



21 JUNI 2025


# Test Planning

Aplikasi Pinjam Sultan

CHANDRA MEDHIKA

QA ENGINEER

JAKARTA, INDONESIA



# Document History

Version	Date	Author	Description of Change
V1			Draft
V2			Draft- Reviewed

## Table of Contents

<b>Objective:</b> .....	<b>3</b>
<b>Scope</b> .....	<b>3</b>
<b>Test Environment</b> .....	<b>6</b>
<b>Test Data:</b> .....	<b>7</b>
<b>Test Strategy</b> .....	<b>8</b>
<b>Defect Reporting Procedure:</b> .....	<b>10</b>
<b>Test Schedule:</b> .....	<b>11</b>
<b>Test Coverages:</b> .....	<b>11</b>
<b>Risk and Mitigation:</b> .....	<b>12</b>
<b>Testing Tools</b> .....	<b>13</b>
<b>Roles and Responsibilities</b> .....	<b>13</b>
<b>Test Deliverables</b> .....	<b>14</b>
<b>Approvers List</b> .....	<b>15</b>

## Objective:

Rencana pengujian ini bertujuan untuk memastikan agar kualitas, keandalan dan fungsionalitas dari fitur seperti: Term&Conditions, Product Catalog, Product Information, Rental Shop Information, Register & Verification, Chat, Booking, Product Categorization, Product Filter dapat berfungsi dengan baik dan sesuai kebutuhan pengguna serta mencapai target unduhan 1.200.

## Scope:

Ruang lingkup Test Plan dari Aplikasi Pinjam Sultan:

### ❖ Terms and Condition:

- Pastikan konten mencakup aturan transaksi utama (kebijakan pembatalan, jaminan, dll.) berdasarkan data dari pemilik layanan.
- Verifikasi aksesibilitas tautan Terms and Conditions dari halaman utama, registrasi, dan checkout.
- Uji keterbacaan dan mekanisme penerimaan (checkbox "Saya setuju").

### ❖ Product Catalog:

- Pastikan halaman page Product Catalog dapat diakses dengan tepat
- Verifikasi bahwa setiap produk menampilkan gambar, nama, harga sewa, jenis aktivitas, lokasi sewa, rating, dan jumlah yang disewakan.
- Pastikan daftar "best selling goods and vehicles" ditampilkan dengan benar dan diperbarui.
- Uji kemampuan pengguna untuk mencari produk menggunakan search bar (misalnya, dengan kata kunci seperti "tenda" atau "mobil").
- Pastikan hasil pencarian relevan dengan input pengguna.
- Uji filter berdasarkan harga, lokasi, atau kategori aktivitas (meskipun filter rinci seperti di "Product Filter" task akan lebih spesifik, ini mencakup filter dasar di katalog).
- Verifikasi bahwa filter dapat diterapkan dan hasilnya diperbarui dengan benar.
- Pastikan pengguna dapat mengakses Product Catalog dari halaman utama atau menu navigasi.
- Uji transisi halus ke detail produk saat pengguna memilih item.
- Validasi bahwa fitur ini berfungsi dengan baik di platform mobile (Android/iOS) sesuai target aplikasi.
- Pastikan data yang ditampilkan (misalnya, harga, lokasi) sesuai dengan data yang dikumpulkan dari pemilik layanan (sesuai notes PRD).

#### ❖ Product Information:

- Verifikasi bahwa setiap produk yang dipilih dapat menampilkan informasi penting seperti gambar, harga ,spesifikasi, ulasan dan nama supliar.
- Pastikan update informasi ketersediaan sesuai dengan database.
- Pastikan fungsi filter dengan ulasan positif dan negatif bekerja dengan baik
- Verifikasi bahwa setiap ulasan yang menggunakan istilah tidak pantas dapat disensor
- Pastikan fitur "other recommendations" menampilkan saran produk terkait yang relevan.
- Uji tombol share dan pastikan pilihan platform sharing adalah yang umum digunakan.
- Uji tombol chat dan pastikan terintegrasi dengan fitur chat sehingga pengguna dapat mengirim pesan dan menerima pesan dari supliar.
- Uji tombol order dan pastikan navigasi ke halaman pembayaran dan memiliki fungsionalitas yang terintegrasi dengan fitur booking.
- Uji tombol "Add to Cart" dan pastikan navigasi ke halaman keranjang.

#### ❖ Rental Shop Information:

- Verifikasi bahwa setiap toko rental yang dipilih menampilkan informasi penting seperti nama toko, lokasi ,ulasan, produk yang disewa.
- Pastikan gambar produk yang disewakan menggunakan gambar yang berkualitas tinggi namun tetap memiliki performa load yang cepat.
- Verifikasi bahwa setiap chat antara admin dan pembeli mempunyai keterangan waktu yang real-time saat berkomunikasi dan juga informasi waktu dan tanggal kapan terakhir kali admin aktif.

#### ❖ Register and Verification:

- Uji dan pastikan kolom pengisian nama hanya menerima tipe data string.
- Uji dan pastikan kolom pengisian email hanya menerima data berupa email dengan ketentuan-ketentuan seperti kombinasi huruf dan angka, simbol "@" dan adanya sebuah domain.
- Uji dan pastikan gambar yang diunggah sesuai dengan ketentuan yang diminta dan hanya menerima gambar dengan format \*png, \*jpg, \*jpg.

### ❖ Chat:

- Verifikasi bahwa fitur chat dapat diakses dari navigasi utama, halaman Product Info, dan Service Provider Store Info.
- Pastikan tombol atau ikon chat terlihat jelas dan mudah ditemukan.
- Uji keberadaan dan akurasi informasi seperti nama toko penyedia, status online, dan detail produk (nama, harga, rating, tombol pesan langsung, tombol tambah ke keranjang).
- Verifikasi bahwa bubble chat, chatbox, dan tombol send ditampilkan dengan benar.
- Uji kemampuan pengguna untuk mengetik pesan di chatbox dan mengirimnya menggunakan tombol send.
- Pastikan pesan yang dikirim muncul di bubble chat dengan urutan yang benar.
- Uji tombol "direct message button" dan "add to cart" untuk memastikan navigasi dasar ke fitur terkait (misalnya, ke halaman pemesanan atau keranjang).
- Validasi bahwa fitur ini berfungsi dengan baik di platform mobile (Android/iOS) sesuai target aplikasi.
- Pastikan antarmuka chat intuitif dan responsif saat pengguna berinteraksi (misalnya, transisi mulus saat mengklik tombol).

### ❖ Booking:

- Verifikasi bahwa setiap produk menampilkan gambar, nama, harga sewa, stok, lokasi, jumlah yang disewakan, dan tanggal peminjaman (kalender).
- Pastikan total pembayaran, informasi batas waktu pembayaran, dan kode pembayaran ditampilkan dengan benar.
- Uji opsi pick-up method: Go-send (hanya untuk alat), alamat pelanggan, ambil sendiri, alamat rental, dan integrasi dengan Google Maps untuk navigasi.
- Verifikasi bahwa opsi dipilih dengan benar dan diperbarui di antarmuka.
- Uji semua metode pembayaran: Mandiri Virtual Account, BCA Virtual Account, CIMB Niaga Virtual Account, dan BNI Virtual Account.
- Pastikan total pembayaran sesuai dengan harga dan jumlah yang dipilih.
- Verifikasi tombol "Check payment status" menampilkan status pembayaran (misalnya, "Pending" atau "Paid").
- Uji tombol "Shop again button" untuk memastikan pengguna dapat kembali ke katalog.
- Pastikan pengguna dapat mengakses halaman Booking dari Product Information atau Product Catalog.
- Uji transisi halus ke konfirmasi pembayaran atau halaman berikutnya.
- Validasi bahwa fitur ini berfungsi dengan baik di platform mobile (Android/iOS) sesuai target aplikasi.
- Pastikan data yang ditampilkan (harga, stok, dll.) konsisten dengan data dari pemilik layanan dan mitra (Go-Send, bank).

### ❖ Product Categorization:

- Verifikasi bahwa produk direkomendasikan berdasarkan aktivitas
- Pastikan setiap kategori menampilkan daftar produk yang relevan.
- Uji keberadaan dan akurasi informasi penyedia, termasuk nama penyedia, lokasi, tingkat kepercayaan pengguna lain, dan produk yang disewakan.
- Verifikasi bahwa "how responsive is the service provider" ditampilkan (misalnya, indikator responsivitas atau rating).
- Pastikan pengguna dapat mengakses fitur Product Categorization dari halaman utama, Product Catalog, atau menu navigasi.
- Uji transisi halus ke detail produk atau halaman penyedia saat pengguna memilih kategori.
- Validasi bahwa fitur ini berfungsi dengan baik di platform mobile (Android/iOS) sesuai target aplikasi.
- Pastikan data yang ditampilkan (nama penyedia, lokasi, dll.) konsisten dengan data yang dikumpulkan dari pemilik toko rental, sesuai notes PRD.
- Uji apakah rekomendasi sesuai dengan aktivitas yang dipilih (misalnya, peralatan panjat tebing untuk kategori "mountain climbing").

### ❖ Product Filter:

- Uji dan pastikan input harga pada bilah pencarian dapat memfilter atau mencari harga yang telah ditentukan yaitu harga maksimum atau minimum.
- Uji dan pastikan input lokasi pada bilah pencarian dapat memfilter atau mencari lokasi sesuai dengan tempat/daerah yang dipilih.
- Pastikan kombinasi filter (harga + lokasi) berfungsi dengan benar.
- Verifikasi aksesibilitas dari Product Catalog dan transisi saat menerapkan filter.
- Periksa integritas data lokasi dan harga sesuai data yang dikumpulkan dari pemilik toko rental.

### Test Environment:

- Sistem Operasi yang akan digunakan dalam pengujian adalah Android dan IOS.
- Penggunaan virtual machine untuk Android emulator seperti: NOXPlayer, BlueStack, GenyMotion dan IOS emulator seperti: Ipadian, Iphone Air, Xamarin Testflight, MobieOne Studio, Smartface.
- Perangkat pengujian fisik dapat menggunakan smartphone mobile sesuai dengan OS masing-masing, yang mana akan memanfaatkan developer mode di setiap perangkat.
- Persyaratan perangkat keras dan perangkat lunak untuk menjalankan kasus uji, seperti prosesor tertentu, memori, atau kapasitas penyimpanan.

- Konektivitas jaringan dan bandwidth yang akan tersedia untuk pengujian, seperti Wi-Fi, seluler, atau koneksi kabel.
- Protokol keamanan dan metode otentikasi yang akan digunakan untuk mengakses lingkungan pengujian, seperti kata sandi, token, atau sertifikasi.

## Test Data:

- Data uji (valid, tidak valid) dapat dibuat secara manual oleh tim pengujian untuk mencakup skenario dan kasus tepi tertentu.
- Jenis data yang dibutuhkan: Data Produk, Data Penyedia Layanan, Data Pengguna, Data Transaksi, Data Chat
- Sumber Data:
  - **Data Dummy:** Dibuat oleh tim QA berdasarkan PRD (misalnya, nama produk, harga, lokasi).
  - **Data dari Tim Produk:** Data yang dikumpulkan dari pemilik layanan (sesuai notes PRD, seperti deskripsi produk, regulasi Terms and Conditions).
  - **Data Mitra:** Koordinasi dengan Go-Send untuk data pengiriman dan bank (Mandiri, BCA, dll.) untuk data pembayaran virtual account.
- Skenario Pengujian:
  - Sukses: Pemesanan "Tenda Hiking" dengan filter harga Rp100.000-Rp300.000 dan lokasi Jakarta.
  - Gagal: Pemesanan produk dengan stok 0.
  - Batas: Filter harga Rp0-Rp1.000.000.
- Persyaratan Data:
  - Data harus mencerminkan realitas (misalnya, harga realistis, lokasi valid).
  - Sertakan data untuk semua fitur (misalnya, produk untuk panjat tebing dan pantai, lokasi Jakarta dan Aceh).
  - Gunakan data dummy untuk menghindari pelanggaran privasi (misalnya, nomor telepon atau email fiktif).
  - Pastikan data siap sebelum pengujian dimulai (koordinasi dengan tim produk).



## Test Strategy:

### Step-1 Perencanaan:

- Identifikasi Tujuan Pengujian dengan memastikan tujuan mencakup verifikasi User Acceptance Criteria, pengalaman pengguna mobile (sesuai preferensi 93.3% dari PRD), dan dukungan untuk target unduhan 1.200.
- Definisikan Lingkungan Pengujian dengan menyiapkan emulator Android (versi 10-13) dan iOS (versi 14-17), perangkat fisik (misalnya, Samsung Galaxy, iPhone 12), dan jaringan Wi-Fi/4G. Gunakan data dummy (produk, penyedia, transaksi) dari bagian Test Data sebelumnya.
- Gunakan Selenium/Appium untuk otomatisasi, JMeter untuk performa (jika diperlukan), Jira untuk pelacakan bug, dan Maze untuk usability testing.
- Berkolaborasi dengan tim produk (Elfrida, Wahid) untuk data penyedia dan mitra (Go-Send, bank), serta tim desain (Adinda, Flonetta) untuk umpan balik UI/UX.
- Alokasikan 2-3 minggu untuk pengujian prototype, sesuai jadwal di Test Schedule.

### Step-2 Desain Test Case:

- Buat test case untuk skenario sukses, gagal, dan batas (edge case) berdasarkan User Acceptance Criteria. Contoh:
  - Contoh sukses: "Pengguna berhasil memesan 'Tenda Hiking' dengan filter harga Rp100.000-Rp300.000" (sukses).
  - Contoh gagal: "Pengguna mencoba memesan produk dengan stok 0, sistem menampilkan error" (gagal).
  - Contoh edge case: "Pengguna mengatur filter harga Rp0-Rp1.000.000" (batas).
- Gunakan Functional Testing (untuk fitur seperti Booking), Usability Testing (untuk navigasi Chat), Compatibility Testing (untuk mobile), Boundary Value Analysis (untuk filter), dan Equivalence Partitioning (untuk harga).
- Gunakan Excel atau TestRail untuk mendokumentasikan test case.

### Step-3 Pengaturan Environment dan Data:

- Siapkan staging environment dengan API mock jika diperlukan, dan uji konektivitas jaringan.
- Muat data dummy (produk, penyedia, transaksi) ke dalam sistem. Contoh: "Tenda Hiking" (Rp150.000, Jakarta Selatan), "Papan Selancar" (Rp200.000, Banda Aceh).
- Pastikan semua fitur dapat diakses dengan data dummy sebelum pengujian utama.

#### **Step-4 Eksekusi Pengujian Manual:**

- Uji kemudahan penggunaan fitur seperti Terms and Conditions (keterbacaan), Chat (navigasi), dan Booking (pilihan pembayaran). Gunakan Maze atau wawancara langsung.
- Jalankan test case manual untuk memverifikasi fungsionalitas dasar (misalnya, filter harga di Product Filter, tampilan katalog di Product Catalog).
- Uji aplikasi di berbagai perangkat mobile (Android 10+, iOS 14+) untuk memastikan kompatibilitas.
- Catat bug atau isu di Jira dengan detail (langkah, ekspektasi, hasil aktual).

#### **Step-5 Eksekusi Pengujian Automation:**

- Buat skrip otomatisasi menggunakan Selenium/Appium untuk fitur berulang seperti pencarian di Product Catalog, filter di Product Filter, dan tombol aksi di Product Information.
- Eksekusi skrip untuk menguji integrasi antar fitur (misalnya, dari Product Categorization ke Booking).
- Periksa laporan otomatisasi untuk kegagalan dan bandingkan dengan ekspektasi.
- Tambahkan hasil otomatisasi ke Jira.

#### **Step-6 Exploratory Testing:**

- Tester bereksperimen dengan alur tidak terdefinisi (misalnya, kombinasi filter aneh atau input tidak valid di Booking).
- Dokumentasikan temuan seperti crash atau perilaku aneh.
- Masukkan isu ke Jira untuk analisis lebih lanjut.

#### **Step-7 Pengujian Performa:**

- Gunakan JMeter untuk mensimulasikan 1.200 pengguna bersamaan (jika memungkinkan pada prototype).
- Uji waktu respons (target < 2 detik) dan stabilitas aplikasi.
- Catat hasil performa dan rekomendasi perbaikan.

#### **Step-8 Analisis Hasil Dan Pelaporan:**

- Gabungkan hasil dari tes manual, otomatisasi, exploratory, dan performa (jika ada).
- Urutkan berdasarkan severity (kritis, mayor, minor) dan prioritas perbaikan.

- Buat dokumen ringkasan dengan status test case (lolos/gagal), daftar bug, dan rekomendasi.
- Presentasikan hasil ke tim produk dan desain untuk tindak lanjut.

### Step-9 Retest dan Penutupan:

- Jalankan ulang test case untuk bug yang telah diperbaiki.
- Pastikan 90% test case lolos, semua User Acceptance Criteria terpenuhi, dan tidak ada bug kritis.
- Setujui dokumen Test Plan dengan Approvers List (Elfrida, Wahid) dan arsipkan hasil.

Izin akses dan peran anggota tim yang akan menggunakan lingkungan pengujian, seperti penguji, pengembang, atau pemangku kepentingan.

Name	Credential	Environment
<b>Developer</b>		
<b>QA</b>	Username: Password:	pre-staging
<b>QA/Stakeholders</b>	Username: Password:	staging
<b>End User</b>		

### Defect Reporting Procedure:

- Jika defect ditemukan oleh tester selama pengujian, catat langkah, ekspektasi, hasil.
- Laporkan di Jira dengan judul, ID, deskripsi, severity (kritis/mayor/minor), prioritas, tangkapan layar, lingkungan, tanggal.
- Tetapkan status "New", assignee ke pengembang, notifikasi tim dalam 24 jam.
- Product owner dan pengembang tinjau dalam 48 jam, ubah ke "In Progress" atau "Rejected".
- Pengembang perbaiki dalam 3-5 hari (kritis/mayor) atau 7-10 hari (minor), uji internal.
- QA retest dalam 2 hari, ubah ke "Closed" atau "Reopened".
- Arsipkan defect, buat laporan ringkasan, setujui dengan Approvers.
- Tunda pelaporan (tenggat 24 jam), severity salah (klarifikasi), retest gagal (eskalisasi).
- Semua defect kritis/mayor ditutup, laporan disetujui.

Defect dan Orang yang Melaporkan:

Defect Process	POC
Front End	Mr. X
Back End	Mr. Y
Dev Ops	Mr. Z
Mobile App	Mr. M

## Test Schedule:

Dua sprint (Satu Sprint = 5 hari kerja) akan dibutuhkan untuk menguji aplikasi.

Berikut adalah jadwal uji coba yang direncanakan untuk proyek tersebut:

No	Task	Date	Total Days
1	Perencanaan dan Persiapan	1 Jul 2025 - 2 Jul 2025	1
2	Desain Test Case	3 Jul 2025 - 4 Jul 2025	1
3	Set Environment	7 Jul 2025	
4	Eksekusi Manual	8 Jul 2025 - 11 Jul 2025	3
5	Eksekusi Automation	14 Jul 2025 - 16 Jul 2025	2
6	Exploratory Testing	17 Jul 2025	
7	Pengujian Performa	18 Jul 2025	
8	Retest dan Penutupan	21 Jul 2025 - 22 Jul 2025	1

- **Milestone 1:** Lingkungan siap (7 Juli 2025).
- **Milestone 2:** 50% test case dieksekusi (11 Juli 2025).
- **Milestone 3:** Semua tes selesai (18 Juli 2025).
- **Milestone 4:** Penutupan dan persetujuan (22 Juli 2025).

## Test Coverages:

Test Coverages menentukan apa yang akan diuji, tingkat kedalaman pengujian, dan metrik keberhasilan. Fokusnya adalah memastikan cakupan menyeluruh pada fungsionalitas, usability, kompatibilitas, dan batas input, dengan mempertimbangkan tahap prototype.

### ❖ Cakupan Fungsionalitas:

- Terms and Conditions: Akses, keterbacaan, penerimaan (100% UAC).
- Product Catalog: Tampilan, pencarian, filter (100% UAC).
- Product Information: Detail, tombol aksi, berbagi (100% UAC).
- Chat: Akses, konten, pengiriman (100% UAC).

- Booking: Informasi, pengambilan, pembayaran (100% UAC).
  - Product Categorization: Rekomendasi, penyedia (100% UAC).
  - Product Filter: Search, harga, lokasi (100% UAC).
- ❖ Cakupan Non-Fungsionalitas:
- Usability: 90% pengguna baru selesai alur dalam 5 menit.
  - Compatibility: 100% fitur stabil di Android 10-13, iOS 14-17.
  - Performance: 95% transaksi < 2 detik (jika diuji).
  - Security: 80% input sensitif divalidasi.
- ❖ Cakupan Skenario:
- Sukses (100%), gagal (100%), batas (80%).
- ❖ Tingkat Cakupan:
- 100% fitur, 90% test case, 70% code (opsional), 100% data.
- ❖ Batasan Cakupan:
- Tidak mencakup desain mendalam, performa rinci, integrasi mitra, fitur di luar prototype.
- ❖ Metrik Keberhasilan:
- 100% UAC, usability  $\geq$  85%, kompatibilitas 100%, tanpa bug kritis.
- ❖ Risiko dan Mitigasi:
- Data terbatas (gunakan dummy), waktu edge case (prioritas utama), kompatibilitas (uji awal).

## Risiko dan Mitigasi:

Berikut adalah beberapa risiko umum dan strategi mitigasi untuk mengatasinya.

No	Risiko	Mitigasi
1.	Penyerahan Prototype Terlambat	Persiapan awal (1-2 Jul), update status mingguan, sesuaikan jadwal
2.	Data Tidak Lengkap	Gunakan data dummy, rapat 30 Jun dengan tim produk
3.	Banyak Defect Kritis	Prioritas tes awal (8-11 Jul), cadangan 2 hari (21-22 Jul)
4.	Kompatibilitas Terbatas	Uji awal (8 Jul) di emulator/perangkat utama

5.	Koordinasi Buruk	Rapat harian (7-22 Jul), notifikasi Jira.
6.	Waktu Edge Case	Exploratory testing (17 Jul), prioritas utama

## Testing Tools:

- **Manual Testing:** Google Sheets/Excel (dokumentasi test case, hasil), Maze (usability), Wawancara Langsung (umpan balik).
- **Automated Testing:** Selenium (fungsi browser), Appium (aplikasi mobile).
- **Defect Tracking:** Jira (pelacakan utama), Bugzilla (cadangan).
- **Usability Testing:** Maze (tes pengguna), Wawancara Langsung (pasca-peluncuran).
- **Performance Testing:** JMeter (simulasi beban), LoadRunner (opsional).
- **Security Testing:** OWASP ZAP (validasi input dasar).
- **Kriteria Pemilihan:** Kompatibel mobile, mudah digunakan, gratis, terintegrasi.
- **Rencana Penggunaan:** Siapkan 1-2 Jul, manual 8-11 Jul, otomatis 14-16 Jul, dll.
- **Risiko dan Mitigasi:** Kompatibilitas (uji awal), keahlian (latih tim), infrastruktur (cadangan lokal).
- **Kriteria Kelulusan:** Alat berfungsi, data akurat, tanpa gangguan signifikan.

## Roles and Responsibilities:

Name	Role	Responsibilities
	(SQA) Lead	<ul style="list-style-type: none"> <li>- Merancang Test Plan</li> <li>- Tentukan Ruang Lingkup dan Tujuan</li> <li>- Memantau kemajuan pengujian dan memastikan kriteria kelulusan</li> <li>- Meninjau dan menyetujui test cases dan test data</li> <li>- Menyusun laporan akhir dan mendapatkan persetujuan dari Approvers</li> </ul>
	SQA Engineer	<ul style="list-style-type: none"> <li>- Menjalankan test case manual (usability, fungsional)</li> <li>- Mengembangkan dan menjalankan script automation (Appium/Selenium)</li> <li>- Melakukan exploratory testing</li> <li>- Mendokumentasikan defect di Jira sesuai prosedur pelaporan</li> <li>- Melakukan retest defect yang diperbaiki</li> <li>- Memberikan umpan balik pada lingkungan pengujian dan data dummy</li> </ul>

	Performance Tester	<ul style="list-style-type: none"> <li>- Mengatur dan menjalankan JMeter untuk simulasi beban</li> <li>- Mengukur waktu respons (&lt; 2 detik) dan stabilitas untuk 1.200 pengguna</li> <li>- Melaporkan hasil performa dan rekomendasi perbaikan</li> </ul>
--	--------------------	--

## Test Deliverables:

Berikut adalah yang akan disampaikan kepada klien:

Deliverables	Description	Completion Date
<b>Test Plan</b>	Rincian tentang ruang lingkup proyek, strategi pengujian, jadwal pengujian, dan hasil pengujian	
<b>Functional Test Cases</b>	Kasus uji dibuat untuk cakupan yang ditentukan.	
<b>Defect Report</b>	Deskripsi rinci tentang cacat yang diidentifikasi beserta tangkapan layar dan langkah-langkah untuk mereproduksinya setiap hari.	
<b>Summary Report</b>	Laporan ringkasan proyek	

Approvers List:

No	Name	Role	Approver Review	Date