Terbit *online* pada laman web jurnal :
http://ejournal.amikompurwokerto.ac.id/index.php/telematika/

Telematika

Terakreditasi Sinta "3" KEMENRISTEKDIKTI, No. 21/E/KPT/2018

UNIVERSITAS AMIKOM PURWOKERTO

Desain dan Implementasi Penandatangan Elektronik Sertifikat X509 Menggunakan Platform Bot Telegram

Herman Kabetta

Program Studi Rekayasa Kriptografi
Sekolah Tinggi Sandi Negara
Email : herman.kabetta@stsn-nci.ac.id

INFO ARTIKEL

Sejarah artikel:
Menerima 6 Desember, 2019
Revisi 29 Januari, 2020
Diterima 24 Januari 24, 2020
Online 28 Februari 2020

***Keyword*:**
Digital Signature,
Digital Certificate,
X509 Certificate,
Chatbot,
Rapid Application Development,
System Usability Scale

ABSTRACT

Balai Sertifikasi Elektronik (BSrE) as one of the Certificate Authorities in Indonesia has been released a desktop-based digital signing application, but one of the weaknesses of desktop applications is its low portability. BSrE has also released a digital signing application for mobile operating systems, but it is only intended for users of the Android operating system. The aim of this research is to develop a digital signing application with Telegram Bot platform, that can be used to sign electronic documents using X509 certificates wherever and whenever, and also it can be run on all operating system platforms. The research methodology is using Rapid Application Development (RAD) which consists of four stages, Requirements Planning, User Design, Construction and Cutover. The backend system of the bot is built using Java programming language, and integrated with the MySQL database as conversation sessions storage. There are three main functions of the designed system, sign, verify and setting. Signed documents also have been tested in several pdf reader applications and digital signatures can be recognized and validated. Bot can also verify documents signed by other applications. Testing uses a black-box method, the results of functional testing and non-functional testing show the system can run properly as expected in requirements planning. Evaluation using System Usability Scale (SUS) indicate that the system is suitable for use.

Kata Kunci*:*
Tanda Tangan Elektronik,
Sertifikat Elektronik,
Sertifikat X509,
Chatbot,
Rapid Application Development,
System Usability Scale

ABSTRAK

Balai Sertifikasi Elektronik (BSrE) sebagai salah satu *Certificate Authority* di Indonesia telah merilis aplikasi penandatangan elektronik berbasis desktop kepada publik, namun salah satu kekurangan aplikasi desktop adalah rendahnya portabilitas dalam penggunaan. BSrE juga telah merilis aplikasi penandatangan elektronik untuk sistem operasi mobile, namun hanya diperuntukkan bagi pengguna sistem operasi Android. Penelitian ini bertujuan mengembangkan sebuah Bot Telegram yang dapat digunakan untuk menandatangani dokumen elektronik menggunakan sertifikat X509 dimanapun dan kapanpun, serta dapat berjalan pada semua platform sistem operasi. Metode yang digunakan dalam penelitian ini adalah Rapid Application Development (RAD) yang terdiri dari empat tahap, Requirements Planning, User Design, Construction dan Cutover. Sistem backend bot dibangun dengan menggunakan bahasa pemrograman Java yang terintegrasi dengan basis data MySQL untuk menyimpan sesi percakapan. Penelitian menghasilkan sebuah sistem Bot Telegram yang memiliki tiga fungsi utama, yakni tanda tangan, verifikasi dan pengaturan. Dokumen yang ditandatangani telah diuji pada beberapa aplikasi pembaca berkas pdf dan tanda tangan elektronik dapat dikenali dan divalidasi. Bot juga dapat memverifikasi dokumen yang ditandatangani oleh aplikasi lain. Hasil pengujian terhadap komponen fungsional dan non-fungsional dengan metode black-box testing menunjukkan sistem dapat berjalan dengan baik sesuai yang diharapkan pada requirements planning. Hasil evaluasi kelayakan menggunakan System Usability Scale (SUS) menunjukkan sistem berada dalam kategori baik dan layak untuk digunakan.

Korespondensi:
Telepon: +62 (813) 94616622
E-mail:
hermanka.beta@gmail.com

PENDAHULUAN

Menurut Undang-Undang Informasi dan Transaksi Elektronik (UU ITE), tanda tangan elektronik adalah tanda tangan yang terdiri atas informasi elektronik yang dilekatkan, terasosiasi atau terkait dengan informasi elektronik lainnya yang digunakan sebagai alat verifikasi dan autentikasi, dengan kata lain, tanda tangan elektronik dapat digunakan sebagai jaminan keabsahan dan keaslian sebuah dokumen elektronik (Dhagat, 2016) (Pereira, 2018). Pada proses penandatangan dokumen elektronik, pengguna membutuhkan sebuah sertifikat elektronik yang diterbitkan oleh *Certificate Authority* (CA). Peran CA tidak hanya menerbitkan, namun juga melakukan verifikasi terhadap sertifikat elektronik. Balai Sertifikasi Elektronik (BSrE) merupakan salah satu CA di Indonesia. BSrE adalah salah satu unit pelaksana teknis BSSN (Badan Siber dan Sandi Negara) yang bertugas dalam pemberian pelayanan penerbitan dan pengelolaan sertifikat elektronik kepada publik (Yusandy, 2019). BSrE telah merilis aplikasi penandatangan elektronik berbasis *desktop* kepada publik, namun salah satu kekurangan dari aplikasi berbasis *desktop* adalah rendahnya portabilitas (Singh, 2017). Pada beberapa wawancara yang telah dilakukan, pengguna sertifikat elektronik membutuhkan sebuah aplikasi penandatangan yang dapat dijalankan dimanapun berada, kapanpun dibutuhkan dan dapat berjalan pada semua platform sistem operasi.

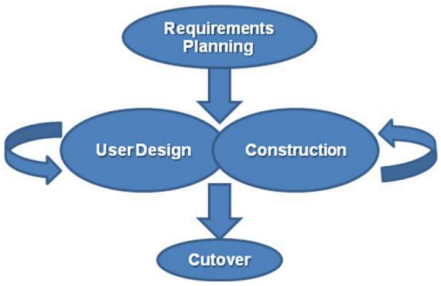
Salah satu solusi tercepat untuk membangun sebuah aplikasi *multiplatform* adalah dengan memanfaatkan fasilitas *chatbot* yang disediakan oleh aplikasi perpesanan (Dubosson, 2017). Penelitian ini akan menggunakan fasilitas *Chatbot* dari aplikasi Telegram. Istilah *Chatbot* sendiri merujuk pada *Bot* yang khusus berjalan pada aplikasi perpesanan yang dapat diintegrasikan dengan sistem-sistem eksternal (Korotaeva, 2018). Penelitian mengenai penggunaan *chatbot* Telegram telah banyak dilakukan, beberapa di antaranya pada bidang pelayanan (Rosid, 2018), pusat informasi (Calvo, 2017), hingga pengawasan (Husni, 2018). Pada penelitian yang dilakukan oleh Rosid (2018), peneliti mencoba meng-integrasikan *chatbot* Telegram dengan aplikasi *e-complaint* berbasis web yang sudah ada. Tujuan dari penelitian ini adalah untuk semakin mempermudah pengguna dalam menyampaikan keluhan. Pengguna tidak perlu membuka aplikasi *e-complaint* melalui *web browser*, namun cukup dengan berkomunikasi menggunakan aplikasi Telegram. Selain menangani keluhan, *chatbot* juga diatur dapat memberikan informasi seputar kampus. Pada penelitian Calvo (2017), dibangun sebuah *chatbot* bimbingan karir. Ide penelitiannya adalah menyediakan antarmuka yang mudah dan ramah bagi pengguna, di samping tujuan utamanya sebagai pengumpul data yang diperlukan untuk memberikan panduan karir. Dari kuesioner yang disebar diperoleh hasil responden sangat setuju dengan sistem *chatbot* tersebut, di samping mudah digunakan, responden juga berkomentar bahwa dengan menggunakan *chatbot*, sistem menjadi lebih mudah untuk dipelajari. Selain beberapa penelitian di atas, *Chatbot* juga dapat digunakan sebagai sistem pengawasan. Husni (2018) membangun sebuah *chatbot* yang mampu mengawasi dan memantau keberadaan pengemudi hingga perilaku pengemudi di jalan. Sistem berfungsi sebagai pengawas yang memberi tahu administrator rental mobil ketika pengemudi melakukan kesalahan berdasarkan aturan yang diinginkan dan memberi tahu pengemudi apa yang harus dilakukan.

Tujuan dari penelitian ini adalah merancang dan membangun sebuah aplikasi penandatangan dokumen elektronik yang bersifat *multiplatform* menggunakan *chatbot*, serta melakukan evaluasi kelayakan terhadap penggunaannya. Sertifikat yang digunakan dalam proses penandatanganan adalah standar sertifikat X509 dalam format PKCS#12. Sertifikat X509 telah banyak digunakan pada beberapa penelitian, diantaranya Forsby (2017) dan Karthikeyan (2019) yang menggunakan sertifikat X509 untuk

mengamankan perangkat-perangkat IoT. Sertifikat X509 sendiri merupakan salah satu standar sertifikat elektronik paling penting yang banyak digunakan pada beberapa mekanisme autentikasi (Vukasović, 2017).

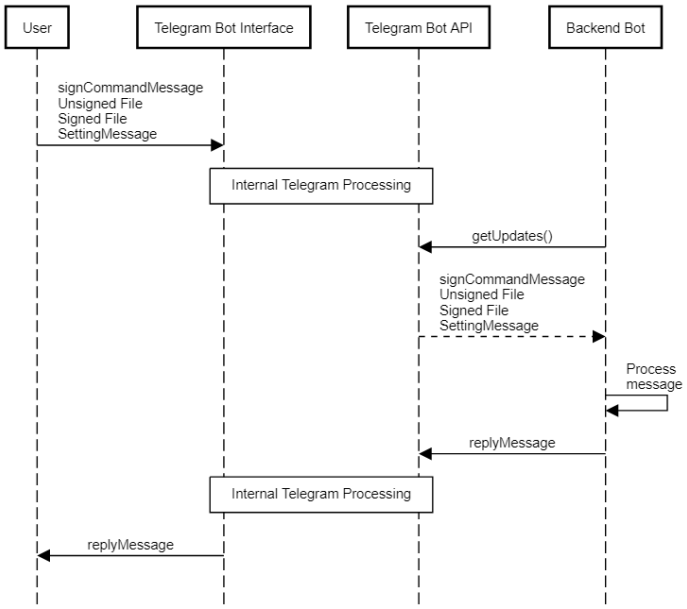
METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah *Rapid Application Development* (RAD). *Rapid Application Development* (RAD) *Life Cycle* pertama kali diperkenalkan oleh James Martin pada awal tahun 1990 (Kneuper, 2018). Menurut James Martin (Hassan, 2015) (Setyatama, 2018), *Rapid Application Development* (RAD) terdiri dari empat tahap yaitu *Requirements Planning, User Design, Construction dan Cutover*.



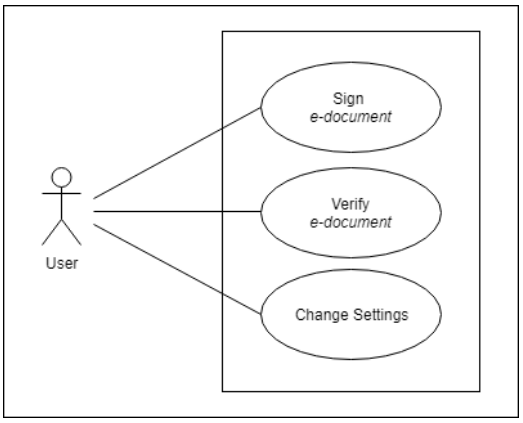
Gambar 1. Model *Rapid Application Development* (RAD) James Martin (Martin, 1991)

Tahapan penelitian pada Gambar 1. dapat dijelaskan sebagai berikut: 1) *Requirements Planning*, merupakan tahap pertama dari model RAD James Martin. Pada tahap ini dilakukan perencanaan berdasarkan kebutuhan yang diinginkan oleh pengguna serta analisis terhadap permasalahan. Data dikumpulkan melalui proses wawancara dan kuesioner yang melibatkan pengguna layanan dari BSrE khususnya para pemegang sertifikat elektronik. Berdasarkan hasil kuesioner dan wawancara, pengguna menginginkan sebuah aplikasi penandatangan dan pemverifikasi dokumen elektronik dengan mobilitas dan portabilitas yang tinggi, *user friendly*, serta dapat berjalan pada semua platform sistem operasi. Sebagai pemecahan masalah, maka akan dikembangkan sebuah *chatbot* Telegram yang dapat menandatangani dan memverifikasi dokumen elektronik.



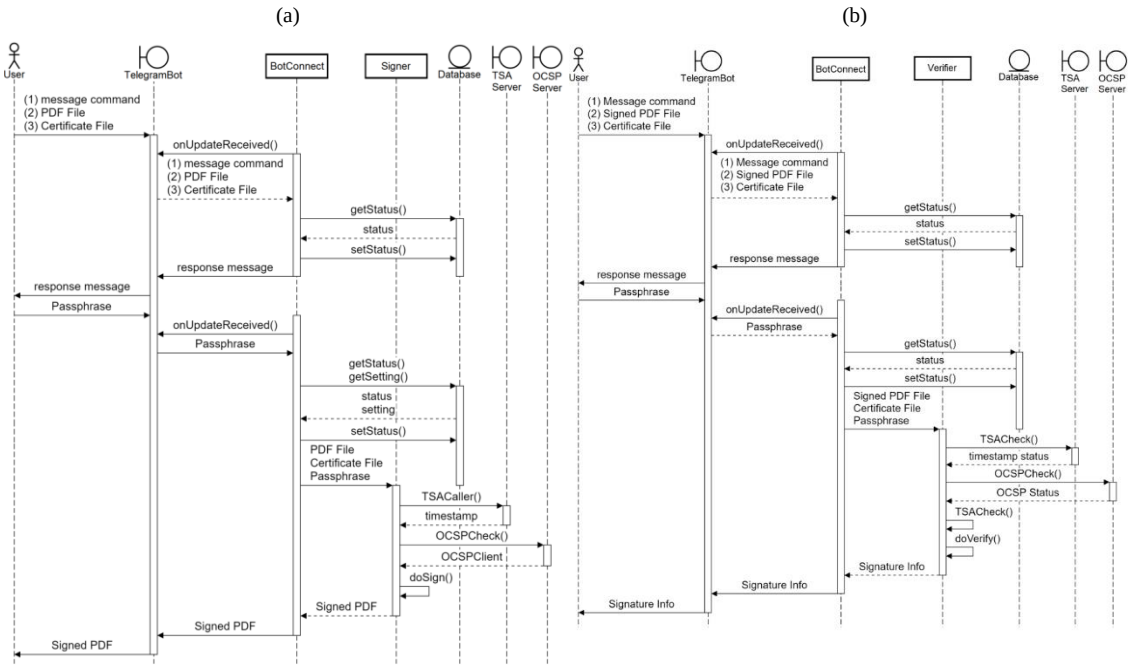
Gambar 2. Gambaran umum sistem

Gambaran umum sistem *chatbot* dapat dilihat pada Gambar 2 yang menunjukkan interaksi antara *user* dengan *chatbot*. *Chatbot* sendiri terdiri dari tiga entitas yang tidak dapat dipisahkan yakni *Telegram Bot Interface*, *Telegram Bot API* dan *Backend Bot*. Telegram Bot *Interface* berperan sebagai antarmuka user yang dalam hal ini adalah aplikasi Telegram itu sendiri, sedangkan Telegram Bot API berperan sebagai penghubung antara program *backend* dengan Telegram Bot *Interface*. Telegram Bot API berada di server Telegram, sedangkan *backend* terpasang di server peneliti. Program *backend* digunakan sebagai pemroses pesan, proses penandatangan dan verifikasi dokumen berjalan pada program *backend*. User mengirimkan pesan melalui Bot Telegram *Interface* yang dapat berupa pesan perintah atau pesan lampiran dokumen elektronik. Pesan-pesan yang diterima oleh antarmuka *chatbot* kemudian akan diproses secara internal untuk selanjutnya diteruskan ke Telegram Bot API. Secara berkala program *backend* akan meminta *update* pesan dari Telegram Bot API untuk kemudian diproses sesuai isi pesan yang masuk didalam program *backend*. Perlu diperhatikan bahwa pengiriman pesan dilakukan satu-persatu, alur pengiriman pesan sama seperti proses *chatting*. Alur dan urutan pengiriman secara detail dijabarkan pada rancangan *sequence diagram* pada tahap RAD selanjutnya.



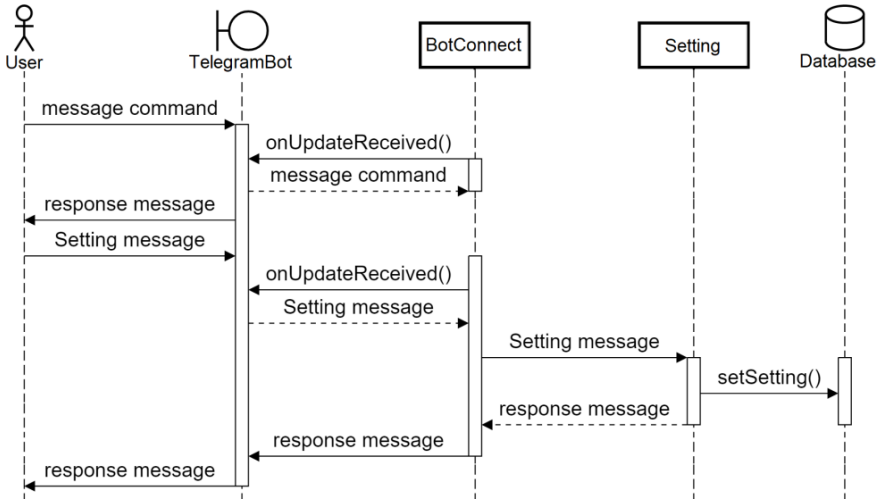
Gambar 3. *Use case diagram*

Tahapan kedua dari metode RAD yakni 2) *User Design*, tahap ini memastikan desain sistem sesuai dengan kebutuhan pengguna berdasarkan data pada tahap sebelumnya. Pada tahap ini ditentukan *input*, proses dan *output* yang diinginkan beserta layanan-layanan yang nantinya akan didukung oleh *chatbot*. Desain sistem ditetapkan menggunakan model UML. Diagram yang digunakan antara lain *Use Case Diagram*, *Sequence Diagram*, *Class Diagram* dan *Deployment Diagram*. Gambar 3 merupakan rancangan *use case* yang telah dibuat. Desain sistem memiliki tiga fungsi yaitu fungsi untuk menandatangani dokumen, memverifikasi dokumen, dan mengubah pengaturan khususnya tata letak tanda tangan elektronik. Rancangan skenario untuk setiap *use case* dituangkan dalam tiga *sequence diagram* di Gambar 4(a), Gambar 4(b) dan Gambar 5.



Gambar 4. *Sequence diagram* (a) skenario penandatanganan; (b) skenario verifikasi

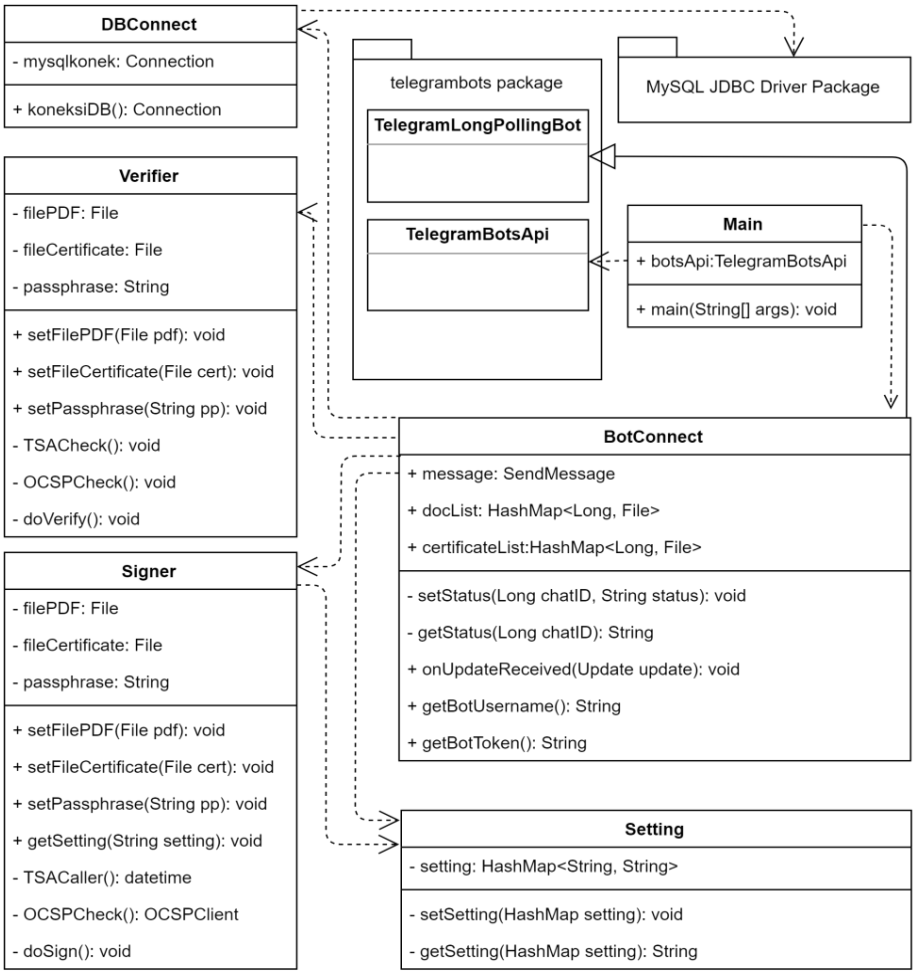
Berdasarkan rancangan *sequence diagram* (Gambar 4(a), 4(b) dan 5), dapat kita amati entitas-entitas yang terlibat antara lain, *User* sebagai aktor pengguna sistem, TelegramBot yang merupakan sistem internal Telegram, yang mencakup aplikasi Telegram, Bot Telegram dan Telegram Bot API. Aplikasi *backend* diwakilkan oleh dua entitas yakni BotConnect dan satu entitas untuk masing-masing skenario yaitu *Signer* untuk skenario penandatanganan, *Verifier* untuk skenario verifikasi dan *Setting* untuk skenario pengaturan letak posisi tanda tangan elektronik. *Database* sebagai entitas penyimpanan data sesi dan status *chat*. Entitas lain pada skenario penandatangan dan verifikasi yaitu TSAServer dan OCSPServer yang merupakan entitas eksternal yang dikelola oleh *Certificate Authority*. TSAServer dan OCSPServer berfungsi sebagai entitas yang memiliki otoritas untuk memeriksa validitas sertifikat elektronik dan *timestamp* pada tanda tangan elektronik.



Gambar 5. *Sequence diagram* skenario pengaturan

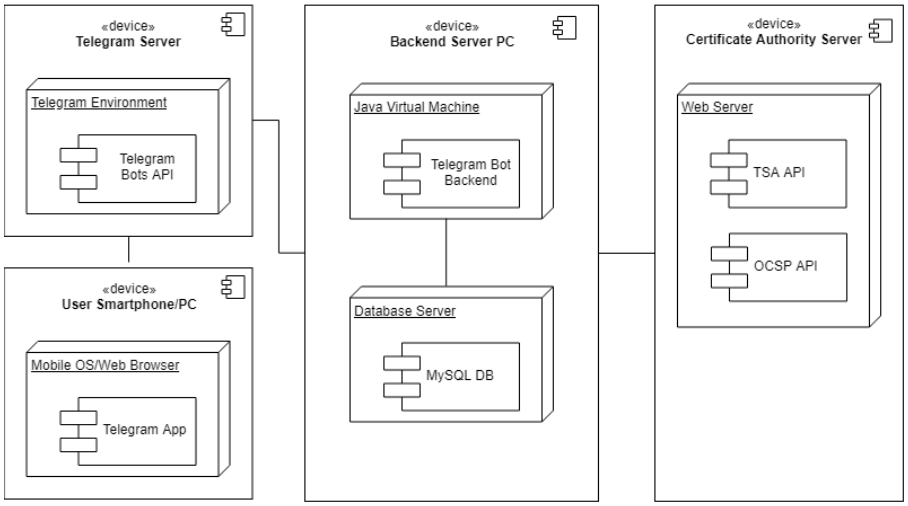
Aplikasi *backend chatbot* akan dikembangkan menggunakan bahasa pemrograman Java. Rancangan *class* untuk aplikasi *backend* dapat dilihat pada Gambar 6. Terdapat enam *class* yang terlibat, tiga class di antaranya yakni *Signer*, *Verifier* dan *Setting* berhubungan dengan skenario pada diagram *use case*. Selain

class internal, aplikasi *backend* juga membutuhkan *package* eksternal diantaranya MySQL JDBC *Driver* yang berfungsi sebagai *driver* koneksi dari aplikasi ke basis data, dan telegrambots yang merupakan *library* khusus untuk berkomunikasi dengan Telegram Bot API.



Gambar 6. *Class diagram*

Gambar 7. adalah rancangan *deployment*. *Deployment diagram* digunakan sebagai gambaran arsitektur sistem secara keseluruhan hingga lapisan perangkat keras yang digunakan. Dapat dilihat pada *deployment diagram*, peneliti hanya menyiapkan satu buah komputer *server* yang bertindak sebagai *backend*, dan sebuah *smartphone* untuk pengujian. Sedangkan server lain merupakan *server* eksternal dari pihak Telegram maupun dari pihak *Certificate Authority*.



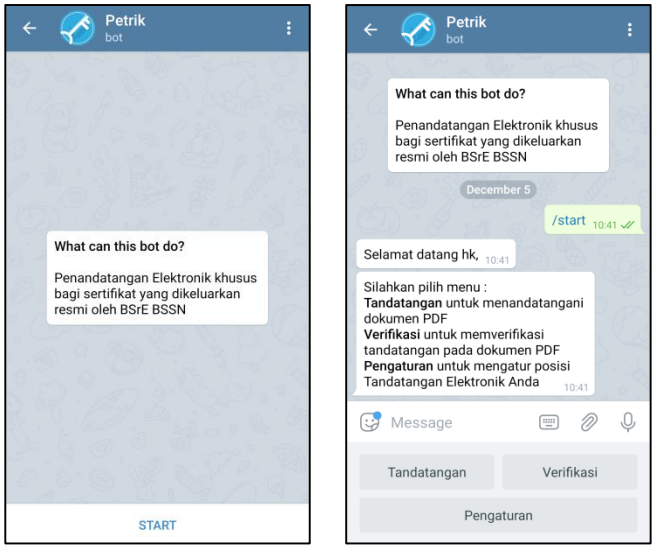
Gambar 7. *Deployment diagram*

Desain sistem yang telah dirancang pada tahap sebelumnya kemudian diimplementasikan menjadi sebuah sistem utuh, tahap ini pada metode RAD disebut dengan 3) *Construction*. Langkah pertama yang dilakukan adalah mendaftarkan *chatbot* melalui akun BotFather di Telegram. Selanjutnya, menuliskan kode untuk sistem *backend* menggunakan bahasa pemrograman Java. Koneksi *backend* dengan Telegram Bot API dilakukan menggunakan metode *long-polling*. Metode *long polling* adalah metode *default* yang digunakan untuk interaksi antara *backend* dengan Telegram API (Sucipto, 2019). Keuntungan penggunaan metode *long-polling* adalah tidak diperlukan sebuah *web server*, sedangkan kelemahannya adalah metode ini tidak berjalan secara *realtime*. Implementasi dan pengujian dilakukan berulang secara interaktif pada tahap *construction*. Tahap terakhir pada metode *Rapid Application Development* yaitu 4) *Cutover*, tahap ini terdiri dari proses *maintenance* dan *deployment*. Pada tahap *cutover* dilakukan pengujian akhir setelah *chatbot* melalui proses *deployment*. Pengujian fungsional dan non-fungsional dilakukan dengan metode *black-box* terhadap sistem *chatbot*. Evaluasi sistem menggunakan *System Usability Scale* (SUS) juga dilakukan untuk mengetahui tingkat kelayakan dari sistem yang akan dibangun.

HASIL DAN PEMBAHASAN

Berdasarkan rancangan sistem, terdapat tiga proses yang akan diakomodasi oleh sistem yaitu proses tanda tangan, verifikasi dokumen, dan pengaturan letak posisi tanda tangan. Desain antarmuka *chatbot* tidak terlalu rumit karena antarmuka telah diakomodasi secara penuh oleh pihak Telegram. Pengembang hanya perlu fokus pada pesan-pesan balasan untuk setiap tahap proses yang berlangsung. Gambar 8(a) merupakan tampilan awal *chatbot* ketika pertama kali diakses atau ditambahkan pada *chatroom* Telegram. Pada tampilan awal ini, pengguna akan diberikan informasi sekilas mengenai kegunaan *chatbot* dan disediakan tombol "start" untuk memulai obrolan dengan *chatbot*. Gambar 8.(b) adalah tampilan ketika pengguna telah menekan tombol "start", kemudian muncul pesan balasan dari *chatbot* disertai pilihan menu yang disediakan di area *keyboard* yaitu "Tandatangan", "Verifikasi" dan "Pengaturan".

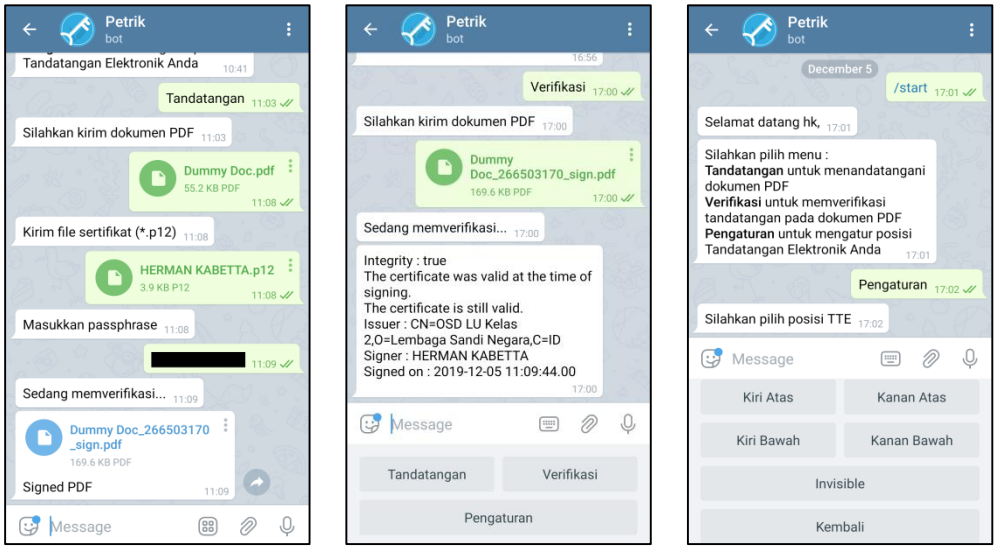
(a) (b)



Gambar 8. Tampilan awal *chatbot*, (a) Tampilan ketika pertama kali diakses; (b) Tampilan ketika tombol "start" ditekan.

Proses penandatanganan dokumen dimulai ketika pengguna menekan tombol "Tandatangan" pada menu awal. Sebelum memulai penandatanganan dokumen, pengguna harus memiliki sertifikat elektronik yang dikeluarkan oleh Balai Sertifikasi Elektronik (BSrE). Sertifikat elektronik biasanya diberikan berpasangan dengan *passphrase*-nya. *Passphrase* adalah kata sandi yang digunakan sebagai autentikasi kepemilikan sertifikat elektronik.

(a) (b) (c)



Gambar 9. Tampilan percakapan (a) Proses tanda tangan dokumen; (b) Proses verifikasi dokumen; (c) Menu pengaturan posisi tanda tangan

Gambar 9(a) merupakan tampilan saat percakapan penandatanganan dokumen. Pada awal percakapan, pengguna diminta untuk mengirimkan dokumen pdf yang akan ditandatangani, dilanjutkan kemudian mengirim berkas sertifikat elektronik dan diteruskan dengan memasukkan *passphrase* dari sertifikat elektronik tersebut. Percakapan berlangsung satu per satu bergantian balas membalas antara *chatbot* dan pengguna. Pada akhir percakapan, *chatbot* mengirimkan dokumen yang telah ditandatangani. Fitur kedua *chatbot* adalah verifikasi tanda tangan elektronik pada dokumen pdf. Gambar 9(b) merupakan

tampilan percakapan saat proses verifikasi dokumen. Pada awal percakapan, pengguna diminta untuk mengirimkan dokumen pdf yang bertandatangan. Selanjutnya *chatbot* akan memproses dan melakukan pengecekan terhadap dokumen, kemudian mengirimkan hasil verifikasi dokumen kepada pengguna. Hasil verifikasi berupa integritas dokumen, validitas sertifikat elektronik, identitas *Certificate Authority*, penandatangan dokumen serta waktu penandatanganan. Gambar 9(c) merupakan tampilan percakapan dan menu untuk mengatur posisi letak tanda tangan elektronik. Terdapat lima menu pilihan posisi, kiri atas, kanan atas, kiri bawah, kanan bawah dan *invisible* (tidak nampak). Pengaturan posisi tanda tangan akan tersimpan pada basis data, sehingga pengguna tidak perlu mengatur berulang kali setiap berkomunikasi dengan *chatbot*.

Pengujian sistem dilakukan untuk memastikan *chatbot* dapat berjalan sesuai dengan spesifikasi dan ekspektasi pengguna. Menurut Ul Haq (2019), terdapat banyak variasi pengujian perangkat lunak, salah satunya dapat dilakukan dengan metode *black-box.* Berikut ini skenario pengujian fungsional dengan metode *black-box* yang dilakukan terhadap sistem *chatbot* dengan daftar disajikan pada Tabel 1.

Tabel 1. Daftar skenario pada pengujian fungsional *chatbot*

No.	***Spesifikasi***	***Test Case ID***	***Test Case***
1	Tanda tangan dokumen	C01	Proses tanda tangan dengan format dokumen valid, sertifikat valid dan *passphrase* benar
		C02	Proses tanda tangan dengan format dokumen valid, sertifikat valid dan *passphrase* salah atau *invalid*
		C03	Proses tanda tangan dengan format dokumen valid, sertifikat *invalid* atau kadaluwarsa
		C04	Proses tanda tangan dengan format dokumen *invalid*
2	Verifikasi dokumen	C05	Proses verifikasi dengan format dokumen valid
		C06	Proses verifikasi dengan format dokumen *invalid*
		C07	Proses verifikasi dengan dokumen yang belum ditandatangani
3	Pengaturan posisi tanda tangan	C08	Proses pengaturan dengan perintah valid
		C09	Proses pengaturan dengan perintah *invalid*
		C10	Menandatangani dokumen setelah perubahan pengaturan posisi tanda tangan
4	*Illegal message handling*	C11	Menuliskan pesan perintah ilegal pada awal percakapan

Pada spesifikasi pertama yaitu proses tanda tangan dokumen. Masing-masing skenario mewakili tiap tahap percakapan yang berjalan secara berurutan, sehingga percakapan tidak akan berlanjut apabila percakapan sebelumnya tidak valid. Hal ini berlaku pula pada spesifikasi pengujian yang lainnya. Urutan tahap percakapan *user* pada proses tanda tangan adalah, 1) Mengirimkan pesan perintah "Tandatangan", 2) Mengirimkan dokumen PDF, 3) Mengirimkan berkas sertifikat elektronik, 4) Mengirimkan *passphrase*. Masing-masing tahap percakapan kemudian diuji dengan data valid dan tidak valid sehingga terdapat empat skenario yang diuji, daftar skenario secara lengkap disajikan pada Tabel 2.

Tabel 2. Skenario pengujian untuk spesifikasi tanda tangan dokumen

ID	***Test Case***	***Step Detail / Input Value***	***Expected Result***	***Actual Result***
C01	Proses tanda tangan dengan format	Mengirimkan pesan teks "Tandatangan"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai

	dokumen valid, sertifikat valid dan *passphrase* benar	Mengirimkan dokumen PDF	Chatbot memberikan pesan balasan "Kirim file sertifikat (*.p12)"	Sesuai
		Mengirimkan berkas sertifikat elektronik (*.p12) yang masih berlaku	Chatbot memberikan pesan balasan "Masukkan passphrase"	Sesuai
		Mengirimkan pesan teks berupa passphrase yang benar dari sertifikat elektronik	*Chatbot* memberikan dokumen yang telah ditandatangani	Sesuai
C02	Proses tanda tangan dengan format dokumen valid, sertifikat valid dan *passphrase* salah atau *invalid*	Mengirimkan pesan teks "Tandatangan"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai
		Mengirimkan dokumen PDF	Chatbot memberikan pesan balasan "Kirim file sertifikat (*.p12)"	Sesuai
		Mengirimkan berkas sertifikat elektronik (*.p12) yang masih berlaku	Chatbot memberikan pesan balasan "Masukkan passphrase"	Sesuai
		Mengirimkan pesan teks berupa *passphrase* yang salah	Muncul peringatan *passphrase* tidak sesuai	Sesuai
C03	Proses tanda tangan dengan format dokumen valid, sertifikat *invalid* atau kadaluwarsa	Mengirimkan pesan teks "Tandatangan"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai
		Mengirimkan dokumen PDF	Chatbot memberikan pesan balasan "Kirim file sertifikat (*.p12)"	Sesuai
		Mengirimkan berkas selain sertifikat p12 (pdf, doc, jpg, xls) atau mengirimkan berkas sertifikat elektronik yang kadaluwarsa	Chatbot memberikan pesan peringatan "Sertifikat kadaluwarsa atau tidak valid"	Sesuai
C04	Proses tanda tangan dengan format dokumen *invalid*	Mengirimkan pesan teks "Tandatangan"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai
		Mengirimkan dokumen selain PDF (doc, jpg, xls)	Chatbot memberikan pesan peringatan "Hanya dapat menandatangani berkas PDF"	Sesuai

Spesifikasi kedua adalah proses verifikasi dokumen, terdiri dari tiga skenario yang disajikan pada tabel 3. Urutan tahap percakapan *user* pada proses tanda tangan adalah, 1) Mengirimkan pesan perintah "Verifikasi", 2) Mengirimkan dokumen PDF. Pada proses ini, terdapat tiga masukan data berbeda yang diujikan, yaitu berupa dokumen PDF yang telah ditandatangai, dokumen PDF yang belum ditandatangani dan dokumen selain PDF.

Tabel 3. Skenario pengujian untuk spesifikasi verifikasi dokumen

ID	***Test Case***	***Step Detail / Input Value***	***Expected Result***	***Actual Result***
C05	Proses verifikasi dengan format dokumen valid	Mengirimkan pesan teks "Verifikasi"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai
		Mengirimkan dokumen PDF yang telah ditandatangani	*Chatbot* memberikan pesan balasan yang berisi informasi sertifikat elektronik	Sesuai
C06	Proses verifikasi dengan format dokumen *invalid*	Mengirimkan pesan teks "Verifikasi"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai
		Mengirimkan dokumen selain PDF	Chatbot memberikan pesan peringatan "Hanya dapat memverifikasi berkas PDF"	Sesuai
C07	Proses verifikasi dengan dokumen yang belum ditandatangani	Mengirimkan pesan teks "Verifikasi"	Chatbot memberikan pesan balasan "Silahkan kirim dokumen PDF"	Sesuai
		Mengirimkan dokumen PDF yang belum ditandatangani	Chatbot memberikan pesan peringatan "Tidak ditemukan tanda tangan elektronik"	Sesuai

Spesifikasi ketiga adalah proses pengaturan posisi tanda tangan, terdiri dari tiga skenario yang disajikan pada tabel 4. Urutan tahap percakapan *user* pada proses pengaturan adalah, 1) Mengirimkan pesan perintah "Pengaturan", 2) Mengirimkan pesan pengaturan. Terdapat lima pesan pengaturan yang dapat dipilih, yaitu "Kiri Atas", "Kanan Atas", "Kiri Bawah", "Kanan Bawah" dan "Invisible". Pengujian dilakukan untuk masing-masing pesan pengaturan tersebut beserta pesan pengaturan yang tidak terdapat pada daftar, kemudian dilakukan proses tanda tangan untuk menguji posisi tanda tangan sudah berada pada posisi yang semestinya.

Tabel 4. Skenario pengujian untuk spesifikasi pengaturan posisi tanda tangan

ID	*Test Case*	*Step Detail / Input Value*	*Expected Result*	*Actual Result*
C08	Proses pengaturan dengan perintah valid	Mengirimkan pesan teks “Pengaturan”	Chatbot memberikan pesan balasan “Silahkan pilih posisi TTE”	Sesuai
		Mengirimkan salah satu pesan pengaturan yaitu “Kiri Atas”, “Kanan Atas”, “Kiri Bawah”, Kanan Bawah” atau ”invisible”	*Chatbot* memberikan pesan balasan “Pengaturan telah disimpan”	Sesuai
C09	Proses pengaturan dengan perintah *invalid*	Mengirimkan pesan teks “Pengaturan”	Chatbot memberikan pesan balasan “Silahkan pilih posisi TTE”	Sesuai
		Mengirimkan pesan teks selain “Kiri Atas”, “Kanan Atas”, “Kiri Bawah”, Kanan Bawah” atau ”invisible”	*Chatbot* memberikan pesan balasan “Pengaturan tidak dikenal”	Sesuai
C10	Menandatangani dokumen setelah perubahan pengaturan posisi tanda tangan	Mengirimkan pesan teks “Tandatangan”	Chatbot memberikan pesan balasan “Silahkan kirim dokumen PDF”	Sesuai
		Mengirimkan dokumen PDF	Chatbot memberikan pesan balasan “Kirim file sertifikat (*.p12)”	Sesuai
		Mengirimkan berkas sertifikat elektronik (*.p12) yang masih berlaku	Chatbot memberikan pesan balasan “Masukkan passphrase”	Sesuai
		Mengirimkan pesan teks berupa passphrase yang benar dari sertifikat elektronik	Chatbot memberikan dokumen yang telah ditandatangani dengan posisi sesuai pengaturan yang baru	Sesuai

Illegal message handling merupakan skenario untuk menangani kesalahan penulisan perintah pada awal percakapan dengan *chatbot*. Awal percakapan dengan *chatbot* dimulai setelah *user* menekan tombol ”START”. Pada awal percakapan, *user* diminta untuk memilih satu diantara tiga menu yaitu “Tandatangan”, “Verifikasi” dan “Pengaturan”. Penanganan perlu dilakukan apabila *user* tidak memilih satu diantara tiga daftar menu tersebut, atau salah menuliskan nama menu. Detail skenario disajikan pada tabel 5.

Tabel 5. Skenario pengujian untuk spesifikasi *illegal message handling*

ID	*Test Case*	*Step Detail / Input Value*	*Expected Result*	*Actual Result*
C11	Menuliskan perintah ilegal pada awal percakapan	Mengirimkan pesan berupa teks yang tidak terdaftar, contoh : • “tanda tangan” • “sign”	Chatbot memberikan pesan peringatan “Perintah tidak dikenali”	Sesuai

Pengujian non-fungsional dilakukan untuk menguji aspek *compatibility chatbot* apakah dapat berjalan dengan baik pada semua platform sistem operasi, dan apakah tanda tangan elektronik yang tersimpan pada dokumen dapat dikenali oleh aplikasi pembaca berkas pdf lainnya. Dua skenario pengujian non-fungsional diujikan pada sistem *chatbot* dengan hasil disajikan pada Tabel 6 dan Tabel 7.

Tabel 6. Daftar skenario pada pengujian non-fungsional *chatbot*

No.	*Domain*	*Test Case ID*	*Test Case*
1	*Compatibility*	C12	*Chatbot* dapat berjalan di sistem operasi *mobile* (Android dan iOS), sistem operasi *desktop* (Windows, MacOS) dan web.
		C13	Tanda tangan elektronik dapat dikenali oleh aplikasi pembaca PDF

Tabel 7. Skenario pengujian untuk domain *compatibility*

ID	*Test Case*	*Step Detail / Input Value*	*Expected Result*	*Actual Result*
C12	*Chatbot* dapat berjalan di sistem operasi *mobile* (Android dan iOS), sistem operasi *desktop* (Windows, MacOS) dan web.	Menjalankan proses tanda tangan, verifikasi dan pengaturan pada setiap platform	*Chatbot* dapat berjalan dengan baik pada semua platform sistem operasi dan web	Sesuai
C13	Tanda tangan elektronik dapat dikenali oleh aplikasi pembaca PDF	Memvalidasi tanda tangan elektronik menggunakan aplikasi Adobe Acrobat Reader dan Foxit PDF Reader	Tanda tangan elektronik dapat dikenali pada aplikasi Adobe Acrobat Reader dan Foxit PDF Reader	Sesuai

Evaluasi sistem dilakukan menggunakan *System Usability Scale* (SUS) untuk mengetahui tingkat kelayakan dan kegunaan dari *chatbot*. *System Usability Scale* (SUS) yang diciptakan oleh John Brooke pada tahun 1986, merupakan salah satu metode yang dapat digunakan untuk mengukur persepsi kegunaan sebuah perangkat keras maupun perangkat lunak (Boyd, 2018). Responden kuesioner SUS berjumlah sepuluh orang yang merupakan pengguna sertifikat elektronik BsrE dan pengumpulan data dilakukan secara daring menggunakan kuesioner dari *Google Forms*. Hasil pengumpulan data disajikan pada tabel 8.

Tabel 8. Hasil kuesioner *System Usability Scale*

No.	*Pertanyaan*	*Rata-rata skor*
1	Saya sepertinya akan sering menggunakan *chatbot* ini	2,4
2	Ada fitur pada *chatbot* yang sebenarnya tidak perlu	3,9
3	Saya merasa mudah menggunakan *chatbot* ini	1,9
4	Saya sepertinya perlu bantuan teknis untuk mengoperasikan *chatbot* ini	4,1
5	Saya menemukan berbagai fungsi dalam *chatbot* ini telah terintegrasi dengan baik	2,5
6	Saya pikir ada terlalu banyak ketidaksesuaian dalam *chatbot* dan sistem pendukungnya	4,5
7	Saya rasa mayoritas pengguna akan dapat mempelajari *chatbot* ini dengan cepat	2,4
8	Saya merasa *chatbot* ini sangat tidak praktis ketika digunakan	4,3
9	Saya sangat yakin dapat menggunakan *chatbot* ini	2,5
10	Sepertinya saya harus belajar banyak hal terlebih dahulu sebelum saya dapat menggunakan *chatbot* ini	4,3
	Total	32,8
	Skor SUS (2,5 * Total)	82,0

SUS terdiri dari 10 pertanyaan, dengan lima pertanyaan positif dan 5 pertanyaan negatif serta setiap pertanyaan memiliki bobot 0 sampai 4. Pertanyaan nomor ganjil merupakan pertanyaan positif, skor setiap pertanyaan dihitung dengan cara bobot tiap pertanyaan dikurangi dengan nilai 1 (bobot - 1). Pertanyaan genap yang merupakan pertanyaan negatif, skor dihitung dengan cara 5 dikurangi bobot setiap pertanyaan (5 - bobot). Total skor diperoleh dari jumlah rata-rata skor setiap pertanyaan, kemudian total skor dikalikan 2,5 untuk mendapatkan skor SUS antara 0-100. Menurut Baumgartner (2019), standar minimum skor SUS adalah 65 untuk memastikan bahwa produk dapat diterima oleh pengguna. Dari hasil perhitungan diperoleh skor SUS sebesar 82,0 yang menunjukkan bahwa sistem yang dibangun berada dalam kategori baik dan layak untuk digunakan.

KESIMPULAN DAN SARAN

Berdasarkan penelitian dan pengujian yang telah dilakukan, dapat disimpulkan:

1. *Chatbot* penandatangan elektronik dapat dibangun dengan metode *Rapid Application Development* (RAD) dan dapat diakses dengan mudah melalui aplikasi perpesanan Telegram.
2. Dokumen yang telah ditandatangani *chatbot* terbukti dapat dikenali dan divalidasi oleh aplikasi pembaca pdf lainnya.
3. Pengujian dilakukan pada komponen fungsional dan non-fungsional. Hasil pengujian terhadap tiga fungsi utama (fungsional) *chatbot* yaitu tanda tangan, verifikasi dan pengaturan, menunjukkan hasil yang sesuai dengan *requirements planning* dan *chatbot* dapat berjalan dengan baik. Hasil pengujian terhadap komponen non-fungsional menunjukkan *chatbot* dapat berjalan dengan baik pada semua platform sistem operasi maupun pada platform web.
4. Evaluasi sistem menggunakan *System Usability Scale* (SUS) menghasilkan skor sebesar 82,0 yang menunjukkan bahwa *chatbot* berada dalam kategori baik dan layak untuk digunakan.

Adapun saran untuk penelitian lebih lanjut adalah perlunya pengembangan terhadap mekanisme pengamanan data, mengingat *chatbot* menggunakan server dari pihak ketiga yakni Telegram, serta pengembangan lebih lanjut agar *chatbot* dapat menandatangani sertifikat dari *Certificate Authority* selain BSrE.

DAFTAR PUSTAKA

Baumgartner, J., Frei, N., Kleinke, M., Sauer, J., & Sonderegger, A. (2019). Pictorial System Usability Scale (P-SUS) Developing an Instrument for Measuring Perceived Usability. *In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1-11)*.

Boyd, K., Bond, R., Magee, J., & McCormack, P. (2018). Can users recall their user experience with a technology? Temporal bias and the system usability scale. *In Proceedings of the 32nd International BCS Human Computer Interaction Conference 32 (pp. 1-6)*.

Calvo D., Quesada L., López G., Guerrero L.A. (2017) Multiplatform Career Guidance System Using IBM Watson, Google Home and Telegram. In: Ochoa S., Singh P., Bravo J. (eds) Ubiquitous Computing and Ambient Intelligence. UCAmI 2017. *Lecture Notes in Computer Science, vol 10586*. Springer, Cham.

Dhagat, R. and Joshi, P., (2016). New Approach of User Authentication Using Digital Signature. *Symposium on Colossal Data Analysis and Networking (CDAN)* (pp. 1-3). Indore.

Dubosson, F., Schaer, R., Savioz, R., & Schumacher, M. (2017). Going beyond the relapse peak on social network smoking cessation programmes: ChatBot opportunities. *Swiss medical informatics*, *33*(00).

Forsby, F., Furuhed, M., Papadimitratos, P., & Raza, S. (2017). Lightweight X. 509 digital certificates for the Internet of Things. *In Interoperability, Safety and Security in IoT* (pp. 123-133). Springer, Cham.

Hassan, S., Qamar, U., & Idris, M. A. (2015). Purification of requirement engineering model for rapid application development. In *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 357-362). IEEE.

Husni, E., & Hasibuan, F. (2018). Driver Supervisor System with Telegram Bot Platform. In *International Conference on Computational Collective Intelligence* (pp. 436-444). Springer, Cham.

Karthikeyan, S., Patan, R., & Balamurugan, B. (2019). Enhancement of Security in the Internet of Things (IoT) by Using X. 509 Authentication Mechanism. *In Recent Trends in Communication, Computing, and Electronics* (pp. 217-225). Springer, Singapore.

Kneuper R. (2018). Software Processes in the Software Product Life Cycle. In: Software Processes and Life Cycle Models. Springer, Cham.

Korotaeva, D., Khlopotov, M., Makarenko, A., Chikshova, E., Startseva, N., & Chemysheva, A. (2018). Botanicum: a Telegram Bot for Tree Classification. In *2018 22nd Conference of Open Innovations Association (FRUCT)* (pp. 88-93). IEEE.

Martin, J. (1991). *Rapid application development*. Macmillan Publishing Co., Inc..

Pemerintah Indonesia. (2016). Undang-Undang Nomor 19 Tahun 2016 Tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 Tentang Informasi dan Transaksi Elektronik. Sekretariat negara, Jakarta.

Pereira, C., Barbosa, L., Martins, J., & Borges, J. (2018). Digital Signature Solution for Document Management Systems-The University of Trás-os-Montes and Alto Douro. *In World Conference on Information Systems and Technologies* (pp. 16-25). Springer, Cham.

Rosid, M. A., Rachmadany, A., Multazam, M. T., Nandiyanto, A. B. D., Abdullah, A. G., & Widiaty, I. (2018). Integration Telegram Bot on E-Complaint Applications in College. In *IOP Conference Series: Materials Science and Engineering* (Vol. 288, No. 1, p. 012159). IOP Publishing.

Setyatama, F., & IrwanKurnia, A. (2018). Rapid Application Development (RAD) Method For Developing Clinical Laboratory Information System (Case Study: PT. Populer Sarana Medika). *Journal of Electrical Engineering And Computer Sciences, Vol. 3 Number 2*, 3(2).

Singh, D. A. A. G., Leavline, E. J., & Vijayan, P. M. (2017). Mobile Application for Student Attendance and Mark Management System. *International Journal of Computational Intelligence Research, 13(3)*, 425-432.

Sucipto, S., Resti, N. C., Andriyanto, T., Karaman, J., & Qamaria, R. S. (2019). Transactional Database Design Information System Web-Based Tracer Study Integrated Telegram Bot. *Journal of Physics: Conference Series (Vol. 1381, No. 1, p. 012008)*. IOP Publishing.

Ul Haq, S., & Qamar, U. (2019). Ontology Based Test Case Generation for Black Box Testing. *Proceedings of the 2019 8th International Conference on Educational and Information Technology (pp. 236-241).* Association for Computing Machinery (ACM).

Vukasović, M., Veselinović, B., & Stanisavljević, Ž. (2017). A development of a configurable system for handling X509 certificates. In *2017 25th Telecommunication Forum (TELFOR)* (pp. 1-4). IEEE.

Yusandy, T. (2019). Kedudukan dan Kekuatan Pembuktian Alat Bukti Elektronik dalam Hukum Acara Perdata Indonesia. *Jurnal Serambi Akademica*, 7(5), 645-656.