

PEMBUATAN APLIKASI CHATBOT KOLEKTOR DENGAN METODE EXTREME PROGRAMMING DAN STRATEGI FORWARD CHAINING.

Muhamad Sidik*¹, Bambang Gunawan², Dina Anggraini³

^{1,2,3} Magister Manajemen Sistem Informasi, Universitas Gunadarma Depok
Email: ¹ m.sidik19@yahoo.com, ² bambang_gunawan@staff.gunadarma.ac.id,
³ dina_anggraini@staff.gunadarma.ac.id
*Penulis Korespondensi

(Naskah masuk: 05 November 2020, diterima untuk diterbitkan: 22 Maret 2021)

Abstrak

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 tentu akan menjadi sebuah masalah. Untuk menangani permasalahan menumpuknya permintaan layanan informasi seperti keluhan pelanggan, permintaan data konsumen, pelayanan nasabah baru, dan permasalahan serupa maka banyak diterapkan bantuan asisten virtual atau biasa disebut dengan *chatbot*. *Chatbot* merupakan aplikasi asisten virtual yang mampu melakukan interaksi secara langsung kepada setiap pesan yang masuk tanpa perlu menunggu operator untuk membalas pesan-pesan tersebut, sehingga *chatbot* merupakan solusi yang dinilai efektif untuk menangani permasalahan dalam permintaan layanan informasi yang menumpuk. Selain itu, pengembangan *chatbot* juga berguna untuk memberikan layanan pemberian informasi yang lebih responsif bagi suatu perusahaan. Pada penelitian ini, dilakukan pembuatan suatu aplikasi *chatbot* dengan *platform* LINE menggunakan metode *extreme programming* dan strategi *forward chaining* yang bertujuan untuk memenuhi kebutuhan informasi bagi kolektor di PT. Indomobil Finance Indonesia (PT.IMFI) serta memberikan layanan informasi dalam waktu singkat. Hasil dari pengembangan sistem berjalan pada layanan informasi kolektor PT. IMFI menghasilkan sebuah aplikasi *chatbot* berbasis LINE. Hasil dari pengujian *blackbox testing* serta *user acceptance test* menghasilkan persentase keberhasilan aplikasi Chatbot Kolektor mencapai 95%, dan persentase kegagalan aplikasi 5%, angka 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.

Kata kunci: *chatbot, line, asisten virtual, extreme programming, forward chaining.*

MANUFACTURING OF CHATBOT FOR COLLECTOR APPLICATION WITH EXTREME PROGRAMMING METHOD AND FORWARD CHAINING STRATEGY

Abstract

Requests for information services at a certain point in a company are certainly a routine activity, but if requests for information services are carried out in large numbers, it will certainly be a problem. To deal with the problem of accumulating requests for information services As customer complaints, request for data, consumer demand service, new borrowers and problems like it so, virtual assistants, commonly known as chatbots, is applied. Chatbot is a virtual assistant application that is able to interact directly with every incoming message without the need to wait for the operator to reply the messages, so chatbot is a solution that can solve problems in accumulating information service requests. In addition, chatbot development is also useful for providing services to be more responsive for a company. In this study, a chatbot application was made with the LINE platform using the extreme programming method and forward chaining strategy which has the intention of meeting information needs by collectors at PT. Indomobil Finance Indonesia (PT. IMFI) And provide the information in a short time. The results of the development of the system running on the collector information service of PT. IMFI produces a LINE-based chatbot application. The results of the blackbox testing and user acceptance test resulted in a successful percentage of the Collector's Chatbot application reaching 95%, and the percentage of application failure was 5%, this figure shows that the process of making a chatbot application to meet the information service needs of PT. Indomobil Finance Indonesia was successfully carried out and all functions were running as expected

Keywords: *chatbot, line, virtual assistant, extreme programming, forward chaining*

1. PENDAHULUAN

Perkembangan teknologi komunikasi dan informasi 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 dan juga bertujuan untuk menyelesaikan berbagai permasalahan yang dihadapi oleh *customer relation* mereka, salah satunya adalah permasalahan menumpuknya permintaan layanan informasi dari konsumen. Pembaharuan teknologi ini dilakukan karena *customer relation* adalah kunci sukses bagi suatu perusahaan, dimana *customer relation* adalah pintu gerbang pertama antara perusahaan dengan pelanggan, 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 & E. L. Amalia (2017), disebutkan bahwa *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 (Jati & Maarif, 2018). Menurut Jati & Maarif (2018) disebutkan bahwa *Chatbot* (juga dikenal sebagai *talkbot*, *chatterbot*, *bot*) dapat diartikan sebagai sebuah perangkat lunak yang diperuntukkan untuk mereplikasi percakapan intelek kepada satu manusia atau lebih (LINE, 2016). Sedangkan menurut literasi lain disebutkan bahwa *chatbot* dapat diartikan sebagai suatu program aplikasi yang dibuat secara khusus untuk memberikan respon pesan kepada manusia baik pesan tersebut berupa teks atau tulisan, suara atau *audio*, bahkan keduanya (Setiawan, 2018; Akhsan & Faizah, 2017).

Chatter Bot atau *Chatbot* yang menggunakan *machine learning* atau biasa dikenal sebagai *smartbot* adalah *chatbot* yang mampu memberikan respon terhadap pesan dalam bahasa manusia melalui proses pembelajaran, tidak seperti *chatbot* biasa yang jawaban dari pesan yang masuk sudah ditentukan sebelumnya (Ananda, Firdha, Yusuf & Ardiansyah, 2018).

Elisabet dan Afrianto berpendapat dalam jurnalnya yang berjudul "Rancang Bangun Aplikasi *Chatbot* Informasi Objek Wisata Kota Bandung dengan Pendekatan *Natural Language Processing*"

bahwa pada dunia bisnis dan industri, pelayanan yang berupa layanan pelanggan sudah mulai menggunakan bantuan *chatbot* untuk memberikan respon pertanyaan dari konsumen mereka. Pelayanan menggunakan *chatbot* dinilai cukup efektif karena *chatbot* akan memberikan respon terhadap pertanyaan dari konsumen dengan sangat responsif, hal ini membuat konsumen akan memperoleh informasi yang mereka butuhkan saat itu juga (Elisabet & Afrianto, 2015). Manfaat dari *chatbot* adalah untuk pelayanan informasi kepada pengguna tanpa melibatkan operator sebagai penjawab, jadi pekerjaan operator sudah di alihkan ke aplikasi *chatbot* (Wijaya, Sotyohadi & Setiawan, 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 permintaan 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.

Bedasarkan data dari hasil penelitian pada tahun 2020 yang dikerjakan oleh We Are Social yang membahas tren media sosial yang digunakan di Indonesia, sosial media terfavorit adalah sebagai

berikut : 1) Youtube digunakan oleh 88%, 2) Whatsapp digunakan sebanyak 84%, 3) Facebook dengan jumlah 82%, 4) Instagram sebanyak 79%, 5) Twitter dengan jumlah penggunaan 56%, 6) LINE dengan pengguna 56%, 7) FB Messenger sebanyak 50%, 8) LinkedIn digunakan oleh 35%, 9) Pinterest sebanyak 34%, 10) WeChat dengan pengguna 29%, 11) Snapchat digunakan oleh 28%, 12) Skype digunakan sebanyak 25%, 13) Tiktok dengan jumlah pengguna 25%, 14) Tumblr dengan pengguna 22%, 15) Reddit sebanyak 18%, dan 16) Sina Weibo dengan jumlah pengguna sebanyak 17% (Kemp, 2020; Bagus, Ari & Widhi, 2018).

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. Sementara, media sosial LINE memberikan akses Messaging API-nya secara bebas sejak tahun 2016 bulan September, hal ini memungkinkan para *developer* program untuk membuat chatbot pada aplikasi LINE (Lestari, 2016).

Chatbot berbasis LINE memanfaatkan teknologi restful API untuk berkomunikasi. REST atau kependekan dari *Representational State Transfer* adalah suatu fungsi dari protokol *Hypertext Transfer Protocol* atau yang lebih dikenal dengan HTTP dimana REST digunakan untuk berkomunikasi secara penuh dalam proses pertukaran representasi seperti mendapatkan suatu nilai dengan mengirimkan suatu parameter kepada suatu server REST (Junadhi & Mardainis, 2019). Dengan adanya REST maka setiap pesan yang dikirim pengguna mampu dikenali oleh *chatbot* dan *chatbot* mampu memberikan respon atas pesan yang dikirimkan oleh pengguna.

Dalam memberikan respon, metode populer yang digunakan adalah *Forward Chaining*, Carolina dan Adi mengungkapkan bahwa *Forward Chaining* adalah suatu metode pencarian dimana dalam proses pencarian tersebut dilakukan dengan cara pelacakan secara *forwarding* atau ke depan. Cara tersebut memungkinkan suatu proses dimulai dengan mengumpulkan informasi yang sudah didapatkan sebelumnya kemudian digabungkan dengan suatu aturan tertentu untuk mendapatkan hasil berupa kesimpulan (Carolina & Adi, 2019). Dengan menggunakan *forward chaining*, maka *chatbot* mampu memberikan jawaban yang sudah dipersiapkan oleh *developer*.

Salah satu perusahaan yang membutuhkan adanya pengembangan aplikasi *chatbot* adalah PT. IMFI (Indomobil Finance Indonesia). PT. IMFI adalah suatu perusahaan yang bidang bisnisnya berupa jasa pembiayaan, baik itu kendaraan bermotor, pembiayaan 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 suatu perusahaan 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 bervariasi, hal ini sesuai dengan *jobdesk* yang telah ditentukan masing-masing cabang bagi 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 atau *simcard* yang di pergunakan, jika *simcard* atau kartu sim yang di pergunakan 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, Firdha, Yusuf, dan Ardiansyah tentang MILKI BOT (Ananda, Firdha, Yusuf & Ardiansyah, 2018). Penelitian yang dikerjakan telah berhasil menyelesaikan permasalahan yang dihadapi oleh *customer service* pada UKM Misnu. Selain itu penelitian sejenis juga dikerjakan oleh Junadhi, dan Mardainis tentang LINE Chatbot Informasi Cuaca Wilayah Indonesia (Wijaya, Sotyohadi & Setiawan, 2015), 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 dibutuhkan sebuah layanan *chatbot* dengan platform LINE untuk memenuhi permintaan layanan informasi bagi kolektor PT. Indomobil Finance Inonesia 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.

2. METODE PENELITIAN

Proses pembuatan aplikasi *chatbot* yang dikerjakan pada penelitian ini secara garis besar menggunakan 2 metode, metode tersebut yaitu metode *Extreme Programming* serta *Forward Chaining*. Pada proses pengerjaan pembuatan aplikasi ini dilakukan 4 tahapan sesuai dengan tahapan yang ada pada metode *Extreme Programming* (Benedictus, Wowor & Sambul, 2017), tahapan tersebut adalah:

2.1 Tahap Planning

Tahap *Planning* ini terdiri dari pengumpulan data yang terdiri dari *manual* penggunaan aplikasi dari program yang saat ini sudah berjalan di PT. Indomobil Finance Indonesia, juga kebutuhan *user* yang didapatkan dengan wawancara langsung kepada *user* dari departemen *Remedial and Collection* dengan menanyakan apa saja yang perlu dikembangkan dan dihilangkan dari program yang saat ini sudah berjalan di PT. Indomobil Finance Indonesia, selain itu, pada tahap ini juga peneliti mempelajari alur program yang sudah tersedia dengan cara melakukan proses *debug* pada sistem yang ada untuk memahami proses dan alur kerja yang saat ini sudah berjalan.

2.2 Tahap Design

Tahap *Design* pada penelitian ini adalah dengan melakukan perancangan tampilan serta perancangan *input* pada aplikasi

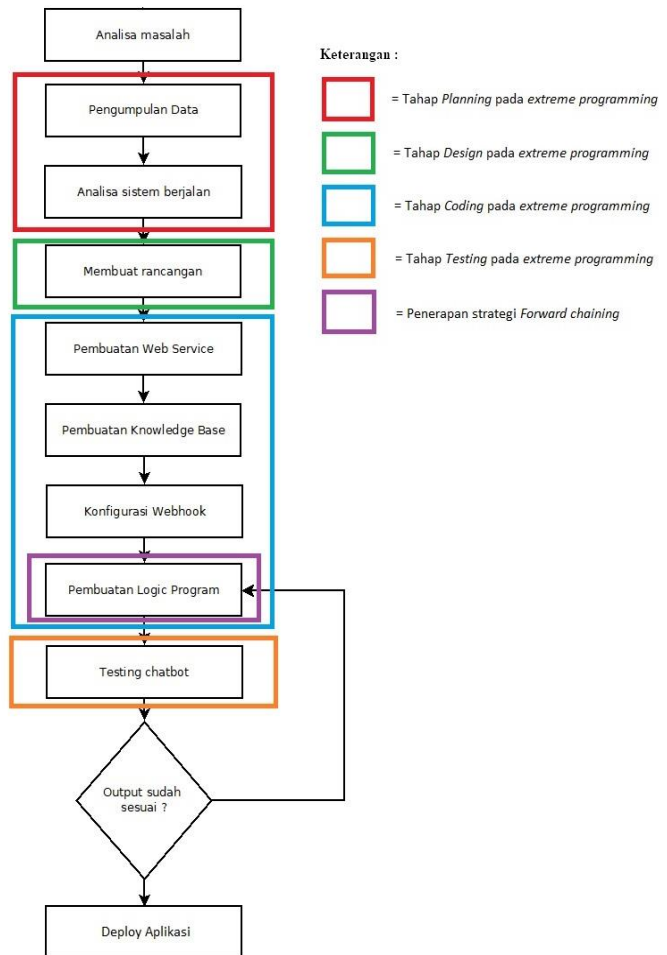
2.3 Tahap Coding

Pada tahap *Coding*, kegiatan yang dilakukan terdiri dari pembuatan *web service*, penentuan akses database sebagai *knowledge base*, melakukan konfigurasi *webhook*, serta pembuatan *logic* program.

2.4 Tahap Testing

Tahap *testing* dilakukan dengan memasukkan *input* pada *chatbot*.

Strategi *forward chaining* diterapkan untuk menentukan alur percakapan yang akan diterapkan



Gambar 1. Tahapan Penelitian

Gambar 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 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.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

Untuk mendapatkan jawaban 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 1 dan tabel 2 menunjukkan hasil perbandingan aplikasi SMSCenter dengan Chatbot Kolektor.

Tabel 1. Hasil Perbandingan Aplikasi Berjalan Dengan Aplikasi Yang Dibangun Pada Penelitian Ini

No	Permasalahan	Aplikasi Berjalan	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 gangguan	Ketika Perlu hard reset modem gsm	Hanya melakukan pengecekan apakah webhook dapat diverifikasi.

Berdasarkan pada Tabel 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 menggunakan pesan singkat atau lebih dikenal dengan *SMS*. Hal ini membuat aplikasi SMSCenter bergantung secara menyeluruh pada kartu sim, sedangkan pada aplikasi yang dikembangkan di penelitian ini yaitu Chatbot Kolektor, penggunaan kartu sim hanya bergantung sebagian, yaitu ketika chatbot kolektor perlu mengirimkan pesan *One Time Password* atau yang lebih dikenal dengan *OTP*, *OTP* ini merupakan sebuah pesan singkat yang berisikan sebuah deret angka yang berfungsi sebagai *Password*, *password*

ini hanya akan berlaku 1 kali saja ketika dimasukkan oleh pengguna di sistem chatbot kolektor, pengiriman pesan *OTP* ini ditujukan 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 pembiayaan, 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 membebaskan 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 Januari 2020, 3 Januari 2020, 4 Januari 2020, 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* mengalami *system failure* sehingga modem tidak dapat berjalan dengan semestinya 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. Permasalahan yang terjadi selama ini disebabkan oleh kondisi simcard yang *hang*, ataupun *modem gsm* yang mengalami kondisi *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 2. Hasil Perbandingan *Response Time* Antara Aplikasi Berjalan Dengan Aplikasi Yang Dibangun Pada Penelitian Ini

No	Tugas	Response Time(detik)	
		SMSCenter	Chatbot Kolektor
1	Modul Kolektor	26.10	1.97
2	Modul Blacklist	26.35	1.21
3	Modul Mesin	19.02	3.75
4	Modul Rangka	25.54	6.76
Jumlah		97.01	13.69
Rata-rata waktu tes		24.25	3.42

Hasil diatas merupakan hasil *response time* yang didapatkan dari proses perhitungan secara *manual* menggunakan *stopwatch* yang difungsikan untuk menunggu hingga system mengirimkan balasan dari pesan yang masuk, dan berdasarkan hasil perbandingan *response time* yang ditunjukkan pada tabel 2, maka didapatkan data bahwa pada aplikasi SMSCenter waktu yang diperlukan untuk menampilkan respon pada modul kolektor adalah 26.10 detik, kemudian pada modul blacklist aplikasi SMSCenter membutuhkan waktu 26.35 detik untuk menampilkan respon. Pada modul mesin, aplikasi SMSCenter membutuhkan waktu 19.02 detik, dan pada modul rangka waktu yang diperlukan oleh aplikasi SMSCenter adalah 25.54.

Sementara untuk aplikasi yang dibangun pada penelitian ini yaitu Aplikasi Chatbot Kolektor membutuhkan waktu 1.97 detik untuk menampilkan respon pada modul kolektor. Untuk modul blacklist, aplikasi Chatbot Kolektor memerlukan waktu 1.21 detik untuk membalas respon. Untuk modul mesin, waktu yang dibutuhkan adalah 3.75 detik, dan pada modul rangka, waktu yang dibutuhkan aplikasi Chatbot Kolektor adalah 6.76 detik.

Dari perbandingan tersebut, maka rata-rata respon yang diperlukan oleh aplikasi SMSCenter untuk menampilkan respon adalah 24.25 detik, sementara pada aplikasi Chatbot Kolektor waktu rata-rata yang dibutuhkan untuk menampilkan respon adalah 3.42 detik, sehingga dapat disimpulkan bahwa aplikasi Chatbot Kolektor mampu menampilkan respon balasan dari pengguna lebih cepat dibandingkan dengan aplikasi yang saat ini berjalan yaitu SMSCenter.

Berdasarkan hasil yang terdapat pada tabel 1 dan tabel 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.

3.2 PEMBAHASAN

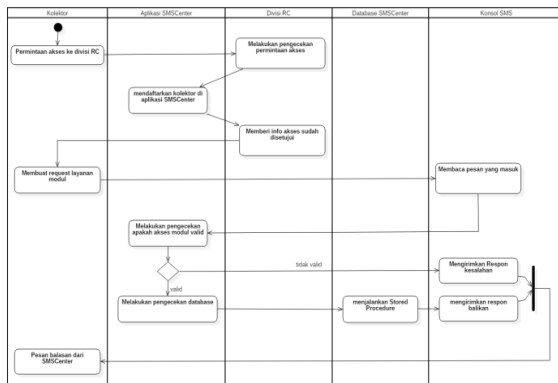
3.2.1 Pengumpulan Data (Tahap *Planning* Pada *Extreme Programming*)

Tahapan penelitian yang dimulai dengan tahapan *planning*, pada tahap *planning* ini peneliti mengumpulkan kebutuhan yang diperlukan untuk membangun aplikasi Chatbot Kolektor. Pada penelitian ini pengumpulan data dilakukan secara observasi. Teknik observasi ini 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 diproses oleh peneliti dalam penelitian ini berupa data primer yaitu data kolektor dan data kendaraan yang terdaftar di PT. Indomobil Finance Indonesia.

Adapun pengumpulan data yang dikerjakan oleh peneliti dalam penelitian ini adalah dengan teknik wawancara, teknik wawancara ini 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.

3.2.2 Analisa Sistem Berjalan (Tahap Planning Pada Extreme Programming)



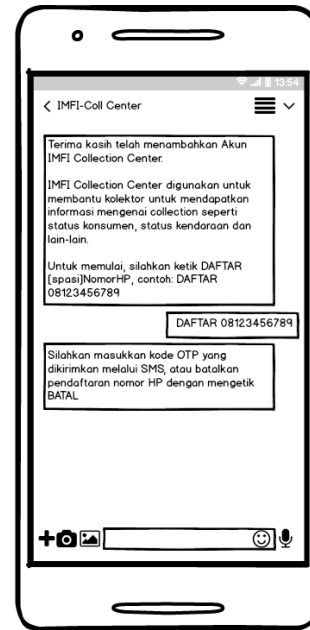
Gambar 2. Alur dari Sistem yang saat ini berjalan

Berdasarkan gambar 2, 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 membuat 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.

3.2.3 Membuat Rancangan (Tahap Design pada Extreme Programming)

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*. Contoh rancangan yang dilakukan pada tahap *design* pada penelitian ini dapat dilihat pada gambar 3.



Gambar 3. Rancangan Tampilan Aplikasi Chatbot Kolektor

3.2.4 Tahap Pembuatan Coding

3.2.4.1 Pembuatan web service (Tahap Coding pada Extreme Programming)

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 atau VS2010 dan bahasa pemrograman menggunakan visual basic atau VB.Net 2010.

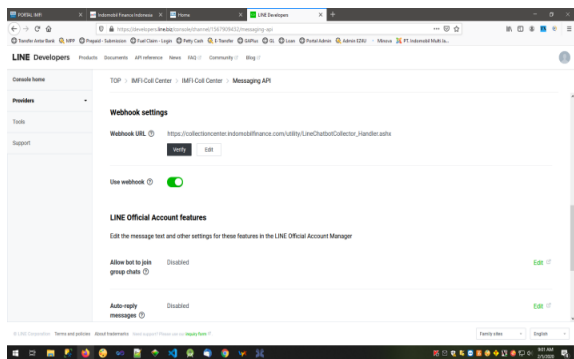
3.2.4.2 Pembuatan Knowledge Base (Tahap Coding pada Extreme Programming)

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.

3.2.4.3 Konfigurasi Webhook (Tahap Coding pada Extreme Programming)

Konfigurasi *webhook* pada penelitian ini dilakukan untuk menghubungkan *web service* dengan layanan *API LINE*. Layanan *API (Application Programming Interface) LINE* adalah sebuah layanan yang memungkinkan sebuah sistem dapat berkomunikasi dengan platform milik LINE

sehingga *developer* program mampu membangun suatu aplikasi berbasis LINE yang dapat berkomunikasi langsung dengan Fitur yang terdapat pada LINE. Untuk bisa melakukan konfigurasi, maka dibutuhkan akun LINE developer, didalam akun LINE developer ini juga peneliti melakukan konfigurasi lain untuk *chatbot*, misal ikon *chatbot*, nama *chatbot*, dan lain-lain. Salah satu konfigurasi yang dilakukan pada penelitian ini adalah memasukkan alamat *URL webhook*, kegiatan ini dapat dilihat pada gambar 4.



Gambar 4. Konfigurasi *Webhook* pada halaman LINE Developer

3.2.4.4 Pembuatan Logic Program (Tahap Coding pada Extreme Programming)

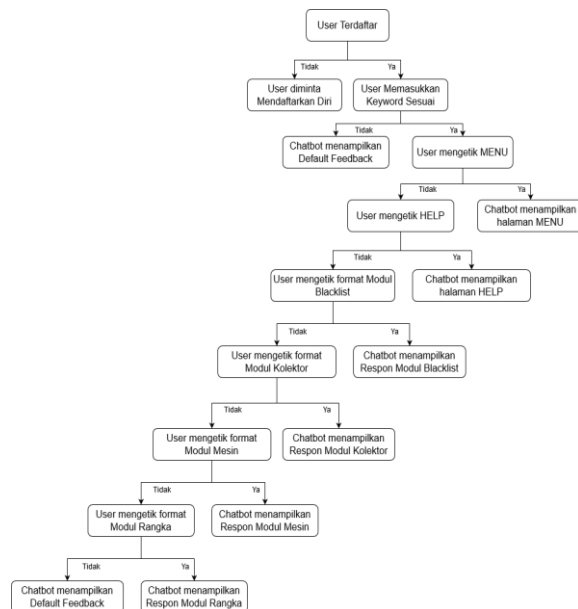
Pembuatan *logic* program dilakukan pada *web service* yang sudah disiapkan sebelumnya, pada proses pembuatan *logic* program ini strategi *forward chaining* diterapkan. Dengan menerapkan strategi jika-maka yang ada pada *forward chaining*, seluruh input yang akan dimasukan oleh *user* untuk berinteraksi dengan *chatbot* akan ditentukan pada tahap ini, respon yang sudah disiapkan pada *knowledge base* juga akan diatur sebagai respon dalam tahap ini.

3.2.4.5 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 pesan atau kata yang diinginkan, semisal “saya tidak mengerti maksud anda?”, “mohon masukkan kata kunci yang sesuai”. Setelah ditampilkan hasil dari masukan tersebut, *user* bisa memasukkan kata kunci kembali atau mengakhiri percakapan.

Gambar 5 menunjukkan alur aplikasi Chatbot Kolektor dengan menerapkan aturan “jika-maka” yang terdapat pada metode *forward chaining*.

Aturan “jika-maka” dalam *forward chaining* ini memungkinkan aplikasi chatbot untuk menentukan bagaimana respon yang akan diberikan jika *input* yang dilakukan oleh *user* adalah *input* yang sesuai atau tidak sesuai. Dengan *forward chaining*, aplikasi Chatbot Kolektor juga mampu menentukan default *feedback* yang akan diberikan jika *keyword* pesan yang dimasukan oleh *user* tidak dikenali.



Gambar 5. Bagan *Network Forward Chaining*

3.2.5 Tahap Testing Aplikasi (Tahap Testing pada metode Extreme Programming)

3.2.5.1 Blackbox Testing

a) Analisa kebutuhan dan spesifikasi.

Perencanaan pengujian perangkat lunak dengan *blackbox testing* dilakukan dengan melakukan 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 3 menunjukkan hasil dari analisa kebutuhan dan spesifikasi yang digunakan pada pengujian perangkat lunak dengan menggunakan *blackbox testing*.

Tabel 3. Analisa Kebutuhan *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

b) 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.

c) Pemilihan *Output*.

Proses pemilihan *output* dilakukan untuk memberikan jawaban dari tiap pesan yang masuk serta alur dari proses yang akan berjalan pada halaman pesan Chatbot Kolektor.

d) 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.

e) Review Hasil

Berdasarkan tabel 4 yang berisi hasil pengujian aplikasi dengan menggunakan *blackbox testing*, didapatkan kesimpulan bahwa seluruh fungsi yang berjalan pada aplikasi Chatbot Kolektor sudah berjalan sesuai dengan fungsinya dan respon yang diharapkan telah ditampilkan dengan sesuai.

Tabel 4. Hasil Pengujian *Blackbox Testing*

No	Langkah Skenario	Hasil Diharapkan	Yang 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 1234567890123456 (NIK ketik asal)	Chatbot merespon data NIK tidak ada	Chatbot merespon data OK	Pengujian Gagal
3	User input KOL K640808450467002 (Data Tidak Tersedia)	Chatbot merespon format pesan error	Chatbot merespon format sms salah	Pengujian Gagal
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

No	Langkah Skenario	Hasil Diharapkan	Yang 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

3.2.5.2 User Acceptance Test

UAT atau *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 selaku staff pada divisi *Remedial and Collection* dan Bapak Hengky selaku Kepala Departemen *Remedial and Collection*, serta kepada 98 partisipan lain yang terdiri dari Staff, Coordinator, dan Supervisor yang ada di Departemen *Remedial and Collection* sehingga total partisipan untuk UAT dalam penelitian ini adalah 100 responden untuk melakukan pemeriksaan tiap fungsi yang ada pada Chatbot Kolektor.

Dari 100 data yang dilakukan pengujian, terdapat 5 data yang mengalami gagal uji karena respon yang diberikan tidak sesuai dengan yang diharapkan, gagal uji tersebut dikarenakan terdapat format balasan tidak sesuai dengan yang seharusnya pada 4 pesan yang dikirimkan, selain itu terdapat juga gagal uji pada 1 pesan masuk ketika data yang dimasukkan adalah data nik berisi angka acak, *chatbot* merespon dengan pesan OK, seharusnya terdapat validasi terlebih dahulu apakah data nik yang dimasukkan sesuai dengan format nik yang sesuai.

Berikut perhitungan persentase keberhasilan Aplikasi Chatbot Kolektor berdasarkan pengujian pada 100 data.

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

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

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

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

4. KESIMPULAN

Menurut data yang didapatkan dari proses penelitian ini, penulis menyimpulkan bahwa 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*, hasil pengujian menunjukkan persentase keberhasilan aplikasi Chatbot Kolektor mencapai 95% dan persentase kegagalan aplikasi adalah 5%. Dari angka tersebut, maka dapat disimpulkan 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 berhasil dibuat dalam penelitian ini, maka penulis dapat menyimpulkan aplikasi Chatbot Kolektor telah mampu mengatasi permasalahan seringkali gangguan pada sistem pemberi informasi serta mampu melakukan penghematan biaya operasional yang dikeluarkan oleh PT. Indomobil Finance Indonesia.

Saran bagi penelitian berikutnya, pembuatan aplikasi Chatbot Kolektor yang dilakukan dalam penelitian ini sebaiknya dapat disusun juga menggunakan berbagai bahasa pemrograman yang lainnya dalam pembuatan *web service*. 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 dalam penelitian ini bisa dipergunakan sebagai sebuah acuan untuk dapat mengembangkan aplikasi *chatbot* di berbagai *platform* lainnya seperti Whatsapp, dan Telegram

DAFTAR PUSTAKA

- AKHSAN, A. dan FAIZAH, F. 2017. Analisa dan perancangan interaksi chatbot reminder dengan user-centered design. *Jurnal Sistem Informasi*, 2(13), pp. 7-89.
- ANANDA, D. R., FIRDHA, I., YUSUF, M. A. S., dan ARDIANSYAH. 2018. Aplikasi chatbot (milki bot) yang terintegrasi dengan web cms untuk customer service pada ukm minsu. *Jurnal Cendikia*, XVI, pp. 100-106.
- BAGUS, S., ARI, K., dan 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).
- BENEDICTUS, R. R., WOWOR, H. Dan SAMBUL, A. 2017. Rancang bangun chatbot helpdesk untuk sistem informasi terpadu universitas sam ratulang. *EJournal Teknik Informatika*, 11(1).
- CAROLINA, I. dan ADI, S. 2019. Penerapan Metode Extreme Programming Dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen. *Jurnal IKRA-ITH*, 3(1).
- 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), pp. 49-54.
- HORMANSYAH, D. S. dan 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, pp. 24-28.
- JATI, D. dan MAARIF, M. 2018. The development of chatbot application on line messaging platform for customer service in jogja sewa kamera. *Compiler*, 7(2), pp. 91-98.
- JUNADHI dan MARDAINIS. 2019. Line chatbot informasi cuaca wilayah indonesia. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 10(1), pp. 101-109.
- KEMP, S. 2020. Digital 2020: Indonesia. [online] Tersedia di: WeAreSocial <<https://datareportal.com/digital-in-indonesia>> [Diakses 26 Januari 2021]
- LESTARI, J. 2016. Analisis sistem deteksi kerusakan komputer dengan menggunakan metode forward chaining. *Jurnal Inspiraton*, 6(1), pp. 19-27.
- LINE. 2016. Line announces plans for expanded development and proliferation of chatbots. [online] Tersedia di: <<https://linecorp.com/zh-hans/pr/news/en/2016/1517>> [Diakses 23 Januari 2020]
- SETIAWAN, D., 2018. Dampak perkembangan teknologi informasi dan komunikasi terhadap budaya. *JURNAL SIMBOLIKA: Research and Learning in Communication Study*. 4(62).
- SURYANI, D. dan AMALIA, E. 2017. Aplikasi chatbot objek wisata jawa timur berbasis aiml. *SMARTICS Journal*, 3(2), pp. 4-54.
- WIJAYA, M. H., SOTYOHADI, dan SETIAWAN, R. S. 2015. Perancangan chatbot pembelajaran pemrograman berorientasi object berbasis sistem modular. *SENATEK (Seminar Nasional Teknologi)*, 1(A), pp. 379-386.