

Implementasi Automated Testing untuk Proses Pengujian pada SAP ECC dengan Menggunakan Software TestComplete (Studi Kasus PT Telkom Indonesia)

1st Rahman Dilan Syaukanie
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

syaukanied@student.telkomuniversity.ac.id

2nd Wahjoe Witjaksono
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

wahyuwicaksono@telkomuniversity.ac.id

3rd Tien Fabrianti Kusumasari
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia

tienkusumasari@telkomuniversity.ac.id

Abstrak — PT Telkom Indonesia Tbk. PT Telkom Indonesia menggunakan SAP TREMS (*Telkom Revenue Management System*) untuk mengelola berbagai aspek penting dalam bisnis telekomunikasi mereka. Di dalam SAP TREMS, terdapat beberapa proses bisnis yang krusial seperti pencatatan data pelanggan, pembuatan tagihan, pembuatan data pembayaran, dan menampilkan informasi transaksi, yang semuanya memainkan peran penting dalam operasional sehari-hari PT Telkom Indonesia. Namun, pengujian yang saat ini dilakukan pada SAP TREMS oleh perusahaan masih dilakukan secara manual, yang dimana bisa berakibat memakan waktu yang lama dan kurang efisien. Hal ini menimbulkan kebutuhan akan solusi yang lebih efisien. Implementasi pengujian otomatis dengan menggunakan perangkat lunak TestComplete diharapkan dapat mengatasi tantangan ini. Penelitian ini bertujuan untuk menerapkan pengujian otomatis dalam lingkungan SAP TREMS ECC di PT Telkom Indonesia, dengan menggunakan pendekatan *Software Testing Life Cycle* (STLC). Proses ini mencakup analisis kebutuhan, perencanaan pengujian, pengembangan kasus uji, persiapan lingkungan pengujian, eksekusi pengujian, dan penutupan siklus pengujian. Melalui penggunaan *Keyword Test* dan *VBScript* dalam TestComplete, diharapkan pengujian dapat dijalankan berulang, sehingga menghasilkan pengujian yang lebih efisien. Hasil dari penelitian ini juga akan melakukan perbandingan waktu pengujian antara metode manual dan otomatis, untuk menunjukkan bahwa pengujian otomatis menggunakan TestComplete dapat mengurangi waktu yang diperlukan untuk pengujian.

Kata kunci—pengujian perangkat lunak, pengujian otomatis, SAP TREMS, *Software Testing Life Cycle*, TestComplete.

I. PENDAHULUAN

Sistem perangkat lunak telah menjadi suatu komoditas yang vital dari segi profesional dan pribadi. Apa yang dulunya dianggap sebagai sesuatu yang tidak terlalu penting,

sekarang telah menjadi sangat berperan dalam kehidupan kita hingga pada titik di mana kita sangat bergantung padanya [1]. Sebagai hasilnya, pengguna yang merasa terdorong untuk mempercayai sistem-sistem ini, yang dapat menjadi berisiko jika sistem perangkat lunak tidak berfungsi sebagaimana mestinya. Dalam situasi seperti ini, perangkat lunak yang tidak berfungsi sebagaimana mestinya dapat menyebabkan banyak ketidaknyamanan bagi pengguna. Terkadang, hal ini dapat berujung pada kebocoran informasi sensitif, atau dalam kasus perangkat lunak yang digunakan di sektor medis, dapat mengancam nyawa kita [2]. Hal ini menyoroti betapa pentingnya jaminan kualitas untuk mencegah kegagalan perangkat lunak.

Salah satu metode untuk mencapai tujuan ini adalah dengan melakukan pengujian perangkat lunak, yang memegang peran penting dalam proses pengembangan perangkat lunak dengan tujuan meningkatkan mutu perangkat lunak [1]. Pengujian perangkat lunak adalah proses yang bertujuan untuk mengevaluasi program perangkat lunak dengan maksud menemukan kesalahan atau kekurangan dalam perangkat lunak. Pengujian dilakukan untuk memastikan bahwa perangkat lunak menjalankan tujuannya dengan benar, mengakses, mencapai, dan menjaga kualitas perangkat lunak [3]. Pengujian perangkat lunak dapat dilakukan secara manual atau otomatis menggunakan alat pengujian perangkat lunak. Dalam pengujian manual, seorang penguji bertindak sebagai pengguna akhir, menguji fitur-fitur aplikasi. Namun, metode ini memakan waktu dan tidak selalu efektif dalam mendeteksi semua kesalahan. Pengujian manual lebih sesuai untuk perusahaan kecil yang memiliki keterbatasan sumber daya finansial. Pengujian otomatis mengatasi tantangan ini dengan memungkinkan pembuatan skenario pengujian yang dapat diulang dan dieksekusi sesuai kebutuhan. Dalam menghadapi kompleksitas pengembangan perangkat lunak yang semakin meningkat, tim pengembang perangkat lunak perlu mengadopsi alat pengujian otomatis untuk memastikan

kualitas dan fungsionalitas perangkat lunak secara lebih efektif, mengoptimalkan waktu, biaya, dan kegunaan [4].

Software Testing Life Cycle (STLC) hadir sebagai metode yang sistematis dan terstruktur untuk melakukan pengujian perangkat lunak (Ardan et al., 2021). Kerangka ini dirancang untuk memastikan bahwa setiap tahap dalam pengujian perangkat lunak dilakukan dengan cermat dan menyeluruh, sehingga dapat mendeteksi dan mengeliminasi bug serta kesalahan dalam perangkat lunak sebelum dirilis kepada pengguna akhir [5].

Tantangan yang dihadapi oleh PT Telkom Indonesia Tbk, perusahaan telekomunikasi terkemuka di Indonesia, mencakup berbagai aspek dalam bisnis telekomunikasi, termasuk proses registrasi pelanggan, pembuatan tagihan, hingga menampilkan setiap transaksi. Dalam upaya mengatasi tantangan ini, perusahaan telah mengembangkan Telkom Revenue Management System (TREMS), sebuah solusi perangkat lunak berbasis SAP, yang dirancang khusus untuk mengelola transaksi dari berbagai layanan telekomunikasi yang mereka tawarkan kepada pelanggan. SAP, yang merupakan singkatan dari System, Application, and Product in Data Processing, adalah perangkat lunak yang dikembangkan oleh SAP AG dalam ranah Enterprise Resource Planning (ERP). SAP adalah suatu solusi perangkat lunak yang merevolusi praktik pengumpulan, penyimpanan, dan penyebaran data serta meningkatkan kapasitasnya dalam pengambilan keputusan [6]. Keberadaan teknologi seperti SAP memegang peranan penting dalam menyokong efisiensi dalam pengelolaan pada transaksi ini. Dalam menjalankan proses sehari-hari, SAP TREMS menggunakan modul Revenue Management and Contract Accounting (RMCA). Modul ini memungkinkan perusahaan untuk membantu dalam pencatatan data pelanggan, pembuatan tagihan, dan pencatatan transaksi penerimaan dana dengan baik.

Dalam pengembangan perangkat lunak SAP TREMS di PT Telekomunikasi Indonesia, tahap pengujian perangkat lunak menjadi penting karena menghadapi tingkat kompleksitas dan skalabilitas yang tinggi, dan dimana pengujian yang dilakukan saat ini masih dilakukan secara manual yang bisa berakibat memakan waktu yang lama dan kurang efisien. Untuk mengatasi masalah ini, terutama kurangnya efisiensi dalam pengujian manual, pengujian otomatis dapat menjadi solusi yang sangat tepat. Pengujian otomatis memungkinkan tim pengujian untuk menggunakan berbagai alat untuk menjalankan skenario pengujian berulang kali [7]. Keunggulan ini diharapkan dapat memberikan kontribusi signifikan pada peningkatan efisiensi dalam proses pengujian, mempercepat waktu siklus pengujian, dan memastikan kualitas perangkat lunak yang lebih baik.

TestComplete muncul sebagai elemen kunci yang memungkinkan perusahaan untuk mengoptimalkan efisiensi proses pengujian perangkat lunak. Alat pengujian ini dirancang oleh SmartBear untuk mengotomatisasi aplikasi desktop, mobile, dan web. Dengan TestComplete, pengujian perangkat lunak dapat dengan cepat membangun dan menjalankan kasus pengujian otomatis untuk berbagai platform. TestComplete adalah alat yang ramah pengguna,

memungkinkan tim pengujian perangkat lunak dari berbagai tingkatan keahlian untuk dengan mudah membangun dan menjalankan kasus pengujian otomatis. Hal ini memungkinkan percepatan siklus pengujian, peningkatan kualitas perangkat lunak, dan memastikan aplikasi berfungsi sesuai desain [8].

Dalam rangka mencapai tujuan tersebut, penelitian ini akan melakukan analisis mendalam terhadap kasus PT Telkom Indonesia. PT Telkom Indonesia memiliki jaringan pelanggan yang luas, menjadikan proses pencatatan data pelanggan, pembuatan tagihan, dan pencatatan transaksi penerimaan dana sebagai elemen utama dalam jalannya operasional perusahaan yang dimana akan menjadi fokus pengujian tersebut. Implementasi pengujian otomatis dengan software TestComplete di lingkungan SAP TREMS PT Telkom Indonesia dapat memberikan wawasan berharga tentang bagaimana teknologi ini dapat meningkatkan efisiensi proses pengujian perangkat lunak. Penelitian ini juga akan mempertimbangkan faktor-faktor seperti kecepatan dalam pengujian dan dilakukan perbandingan waktu pengujian untuk memberikan panduan yang lebih baik terkait dengan penggunaan teknologi pengujian.

II. KAJIAN TEORI

Berikut merupakan kajian teori yang digunakan dalam penelitian ini.

A. Pengujian Perangkat Lunak

Pengujian Perangkat Lunak adalah metode untuk memeriksa apakah produk perangkat lunak sebenarnya sesuai dengan persyaratan yang diharapkan dan untuk memastikan bahwa produk perangkat lunak bebas dari cacat. Ini melibatkan eksekusi komponen perangkat lunak/sistem menggunakan alat manual atau otomatis untuk mengevaluasi satu atau lebih properti yang diinginkan. Tujuan pengujian perangkat lunak adalah untuk mengidentifikasi kesalahan, kesenjangan, atau persyaratan yang hilang dibandingkan dengan persyaratan sebenarnya [9].

B. Pengujian Otomatis Perangkat Lunak

Pengujian Otomatis Perangkat Lunak adalah proses menulis program dalam bahasa pemrograman atau Skriping apa pun yang menduplikasi langkah-langkah uji manual dengan bantuan alat bantu otomatisasi eksternal. Ini melibatkan pembuatan tool kit untuk menguji kode sumber yang sudah diimplementasikan. Tujuannya adalah membuat proses pengujian lebih otomatis [7].

C. SAP ECC

SAP ERP Central Component (SAP ECC) adalah Enterprise Resource Planning (ERP) yang berbasis on-premises. ERP mengintegrasikan informasi digital yang dihasilkan di satu area bisnis dengan data dari area lain dalam bisnis yang sama secara real-time. Ini berarti pembaruan di satu area bisnis, seperti penjualan, akan memicu pembaruan di area terkait, seperti persediaan. Memiliki pandangan yang terpadu terhadap sumber daya perusahaan memungkinkan para manajer membuat keputusan berbasis data yang mengoptimalkan proses bisnis inti [9].

D. TestComplete

TestComplete merupakan alat pengujian yang dirancang oleh SmartBear untuk mengotomatisasi aplikasi desktop, mobile, dan web. Dengan TestComplete, tim pengujian perangkat lunak dapat dengan mudah membangun dan menjalankan kasus pengujian otomatis untuk berbagai platform. Alat ini memungkinkan pengguna dari berbagai tingkatan keahlian untuk membuat dan menjalankan kasus pengujian otomatis dengan cepat. Dukungan terhadap berbagai bahasa Skriping seperti VBScript, Python, JScript, C++, dan DelphiScript memperluas fleksibilitas dalam pembuatan dan eksekusi tes. Selain itu, TestComplete juga menyediakan fitur untuk merekam dan memutar kembali proses pengujian, memungkinkan pengguna untuk mengotomatisasi pengujian, pengujian unit, pengujian kata kunci, dan pengujian regresi. Ini memberikan kontribusi dalam mempercepat siklus pengujian, meningkatkan kualitas perangkat lunak, dan memastikan aplikasi berfungsi sesuai desain [8]. Pada pengujian otomatis TestComplete terdapat pengujian Keyword Test, ini menggunakan Keyword-driven Tests, dimana tes berbasis kata kunci dalam TestComplete adalah serangkaian tindakan yang ditentukan oleh kata kunci tertentu [10].

III. METODE

Penelitian ini diawali dengan identifikasi masalah dan menentukan tujuan penelitian, serta didukung dengan studi literatur dan pengumpulan data melalui wawancara dan observasi. Penelitian ini akan meneliti implementasi pengujian otomatis dengan menggunakan *software* TestComplete untuk meningkatkan efisiensi pengujian pengujian di PT Telkom Indonesia Tbk dengan menerapkan metode *Software Testing Life Cycle* (STLC) guna mendukung proses perencanaan, perancangan, dan pelaksanaan proses pengujian perangkat lunak. STLC terdiri dari beberapa tahapan yang membantu dalam mengorganisir dan menyempurnakan proses pengujian. Tahapan-tahapan dalam STLC meliputi *Requirement Analysis*, *Test Planning*, *Test case Development*, *Environment Setup*, *Test Execution* dan *Test Cycle Closure*. Berikut merupakan penjabaran dari setiap tahapan yang dilakukan

A. Requirement Analysis

Tahap Requirement Analysis melibatkan pengumpulan informasi tentang persyaratan perangkat lunak. Pada tahap ini dilakukan pemahaman persyaratan produk perangkat lunak dalam hal fungsi, spesifikasi, dan desainnya serta dilakukan perancangan *Requirement Traceability Matrix* untuk memastikan setiap kebutuhan dan fitur yang diidentifikasi dapat diuji dan dilacak dengan efektif.

B. Test planning

Test planning merupakan tahapan di mana rencana pengujian disusun secara terperinci untuk mengarahkan dan mengelola seluruh proses pengujian. Pada tahap ini, penguji akan menentukan tujuan pengujian, ruang lingkup pengujian, jenis pengujian yang akan dilakukan, serta alat pengujian yang akan digunakan

C. Test case Development

Pada tahap *Test case Development*, dilakukan pembuatan dan pengerjaan *Test case* serta skrip pengujian. Setelah rencana pengujian siap, penguji mulai merancang *Test case* untuk setiap fitur yang akan diuji. Selain itu, data uji yang relevan juga akan disiapkan untuk mendukung setiap *Test case* yang telah dirancang. Dalam tahap ini juga akan dilakukan pembuatan Skrip otomatisasi pengujian yang akan digunakan pada TestComplete.

D. Environment setup

Pada tahap *environment setup* akan dilakukan pemeriksaan bahwa Environment Test yang akan dijalankan baik hardware maupun software, berjalan sesuai dengan rencana. Tujuan dari tahap ini adalah untuk menciptakan lingkungan pengujian yang dengan lancar dan menghasilkan hasil yang sesuai.

E. Test Execution

Pada tahap ini, tim penguji mulai mengeksekusi pengujian berdasarkan Test case yang sudah dibuat. Setiap pengujian dijalankan menggunakan data uji untuk memastikan bahwa kasus tersebut memberikan hasil yang diinginkan. Software dianggap lulus pengujian apabila dapat memberikan hasil yang sesuai harapan. Tester akan memberikan keterangan pass untuk Test case yang berhasil dieksekusi sesuai dengan harapan, untuk Test case yang tidak sesuai, maka akan diberi status fail.

F. Test Cycle Closure

Test Cycle Closure atau Penutupan Siklus Tes, seperti yang ditunjukkan oleh namanya, adalah tahap penutup dari *Software Testing Life Cycle* (STLC). Ketika tahap pelaksanaan pengujian selesai, dilakukan evaluasi pengujian berdasarkan RTM serta dilakukan perbandingan waktu pengujian.

IV. HASIL DAN PEMBAHASAN

Berikut merupakan penjabaran mengenai hasil dan pembahasan yang telah dilakukan dalam penelitian ini dengan menerapkan metode *Software Testing Life Cycle* :

A. Requirement Analysis

Requirement Analysis adalah langkah pertama dari *Software Testing Life Cycle* (STLC). Pada tahap ini, penguji melakukan analisis proses bisnis untuk memahami alur kerja yang ada, analisis Business Requirement untuk memastikan semua kebutuhan bisnis diidentifikasi, analisis Test case yang akan dibuat untuk menentukan skenario pengujian, dan perancangan *Requirement Traceability Matrix* untuk melacak setiap kebutuhan dari tahap analisis hingga pengujian. Selain itu, penguji akan berkolaborasi dengan pihak divisi Information Technology untuk memastikan pemahaman yang baik tentang produk yang akan diuji.'

TABEL 1
REQUIREMENT TRACEABILITY MATRIX

Requirement Traceability		Business Requirement				
		BR1	BR2	BR3	BR4	BR5
Test case	TC1					
	TC2					
	TC3					
	TC4					
	TC5					
	TC6					
	TC7					
	TC8					
	TC9					
	TC10					
	TC11					
	TC12					
	TC13					
	TC14					

B. Test Planning

Test Planning merupakan tahap dalam *Software Testing Life Cycle* di mana semua rencana pengujian akan ditentukan. Pada tahap perencanaan pengujian, proses merancang metode pengujian yang paling sesuai dimulai, serta merencanakan upaya pengujian dari berbagai aspek, mulai dari menentukan tujuan pengujian, ruang lingkup, jenis pengujian yang akan dilakukan, pemilihan alat pengujian. Sistem yang akan diuji adalah SAP TREMS (*Telkom Revenue Management System*).

Tujuan utama dari pengujian ini adalah memastikan bahwa SAP TREMS beroperasi sesuai dengan harapan dan memenuhi kebutuhan bisnis Telkom Indonesia. Dalam kerangka ini, pengujian bertujuan untuk mengidentifikasi cacat atau kekurangan dalam fungsionalitas sistem. Pengujian ini juga bertujuan untuk mengukur waktu respons sistem guna memastikan bahwa operasi sistem berlangsung dengan cepat dan efisien, sehingga dapat mendukung proses bisnis secara optimal. Perbandingan waktu pengujian juga dilakukan berdasarkan metode yang berbeda untuk mengetahui metode pengujian mana yang paling efisien dan mengetahui selisih waktu pengujian.

Ruang lingkup pengujian akan terfokus pada fitur-fitur utama dan kinerja pada SAP TREMS yang mencakup pencatatan data pelanggan, pembuatan tagihan, pembuatan data pembayaran, dan menampilkan transaksi.

Pengujian SAP TREMS ini akan menggunakan jenis pengujian *Functional Testing*, yang berfokus pada pengujian fitur dan fungsi perangkat lunak untuk memastikan bahwa mereka bekerja sesuai dengan spesifikasi yang ditetapkan, serta *Performance Testing*, yang bertujuan untuk mengukur waktu respons. Teknik pengujian yang digunakan adalah *Blackbox Testing*, yang menguji perangkat lunak berdasarkan input dan output tanpa memerlukan pemahaman tentang implementasi internal dari perangkat lunak. Pengujian ini akan memastikan bahwa semua fungsi utama dalam sistem, seperti pencatatan data pelanggan, pembuatan tagihan, pembuatan data pembayaran, dan menampilkan

transaksi, berfungsi dengan benar dan kinerja sistem tetap optimal.

Dalam pengujian ini, alat pengujian yang dipilih adalah *TestComplete*. Keputusan ini didasarkan pada kemampuan alat ini dalam otomatisasi pengujian, yang akan mendukung pengujian berbagai skenario penggunaan dalam SAP TREMS. Dengan menggunakan *TestComplete*, dapat dibuat skrip pengujian yang efisien untuk memastikan pengujian yang menyeluruh terhadap fungsionalitas sistem yang dituju.

C. Test case Development

Tahap *Test case Development* dimulai setelah tahap *Test Planning* selesai. Dalam tahap ini, peneliti akan membuat *Test case* yang lengkap yang sesuai dengan RTM dan rencana pengujian sebelumnya. Penguji juga akan mempersiapkan data pengujian yang diperlukan untuk menjalankan kasus uji yang telah dirancang. *Test case* mencakup prosedur yang diperlukan untuk menguji kinerja sistem dengan hasil yang diharapkan dan nyata. Setiap *Test case* mengandung skenario positif dan negatif, yang dikonfirmasi oleh RTM, yang menggabungkan persyaratan untuk memastikan pengujian dilakukan secara menyeluruh. Selain itu, semua data uji juga akan disiapkan untuk mendukung setiap kasus *Test case* yang telah dirancang. Pada tahap *Test case Development* ini karena pengujian ini menggunakan pengujian otomatis, maka penguji juga akan membuat skrip pengujian pada setiap *Test case*.

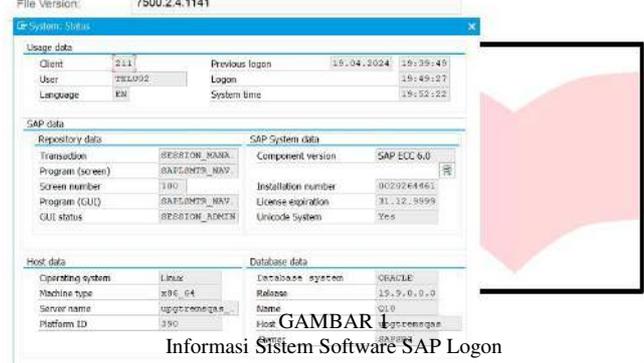
TABEL 2
TEST CASE

Test Case ID	Deskripsi Test Case
TC1	Login berhasil
TC2	Login Gagal dengan password salah
TC3	Create Customer Data berhasil
TC4	Create Customer Data gagal dengan Field I_CCA tidak terisi
TC5	Request Number berhasil
TC6	Request Number gagal dengan Field ZCUSTOMERCODE tidak terisi
TC7	Berhasil membuat billing
TC8	Gagal membuat billing dengan Field ZIDSC tidak terisi
TC9	Request Billing berhasil
TC10	Request Billing gagal dengan Field ZCUSTOMERID tidak diisi
TC11	Berhasil membuat data pembayaran
TC12	Gagal membuat data pembayaran dengan ZSESSIONNO tidak diisi
TC13	Berhasil mendisplay transaksi
TC14	Gagal mendisplay transaksi dengan Contract Account tidak ditemukan

D. Environment Setup

Tahap *Environment Setup* merupakan tahapan dalam proses pengujian yang menentukan kondisi perangkat lunak serta perangkat keras di bawah mana produk kerja diuji dan ini merupakan aspek yang penting dalam pengujian. Dalam lingkungan SAP TREMS, tim pengembang telah mempersiapkan lingkungan uji yang diperlukan, selanjutnya yang tim pengujian perlu lakukan adalah verifikasi untuk memastikan bahwa semua perangkat lunak dan sistem yang diperlukan telah terinstal dan berfungsi sesuai dengan

spesifikasi yang ditetapkan, sehingga pengujian dapat berjalan dengan lancar dan akurat. Pada prakteknya, hardware yang digunakan untuk pengujian adalah menggunakan laptop pengujian sendiri, sementara itu, software dan sistem pengujian telah disediakan oleh tim pengembang.



Informasi Sistem Software SAP Logon



GAMBAR 2 Informasi Sistem Software SAP Logon

E. Environment Setup

Pada tahap ini proses pengujian dilakukan berdasarkan Test case yang sudah dibuat dan dijalankan pada Environment yang sudah di tentukan sebelumnya. Pada tahap ini akan menampilkan hasil yang diharapkan dari sistem yang diuji dan hasil dari proses pengujian itu sendiri. Pada tahap Test Execution ini akan menampilkan table Test case dengan kolom Pass/Fail yang sudah terisi yang menandakan berhasil tidak nya pengujian. Serta dilakukan Performance Testing untuk mengukur waktu Test Execution menggunakan tiga metode: Test manual, Automation Testing dengan Keyword Test, dan Automation Testing dengan VBScript.

TABEL 3 TEST EXECUTION

Test Case ID	Deskripsi Test Case	Hasil Pengujian (Pass/Fail)
TC1	Login berhasil	Pass
TC2	Login Gagal dengan password salah	Pass
TC3	Create Customer Data berhasil	Pass
TC4	Create Customer Data gagal dengan Field I_CCA tidak terisi	Pass
TC5	Request Number berhasil	Pass

TC6	Request Number gagal dengan Field ZCUSTOMERCODE tidak terisi	Pass
TC7	Berhasil membuat billing	Pass
TC8	Gagal membuat billing dengan Field ZIDSC tidak terisi	Pass
TC9	Request Billing berhasil	Pass
TC10	Request Billing gagal dengan Field ZCUSTOMERID tidak diisi	Pass
TC11	Berhasil membuat data pembayaran	Pass
TC12	Gagal membuat data pembayaran dengan ZSESSIONNO tidak diisi	Pass
TC13	Berhasil mendisplay transaksi	Pass
TC14	Gagal mendisplay transaksi dengan Contract Account tidak ditemukan	Pass

F. Test Cycle Closure

Test Cycle Closure adalah tahap akhir dalam Software Testing Life Cycle (STLC) di mana semua aktivitas pengujian telah diselesaikan dan akan dievaluasi. Di tahap ini, Requirement Traceability Matrix (RTM) yang sudah terisi akan ditampilkan untuk memeriksa apakah tidak ada pengujian yang terlewatkan, lalu menampilkan hasil pengujian dengan status pass atau fail untuk setiap Test case. Selain itu, perbandingan antara pengujian manual dan otomatis juga akan disajikan untuk menilai efisiensi masing-masing metode

TABEL 4 REQUIREMENT TRACEABILITY MATRIX

Requirement Traceability	Business Requirement				
	BR1	BR2	BR3	BR4	BR5
TC1	✓				
TC2	✓				
TC3		✓			
TC4		✓			
TC5		✓			
TC6		✓			
TC7			✓		
TC8			✓		
TC9			✓		
TC10			✓		
TC11				✓	
TC12				✓	
TC13					✓
TC14					✓

Pada tabel 4 Requirement Traceability Matrix di atas menunjukkan hasil pengujian untuk setiap Test case yang telah dilakukan. Dari total 14 Test case yang ada, masing-masing Test case telah berhasil diuji dan hasilnya sesuai dengan persyaratan bisnis yang telah ditetapkan. Setiap ceklis dalam tabel menunjukkan bahwa Test case tersebut memenuhi kebutuhan bisnis yang diidentifikasi, yaitu BR1, BR2, BR3, BR4, dan BR5. Ini memastikan bahwa semua aspek penting dari sistem telah diuji dengan baik dan memenuhi standar yang diharapkan.

TABEL 5
PERBANDINGAN WAKTU PENGUJIAN

Test Case ID	Pengujian Manual	Pengujian Otomatis <i>Keyword Test</i>	Pengujian Otomatis <i>VBScript</i>
TC1	24.6 detik	19.3 detik	15 detik
TC2	22.6 detik	16 detik	15 detik
TC3	385.6 detik	27.3 detik	26.3 detik
TC4	356 detik	26 detik	26 detik
TC5	96 detik	5 detik	5 detik
TC6	83.3 detik	5 detik	4.6 detik
TC7	220 detik	29 detik	28.3 detik
TC8	200.3 detik	28.6 detik	27.3 detik
TC9	122.3 detik	20 detik	18 detik
TC10	102 detik	18 detik	16.6 detik
TC11	355.6 detik	54.3 detik	41.3 detik
TC12	329.3 detik	41 detik	41 detik
TC13	25 detik	2.6 detik	2 detik
TC14	18.6 detik	2 detik	2 detik
Total	39 menit 41.2 detik	5 menit 16.2 detik	5 menit 11.9 detik

Tabel 5 Waktu Pengujian menunjukkan perbandingan waktu pengujian antara pengujian manual, pengujian otomatis menggunakan *Keyword Test*, dan pengujian otomatis menggunakan *VBScript*. Dari grafik tersebut, terlihat jelas bahwa pengujian manual memakan waktu paling lama, mencapai hampir 40 menit. Sebaliknya, pengujian otomatis baik dengan *Keyword Test* maupun *VBScript* membutuhkan waktu yang jauh lebih singkat, masing-masing sekitar 5 menit. Grafik ini mengilustrasikan efisiensi dari pengujian otomatis dibandingkan dengan pengujian manual, dengan *VBScript* sedikit lebih cepat daripada *Keyword Test*. Hal ini menunjukkan bahwa penggunaan metode otomatisasi dalam pengujian dapat mengurangi waktu yang diperlukan untuk melakukan pengujian, yang sangat bermanfaat dalam proses pengembangan perangkat lunak.

V. KESIMPULAN

Berdasarkan hasil penelitian yang mencakup analisis dan implementasi pengujian otomatis untuk proses bisnis pada SAP TREMS menggunakan *software TestComplete* di PT Telkom Indonesia, serta perbandingan antara pengujian manual, pengujian otomatis dengan *Keyword Test*, dan pengujian otomatis dengan VB Skrip dengan metodologi *Software Testing Life Cycle (STLC)*, dapat ditarik kesimpulan bahwa pada implementasi pengujian otomatis menggunakan perangkat lunak *TestComplete* pada SAP TREMS. Semua 14 test case yang direncanakan telah dilakukan secara menyeluruh tanpa menemukan defect selama proses pengujian. Implementasi ini menggunakan metode *Software Testing Life Cycle (STLC)*, memastikan

setiap tahap pengujian dilakukan secara sistematis. Penggunaan *TestComplete* membawa dampak positif dengan membuat proses pengujian lebih efisien, menghemat waktu dan sumber daya yang sebelumnya dihabiskan untuk tugas-tugas manual yang repetitif. Data waktu pengujian menunjukkan bahwa pengujian otomatis menggunakan *Keyword Test* memakan waktu 5 menit 16,2 detik, sedangkan dengan *VBScript* hanya 5 menit 11,9 detik. Sebagai perbandingan, pengujian manual untuk keseluruhan proses memakan waktu 39 menit 41,2 detik. Dengan demikian, pengujian otomatis, baik menggunakan *Keyword Test* maupun *VBScript*, telah meningkatkan efisiensi waktu pengujian secara signifikan.

REFERENSI

- [1] T. Dias, A. Batista, E. Maia, and I. Praça, "TestLab: An Intelligent Automated Software Testing Framework." arXiv, Jun. 06, 2023. Accessed: Dec. 07, 2023. [Online]. Available: <http://arxiv.org/abs/2306.03602>
- [2] A. Lal and G. Kumar, "Intelligent Testing in Software Industry," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India: IEEE, Jul. 2021, pp. 01–06. doi: 10.1109/ICCCNT51525.2021.9580012.
- [3] M. A. Umar, "Comprehensive study of software testing: Categories, levels, techniques, and types," preprint, Jun. 2020. doi: 10.36227/techrxiv.12578714.
- [4] D. Ateşoğulları and A. Mishra, "AUTOMATION TESTING TOOLS: A COMPARATIVE VIEW," vol. 12, 2020.
- [5] Y. Altaf, "Software Testing Life Cycle - A Beginner's Guide," GoodCore Blog. Accessed: Nov. 29, 2023. [Online]. Available: <https://www.goodcore.co.uk/blog/software-testing-life-cycle/>
- [6] L. T. Reddi, "TRANSFORMING MANAGEMENT ACCOUNTING: ANALYZING THE IMPACTS OF INTEGRATED SAP IMPLEMENTATION," *Int. Res. J. Mod. Eng. Technol. Sci.*, Aug. 2023, doi: 10.56726/IRJMETS44178.
- [7] D. Vijayasree and A. Arun, "A Review on the Process of Automated Software Testing," Sep. 2022.
- [8] M. Viswanath V S, "TestComplete Tutorial for Beginners | Learn Software Testing," Mindmajix. Accessed: Mar. 27, 2024. [Online]. Available: <https://mindmajix.com/testcomplete-tutorial>
- [9] D. Daniel, "What is SAP ERP Central Component (SAP ECC)?," SearchSAP. Accessed: Nov. 29, 2023. [Online]. Available: <https://www.techtarget.com/searchsap/definition/SAP-ERP-Central-Component-SAP-ECC>
- [10] "Keyword Tests | TestComplete Documentation." Accessed: May 26, 2024. [Online]. Available: <https://support.smartbear.com/testcomplete/docs/keyword-testing/index.html>