

## **SOP – Technical Scripting Automation Test**



Document Version: 1.0

Effective Date: [Tanggal Revisi]

Prepared By: Suhendar - Automation Test Lead

Approved By: ... – IT QA Manager

## 1. Purpose

Memberikan panduan teknis standar dalam pembuatan, pemeliharaan, dan eksekusi skrip automation testing agar kode konsisten, mudah dibaca, dapat di-maintain, dan sesuai dengan framework yang digunakan secara berkelanjutan.

## 2. Scope

SOP ini berlaku untuk semua anggota QA Automation Team yang melakukan:

- Penulisan skrip baru untuk Web, Mobile, API.
- Pemeliharaan skrip untuk menjalankan regression test.

## 3. Tools & Framework

Automation tester wajib menggunakan tool & framework resmi yang telah disetujui:

- Framework: Katalon Studio V10 (Free).
- Report : iTextPDF
- Language: Groovy, Java
- Repository : GitHub
- Version Control: Git

## 4. Scripting Standards

### 4.1 Penamaan (Naming Convention)

- Class Name: PascalCase (contoh: LoginPageTest).
- Method Name: camelCase (contoh: verifyLoginFunction).
- Variable Name: camelCase (contoh: userNameInput).
- Hindari singkatan tidak jelas.

## 4.2 Struktur Kode

Page Object Model (POM) + WebActions digunakan untuk memisahkan logika test automation menjadi dua lapisan: POM berfungsi menyimpan representasi halaman web (locator dan action spesifik per halaman), sedangkan WebActions / MobileActions berisi kumpulan keyword generik yang dapat digunakan ulang seperti click, setText, verify, dan screenshot. Dengan pendekatan ini, script test case menjadi lebih singkat, mudah dibaca, serta lebih mudah dirawat karena jika ada perubahan elemen atau fungsi umum, cukup diperbarui di satu tempat saja tanpa perlu mengubah banyak test case. Struktur ini menjadikan automation framework lebih scalable, reusable, dan maintainable.

Gambaran Struktur File :

```
Keywords/
├── customKeywords/
│   └── WebActions.groovy ← helper keyword umum
├── pageObjects/
│   ├── LoginPage.groovy
│   ├── DashboardPage.groovy
│   └── LogoutPage.groovy
Test Cases/
└── TC_Login_Dashboard_Logout.groovy
Object Repository/
├── Page_Login/
│   ├── input_username
│   ├── input_password
│   └── btn_login
├── Page_Dashboard/
│   └── lbl_welcome
├── Page_Logout/
└── btn_logout
```

## 4.3 Handling Test Data

- Simpan test data di file Excel dengan penamaan TD\_NamaFeature.
- Jangan hardcode data input di dalam skrip.

## 4.4 Standarisasi Penamaan Test Object

Gunakan penamaan test object yang konsisten dan mudah dipahami oleh semua anggota tim, sehingga memudahkan kolaborasi dan mengurangi risiko kesalahan. Standarisasi ini memastikan setiap object dapat cepat dicari saat automation script bertambah banyak, tidak menimbulkan ambiguitas antara object yang mirip, serta tetap selaras dengan praktik terbaik QA melalui naming convention yang jelas, rapi, dan profesional.

### • Struktur Dasar

**<Page/Module>/<ELEMENTTYPE>\_<LogicalName>**

Penjelasan Struktur :

1. Page/Module → Nama folder halaman atau modul tempat object berada.  
Contoh: Login, Dashboard, Cart, API.

2. ELEMENTTYPE → Tipe element UI menggunakan prefix huruf besar semua.  
Contoh: BTN, INPUT, TXT, DD, CB, RB, LINK, ICON, TBL, MDL, IMG, TAB, ALT.
3. LogicalName → Deskripsi singkat sesuai fungsi element, ditulis dengan PascalCase.  
Contoh: Login, Password, WelcomeMsg, AddToCart.

- **Aturan Umum**

1. Gunakan Bahasa Inggris → Semua nama object harus dalam bahasa Inggris agar universal.
2. Hindari singkatan aneh / angka → Jangan gunakan BTN1 atau INPUTX. Gunakan nama jelas: Login\_BTN\_Login.
3. Gunakan PascalCase pada LogicalName setelah underscore.
4. ELEMENTTYPE harus huruf besar (UPPERCASE) sesuai Hungarian Style.

- **Test Object dengan Hungarian**

Berikut ini adalah tabel Test Object dengan Hungarian Style yang harus diikuti sebagai standarisasi penamaan Test Object.

ELEMENT TYPE	Nama Lengkap	Deskripsi Penggunaan	Contoh Nama Object
<b>BTN</b>	Button	Dipakai untuk tombol aksi.	Login_BTN_Login, Cart_BTN_Checkout
<b>INPUT</b>	Input / TextField	Field untuk teks biasa (username, email, dsb).	Login/INPUT_Email Login/ INPUT_Password
<b>TXT</b>	Text / Label	Elemen statis berupa teks, label, atau pesan.	Dashboard/TXT_WelcomeMsg
<b>LINK</b>	Hyperlink	Tautan navigasi.	Login/LINK_ForgotPassword
<b>ICON</b>	Icon / Symbol	Ikon navigasi atau ikon fungsional.	Dashboard/ICON_Profile
<b>DD</b>	Dropdown	Elemen pilih nilai dari daftar (combobox).	Register/DD_Country
<b>CB</b>	Checkbox	Elemen centang pilihan ganda.	Register/CB_AgreeTerms
<b>RB</b>	Radio Button	Pilihan tunggal dari sekumpulan opsi.	Register/RB_GenderMale
<b>TBL</b>	Table	Tabel data utama.	Cart/TBL_ItemList
<b>ROW</b>	Table Row	Baris dalam tabel.	Cart/ROW_Item
<b>COL</b>	Table Column	Kolom dalam tabel.	Cart/COL_Price
<b>MDL</b>	Modal / Popup	Dialog konfirmasi, popup window.	Checkout/MODAL_Confirm
<b>IMG</b>	Image	Elemen gambar.	Home/IMG_Logo

<b>TAB</b>	Tab / Menu	Tab navigasi atau menu.	Report/TAB_Transactions
<b>ALT</b>	Alert / Notification	Pesan peringatan, error, atau notifikasi.	Login/ALT_Credentials

#### 4.4 Logging & Reporting

- Jalankan menggunakan test suite
- Centang berdasarkan test case yang ingin diuji
- Gunakan Standarisasi PDF Report.
- Pastikan setiap test case memberikan laporan.

#### 5. Development Workflow

1. Buat branch baru dari develop atau feature sesuai task.
2. Tulis skrip mengikuti standar.
3. Self-review dan jalankan di lokal.
4. Push ke repository dan buat Pull Request.
5. Code review oleh QA Lead/Peer.
6. Merge setelah approval.

#### 6. Test Execution

- Eksekusi skrip di local sebelum commit.
- Gunakan filter untuk memilih test case yang relevan.

#### 7. Maintenance

- Review skrip setiap sprint atau minimal setiap bulan.
- Refactor skrip yang terlalu panjang atau kompleks.
- Hapus skrip yang obsolete.

#### 8. References

- BAD Test Strategy Automation