**Technical Test – QA**

1. **Manual Test Case**

| **No** | **Test Case ID** | **Test Case Descriptions** | **Precondition** | **Test Data** | **Test Steps** | **Post Condition** | **Expected Result** | **Actual Result** | **Screenshoot** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC\_Login\_001 | Untuk memastikan bahwa Valid Email dan Password | Pengguna sudah terdaftar dan memiliki kredensial yang valid. | Email : [admin@example.com](mailto:admin@example.com)  Password : password | 1. Buka halaman Sign In pada URL : <https://lapor.folkatech.com/> 2. Masukkan email yang valid. 3. Masukkan password yang valid. 4. Klik tombol Sign In. | Pengguna diarahkan ke halaman dashboard setelah login berhasil. | Login Berhasil |  |  | Passed |
| 2 | TC\_Login\_002 | Untuk memastikan bahwa Email Invalid | Pengguna gagal login dengan kredensial yang salah, dan pesan error yang sesuai muncul. | Email : [cindy@example.com](mailto:cindy@example.com) | 1. Buka halaman Sign In pada URL : <https://lapor.folkatech.com/> 2. Masukkan email yang tidak valid. 3. Masukkan password yang valid. 4. Klik tombol Sign In. | Muncul pesan error “**Login Gagal! Akun tidak ada.**” | Menampilkan pesan error **“Login Gagal! Akun tidak ada.”** |  |  | Failed |
| 3 | TC\_Login\_003 | Untuk memastikan bahwa Password Invalid | Pengguna gagal login dengan kredensial yang salah, dan pesan error yang sesuai muncul. | Password : Cindy | 1. Buka halaman Sign In pada URL : <https://lapor.folkatech.com/> 2. Masukkan email yang valid. 3. Masukkan password yang tidak valid. 4. Klik tombol Sign In. | Muncul pesan error **“Login Gagal! Kata sandi salah.”** | Menampilkan pesan error **“Login Gagal! Kata sandi salah.”** |  |  | Failed |
| 4 | TC\_Login\_004 | Kosongkan Email dan Password | Tidak ada input yang dimasukkan. | Tidak ada input yang dimasukkan. | 1. Buka halaman Sign In pada URL : <https://lapor.folkatech.com/> 2. Tidak mengisi kolom email dan password. 3. Klik tombol Sign In. | Muncul pesan error **“Email harus diisi”** dan **“Password harus diisi”.** | Menampilkan pesan error **“Email harus diisi”** dan **“Password harus diisi”.** |  |  | Failed |
| 5 | TC\_Login\_005 | Tampilkan password | Pengguna memasukkan password di kolom **"Password".** | Email : [admin@example.com](mailto:admin@example.com)  Password : password | 1. Buka halaman Sign In pada URL : <https://lapor.folkatech.com/> 2. Masukkan password. 3. Klik ikon mata pada kolom password di sebelah kanan. | Password terlihat jelas di kolom password. | Password ditampilkan. |  |  | Passed |
| 6 | TC\_Login\_006 | Password Masked Setelah Klik Mata Lagi | Pengguna sudah melihat password setelah klik ikon mata. | Email : [admin@example.com](mailto:admin@example.com)  Password : password | 1. Buka halaman Sign In pada URL : <https://lapor.folkatech.com/> 2. Masukkan password. 3. Klik kembali ikon mata pada kolom password di sebelah kanan. | Password kembali disembunyikan (ditampilkan dalam bentuk titik). | Password disembunyikan. |  |  | Passed |
| 7 | TC\_Dahboard\_007 | Memuat Halaman Dashboard Setelah Login | Pengguna berhasil Login. | Email : [admin@example.com](mailto:admin@example.com)  Password : password | 1. Login menggunakan akun yang valid. 2. Sistem menampilkan halaman dashboard setelah login berhasil. | Pengguna di halaman dashboard | Halaman dashboard muncul dengan informasi jumlah laporan, statistik, dan performa. | Halaman dashboard muncul sesuai dengan yang diharapkan. |  | Passed |
| 8 | TC\_Dashboard\_008 | Verifikasi Tampilan Statistik di Dashboard | Pengguna berada di halaman dashboard | Data laporan dari database | 1. Bandingkan jumlah laporan yang belum diproses pada dashboard dengan data di database 2. Periksa statistik laporan lainnya. | Data di dashboard sesuai | Data statistik dan jumlah laporan sesuai dengan data di database. | Data statistik dan jumlah laporan sesuai. |  | Passed |
| 9 | TC\_Dashboard\_009 | Filter laporan berdasarkan tanggal mulai dan tanggal akhir. | User sudah login dan berada di halaman Dashboard | - Tanggal mulai: 09/09/202  - Tanggal akhir: 10/09/2024 | 1. Klik tombol "Filter" pada dashboard 2. Isi tanggal mulai dan tanggal akhir 3. Klik tombol "Filter" | Menampilkan data laporan sesuai tanggal yang difilter | Data laporan yang ditampilkan sesuai dengan rentang tanggal yang dipilih |  |  | Passed |
| 10 | TC\_Dahsboard\_010 | Filter laporan berdasarkan provinsi tertentu | User sudah login dan berada di halaman Dashboard | Provinsi: Jawa Timur | 1. Klik tombol "Filter" pada dashboard 2. Isi Provinsi “Jawa Timur” 3. Klik tombol "Filter" | Menampilkan data laporan sesuai provinsi yang dipilih | Data laporan yang ditampilkan sesuai dengan provinsi yang difilter. |  |  | Passed |
| 11 | TC\_Dashboard\_011 | Filter laporan berdasarkan kategori tertentu | User sudah login dan berada di halaman Dashboard | Kategori : Category C | 1. Klik tombol “Filter” pada dashboard. 2. Isi kategori “Category C” 3. Klik tombol “Filter” | Menampilkan data laporan sesuai kategori yang dipilih | Data laporan yang ditampilkan sesuai dengan kategori yang difilter. |  |  | Passed |
| 12 | TC\_Dahsboard\_012 | Reset filter laporan untuk menampilkan semua data | User sudah login dan berada di halaman Dashboard | Tidak ada test data | Klik tombol “Reset” pada dashboard untuk menghapus semua filter. | Menampilkan semua data laporan tanpa filter | Semua data laporan ditampilkan kembali setelah filter direset |  |  | Passed |
| 13 | TC\_Dahsboard\_013 | Menampilkan detail laporan dengan klik pada salah satu kartu laporan | User sudah login dan berada di halaman Dashboard | Tidak ada test data | 1. Klik salah satu kartu laporan (misal "Kurang dari 1 Hari Belum Diproses") 2. Lihat daftar yang ditampilkan. 3. Klik tombol “Detail” pada salah satu daftar nama yang ditampilkan. | Menampilkan detail laporan sesuai dengan nama yang diklik. | Detail laporan ditampilkan sesuai dengan kategori laporan yang dipilih (misal "Cindy Thresya") |  |  | Passed |

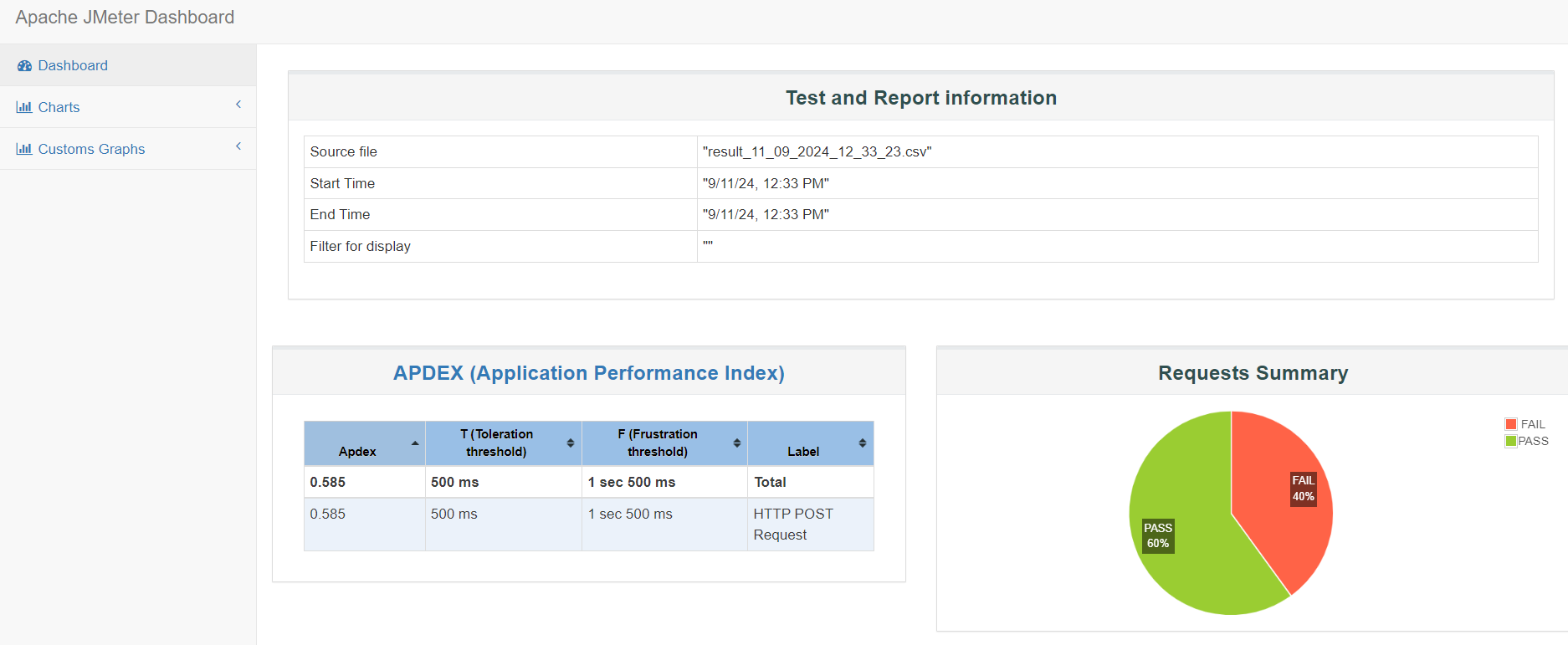
1. **API Testing**

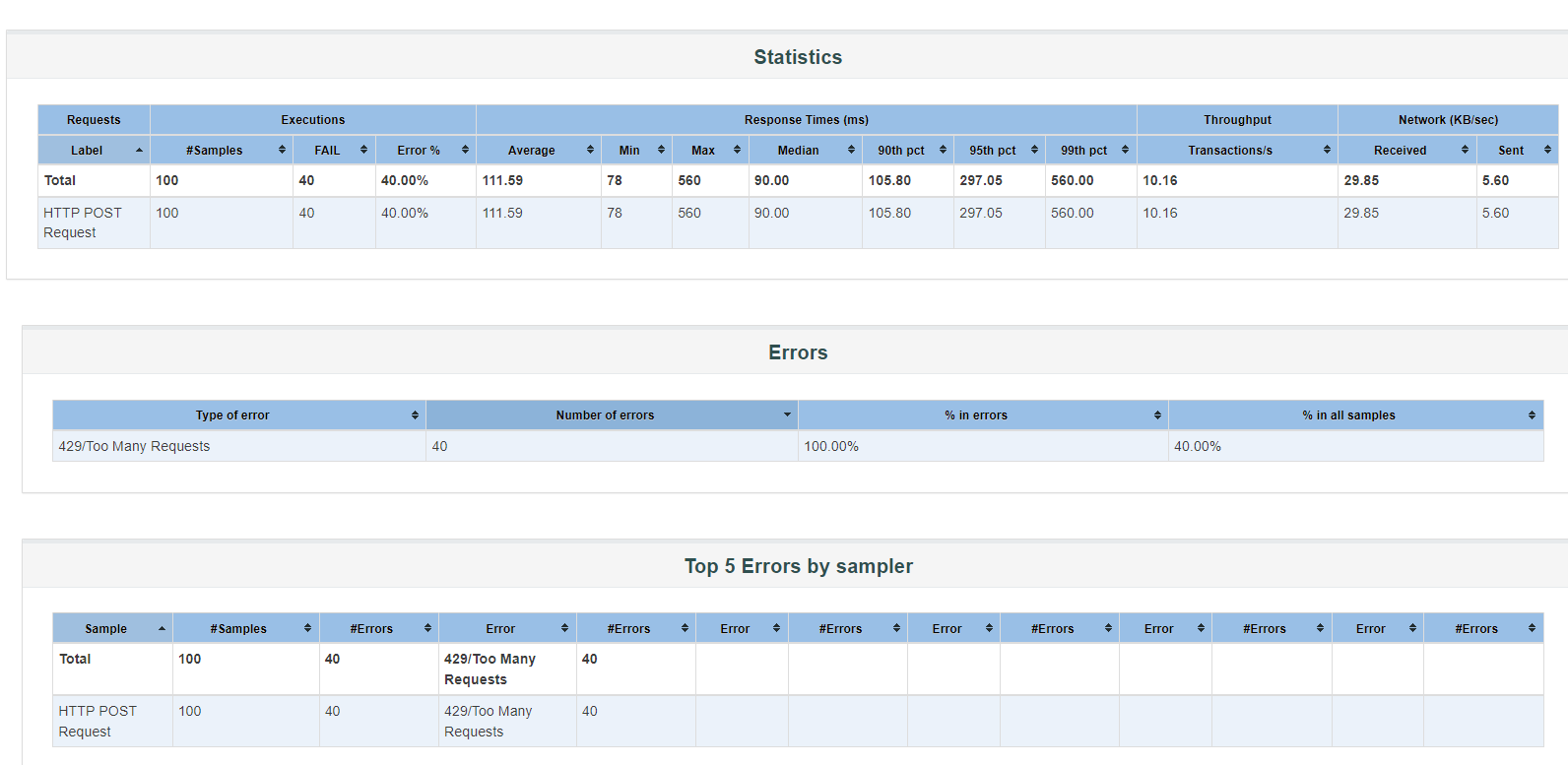
| **No** | **Test Case ID** | **Title** | **Test Case Descriptions** | **Preconditions** | **Test Steps** | **Postconditions** | **Expected Result** | **Expected Response** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | TC\_API\_001 | Testing API Register Endpoint | Verifikasi apakah pengguna dapat melakukan registrasi menggunakan API Register dengan data yang valid dan menerima respons yang benar. | 1. Endpoint API Register sudah tersedia. 2. Tools postman siap untuk menguji API. | 1. Buka Postman untuk pengujian API. 2. Pilih metode POST dan masukkan URL endpoint: {{api}}/api/register. 3. Pada tab Body, pilih x-www-form-urlencoded dan masukkan parameter berikut:  * **name:** Cindy Thresya * **email:** [cindythresya@gmail.com](mailto:cindythresya@gmail.com) * **password:** Qwerty123! * **password\_confirmation:** Qwerty123! * **phone:** 0813  1. Klik tombol send untuk mengirim request. 2. Periksa respons dari server. 3. Verifikasi data respons:  * **code:** 200 * **message:** Usre registered successfully * **success:** true * **data:** name, phone\_number, email, created\_at, id. | Akun baru dibuat dalam database dengan data yang sesuai. | Data yang diterima sesuai dan server memberikan respons dengan status **200 OK.** | {  "code": 200,  "message": "user registered successfully",  "success": true,  "data": {  "name": "Cindy Thresya",  "phone\_number": "0813",  "email": "cindythresya@gmail.com",  "created\_at": "2024-09-11T04:20:36.000000Z",  "id": 5  }  } | Passed |
| 2 | TC\_API\_002 | Testing API Login Endpoint | Verifikasi apakah pengguna dapat login menggunakan API Login dengan data yang valid dan menerima respons yang benar. | 1. Endpoint login Sudah tersedia. 2. Tools postman digunakan untuk menguji API. 3. Akun pengguna sudah terdaftar pada sistem. | 1. Buka Postman atau tool lain untuk pengujian API. 2. Pilih metode **POST** dan masukkan URL endpoint: {{api}}/api/login. 3. Pada tab **Body**, pilih x-www-form-urlencoded dan masukkan parameter berikut:  * **email**: [cindythresya@gmail.com](mailto:cindythresya@gmail.com) * **password**: Qwerty123! * **reg\_id**: 12345  1. Klik tombol **Send** untuk mengirim request. 2. Periksa respons dari server. 3. Verifikasi data respons:  * **code**: 200 * **token**: Token yang dihasilkan untuk sesi login * **phone\_number**: "0813" * **last\_login**: Timestamp login terakhir | Pengguna berhasil login dan token autentikasi dikembalikan. | Server memberikan respons **200 OK.** | {  "created\_at": "2024-09-11T04:20:36.000000Z",  "updated\_at": "2024-09-11T04:22:10.000000Z",  "phone\_number": "0813",  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...",  "last\_login": "2024-09-11 11:22:10",  "is\_active": 1  } | Passed |
| 3 | TC\_API\_003 | Testing API Report Endpoint | Validasi keberhasilan pembuatan ticket laporan menggunakan endpoint POST /api/report dengan token Bearer yang valid. | 1. Server API beroperasi. 2. Pengguna memiliki token Bearer yang valid. | 1. Siapkan request API:  * Metode: POST * URL: /api/report * Autentikasi: Bearer Token  1. Atur header:  * Content-Type: application/json * Authorization: Bearer token  1. Siapkan body request dengan field yang diperlukan. 2. Kirim request. | Ticket laporan baru harus dibuat dalam sistem dengan detail yang sesuai. | Kode Status HTTP: 200 Created | {  "message": "Ticket created successfully",  "ticketId": "<some-id>",  "status": "open"  } | Passed |
| 4 | TC\_API\_004 | Buat Ticket Laporan - Data Form Valid | Menguji keberhasilan pembuatan ticket laporan menggunakan endpoint POST /api/report dengan form-data yang valid. | 1. Server API aktif. 2. Pengguna memiliki token Bearer yang valid. 3. Format file gambar didukung oleh API. | 1. Siapkan request API:  * Metode: POST * URL: /api/report * Autentikasi: Bearer Token  1. Atur header:  * Content-Type: multipart/form-data * Authorization: Bearer token  1. Siapkan form-data dengan field yang valid seperti: fullname, lat, lng, userfile, dll. 2. Kirim request. | Ticket laporan baru harus dibuat dalam sistem dengan detail yang disediakan. | Kode Status HTTP: 200 Created | json { "message": "Ticket berhasil dibuat", "ticketId": "<id-ticket>", "status": "open" } | Passed |
| 5 | TC\_API\_005 | Mengirim data yang Invalid | Menguji apakah API memberikan respons error ketika report\_category\_id tidak valid. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Siapkan request API:  * Metode: POST * URL: /api/report * Autentikasi: Bearer Token  1. Atur header:  * Content-Type: multipart/form-data * Authorization: Bearer token  1. Siapkan form-data dengan field yang invalid seperti: mengosongkan report\_category\_id. 2. Kirim request. | Laporan tidak berhasil dibuat dan error pesan terkait ID kategori muncul. | Status code: 400 Bad Request | {"status": false, "message": "Data kategori tidak ditemukan."} | Failed |
| 6 | TC\_API\_006 | Missing Required Field (fullname) | Menguji apakah API memberikan respons error ketika field fullname kosong. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Biarkan field fullname kosong. 2. Kirim request ke endpoint /api/report dengan metode POST. | Laporan tidak berhasil dibuat, dan API mengembalikan pesan error terkait missing field. | Status code: 500 Internal server Error. | {"status": false, "message": "Field fullname wajib diisi."} | Failed |
| 7 | TC\_API\_007 | Invalid Data Type for gender | Menguji apakah API memberikan respons error ketika tipe data pada field gender kosong. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Gunakan tipe data yang salah untuk gender kosong. 2. Kirim request ke endpoint /api/report dengan metode POST. | Laporan tidak berhasil dibuat, dan API mengembalikan pesan error terkait tipe data yang salah. | Status code: 500 Internal server Error. | {"status": false, "message": "Tipe data gender tidak valid."} | Failed |
| 8 | TC\_API\_008 | Invalid province\_id | Menguji apakah API memberikan respons error ketika province\_id tidak ditemukan atau invalid. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Pastikan province\_id kosong. 2. Kirim request ke endpoint /api/report dengan metode POST. | Laporan tidak berhasil dibuat dan API memberikan pesan error terkait province\_id. | Status code: 200 OK  Response body: {"status": false, "message": "Data provinsi tidak ditemukan."} | {"status": false, "message": "Data provinsi tidak ditemukan."} | Failed |
| 9 | TC\_API\_009 | Invalid tipe\_korban id | Menguji apakah API memberikan respons error ketika tipe\_korban diisi dengan data tidak valid. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Pastikan semua field diisi dengan benar, tetapi tipe\_korban kosong. 2. .Kirim request POST. | Laporan tidak berhasil dibuat dan API memberikan pesan error terkait tipe data tipe\_korban. | Status code: 500 Internal server Error. | {"status": false, "message": "Tipe data tipe\_korban tidak valid."} | Failed |
| 10 | TC\_API\_010 | Missing description | Menguji apakah API memberikan respons error ketika description diisi dengan data kosong. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. astikan semua field diisi dengan benar, tetapi description kosong. 2. Kirim request POST. | Laporan tidak berhasil dibuat dan API memberikan pesan error terkait tipe data atau panjang description. | Status code: 500 Internal server Error. | {"status": false, "message": "Panjang field description tidak valid."} | Failed |
| 11 | TC\_API\_011 | Missing lat | Menguji apakah API memberikan respons error ketika field lat dikosongkan. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Pastikan semua field diisi dengan benar kecuali lat yang dikosongkan. 2. Kirim request ke endpoint /api/report dengan metode POST. | Laporan tidak berhasil dibuat dan API memberikan pesan error terkait lat yang kosong. | Status code: 500 Internal server Error. | {"status": false, "message": "Field lat wajib diisi."} | Failed |
| 12 | TC\_API\_012 | Missing lng | Menguji apakah API memberikan respons error ketika field lng dikosongkan. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Pastikan semua field diisi dengan benar kecuali lng yang dikosongkan. 2. Kirim request ke endpoint /api/report dengan metode POST. | Laporan tidak berhasil dibuat dan API memberikan pesan error terkait lng yang kosong. | Status code: 500 Internal server Error. | {"status": false, "message": "Field lng wajib diisi."} | Failed |
| 13 | TC\_API\_013 | Missing address\_location | Menguji apakah API memberikan respons error ketika field address\_location dikosongkan. | Endpoint siap diakses dan API key valid jika dibutuhkan. | 1. Pastikan semua field diisi dengan benar kecuali address\_location yang dikosongkan.   Kirim request ke endpoint /api/report dengan metode POST. | Laporan tidak berhasil dibuat dan API memberikan pesan error terkait address\_location yang kosong. | Status code: 500 Internal server Error. | {"status": false, "message": "Field address\_location wajib diisi."} | Failed |

| **LAMPIRAN** | | |
| --- | --- | --- |
| **Positive Scenario** | **TC\_API\_001** |  |
| **TC\_API\_002** |  |
| **TC\_API\_003** |  |
| **TC\_API\_004** |  |
| **TC\_API\_004** |  |
| **Negative Scenario** | **TC\_API\_005** |  |
| **TC\_API\_006** |  |
| **TC\_API\_007** |  |
| **TC\_API\_008** |  |
| **TC\_API\_009** |  |
| **TC\_API\_010** |  |
| **TC\_API\_011** |  |
| **TC\_API\_012** |  |
| **TC\_API\_013** |  |

1. **Performance Testing**

Dari hasil testing sesuai dengan test plan Dimana simulasi beban 100 pengguna yang mengirimkan laporan secara bersamaan didapatkan hasil dari 100 data yang dikirimkan 60% data berhasil dan 40% data gagal. Dari 40% data yang gagal mengalami error to many request artinya website atau aplikasi membatasi total request secara bersamaan.

****

****

