplikasi Chatbot Kolektor yang telah dibangun pada penelitian ini, maka dapat disimpulkan aplikasi Chatbot Kolektor telah mampu mengatasi permasalahan seringnya gangguan pada sistem pemberi informasi serta mampu melakukan penghematan biaya operasional yang dikeluarkan oleh PT. Indomobil Finance Indonesia.

5.2 Saran

Pembuatan aplikasi chatbot kolektor yang dilakukan pada penelitian ini sebaiknya dimodifikasi dengan menggunakan berbagai bahasa pemrograman dalam pembuatan *webservice*. Pengembangan aplikasi chatbot juga dapat menambahkan kecerdasan buatan (*Artificial Intelligence*) dan *Natural Language Programming* (NLP) agar *chatbot* mampu belajar dan menghasilkan *knowledge*nya sendiri. Pada penelitian berikutnya, pengembangan chatbot yang dilakukan pada penelitian ini dapat digunakan sebagai acuan untuk mengembangkan aplikasi chatbot pada platform lainnya seperti Whatsapp, dan Telegram.

DAFTAR PUSTAKA

- Akhsan, A. A. and Faizah, F. (2017). Analisa dan perancangan interaksi chatbot reminder dengan user-centered design. *Jurnal Sistem Informasi*, 2(13):78–89.
- Ananda, D. R., Firdha, I., Yusuf, M. A. S., and Ardiansyah (2018). Aplikasi chatbot (milki bot) yang terintegrasi dengan web cms untuk customer service pada ukm minsu. *Jurnal Cendikia*, XVI:100–106.
- Bagus, S., Ari, K., and Widhi, Y. (2018). Implementasi algoritme blake2s pada json web token (jwt) sebagai algoritme hashing untuk mekanisme autentikasi layanan rest-api. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(12):6269–6276.
- Benedictus, R. R., Wowor, H., and Sambul, A. (2017). Rancang bangun chatbot helpdesk untuk sistem informasi terpadu universitas sam ratulang. *EJournal Teknik Informatika*, 11(1).
- Carolina, I. and Adi, S. (2019). Penerapan metode extreme programming dalam perancangan aplikasi perhitungan kuota sks mengajar dosen. *Jurnal IKRA-ITH*, 3(1).
- Chen, H., editor (2016). *Platform strategies perspective on the OTT messaging services: A Case study of WeChat and Line*. Chen, H. H. In annual conference of International Telecommunication Society, Taipei, Taiwan.
- Elisabet, N. S. C. P. and Afrianto, M. (2015). Rancang bangun aplikasi chatbot informasi objek wisata kota bandung dengan pendekatan natural language processing. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 4(1):49–54.
- Extise, N. and Supriandi, A. (2016). Sistem informasi pengolahan data pendidikan anak usia dini (paud) terpadu amalia syukra padang. *Jurnal Edik Informatika*, pages 203–212.

- Hormansyah, D. S. and Utama, Y. P. (2018). Aplikasi chatbot berbasis web pada sistem informasi layanan publik kesehatan di malang dengan menggunakan metode tf-idf. *Jurnal Informatika Polinema*, 4:24–28.
- Jati, D. and Maarif, M. (2018). The development of chatbot application on line messaging platform for customer service in jogja sewa kamera. *Compiler*, 7(2):91–98.
- Junadhi and Mardainis. (2019). Line chatbot informasi cuaca wilayah indonesia. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 10(1):101–109.
- Lestari, J. (2016). Analisis sistem deteksi kerusakan komputer dengan menggunakan metode forward chaining. *Jurnal Inspiraton*, 6(1):19–27.
- LINE (2016). Line announces plans for expanded development and proliferation of chatbots. diakses pada 23 Januari 2020.
- Mufrizal, R. and Indarti, D. (2019). Refactoring arsitektur microservice pada aplikasi absensi pt. graha usaha teknik. *Jurnal Nasional Teknologi dan Sistem Informasi*, 5:57–68.
- Muhammad, S. S. S. and Rendra, S. (2018). Perancangan sistem informasi pariwisata kota sorong menggunakan web dan line@ chatbot sebagai media informasi pariwisata. *Jurnal Teknik Informasi dan Keamanan*, 4(2):71–77.
- Novella, A. (2015). Introduction microsoft sql server 2012.
- Setiawan, D. (2018). Dampak perkembangan teknologi informasi dan komunikasi terhadap budaya. *JURNAL SIMBOLIKA: Research and Learning in Communication Study*, 4:62.
- Sidi, M., Roeri, F., and Hendra, R. (2015). Pengujian aplikasi menggunakan black box testing boundary value analysis(studi kasus : Aplikasi prediksi kelulusan snmptn). *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3).
- S.Kemp (2019). All the data and trends you need to understand internet, social media, mobile, and e-commerce behaviours in 2019.
- Suryani, D. and Amalia, E. (2017). Aplikasi chatbot objek wisata jawa timur berbasis aiml. *SMARTICS Journal*, 3(2):47–54.

- Vallabhaneni, S. R. (2015). *Wiley CIAexcel Exam Review 2015, Part 3: Internal Audit Knowledge Elements*. John Wiley & Sons.
- Wijaya, M., Junaedy, and Arfandy, H. (2019). Perancangan chatbot untuk informasi penerimaan mahasiswa baru pada stmiik kharisma makassar. *KHARISMA Tech*, 14(1):14–23. Tersedia pada: <https://jurnal.kharisma.ac.id/kharismatech/article/view/14> (Diakses: 23 Januari 2020).
- Wijaya, M. H., Sotyohadi, and Setiawan, R. S. (2015). Perancangan chatbot pembelajaran pemrograman berorientasi object berbasis sistem modular. *SENATEK (Seminar Nasional Teknologi)*, 1(A):379–386.
- Yuniar, E. and Purnomo, H. (2019). Implementasi chatbot "alitta" asisten virtual dari balittas sebagai pusat informasi di balittas. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 13(1):24–35.

Lampiran 1

LineChatbotCollector_Handler.ashx.vb

```
Imports System.Web
Imports System.Web.Services
Imports System.IO
Imports System.Net

Public Class LineChatbotCollector_Handler
    Implements System.Web.IHttpHandler

    Sub ProcessRequest(ByVal context As HttpContext) Implements
        IHttpHandler.ProcessRequest
        Dim oJSONTextFormat As String = New StreamReader(context.
            Request.InputStream).ReadToEnd
        Dim oURL As String = ConfigurationManager.AppSettings("
            URL_LINE_LOCAL")
        Dim oJSON As String = SendPOST(oURL, oJSONTextFormat)
        Dim sendLine As String = SendPOSTLine("https://api.line.me
            /v2/bot/message/reply", oJSON)

        context.Response.ContentType = "application/json"
        context.Response.Headers.Add("Authorization", "Bearer " +
            "rvOLmlA7Q+
            qAF3x3PaxtXgHm8KTruPFtVwz3p9lUQpy5gEEZTC3Ghgp2w0bodKEgepe5k
            /tHAJT3XLCLKUDYFnBNkabojCruaYJfcUx/fA2Lym6EB9/5
            yZlkfH7FkXTBZeY15VON/j2ITcZKkvjBwwdB04t89/1O/w1cDnyilFU
            =")
        context.Response.Write(oJSON)
    End Sub

    Private Function SendPOST(_URL As String, _JSONFormat As
        String)
        Try
            Dim oHttpWebRequest As HttpWebRequest = DirectCast(
                WebRequest.Create(_URL), HttpWebRequest)
```

```

oHttpWebRequest.ContentType = "application/json"
oHttpWebRequest.Method = "POST"

Dim oStreamWriter As StreamWriter = New StreamWriter(
    oHttpWebRequest.GetRequestStream())

oStreamWriter.WriteLine(_JSONFormat)
oStreamWriter.Flush()
oStreamWriter.Close()

Dim oHttpResponse As HttpWebResponse = oHttpWebRequest
    .GetResponse()
Dim oStreamReader As StreamReader = New StreamReader(
    oHttpResponse.GetResponseStream())
Dim oResult = oStreamReader.ReadToEnd()

Return oResult.ToString
Catch ex As WebException
    Return ex.Message
End Try
End Function

Private Function SendPOSTLine(_URL As String, _JSONFormat As
String)
Try
    Dim oHttpWebRequest As HttpWebRequest = DirectCast(
        WebRequest.Create(_URL), HttpWebRequest)

    oHttpWebRequest.ContentType = "application/json"
    oHttpWebRequest.Headers.Add("Authorization", "Bearer "
        + "rvOLmIA7Q+
qAF3x3PaxtXgHm8KTruPFtVwz3p9lUQpy5gEEZTC3Ghgp2w0bodKEgepe5k
/tHAJT3XLCLKUdYFnBNkabojCruaYJfcUx/fA2Lym6EB9/5
yZlkfH7FkXTBZeY15VON/j2ITcZKkjvBwwdB04t89/1O/
w1cDnyilFU=")
    oHttpWebRequest.Method = "POST"

    Dim oStreamWriter As StreamWriter = New StreamWriter(
        oHttpWebRequest.GetRequestStream())

    oStreamWriter.WriteLine(_JSONFormat)
    oStreamWriter.Flush()
    oStreamWriter.Close()

    Dim oHttpResponse As HttpWebResponse = oHttpWebRequest

```

```
    .GetResponse()
Dim oStreamReader As StreamReader = New StreamReader(
    oHttpResponse.GetResponseStream())
Dim oResult = oStreamReader.ReadToEnd()

    Return oResult.ToString
Catch ex As WebException
    Return ex.Message
End Try
End Function

ReadOnly Property IsReusable() As Boolean Implements
    IHttpHandler.IsReusable
    Get
        Return False
    End Get
End Property

End Class
```

Lampiran 2

LineChatbotCollector_GetInfo.ashx.vb

```
Imports System.Web
Imports System.Web.Services
Imports System.IO
Imports System.Net

Public Class LineChatbotCollector_Handler
    Implements System.Web.IHttpHandler

    Sub ProcessRequest(ByVal context As HttpContext) Implements
        IHttpHandler.ProcessRequest
        Dim oJSONTextFormat As String = New StreamReader(context.
            Request.InputStream).ReadToEnd
        Dim oURL As String = ConfigurationManager.AppSettings("
            URL_LINE_LOCAL")
        Dim oJSON As String = SendPOST(oURL, oJSONTextFormat)
        Dim sendLine As String = SendPOSTLine("https://api.line.me
            /v2/bot/message/reply", oJSON)

        context.Response.ContentType = "application/json"
        context.Response.Headers.Add("Authorization", "Bearer " +
            "rvOLmlA7Q+
            qAF3x3PaxtXgHm8KTruPFtVwz3p9lUQpy5gEEZTC3Ghgp2w0bodKEgepe5k
            /tHAJT3XLCLKUDYFnBNkabojCruaYJfcUx/fA2Lym6EB9/5
            yZlkfH7FkXTBZeY15VON/j2ITcZKkvjBwwdB04t89/1O/w1cDnyilFU
            =")
        context.Response.Write(oJSON)
    End Sub

    Private Function SendPOST(_URL As String, _JSONFormat As
        String)
        Try
            Dim oHttpWebRequest As HttpWebRequest = DirectCast(
                WebRequest.Create(_URL), HttpWebRequest)
```

```

oHttpWebRequest.ContentType = "application/json"
oHttpWebRequest.Method = "POST"

Dim oStreamWriter As StreamWriter = New StreamWriter(
    oHttpWebRequest.GetRequestStream())

oStreamWriter.WriteLine(_JSONFormat)
oStreamWriter.Flush()
oStreamWriter.Close()

Dim oHttpResponse As HttpWebResponse = oHttpWebRequest
    .GetResponse()
Dim oStreamReader As StreamReader = New StreamReader(
    oHttpResponse.GetResponseStream())
Dim oResult = oStreamReader.ReadToEnd()

Return oResult.ToString
Catch ex As WebException
    Return ex.Message
End Try
End Function

Private Function SendPOSTLine(_URL As String, _JSONFormat As
String)
Try
    Dim oHttpWebRequest As HttpWebRequest = DirectCast(
        WebRequest.Create(_URL), HttpWebRequest)

    oHttpWebRequest.ContentType = "application/json"
    oHttpWebRequest.Headers.Add("Authorization", "Bearer "
        + "rvOLmIA7Q+
qAF3x3PaxtXgHm8KTruPFtVwz3p9lUQpy5gEEZTC3Ghgp2w0bodKEgepe5k
/tHAJT3XLCLKUdYFnBNkabojCruaYJfcUx/fA2Lym6EB9/5
yZlkfH7FkXTBZeY15VON/j2ITcZKkjvBwwdB04t89/1O/
w1cDnyilFU=")
    oHttpWebRequest.Method = "POST"

    Dim oStreamWriter As StreamWriter = New StreamWriter(
        oHttpWebRequest.GetRequestStream())

    oStreamWriter.WriteLine(_JSONFormat)
    oStreamWriter.Flush()
    oStreamWriter.Close()

    Dim oHttpResponse As HttpWebResponse = oHttpWebRequest

```

```
    .GetResponse()
Dim oStreamReader As StreamReader = New StreamReader(
    oHttpResponse.GetResponseStream())
Dim oResult = oStreamReader.ReadToEnd()

    Return oResult.ToString
Catch ex As WebException
    Return ex.Message
End Try
End Function

ReadOnly Property IsReusable() As Boolean Implements
    IHttpHandler.IsReusable
    Get
        Return False
    End Get
End Property

End Class
```

Lampiran 3

WSFunction.vb

```
Imports System.Web
Imports System.Web.Services
Imports System.IO
Imports System.Net
Imports IMFI.SCT.Presentation.Web
Imports IMFI.SCT.DomainObjects
Imports IMFI.Framework.Persistence.DomainObjects.Core
Imports System.Reflection
Imports System.Data.SqlClient
Imports IMFI.SCT.Common
Imports System.Text.RegularExpressions

Public Class WSFunction

#Region "Custom Method"

    Public Function CekLineUserByID(ByRef LineID As String) As
        ArrayList
        Dim oSearchCondition As String = "LineID = '" + LineID + "
        ,"
        Dim arLineUser As ArrayList = New LineUserMapper().
            RetrieveWithCondition(oSearchCondition, "LineUserID
            DESC")

        Return arLineUser
    End Function

    Public Function CekLineUserByNoData(ByRef LineUserID As Int64)
        As SCT_LineUser
        Return New LineUserMapper().Retrieve((New
            ObjectTransporter(New SCT_LineUser(LineUserID), (New
                UserIdentification). GetUserCredential)))
    End Function

```

```

Public Function CreateLineUser(ByRef oLine As SCT_LineUser) As
    Int64
    Return New LineUserMapper() . Insert(oLine)
End Function

Public Function CekLineKonsumen(ByRef LineID As String) As
    ArrayList
    Dim oSearchCondition As String = "IDLine = '" + LineID + "'"
    Dim arLineKonsumen As ArrayList = New LineKonsumenMapper()
        . RetrieveWithCondition(oSearchCondition , "NoDataLine
DESC")

    Return arLineKonsumen
End Function

Public Function CreateLineKonsumen(ByRef oLine2 As
    SCT_LineKonsumen) As SCT_LineKonsumen
    Dim oLineInsert = New LineKonsumenMapper() . Insert(oLine2)

    Return oLineInsert
End Function

Public Function PendaftaranLineUser(ByRef oRowLine As
    SCT_LineUser , ByVal incomingmessage As String) As String
    Dim Message = ""

    If oRowLine.DialogFlow = 0 Then
        'DialogFlow = 0 -> user diminta memasukkan nomor HP ke
        SCT_LineUser
        Message = "Silahkan melakukan pendaftaran dengan
mengetik DAFTAR[ spasi ]NomorHP"
        'update dialogflow
        oRowLine.DialogFlow = 1

    Dim LineUserUpdate As SCT_LineUser

    LineUserUpdate = (New LineUserMapper) . Update(oRowLine)

    ElseIf oRowLine.DialogFlow = 1 Then
        'DialogFlow = 1 -> User memasukkan Nomor HP
        If incomingmessage.Length >= 5 Then
            Dim application = incomingmessage.Substring(0, 6)

            If (application .ToUpper = "DAFTAR") Then

```

```

'Mengubah format 0xxxxxx menjadi +62xxxxxx
_____
Dim nomorHP = RemoveWhitespace(incomingmessage
    .Substring(6, incomingmessage.Length - 6))

If nomorHP.Substring(0, 1) = "+" Then
Else
    nomorHP = nomorHP.Substring(1, nomorHP.
        Length - 1)
    nomorHP = "+62" + nomorHP
End If

'Mengecek apakah nomor hp sesuai (semua harus
    integer)
If Regex.IsMatch(nomorHP.Substring(3, nomorHP.
    Length - 3), "^[0-9 ]+$") Then
    'Mengecek apakah nomor HP sudah terdaftar
        dalam database
    Dim SearchCondition = "noHP = '" + nomorHP
        +
    Dim arRegistrasiUserHdr As ArrayList = New
        RegistrasiUserHeaderMapper().
            RetrieveWithCondition(SearchCondition)

    If (arRegistrasiUserHdr.Count > 0) Then
        Dim arLineUser As ArrayList = New
            LineUserMapper().
                RetrieveWithCondition(
                    SearchCondition, "LineUserID ASC")

        If (arLineUser.Count > 0) Then
            Message = "Nomor HP anda sudah
                terdaftar di akun yang berbeda
                silahkan masukkan nomor HP yang
                lain"
        Else
            'Memasukan nomor HP ke database
            LineUser dan generate OTP
            oRowLine.OTP = GenerateOTP()
            oRowLine.NoHP = nomorHP
            oRowLine.DialogFlow = 2

            Dim LineUserUpdate As SCT_LineUser
                = (New LineUserMapper).Update(
                    oRowLine)

```

```
Message = "Silahkan masukkan kode  
OTP yang dikirimkan melalui SMS  
, atau batalkan pendaftaran  
nomor HP dengan mengetik BATAL"
```

```
'2018.12.05 - sidik - ini kodingan  
untuk kirim OTP via SMS  
Dim connectionString As String =  
    ConfigurationManager.  
    ConnectionStrings("DBData").  
   ConnectionString  
Dim connection = New SqlConnection  
(connectionString)  
Dim command As New SqlCommand  
  
connection.Open()  
command.Connection = connection  
command.CommandType = CommandType.  
    StoredProcedure  
  
Dim cmd As SqlCommand = New  
    SqlCommand("SP_Send OTP",  
    connection)  
  
cmd.CommandType = CommandType.  
    StoredProcedure  
cmd.Parameters.AddWithValue("@NoHP  
    ", oRowLine.NoHP)  
cmd.Parameters.AddWithValue("@  
    ReplyMessage", "Silahkan  
masukkan OTP : " + oRowLine.OTP  
)  
cmd.Parameters.AddWithValue("@  
    ApplicationFlag", "KOL")  
cmd.Parameters.AddWithValue("@  
    DeviceID", "3")  
cmd.Parameters.AddWithValue("@  
    IsRequestStatus", "1")  
  
' Ask the command to create an  
SqlDataReader on the result of  
the sproc'  
Using r = cmd.ExecuteReader()
```

```

        End Using
        'Sampe sini kodingan untuk kirim
         OTP via SMS
    End If

    Else
        Message = "Nomor anda Hp tidak
                    terdaftar dalam database kami,
                    silahkan hubungi ITHelpdesk
                    Perusahaan Kami"
    End If

    Else
        Message = "Format nomor HP salah. Silahkan
                    masukkan nomor hp kembali"
    End If

    Else
        Message = "Silahkan melakukan pendaftaran
                    dengan mengetik DAFTAR[ spasi ]NomorHP"
    End If

    Else
        Message = "Silahkan melakukan pendaftaran dengan
                    mengetik DAFTAR[ spasi ]NomorHP"
    End If

ElseIf oRowLine.DialogFlow = 2 Then
    'DialogFlow = 2 -> User memasukkan kode OTP
    Dim kodeOTP = incomingmessage

    If (kodeOTP = oRowLine.OTP) Then
        oRowLine.Status = 1
        oRowLine.DialogFlow = 0

        Dim LineUserUpdate As SCT_LineUser = (New
            LineUserMapper).Update(oRowLine)

        Message = "Pendaftaran Berhasil. Silahkan ketik
                    MENU untuk melihat menu collection yg dapat
                    Anda akses"

    ElseIf incomingmessage.ToUpper = "BATAL" Then
        oRowLine.Status = 0
        oRowLine.DialogFlow = 1

        Dim LineUserUpdate As SCT_LineUser = (New
            LineUserMapper).Update(oRowLine)

```

```

        Message = "Pendaftaran dibatalkan , silahkan
                    melakukan pendaftaran kembali dengan mengetik
                    DAFTAR[ spasi ]Nomor HP"
    Else
        Message = "Kode OTP salah , Silahkan masukkan kode
                    OTP yang dikirimkan melalui SMS, atau batalkan
                    pendaftaran nomor HP dengan mengetik BATAL"
    End If

End If

Return Message
End Function

Public Function GetInfo(ByRef LineID As String , ByVal
incomingmessage As String) As String
Dim message = ""
Dim oResult As Object

'2019.03.20 – sidik – ini deklarasi variabel untuk nyimpen
hasil query
Dim Nama As New List(Of String)
Dim Modul As New List(Of String)
Dim Template As New List(Of String)
Dim Contoh As New List(Of String)

'2019.03.20 – sidik – ini untuk koneksi ke database dan
jalanin query
Dim connectionString As String = ConfigurationManager.
.ConnectionStrings("DBData").ConnectionString
Using connObj As New SqlConnection(
    connectionString)
    Using cmdObj As New SqlCommand()
        "SELECT slu.LineUserID , slu.LineID , slu.NoHP, srh.
        Nama, slu.Status AS isUserAktif ,sm.Keterangan ,
        sm.KodeModul, sm.KodeModul + '[ spasi ]' +
        replace(replace(sm.[ Parameter ], '@' , '' ) , ',' , '[
        spasi ]') AS Template ,sm.SMSTemplate AS Contoh ,
        srh.IsAktif AS isModulAktif FROM dbo.
        SCT_LineUser slu INNER JOIN dbo.
        SCT_Registrasi_Hdr srh ON srh.NoHP = slu.NoHP
        INNER JOIN dbo.SCT_Registrasi_Dtl srd ON srd.
        NoDataRegistrasiHdr = srh.NoDataRegistrasiHdr
        INNER JOIN dbo.SCT_Modul sm ON sm.NoDataModul =

```

```

        srd.NoDataModul WHERE slu.LineID ='" + LineID
        + "'",
        connObj)

        connObj.Open()
        Using readerObj As SqlClient.SqlDataReader =
            cmdObj.ExecuteReader
            'This will loop through all returned records
            While readerObj.Read
                Nama.Add(readerObj("Nama").ToString)
                Modul.Add(readerObj("Keterangan").ToString
                    )
                Template.Add(readerObj("Template").
                    ToString)
                Contoh.Add(readerObj("Contoh").ToString)
                'handle returned value before next loop
                here
            End While
        End Using
        connObj.Close()
    End Using
End Using
'jalankan query selesai sampe disini

'2019.03.20 – sidik – ngecek pesan yg masuk dari user
lebih dari 3 karakter atau ngga
If incomingmessage.Length > 3 Then
    Dim pesanMasuk = incomingmessage.Substring(0, 4)
    '2019.03.20 – sidik – ngecek apakah pesan yg masuk
    dari user termasuk ke kata kunci pengecekan
    KOLEKTOR
    If pesanMasuk.ToUpper = "KOL " Or pesanMasuk.ToUpper =
        "BLK " Or pesanMasuk.ToUpper = "MSN " Or
        pesanMasuk.ToUpper = "RNK " Then
        '2019.03.20 – sidik – ini untuk proses inserting
        ke SCT_InboxLine

        'mengambil data dari tabel SCT_LineUser
        Dim oLineNoHP As ArrayList = New LineUserMapper().
            RetrieveWithCondition("LineID = '" + LineID +
            "'", "LineID desc")

        'memasukkan data ke tabel SCT_InboxLineuser
        Dim dateToday As DateTime = DateTime.Today
        Dim oNewInboxLine = New SCT_InboxLine

```

```

' oNewInboxLine.InboxLineID = nextID
oNewInboxLine.SMSMessage = incomingmessage
oNewInboxLine.NoHP = CType(oLineNoHP(0),
    SCT_LineUser).NoHP
oNewInboxLine.WaktuSMS = dateToday

If incomingmessage.Length > 3 Then
    oNewInboxLine.ApplicationFlag =
        incomingmessage.Substring(0, 3).ToUpper
Else
    oNewInboxLine.ApplicationFlag = "Invalid"
End If

oResult = New InboxLineMapper().Insert(
    oNewInboxLine)

'mengambil balasan dari sistem
Dim oReplyInboxLine As SCT_InboxLine = New
    InboxLineMapper().Retrieve(oResult)

If Not IsNothing(oReplyInboxLine) Then
    message = oReplyInboxLine.ReplyMessage
Else
    message = "Reply tidak ditemukan"
End If

'2019.03.20 – sidik – ini untuk ngecek kalo pesan
yg diinput user adalah HELP
ElseIf pesanMasuk.ToUpper = "HELP" Then
    If Modul.Count <> 0 Then
        message = "Untuk Template Modul, dapat diketik
            sebagai berikut : \n"

        For Each templateModul As String In Template
            message = message + "\n" _
                & templateModul
        Next

        message = message + "\n\n" _
            & "Untuk Contoh pengecekan Modul,
                dapat diketik sebagai berikut :
                \n"

```

```

For Each contohModul As String In Contoh
    message = message + "\n" _
        & contohModul
    Next

End If
Else
    If Modul.Count <> 0 Then
        message = "Hai, " + Nama.Item(0) + "\n" _
            & "Menu yang dapat anda akses
            adalah : "

        For Each isiModul As String In Modul
            message = message + "\n" _
                & isiModul
        Next

        message = message + "\n\n" _
            & "Jika anda ingin mengetahui
            template pesan, silahkan ketik
            HELP"
    End If
End If
Else
    If Modul.Count <> 0 Then
        message = "Hai, " + Nama.Item(0) + "\n" _
            & "Menu yang dapat anda akses adalah :
            "

        For Each isiModul As String In Modul
            message = message + "\n" _
                & isiModul
        Next

        message = message + "\n\n" _
            & "Jika anda ingin mengetahui template
            pesan, silahkan ketik HELP"
    End If
End If

Return message
End Function

```

```

Protected Function GenerateOTP() As String
    Dim alphabets As String = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

```

```

Dim small_alphabets As String = "
    abcdefghijklmnopqrstuvwxyz"
Dim numbers As String = "1234567890"
Dim characters As String = numbers
Dim length As Integer = Integer.Parse(6)
Dim otp As String = String.Empty

For i As Integer = 0 To length - 1
    Dim character As String = String.Empty

    Do
        Dim index As Integer = New Random().Next(0,
            characters.Length)

        character = characters.ToCharArray()(index) .
            ToString()
    Loop While otp.IndexOf(character) <> -1
    otp += character
    Next

    Return otp
End Function

Function RemoveWhitespace(fullString As String) As String
    Return New String(fullString.Where(Function(x) Not Char.
        IsWhiteSpace(x)).ToArray())
End Function

#End Region
End Class

```

Lampiran 4

SCT_Modul.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
'*          Copyright © 2006 IMF
'*          *
'*          *
' *****

#End Region

#Region "Summary"

' *****
'*          Author      : IMF Development Team
'*          Purpose     : SCT_Modul Domain Object.
'*          Called By   :
'*          *
' *****

#End Region

#Region "History"

' *****
'*          Created On  : 26/4/2017 – 3:15:44 PM
'*          Modify By   : {Name} – {Date}
'*                      – {Description Here}
'*          *
' *****
```

```

' *
'* ****
* /



#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistance.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DomainObjects
    <Serializable ()> _
    Public Class SCT_Modul
        Inherits AbstractDomainObject

#Region "Constructors/Destructors/Finalizers"
        Public Sub New()
            End Sub

        Public Sub New(NoDataModul As Int32)

            _NoDataModul = NoDataModul
        End Sub
#End Region

#Region "Private Variables"
    Private _NoDataModul As int32
    Private _KodeModul As String = String.Empty
    Private _Keterangan As String = String.Empty
    Private _SPName As String = String.Empty
    Private _SPLine As String = String.Empty

```

```

Private _JumlahParameter As int32
Private _Parameter As String = String.Empty
Private _SMSTemplate As String = String.Empty
Private _DibuatOleh As String = String.Empty
Private _WaktuDibuat As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)
Private _DiubahOleh As String = String.Empty
Private _WaktuDiubah As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)

#End Region

#Region "Public Properties"
    Public Property NoDataModul As int32
        Get
            Return _NoDataModul
        End Get
        Set(value As int32)
            _NoDataModul = value
        End Set
    End Property

    Public Property KodeModul As String
        Get
            Return _KodeModul
        End Get
        Set(value As String)
            _KodeModul = value
        End Set
    End Property

    Public Property Keterangan As String
        Get
            Return _Keterangan
        End Get
        Set(value As String)
            _Keterangan = value
        End Set
    End Property

    Public Property SPName As String
        Get
            Return _SPName

```

```
    End Get
    Set(value As String)
        _SPName = value
    End Set
End Property
```

```
Public Property SPLine As String
    Get
        Return _SPLine
    End Get
    Set(value As String)
        _SPLine = value
    End Set
End Property
```

```
Public Property JumlahParameter As int32
    Get
        Return _JumlahParameter
    End Get
    Set(value As int32)
        _JumlahParameter = value
    End Set
End Property
```

```
Public Property Parameter As String
    Get
        Return _Parameter
    End Get
    Set(value As String)
        _Parameter = value
    End Set
End Property
```

```
Public Property SMSTemplate As String
    Get
        Return _SMSTemplate
    End Get
    Set(value As String)
        _SMSTemplate = value
    End Set
```

```

End Property

Public Property DibuatOleh As String
    Get
        Return _DibuatOleh
    End Get
    Set(value As String)
        _DibuatOleh = value
    End Set
End Property

Public Property WaktuDibuat As DateTime
    Get
        Return _WaktuDibuat
    End Get
    Set(value As DateTime)
        _WaktuDibuat = value
    End Set
End Property

Public Property DiubahOleh As String
    Get
        Return _DiubahOleh
    End Get
    Set(value As String)
        _DiubahOleh = value
    End Set
End Property

Public Property WaktuDiubah As DateTime
    Get
        Return _WaktuDiubah
    End Get
    Set(value As DateTime)
        _WaktuDiubah = value
    End Set
End Property

#End Region
End Class
End Namespace

```

Lampiran 5

SCT_LineUser.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
'*          Copyright © 2006 IMF
'*          *
'*          *
' *****

#End Region

#Region "Summary"

' *****
'*          Author      : IMF Development Team
'*          *
'*          Purpose     : SCT_LineUser Domain Object.
'*          Called By   :
'*          *
' *****

#End Region

#Region "History"

' *****
'*          Created On  : 10/5/2018 - 1:17:33 PM
'*          *
'*          Modify By   : {Name} - {Date}
'*          - {Description Here}
'*          *
' *****
```

```

'*                                         *
'*****/




#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.ComponentModel.DataAnnotations

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistence.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DomainObjects
    <Serializable ()> _
    Public Class SCT_LineUser
        Inherits AbstractDomainObject

#Region "Constructors/Destructors/Finalizers"
        Public Sub New()
            End Sub

        Public Sub New(LineUserID As Int32)

            _LineUserID = LineUserID
            End Sub
#End Region

#Region "Private Variables"
        Private _LineUserID As int32
        Private _LineID As String = String.Empty
        Private _NoHP As String = String.Empty
        Private _Status As int32

```

```
Private _DialogFlow As Int32
Private _OTP As String = String.Empty
Private _DibuatOleh As String = String.Empty
Private _WaktuDibuat As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)
Private _DiubahOleh As String = String.Empty
Private _WaktuDiubah As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)
#End Region
```

```
#Region "Public Properties"
```

```
<Key()> _
Public Property LineUserID As int32
    Get
        Return _LineUserID
    End Get
    Set(value As int32)
        _LineUserID = value
    End Set
End Property
```

```
Public Property LineID As String
    Get
        Return _LineID
    End Get
    Set(value As String)
        _LineID = value
    End Set
End Property
```

```
Public Property NoHP As String
    Get
        Return _NoHP
    End Get
    Set(value As String)
        _NoHP = value
    End Set
End Property
```

```
Public Property Status As int32
    Get
        Return _Status
    End Get
    Set(value As int32)
        _Status = value
    End Set
End Property
```

```
Public Property DialogFlow As int32
    Get
        Return _DialogFlow
    End Get
    Set(value As int32)
        _DialogFlow = value
    End Set
End Property
```

```
Public Property OTP As String
    Get
        Return _OTP
    End Get
    Set(value As String)
        _OTP = value
    End Set
End Property
```

```
Public Property DibuatOleh As String
    Get
        Return _DibuatOleh
    End Get
    Set(value As String)
        _DibuatOleh = value
    End Set
End Property
```

```
Public Property WaktuDibuat As DateTime
    Get
        Return _WaktuDibuat
    End Get
    Set(value As DateTime)
        _WaktuDibuat = value
    End Set
End Property
```

```
Public Property DiubahOleh As String
    Get
        Return _DiubahOleh
    End Get
    Set(value As String)
        _DiubahOleh = value
    End Set
End Property
```

```
Public Property WaktuDiubah As DateTime
    Get
        Return _WaktuDiubah
    End Get
    Set(value As DateTime)
        _WaktuDiubah = value
    End Set
End Property
```

```
#End Region
End Class
End Namespace
```

Lampiran 6

SCT_LineMessage.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
'*          Copyright © 2006 IMF
'*          *
'*          *
' *****

#End Region

#Region "Summary"

' *****
'*          Author      : IMF Development Team
'*          *
'*          Purpose     : SCT_LineMessage Domain Object.
'*          Called By   :
'*          *
' *****

#End Region

#Region "History"

' *****
'*          Created On  : 10/8/2018 - 2:08:38 PM
'*          *
'*          Modify By   : {Name} - {Date}
'*          - {Description Here}
'*          *
' *****
```

```

'*                                         *
'*****/




#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.ComponentModel.DataAnnotations

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistence.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DomainObjects
    <Serializable ()> _
    Public Class SCT_LineMessage
        Inherits AbstractDomainObject

#Region "Constructors/Destructors/Finalizers"
        Public Sub New()
            End Sub

        Public Sub New(NoDataID As Int32)

            _NoDataID = NoDataID
            End Sub
#End Region

#Region "Private Variables"
        Private _NoDataID As int32
        Private _LineID As String = String.Empty
        Private _Message As String = String.Empty
        Private _ReplyToken As String = String.Empty

```

```
#End Region

#Region "Public Properties"

'<Key()> _
Public Property NoDataID As Int32
    Get
        Return _NoDataID
    End Get
    Set(value As Int32)
        _NoDataID = value
    End Set
End Property
```

```
Public Property LineID As String
    Get
        Return _LineID
    End Get
    Set(value As String)
        _LineID = value
    End Set
End Property
```

```
Public Property Message As String
    Get
        Return _Message
    End Get
    Set(value As String)
        _Message = value
    End Set
End Property
```

```
Public Property ReplyToken As String
    Get
        Return _ReplyToken
    End Get
    Set(value As String)
        _ReplyToken = value
    End Set
```

```

        End Property

#End Region
    End Class

    Public Class SCT_LineLocation
        Inherits AbstractDomainObject

#Region "Constructors/Destructors/Finalizers"
    Public Sub New()
        End Sub

        Public Sub New(NoDataID As Int32)

            _NoDataID = NoDataID
        End Sub
#End Region

#Region "Private Variables"
    Private _NoDataID As Int32
    Private _Type As String = String.Empty
    Private _Address As String = String.Empty
    Private _Latitude As String = String.Empty
    Private _Longitude As String = String.Empty
    Private _ReplyToken As String = String.Empty
#End Region

#Region "Public Properties"

    '<Key()> _
    Public Property NoDataID As Int32
        Get
            Return _NoDataID
        End Get
        Set(value As Int32)
            _NoDataID = value
        End Set
    End Property

    Public Property Type As String
        Get
            Return _Type

```

```

        End Get
        Set(value As String)
            _Type = value
        End Set
    End Property

    Public Property Address As String
        Get
            Return _Address
        End Get
        Set(value As String)
            _Address = value
        End Set
    End Property

    Public Property Latitude As String
        Get
            Return _Latitude
        End Get
        Set(value As String)
            _Latitude = value
        End Set
    End Property

    Public Property Longitude As String
        Get
            Return _Longitude
        End Get
        Set(value As String)
            _Longitude = value
        End Set
    End Property

    Public Property ReplyToken As String
        Get
            Return _ReplyToken
        End Get
        Set(value As String)
            _ReplyToken = value
        End Set
    End Property
#End Region

```

End Class
End Namespace

Lampiran 7

SCT_InboxLine.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
'*          Copyright © 2006 IMF
'* *****
'*End Region

#Region "Summary"

' *****
'*      Author      : IMF Development Team
'*      Purpose     : SCT_InboxLine Domain Object.
'*      Called By   :
'* *****
'*End Region

#Region "History"

' *****
'*      Created On  : 10/8/2018 - 10:39:11 AM
'*      Modify By   : {Name} - {Date}
'*                  - {Description Here}
'******
```

```

'*
'* ****
#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.ComponentModel.DataAnnotations

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistence.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DomainObjects
    <Serializable ()> _
    Public Class SCT_InboxLine
        Inherits AbstractDomainObject

#Region "Constructors/Destructors/Finalizers"
        Public Sub New()
            End Sub

        Public Sub New(InboxLineID As Int32)
            _InboxLineID = InboxLineID
        End Sub
#End Region

#Region "Private Variables"
    Private _InboxLineID As int32
    Private _NoHP As String = String.Empty
    Private _Register As Boolean
    Private _SMSMessage As String = String.Empty

```

```

Private _WaktuSMS As DateTime = CType(System.Data.SqlTypes
    .SqlDateTime.MinValue.Value, DateTime)
Private _Proses As Boolean
Private _ReplyMessage As String = String.Empty
Private _WaktuReply As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)
Private _ApplicationFlag As String = String.Empty
Private _DibuatOleh As String = String.Empty
Private _WaktuDibuat As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)
Private _DiubahOleh As String = String.Empty
Private _WaktuDiubah As DateTime = CType(System.Data.
    SqlTypes.SqlDateTime.MinValue.Value, DateTime)

#End Region

#Region "Public Properties"

'<Key()> _
Public Property InboxLineID As Int32
    Get
        Return _InboxLineID
    End Get
    Set(value As Int32)
        _InboxLineID = value
    End Set
End Property

Public Property NoHP As String
    Get
        Return _NoHP
    End Get
    Set(value As String)
        _NoHP = value
    End Set
End Property

Public Property Register As Boolean
    Get
        Return _Register
    End Get
    Set(value As Boolean)

```

```
    _Register = value  
End Set  
End Property
```

```
Public Property SMSMessage As String  
Get  
    Return _SMSMessage  
End Get  
Set(value As String)  
    _SMSMessage = value  
End Set  
End Property
```

```
Public Property WaktuSMS As DateTime  
Get  
    Return _WaktuSMS  
End Get  
Set(value As DateTime)  
    _WaktuSMS = value  
End Set  
End Property
```

```
Public Property Proses As Boolean  
Get  
    Return _Proses  
End Get  
Set(value As Boolean)  
    _Proses = value  
End Set  
End Property
```

```
Public Property ReplyMessage As String  
Get  
    Return _ReplyMessage  
End Get  
Set(value As String)  
    _ReplyMessage = value
```

```
    End Set  
End Property
```

```
Public Property WaktuReply As DateTime  
    Get  
        Return _WaktuReply  
    End Get  
    Set(value As DateTime)  
        _WaktuReply = value  
    End Set  
End Property
```

```
Public Property ApplicationFlag As String  
    Get  
        Return _ApplicationFlag  
    End Get  
    Set(value As String)  
        _ApplicationFlag = value  
    End Set  
End Property
```

```
Public Property DibuatOleh As String  
    Get  
        Return _DibuatOleh  
    End Get  
    Set(value As String)  
        _DibuatOleh = value  
    End Set  
End Property
```

```
Public Property WaktuDibuat As DateTime  
    Get  
        Return _WaktuDibuat  
    End Get  
    Set(value As DateTime)  
        _WaktuDibuat = value  
    End Set
```

```
End Property
```

```
Public Property DiubahOleh As String
    Get
        Return _DiubahOleh
    End Get
    Set(value As String)
        _DiubahOleh = value
    End Set
End Property
```

```
Public Property WaktuDiubah As DateTime
    Get
        Return _WaktuDiubah
    End Get
    Set(value As DateTime)
        _WaktuDiubah = value
    End Set
End Property
```

```
#End Region
End Class
End Namespace
```

Lampiran 8

SCT_JsonObject.vb

```
Public Class SCT_JsonObject
    Public Property events() As Events()

End Class

Public Class Events
    Public type As String
    Public replyToken As String
    Public source As Source
    Public timestamp As String
    Public message As Message
End Class

Public Class Source
    Public userID As String
    Public type As String
End Class

Public Class Message
    Public type As String
    Public id As String
    Public text As String
End Class

Public Class SCT_JsonObjectLocationLine
    Public Property events() As EventsLocation()

End Class

Public Class EventsLocation
    Public type As String
    Public replyToken As String
    Public source As SourceLocation

```

```

    Public timestamp As String
    Public message As MessageLocation
End Class

Public Class SourceLocation
    Public userID As String
    Public type As String
End Class

Public Class MessageLocation
    Public type As String
    Public title As String
    Public address As String
    Public latitude As String
    Public longitude As String
End Class

Public Class SCT_JsonObjectPostbackLine
    Public Property events() As EventsPostback()
End Class

Public Class EventsPostback
    Public type As String
    Public replyToken As String
    Public source As SourcePostback
    Public timestamp As String
    Public postback As Postback
End Class

Public Class SourcePostback
    Public userID As String
    Public type As String
End Class

Public Class Postback
    Public data As String
End Class

Public Class SCT_JsonObjectTelegram
    Public Property update_id As String
    Public Property message As Messages

```

```
End Class

Public Class Messages
    Public message_id As String
    Public from As From
    Public chat As Chat
    Public date_chat As DateTime
    Public text As String
End Class

Public Class From
    Public id As String
    Public is_Bot As Boolean
    Public first_name As String
    Public last_name As String
    Public language_code As String
End Class

Public Class Chat
    Public id As String
    Public first_name As String
    Public last_name As String
    Public type As String
End Class
```

Lampiran 9

SCT_ModulMapper.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
'*          Copyright © 2006 IMF
'*          *
'*          *
' *****

#End Region

#Region "Summary"

' *****
'*          Author      : IMF Development Team
'*          Purpose     : SCT_Modul Objects Mapper.
*
'*          Called By   :
'*          *
' *****

#End Region

#Region "History"

' *****
'*          Created On  : 26/4/2017 – 3:16:06 PM
'*          *
'*          Modify By   : {Name} – {Date}
'*                      – {Description Here}
'*          *
' *****
```

```

'*                                         *
'*****/




#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.Data
Imports System.Collections

#End Region

#Region "Custom Namespace Imports"

Imports Microsoft.Practices.EnterpriseLibrary.Data
'Imports Microsoft.Practices.EnterpriseLibrary.ExceptionHandling
'Imports Microsoft.Practices.EnterpriseLibrary.Logging

Imports IMFI.Framework.Persistence.DomainObjects.Core
Imports IMFI.Framework.Persistence.Mapper
Imports IMFI.SCT.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DataMapper
    Public Class SCT_ModulMapper
        Inherits AbstractMapper
    End Class
End Namespace

#Region "Constructors/Destructors/Finalizers"

    Public Sub New()
        Db = DatabaseFactory.CreateDatabase()
    End Sub

#End Region

#Region "Private Variables"

```

```

Private m_InsertStatement As String = "USP_SCT_ModulInsert
"
Private m_UpdateStatement As String = "USP_SCT_ModulUpdate
"
Private m_RetrieveStatement As String = "
    USP_SCT_ModulRetrieve"
Private m_RetrieveListStatement As String = "
    USP_SCT_ModulRetrieveList"
Private m_DeleteStatement As String = "USP_SCT_ModulDelete
"
Private m_RetrieveListPagingStatement As String = "
    USP_SCT_ModulRetrievePagingList"
Private m_RetrieveListPagingRowCountStatement As String =
    "USP_SCT_ModulRetrievePagingRowCount"

#End Region

#Region "Protected Methods"

Protected Overrides Function GetNewID(ByVal
    oDBCommandWrapper As System.Data.Common.DbCommand) As
Object
    Return Db.GetParameterValue(DBCommand, "@NoDataModul")
End Function

Protected Overrides Function GetInsertParameter(ByVal
    oObjectTransporter As IObjectTransporter) As System.
    Data.Common.DbCommand

    Dim oUserCredential As IUserCredential = CType(
        oObjectTransporter, IObjectTransporter).
        UserCredential
    Dim oSCT_Modul As SCT_Modul = CType(CType(
        oObjectTransporter, IObjectTransporter).
        DomainObject, SCT_Modul)

    DBCommand = Db.GetStoredProcCommand(Me.
        m_InsertStatement)

    Db.AddInParameter(DBCommand, "@NoDataModul", DbType.
        Int32, oSCT_Modul.NoDataModul)

    Db.AddInParameter(DBCommand, "@KodeModul", DbType.

```

```

        AnsiString , oSCT_Modul.KodeModul)

Db.AddInParameter(DBCommand, "@Keterangan" , DbType.
        AnsiString , oSCT_Modul.Keterangan)

Db.AddInParameter(DBCommand, "@SPName" , DbType.
        AnsiString , oSCT_Modul.SPName)

Db.AddInParameter(DBCommand, "@SPLine" , DbType.
        AnsiString , oSCT_Modul.SPLine)

Db.AddInParameter(DBCommand, "@JumlahParameter" ,
        DbType.Int32 , oSCT_Modul.JumlahParameter)

Db.AddInParameter(DBCommand, "@Parameter" , DbType.
        AnsiString , oSCT_Modul.Parameter)

Db.AddInParameter(DBCommand, "@SMSTemplate" , DbType.
        AnsiString , oSCT_Modul.SMSTemplate)

    Return DBCommand
End Function

Protected Overrides Function GetUpdateParameter(ByVal
    oObjectTransporter As IObjectTransporter) As System.
    Data.Common.DbCommand
    Dim objUserCredential As IUserCredential = CType(
        oObjectTransporter , IObjectTransporter).
        UserCredential
    Dim oSCT_Modul As SCT_Modul = CType(CType(
        oObjectTransporter , IObjectTransporter).
        DomainObject , SCT_Modul)

    DBCommand = Db.GetStoredProcCommand(Me.
        m_UpdateStatement)
    Db.AddInParameter(DBCommand, "@NoDataModul" , DbType.
        Int32 , oSCT_Modul.NoDataModul)
    Db.AddInParameter(DBCommand, "@KodeModul" , DbType.
        AnsiString , oSCT_Modul.KodeModul)
    Db.AddInParameter(DBCommand, "@Keterangan" , DbType.
        AnsiString , oSCT_Modul.Keterangan)
    Db.AddInParameter(DBCommand, "@SPName" , DbType.
        AnsiString , oSCT_Modul.SPName)
    Db.AddInParameter(DBCommand, "@SPLine" , DbType.
        AnsiString , oSCT_Modul.SPLine)

```

```

        Db.AddInParameter(DBCommand, "@JumlahParameter",
                           DbType.Int32, oSCT_Modul.JumlahParameter)
        Db.AddInParameter(DBCommand, "@Parameter", DbType.
                           AnsiString, oSCT_Modul.Parameter)
        Db.AddInParameter(DBCommand, "@SMSTemplate", DbType.
                           AnsiString, oSCT_Modul.SMSTemplate)

    End Function

Protected Overrides Function GetRetrieveParameter(ByVal
oObjectTransporter As IObjectTransporter) As System.
Data.Common.DbCommand

    Dim oSCT_Modul As SCT_Modul
    If Not IsNothing(oObjectTransporter) Then
        oSCT_Modul = CType(CType(oObjectTransporter,
                           IObjectTransporter).DomainObject, SCT_Modul)
    End If
    DBCommand = Db.GetStoredProcCommand(Me.
                                         m_RetrieveStatement)

    Db.AddInParameter(DBCommand, "@NoDataModul", DbType.
                           Int32, oSCT_Modul.NoDataModul)

    Return DBCommand
End Function

Protected Overrides Function GetRetrieveListParameter(
    ByVal oObjectTransporter As IObjectTransporter) As
System.Data.Common.DbCommand
    Dim oSCT_Modul As SCT_Modul
    If Not IsNothing(oObjectTransporter) Then
        oSCT_Modul = CType(CType(oObjectTransporter,
                           IObjectTransporter).DomainObject, SCT_Modul)
    End If
    DBCommand = Db.GetStoredProcCommand(Me.
                                         m_RetrieveListStatement)
    Return DBCommand
End Function

Protected Overrides Function
GetRetrieveListPagingParameter(ByVal PageSize As

```

```

Integer , ByVal CurrentPage As Integer , ByVal OrderBy As
String , ByVal oObjectTransporter As IObjectTransporter
, ByVal SearchCondition As String) As System.Data.
Common.DbCommand
Dim oSCT_Modul As SCT_Modul
If Not IsNothing(oObjectTransporter) Then
    oSCT_Modul = CType(CType(oObjectTransporter ,
    IObjectTransporter).DomainObject , SCT_Modul)
End If
DBCommand = Db.GetStoredProcCommand(Me.
    m_RetrieveListPagingStatement)

Db.AddInParameter(DBCommand, "@PageSize" , DbType.Int32
    , PageSize)
Db.AddInParameter(DBCommand, "@CurrentPage" , DbType.
    Int32 , CurrentPage)
Db.AddInParameter(DBCommand, "@OrderBy" , DbType.String
    , OrderBy.Split(" "c)(0))
Db.AddInParameter(DBCommand, "@Order" , DbType.String ,
    OrderBy.Split(" "c)(1))
Db.AddInParameter(DBCommand, "@SearchCondition" ,
    DbType.String , IIf(IsNothing(SearchCondition) , "" ,
    SearchCondition))

Return DBCommand
End Function

Protected Overrides Function
GetRetrieveListPagingRowCountParameter(ByVal
oObjectTransporter As IObjectTransporter , ByVal
SearchCondition As String) As System.Data.Common.
DbCommand
Dim oSCT_Modul As SCT_Modul
If Not IsNothing(oObjectTransporter) Then
    oSCT_Modul = CType(CType(oObjectTransporter ,
    IObjectTransporter).DomainObject , SCT_Modul)
End If
DBCommand = Db.GetStoredProcCommand(Me.
    m_RetrieveListPagingRowCountStatement)

Db.AddInParameter(DBCommand, "@SearchCondition" ,
    DbType.String , IIf(IsNothing(SearchCondition) , "" ,
    SearchCondition))

Return DBCommand

```

```

End Function

Protected Overrides Function GetDeleteParameter(
    oObjectTransporter As IObjectTransporter) As System.
Data.Common.DbCommand
    Dim oUserCredential As IUserCredential = CType(
        oObjectTransporter, IObjectTransporter).
    UserCredential
    Dim oSCT_Modul As SCT_Modul = CType(CType(
        oObjectTransporter, IObjectTransporter).
    DomainObject, SCT_Modul)
    DBCommand = Db.GetStoredProcCommand(Me.
        m_DeleteStatement)

Db.AddInParameter(DBCommand, "@NoDataModul", DbType.
    Int32, oSCT_Modul.NoDataModul)
Return DBCommand
End Function

Protected Overrides Function DoRetrieve(ByVal dr As
    IDataReader) As Object
    Dim oSCT_Modul As SCT_Modul = Nothing
    While (dr.Read())
        oSCT_Modul = Me.CreateObject(dr)
    End While

    Return oSCT_Modul
End Function

Protected Overrides Function DoRetrieveList(ByVal dr As
    IDataReader) As ArrayList
    Dim oSCT_ModulList As ArrayList = New ArrayList()
    While (dr.Read())
        Dim oSCT_Modul As SCT_Modul = Me.CreateObject(dr)
        oSCT_ModulList.Add(oSCT_Modul)
    End While

    Return oSCT_ModulList
End Function

#End Region

```

```

#Region "Private Methods"

    Private Function CreateObject(ByVal dr As IDataReader) As
        SCT_Modul
        Dim oSCT_Modul As SCT_Modul = New SCT_Modul()

        oSCT_Modul.NoDataModul = CType(dr("NoDataModul"), 
            int32)
        If (Not dr.IsDBNull(dr.GetOrdinal("KodeModul"))) Then
            oSCT_Modul.KodeModul = dr("KodeModul").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("Keterangan"))) Then
            oSCT_Modul.Keterangan = dr("Keterangan").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("SPName"))) Then
            oSCT_Modul.SPName = dr("SPName").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("SPLine"))) Then
            oSCT_Modul.SPLine = dr("SPLine").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("JumlahParameter")))
            Then oSCT_Modul.JumlahParameter = CType(dr("JumlahParameter"), 
                int32)
        If (Not dr.IsDBNull(dr.GetOrdinal("Parameter"))) Then
            oSCT_Modul.Parameter = dr("Parameter").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("SMSTemplate")))
            Then oSCT_Modul.SMSTemplate = dr("SMSTemplate").
                ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("DibuatOleh")))
            Then oSCT_Modul.DibuatOleh = dr("DibuatOleh").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("WaktuDibuat")))
            Then oSCT_Modul.WaktuDibuat = CType(dr("WaktuDibuat"),
                DateTime)
        If (Not dr.IsDBNull(dr.GetOrdinal("DiubahOleh")))
            Then oSCT_Modul.DiubahOleh = dr("DiubahOleh").ToString()
        If (Not dr.IsDBNull(dr.GetOrdinal("WaktuDiubah")))
            Then oSCT_Modul.WaktuDiubah = CType(dr("WaktuDiubah"),
                DateTime)

        Return oSCT_Modul
    End Function

#End Region
    End Class
End Namespace

```

Lampiran 10

SCT_LineUserMapper.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
',
,
Copyright © IMF

',
'

#End Region

#Region "Summary"

' *****
',
,
Author : IMF IT Development Team
Purpose : SCT_LineUser Objects Mapper.
Called By :
',
'

#End Region

#Region "History"

' *****
',
,
Created On : 10/5/2018 – 1:17:51 PM
',
,
Modify By : {Name} – {Date}
– {Description Here}
',
'

******/
```

```

#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.Data
Imports System.Collections

#End Region

#Region "Custom Namespace Imports"

Imports Microsoft.Practices.EnterpriseLibrary.Data
'Imports Microsoft.Practices.EnterpriseLibrary.ExceptionHandling
'Imports Microsoft.Practices.EnterpriseLibrary.Logging

Imports IMFI.Framework.Persistence.DomainObjects.Core
Imports IMFI.Framework.Persistence.Mapper
Imports IMFI.SCT.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DataMapper
    Public Class SCT_LineUserMapper
        Inherits AbstractMapper2

#Region "Constructors/Destructors/Finalizers"

        Public Sub New()
            Db = DatabaseFactory.CreateDatabase()
        End Sub

#End Region

#Region "Private Variables"

        Private m_InsertStatement As String =
            "USP_SCT_LineUserInsert"

```

```

Private m_UpdateStatement As String = "
    USP_SCT_LineUserUpdate"
Private m_RetrieveStatement As String = "
    USP_SCT_LineUserRetrieve"
Private m_RetrieveListStatement As String = "
    USP_SCT_LineUserRetrieveList"
Private m_DeleteStatement As String = "
    USP_SCT_LineUserDelete"
Private m_RetrieveListPagingStatement As String = "
    USP_SCT_LineUserRetrievePagingList"
Private m_RetrieveWithConditionStatement As String = "
    USP_SCT_LineUserRetrieveWithCondition"
Private m_RetrieveListPagingRowCountStatement As String =
    "USP_SCT_LineUserRetrievePagingRowCount"

#End Region

#Region "Protected Methods"

Protected Overrides Function GetNewID(ByVal
    oDBCommandWrapper As System.Data.Common.DbCommand) As
Object
    Return Db.GetParameterValue(DBCommand, "@LineUserID")
End Function

Protected Overrides Function GetInsertParameter(ByVal
    oObjectTransporter As IObjectTransporter) As System.
Data.Common.DbCommand
    Dim oUserCredential As IUserCredential = CType(
        oObjectTransporter, IObjectTransporter).
        UserCredential
    Dim oSCT_LineUser As SCT_LineUser = CType(CType(
        oObjectTransporter, IObjectTransporter).
        DomainObject, SCT_LineUser)

    DBCommand = Db.GetStoredProcCommand(Me.
        m_InsertStatement)

    Db.AddInParameter(DBCommand, "@LineUserID", DbType.
        Int32, oSCT_LineUser.LineUserID)

    Db.AddInParameter(DBCommand, "@LineID", DbType.
        AnsiString, oSCT_LineUser.LineID)

```

```

Db.AddInParameter(DBCommand, "@NoHP", DbType.
    AnsiString, oSCT_LineUser.NoHP)

Db.AddInParameter(DBCommand, "@Status", DbType.Int32,
    oSCT_LineUser.Status)

Db.AddInParameter(DBCommand, "@DialogFlow", DbType.
    Int32, oSCT_LineUser.DialogFlow)

Db.AddInParameter(DBCommand, "@OTP", DbType.AnsiString
    , oSCT_LineUser.OTP)

Return DBCommand
End Function

Protected Overrides Function GetUpdateParameter(ByVal
    oObjectTransporter As IObjectTransporter) As System.
    Data.Common.DbCommand
    Dim objUserCredential As IUserCredential = CType(
        oObjectTransporter, IObjectTransporter).
        UserCredential
    Dim oSCT_LineUser As SCT_LineUser = CType(CType(
        oObjectTransporter, IObjectTransporter).
        DomainObject, SCT_LineUser)

    DBCommand = Db.GetStoredProcCommand(Me.
        m_UpdateStatement)
    Db.AddInParameter(DBCommand, "@LineUserID", DbType.
        Int32, oSCT_LineUser.LineUserID)
    Db.AddInParameter(DBCommand, "@LineID", DbType.
        AnsiString, oSCT_LineUser.LineID)
    Db.AddInParameter(DBCommand, "@NoHP", DbType.
        AnsiString, oSCT_LineUser.NoHP)
    Db.AddInParameter(DBCommand, "@Status", DbType.Int32,
        oSCT_LineUser.Status)
    Db.AddInParameter(DBCommand, "@DialogFlow", DbType.
        Int32, oSCT_LineUser.DialogFlow)
    Db.AddInParameter(DBCommand, "@OTP", DbType.AnsiString
        , oSCT_LineUser.OTP)

Return DBCommand
End Function

```

```

Protected Overrides Function GetRetrieveParameter(ByVal
    oObjectTransporter As IObjectTransporter) As System.
    Data.Common.DbCommand
    Dim oSCT_LineUser As SCT_LineUser

    If Not IsNothing(oObjectTransporter) Then
        oSCT_LineUser = CType(CType(oObjectTransporter,
            IObjectTransporter).DomainObject, SCT_LineUser)
    End If
    DBCommand = Db.GetStoredProcCommand(Me.
        m_RetrieveStatement)

    Db.AddInParameter(DBCommand, "@LineUserID", DbType.
        Int32, oSCT_LineUser.LineUserID)

    Return DBCommand
End Function

Protected Overrides Function GetRetrieveListParameter(
    ByVal oObjectTransporter As IObjectTransporter) As
    System.Data.Common.DbCommand
    Dim oSCT_LineUser As SCT_LineUser

    If Not IsNothing(oObjectTransporter) Then
        oSCT_LineUser = CType(CType(oObjectTransporter,
            IObjectTransporter).DomainObject, SCT_LineUser)
    End If
    DBCommand = Db.GetStoredProcCommand(Me.
        m_RetrieveListStatement)

    Return DBCommand
End Function

Protected Overrides Function
    GetRetrieveListPagingParameter(ByVal PageSize As
        Integer, ByVal CurrentPage As Integer, ByVal OrderBy As
        String, ByVal oObjectTransporter As IObjectTransporter
        , ByVal SearchCondition As String) As System.Data.
        Common.DbCommand
    Dim oSCT_LineUser As SCT_LineUser

    If Not IsNothing(oObjectTransporter) Then
        oSCT_LineUser = CType(CType(oObjectTransporter,
            IObjectTransporter).DomainObject, SCT_LineUser)

```

```

End If
DBCommand = Db.GetStoredProcCommand(Me.
    m_RetrieveListPagingStatement)

Db.AddInParameter(DBCommand, "@PageSize", DbType.Int32
    , PageSize)
Db.AddInParameter(DBCommand, "@CurrentPage", DbType.
    Int32, CurrentPage)
Db.AddInParameter(DBCommand, "@OrderBy", DbType.String
    , OrderBy.Split("c)(0))
Db.AddInParameter(DBCommand, "@Order", DbType.String,
    OrderBy.Split("c)(1))
Db.AddInParameter(DBCommand, "@SearchCondition",
    DbType.String, If(IsNothing(SearchCondition), "", SearchCondition))

Return DBCommand
End Function

Protected Overrides Function
GetRetrieveWithConditionParameter(ByVal SearchCondition
    As String, ByVal OrderBy As String, ByVal
    oObjectTransporter As IObjectTransporter) As System.
    Data.Common.DbCommand
Dim oSCT_LineUser As SCT_LineUser

If Not IsNothing(oObjectTransporter) Then
    oSCT_LineUser = CType(CType(oObjectTransporter,
        IObjectTransporter).DomainObject, SCT_LineUser)
End If
DBCommand = Db.GetStoredProcCommand(Me.
    m_RetrieveWithConditionStatement)

Db.AddInParameter(DBCommand, "@OrderBy", DbType.String
    , OrderBy)
Db.AddInParameter(DBCommand, "@SearchCondition",
    DbType.String, If(IsNothing(SearchCondition), "", SearchCondition))

Return DBCommand
End Function

Protected Overrides Function
GetRetrieveListPagingRowCountParameter(ByVal
    oObjectTransporter As IObjectTransporter, ByVal

```

```

SearchCondition As String) As System.Data.Common.
DbCommand
Dim oSCT_LineUser As SCT_LineUser

If Not IsNothing(oObjectTransporter) Then
    oSCT_LineUser = CType(CType(oObjectTransporter,
        IObjectTransporter).DomainObject, SCT_LineUser)
End If
DBCommand = Db.GetStoredProcCommand(Me.
    m_RetrieveListPagingRowCountStatement)

Db.AddInParameter(DBCommand, "@SearchCondition",
    DbType.String, If(IsNothing(SearchCondition), "", SearchCondition))

Return DBCommand
End Function

Protected Overrides Function GetDeleteParameter(
    oObjectTransporter As IObjectTransporter) As System.
Data.Common.DbCommand
Dim oUserCredential As IUserCredential = CType(
    oObjectTransporter, IObjectTransporter).
    UserCredential
Dim oSCT_LineUser As SCT_LineUser = CType(CType(
    oObjectTransporter, IObjectTransporter).
    DomainObject, SCT_LineUser)
DBCommand = Db.GetStoredProcCommand(Me.
    m_DeleteStatement)

Db.AddInParameter(DBCommand, "@LineUserID", DbType.
    Int32, oSCT_LineUser.LineUserID)

Return DBCommand
End Function

Protected Overrides Function DoRetrieve(ByVal dr As
    IDataReader) As Object
Dim oSCT_LineUser As SCT_LineUser = Nothing

While (dr.Read())
    oSCT_LineUser = Me.CreateObject(dr)
End While

```

```

        Return oSCT_LineUser
    End Function

Protected Overrides Function DoRetrieveList(ByVal dr As
    IDataReader) As ArrayList
    Dim oSCT_LineUserList As ArrayList = New ArrayList()

    While (dr.Read())

        Dim oSCT_LineUser As SCT_LineUser = Me.
            CreateObject(dr)
        oSCT_LineUserList.Add(oSCT_LineUser)
    End While

    Return oSCT_LineUserList
End Function

Protected Overrides Function DoRetrieveWithCondition(ByVal
    dr As IDataReader) As ArrayList
    Dim oSCT_LineUserList As ArrayList = New ArrayList()

    While (dr.Read())

        Dim oSCT_LineUser As SCT_LineUser = Me.
            CreateObject(dr)
        oSCT_LineUserList.Add(oSCT_LineUser)
    End While

    Return oSCT_LineUserList
End Function

#End Region

#Region "Private Methods"

Private Function CreateObject(ByVal dr As IDataReader) As
    SCT_LineUser
    Dim oSCT_LineUser As SCT_LineUser = New SCT_LineUser()

    oSCT_LineUser.LineUserID = CType(dr("LineUserID"), int32)
    If (Not dr.IsDBNull(dr.GetOrdinal("LineID"))) Then
        oSCT_LineUser.LineID = dr("LineID").ToString()
    If (Not dr.IsDBNull(dr.GetOrdinal("NoHP"))) Then
        oSCT_LineUser.NoHP = dr("NoHP").ToString()

```

```

If (Not dr.IsDBNull(dr.GetOrdinal("Status"))) Then
    oSCT_LineUser.Status = CType(dr("Status"), int32)
If (Not dr.IsDBNull(dr.GetOrdinal("DialogFlow"))) Then
    oSCT_LineUser.DialogFlow = CType(dr("DialogFlow"), 
    Int32)
If (Not dr.IsDBNull(dr.GetOrdinal("OTP"))) Then
    oSCT_LineUser.OTP = dr("OTP").ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("DibuatOleh"))) Then
    oSCT_LineUser.DibuatOleh = dr("DibuatOleh").
    ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("WaktuDibuat")))
    Then oSCT_LineUser.WaktuDibuat = CType(dr("WaktuDibuat"), DateTime)
If (Not dr.IsDBNull(dr.GetOrdinal("DiubahOleh")))
    Then oSCT_LineUser.DiubahOleh = dr("DiubahOleh").
    ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("WaktuDiubah")))
    Then oSCT_LineUser.WaktuDiubah = CType(dr("WaktuDiubah"), DateTime)

Return oSCT_LineUser
End Function

#End Region
End Class
End Namespace

```

Lampiran 11

SCT_InboxLineMapper.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
',
,
Copyright © IMF
,

' *****

#End Region

#Region "Summary"

' *****
',
,
Author : IMF IT Development Team
,Purpose : SCT_InboxLine Objects Mapper.
,Called By :
,

' *****

#End Region

#Region "History"

' *****
',
,
Created On : 10/8/2018 – 10:39:49 AM
,
Modify By : {Name} – {Date}
– {Description Here}
,
' *****/

```

```

#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.Data
Imports System.Collections

#End Region

#Region "Custom Namespace Imports"

Imports Microsoft.Practices.EnterpriseLibrary.Data
'Imports Microsoft.Practices.EnterpriseLibrary.ExceptionHandling
'Imports Microsoft.Practices.EnterpriseLibrary.Logging

Imports IMFI.Framework.Persistence.DomainObjects.Core
Imports IMFI.Framework.Persistence.Mapper
Imports IMFI.SCT.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.DataMapper
    Public Class SCT_InboxLineMapper
        Inherits AbstractMapper2

#Region "Constructors/Destructors/Finalizers"

        Public Sub New()
            Db = DatabaseFactory.CreateDatabase()
        End Sub

#End Region

#Region "Private Variables"

        Private m_InsertStatement As String =
            "USP_SCT_InboxLineInsert"

```

```

Private m_UpdateStatement As String = "
    USP_SCT_InboxLineUpdate"
Private m_RetrieveStatement As String = "
    USP_SCT_InboxLineRetrieve"
Private m_RetrieveListStatement As String = "
    USP_SCT_InboxLineRetrieveList"
Private m_DeleteStatement As String = "
    USP_SCT_InboxLineDelete"
Private m_RetrieveListPagingStatement As String = "
    USP_SCT_InboxLineRetrievePagingList"
Private m_RetrieveWithConditionStatement As String = "
    USP_SCT_InboxLineRetrieveWithCondition"
Private m_RetrieveListPagingRowCountStatement As String =
    "USP_SCT_InboxLineRetrievePagingRowCount"

#End Region

#Region "Protected Methods"

Protected Overrides Function GetNewID(ByVal
    oDBCommandWrapper As System.Data.Common.DbCommand) As
    Object
    Return Db.GetParameterValue(DBCommand, "@InboxLineID")
End Function

Protected Overrides Function GetInsertParameter(ByVal
    oObjectTransporter As IObjectTransporter) As System.
    Data.Common.DbCommand
    Dim oUserCredential As IUserCredential = CType(
        oObjectTransporter, IObjectTransporter).
        UserCredential
    Dim oSCT_InboxLine As SCT_InboxLine = CType(CType(
        oObjectTransporter, IObjectTransporter).
        DomainObject, SCT_InboxLine)

    DBCommand = Db.GetStoredProcCommand(Me.
        m_InsertStatement)

    Db.AddInParameter(DBCommand, "@InboxLineID", DbType.
        Int32, oSCT_InboxLine.InboxLineID)

    Db.AddInParameter(DBCommand, "@NoHP", DbType.
        AnsiString, oSCT_InboxLine.NoHP)

```

```

Db.AddInParameter(DBCommand, "@Register", DbType.
Boolean, oSCT_InboxLine.Register)

Db.AddInParameter(DBCommand, "@SMSMessage", DbType.
AnsiString, oSCT_InboxLine.SMSMessage)

Db.AddInParameter(DBCommand, "@WaktuSMS", DbType.
DateTime, IIf(oSCT_InboxLine.WaktuSMS > #1/1/1753#, 
oSCT_InboxLine.WaktuSMS, System.DBNull.Value))

Db.AddInParameter(DBCommand, "@Proses", DbType.Boolean
, oSCT_InboxLine.Proses)

Db.AddInParameter(DBCommand, "@ReplyMessage", DbType.
AnsiString, oSCT_InboxLine.ReplyMessage)

Db.AddInParameter(DBCommand, "@WaktuReply", DbType.
DateTime, IIf(oSCT_InboxLine.WaktuReply >
#1/1/1753#, oSCT_InboxLine.WaktuReply, System.
DBNull.Value))

Db.AddInParameter(DBCommand, "@ApplicationFlag",
DbType.AnsiString, oSCT_InboxLine.ApplicationFlag)

Return DBCommand
End Function

Protected Overrides Function GetUpdateParameter(ByVal
oObjectTransporter As IObjectTransporter) As System.
Data.Common.DbCommand
Dim objUserCredential As IUserCredential = CType(
oObjectTransporter, IObjectTransporter).
UserCredential
Dim oSCT_InboxLine As SCT_InboxLine = CType(CType(
oObjectTransporter, IObjectTransporter).
DomainObject, SCT_InboxLine)

DBCommand = Db.GetStoredProcCommand(Me.
m_UpdateStatement)
Db.AddInParameter(DBCommand, "@InboxLineID", DbType.
Int32, oSCT_InboxLine.InboxLineID)
Db.AddInParameter(DBCommand, "@NoHP", DbType.
AnsiString, oSCT_InboxLine.NoHP)
Db.AddInParameter(DBCommand, "@Register", DbType.

```

```

        Boolean , oSCT_InboxLine . Register )
Db . AddInParameter (DBCommand , "@SMSMessage" , DbType .
        AnsiString , oSCT_InboxLine . SMSMessage )
Db . AddInParameter (DBCommand , "@WaktuSMS" , DbType .
        DateTime , IIf (oSCT_InboxLine . WaktuSMS > #1/1/1753# ,
            oSCT_InboxLine . WaktuSMS , System . DBNull . Value ))
Db . AddInParameter (DBCommand , "@Proses" , DbType . Boolean
        , oSCT_InboxLine . Proses )
Db . AddInParameter (DBCommand , "@ReplyMessage" , DbType .
        AnsiString , oSCT_InboxLine . ReplyMessage )
Db . AddInParameter (DBCommand , "@WaktuReply" , DbType .
        DateTime , IIf (oSCT_InboxLine . WaktuReply >
            #1/1/1753# , oSCT_InboxLine . WaktuReply , System .
            DBNull . Value ))
Db . AddInParameter (DBCommand , "@ApplicationFlag" ,
        DbType . AnsiString , oSCT_InboxLine . ApplicationFlag )

Return DBCommand
End Function

Protected Overrides Function GetRetrieveParameter ( ByVal
    oObjectTransporter As IObjectTransporter ) As System .
    Data . Common . DbCommand
Dim oSCT_InboxLine As SCT_InboxLine

If Not IsNothing (oObjectTransporter) Then
    oSCT_InboxLine = CType (CType (oObjectTransporter ,
        IObjectTransporter) . DomainObject , SCT_InboxLine
    )
End If
DBCommand = Db . GetStoredProcedure (Me .
    m_RetrieveStatement)

Db . AddInParameter (DBCommand , "@InboxLineID" , DbType .
    Int32 , oSCT_InboxLine . InboxLineID)

Return DBCommand
End Function

Protected Overrides Function GetRetrieveListParameter (
    ByVal oObjectTransporter As IObjectTransporter ) As
System . Data . Common . DbCommand
Dim oSCT_InboxLine As SCT_InboxLine

```

```

If Not IsNothing(oObjectTransporter) Then
    oSCT_InboxLine = CType(CType(oObjectTransporter,
        IObjectTransporter).DomainObject, SCT_InboxLine
    )
End If
DBCommand = Db.GetStoredProcCommand(Me.
    m_RetrieveListStatement)

Return DBCommand
End Function

Protected Overrides Function
    GetRetrieveListPagingParameter(ByVal PageSize As
        Integer, ByVal CurrentPage As Integer, ByVal OrderBy As
        String, ByVal oObjectTransporter As IObjectTransporter
        , ByVal SearchCondition As String) As System.Data.
    Common.DbCommand
    Dim oSCT_InboxLine As SCT_InboxLine

    If Not IsNothing(oObjectTransporter) Then
        oSCT_InboxLine = CType(CType(oObjectTransporter,
            IObjectTransporter).DomainObject, SCT_InboxLine
        )
    End If
    DBCommand = Db.GetStoredProcCommand(Me.
        m_RetrieveListPagingStatement)

    Db.AddInParameter(DBCommand, "@PageSize", DbType.Int32
        , PageSize)
    Db.AddInParameter(DBCommand, "@CurrentPage", DbType.
        Int32, CurrentPage)
    Db.AddInParameter(DBCommand, "@OrderBy", DbType.String
        , OrderBy.Split(" "c)(0))
    Db.AddInParameter(DBCommand, "@Order", DbType.String,
        OrderBy.Split(" "c)(1))
    Db.AddInParameter(DBCommand, "@SearchCondition",
        DbType.String, If(IsNothing(SearchCondition), "", SearchCondition))

    Return DBCommand
End Function

Protected Overrides Function
    GetRetrieveWithConditionParameter(ByVal SearchCondition

```

```

        As String , ByVal OrderBy As String , ByVal
        oObjectTransporter As IObjectTransporter) As System.
        Data.Common.DbCommand
        Dim oSCT_InboxLine As SCT_InboxLine

        If Not IsNothing(oObjectTransporter) Then
            oSCT_InboxLine = CType(CType(oObjectTransporter ,
                IObjectTransporter).DomainObject , SCT_InboxLine
            )
        End If
        DBCommand = Db.GetStoredProcCommand(Me.
            m_RetrieveWithConditionStatement)

        Db.AddInParameter(DBCommand, "@OrderBy" , DbType.String
            , OrderBy)
        Db.AddInParameter(DBCommand, "@SearchCondition" ,
            DbType.String , IIf(IsNothing(SearchCondition) , "" ,
            SearchCondition))

        Return DBCommand
    End Function

    Protected Overrides Function
        GetRetrieveListPagingRowCountParameter(ByVal
            oObjectTransporter As IObjectTransporter , ByVal
            SearchCondition As String) As System.Data.Common.
        DbCommand
        Dim oSCT_InboxLine As SCT_InboxLine

        If Not IsNothing(oObjectTransporter) Then
            oSCT_InboxLine = CType(CType(oObjectTransporter ,
                IObjectTransporter).DomainObject , SCT_InboxLine
            )
        End If
        DBCommand = Db.GetStoredProcCommand(Me.
            m_RetrieveListPagingRowCountStatement)

        Db.AddInParameter(DBCommand, "@SearchCondition" ,
            DbType.String , IIf(IsNothing(SearchCondition) , "" ,
            SearchCondition))

        Return DBCommand
    End Function

    Protected Overrides Function GetDeleteParameter(

```

```

oObjectTransporter As IObjectTransporter) As System.
Data.Common.DbCommand
Dim oUserCredential As IUserCredential = CType(
    oObjectTransporter, IObjectTransporter).
    UserCredential
Dim oSCT_InboxLine As SCT_InboxLine = CType(CType(
    oObjectTransporter, IObjectTransporter).
    DomainObject, SCT_InboxLine)
DBCommand = Db.GetStoredProcCommand(Me.
    m_DeleteStatement)

Db.AddInParameter(DBCommand, "@InboxLineID", DbType.
    Int32, oSCT_InboxLine.InboxLineID)

Return DBCommand
End Function

Protected Overrides Function DoRetrieve(ByVal dr As
    IDataReader) As Object
Dim oSCT_InboxLine As SCT_InboxLine = Nothing

While (dr.Read())
    oSCT_InboxLine = Me.CreateObject(dr)
End While

Return oSCT_InboxLine
End Function

Protected Overrides Function DoRetrieveList(ByVal dr As
    IDataReader) As ArrayList
Dim oSCT_InboxLineList As ArrayList = New ArrayList()

While (dr.Read())
    Dim oSCT_InboxLine As SCT_InboxLine = Me.
        CreateObject(dr)
    oSCT_InboxLineList.Add(oSCT_InboxLine)
End While

Return oSCT_InboxLineList
End Function

Protected Overrides Function DoRetrieveWithCondition(ByVal
    dr As IDataReader) As ArrayList

```

```

Dim oSCT_InboxLineList As ArrayList = New ArrayList()

While (dr.Read())

    Dim oSCT_InboxLine As SCT_InboxLine = Me.
        CreateObject(dr)
    oSCT_InboxLineList.Add(oSCT_InboxLine)
End While

Return oSCT_InboxLineList
End Function

#End Region

#Region "Private Methods"

Private Function CreateObject(ByVal dr As IDataReader) As
SCT_InboxLine
    Dim oSCT_InboxLine As SCT_InboxLine = New
SCT_InboxLine()

oSCT_InboxLine.InboxLineID = CType(dr("InboxLineID"), 
int32)
If (Not dr.IsDBNull(dr.GetOrdinal("NoHP"))) Then
    oSCT_InboxLine.NoHP = dr("NoHP").ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("Register"))) Then
    oSCT_InboxLine.Register = CType(dr("Register"), 
Boolean)
If (Not dr.IsDBNull(dr.GetOrdinal("SMSMessage"))) Then
    oSCT_InboxLine.SMSMessage = dr("SMSMessage").
ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("WaktuSMS"))) Then
    oSCT_InboxLine.WaktuSMS = CType(dr("WaktuSMS"), 
DateTime)
If (Not dr.IsDBNull(dr.GetOrdinal("Proses"))) Then
    oSCT_InboxLine.Proses = CType(dr("Proses"), Boolean)
)
If (Not dr.IsDBNull(dr.GetOrdinal("ReplyMessage")))
Then oSCT_InboxLine.ReplyMessage = dr("ReplyMessage").
ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("WaktuReply"))) Then
    oSCT_InboxLine.WaktuReply = CType(dr("WaktuReply"),
DateTime)
If (Not dr.IsDBNull(dr.GetOrdinal("ApplicationFlag")))
Then oSCT_InboxLine.ApplicationFlag = dr(""

```

```
        ApplicationFlag") . ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("DibuatOleh"))) Then
    oSCT_InboxLine.DibuatOleh = dr("DibuatOleh") .
        ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("WaktuDibuat")))
    Then oSCT_InboxLine.WaktuDibuat = CType(dr("
        WaktuDibuat"), DateTime)
If (Not dr.IsDBNull(dr.GetOrdinal("DiubahOleh")))
    Then oSCT_InboxLine.DiubahOleh = dr("DiubahOleh") .
        ToString()
If (Not dr.IsDBNull(dr.GetOrdinal("WaktuDiubah")))
    Then oSCT_InboxLine.WaktuDiubah = CType(dr("
        WaktuDiubah"), DateTime)

    Return oSCT_InboxLine
End Function

#End Region
End Class
End Namespace
```

Lampiran 12

SCT_ModulBusinessFacade.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
'*          Copyright © IMFIMF
'*          *
'*          *
' *****

#End Region

#Region "Summary"

' *****
'*          Author      : IMFIMF Development Team
'*          Purpose     : SCT_Modul Business Facade.
*
'*          Called By   :
'*          *
' *****

#End Region

#Region "History"

' *****
'*          Created On  : 26/4/2017 – 3:16:30 PM
'*          *
'*          Modify By   : {Name} – {Date}
'*          – {Description Here}
'*          *
' *****
```

```

' *
'* ****
* /



#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.Collections

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistance.Mapper
Imports IMFI.Framework.Persistance.DomainObjects.Core
Imports IMFI.Framework.Persistance.BusinessFacade
Imports IMFI.SCT.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.BusinessFacade

    Public Class SCT_ModulBusinessFacade
        #Region "Private Variables"

            Private SCT_ModulMapper As IMapper
            Private _Transaction As IDbTransaction

        #End Region

        #Region "Constructors/Destructors/Finalizers"

            Public Sub New()
                SCT_ModulMapper = MapperFactory.GetInstance().

```

```

        GetMapper(GetType(SCT_Modul).ToString())
    End Sub
#End Region

#Region "Public Methods"

    Public Function Retrieve(ByVal oObjectTransporter As
        IObjectTransporter) As SCT_Modul
        Return CType(SCT_ModulMapper.Retrieve(
            oObjectTransporter), SCT_Modul)
    End Function

    Public Function RetrieveListPaging(ByVal PageSize As Int32
        , ByVal CurrentPage As Int32, ByVal OrderBy As String ,
        ByVal oObjectTransporter As IObjectTransporter , ByVal
        SearchCondition As String , ByRef RowCount As Int32) As
        ArrayList
        Return SCT_ModulMapper.RetrieveListPaging(PageSize ,
            CurrentPage , OrderBy , oObjectTransporter ,
            SearchCondition , RowCount)
    End Function

    Public Function RetrieveList(ByVal oObjectTransporter As
        IObjectTransporter) As ArrayList
        Return SCT_ModulMapper.RetrieveList(oObjectTransporter
            )
    End Function

    Public Function Create(ByVal oObjectTransporter As
        IObjectTransporter) As Object
        Return SCT_ModulMapper.Insert(oObjectTransporter)
    End Function

    Public Function Delete(ByVal oObjectTransporter As
        IObjectTransporter) As Int32
        Return SCT_ModulMapper.Delete(oObjectTransporter)
    End Function

    Public Function Update(ByVal oObjectTransporter As
        IObjectTransporter) As Int32
        Return SCT_ModulMapper.Update(oObjectTransporter)
    End Function

    Public Sub UseTransaction(ByVal oTransaction As

```

```
    IDbTransaction)
    SCT_ModulMapper.UseTransaction(oTransaction)
    _Transaction = oTransaction
End Sub

#End Region
End Class
End Namespace
```

Lampiran 13

SCT_LineUserBusinessFacade.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
',
'          Copyright © IMF
',
' *****

#End Region

#Region "Summary"

' *****
',
'      Author      : IMF IT Development Team
'      Purpose     : SCT_LineUser Business Facade.
'      Called By   :
',
' *****

#End Region

#Region "History"

' *****
',
'      Created On  : 10/5/2018 – 1:18:08 PM
',
'      Modify By   : {Name} – {Date}
'                  – {Description Here}
'


```

```

' ****
#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.Collections

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistance.Mapper
Imports IMFI.Framework.Persistance.DomainObjects.Core
Imports IMFI.Framework.Persistance.BusinessFacade
Imports IMFI.SCT.DomainObjects
Imports IMFI.SCT.Common
Imports System.Text.RegularExpressions

#End Region

#End Region

Namespace IMFI.SCT.BusinessFacade

    Public Class SCT_LineUserBusinessFacade
        #Region "Private Variables"

            Private SCT_LineUserMapper As IMapper2
            Private _Transaction As IDbTransaction

       #End Region

        #Region "Constructors/Destructors/Finalizers"

            Public Sub New()
                SCT_LineUserMapper = MapperFactory2.GetInstance() .
                    GetMapper(GetType(SCT_LineUser).ToString())
            End Sub

       #End Region
    End Class

```

```

#Region "Public Methods"

    Public Function Create(ByVal oObjectTransporter As
        IObjectTransporter) As Object
        Return SCT_LineUserMapper.Insert(oObjectTransporter)
    End Function

    Public Function Retrieve(ByVal oObjectTransporter As
        IObjectTransporter) As SCT_LineUser
        Return CType(SCT_LineUserMapper.Retrieve(
            oObjectTransporter), SCT_LineUser)
    End Function

    Public Function RetrieveListPaging(ByVal PageSize As Int32
        , ByVal CurrentPage As Int32, ByVal OrderBy As String,
        ByVal oObjectTransporter As IObjectTransporter, ByVal
        SearchCondition As String, ByRef RowCount As Int32) As
        ArrayList
        Return SCT_LineUserMapper.RetrieveListPaging(PageSize,
            CurrentPage, OrderBy, oObjectTransporter,
            SearchCondition, RowCount)
    End Function

    Public Function RetrieveWithCondition(ByVal
        SearchCondition As String, ByVal OrderBy As String,
        ByVal oObjectTransporter As IObjectTransporter) As
        ArrayList
        Return SCT_LineUserMapper.RetrieveWithCondition(
            SearchCondition, OrderBy, oObjectTransporter)
    End Function

    Public Function RetrieveList(ByVal oObjectTransporter As
        IObjectTransporter) As ArrayList
        Return SCT_LineUserMapper.RetrieveList(
            oObjectTransporter)
    End Function

    Public Function Update(ByVal oObjectTransporter As
        IObjectTransporter) As Int32
        Return SCT_LineUserMapper.Update(oObjectTransporter)
    End Function

'public Function Delete(ByVal oObjectTransporter as
    IObjectTransporter) as Int32

```

```

'    return SCT_LineUserMapper.Delete(oObjectTransporter)
'End Function

Public Function Delete(ByVal oObjectTransporter As
IObjectTransporter) As Result
Dim oResult As Result

Try
    oResult = New Result(SCT_LineUserMapper.Delete(
        oObjectTransporter), "")
Catch ex As Exception
    If ex.GetType().Name = "ConstraintException" Then
        oResult = New Result(-1, "Data sudah digunakan
        .")
    Else
        Throw ex
    End If
End Try

Return oResult
End Function

Public Sub UseTransaction(ByVal oTransaction As
IDbTransaction)
    SCT_LineUserMapper.UseTransaction(oTransaction)
    _Transaction = oTransaction
End Sub

#End Region
#Region "Custom Method"
    Public Function CekLineUser(ByRef LineID As String) As
ArrayList

    Dim oSearchCondition As String = "LineID = '" + LineID
    + "'"
    Dim arLineUser As ArrayList = Me.RetrieveWithCondition
    (oSearchCondition, "LineUserID DESC", (New
    ObjectTransporter(New SCT_LineUser, Nothing)))

    Return arLineUser
End Function

Public Function CreateLineUser(ByRef oLine As SCT_LineUser
) As SCT_LineUser

```

```

Dim oLineInsert = Me.Create(New ObjectTransporter(
    oLine, (New UserIdentification).GetUserCredential))

    Return oLineInsert
End Function

Public Function PendaftaranLineUser(ByRef oRowLine As
    SCT_LineUser, ByVal incomingmessage As String) As
String
Dim Message = ""

If oRowLine.DialogFlow = 0 Then
    'DialogFlow = 0 -> user belum memasukkan nomor HP
    ke SCT_LineUser

    Message = "Silahkan masukkan nomor telepon anda"

    'update dialogflow
    oRowLine.DialogFlow = 1
    Dim LineUserUpdate As Integer = (New
        SCT_LineUserBusinessFacade).Update(New
        ObjectTransporter(oRowLine, (New
        UserIdentification).GetUserCredential))

ElseIf oRowLine.DialogFlow = 1 Then
    'DialogFlow = 1 -> User memasukkan Nomor HP

    Dim nomorHP = incomingmessage
    Dim tempNomorHP = nomorHP
    'Mengecek apakah format nomor hp sudah benar dan
    'Mengubah format 0xxxxxx menjadi +62xxxxx

    If nomorHP.Substring(0, 1) = "+" Then
        Else
            nomorHP = nomorHP.Substring(1, nomorHP.Length
                - 1)
            nomorHP = "+62" + nomorHP

    End If

    If Regex.IsMatch(nomorHP.Substring(3, nomorHP.
        Length - 3), "^[0-9 ]+$") Then
        'Jika format benar
        Message = " Apakah nomor " + nomorHP + " sudah
        benar? [ya/tidak]"
    End If
End If

```

```

'Update nomor HP
oRowLine.NoHP = nomorHP
oRowLine.DialogFlow = 2
Dim LineUserUpdate As Integer = (New
    SCT_LineUserBusinessFacade).Update(New
    ObjectTransporter(oRowLine, (New
        UserIdentification).GetUserCredential))

Else
    Message = "Format nomor HP salah. Silahkan
        masukkan nomor hp kembali"

End If

ElseIf oRowLine.DialogFlow = 2 Then
    'DialogFlow = 2 -> Konfirmasi Nomor HP

    Dim konfirmasi = incomingmessage

    If konfirmasi.ToUpper = "YA" Then

        'Cek apakah nomor HP sudah terdaftar
        Dim nomorHP = oRowLine.NoHP

        If (nomorHP.Substring(0, 1)) = "0" Then
            nomorHP = nomorHP.Substring(1, oRowLine.
                NoHP.Length)
            nomorHP = "+62" & nomorHP

    End If

    Dim condition = "NoHP = '" + nomorHP + "'"
    Dim oRowRegistrasiHdr As ArrayList = New
        SCT_Registrasi_HdrBusinessFacade().
        RetrieveListPaging(-1, 0, "
            NoDataRegistrasiHdr ASC", (New
            ObjectTransporter(New SCT_Registrasi_Hdr, (
            New UserIdentification).GetUserCredential))
            , condition, 10)

    If (oRowRegistrasiHdr.Count > 0) Then
        Message = "Silahkan masukkan kode OTP yang
            dikirimkan melalui SMS"

```

```

        oRowLine.OTP = GenerateOTP()
        oRowLine.DialogFlow = 3
        Dim LineUserUpdate As Integer = (New
            SCT_LineUserBusinessFacade).Update(New
            ObjectTransporter(oRowLine, (New
            UserIdentification). GetUserCredential))

    Else
        Message = "Nomor HP tidak terdaftar.
                    Silahkan masukkan nomor HP lain"

        oRowLine.DialogFlow = 1
        Dim LineUserUpdate As Integer = (New
            SCT_LineUserBusinessFacade).Update(New
            ObjectTransporter(oRowLine, (New
            UserIdentification). GetUserCredential))

    End If

ElseIf konfirmasi.ToUpper = "TIDAK" Then
    Message = "Silahkan masukkan nomor HP anda"

    oRowLine.NoHP = ""
    oRowLine.DialogFlow = 1
    Dim LineUserUpdate As Integer = (New
        SCT_LineUserBusinessFacade).Update(New
        ObjectTransporter(oRowLine, (New
        UserIdentification). GetUserCredential))

Else
    Message = "Apakah nomor " + oRowLine.NoHP +
        " tersebut sudah benar? [ya/tidak]"

End If

ElseIf oRowLine.DialogFlow = 3 Then
    'DialogFlow = 3 -> User input kode OTP

    Dim kodeOTP = incomingmessage
    Message = "Kode OTP salah"

    If (kodeOTP = oRowLine.OTP) Then
        Message = "Pendaftaran Berhasil"

        oRowLine.Status = 1

```

```

        oRowLine.DialogFlow = 0
        Dim LineUserUpdate As Integer = (New
            SCT_LineUserBusinessFacade).Update(New
            ObjectTransporter(oRowLine, (New
            UserIdentification).GetUserCredential))

    Else
        Message = "Kode OTP salah"

    End If
End If
Return Message
End Function

Public Function GetInfo(ByRef LineID As String, ByVal
    incomingmessage As String) As String

    Dim message = ""
    Dim oResult As Object

    'mengambil InboxLineID terakhir di tabel SCT_InboxLine
    Dim oInboxLine As ArrayList = New
        SCT_InboxLineBusinessFacade().RetrieveWithCondition
        ("", "InboxLineID desc", New ObjectTransporter(New
        SCT_InboxLine, (New UserIdentification).
        GetUserCredential))
    Dim nextID = (CType(oInboxLine(0), SCT_InboxLine).
        InboxLineID + 1)

    'mengambil data dari tabel SCT_LineUser
    Dim oLineNoHP As ArrayList = Me.RetrieveWithCondition
        ("LineID = '" + LineID + "'", "LineID desc", New
        ObjectTransporter(New SCT_LineUser, (New
        UserIdentification).GetUserCredential))

    Dim dateToday As DateTime = DateTime.Today

    Dim oNewInboxLine = New SCT_InboxLine

    'memasukkan data ke tabel SCT_InboxLineuser
    oNewInboxLine.InboxLineID = nextID
    oNewInboxLine.SMSMessage = incomingmessage
    oNewInboxLine.NoHP = CType(oLineNoHP(0), SCT_LineUser)
        .NoHP
    oNewInboxLine.WaktuSMS = dateToday

```

```

If incomingmessage.Length > 3 Then
    oNewInboxLine.ApplicationFlag = incomingmessage.
        Substring(0, 3).ToUpper
Else
    oNewInboxLine.ApplicationFlag = "Invalid"
End If
oResult = New SCT_InboxLineBusinessFacade().Create(New
    ObjectTransporter(oNewInboxLine, New
        UserIdentification().GetUserCredential))

'mengambil balasan dari sistem
Dim oReplyInboxLine As SCT_InboxLine = New
    SCT_InboxLineBusinessFacade().Retrieve(New
        ObjectTransporter(New SCT_InboxLine(nextID), (New
            UserIdentification).GetUserCredential))

message = oReplyInboxLine.ReplyMessage

Return message
End Function

Protected Function GenerateOTP() As String
    Dim alphabets As String = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    Dim small_alphabets As String = "
        abcdefghijklmnopqrstuvwxyz"
    Dim numbers As String = "1234567890"

    Dim characters As String = numbers

    Dim length As Integer = Integer.Parse(6)
    Dim otp As String = String.Empty
    For i As Integer = 0 To length - 1
        Dim character As String = String.Empty
        Do
            Dim index As Integer = New Random().Next(0,
                characters.Length)
            character = characters.ToCharArray()(index).
                ToString()
        Loop While otp.IndexOf(character) <> -1
        otp += character
    Next
    Return otp
End Function

#End Region

```

End Class

End Namespace

Lampiran 14

SCT_InboxLineBusinessFacade.vb

```
#Region "Header Info"

#Region "Code Disclaimer"

' *****
',
'          Copyright © IMF
',
' *****

#End Region

#Region "Summary"

' *****
',
'      Author      : IMF IT Development Team
'      Purpose     : SCT_InboxLine Business Facade.
'      Called By   :
',
' *****

#End Region

#Region "History"

' *****
',
'      Created On  : 10/8/2018 – 10:40:14 AM
',
'      Modify By   : {Name} – {Date}
'                  – {Description Here}
'


```

```

' ****
#End Region

#End Region

#Region "Namespace Imports"

#Region ".NET Base Class Namespace Imports"

Imports System
Imports System.Collections

#End Region

#Region "Custom Namespace Imports"

Imports IMFI.Framework.Persistance.Mapper
Imports IMFI.Framework.Persistance.DomainObjects.Core
Imports IMFI.Framework.Persistance.BusinessFacade
Imports IMFI.SCT.DomainObjects

#End Region

#End Region

Namespace IMFI.SCT.BusinessFacade

    Public Class SCT_InboxLineBusinessFacade
        #Region "Private Variables"

            Private SCT_InboxLineMapper As IMapper2
            Private _Transaction As IDbTransaction

       #End Region

        #Region "Constructors/Destructors/Finalizers"

            Public Sub New()
                SCT_InboxLineMapper = MapperFactory2.GetInstance() .
                    GetMapper(GetType(SCT_InboxLine).ToString())
            End Sub
       #End Region

        #Region "Public Methods"

```

```

Public Function Create(ByVal oObjectTransporter As
    IObjectTransporter) As Object
    Return SCT_InboxLineMapper.Insert(oObjectTransporter)
End Function

Public Function Retrieve(ByVal oObjectTransporter As
    IObjectTransporter) As SCT_InboxLine
    Return CType(SCT_InboxLineMapper.Retrieve(
        oObjectTransporter), SCT_InboxLine)
End Function

Public Function RetrieveListPaging(ByVal PageSize As Int32
    , ByVal CurrentPage As Int32, ByVal OrderBy As String ,
    ByVal oObjectTransporter As IObjectTransporter , ByVal
    SearchCondition As String , ByRef RowCount As Int32) As
    ArrayList
    Return SCT_InboxLineMapper.RetrieveListPaging(PageSize
        , CurrentPage , OrderBy , oObjectTransporter ,
        SearchCondition , RowCount)
End Function

Public Function RetrieveWithCondition(ByVal
    SearchCondition As String , ByVal OrderBy As String ,
    ByVal oObjectTransporter As IObjectTransporter) As
    ArrayList
    Return SCT_InboxLineMapper.RetrieveWithCondition(
        SearchCondition , OrderBy , oObjectTransporter)
End Function

Public Function RetrieveList(ByVal oObjectTransporter As
    IObjectTransporter) As ArrayList
    Return SCT_InboxLineMapper.RetrieveList(
        oObjectTransporter)
End Function

Public Function Update(ByVal oObjectTransporter As
    IObjectTransporter) As Int32
    Return SCT_InboxLineMapper.Update(oObjectTransporter)
End Function

' public Function Delete(ByVal oObjectTransporter as
    IObjectTransporter) as Int32
'     return SCT_InboxLineMapper.Delete(oObjectTransporter)
'End Function

```

```

Public Function Delete(ByVal oObjectTransporter As
    IObjectTransporter) As Result
    Dim oResult As Result

    Try
        oResult = New Result(SCT_InboxLineMapper.Delete(
            oObjectTransporter), "")
    Catch ex As Exception
        If ex.GetType().Name = "ConstraintException" Then
            oResult = New Result(-1, "Data sudah digunakan
                .")
        Else
            Throw ex
        End If
    End Try

    Return oResult
End Function

Public Sub UseTransaction(ByVal oTransaction As
    IDbTransaction)
    SCT_InboxLineMapper.UseTransaction(oTransaction)
    _Transaction = oTransaction
End Sub

#End Region

End Class

End Namespace

```

Lampiran 15

Form UAT Page 1

No	Test Case	Checkmark	Remark
Webapps SMS Center (Penambahan Status Registrasi Line dan Status Inbox dari Line / SMS)			
1	MASTER - USER REGISTRATION	-	-
	A. Halaman Browse User Registration	-	-
	● Status Registrasi Akun Line	✓	✓
2	MASTER - User Registration Input	-	-
	A. Halaman Browse Module	-	-
	● Status Registrasi Akun Line	✓	✓
3	MAILBOX - INBOX	-	-
	A. Halaman Browse Inbox	-	-
	● Status sumber pesan masuk	✓	✓
Line Chatbot Collection Center			
1	Add User	-	-
	● Menambahkan Friend IMFI-Coll Center di Line User (@138jkzap)	✓	✓
2	Pendaftaran User	-	-
	A. User Terdaftar di SMS Center	-	-
	● Melakukan pendaftaran dengan ketik Daftar[spasi]No. Telpon yang sudah ada di SMS Center	✓	✓
	B. User Tidak Terdaftar di SMS Center	-	-
	● Melakukan pendaftaran dengan ketik Daftar[spasi]No. Telpon yang belum ada di SMS Center	✓	✓
3	Memasukkan Keyword Module Terdaftar	-	-
	A. Module KOL	-	-
	● Melakukan pengecekan dengan mengetik KOL[spasi]Plat Nomor terdaftar	✓	✓
	B. Module BLK	-	-
	● Melakukan pengecekan dengan mengetik BLK[spasi]tipe konsumen[spasi]no ktp konsumen OK	✓	✓
	C. Module RNK	-	-
	● Melakukan pengecekan dengan mengetik RNK[spasi]Nomor Rangka Terdaftar	✓	✓

Lampiran 16

Form UAT Page 2

INDOMOBIL finance
PT INDOMOBIL FINANCE INDONESIA

	D. Module MSN	-	-
	● Melakukan pengecekan dengan mengetik MSN[spasi]Nomor Mesin Terdaftar	✓	✓
4	Memasukkan Keyword Help	-	-
	A. HELP	-	-
	● Menampilkan bantuan dengan mengetik HELP	✓ ✕	✓ <i>28/09</i>

Note :

1. Remark dapat diisi dengan kekurangan ataupun kesalahan yang masih ditemukan pada saat UAT.
2. Remark tidak boleh dilis dengan penambahan baru yang tidak sesuai dengan rancangan awal yang sudah disetujui.
3. Jika ada penambahan baru yang menurut User dibutuhkan, dapat dilakukan pengajuan Form Penambahan atau Perubahan Program kembali.

Lampiran 17

Tabel Data Pengujian User Acceptance Test Kolektor ke 31-100.

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
No	Pesan Masuk	Respon Balasan	Respon Seharusnya	Keterangan
31	MSN 1KRA368194	135 - CIANJUR \n1602147 \n(APEP HADIYANA) \nOD 895 HARI \nTOYOTA \nAGYA G M/T \nF 1430 YD \nMH-KA4DA3JGJ111135 \n1KRA368194	(sudah sesuai)	Passed
32	KOL BA4778OD	340 - PADANG \n1701627 \n(BAMBANG KRISTIAN) \nSUDAH DITARIK \nHONDA \nNEW BEAT FI STREET \nBA4778OD \nMH1JFZ212HK077279 \nJFZ2E1082363	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
33	RNK 261460	131 - BEKASI \n300466 \n(MOH.IMRON) \nSUDAH LUNAS \nSUZUKI \nSMASH FD 110 XCDF \nB 5171 ZP \n261460 \n261174	(sudah sesuai)	Passed
34	Kol b1064ujf	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
35	Kol d1108ptw	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
36	Kol b1307cs	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
37	KOL B144ABS	700 - JAKARTA R4 \n1501188 \n(ANDI BACHTIAR) \nOD 370 HARI \nNISSAN \nX-TRAIL 2.5L NEW XT A/T \nB144ABS \nMHBF3CF1CFJ004587 \nQR25236401L	(sudah sesuai)	Passed
38	Kol KH8585NP	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
39	MSN 1KRA368194	135 - CIANJUR \n1602147 \n(APEP HADIYANA) \nOD 894 HARI \nTOYOTA \nAGYA G M/T \nF 1430 YD \nMH-KA4DA3JGJ111135 \n1KRA368194	(sudah sesuai)	Passed
40	KOL F1088CL	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
41	MSN 1KRA368194	135 - CIANJUR \n1602147 \n(APEP HADIYANA) \nOD 895 HARI \nTOYOTA \nAGYA G M/T \nF 1430 YD \nMH-KA4DA3JGJ111135 \n1KRA368194	(sudah sesuai)	Passed
42	KO1 D1652AEN	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
43	MSN INZZ086716	No.Mesin INZZ086716 Tidak Ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
44	Kol b1104vkv	104 - PURI INDAH \n1700494 \n(SALBINI) \nOD 609 HARI \nSUZUKI \nNEW ERTIGA GL M/T AIRBAG \nB1104VKV \nMHYZE81SHJ303670 \nK14BT1231779	(sudah sesuai)	Passed
45	Kol f1223ew	130 - BOGOR \n1501363 \n(BUDI SANTOSO) \nLANCAR \nDATSUN \nGO TYPE T ACTIVE M/T \nF1223EW \nMHB2CH2FFJ010343 \nHR12746235T	(sudah sesuai)	Passed
46	Kol B1985URY	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
47	KOL B1451UZM	231 - YOGYAKARTA USED CAR \n1700119 \n(MUHAMMAD ANDI ARIFUDDIN) \nSUDAH DITARIK \nHONDA \nBRIO 1.3 S M/T \nB1451UZM \nMR-HDD2720DP311191 \nL13Z51211018	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
48	KOL AD9448NJ	231 - YOGYAKARTA USED CAR \n1600132 \n(MUSRIFIN) \nOD 846 HARI \nDAIHATSU \nXENIA Li VVT-i \nAD9448NJ \nMH-KV1AA2J9K050730 \nDN93821	(sudah sesuai)	Passed
49	KOL B9087WAF	753 - TANGERANG R4 \n1500593 \n(EDISON RAFLES) \nOD 706 HARI \nSUZUKI \nPICK UP CARRY F/D \nB9087WAF \nMHYESL415FJ711866 \nG15AID997296	(sudah sesuai)	Passed
50	Kol f1030an	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
51	Kol f1030hn	130 - BOGOR \n1600520 \n(DEDEN ABDURRAHMAN) \nOD 614 HARI \nDAIHATSU \nXENIA F651 RV \nF1030HN \nMH-KV1BA2JCK007423 \nDK20641	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
52	Kol b1982wov	753 - TANGERANG R4 \n1700360 \n(HAERONI) \nOD 864 HARI \nSUZUKI \nNEW ERTIGA AIRBAG ABS GX M/T \nB1982WOW \nMHYKZE81SHJ308465 \nK14BT1239067	(sudah sesuai)	Passed
53	MSN JM51E212911	No.Mesin JM51E212911 Tidak Ditemukan	(sudah sesuai)	Passed
54	MSN JM51E1212911	226 - SALATIGA \n1901142 \n(TITIK ARFIANI) \nOD 129 HARI \nHONDA \nVARIO TECHNO 125 PGM FI CBS ISS \nH 4174 ARC \nMH1JM5118KK213082 \nJM51E1212911	(sudah sesuai)	Passed
55	MSN KC41E1018944	No.Mesin KC41E1018944 Tidak Ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
56	KOL B1695KRE	702 - KELAPA GADING R4 \n1500030 \n(SUHAEBATUL ASLAMIAH) \nOD 1578 HARI \nNISSAN \nALL NEW GRAND LIVINA 1.5L SV CVT \nB1695KRE \nMHBG3CG1CEJ034903 \nHR15723640T	(sudah sesuai)	Passed
57	Kol b2910sfs	753 - TANGERANG R4 \n1501623 \n(KARNO ARIWIBOWO) \nOD 447 HARI \nDATSUN \nGO+ TYPE T OPTION \nB2910SFS \nMHBJ1CH2FFJ027391 \nHR12735851T	(sudah sesuai)	Passed
58	Kol n8375kf	241 - MALANG \n1700172 \n(HERNI RINA JAYAWATI) \nOD 494 HARI \nSUZUKI \nPICK UP CARRY F/D \nN8375KF \nMHYESL415HJ781008 \nG15AID1070724	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
59	kol b1621zfx	753 - TANGERANG R4 \n1600777 \n(JHONI SIREGAR) \nOD 949 HARI \nSUZUKI \nNEW ERTIGA GL M/T AIRBAG \nB1621ZFX \nMHYKZE81SGJ312642 \nK14BT1186098	(sudah sesuai)	Passed
60	Kol f8105vl	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
61	Kol f111oa	138 - PELABUHAN RATU \n1900954 \n(HELMI SAPUTRA) \nOD 6 HARI \nMITSUBISHI \nPAJERO SPORT 2.5D EXCEED A/T \nF111OA \nMMBGR-KH40AF007443 \n4D56UCBW0398	(sudah sesuai)	Passed
62	KOL D111NR	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
63	Kol b2746tox	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
64	Kol D1282vbt	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
65	Kol D1074wm	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
66	Kol D8098ZG	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
67	KOL D1472ADP	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
68	Kol f9973ck	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
69	Kol f1384lf	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
70	Kol h1734RV	227 - KENDAL \n1800813 \n(MUGIYONO) \nOD 114 HARI \nSUZUKI \nAPV PICK UP (MEGA CARRY) \nH1734RV \nMHYGDN41TJJ400442 \nG15AID409130	(sudah sesuai)	Passed
71	Kol d1639oe	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
72	Kol h3871aqc	226 - SALATIGA \n1900339 \n(NURUL CHOBIBAH) \nOD 42 HARI \nYAMAHA \nJUPITER Z1 CW FI \nH3871AQC \nMH3UE1120JJ192316 \nE3R5E0203030	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
74	kol ba1552ow	340 - PADANG \n1701614 \n(SUKARDI) \nOD 642 HARI \nHONDA \nMOBILIO E M/T \nBA1552OW \nMHRDD4750HJ704105 \nL15Z136611741	(sudah sesuai)	Passed
75	KOL BA1552OW	340 - PADANG \n1701614 \n(SUKARDI) \nOD 642 HARI \nHONDA \nMOBILIO E M/T \nBA1552OW \nMHRDD4750HJ704105 \nL15Z136611741	(sudah sesuai)	Passed
76	Kol h4453bje	220 - SEMARANG \n1901822 \n(AFIFUDIN) \nOD 62 HARI \nHONDA \nVARIO 150 ESP EXCLUSIVE \nH4453BJE \nMH1KF4118KK545984 \nKF41E1546409	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
77	KOL BA1024FA	340 - PADANG \n1900463 \n(SYAMSU AKIKURATA) \nOD 134 HARI \nMITSUBISHI \nXPANDER GLS M/T \nBA1024FA \nMK2NCWMANKJ000264 \n4A91GJ3839	(sudah sesuai)	Passed
78	kol b2607sob	753 - TANGERANG R4 \n1602140 \n(DEWI TRISNOWATI) \nOD 346 HARI \nDATSUN \nGO+ T OPTION M/T AIRBAG \nB2607SOB \nMHBJ1CH2FGJ052156 \nHR12771837T	(sudah sesuai)	Passed
79	KOL BA1602OY	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
80	KOL BA6217GJ	341 - PAINAN \n1800105 \n(MAISIR) \nSUDAH DITARIK \nHONDA \nVARIO 125 ESP CBS ISS \nBA6217GJ \nMH1JFV111JK844961 \nJFV1E1850357	(sudah sesuai)	Passed
81	Kol bn1417pb	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
82	Kol l9766wa	770 - SURABAYA R4 \n1800230 \n(DIDIK HARIYANTO) \nOD 442 HARI \nSUZUKI \nFUTURA PICK UP FD \nL9766WA \nMHYESL415JJ701404 \nG15AID1101734	(sudah sesuai)	Passed
83	Kol b1285syb	706 - JAKARTA USED CAR \n1600023 \n(ABDUL KODIR JAELANI) \nOD 1254 HARI \nDAIHATSU \nLUXIO 1.5 D M/T \nB1285SYB \nMH- KW3CA1JDK006797 \nDDP7114	(sudah sesuai)	Passed
84	Kol bp173ty	360 - BATAM \n1300294 \n(NETTY) \nOD 891 HARI \nNISSAN \nJUKE 1.5 RX RED EDITION \nBP173TY \nMHBJ1CG1ADJ016786 \nHR15393386C	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
85	Kol f1501el	130 - BOGOR \n1600451 \n(RIMA MULYATI) \nOD 261 HARI \nSUZUKI \nKARIMUN WAGON R GL \nF1501EL \nMHYHMP31SEJ118442 \nK10BT1022676	(sudah sesuai)	Passed
86	KOL BA6217GJ	341 - PAINAN \n1800105 \n(MAISIR) \nSUDAH DITARIK \nHONDA \nVARIO 125 ESP CBS ISS \nBA6217GJ \nMH1JFV111JK844961 \nJFV1E1850357	(sudah sesuai)	Passed
87	KOL B1112VKJ	700 - JAKARTA R4 \n1502597 \n(ANUGERAH PEKERTI) \nOD 38 HARI \nNISSAN \nALL NEW MARCH 1.2 A/T \nB1112VKJ \nMN-TFBUK13Z0087565 \nHR12455315B	(sudah sesuai)	Passed
88	KOL BA4017OS	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
89	KOL BA6244BR	340 - PADANG \n1202192 \n(KARIMIL ARSYAD) \nSUDAH LUNAS \nHONDA \nSCOOPY A/T \nBA6244BR \nMH1JF6114CK405326 \nJF61E1400076	(sudah sesuai)	Passed
90	KOL BA4327QS	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
91	KOL BA3385BW	340 - PADANG \n1101365 \n(FITRI NINGSIH) \nSUDAH DITARIK \nSUZUKI \nHAYATE UW 125 SC1 \nBA3385BW \nMH8CF4EJABJ104381 \nF4E2ID104625	(sudah sesuai)	Passed
92	KOL BA9073RC	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
93	KOL BA9725BB	340 - PADANG \n1700665 \n(YESI ANWAR) \nSUDAH DITARIK \nMITSUBISHI \nCOLT T120SS STD PU \nBA9725BB \nMH-MU5TU2EHK195593 \n4G15R22001	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
94	KOL BA8049AC	340 - PADANG \n1801442 \n(NURHAYATI) \nSUDAH DITARIK \nSUZUKI \nFUTURA PICK UP FD \nBA8049AC \nMHYESL415JJ719978 \nG15AID1121450	(sudah sesuai)	Passed
95	KOL BA1448QA	340 - PADANG \n1401934 \n(YUDI) \nSUDAH DITARIK \nNISSAN \nALL NEW GRAND LIVINA 1.5L SV M/T \nBA1448QA \nMHBG3CG1FEJ028304 \nHR15716522T	(sudah sesuai)	Passed
96	KOL BA2454OC	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
97	KOL BA2579AA	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed

73	KOL B1641UZY	706 - JAKARTA USED CAR \n1601741 \n(SUKAMTO) \nOD 950 HARI \nTOYOTA \nETIOS VALCO 1.2G M/T \nB1641UZY \nMHFK39BTXD2003126 \n3NRV111766	(sudah sesuai)	Passed
98	KOL BA4778OD	340 - PADANG \n1701627 \n(BAMBANG KRISTIAN) \nSUDAH DITARIK \nHONDA \nNEW BEAT FI STREET \nBA4778OD \nMH1JFZ212HK077279 \nJFZ2E1082363	(sudah sesuai)	Passed
99	KOL BA5978OU	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed
100	KOL BA4752QB	Data yang ditelusuri tidak ditemukan	(sudah sesuai)	Passed