

ROBOTIC PROCESS AUTOMATION FOR QUALITY CONTROL ASSESSMENT USING SELENIUM

Refgiufi Patria Avrianto^{*1}, Mochamad Isnin Faried², Erick Dazki³, Richardus Eko Indrajit⁴

^{1,3,4}Teknologi Informasi, Universitas Pradita, Indonesia

²Teknik Informatika, Fakultas Teknik, Institut Perbanas, Indonesia

Email: ¹refgiufi.patria@student.pradita.ac.id, ²isninfaried@perbanas.id, ³erick.dazki@pradita.ac.id,
⁴eko.indrajit@pradita.ac.id

(Naskah masuk: 24 Mei 2022, Revisi: 2 Juni 2022, diterbitkan: 24 Oktober 2022)

Abstract

Robotic Process Automation (RPA) is a form of technology to automate tasks and routines performed by humans, which can be instructed to machine. Along with the rapid development of technology, the time that can be used to maintain the quality of a technology is becoming increasingly narrow. Therefore, RPA will imitate activities or tasks performed by humans on a computer with a much faster processing time and 100% accuracy without "human error". The agile method was chosen in this study because of its focus on flexibility and more responsiveness to required changes. The system developed in this paper uses the Selenium library with the Python programming language and retrieves task data from the MNC IT team, especially the QC division for the RCTI+ website Quality Assessment process. From the results of testing this system, it shows that the testing process using this system and testing from a tester / examiner get the same results. However, because this system is programmed, it will have more stable and efficient results.

Keywords: *Robotic Process Automation, Selenium, Software Quality Assurance, Testing System, Website.*

ROBOTIC PROCESS AUTOMATION UNTUK QUALITY CONTROL ASSESSMENT MENGGUNAKAN SELENIUM

Abstrak

Robotic Process Automation (RPA) adalah bentuk teknologi otomatisasi tugas dan rutinitas yang dilakukan manusia, yang dapat diinstruksikan ke mesin. Seiring perkembangan teknologi yang cepat, waktu yang dapat digunakan untuk mempertahankan kualitas suatu teknologi menjadi semakin sempit. Oleh karena itu, RPA akan meniru aktivitas atau tugas yang dilakukan manusia di dalam komputer dengan waktu pengerjaan yang jauh lebih cepat dan akurasi 100% tanpa "kesalahan manusia" (human error). Metode Agile dipilih dalam penelitian ini karena fokusnya pada fleksibilitas dan lebih responsif terhadap perubahan yang diperlukan. Sistem yang dikembangkan pada penulisan ini menggunakan library Selenium dengan bahasa pemrograman Python dan mengambil data tugas/task dari tim IT MNC khususnya divisi QC untuk proses Quality Assessment website RCTI+. Dari hasil pengujian sistem ini menunjukkan bahwa proses testing menggunakan sistem ini dan testing dari seorang tester/penguji mendapatkan hasil yang sama. Namun karena sistem ini terprogram, maka sistem ini akan memiliki hasil yang lebih stabil dan efisien.

Kata kunci: *Robotic Process Automation, Selenium, Software Quality Assurance, Sistem Pengujian, Website.*

1. PENDAHULUAN

Perkembangan teknologi informasi memberikan dampak besar dalam kehidupan masyarakat. Sejak ditemukannya komputer pada tahun 1955, peradaban dunia telah memasuki periode informasi. Teknologi informasi dengan komputer sebagai motor penggerak telah mengubah segalanya. Pemrosesan informasi berbasis komputer mulai dikenal dan sampai sekarang ada banyak aplikasi yang dapat digunakan sebagai alat

pengolah data untuk menghasilkan informasi. Dengan berbagai banyaknya jumlah aplikasi yang dapat membantu pekerjaan manusia, pekerjaan manusia itu menjadi tidak terlalu berat. Karena hal inilah, sampai saat ini masih banyak pekerjaan yang dilakukan manusia dengan mengandalkan aplikasi walaupun prosesnya selalu diulangi. Tetapi tidak dapat dipungkiri ketika manusia dipaksa untuk mengerjakan tugas satu-persatu dan membosankan secara berulang kali, hal ini dapat menyebabkan frustrasi dari waktu ke waktu dan menjadi rentan

terhadap kesalahan [1]. Tidak seperti manusia, robot tidak memiliki perasaan sehingga robot akan tetap menghasilkan kualitas pekerjaan yang sama seiring dengan berjalannya waktu. *Robotic Process Automation* (RPA) dapat mengatasi hal tersebut. RPA adalah istilah umum untuk sistem yang beroperasi pada antarmuka pengguna sistem komputer seperti yang dilakukan manusia. RPA bertujuan untuk melakukan pekerjaan manusia dengan cara otomatisasi. Gartner yang merupakan perusahaan teknologi informasi dan firma penasihat Amerika Serikat, mendefinisikan RPA sebagai berikut: "RPA merupakan alat yang melakukan pernyataan *[if, then, else]* pada data terstruktur, biasanya menggunakan kombinasi interaksi antarmuka pengguna, atau dengan menghubungkan ke API atau menggerakkan *server-client, mainframe* atau kode HTML. RPA beroperasi dengan memetakan proses dengan menyesuaikan bahasa pemrograman RPA untuk diikuti oleh robot perangkat lunak, dengan runtime yang dialokasikan untuk mengeksekusi program" [2]. Oleh karena itu, RPA bertujuan untuk mengurangi beban tugas-tugas sederhana yang berulang pada karyawan [3]. Dengan RPA memungkinkan untuk mengotomatisasi tugas-tugas yang biasanya dilakukan oleh manusia dalam semua aplikasi dan sistem, baik di *front office* dan *back office* sehingga dapat meningkatkan efisiensi dan efektivitas agar lebih kompetitif [1].

RCTI merupakan stasiun televisi yang memiliki jangkauan terluas di Indonesia, melalui 54 stasiun *relay*nya program-program RCTI disaksikan oleh lebih dari 191 juta pemirsa yang tersebar di 452 kota di seluruh Nusantara, atau kira-kira 80.7% dari jumlah penduduk Indonesia [4]. Saat ini, RCTI sedang mengembangkan sebuah aplikasi bernama RCTI+ yang bertujuan untuk mengembangkan dunia hiburan di Indonesia menggunakan pengembangan dari teknologi informasi di bidang media pertelevisian. RCTI+ adalah sebuah aplikasi tv streaming yang memungkinkan setiap orang dapat menyaksikan siaran yang ada pada MNC Group (RCTI, MNCTV, GTV dan iNews) secara *live streaming*. Dan juga, di RCTI+ menghadirkan konten-konten lawas atau konten yang pernah disiarkan agar dapat disaksikan kembali oleh pengguna. Tidak berhenti sampai disitu, RCTI+ juga menghadirkan konten-konten eksklusif yang tidak ditayangkan pada tv dan hanya dapat disaksikan pada aplikasi RCTI+. Dengan berbagai konten yang disajikan, maka aplikasi RCTI harus meminimalisir adanya *bug* pada aplikasi agar para pengguna dapat menggunakan aplikasi tanpa hambatan ataupun kendala. Untuk memastikan kelancaran aplikasi berjalan, mutu aplikasi harus dikendalikan agar kualitasnya tetap baik dan terjaga. Kualitas baik maupun buruk produk perusahaan tergantung dari pekerjaan karyawan di bidang ini yaitu *quality control*. Untuk alasan inilah RCTI memiliki tim *Software Quality Assurance* (SQA) dalam

perusahaannya. Menjadi seorang SQA bisa dibilang bukan pekerjaan yang mudah karena membutuhkan ketelitian dan kesabaran untuk mencari-cari dimana letak *bug* pada aplikasi yang sedang dikembangkan sebelum bisa dipasarkan.

Kreatifitas pengujian SQA dalam menguji sebuah aplikasi atau modul sangat penting. Seorang pengujian SQA harus mempunyai kemampuan analitis yang baik dalam memprediksi dimana letak kesalahan suatu program. Seorang pengujian SQA juga harus mampu melakukan uji kelayakan program dengan tujuan mencari hal yang salah dalam sebuah program, bukan melakukan pengujian untuk mendapatkan hasil bahwa program sudah dikembangkan dengan benar. Seorang pengujian SQA harus secara proaktif dan ulet untuk mencoba mencari kesalahan sebuah program. Seiring berjalannya waktu, durasi pengembangan aplikasi semakin menyempit. Pola pengembangan pemrograman yang lebih fleksibel dapat mempersingkat waktu pengembangan. Kode pemrograman dapat berubah dalam hitungan minggu atau bahkan hari. Hal ini mengakibatkan meningkatnya resiko kode-kode yang baru merusak program yang sudah dibuat. Menguji semua modul satu-persatu setiap hari atau setiap minggu akan membutuhkan banyak sumber daya dan sangat tidak efektif. Hal ini dikarenakan pengujinya adalah manusia yang sangat rentan akan kebosanan menjalankan pengujian yang itu saja dan akan lelah seiring berjalannya waktu.

Maka dari itu untuk mengatasi tenggat jadwal yang semakin singkat dan membantu dalam melakukan uji coba aplikasi, perlu dibuatnya sistem yang dapat melakukan tugas seorang pengujian SQA. Untuk membantu peran manusia sebagai pengujian SQA maka bisa menggunakan robot yang digerakkan dengan teknologi *Robotic Process Automation* (RPA). RPA adalah program aplikasi yang berjalan sebagai pengguna akhir. RPA tidak memerlukan pengembangan kode, dan juga tidak memerlukan akses langsung ke sistem atau database aplikasi. Pada sebuah studi kasus yang dikeluarkan oleh Infosys, implementasi RPA dapat menghasilkan penurunan *Full Time Equivalent* (FTE) atau beban kerja karyawan sebesar 50%, dan menurunkan pekerjaan yang dilakukan seseorang sebesar 58%. Aktivitas robot akan direkam dan disimpan dalam log yang tersedia, dan dari data tersebut dihasilkan analisis bahwa dengan robot dapat meningkatkan waktu proses mencapai 70%. Sedangkan menurut Prof Leslie Willcocks, *Professor of Technology Work and Globalization and governor of the Information Systems and Innovation Group* di *London School of Economics*, dari hasil penelitiannya yang disajikan di laporan penelitian McKinsey, disampaikan bahwa *Return of Investment* (ROI) atau tingkat pengembalian investasi implementasi RPA di perusahaan bervariasi berkisar

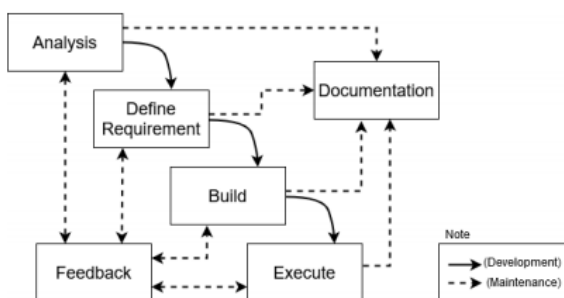
antara 30% sampai dengan 200% pada tahun pertamanya.[1]

Dengan penerapan konsep RPA dalam kegiatan SQA khususnya dalam hal pengujian aplikasi yang bertujuan untuk meningkatkan *outcome* karyawan dari segi waktu dan tenaga sehingga RPA akan menjadi alat bantu yang efektif dan powerful. Berdasarkan deskripsi diatas, maka penulis ingin melakukan penelitian berjudul “*Robotic Process Automation Untuk Quality Control Assessment Menggunakan Selenium*”.

2. METODE PENELITIAN

Penelitian ini melakukan pengembangan sistem menggunakan model *Software Development Life Cycle* (SDLC). SDLC adalah proses pembuatan dan pengembangan perangkat lunak dengan menggunakan model dan solusi dari permasalahan yang dialami pengguna pada perangkat lunak sebelumnya[5]. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari: perencanaan (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pemeliharaan (*maintenance*).

Dalam penelitian ini, tahapan yang dilakukan peneliti menggunakan metode Agile. Agile memungkinkan mengembangkan perangkat lunak yang memiliki perencanaan adaptif, interaksi berkelanjutan, kolaborasi, dan tanggap terhadap perubahan. Peneliti merumuskan beberapa tahapan yang dapat digambarkan sebagai berikut:



Gambar 1. Tahapan penelitian dengan Metode Agile

Berdasarkan tahap-tahap diatas dapat diketahui bahwa terdapat beberapa langkah dalam penelitian, berikut penjelasan untuk masing-masing langkah[6]:

- *Analysis*

Analisis sistem dapat diidentifikasi sebagai penguraian dari suatu sistem informasi kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan pengembangannya.

- *Defied Requirement*

Mengumpulkan informasi sebanyak-banyaknya yang dibutuhkan (misalnya *form-form* yang

digunakan) pada sistem yang dibuat dalam proses penelitian pengembangan sistem.

- *Build*

Di tahap ini, pengembang akan mentransformasikan kebutuhan secara terperinci. Pembangunan sistem berfokus pada bagaimana caranya agar dapat memenuhi berbagai fungsi yang dibutuhkan oleh sebuah sistem.

- *Execute and Feedback*

Setelah proses pembuatan selesai dan sistem dapat berjalan, maka tahap pengujian dapat dimulai, pengujian memfokuskan pada logika internal dan fungsi eksternal serta mencari kesalahan pada sistem yang dikembangkan sehingga memenuhi kebutuhan *client*.

- *Documentation*

Setelah dilakukan tes perangkat lunak langkah selanjutnya yaitu proses dokumentasi perangkat lunak untuk mempermudah proses *maintenance* kedepannya.

- *Maintenance*

Tahap terakhir yaitu pemeliharaan. Tidak ada perangkat lunak yang 100% bebas dari *bug*, oleh karena itu sangatlah penting agar perangkat lunak dipelihara secara berkala. Pemeliharaan bertujuan agar sistem yang telah dikembangkan dapat mengikuti perubahan apapun sesuai keinginan *client* dan untuk mencapai tujuan penggunaannya yang lebih efisien, efektif dan akurat serta tepat guna.

A. *Robotic Process Automation*

Robotic Process Automation atau sering disebut dengan RPA adalah ilmu/teknik di mana perangkat lunak komputer digunakan untuk menyelesaikan proses tertentu yang sebelumnya dilakukan oleh manusia. Perangkat lunak yang dijalankan oleh robot tidak menggantikan sistem. Sebaliknya, ia bekerja dengan sistem dan melakukan tugas tertentu dengan cara yang sama seperti yang diminta untuk diselesaikan. RPA merupakan sebuah perangkat lunak yang mampu untuk meniru aktivitas pekerjaan manusia dengan berinteraksi melalui antarmuka pengguna untuk mengambil data dan memanipulasi aplikasi seperti halnya manusia [7]. Alih-alih mempekerjakan karyawan yang dibayar gaji untuk melakukan tugas yang berulang di komputer, RPA dapat digunakan untuk melakukan proses yang mencakup pengetikan dan mengklik dengan cara yang sama dengan manusia. RPA tidak perlu mengubah sistem lama. RPA dapat diintegrasikan dengan perangkat lunak apapun yang digunakan oleh manusia dan dapat diimplementasikan dalam waktu singkat untuk tujuan melaksanakan prosedur operasional[8].

RPA ini dimaksudkan untuk mengurangi tugas yang diberikan kepada pekerja. Teknologi yang digunakan dalam proses otomatis ini adalah robot yang dimodifikasi untuk melakukan tugas dalam proses kerja tertentu yang biasanya dikerjakan karyawan. Perangkat lunak ini tidak belajar mandiri

atau mengubah keefektifan baru atau wawasan baru, misalnya, analisis data yang besar atau perangkat lunak manajemen sumber daya perusahaan. RPA bekerja seperti asisten digital untuk pekerja dengan menyelesaikan tugas sederhana dan berulang yang menjalani sebagian besar hari mereka sebagai pekerja kantor.

Dengan demikian, RPA adalah sistem yang lebih sederhana daripada sistem berbasis kecerdasan buatan atau program yang menopang semua informasi dalam satu tempat [1]. Ini juga membuatnya menjadi barang yang relatif lebih murah daripada perangkat lunak AI. Kesederhanaan dan murahnya relatif ini bisa membuat RPA menjadi solusi yang lebih menarik bagi banyak perusahaan, terutama jika perusahaan memiliki sistem dan aplikasi lawas. RPA dirancang untuk dapat bekerja dengan kebanyakan sistem dan aplikasi lawas, sehingga memudahkan dalam penerapan. RPA pada akhirnya adalah tentang mengotomatisasi beberapa tugas dan proses berbasis komputer yang paling biasa dan berulang di tempat kerja. Contohnya tugas *copy-paste* dan pindahkan *file* dari satu lokasi ke lokasi lain, misalnya.

B. Selenium

Selenium adalah alat otomatisasi perangkat lunak sumber terbuka yang dikembangkan oleh lisensi Apache 2.0, yang mendukung berbagai jenis pengujian dalam aplikasi web. Ini bukan hanya satu alat tetapi terdiri dari empat alat: Selenium IDE, Selenium RC, *driver* Web Selenium dan Selenium Grid. Selenium melakukan tes seperti fungsional dan regresi dan memiliki potensi menghasilkan skrip pengujian dalam bahasa pemrograman seperti C#, Groovy, Java, Perl, PHP dll. Pengujian mereka dapat berjalan di browser web Firefox dengan platform Windows, Linux, dan OS X. Itu juga dapat diintegrasikan dengan berbagai kerangka kerja untuk menyediakan kerangka kerja hibrida, yang membuat pengujian lebih sederhana.[9]

C. Testing System

Testing atau pengujian merupakan hal yang tidak dapat dipisahkan dari sebuah perangkat lunak maupun sistem. Pengujian perangkat lunak mengacu pada proses mengevaluasi perangkat lunak dengan maksud untuk menemukan kesalahan di dalamnya. Pengujian perangkat lunak adalah teknik yang bertujuan untuk mengevaluasi atribut atau kemampuan suatu program atau produk dan menentukan bahwa itu memenuhi kualitasnya. Pengujian perangkat lunak juga digunakan untuk menguji perangkat lunak untuk faktor kualitas perangkat lunak lain seperti keandalan, kegunaan, integritas, keamanan, kemampuan, efisiensi, portabilitas, rawatan, kompatibilitas, dll [10].

Secara umum, pengujian (pengujian) perangkat lunak dapat dibagi dalam tiga kategori, yaitu: Pengujian saat *Input Data*, Pengujian saat Pemrosesan, dan Pengujian saat *Output* [11]. Pengujian saat *Input Data* adalah tindakan untuk

mengedit dan mengontrol dalam pemasukan data, misalnya: validasi, cek digit. Pengujian saat Pemrosesan untuk melihat apakah program sudah bekerja seperti yang diharapkan. Misalnya: Tabel yang dibuat benar-benar direvisi dengan persyaratan masukan terakhir, dan program kalkulasi benar sesuai dengan rumus yang digunakan. Sedangkan Pengujian saat *Output* berguna untuk membuat laporan yang dibuat telah dibuat dengan format yang benar dan memiliki informasi yang valid/sah.

Tujuan dari proses pengujian perangkat lunak adalah untuk mengidentifikasi semua cacat yang ada dalam produk perangkat lunak. Inilah proses melaksanakan dan mengevaluasi suatu sistem atau komponen sistem dengan cara manual maupun otomatis untuk memverifikasi bahwa itu memenuhi persyaratan yang ditentukan atau untuk mengidentifikasi perbedaan antara hasil yang diharapkan dan hasil aktual.

D. Software Quality Assurance

Quality Assurance atau penjamin mutu merupakan serangkaian kegiatan yang terencana dan sistematis yang memastikan bahwa proses dan produk perangkat lunak sesuai dengan persyaratan, standar, dan prosedur [12]. *Software Quality Assurance* atau SQA memerlukan pola yang terencana dan sistematis dari semua tindakan untuk memenuhi harapan bahwa suatu perangkat lunak sesuai dengan persyaratan teknis yang ditetapkan.

Berdasarkan ISO 9000:2000 (*QMS-Fundamentals and Vocabulary*) (section 3.2.11) *Quality Assurance* “*Part of quality management focused on providing confidence that quality requirements will be fulfilled*”. Jika diterjemahkan *Quality Assurance* terfokus pada pemberian jaminan/keyakinan bahwa persyaratan mutu akan dipenuhi.

E. Website Application

Aplikasi website didefinisikan sebagai aplikasi yang dibangun untuk dijalankan di web. Melalui mereka, pengguna dapat berinteraksi lebih baik dengan situs web [13] dan juga, berkat itu, pengguna memiliki banyak alat yang berjalan online tanpa harus menginstal perangkat lunak di komputer. Tujuan dari aplikasi web adalah untuk menanggapi interaksi pengguna serta memberi mereka kemampuan untuk memanipulasi data. Jadi, dengan aplikasi web, kita dapat membaca, mendengarkan, dan memanipulasi data pada halaman dengan menekan tombol, mengirimkan formulir, mendapatkan tanggapan, pesan instan, melakukan pembayaran, dan banyak lagi.[14]

Secara teknis, website adalah sebuah sistem dengan informasi yang disajikan dalam bentuk teks, gambar, suara dan lain-lain yang tersimpan dalam sebuah server web Internet yang disajikan dalam bentuk hypertext. Informasi web dalam bentuk teks ditulis dalam *Hypertext Markup Language* (HTML). Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG,

PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, *Shockwave*, *Quicktime Movie*, Dunia 3D) [15].

Dengan RPA, aplikasi dapat mengerjakan proses data yang biasa dikerjakan oleh manusia, sehingga dapat mempersingkat pekerjaan berulang dengan skala yang besar. Contohnya seperti melakukan pengujian perangkat lunak aplikasi. Pentingnya pengujian perangkat lunak akan mengacu pada kualitas perangkat lunak itu sendiri.

Pengujian dilakukan dengan mengambil sampel dari salah satu *form* di dalam aplikasi RCTI+ yaitu login user yang digunakan untuk memeriksa sebuah masukan *username* dan *password* dari pengguna. Kemudian membuat sebuah contoh skenario uji yang akan menjadi acuan pada proses pengujian.

Tabel 1. Contoh Tabel Skenario

Module	Input 1	Input 2	Result
Login	refgiufi	12345678	Login gagal
Login	patria	passw0rd	Login berhasil

Pada contoh pengujian ini dilakukan dengan menggunakan input *username* dan *password* secara berturut-turut sesuai dengan skenario yang ada pada table dengan memberikan kondisi dimana *username* dan *password* yang benar adalah “patria dan passw0rd”. Dan setelah memasukkan *username* dan *password* yang berbeda dari yang sudah ditentukan maka akan muncul suatu pesan kesalahan jika hasilnya adalah gagal karena tidak sesuai.

3. HASIL DAN PEMBAHASAN

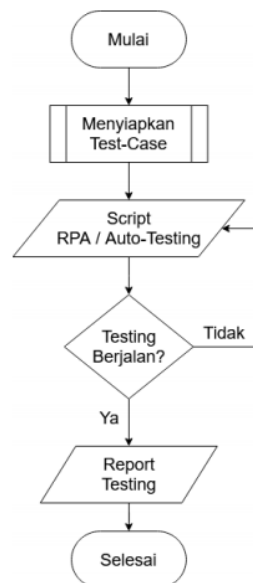
Tim IT dari MNC TV khususnya divisi QC dalam kegiatan pengembangan website RCTI+, saat ini masih menggunakan cara pengujian dengan cara konvensional (tradisional/manual). Dengan cara ini, seorang penguji/tester *Software Quality Assurance* (SQA) akan menguji website RCTI+ secara langsung dan mencoba fitur yang ada satu-persatu untuk memastikan ada atau tidaknya bug di dalam sistem. Hal tersebut tidak diimbangi antara waktu yang diberikan kepada penguji dengan kecepatan pengembangan sistem. Seiring berjalannya waktu, fitur yang ada pada sistem akan terus bertambah, seorang penguji akan melakukan pengujian mulai dari sistem/fitur yang sudah ada ditambah dengan sistem baru yang sedang dikembangkan.

Untuk menjaga kualitas sistem yang dirilis, pengujian/testing adalah pekerjaan utama yang harus dilakukan dalam proses pembuatan/pengembangan aplikasi. Tujuannya adalah untuk menemukan apakah masih ada kesalahan atau *bug* di dalam aplikasi tersebut. Proses ini, bisa dilaksanakan oleh seorang penguji SQA atau oleh seorang SQA *engineer*.

Berdasarkan kondisi awal yang sudah dijelaskan sebelumnya, maka diusulkan suatu sistem yang dapat membantu *tester* untuk melakukan pekerjaan mereka, yaitu pengujian otomatis

(*automated testing*) dengan konsep *Robotic Process Automation* (RPA). Dengan RPA memungkinkan seorang penguji untuk mengotomatiskan tugas-tugas manual rutin tersebut. Sistem pengujian otomatis ini akan melakukan pekerjaan layaknya seorang penguji. Dengan adanya pengujian otomatis ini diharapkan beban seorang penguji menjadi lebih ringan dalam menjalankan *regression test* yang sama berulang kali, tentu ini akan menghemat waktu dan tetap menjaga kualitas sistem sesuai dengan standar yang ada.

Untuk menjalankan konsep RPA, diperlukan tools yang tepat untuk mendukung pengujian berdasarkan aplikasi atau perangkat lunak yang sedang dikembangkan. *Automated testing* sepenuhnya bergantung pada *script* yang berjalan secara otomatis untuk membandingkan hasil aktual dengan hasil yang diharapkan. Hal tersebut membantu SQA *engineer* untuk menentukan apakah suatu aplikasi berfungsi sebagaimana yang diharapkan atau tidaknya. Kelebihannya pengujian otomatis dibuat agar seorang penguji tidak melakukan pengujian secara berulang-ulang karena ada perubahan dari pihak tim *developer*. Bagaimanapun, dalam pembuatan *test script* tidak sesusah membangun aplikasinya. Sistem ini berbasis dengan bahasa pemrograman Python dan alat/*tool* yang digunakan untuk melakukan pengujian otomatis adalah Selenium WebDriver. Adapun proses sistem usulan dapat dilihat pada gambar 2.



Gambar 2. Gambaran sistem yang diusulkan

Berdasarkan Gambar 2 diatas, dapat dijabarkan dari desain desain sistem usulan yaitu :

- 1) Menyiapkan *test-case* berupa hal apa saja yang dilakukan seorang penguji sebelumnya.
- 2) Melakukan pembuatan sistem *Robotic Process Automation* untuk membantu seorang penguji melakukan pekerjaannya.

- 3) Selanjutnya melakukan pengujian apakah sistem tersebut dapat berjalan dengan sebagaimana mestinya atau belum.
- 4) Terakhir membuat *report testing* (laporan pengujian) dengan menggunakan sistem *Robotic Process Automation* ini.

Hal pertama yang dilakukan dalam perancangan sistem ini adalah menyiapkan *test-case*. *Test-case* yang ada diperoleh dari tim IT dari MNC TV khususnya divisi QC. *Test-case* tersebut yang digunakan dalam kegiatan testing sebelumnya, seperti *Test Sign In Form*, *Test Sign Up Form*, dan sebagainya.

3.1 Menyiapkan Test-Case

Tabel 2. Beberapa test-case RCTI+

No.	Sections	Features	Test Scenario	Expectation Detail
1.	Sign In Page	Sign In	1. kosongkan field username 2. klik button sign in	- terdapat text alert dibawah field email/phone number "you have enter an invalid email address or phone number, please try again" , - terdapat text alert dibawah field password "password must be 8 char or more"
2.	Sign In Page	Sign In	1. kosongkan field password 2. klik button sign in	terdapat text alert dibawah field password "password must be 8 char or more"
3.	Sign In Page	Sign In	1. input invalid email (tanpa "@") 2. input invalid password (password kurang dari 8 char) 3. klik button sign in	- terdapat text alert dibawah field password "you have enter an invalid email address, please try again" - terdapat text alert dibawah field password "password must be 8 char or more"
4.	Sign Up Page	Sign Up by Phone Number	1. inputkan valid phone number dan valid password 2. klik button next	masuk ke page interest
5.	Sign Up Page	Text Link Terms & Condition	Klik text link terms & condition	redirect ke webview Terms and Conditions

Dari contoh test pada tabel 2, dapat kita lihat bahwa terdapat 4 kolom harus dicermati (*Sections*, *Features*, *Test Scenario*, dan *Expectation Detail*). Misalnya, *Sections* itu untuk menandakan bagian mana yang akan dilakukan pengujian, *Features* artinya ada fitur apa pada bagian pengujian tersebut, lalu ada *Test Scenario* yaitu apa yang harus dilakukan pada bagian dan fungsi itu, dan terakhir adalah *Expectation Detail* yang menjelaskan hasil yang diharapkan setelah kita melakukan *Test Scenario* tadi.

Untuk memperjelas bagaimana teknis pengujian yang dilakukan selama ini, penulis akan melakukan pengujian no. 1 berdasarkan *test-case* yang ada pada tabel 2. Gambar 3 dibawah ini adalah tampilan awal website RCTI+ (<https://www.rctiplus.com/>).



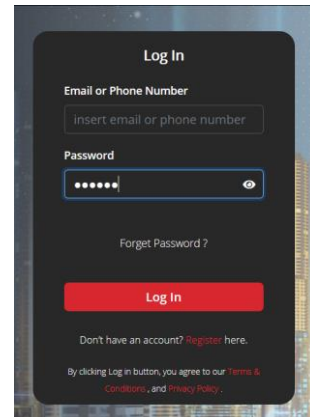
Gambar 3. Tampilan awal website RCTI+

Sesuai dengan *test-case* no. 1 di tabel 2, maka penulis akan membuka halaman Sign In RCTI+ sebagai fitur yang akan diuji (<https://www.rctiplus.com/login>) seperti yang dapat dilihat pada gambar 4 dibawah ini.



Gambar 4. Tampilan Sign-In website RCTI+

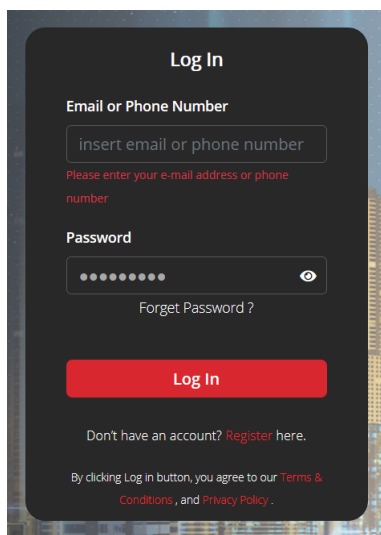
Setelah halaman *Sign-In* terbuka, lalu penulis akan mengikuti *test-scenario* yang ada berdasarkan *test-case* no. 1 yaitu mengosongkan *field username* lalu klik *button sign-in*.



Gambar 5. Form Login RCTI+

Lalu setelah mengklik *button sign-in*, berdasarkan *test-case* no. 1 seharusnya muncul text

dibawah *field username* berupa “*you have enter an invalid email address or phone number, please try again*”. Namun kenyataan saat penulis melakukan pengujian, *text* yang muncul berbeda dari *Expectation Detail* yang ada (gambar 6). Maka dari itu hasil pengujian ini seharusnya dilaporkan karena tidak sesuai dengan *Expectation Detail* yang sudah ditentukan.



Gambar 6. Pengujian Form Login

Jika *test-case* no. 1 selesai maka seorang pengujian harus lanjut ke *test-case* selanjutnya sampai semua *test-case* yang ada sudah dikerjakan. Setelah itu seorang pengujian akan memberikan *report/laporan* bagian mana saja atau *test-case* mana saja yang tidak sesuai dengan *Expectation Detail* yang telah ditentukan.

3.2 Membangun Sistem RPA

Dalam membangun sistem RPA dibutuhkan *library* Python yang akan digunakan untuk melakukan otomatisasi interaksi aplikasi Web, yaitu Selenium WebDriver. Untuk memastikan Selenium sudah terinstall dan siap digunakan bisa dilakukan dengan menggunakan perintah ini di Command Prompt Windows.

```
python -c "import selenium;
print(selenium.__version__)"

C:\Users\refgi>python --version
Python 3.7.4
C:\Users\refgi>python -c "import selenium; print(selenium.__version__)"
3.141.0
```

Gambar 7. Pengecekan versi Python dan Selenium

Selanjutnya kita memerlukan ChromeDriver, ini adalah *tool* yang digunakan dalam navigasi ke halaman web dan input pengguna. ChromeDriver dapat didownload langsung dari website resminya (<https://sites.google.com/a/chromium.org/chromedriver/downloads>) dan ditempatkan dalam satu folder tersendiri untuk digunakan nanti.

Langkah yang perlu dilakukan sebelum menginputkan hal apa saja yang akan dilakukan adalah memanggil *library* yang akan digunakan dalam sistem ini. *Library* yang akan digunakan dalam penelitian ini yaitu Selenium yang tadi sudah kita lakukan pengecekan. Selain Selenium, penulis juga menggunakan *library* “Time” dan “OS” yang sudah ada saat instalasi Python. *Library* “Time” nantinya akan digunakan sebagai waktu jeda saat sistem ini berjalan, sedangkan *library* OS adalah *library* yang akan mengarahkan dimana path ChromeDriver itu berada.

```
from selenium import webdriver
import time
import os
```

Gambar 8. Library yang akan digunakan

Setelah mengaktifkan *library* yang akan digunakan, maka selanjutnya adalah menyiapkan *browser* yang digunakan. *Browser* yang penulis maksud adalah ChromeDriver yang tadi sudah disiapkan (<https://chromedriver.chromium.org/>). Persiapan *browser* dilakukan dengan mengarahkan path direktori dan diikuti pengaturan apa saja yang ingin digunakan dalam browser tersebut. Disini penulis membuat file ChromeDriver dan Script yang sedang dibuat menjadi 1 folder.

```
options = webdriver.ChromeOptions()

dir_path =
os.path.dirname(os.path.realpath(__file__
))

chromedriver = dir_path+"/chromedriver"

os.environ["webdriver.chrome.driver"] =
chromedriver

driver =
webdriver.Chrome(chrome_options=
options, executable_path=chromedriver)
```

Gambar 9. Mengarahkan direktori webdriver yang digunakan

Selanjutnya mengatur input yang telah ditentukan sebelumnya, disini penulis akan membuat *script* untuk *test-case* no. 1 berdasarkan tabel 2 sebelumnya. Yaitu *username* kosong, dan *password* disini penulis menggunakan “Abc123@@”. Username dan *password* tersebut oleh penulis dimasukkan ke variabel “user” untuk username dan “pwd” untuk password.

```
user = ""
pwd = "Abc123@@"
```

Gambar 10. Mengatur input sesuai *test-case*

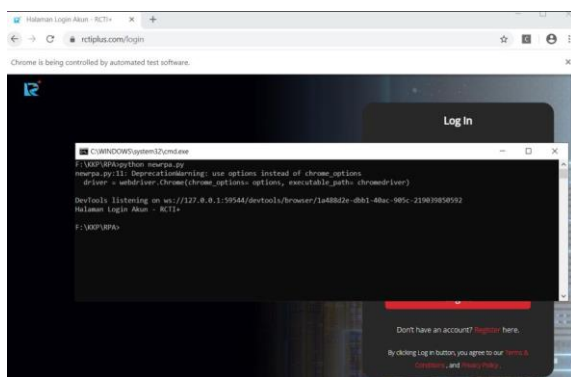
Dibaris selanjutnya mulai masuk *script* untuk otomasinya. Setelah semua persiapan diatas, penulis akan membuat programnya diam selama 5 detik agar memastikan persiapannya tidak ada yang tertinggal.

Lalu browser diarahkan ke url *Sign-In* RCTI+ (<https://www.rctiplus.com/login>). Dan untuk menguji apakah sistem RPA penulis sampai tahap ini, maka pengujian akan mencetak judul website yang sedang diproses saat ini.

```
time.sleep(5)
driver.get("https://ssr.rctiplus.com/login")
print(driver.title)
```

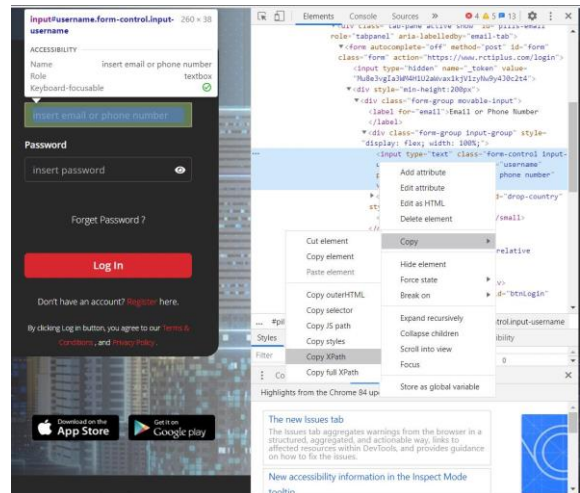
Gambar 11. Mengarahkan ke url *sign-in*

Sampai tahap ini seharusnya program sudah dapat dijalankan. Untuk menjalankannya cukup memasukkan perintah di Command Prompt dan hasilnya ditunjukkan pada gambar 12.



Gambar 12. Percobaan menjalankan program RPA

Dapat dilihat pada gambar 12 bahwa script yang disusun sampai tahap ini dapat berjalan. Namun input yang sudah disiapkan tadi belum bekerja, oleh karena itu selanjutnya yaitu melakukan proses *input* data ke browser. Tapi sebelum melakukan input, penulis akan menambah waktu jeda 3 detik untuk mengantisipasi kecepatan internet melambat. Lalu untuk mencari *form username* dan *password input*, disini penulis menggunakan *syntax XPath*. XPath adalah *syntax* untuk mendeskripsikan lokasi dari elemen website, jadi dengan menggunakan XPath dapat mengarahkan/menentukan bagian tertentu dari sebuah website dan mendapatkan nilainya. Untuk mencari XPath suatu form dapat dilakukan dengan melakukan *inspect element* pada web tersebut (dalam contoh ini: <https://www.rctiplus.com/login>), lalu cari bagian dimana form input itu dan lakukan klik kanan, pilih *Copy* lalu *Copy XPath* (gambar 13).



Gambar 13. Mencari XPath Login Form

XPath yang sudah dicopy tadi ditambahkan pada baris code sebelumnya. Pastikan XPath yang dimasukkan adalah untuk form *username* dan *password*. Dan jangan lupa XPath yang digunakan untuk meng-klik tombol *Log-In*. Disini penulis juga menambahkan waktu jeda setelah menginput data.

```
driver.find_element_by_xpath("//*[@id='username']").send_keys(user)

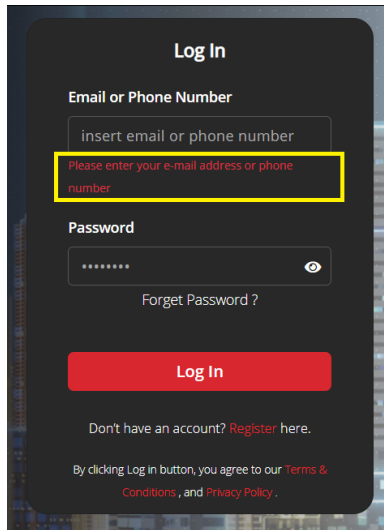
driver.find_element_by_xpath("//*[@id='password']").send_keys(pwd)

time.sleep(3)

driver.find_element_by_xpath("//*[@id='btnLogin']").submit()
```

Gambar 14. Menginput data ke form sesuai XPath

Sampai disini *test-scenario* sudah disusun/dijalankan, sisanya yaitu *Expectation Detail*. Untuk menyamakan data *Expectation Detail* yang disusun dengan hasil yang ada di browser, caranya tidak berbeda dengan mencari bagian form *username/password* tadi. Setelah login dengan test scenario tadi, akan muncul *text alert* atau pesan yang memberitahukan kesalahan yang dilakukan pengguna (gambar 15). *Text alert* inilah yang akan diambil dan disamakan sesuai dengan *Expectation Detail*.



Gambar 15. Text Alert pada Form Login

Text alert tersebut oleh penulis dimasukkan kedalam variabel “elem” dan *Expectation Detail* yang sudah ditentukan sebelumnya dimasukkan ke variabel “notif”. Setelah itu variabel “notif” dan “elem” dibandingkan.

```
time.sleep(3)

elem =
driver.find_element_by_class_name("text-
danger")
notif = "Please enter your e-mail address
or phone number"

if elem.text == notif:
    print("SESUAI TEST-CASE")
else:
    print("TIDAK SESUAI TEST-CASE")
```

Gambar 16. Bagian mencocokkan *Expectation Detail*

Secara keseluruhan, untuk *test-case* no. 1 berdasarkan tabel 2 sebelumnya sudah dapat berjalan melalui script yang sudah disusun. Untuk menjalankannya cukup mudah, yaitu dengan perintah Command Prompt “python namafilename.py”. Dan hasilnya seperti gambar 17.

```
F:\KKP\RPA>python newrpa.py
newrpa.py:11: DeprecationWarning: use options instead of chrome_options
driver = webdriver.Chrome(chrome_options= options, executable_path= chromedriver)

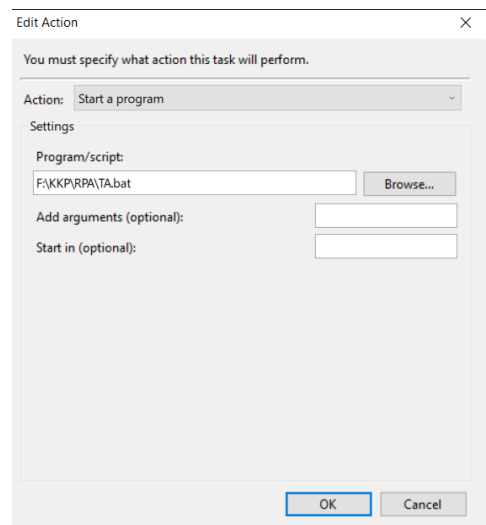
DevTools listening on ws://127.0.0.1:55492/devtools/browser/973f3ec3-89f9-4599-8315-837b9f63d123
Halaman Login Akun - RCTI+
SESUAI TEST-CASE
```

Gambar 17. Menjalankan Script Test-Case 1

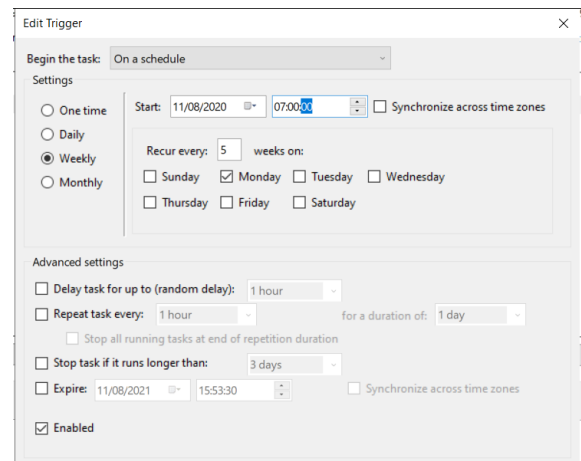
Selain script *Test-Case* 1 yang sudah dijelaskan penulis, disini penulis juga sudah membuat *script* untuk beberapa *test-case* lainnya. Semua *script* yang sudah dibuat itu akan diotomasi agar berjalan setiap waktu yang ditentukan menggunakan *Task Scheduler* di Windows. *Task Scheduler* dapat digunakan untuk mengatur jadwal aktivitas seperti *script* pada rentang waktu yang telah ditentukan.

Agar semua *script* yang sudah dibuat dapat berjalan sesuai dengan waktu yang ditentukan, Semua *script* akan digabung kedalam satu file bat. Setelah itu cukup mengatur *Task Scheduler* agar menjalankan file bat tersebut (gambar 19) dan atur waktu berjalannya sistem RPA ini (gambar 20)

```
F:
cd F:\KKP\RPA
"C:\Users\refgi\AppData\Local\Programs\Python\Python37\python.exe"
"F:\KKP\RPA\testcase1.py"
"C:\Users\refgi\AppData\Local\Programs\Python\Python37\python.exe"
"F:\KKP\RPA\testcase2.py"
```



Gambar 19. Mengatur file .bat dari *Task Scheduler*



Gambar 20. Mengatur waktu RPA dijalankan

3.3 Laporan Testing

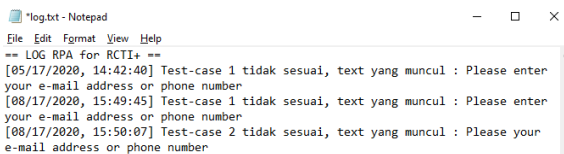
Laporan atau report diperlukan karena walaupun sistem pengujian otomatis sudah berjalan, tapi seorang tester tidak mengetahui dimana letak kesalahan atau hasil yang tidak sesuai dengan *test-case* yang sudah dibuat. Sistem ini akan menggunakan report dalam bentuk *text* (.txt) disimpan pada file log.txt dan kedalam bentuk gambar (.png), juga tak lupa setiap report akan

memberikan waktu dan tanggal kapan laporan ini ditemukan.

```
from datetime import datetime
now = datetime.now()
date_time = now.strftime("%m/%d/%Y,
%H:%M:%S")
f = open('log.txt', 'a')
f.write('\n[' + date_time + ']')
f.write(' Test-case 1 tidak sesuai, text
yang muncul : ' + elem.text)
driver.save_screenshot('Report_TestCase-
1.png')
f.close()
```

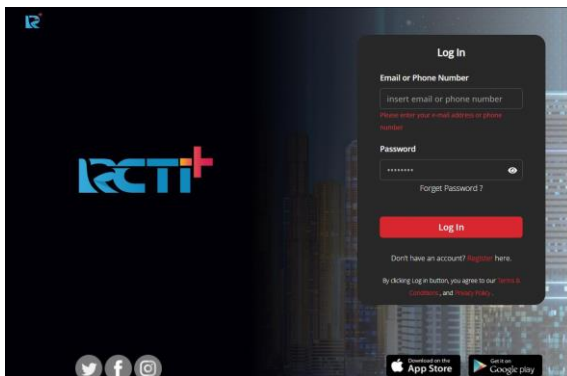
Gambar 21. Source code report testing

Setelah sistem berjalan dan ditemukan hasil yang tidak sesuai dengan Expectation Detail, maka report/laporan dalam bentuk file text dan gambar yang akan diambil seperti gambar 22 dan gambar 23 dibawah ini.



```
*log.txt - Notepad
File Edit Format View Help
-- LOG RPA for RCTI+ --
[05/17/2020, 14:42:40] Test-case 1 tidak sesuai, text yang muncul : Please enter
your e-mail address or phone number
[08/17/2020, 15:49:45] Test-case 1 tidak sesuai, text yang muncul : Please enter
your e-mail address or phone number
[08/17/2020, 15:50:07] Test-case 2 tidak sesuai, text yang muncul : Please your
e-mail address or phone number
```

Gambar 22. Report testing dalam bentuk teks (txt)



Gambar 23. Bentuk Laporan dalam bentuk Gambar (png)

3.4 Hasil

Hasil penelitian ini menunjukkan bahwa tidak terdapat perbedaan yang signifikan antara pengujian dengan cara tradisional maupun menggunakan sistem ini. Dari hasil uji coba sistem ini, dapat diketahui bahwa sistem yang dibangun pada penelitian ini mampu berjalan dengan baik. Sistem dibuat dengan bahasa pemrograman Python, dan mampu melakukan pengidentifikasian layaknya seorang penguji.

Sistem ini dapat melakukan identifikasi perbedaan antara hasil yang sudah ditentukan (*Expectation Detail*) pada tabel 2 dengan hasil yang terjadi pada website yang diujikan (RCTI+). Sistem ini akan memberikan *report*/laporan berupa text dan gambar bagian mana saja yang tidak sesuai dengan *Expectation Detail*. Laporan itulah yang akan

menjadi acuan bagian mana yang akan dilakukan *maintenance*/pemeliharaan. Seperti *Expectation Detail*-nya yang harus diperbaiki ataupun website RCTI+ yang harus diperbarui.

Dalam pengembangan sistem ini juga ada beberapa hal yang dapat mempengaruhi jalannya sistem, seperti:

- Browser yang digunakan (Chrome) tidak boleh di update, karena antara versi ChromeDriver dengan browser tidak boleh berbeda.
- Kecepatan internet dan komputer yang digunakan untuk menjalankan sistem ini akan mempengaruhi, oleh karena itu penulis memberikan waktu jeda agar kerja sistem tidak terlalu padat dan juga memberikan waktu untuk memuat semua hal yang diperlukan oleh sistem ini.

4. KESIMPULAN

Dalam penelitian ini, sebuah sistem untuk menguji perangkat lunak menggunakan RPA secara otomatis telah disajikan. Untuk tujuan ini, kebutuhan pengujian, yang mensimulasikan bagaimana pengujian secara tradisional sebelumnya dilakukan berjalan, dibangun berdasarkan informasi antarmuka pengguna. Sistem ini dapat menjalankan berbagai test-scenario dengan berbagai variasi seperti *statement*, pengkondisian (*if*), dan memberikan laporan. Kemudian sistem ini diuji terhadap sistem yang berjalan agar sesuai dengan kebutuhan, sehingga bisa disimulasikan bagaimana sistem ini berjalan di kemudian hari.

Hasil pengujian menunjukkan evaluasi beban kerja setelah adanya sistem pengujian otomatis, sistem ini mampu mengurangi beban pekerjaan seorang SQA karena pekerjaannya dilakukan oleh robot yaitu melakukan pengujian perangkat lunak dan memberikan laporan hasil pengujian. Dari hasil pengujian menggunakan sistem ini, hasil maupun laporan yang diberikan oleh RPA bisa diterima oleh SQA. Yang berarti seorang SQA bisa menggunakan sistem RPA ini sambil melakukan pekerjaan lainnya karena pengujian yang ia biasa lakukan sekarang dilakukan oleh robot.

Perusahaan yang mengimplementasikan sistem ini juga mendapatkan keuntungan yaitu pemangkasan waktu pengerjaan sehingga hasil keluaran yang diminta dapat dikeluarkan dengan lebih cepat.

DAFTAR PUSTAKA

- [1] D. Fernando and H. Harsiti, "Studi Literatur: Robotic process automation," JSiI (Jurnal Sistem Informasi), vol. 6, no. 1, p. 6, 2019, doi: 10.30656/jsii.v6i1.1071
- [2] C. Tornbohm and R. Dunie, "Market Guide for Robotic Process Automation Software," 2017. https://images.abbyy.com/India/market_guide

- _for_robotic_pro_319864%20(002).pdf (accessed: 24-May-2022).
- [3] E. Kurniadi and H. Ishi, "Harmonic analysis for 4-dimensional real frobenius lie algebras," *Springer Proceedings in Mathematics & Statistics*, pp. 95–109, 2019, doi: 10.1007/978-3-030-26562-5_4
- [4] F. Fitriyani, "Makna Pesan Kesabaran Dalam Sinetron Catatan Harian Aisha Di RCTI," Universitas Islam Negeri Syarif Hidayatullah Jakarta, 2018.
- [5] R. P. Mahardikawati and Nurgiyatna, "Sistem Informasi Industri kecil menengah Pemerintahan Kabupaten Boyolali Berbasis website," *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 53–60, 2020, doi: 10.20884/1.jutif.2020.1.2.13.
- [6] A. Kristanto, *Rekayasa perangkat lunak (Konsep Dasar)*. Gava Media Yogyakarta.
- [7] L. Ivančić, D. Suša Vugec, and V. Bosilj Vukšić, "Robotic Process Automation: Systematic Literature Review," *Business Process Management: Blockchain and Central and Eastern Europe Forum*, vol. 361, pp. 280–295, 2019, doi: 10.1007/978-3-030-30429-4_19.
- [8] G. L., Sigurðardóttir, "Robotic process automation: dynamic roadmap for successful implementation," Reykjavík University, 2018.
- [9] I. Shuaibu, M. Musa, and M. Ibrahim, "Investigation onto the Software Testing Techniques and Tools: An Evaluation and Comparative Analysis," *International Journal of Computer Applications*, vol. 177, no. 23, pp. 24–30, Dec. 2019, doi: 10.5120/ijca2019919685.
- [10] W. Jiang and J. Deng, "Design and Implementation of On-Line Practice System Based on Software Testing," *Journal of Physics: Conference Series*, vol. 1738, no. 1, p. 012115, Jan. 2021, doi: 10.1088/1742-6596/1738/1/012115.
- [11] E. Supriyanto, "Macam – Macam Testing Sistem Informasi," *Dinamik*, vol. 10, no. 1, p. 46, 2005, doi: 10.35315/dinamik.v10i1.10
- [12] Hariyanto, T. Dirgahayu, and H. P. Putro, "Software Quality Assurance pada Perusahaan Pengembang Perangkat Lunak Skala Kecil dan Menengah," *JARTIKA Jurnal Riset Teknologi dan Inovasi Pendidikan*, vol. 3, no. 2, pp. 283–294, Jul. 2020, doi: 10.36765/jartika.v3i2.265.
- [13] C. Nguyen, "New Learning Management Website Application Design," Haaga-Helia University, 2021.
- [14] K. A. Prasetyo and A. K. Nugroho, "Web-Based Technical Supervision And Planning Reports Information System In Cv. Tata Saka Consultant," *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 1, pp. 1-10, 2022, doi: 10.20884/1.jutif.2022.3.1.17.
- [15] I. C. Mauko, N. M. Setiohardjo, and F. P. Noach, "Pengembangan Website Unit Penelitian Dan Pengabdian Kepada Masyarakat Dan Penerapan Jurnal Elektronik Berbasis Open Source Di Politeknik Negeri Kupang," *Jurnal Ilmiah Flash*, vol. 3, no. 2, p. 100, Dec. 2017, doi: 10.32511/jiflash.v3i2.145.

