

## BAB 3

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kegiatan pada proyek T4V (*Transform for Value*) di PT Freeport Indonesia, posisi ini berperan sebagai Quality Assurance (QA) dengan lingkup fungsi Business Analyst. Posisi tersebut berada dalam tim *testing* di bawah Divisi MIS serta berkoordinasi langsung dengan *Module Lead* dan *Functional Consultant* dari area MM, FI/CO, dan PM. QA bertanggung jawab mendukung pelaksanaan pengujian sistem SAP S/4HANA secara menyeluruh, baik pada proses fungsional dasar maupun integrasi antarmodul, untuk memastikan bahwa solusi yang dikembangkan telah memenuhi kebutuhan bisnis serta selaras dengan desain sistem.

Struktur koordinasi dilakukan secara vertikal dan horizontal. Secara vertikal, hasil pelaksanaan pengujian dan analisis *defect* disampaikan kepada *Module Lead*, kemudian diteruskan dalam forum evaluasi lintas tim, termasuk *Technical Consultant* dan *Project Manager*. Secara horizontal, QA berkoordinasi erat dengan *Functional Consultant* dan *Key User* di setiap departemen dalam proses validasi skenario bisnis serta penelusuran akar penyebab (*root cause*) terhadap *defect* yang ditemukan pada *User Acceptance Test* (UAT) maupun *Integration Test*.

Dalam beberapa skenario pengujian yang memerlukan validasi teknis lebih mendalam, QA bekerja sama dengan *Technical Consultant* untuk memastikan bahwa perbaikan (*fix*) telah memenuhi standar yang ditetapkan dan tidak berdampak negatif terhadap alur bisnis lainnya. QA juga terlibat dalam penyusunan *test scenario* serta proses *cross-checking* terhadap data *posting* untuk memastikan integrasi dan akurasi hasil transaksi antarmodul.

Untuk mendukung aktivitas kolaborasi, komunikasi harian dilakukan melalui Microsoft Teams, sedangkan SAP Solution Manager digunakan sebagai platform utama untuk dokumentasi *requirement*, skenario pengujian, pencatatan *defect*, serta pelacakan status perbaikannya. Power BI turut dimanfaatkan untuk menyusun *monitoring dashboard* yang menampilkan status terkini *defect*, progres pengujian, serta kesiapan sistem menjelang fase *go-live*. Rapat koordinasi rutin dilakukan dalam *daily defect meeting* untuk menyampaikan pembaruan progres, kendala teknis, serta prioritas penyelesaian isu.

Selama keterlibatan dalam proyek, QA menerima arahan langsung dari *Module Lead* serta berkolaborasi secara aktif dengan *Key User* dari setiap area bisnis untuk memastikan bahwa seluruh proses pengujian berjalan sesuai dengan prosedur dan standar kualitas perusahaan. Selain itu, pemahaman terkait proses bisnis diperoleh melalui diskusi rutin dengan *Functional Consultant* dan *Key User* selama proses *design validation* dan UAT.

Peran ini memberikan pengalaman profesional dalam pengujian sistem berskala *enterprise*, sekaligus memperkuat pemahaman mengenai pentingnya koordinasi lintas fungsi, pengelolaan *defect* yang sistematis, serta komunikasi yang efektif guna mendukung keberhasilan implementasi SAP S/4HANA. Struktur organisasi dan mekanisme koordinasi antartim terbukti menjadi elemen penting dalam memastikan kualitas sistem serta keselarasan proses bisnis selama transformasi digital perusahaan.

### 3.2 Tugas yang Dilakukan

Selama keterlibatan dalam proyek T4V (*Transform for Value*) di PT Freeport Indonesia, posisi berperan sebagai Quality Assurance (QA) dengan lingkup fungsi Business Analyst. Posisi ini berada pada tim *testing* di bawah Divisi MIS, dengan fokus utama memastikan kualitas sistem SAP S/4HANA melalui proses pengujian terstruktur pada berbagai skenario bisnis.

Peran yang dijalankan melibatkan keterlibatan langsung dalam *User Acceptance Test* (UAT) dan *Integration Test* terhadap modul-modul utama, khususnya *Materials Management* (MM), *Financial Accounting/Controlling* (FI/CO), dan *Plant Maintenance* (PM). Sebelum pelaksanaan pengujian, QA mempelajari dokumen *Business Process Design* (BPD) dan *Functional Specification* untuk memahami alur bisnis serta konfigurasi sistem yang akan divalidasi. Tahap ini menjadi langkah penting dalam memastikan pemahaman menyeluruh terhadap integrasi proses bisnis di dalam SAP S/4HANA.

Fokus utama UAT dan *Integration Test* adalah memvalidasi integrasi antarmodul secara menyeluruh dan memastikan bahwa skenario bisnis berjalan sesuai desain, tanpa menimbulkan *error*, konflik antarmodul, atau ketidaksesuaian terhadap proses operasional. Pengujian ini berperan penting dalam menjamin bahwa perubahan pada suatu konfigurasi atau *enhancement* tidak berdampak negatif terhadap stabilitas proses bisnis lainnya.

Adapun tugas dan kegiatan utama yang dijalankan selama pengujian

meliputi:

- Melakukan pengujian fungsional untuk memastikan setiap proses berjalan sesuai kebutuhan pengguna.
- Menyusun dan mengeksekusi *test scenario* serta *test script* pada lingkungan pengujian.
- Melakukan *System Integration Testing (SIT)* guna memastikan alur bisnis lintas modul berjalan tanpa kendala.
- Mendokumentasikan temuan (*defect*) dan melakukan *tracking* hingga perbaikan ditangani oleh konsultan fungsional.
- Melakukan analisis awal terhadap *defect* untuk membantu mengidentifikasi penyebab serta dampaknya terhadap proses bisnis.
- Berkoordinasi dengan *module lead*, konsultan fungsional, serta tim terkait lainnya dalam penyelesaian isu selama proses pengujian.
- Memahami alur proses bisnis pada area yang diuji sebagai dasar dalam menentukan cakupan serta prioritas pengujian.

Pelaksanaan kegiatan pengujian didukung oleh tahap persiapan yang mencakup pemahaman terhadap alur proses bisnis pada setiap modul serta keterlibatan dalam diskusi internal tim. Diskusi tersebut meliputi pembagian tanggung jawab pengujian dan evaluasi terhadap temuan yang dihasilkan, yang bertujuan untuk menyamakan persepsi mengenai skenario bisnis serta memastikan bahwa cakupan pengujian telah sesuai dengan kebutuhan pengguna.

Selain itu, selama masa magang, penulis turut berpartisipasi dalam penyusunan *test scenario* dan memperoleh pemahaman komprehensif terkait rangkaian aktivitas pengujian sistem berskala *enterprise*. Aktivitas tersebut mencakup proses identifikasi *defect*, validasi integrasi antar modul dalam pelaksanaan *User Acceptance Test (UAT)* maupun *Integration Test*, serta pentingnya dokumentasi yang sistematis dan koordinasi lintas fungsi dalam menjaga kualitas sistem *SAP S/4HANA*.

Di samping pelaksanaan pengujian dan pelaporan *defect*, penulis juga memperoleh pengalaman dalam melakukan verifikasi hasil *posting* transaksi antar modul menggunakan *SAP GUI* dan *Fiori apps*. Kegiatan ini bertujuan untuk memastikan konsistensi data dan keberhasilan integrasi sistem, sekaligus

mendukung analisis awal terhadap permasalahan yang muncul serta membantu proses penelusuran *root cause* bersama konsultan fungsional.

Di luar tugas teknis, kegiatan magang juga mencakup partisipasi dalam berbagai sesi internal, baik yang berorientasi pada fungsi bisnis maupun teknis sistem SAP. Beberapa materi yang diperoleh antara lain pengenalan proses bisnis pada modul *Materials Management (MM)*, *Financial Accounting Controlling (FI/CO)*, serta *Plant Maintenance (PM)*. Materi tersebut membantu memperkuat pemahaman terhadap alur standar SAP serta hubungan antar modul dalam mendukung operasi bisnis.

Selain itu, terdapat pula kegiatan yang mendukung pengembangan *soft skill*, seperti komunikasi dan manajemen waktu, khususnya dalam konteks penanganan *defect* dan interaksi lintas tim. Seluruh rangkaian kegiatan ini memberikan kontribusi signifikan dalam memperluas wawasan serta meningkatkan kompetensi teknis maupun non-teknis yang dibutuhkan untuk menjalankan peran QA/Business Analyst secara optimal dalam implementasi SAP S/4HANA di lingkungan PTFI.

### 3.3 Uraian Pelaksanaan Magang

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Mengikuti proses <i>onboarding</i> , pembuatan akun perusahaan, instalasi <i>tools</i> kerja, serta pengenalan lingkungan kerja; memahami struktur organisasi, ruang lingkup proyek T4V, dan peran sebagai QA/BA.
2	Mempelajari dokumentasi proses bisnis dan arsitektur sistem SAP S/4HANA; memahami peran modul <i>MM</i> , <i>FI/CO</i> , dan <i>SD</i> ; menyusun rangkuman awal hasil pembelajaran.
3	Mengikuti <i>briefing</i> dan <i>weekly team meeting</i> ; menyusun presentasi pemahaman proses bisnis; berdiskusi dengan <i>functional consultant</i> dan <i>key user</i> untuk klarifikasi alur bisnis.
<i>Lanjut di halaman berikutnya</i>	

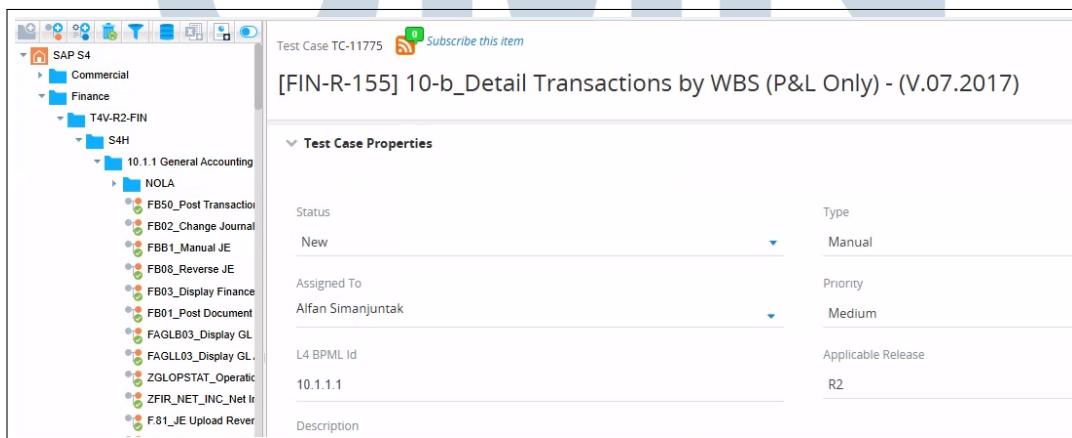
Minggu Ke -	Pekerjaan yang dilakukan
4	Mempelajari penggunaan <i>qTest</i> sebagai <i>test management tool</i> ; memahami alur pencatatan <i>test case</i> , <i>test execution</i> , dan <i>defect tracking</i> .
5–7	Membantu persiapan <i>test environment</i> dan <i>test data</i> ; melakukan konfigurasi akses pengguna pada <i>qTest</i> ; memastikan kesiapan pengujian untuk tahap <i>Pre-SIT</i> .
8	Mendukung pelaksanaan <i>System Integration Test</i> tahap pertama (SIT1); mengeksekusi skenario pengujian; mencatat hasil uji dan <i>defect</i> pada <i>qTest</i> .
9	Melakukan validasi hasil pengujian SIT1; melakukan <i>cross-check</i> integrasi proses bisnis antar-modul; berkoordinasi dengan tim terkait untuk tindak lanjut perbaikan.
10	Melaksanakan <i>retesting</i> terhadap <i>defect</i> yang telah diperbaiki; memperbarui status <i>defect</i> ; menyusun rangkuman progres pengujian untuk <i>daily defect meeting</i> .
11	Melakukan pengujian fungsional modul SAP <i>FI/CO</i> ; mengeksekusi <i>test case</i> sesuai skenario bisnis; memvalidasi kesesuaian sistem dengan <i>business requirement</i> .
12	Melanjutkan pengujian fungsional modul <i>FI/CO</i> ; melakukan verifikasi hasil transaksi; mendokumentasikan hasil pengujian dan temuan pada <i>qTest</i> .
13	Mendukung persiapan <i>System Integration Test</i> tahap kedua (SIT2); melakukan <i>setup test environment</i> dan memastikan kesiapan skenario serta data pengujian.
14	Mendukung pelaksanaan SIT2; mengeksekusi skenario pengujian lanjutan; mencatat hasil uji dan temuan <i>defect</i> .
15	Melakukan proses <i>defect triage</i> ; mengklasifikasikan prioritas dan dampak <i>defect</i> ; berkoordinasi dengan <i>QA Lead</i> , <i>functional consultant</i> , dan <i>developer</i> .
16	Melanjutkan proses <i>defect handling</i> ; memantau status perbaikan; melakukan pengujian ulang dan validasi hasil perbaikan sistem.
<i>Lanjut di halaman berikutnya</i>	

Minggu Ke -	Pekerjaan yang dilakukan
17	Menyusun rangkuman akhir kegiatan magang; merekap hasil pengujian dan pembelajaran yang diperoleh; melakukan evaluasi proses kerja serta kontribusi selama pelaksanaan magang.

### 3.3.1 Pre-System Integration Test 1

Tahap *Pre-System Integration Test 1 (Pre-SIT 1)* merupakan fase persiapan awal sebelum pelaksanaan kegiatan pengujian SIT1 berlangsung. Pada tahap ini, berbagai aspek pendukung pengujian dipersiapkan secara menyeluruh agar pelaksanaan SIT dapat berjalan efektif dan sesuai dengan rancangan. Persiapan dilakukan mencakup aspek logistik dan non-logistik, seperti penyiapan lingkungan sistem (*test environment*), validasi kesiapan data uji (*test data readiness*), serta penyesuaian akses pengguna bagi tim QA, *functional consultant*, dan *key user*.

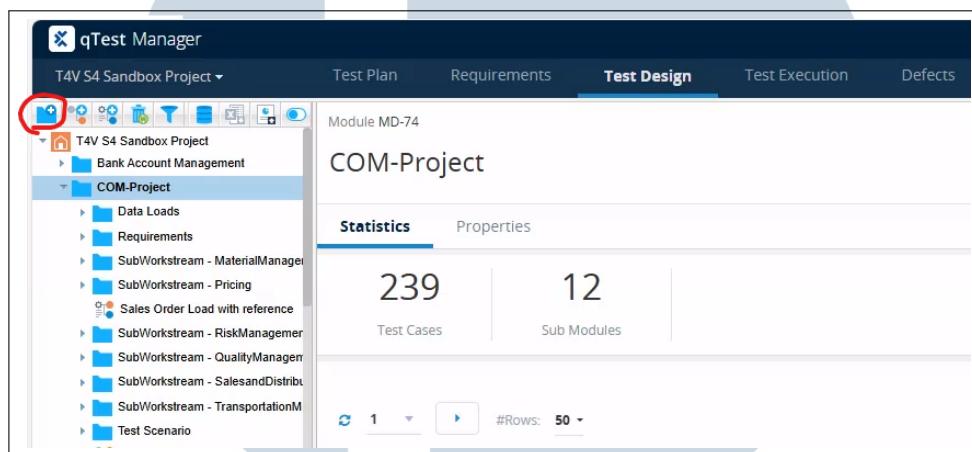
Struktur *test environment* pada SAP S/4HANA disusun berdasarkan hierarki domain fungsional sehingga setiap skenario pengujian dapat diorganisasi secara sistematis. Sebagaimana ditunjukkan pada Gambar 3.1, lingkungan pengujian untuk modul Finance ditempatkan pada direktori T4V-R2-FIN → S4H → 10.1.1 General Accounting → NOLA, yang memuat daftar transaksi (proses bisnis) yang relevan untuk kegiatan pengujian. Contoh transaksi tersebut antara lain *FB50 (Post Transaction)*, *FB02 (Change Journal)*, dan *FBB1 (Manual Journal Entry)*. Struktur ini memastikan bahwa cakupan pengujian tersusun secara terarah dan selaras dengan proses bisnis aktual.



Gambar 3.1. Struktur *Test Environment* SAP S/4HANA pada Modul Finance

Namun, sebelum mencapai struktur seperti ini, *test environment* pada awalnya dalam kondisi kosong. Oleh karena itu, diperlukan proses setup environment terlebih dahulu. Adapun langkah-langkah setup adalah sebagai berikut:

Pertama ketika masuk ke qTest Manager, pada Tab Test Design klik tombol new Design yang di-highlight pada gambar 3.2

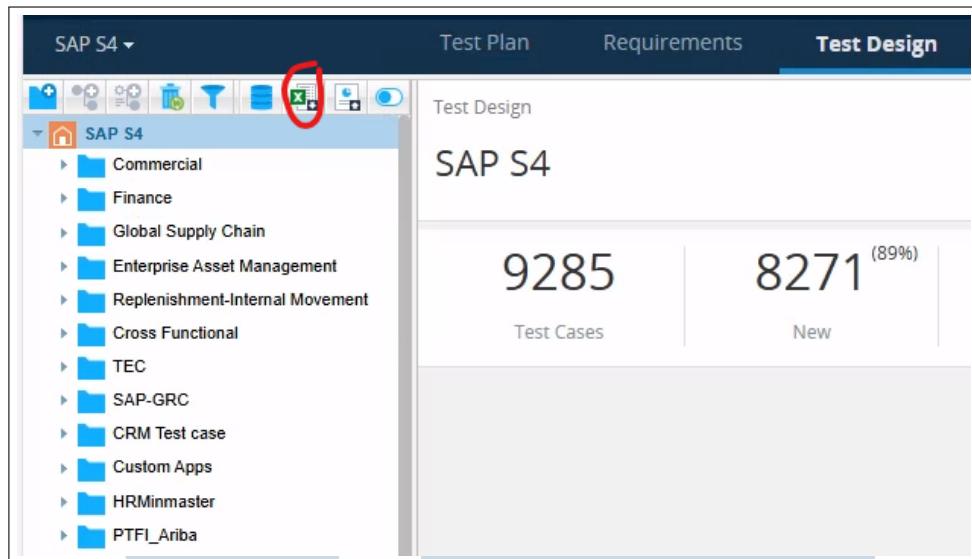


Gambar 3.2. Tombol New Test Design

Pada tab test design, kita dapat mengunggah data uji, test cycle, test cases, hingga test run. Proses penggeraan tahap ini memakan waktu kurang lebih satu setengah bulan sejak awal masa magang, karena dilakukan secara paralel dengan tim fungsional untuk memperoleh validasi serta kesesuaian dengan *Business Requirement Document* (BRD).

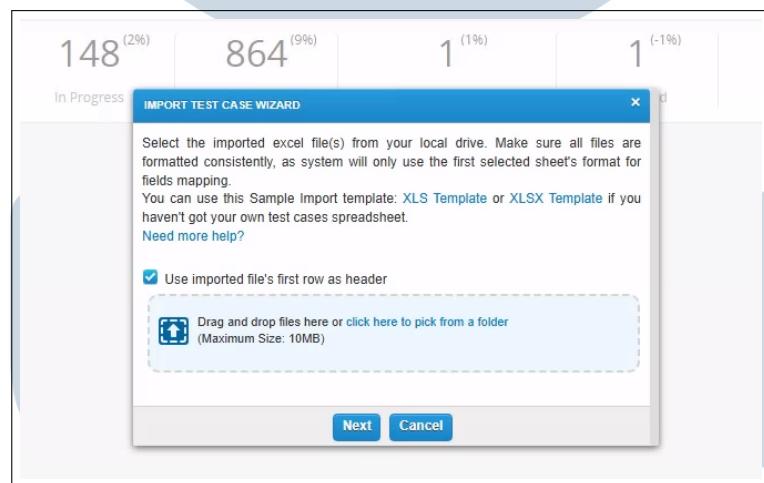
Sebagai gambaran berikut merupakan cara untuk mengupload Test Scenario baru ke testing environment melalui tab test design. Pertama klik tombol excel yang ada pada tampilan seperti yang terlihat pada gambar 3.3

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.3. Tombol untuk import Test Cases

Setelah itu akan muncul wizard seperti ini untuk menjadi uploader test cases anda yang ada dalam file excel

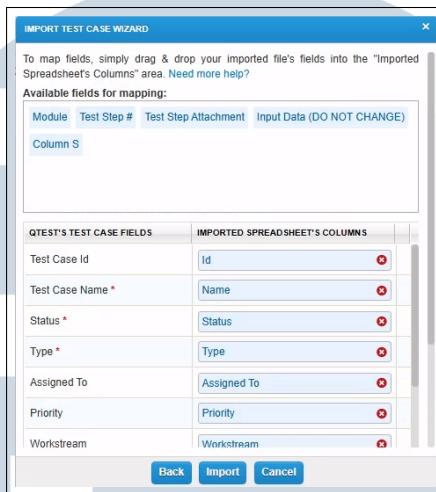


Gambar 3.4. Tombol untuk import Test Cases

Berikut adalah contoh saat penulis mengunggah *Test Scenario* untuk stream Financial menggunakan wizard tersebut. Pada proses ini, setiap kolom pada *test case* perlu dicocokkan dengan menu atau field yang tersedia di dalam wizard. Umumnya, sebuah *test case* memuat beberapa informasi seperti ID *test case*, nama *test case*, status eksekusi, nama tester yang ditugaskan, dan data lainnya sebagaimana ditampilkan pada gambar di bawah.

Setelah seluruh kolom berhasil dipetakan dengan benar, *test case* akan

masuk ke dalam menu *Test Design* dan secara otomatis tersusun dalam bentuk folder, seperti yang ditunjukkan pada Gambar 3.3.



Gambar 3.5. Menu Upload

Selanjutnya, dilakukan verifikasi serta finalisasi *test scenario* dan *test case* berdasarkan *business requirement* yang telah disepakati. Setelah proses finalisasi, seluruh berkas *test case* (TC) dan *test run* (TR) dimuat ke dalam platform manajemen pengujian *qTest* sebagai *project management tool* guna memfasilitasi pelacakan progres pengujian dan pencatatan *defect* secara terstruktur. Akses ke *qTest* juga dipersiapkan agar seluruh *tester* yang terlibat dalam SIT1 dapat melakukan penugasan dan eksekusi tanpa hambatan [13].

Untuk memberikan gambaran lebih jelas mengenai asset pengujian yang digunakan, Gambar 3.7 menampilkan contoh test case yang memuat detail langkah pengujian, prasyarat, data uji, serta hasil yang diharapkan. Penyusunan test case ini memastikan bahwa setiap fungsi bisnis diuji secara sistematis dan konsisten sesuai kebutuhan pengguna.

Module	Name	Id	Status	Type	Description	Precondition	Test Step	Test Step Description	Test Step Expected Result	Requirement Ids	Priority	Workstream	SPML ID (L4)
10.2.3 Corporate Tax (FIN-R-166) Double Tax Code on Invoice TC-11849	New		Manual	Double Tax Code	1) User should:			To launch EDW. Go to this URL: <a href="https://edwqa001/BOE/BI/">https://edwqa001/BOE/BI/</a>	Navigate to EDW Report				
							2	SAP BI Launch Pad - <a href="https://edwqa001/BOE/BI/">https://edwqa001/BOE/BI/</a>					
								Execute EDW: Double Tax Code on Invoice Document					
								Click on Query and Select the required parameters for analysis.	Report output is displayed	RQ-2958	Medium	Finance	10.2.3.1
							3	Parameters: SAP BI Launch Pad - <a href="https://edwqa001/BOE/BI/">https://edwqa001/BOE/BI/</a>					
								Click on 'Export' to download the report.					
								Select 'Excel' and 'All Reports'.					
								Compare report between EDW S4 and EDW ECC on <a href="https://edwqa001/BOE/BI/">https://edwqa001/BOE/BI/</a>	Report is validated. Results match against ECC data				

Gambar 3.6. Contoh Test Case

Setelah finalisasi test case, dilakukan pula pembuatan dan pengelolaan test run (TR) pada *qTest* sebagai struktur pelaksanaan pengujian selama SIT1. Test run

berfungsi sebagai turunan operasional dari test case, yang memuat rincian langkah pelaksanaan, status eksekusi, serta dokumentasi hasil pengujian. Melalui TR, setiap penguji dapat mencatat hasil aktual pengujian secara real time, termasuk status *pass* atau *fail*, serta bukti pendukung berupa tangkapan layar atau catatan hasil uji.

Sebagaimana ditunjukkan pada Gambar 3.7, *Test Run (TR)* merepresentasikan rincian setiap aktivitas eksekusi pengujian (*execution instance*) yang diturunkan dari langkah-langkah terperinci pada *test case*. Melalui struktur ini, setiap skenario pengujian dicatat sebagai unit pelaksanaan yang terpisah, sehingga memungkinkan pengelolaan eksekusi pengujian, pemantauan progres, serta penelusuran *defect* dilakukan secara lebih sistematis dan terukur. Pendekatan ini memastikan bahwa hasil pengujian terdokumentasi dengan baik serta tetap selaras dengan *business requirement* yang berlaku.

ID	NAME	TEST CASE ID	TEST CASE	DEFECTS	TEST CASE VERSION	STATUS	TARGET RELEASE/BUILD	TEST CYCLE
TR-66177	WebGUI_ZFII_JOURNAL_UPLOAD_Journal Entry Upload Program_Internal Order_Cost Center	TC-4750	IC-4-202 WebGUI_ZFII_JOURNAL_UP Entry Upload Program_Internal Order_Cost Center	0	7.0	PASSED	T4VS4_R2_FIN	CL-3047 1460 PTFI
TR-66178	WebGUI_ZFII_JOURNAL_UPLOAD_Journal Entry Upload Program_Cost Center	TC-2640	TC-2640 WebGUI_ZFII_JOURNAL_UP Entry Upload Program_Cost Center	0	4.0	PASSED	T4VS4_R2_FIN	CL-3047 1460 PTFI
TR-66179	WebGUI_FB50_Post Transaction - Validation for Exempt Account	TC-93	TC-93 WebGUI_FB50_Post Transaction -	0	9.0	PASSED	T4VS4_R2_FIN	CL-3047 1460 PTFI
TR-66180	SOX - S4 FIN Report — Total vs Prior Analysis Report (ZFIR_TOTALVSPRIOR)	TC-10703	TC-10703 SOX - S4 FIN Report — Total vs Prior Analysis Report (ZFIR_TOTALVSPRIOR)	0	3.0	PASSED	T4VS4_R2_FIN	CL-3047 1460 PTFI

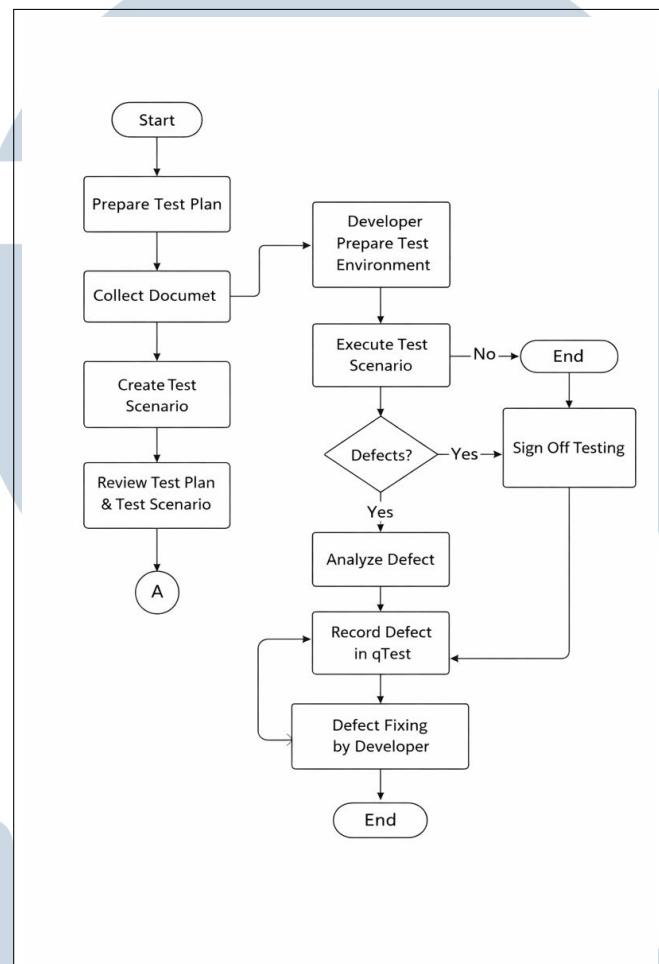
Gambar 3.7. Contoh Test Run

Setiap *Test Run* memuat informasi operasional seperti status eksekusi (*Pass*, *Fail*, *Blocked*), catatan hasil pelaksanaan, serta lampiran bukti pengujian berupa tangkapan layar atau dokumen pendukung. Apabila suatu *Test Run* berstatus *Fail*, maka dilakukan pencatatan *defect* melalui integrasi antara *qTest* dan *SAP Solution Manager*. *Defect* yang tercatat akan secara otomatis terhubung dengan *Test Run* terkait, sehingga memudahkan proses penelusuran akar masalah, penugasan tindak lanjut kepada pihak fungsional maupun teknis, serta pemantauan proses perbaikan.

Untuk memberikan gambaran mengenai alur pelaksanaan pengujian secara *end-to-end*, Gambar 3.8 menampilkan proses *test execution flow* mulai dari perencanaan hingga penyelesaian aktivitas pengujian. Proses dimulai dari penyusunan *test plan* dan pengumpulan dokumen pendukung sebagai dasar perumusan *test scenario*. Setelah melalui proses peninjauan, *developer* menyiapkan *test environment* yang diperlukan sebelum skenario dapat dieksekusi.

Selama eksekusi, apabila ditemukan *defect*, maka dilakukan analisis awal untuk mengidentifikasi akar permasalahan, diikuti pencatatan *defect* pada *qTest*. *Defect* kemudian ditindaklanjuti oleh *developer* hingga proses perbaikan selesai. Jika tidak ditemukan *defect*, atau setelah seluruh *defect* terselesaikan, pengujian

dapat dilanjutkan menuju proses *sign-off*. Alur terstruktur ini memastikan bahwa setiap temuan terdokumentasi dan ditindaklanjuti dengan baik, sehingga kualitas sistem dapat terjaga secara konsisten.



Gambar 3.8. Tahapan dan Flow Testing hingga Defect Handling

### 3.3.2 System Integration Test 1

Pada tahap *System Integration Testing* (SIT) 1, penulis berperan dalam melaksanakan aktivitas pengujian berdasarkan *test scenario* dan *test case* yang telah difinalisasi pada fase *Pre-SIT* 1.

Secara keseluruhan, penulis mengeksekusi 11 *test case* yang mencakup rangkaian proses bisnis pada modul *Finance* (FI/CO) dan *Sales Distribution* (SD) yang termasuk dalam *stream Financial*. Aktivitas pengujian tersebut berfokus pada proses transaksi pencatatan jurnal serta pengelolaan dokumen keuangan.

Meskipun jumlah *test case* utama yang dijalankan relatif terbatas, yaitu

sebanyak 11 skenario, total *test run* yang dihasilkan menjadi cukup banyak. Hal ini disebabkan oleh karakteristik sistem SAP yang bersifat terintegrasi antar modul, sehingga satu *test case* dapat melibatkan beberapa modul dan tahapan pengujian. Setiap skenario pengujian tidak hanya diverifikasi pada modul asalnya, tetapi juga memerlukan validasi lanjutan pada modul lain untuk memastikan bahwa alur data, proses bisnis, serta pencatatan keuangan berjalan dengan benar secara *end-to-end*.

Sebagai tambahan informasi, pada proyek migrasi SAP S/4HANA di PT Freeport Indonesia, terdapat lima modul utama yang menjadi cakupan pengujian, yaitu *Financial* (FIN), *Commercial* (COM), *Global Supply Chain* (GSC), *Enterprise Asset Management* (EAM), dan *Human Resource Program* (HRP).

Dalam pelaksanaannya, penulis mengeksekusi total 11 *test case* utama yang mencakup beberapa modul tersebut. Namun demikian, jumlah *test run* yang dihasilkan melebihi jumlah *test case* yang ada. Hal ini disebabkan oleh adanya pengujian lintas modul serta pelaksanaan *re-run* terhadap *test case* yang sebelumnya berstatus *Fail* maupun *Blocked*.

Setiap *test run* didokumentasikan melalui platform *qTest*, yang mencatat rincian status eksekusi (*Pass*, *Fail*, dan *Blocked*), catatan pelaksanaan, serta lampiran bukti pengujian. Hasil eksekusi pengujian tersebut berkontribusi pada identifikasi beberapa *defect* lintas modul yang kemudian ditindaklanjuti melalui mekanisme *defect management* terintegrasi pada *qTest* dan SAP Solution Manager.



Tabel 3.2. Daftar Test Case yang Dieksekusi pada Tahap SIT 1

No.	Test Case ID	Nama Test Case	Modul
1	TC-1091	WebGUI_ZFIR_FAADD Additions Reports	FI
2	TC-2277	Fiori Manage Credit Accounts – Credit Profile Update	FI
3	TC-44	WebGUI_VA01_Create Sales Order	SD
4	TC-58	WebGUI_VL02N_Update Outbound Delivery – Pack Delivery	SD
5	TC-64	WebGUI_VL02N_Update Outbound Delivery – Pick Delivery	SD
6	TC-68	WebGUI_VL02N_Update Outbound Delivery – Post Goods Issue	SD
7	TC-862	WebGUI_VL02N_Update Outbound Delivery – Release Delivery	SD
8	TC-9668	WEBGUI_ABBA_Unplanned Depreciation	FI
9	TC-2279	ULM_BP New Credit Analyst	FI
10	TC-2283	WebGUI_SCASE_New Credit Analyst Updated	FI
11	TC-93	WebGUI_FB50_Post Transaction	FI

Secara keseluruhan, daftar *test case* yang dieksekusi pada tahap SIT 1 mencakup proses bisnis dari modul FI dan SD. Pada modul Financial (FI), pengujian meliputi eksekusi laporan penambahan aset melalui ZFIR\_FAADD (TC-1091), pembaruan profil kredit pada aplikasi Fiori *Manage Credit Accounts* (TC-2277), pembuatan analis kredit baru melalui ULM\_BP (TC-2279), pemutakhiran data analis kredit pada SCASE (TC-2283), serta proses pencatatan jurnal menggunakan FB50 (TC-93).

Sementara itu, untuk modul Sales and Distribution (SD), pengujian mencakup pembuatan *Sales Order* melalui VA01 (TC-44) serta serangkaian aktivitas terkait *Outbound Delivery* menggunakan VL02N, mulai dari *packing* (TC-58), *picking* (TC-64), *posting goods issue* (TC-68), hingga *release delivery* (TC-862). Selain itu, proses pembuatan *Invoice* melalui VF01 (TC-36) juga turut diuji dalam rangka memastikan integrasi end-to-end pada proses distribusi dan penagihan.

## A Contoh Proses Testing pada Fase SIT 1 yang passed

Berikut merupakan contoh detail proses pengujian pada tahap SIT 1 menggunakan salah satu *Test Run* (TR) dari *Test Case* TC-1091, yaitu TR-1756. TR-1756 digunakan untuk memverifikasi akurasi data pada saat eksekusi laporan penambahan aset melalui *ZFIR FAADD*, yaitu kode transaksi (*custom program*) dalam SAP yang berfungsi untuk memproses penambahan aset.

Dalam TR-1756 terdapat tiga tahapan utama yang harus dijalankan untuk menguji fungsionalitas tersebut, sebagaimana ditunjukkan pada tabel berikut.

Tabel 3.3. Detail Eksekusi Test Run – AW01N

Step	Deskripsi	Expected Result
1	Buka transaction code AW01N.	Halaman “Asset Explorer” tampil.
2	Masukkan nilai pada field berikut, kemudian klik <i>Translate currency amounts</i> :	Input diterima dan proses translasi mata uang berhasil dilakukan.
	<ul style="list-style-type: none"><li>• Company Code: &lt;Company Code&gt;</li><li>• Asset: &lt;Asset&gt;</li></ul>	
	Setelah itu klik <i>Continue</i> .	
3	Lakukan verifikasi terhadap nilai Asset Depreciation dan Transaction Type.	Nilai Asset Depreciation dan Transaction Type berhasil diverifikasi.

Seperti yang ditunjukkan pada Tabel 3.3, Test Run ini terdiri dari tiga langkah utama yang diawali dengan memasukkan transaction code AW01N untuk mengakses menu Asset Explorer.



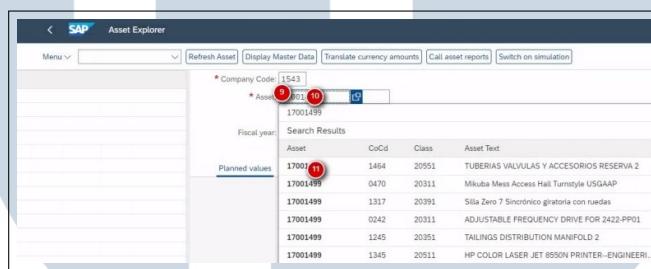
Gambar 3.9. Memasukkan kode transaksi

Setelah berhasil masuk ke menu Asset Explorer, pengguna diminta untuk memasukkan kode perusahaan. Pada skenario pengujian ini, digunakan Company Code 1543.



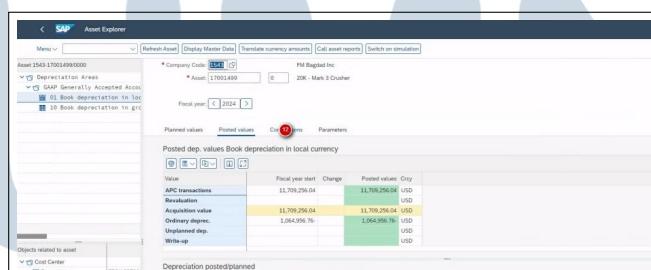
Gambar 3.10. Memasukkan kode perusahaan

Selanjutnya, pada kolom Asset, pengguna memasukkan kode aset. Pada Test Run ini, digunakan Asset Number 17001499, kemudian menekan tombol Enter.



Gambar 3.11. Memasukkan kode asset

Setelah aset berhasil dipilih, sistem menampilkan rincian aset sebagaimana diperlihatkan pada gambar berikut. Kemudian pengguna diarahkan menuju tab Comparison untuk melakukan komparasi nilai.



Gambar 3.12. Tampilan asset

Pada tab Comparison, terlihat adanya perbedaan antara Acquisition Value dan Net Book Value, yang mengindikasikan bahwa aset telah mengalami depresiasi. Informasi tersebut menunjukkan bahwa tab Comparison memberikan laporan yang sesuai dan konsisten dengan status aset. Selanjutnya dilakukan verifikasi terhadap parameter depresiasi untuk memastikan validitasnya.

Fiscal year	APC transactions	Acquisition value	Ordinary deprec.	Net book value	Crcy
<2024	11,709,256.04	11,709,256.04	1,064,956.76-	10,644,299.28	USD
2024		11,709,256.04	390,616.49-	10,253,682.79	USD
2025		11,709,256.04	390,616.49-	9,863,066.30	USD
2026		11,709,256.04	390,616.49-	9,472,449.81	USD
2027		11,709,256.04	390,616.49-	9,081,833.32	USD
2028		11,709,256.04	390,616.49-	8,691,216.83	USD
2029		11,709,256.04	390,616.49-	8,300,600.34	USD
2030		11,709,256.04	390,616.49-	7,909,983.85	USD
2031		11,709,256.04	390,616.49-	7,519,367.36	USD
2032		11,709,256.04	390,616.49-	7,128,750.87	USD
2033		11,709,256.04	390,616.49-	6,738,134.38	USD
2034		11,709,256.04	390,616.49-	6,347,517.89	USD
2035		11,709,256.04	390,616.49-	5,956,901.40	USD
2036		11,709,256.04	390,616.49-	5,566,284.91	USD

Gambar 3.13. Tampilan tab Comparison

Setelah berpindah ke tab Parameters, dapat diamati bahwa parameter depresiasi telah tercatat dengan benar dan sesuai, sebagaimana ditampilkan pada gambar berikut.

Interval From 01/01/1900 To 12/31/9999	
Dep. Key:	ZSTL FCX-Str.-line frm acq.value pro rata w/curb+intrst
Group Asset:	
Life	Start of calculation
Useful Life: 030 / 000	Dep. Start Date: 04/01/2021
Expired UL: 002 / 009	Spec.deprecations: <input type="text"/>
Remaining life: 027 / 003	Int.Calc. Start: <input type="text"/>

Gambar 3.14. Depreciation key

Berdasarkan hasil pengujian, seluruh fungsi yang diuji telah berjalan sebagaimana mestinya, sehingga Test Run ini dinyatakan PASS atau berhasil.

## B Contoh Proses Testing pada Fase SIT 1 yang gagal

Berikut merupakan contoh proses pengujian pada modul test case yang menghasilkan defect atau kegagalan. Pada pengujian ini, dilakukan eksekusi terhadap TC-9668 – WEBGUI ABAA Unplanned Depreciation, yaitu test case yang digunakan untuk memvalidasi proses Unplanned Depreciation Posting melalui SAP WebGUI menggunakan transaction code ABAA.

Tujuan dari test case ini adalah memastikan bahwa user dapat melakukan pencatatan penyusutan tidak terencana (unplanned depreciation) pada suatu aset, mulai dari pengisian parameter, verifikasi nilai penyusutan, pelaksanaan simulasi posting, hingga penyelesaian proses posting akhir. Untuk detail test cases dapat dilihat pada tabel di bawah ini

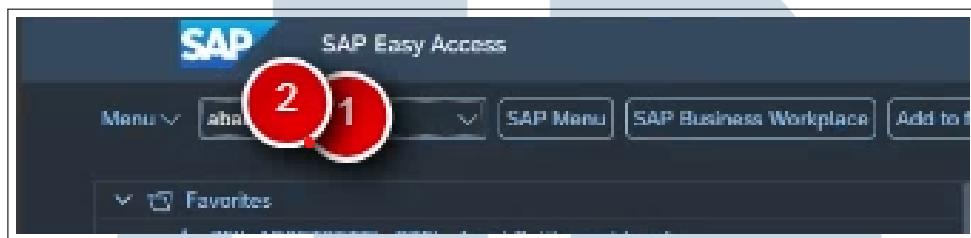
Tabel 3.4. Detail Eksekusi Test Run – ABAA

Step	Deskripsi	Expected Result
1	Masukkan transaction code ABAA.	Tidak memiliki otorisasi untuk ABAA, sehingga perlu dilakukan pemberian akses.
2	Masukkan nilai pada field berikut, kemudian klik <i>Continue</i> : <ul style="list-style-type: none"><li>• Company Code: &lt;company code&gt;</li><li>• Asset: &lt;asset number&gt;</li><li>• Sub-number: &lt;sub-number&gt;</li><li>• Transaction Type: &lt;transaction type&gt;</li></ul>	Parameter berhasil dimasukkan dan halaman utama <i>Unplanned Depreciation</i> muncul.
3	Masukkan nilai pada field berikut: <ul style="list-style-type: none"><li>• Document Date: &lt;document date&gt;</li><li>• Posting Date: &lt;posting date&gt;</li><li>• Asset Value Date: &lt;asset value date&gt;</li><li>• Amount Posted: &lt;amount posted&gt;</li><li>• Text: &lt;text&gt;</li></ul>	Seluruh nilai berhasil diisi dengan benar.
4	Klik <i>Line Items</i> dan verifikasi nilai <i>Amount Posted</i> pada masing-masing Depreciation Area.	Nilai pada setiap Depreciation Area berhasil direview.
5	Klik tombol <i>Simulate</i> dan lakukan review hasil simulasi.	Preview Unplanned Depreciation tampil tanpa error.
6	Klik tombol <i>Post</i> untuk melakukan Unplanned Depreciation Posting.	Transaksi Unplanned Depreciation berhasil diposting.

Seperti yang dapat dilihat pada Tabel di atas, terdapat 6 step utama yang perlu kita lakukan untuk mengeksekusi test cases ini dimulai dari memasukkan

code transaksi ABAA di SAP WebGUI sampai melakukan poisting pada Unplanned Depreciation.

Pada tahap pertama, ketika masuk ke SAP WebGUI masukkan kode transaksi ABAA seperti yang ditunjukkan oleh gambar di bawah



Gambar 3.15. Memeasukkan Kode transaksi ABAA

Setelah kita proceed dari menu ABAA ternyata ada masalah authorisasi yang menyebabkan user tidak bisa mengakses transaksi ABAA seperti yang dilampirkan pada gambar di bawah

INT4 Result of Authorization Check	Additional Information for Check	Access Filtering	CDS Entity	Object	Field 1	Value 1	Field 2	Value 2
12 No authorization in user master record				S_WEBGU	ACTVT	A9		
12 No authorization in user master record				S_WEBGU	ACTVT	AG		
4 Authorization check not successful				S_TCDE	TCD	ABAAL		
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	SAPMS01GENEW
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23		
4 Authorization check not successful				S_TCDE	TCD	ABAAL		
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	SAPMS01GENEW
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23		
4 Authorization check not successful				S_TCDE	TCD	ABAAL		
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23	REPORT	RAGITT_ALV01
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	RAGITT_ALV01
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
4 Authorization check not successful				S_ALV_LAYR	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR
12 No authorization in user master record				S_ALV_LAYO	ACTVT	23	REPORT	ZINT_FL_ASSET_LOAD_PTR

Gambar 3.16. Authorization Error

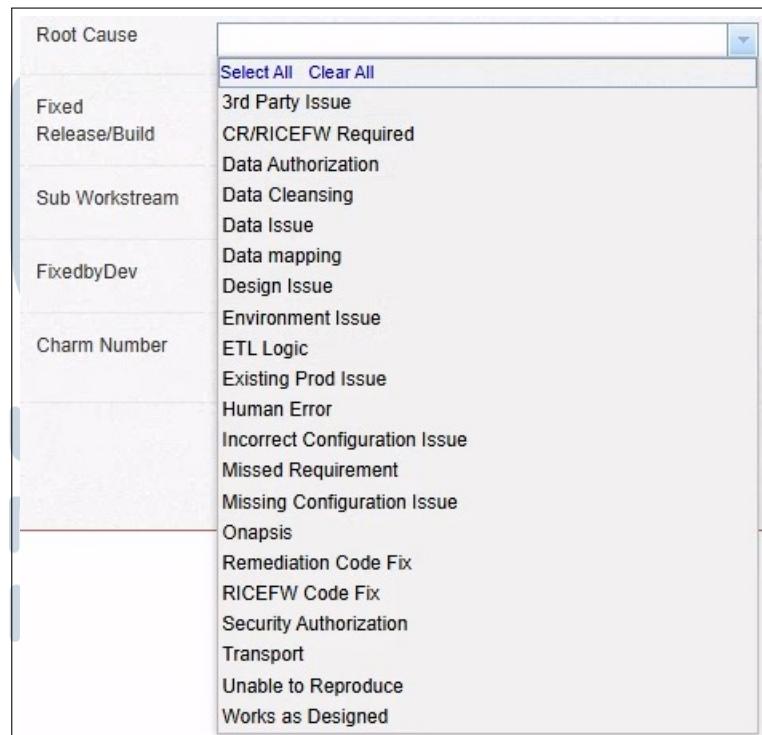
Dikarenakan kita baru berada pada tahap pertama namun error seperti ini sudah terjadi, maka logging defect pun diperlukan pada tahap ini, berikut merupakan tahap untuk logging defect yang dimulai dari masuk ke Page defect logging.

MULTIMEDIA  
NUSANTARA

Gambar 3.17. Page untuk defect logging

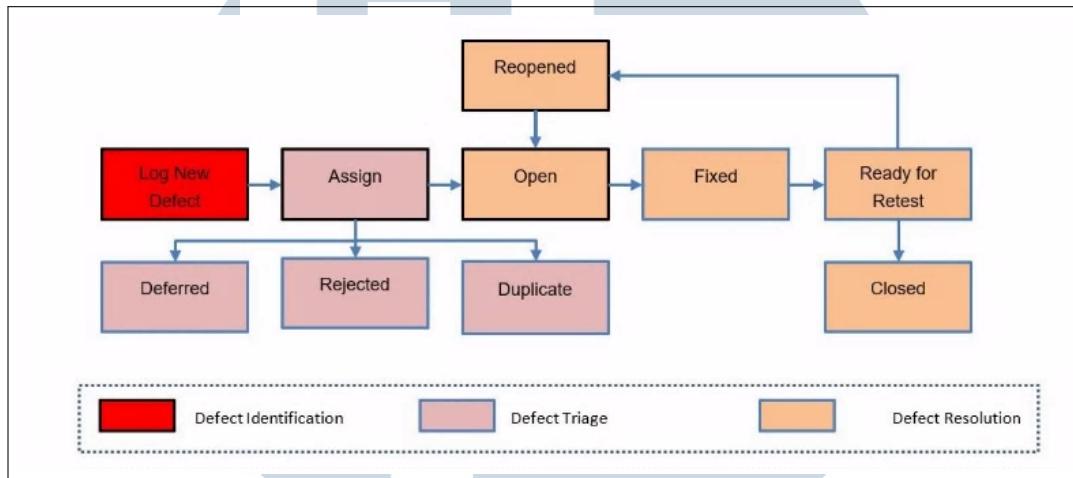
Setelah mengisi data-data dari *defect* tersebut, *defect* akan secara otomatis ter-*attach* pada (TR) tempat *error* tersebut ditemukan. Selanjutnya, *defect* akan diteruskan kepada pihak *Advance Business Application Programmer* (ABAP) untuk dilakukan proses *fixing*.

Dari sisi *tester*, informasi tambahan maupun masukan terkait *root cause* dapat diberikan untuk membantu proses analisis dan perbaikan. Dalam proyek ini, terdapat beberapa kategori *root cause* yang dapat dipilih sebagai bagian dari proses klasifikasi *defect*, seperti pada gambar berikut



Gambar 3.18. Category Root Cause dari Defect

Sebagai contoh, pada kasus *error* yang telah dibahas sebelumnya terjadi *authorization error*. Oleh karena itu, *root cause* dari *defect* tersebut dapat diklasifikasikan sebagai *Security Authorization*. Berdasarkan klasifikasi tersebut, *defect* kemudian diproses lebih lanjut mengikuti tahapan dalam *Defect Management Life Cycle*.



Gambar 3.19. Defect Management Life Cycle

*Defect Management Life Cycle* dimulai ketika *tester* melakukan pencatatan (*logging*) *defect* baru ke dalam sistem *qTest*. Setelah proses pencatatan selesai, *defect* tersebut akan *di-assign* kepada pihak yang bertanggung jawab untuk melakukan perbaikan, baik *ABAP developer* maupun *external functional consultant*, sesuai dengan sumber permasalahan yang teridentifikasi.

Pada tahap ini, *defect* juga melalui proses *triage* yang bertujuan untuk menentukan prioritas, klasifikasi, serta validitas *defect* tersebut. Untuk menentukan tingkat keparahan *defect* dalam proyek ini, digunakan sebuah *severity guideline* yang menjadi acuan dalam mengkategorikan tingkat dampak atau keparahan dari setiap *defect*. Acuan penentuan *severity* ditunjukkan pada

Severity	Description	Business Critical	Process Stop
1 – Urgent	A total system failure that stops any testing. In a production environment, this error would stop production. Error threatens ability to go live by planned date.	Yes	Yes
2 – High	Error stops progress of test script as written. Complicated workarounds are available however, not acceptable for go-live unless reviewed and downgraded by business.	Yes	No
3 – Medium	Impacts functionalities however, acceptable work arounds are available. Product can go live with these defects.	No	No
4 – Low	Cosmetic or minor error. Error does not stop progress of test script as written. Workaround for error is simple and acceptable to the business. It is not necessary to fix this before going live.	No	No

Gambar 3.20. Defect Triage Guidelines (Severity)

Selain severity, terdapat pula acuan untuk menentukan tingkat prioritas

perbaikan (priority) untuk setiap defect berdasarkan tingkat urgensinya. Acuan penentuan priority tersebut ditunjukkan pada Gambar 3.21.

Priority	Description	Business Critical	Process Stop
P1 – Critical	Complete failure of the core functionality, process or system and test execution is completely blocked. Aim to fix as soon as Possible, could impact testing timelines	Yes	Yes
P2 – High	An important piece of functionality fails, and we need to test it, we have a workaround that is limited, however significant extra work is involved. Aim to fix for the next deployment window into test	Yes	No
P3 – Medium	Functionality fails but we do not need to test it right away and we have workaround leaving the impact and extra work marginal. Aim to fix in the next Test run	No	No
P4 – Low	Limited to no impact on test scripts. Fix the defect when all higher priorities have been fixed. Aim to fix and test in the next opportunity. May be a candidate to be delayed	No	No

Gambar 3.21. Defect Triage Guidelines (Priority)

Setelah *severity* dan *priority* ditetapkan, pihak yang bertanggung jawab dalam melakukan perbaikan (*fixer*) akan melaksanakan proses perbaikan berdasarkan tingkat urgensi yang telah ditentukan. Semakin tinggi tingkat *severity* dan *priority*, maka semakin singkat waktu penyelesaian yang ditargetkan. Sebagai contoh, *defect* dengan kategori *SEV 1–P 1* ditargetkan untuk diselesaikan dalam waktu kurang lebih dua hari.

Perbaikan yang telah diselesaikan kemudian dikembalikan kepada *tester* dengan status *Ready for Retest* untuk dilakukan verifikasi melalui pengujian ulang (*retest*). Apabila hasil *retest* menunjukkan bahwa perbaikan telah sesuai dan *defect* tidak lagi ditemukan, maka *defect* akan ditutup dengan status *Closed*. Namun, apabila perbaikan belum efektif atau menimbulkan permasalahan tambahan, *defect* akan dikembalikan dengan status *Reopened*, sehingga proses perbaikan harus diulang hingga hasil yang diperoleh memenuhi standar kualitas yang telah ditetapkan.

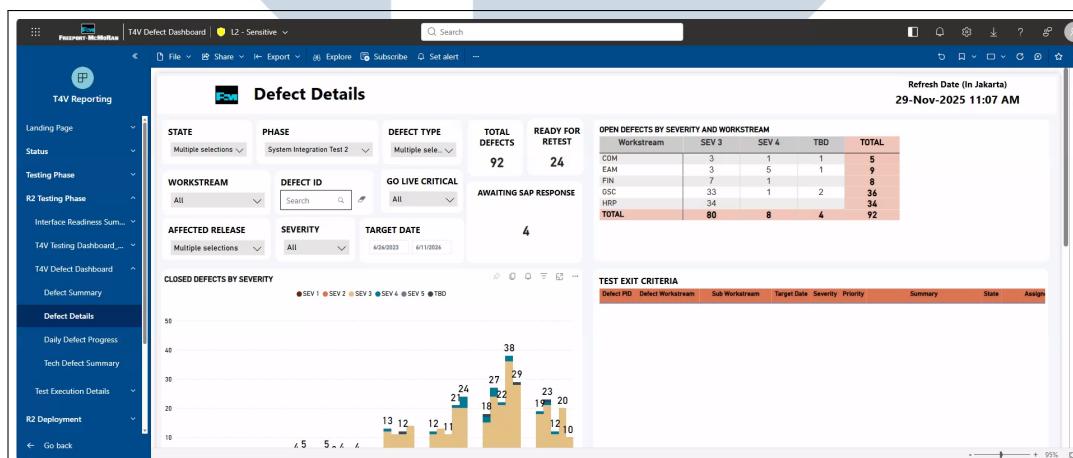
### 3.3.3 SIT 2

Pada proyek ini, jumlah *defect* menunjukkan tren peningkatan, sehingga pada fase *System Integration Testing* (SIT) 2 penulis ditugaskan sebagai *defect handler* yang bertanggung jawab untuk mengelola *defect traffic* serta memonitor keseluruhan *defect life cycle*. Pada fase SIT 2, penulis melakukan *defect tracking* untuk memantau apakah penyelesaian setiap *defect* berjalan sesuai rencana dan memenuhi standar ketepatan waktu yang telah ditetapkan. Aktivitas ini mencakup pemantauan *Service Level Agreement* (SLA) berdasarkan kombinasi *severity–priority*, memastikan kesesuaian proses dengan prosedur *quality governance*, serta mengidentifikasi potensi deviasi terhadap jadwal penyelesaian berdasarkan praktik

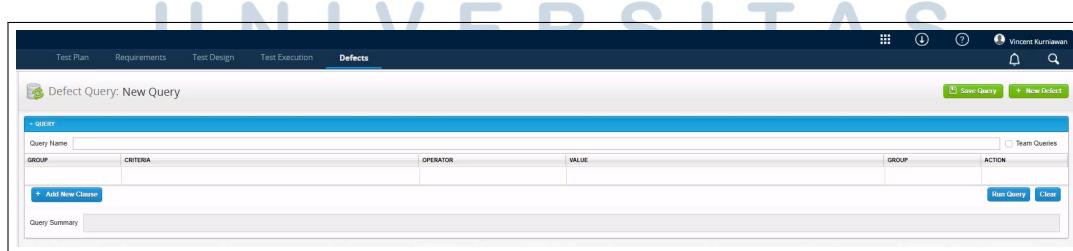
project management yang berlaku.

Untuk mendukung aktivitas *defect tracking* tersebut, penulis menggunakan tiga *tools* utama, yaitu *Power BI*, *Azure DevOps*, dan *ServiceNow*. *Power BI* berfungsi sebagai alat monitoring yang menyajikan visualisasi status *defect*, tren penyelesaian, distribusi *severity-priority*, serta indikator kepatuhan terhadap SLA. *Azure DevOps* digunakan untuk memantau progres penyelesaian *defect* secara detail, termasuk status perbaikan, penugasan *developer*, dan riwayat perubahan (*work item history*). Sementara itu, *ServiceNow* digunakan sebagai sistem *ticketing* untuk mendukung para *fixer* apabila diperlukan pembukaan tiket baru terkait isu tertentu, khususnya jika permasalahan membutuhkan eskalasi atau penanganan lintas tim.

Dalam proses monitoring *defect*, setiap pagi pada pukul 10.00 WIB dilakukan penarikan data *defect* dari *Power BI* dan *qTest* untuk keperluan *tracking*. Berikut merupakan tampilan *Power BI* dan *qTest* pada saat proses penarikan data dilakukan.



Gambar 3.22. Tampilan PowerBI



Gambar 3.23. Tampilan qTest untuk penarikan DF

Pengambilan data di lakukan dari ke-2 platform untuk memastikan

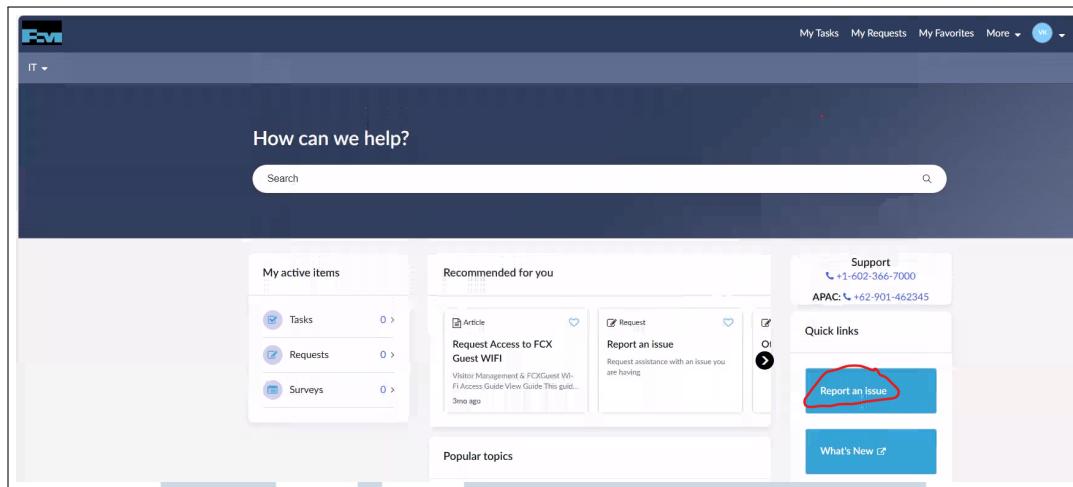
keakuratan data, dikarenakan terkadang PowerBI sering error dan mismatch dalam menampilkan detail dari tiap DF. Untuk pengambilan data via PowerBI kita bisa langsung menggunakan fitur built-in dari PowerBI untuk export data yang ada di dashboard menjadi format data excel sedangkan untuk qTest kita perlu melakukan query untuk mengambil data seperti query SQL. Berikut merupakan pseudocode untuk query di qTest

```
1
2 WHERE
3 (
4     Affected_ReleaseBuild = "T4VS4_R2_COM"
5     OR Affected_ReleaseBuild = "T4VS4_R2_FIN"
6     OR Affected_ReleaseBuild = "T4VS4_R2_GSC"
7     OR Affected_ReleaseBuild = "T4VS4_R2_EAM"
8 )
9
10 AND Test_Phase = "System Integration Test 2"
11
12 AND Status <> "Deferred"
13 AND Status <> "Rejected"
14 AND Status <> "Duplicate"
15 AND Status <> "Closed"
```

Kode 3.1: Query SIT 2 - All Open Defects

Setelah mendapatkan daftar defect, setiap defect akan dicek satu per satu untuk memastikan apakah penyelesaiannya masih on-track atau tidak. Jika sebuah defect sudah overdue, maka defect tersebut akan diarahkan kepada Test Coordinator yang menangani stream terkait untuk meng-expedite resolusi DF ini.

Khusus untuk defect yang berkaitan dengan authorization, pembuatan sebuah tiket diperlukan untuk mengeskalasi isu tersebut kepada tim support global. Pembukaan tiket dilakukan melalui ServiceNow, dan berikut merupakan gambaran umum proses pembukaan tiket tersebut.



Gambar 3.24. Tampilan awal Service Now

Pada tampilan awal aplikasi service now, pilih menu Report an Issue. Menu Report an Issue berguna untuk melaporkan sebuah issue dan membuka Incident Ticket ataupun Request Ticket. Untuk meminta akses request maka kita akan perlu Request Ticket seperti yang diperlihatkan di gambar berikut

Gambar 3.25. Request Ticket

Setelah formulir tersebut diisi, tiket akan diproses oleh Tim Support Global Freeport dan biasanya diselesaikan dalam waktu kurang dari satu hari.

Selain itu, untuk pekerjaan terkait *Defect*, setiap hari pada pukul 15.30 diadakan *Daily Defect Call* yang *dihosting* oleh penulis. Tujuan meeting ini adalah untuk memastikan bahwa seluruh DF yang ditargetkan selesai pada hari tersebut tetap on-track dalam penyelesaiannya.

Dalam diskusi ini, para Test Coordinator memberikan update secara satu per satu terkait DF yang memiliki target penyelesaian pada hari itu. Test Coordinator juga dapat mengajukan masalah terkait proses testing apabila diperlukan.

### 3.4 Kendala yang Solusi yang Ditemukan

Selama pelaksanaan kegiatan magang, penulis menghadapi sejumlah kendala dalam menjalankan tugas pengujian sistem maupun pengembangan otomatisasi. Kendala yang muncul mencakup aspek teknis, pemahaman terhadap sistem, serta penyesuaian dengan lingkungan kerja profesional. Adapun kendala yang ditemui selama proses magang antara lain sebagai berikut:

1. Pemahaman terhadap dunia bisnis menjadi tantangan awal karena latar belakang penulis berasal dari bidang Informatika. Berbagai istilah bisnis, alur proses, serta logika sistem yang digunakan belum sepenuhnya familiar, sehingga diperlukan waktu tambahan untuk memahami konteks pekerjaan secara menyeluruh.
2. Beberapa *test case* memiliki deskripsi yang kurang detail atau bersifat ambigu, yang menyebabkan kesulitan dalam pelaksanaan pengujian serta penentuan hasil yang sesuai dengan ekspektasi.
3. Penanganan terhadap beberapa laporan *defect* memerlukan waktu yang relatif lama. Bahkan, terdapat *defect* yang harus dilakukan *reopen* lebih dari satu kali karena hasil perbaikannya belum memenuhi standar yang ditetapkan oleh tim ST.
4. *Tools* pendukung yang digunakan, seperti *Power BI*, *Azure DevOps*, *ServiceNow*, dan *qTest*, belum sepenuhnya terintegrasi secara otomatis, sehingga sebagian aktivitas masih harus dilakukan secara manual.
5. Proses *retesting* sering kali memerlukan verifikasi dari berbagai pihak, yang berdampak pada lamanya *turnaround time* dalam proses penutupan *defect*.

Berdasarkan kendala-kendala tersebut, sejumlah langkah solusi diterapkan untuk meminimalkan hambatan selama pelaksanaan kegiatan magang, antara lain sebagai berikut:

1. Untuk mempercepat pemahaman terhadap sistem dan domain *multifinance*, dilakukan pembelajaran mandiri melalui dokumen *user manual*, partisipasi dalam sesi pelatihan internal, serta diskusi langsung dengan supervisor dan anggota tim senior.
2. Klarifikasi secara langsung kepada *Test Analyst* dilakukan ketika ditemukan *test case* yang kurang jelas, sehingga proses pengujian dapat dilaksanakan sesuai dengan skenario yang diharapkan.
3. Tindak lanjut dilakukan secara proaktif kepada *Product Owner* dan tim *developer* guna mempercepat proses penyelesaian *defect*, serta melakukan koordinasi ulang dengan *Test Analyst* apabila diperlukan.
4. Penetapan *retesting window* atau jadwal pengecekan rutin diterapkan agar seluruh pihak terkait mengetahui waktu pelaksanaan validasi hasil perbaikan.
5. Apabila terjadi kendala teknis pada *qTest Manager*, langkah penanganan yang dilakukan meliputi instalasi ulang aplikasi, pembersihan *cache* sistem, serta proses *login* ulang menggunakan akun resmi. Selain itu, dokumentasi *troubleshooting* disusun sebagai panduan apabila permasalahan serupa kembali terjadi.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA