

THE DEVELOPMENT OF *CHATBOT* APPLICATION ON LINE MESSAGING PLATFORM FOR CUSTOMER SERVICE IN JOGJA SEWA KAMERA

Dimas Pratama Jati¹, Muhammad Rifqi Ma'arif²

Program Studi D3 Sistem Informasi¹, Program Studi S1 Sistem Informasi²

Fakultas Teknik dan Teknologi Informasi

Universitas Jenderal Achmad Yani Yogyakarta

Jl. Siliwangi Km 0.7, Banyuraden, Gamping, Sleman

dimaspratamajati@gmail.com¹, rifqi@unjaya.ac.id²

Abstract

In recent days, customer care is one of the pivotal aspects in business environment. Quick and reliable response to any of customer question was highly correlated to customer satisfaction and obviously affect the conversion rate of any kind of products or services offered to them. Jogja Sewa Kamera is an small company which the main business is to rent a camera. That company serve no less than 25 booking request and questions about the product they offered to the customer. With the limitation of operational staff, the work for responding the customer booking request and question is a bit overwhelmed. Thus, an automated mechanism was built in this research to help the company to handle the customer service. In this research, a chatbot based on Line Messaging Platform was developed for automatic customer service. This chatbot can answer any of frequent or common question and booking request from the customer.

Keywords: Chatbot, Line Messaging Platform, Customer Service

1. Latar Belakang Masalah

Salah satu faktor kesuksesan dalam bisnis adalah kepuasan pelanggan[1]. Kecepatan dan keauratan dalam menjawab setiap pertanyaan dari calon pelanggan merupakan hal yang krusial dalam menentukan kepuasan pelanggan. Jogja Sewa Kamera merupakan sebuah perusahaan yang bergerak dibidang persewaan kamera yang terletak di jalan Affandi no 36, Gejayan. Jogja Sewa Kamera berdiri sejak Agustus 2017, dalam pengoperasiannya selain melayani sewa secara langsung Jogja Sewa Kamera juga melayani layanan *customer* dan sewa secara *booking* dengan cara menghubungi lewat chat baik WA, Line dll. Dalam sehari statistik chat *customer service* atau *booking* kamera mencapai kurang lebih 25 kali permintaan dan kebanyakan dilayani melalui aplikasi *chat (instant Messaging)*. Dari pertanyaan-pertanyaan yang diajukan oleh pelanggan, kebanyakan pertanyaan-pertanyaan tersebut sama dan berulang-ulang. Sehingga, untuk membantu admin Jogja Sewa Kamera dalam menjawab pertanyaan pelanggan dan melayani permintaan booking, dalam penelitian ini dikembangkan *chatbot* yang dapat melakukan pekerjaan berulang secara otomatis.

Chatbot merupakan sebuah program komputer yang dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada pengguna (manusia) melalui bentuk teks, suara, dan atau visual. Percakapan yang terjadi antara komputer dengan manusia merupakan bentuk respon dari program yang telah dideklarasikan pada database program pada komputer. Kemampuan komputer dalam menyimpan banyaknya data tanpa melupakan satu pun informasi yang disimpannya digabungkan dengan kemudahan bertanya pada sumber informasi langsung dibandingkan dengan mencari informasi sendiri, kemampuan tersebut menyebabkan *chatbot* adalah *customer service* yang handal, hal tersebut yang membuat *chatbot* akan sangat berguna dalam pengoperasian persewaan di Jogja Sewa Kamera ini dan di harapkan dapat meningkatkan kualitas pelayanan di Jogja Sewa Kamera.

2. Metode Penelitian

2.1 Tahapan Pengembangan

Penelitian ini dilaksanakan dengan prinsip *Software Development Lifecycle* (SDLC) menggunakan pendekatan *waterfall*. Adapun tahapan penelitian yang dilakukan dalam pelaksanaan penelitian ini adalah sebagai berikut:

1. Pengumpulan data, tahap pengumpulan data ini penulis mencari dan mengumpulkan semua informasi yang di butuhkan dalam pengembangan aplikasi *Chatbot* ini. kebutuhan tersebut adalah pengetahuan pertanyaan yang sering diajukan oleh customer.
2. Analisa data, pada tahap ini penulis melakukan analisa dari data yang telah terkumpul, sebagai acuan untuk pengembangan sistem.
3. Perancangan, pada tahap ini, penulis melakukan perancangan terhadap aplikasi yang akan dibangun Terdapat dua rancangan, yakni rancangan alur sistem dengan flowchart dan rancangan antarmuka (*user interface*).
4. Implementasi dan pengujian, tahap implementasi merupakan tahap konstruksi aplikasi dengan bahas pemrograman. Aplikasi *chatbot* ini dikembangkan diatas *LINE Messaging Platform* dengan bahasa pemrograman PHP menggunakan framework SLIM. Pengujian terhadap aplikasi dilakukan dengan melakukan uji fungsionalitas pada setiap fitur yang dikembangkan.

2.2 Landasan Teori

2.2.1 Chatbot

Chatbot (juga dikenal sebagai talkbot, chatterbot, bot) merupakan software yang dirancang khusus untuk mensimulasikan percakapan intelektual dengan satu atau lebih manusia baik secara audio maupun . Penggunaan *chatbot* secara spesifik antara lain *weather bot* untuk mengetahui cuaca terkini. Penggunaan lain adalah aplikasi semacam, *grocery bot* untuk membantu memilih dan memesan bahan makanan, *news bot* untuk memberikan informasi paling update, *life advice bot* untuk membantu memberikan solusi atas suatu masalah, *personal finance bot* untuk membantu manage keuangan, *scheduling bot* untuk membantu menyusun jadwal kegiatan sehari-hari, dll[2].

Cara kerja *chatbot* ini adalah dengan mengirim permintaan dalam bentuk audio atau teks lalu mereka akan mengirim kembali (merespons) sesuai kebutuhan yang kita inginkan. 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 memberikan respon kepada perintah yang spesifik dan tingkat kecanggihannya disesuaikan dengan programnya. *Chatbot* yang menggunakan *machine learning* juga sering disebut sebagai smart bot yang dapat menganggapi permintaan dalam bahasa natural yang bebas, tidak sebatas perintah yang sudah ditentukan polanya[3].

2.2.2 Platform Chatbot berbasis Line

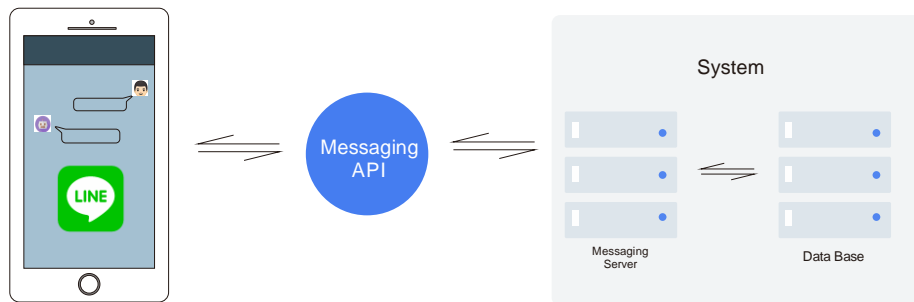
Line adalah sebuah aplikasi pengirim pesan instan gratis yang dapat digunakan pada berbagai platform seperti telepon cerdas, tablet, dan komputer. LINE difungsikan dengan menggunakan jaringan internet sehingga pengguna LINE dapat melakukan aktivitas seperti mengirim pesan teks, mengirim gambar, video, pesan suara, dan lain lain. LINE diklaim sebagai aplikasi pengirim pesan instan terlaris di 42 negara[4].

Untuk mendukung pengembangan *chatbot*, LINE tidak hanya menyediakan *Messaging API* lengkap dengan dokumentasinya, melainkan juga memberikan SDK untuk beberapa bahasa pemrograman di akun GitHub-nya. Diantaranya adalah Java, PHP, Ruby, Go, Perl dan Python. SDK atau LINE Software Development Kit adalah library atau package yang dapat digunakan oleh developer untuk menggunakan *Messaging API*. Dengan menggunakan SDK, developer dapat langsung menerapkan method-method terkait API tanpa harus membuat dari nol. Selain itu ketika ada pembaharuan API dari LINE, developer hanya perlu memperbaharui SDK sehingga meminimalisir perubahan basis kode pada aplikasi.

API Messaging memungkinkan data dilewatkan antara server aplikasi bot dan Platform LINE. LINE platform akan mengirimkan informasi ke server kita setiap kali ada interaksi antara pengguna dengan akun kita, secara tidak langsung *Messaging API* ini sebagai jembatan untuk LINE platform dengan aplikasi *chatbot*. Ada dua jenis interaksi, yakni:

1. Pengguna mengirimkan pesan
2. Pengguna melakukan sebuah aksi, 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. Setiap *request* menyertakan kode access token yang digunakan untuk memverifikasi bahwa *request* yang diterima adalah benar-benar dari platform LINE. Jika access token tidak dikenali, maka *request* dianggap tidak valid. Secara umum, arsitektur dan alur kerja aplikasi *chatbot* pada platform Line, ditunjukkan oleh Gambar 1.



Gambar 1. Arsitektur dan alur kerja *chatbot* berbasis

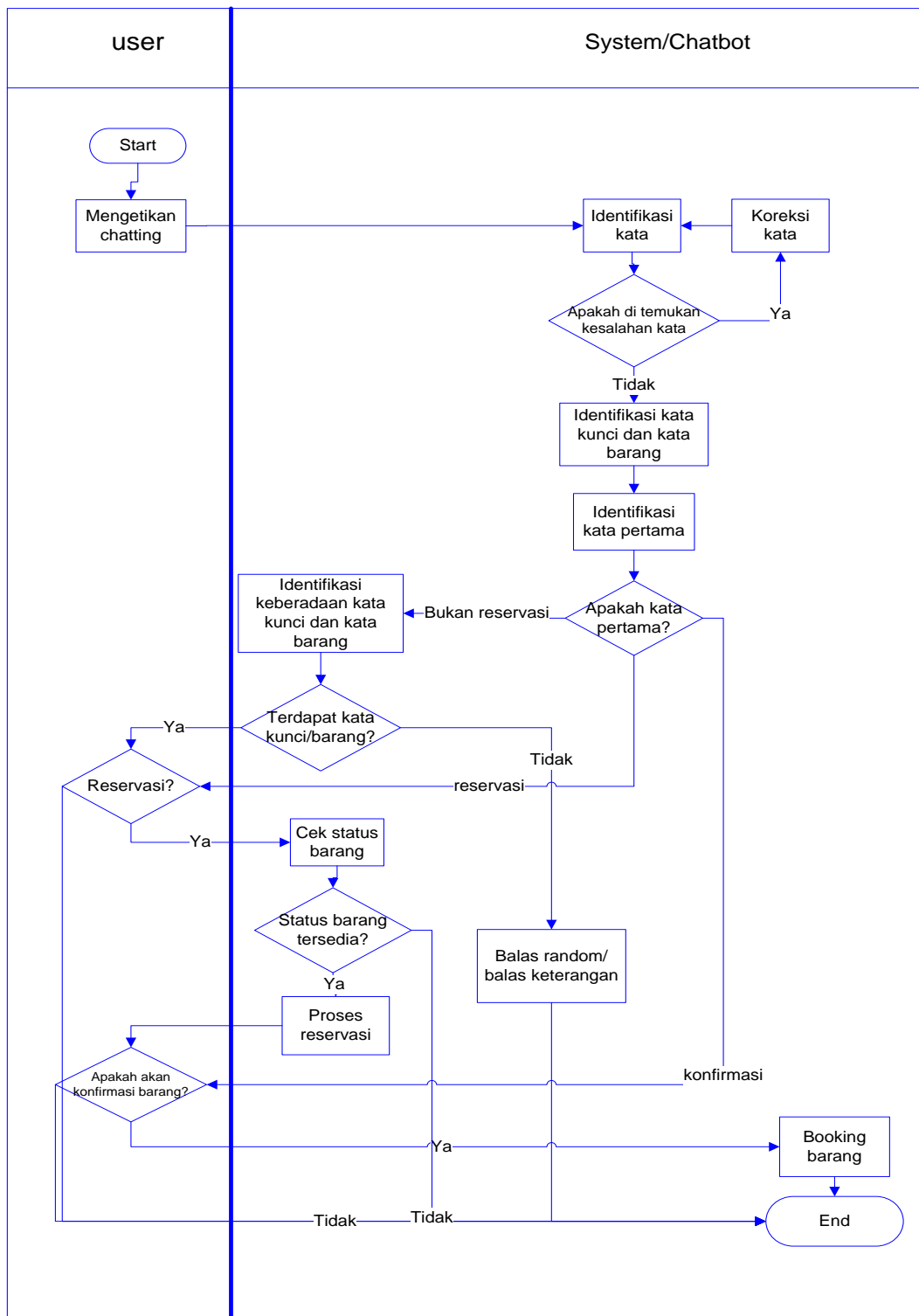
2.2.3 Pengembangan *Chatbot Line* dengan PHP

Seperti disebutkan sebelumnya pengembangan *chatbot* di aplikasi Line didukung SDK dengan beberapa bahasa salah satunya PHP, PHP merupakan singkatan dari PHP Hypertext Preprocessor, yang merupakan bahasa berbentuk skrip yang ditempatkan dalam server dan diproses di server[5]. Layanan diimplementasikan kedalam *web service* yaitu suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada system lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan oleh suatu sistem yang menyediakan web service.

Dalam pengembangan *chatbot* ini menggunakan MySQL sebagai databasenya, MySQL banyak dipakai untuk kepentingan penanganan *database* karena selain handal juga bersifat *open source* [6]. Sebagai basis untuk mengatur routing bisa memanfaatkan SLIM, yaitu microframework PHP untuk membuat aplikasi web atau server, dikatakan microframework karena SLIM adalah framework yang fokus pada kebutuhan pokok yang diperlukan sebuah aplikasi web seperti: menerima sebuah HTTP *request*, mengirimkan *request* tersebut ke code yang sesuai, dan mengembalikan HTTP response[7].

2.3. Rancangan Alur Aplikasi *Chatbot*

Rancangan proses bisnis aplikasi *chatbot* yang dikembangkan dalam penelitian ini, diurikan dengan menggunakan flowchart pada Gambar 2. Dalam flowchart tersebut terdapat dua *actor* yakni *user* adalah pelanggan yang chatting baik bertanya maupun *booking* dan *system/chatbot* adalah aplikasi yang dibuat. Alur kerja tersebut diimplementasikan kedalam tiga jenis antarmuka yakni antarmuka *text message*, antar muka *button message*, dan antarmuka *text confirmation*.



Gambar 2. Alur aplikasi *chatbot*

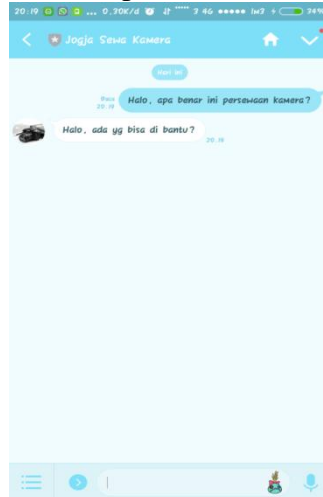
3. Pengujian dan Pembahasan

3.1. Implementasi Antarmuka Aplikasi *Chatbot*

Berdasarkan alur aplikasi *chatbot* yang diuraikan pada Gambar 2, maka antarmuka aplikasi *chatbot* terbagi kedalam 3 bagian yakni antarmuka *text message*, antarmuka *button message*, dan antarmuka *confirm message*. Antarmuka *text message* merupakan antarmuka pertama yang mengawali interaksi antara pengunjung dengan *chatbot*. Ketika ada pesan masuk dari pengunjung

maka *chatbot* akan melakukan tokenisasi dengan memecah kalimat pesan tersebut kedalam kata, misal ketika pelanggan mengirim pesan “Halo, apa benar ini persewaan kamera?”.

- Dari kalimat tersebut akan diurai kedalam bentuk array menjadi setiap kata “halo”, “apa”, “benar”, “ini”, “persewaan”, “kamera”.
- Dari setiap kata tersebut akan dicek pada database apakah ada kata kunci atau produk didalamnya, kata kunci dan daftar produk sudah didefinisikan sebelumnya dan terdapat dalam database, untuk contoh ini karena tidak ada kata kunci maupun produk yang ditemukan dalam database maka akan diberikan jawaban random berupa *Text Message* yang sudah ada dalam database, bisa dilihat pada Gambar 3 berikut.



Gambar 3. Implementasi antarmuka text message

Antarmuka yang kedua adalah *button message*. *Button Message* akan tampil (*triggered*) ketika pesan yang dikirimkan oleh pengunjung mengandung kata kunci tertentu dan/atau nama produk yang ada di dalam basis data. Sebagai contoh, ketika pelanggan mengirim pesan “Apakah canon 600D bisa saya pesan untuk hari ini?”,

- Dari kalimat tersebut akan diurai kedalam bentuk array menjadi setiap kata “apakah”, “canon”, “600D”, “bisa”, “saya”, “pesan”, “untuk”, “hari”, “ini”.
- Dari contoh diatas terdapat kata kunci “pesan” dan produk “600D”, maka *chatbot* akan menanggapi dengan memberikan pesan berupa *Button Message Builder*.
- Jika ditemukan kata kunci dan juga produk maka *chatbot* akan menawarkan produk yang dimaksud, dilengkapi dengan button untuk reservasi seperti Gambar 4 berikut.



Gambar 4. Implementasi button message dengan reservasi

- Jika hanya ditemukan produk maka *chatbot* akan memberikan tanggapan berupa *Button Message Builder* juga, namun tidak disertai dengan *button* untuk reservasi, bisa dilihat pada Gambar 5 berikut.



Gambar 5. Implementasi button message tanpa reservasi

Jenis antarmuka yang terakhir adalah *confirm message*. Antarmuka ini digunakan untuk memberikan pesan balasan (konfirmasi) melalui 2 *button* yang terdiri dari ya dan tidak. Antarmuka *confirm message* akan muncul (*triggered*) jika pelanggan menekan *button* reservasi pada antarmuka *button message*, Gambar 6 menunjukkan implementasi dari antarmuka *confirm message*.



Gambar 6. Implementasi confirm button

3.2. Pengujian

Secara umum, teknis alur penggunaan *chatbot* ini berawal dari antarmuka text message dimana pengguna/pengunjung memasukkan pertanyaan yang kemudian akan direspon oleh *chatbot* secara otomatis pada antarmuka yang sama. Selanjutnya, apabila ada request untuk menampilkan

atau reservasi produk, maka *chatbot* akan menampilkan pilihan produk yang tersedia melalui *button interface* yang dapat diklik oleh pengunjung. Terakhir, proses interaksi akan diakhiri dengan *chatbot* menampilkan antarmuka *confirm message* kepada pengunjung.

Tabel 1. Hasil pengujian fungsionalitas aplikasi *chatbot*.

No	Poin Pengujian	Hasil Uji
1.	Aplikasi <i>chatbot</i> dapat menampilkan respon atas pesan tersrtuktur yang dikirimkan oleh pengguna dengan jawaban yang sudah ditentukan.	✓
2.	Aplikasi <i>chatbot</i> dapat merespon pesan random yang dikirimkan oleh pengguna dengan menampilkan pesan promosional dan daftar produk	✓
3.	Aplikasi <i>chatbot</i> dapat menampilkan daftar produk dan tombol “reservasi” dan “buka/baca” disetiap produk	✓
4.	Aplikasi <i>chatbot</i> dapat merekam pesanan pengguna setelah pengguna mengkilik tombol “ya” pada <i>confirm interface</i> .	✓

Keterangan

✓ = sukses/ berhasil

Tabel 1 menunjukkan hasil pengujian fungsionalitas dari aplikasi *chatbot*. Sesuai dengan tahapan pengembangan yang direncanakan, dan pengujian yang dilakukan pada aplikasi *chatbot* ini adalah pengujian fungsionalitas (*functionality testing*). Pengujian fungsionalitas ini dimaksudkan untuk memastikan bahwa fitur-fitur aplikasi *chatbot* yang dikembangkan berfungsi sebagaimana mestinya sesuai dengan yang direncanakan.

4. Kesimpulan

Dari antarmuka-antarmuka yang sudah diimplementasikan, aplikasi *chatbot* yang dibangun untuk Jogja Sewa Kamera ini dapat digunakan untuk membantu pemilik Jogja Sewa Kamera dalam melayani pertanyaan-pertanyaan pelanggan mengenai produk, terutama pertanyaan-pertanyaan yang sering muncul (*Frequently Asked Question*). Dengan demikian, implementasi aplikasi *chatbot* ini dapat membantu pengelolaan bisnis Jogja Sewa Kamera khususnya dalam hal pelayanan pelanggan (*customer service*). Pemilik usaha tidak perlu menjawab pertanyaan berulang-ulang dan pelanggan juga tidak perlu datang langsung untuk reservasi/booking.

Lebih lanjut, hasil pengujian menunjukkan bahwa semua fungsionalitas utama dari aplikasi *chatbot* dapat berjalan dengan baik sesuai dengan yang direncanakan. Sehingga aplikasi *chatbot* ini sudah siap diimplementasikan pada kondisi real. Meskipun demikian, masih terdapat beberapa kekurangan dalam aplikasi *chatbot* ini yang bisa menjadi bahan pengembangan di masa yang akan datang (*future work*). Keterbatasan aplikasi *chatbot* ini diantaranya masih terpaku pada pola pertanyaan dan jawaban yang sudah terdaftar di basis data sehingga interaksi dengan pengguna hanya terbatas pada pola kosa kata (*dictionary pattern*) yang sudah ditentukan. Implementasi pemrosesan bahasa alami bisa menjadi pengembangan yang sangat menjanjikan untuk fleksibilitas interaksi *chatbot* dengan pengguna di masa yang akan datang.

Daftar Pustaka

- [1] Hidayat, R. (2014). Sistem Informasi Ekspedisi Barang Dengan Metode E-CRM Untuk Meningkatkan Pelayanan Pelanggan. *Jurnal Sisfotek Global*, 4(2).
- [2] Budiharto, W., & Suhartono, D. (2014). Artificial Intelligence: Konsep dan Penerapannya. Yogyakarta: Andi.

- [3] Hadi, W. M., Hadi, S., & Syarif, S. R. (2017). Perancangan *Chatbot* Pembelajaran Pemrograman Berorientasi Object Berbasis Sistem Modular. *Prosiding SENATEK 2015*, 1(A), 379-386.
- [4] Chen, H. H. (2016, June). Platform strategies perspective on the OTT messaging services: A case study of WeChat and Line. In *annual conference of International Telecommunication Society, Taipei, Taiwan*.
- [5] Solichin, A. (2016). *Pemrograman web dengan PHP dan MySQL*. Penerbit Budi Luhur.
- [6] Kadir, A. (2010). Mudah Mempelajari Database MySQL. *Yogyakarta: Andi*.
- [7] Noviantha, F. D. (2017). *Mengenal Slim Framework, Microframework Berbasis PHP*. [Online] Available at: <https://www.codepolitan.com/mengenal-slim-framework-589f4003286cb> [Accessed 17 2018].