

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang dilakukan pada posisi *Fullstak Developer* di LPPM. Dalam masa magang, pengawasan terhadap peserta magang adalah Bapak Dr. Ir. P. M. Winarno, M.Kom. yang menjabat sebagai direktur LPPM.

3.2 Tugas yang Dilakukan

Pelaksanaan Magang diawali dengan pembelajaran tentang Permenkop-kukm-no-8-tahun-2023 yang diberikan oleh Direktur LPPM yang bertujuan untuk mengetahui definisi, jenis dan aturan koperasi. *Training* melakukan pembelajaran tentang bahasa pemrogramman *ReactJS* dan *NodeJS* secara mandiri.

Direktur LPPM meminta *training* untuk membuat laporan proposal situs web koperasi. Tugas ini dibagi menjadi beberapa bagian dimana *training* mendapatkan kesempatan untuk mengerjakan bab satu, lalu sisanya dikerjakan oleh *training* lain. Selama pembuatan laporan proposal *training* membuat prototipe yang menampilkan tampilan depan situs web yang bertujuan untuk memberikan gambaran pada saat pertemuan dengan dinas koperasi dan klien.

Pembuatan laporan proposal diakhiri setelah diberikan revisi oleh Direktur LPPM, dan *training* melanjutkan pembuatan prototipe dan mengrevisi tampilan yang sudah ada. Setelah itu *training* melakukan pembuatan ERD, *useCase*, dan *Activity Diagram* serta melakukan pengembangan prototipe. *Training* dan tim melakukan pertemuan ke dua dengan klien untuk membahas tentang sistem koperasi lebih dalam.

Pertemuan ke dua ini menghasilkan *output* yang tidak terduga oleh *training* dan rekannya, karena sistem yang digunakan oleh klien berbeda dengan rencana awal, maka dari itu *training* dan tim melakukan bimbingan dengan Direktur LPPM untuk memberikan informasi yang didapatkan dari pertemuan ke dua dengan klien dan menanyakan pendapat dari Direktur LPPM.

Perbedaan yang dialami terletak pada sistem simpan dan pinjam, dimana sistem klien masih terhubung dengan pihak dinas koperasi. Setiap kegiatan simpan dan pinjaman klien terhubung dengan gaji/upah kerja yang diberikan oleh Dinas Koperasi. Rencana awal *training* dan tim membuat fitur *real time transaction* yang

terdapat pada situs web.

Dalam rangka perubahan ini maka terjadilah perubahan tampilan dan perubahan fitur yang ada, yaitu fitur transaksi. *Training* dan tim hanya membuat sistem pencatatan jika terjadinya kegiatan simpan dan pinjam, dari pencatatan tersebut akan dikonversi menjadi laporan keuangan untuk Dinas Koperasi.

Training mendapatkan module pekerjaan pembuatan *company profile* dan autentikasi. *Training* memulai bagian pekerjaan yang membuat tampilan *company profile* berupa halaman Beranda, Tentang Kami, Produk, dan Hubungi Kami. Tampilan ini dibuat menggunakan bahasa pemrogramman *ReactJS* dan *TailwindCSS*. Setelah pembuatan tampilan tersebut *training* membuat fitur *login* dan *register* dengan sisi *frontend* menggunakan *ReactJS* dan *TailwindCSS*, lalu pada sisi *backend* menggunakan *nodeJS*.

Pelaksanaan kerja magang diuraikan seperti pada Tabel ??.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Belajar tentang Permenkop-kukm-no-8-tahun-2023 dan belajar ReactJS dan NodeJS.
2	Membuat Proposal pembuatan situs web koperasi
3	Membuat Proposal pembuatan situs web koperasi dan membuat prototipe
4	Revisi prototipe dan membuat ERD, <i>useCase</i> dan <i>Activity Diagram</i>
5	Revisi ERD, <i>useCase</i> , dan <i>activity Diagram</i> , dan mengembangkan prototipe
6	Memulai pembuatan <i>template</i> menggunakan <i>ReactJS</i> dan <i>TailwindCSS</i> serta belajar <i>NodeJS</i>
7	<i>Meeting</i> ke 2 dengan pihak klien untuk membahas sistem koperasi lebih dalam dan revisi prototipe
8	<i>Meeting</i> dengan <i>supervisor</i> untuk validasi dan mulai membuat UI/UX serta mulai membuat komponen UI.
9	Membuat <i>Company Profile</i> seperti halaman beranda, produk, hubungi kami, tentang kami, dan produk.

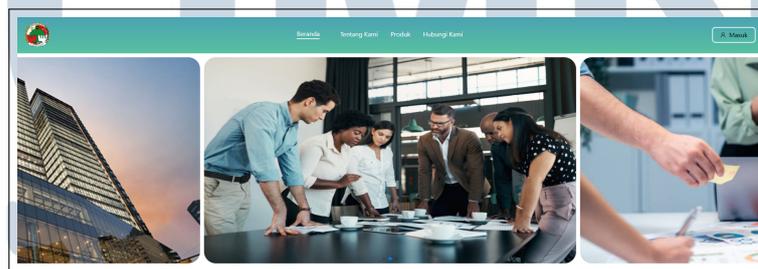
10	Membuat tampilan depan <i>Login</i> dan <i>Register</i> , serta belajar membuat <i>backend</i>
11	Mulai membuat <i>backend</i> untuk fitur <i>Login</i> dan <i>Register</i> .
12	Memperbaiki bug dari <i>backend</i> pada fitur <i>Login</i> dan <i>Register</i> serta implementasikan ke <i>frontend</i> .
13	Membuat proteksi halaman pada setiap role.
14	Memperbaiki bug pada proteksi halaman.
15	Memperbaiki bug pada <i>register</i> dan <i>login</i> serta membuat tampilan halaman menjadi responsive.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan magang di LPPM dilakukan selama 6 bulan (26 minggu), akan tetapi dalam laporan ini akan dibuat menjadi 15 minggu untuk keperluan sidang. Pelaksanaan kerja magang diuraikan seperti pada tabel diatas, berikut adalah timeline pelaksanaan magang di LPPM.

3.3.1 Fase Pertama

Pertama *training* memulai pembuatan tampilan dengan menggunakan *ReactJS* dan *TailwindCSS*, pembuatan dimulai dengan komponen *Header* dan *Footer*. *Header* dapat menampung logo, navigasibar, dan menu *login* seperti dibawah ini.



Gambar 3.1. Tampilan *header*

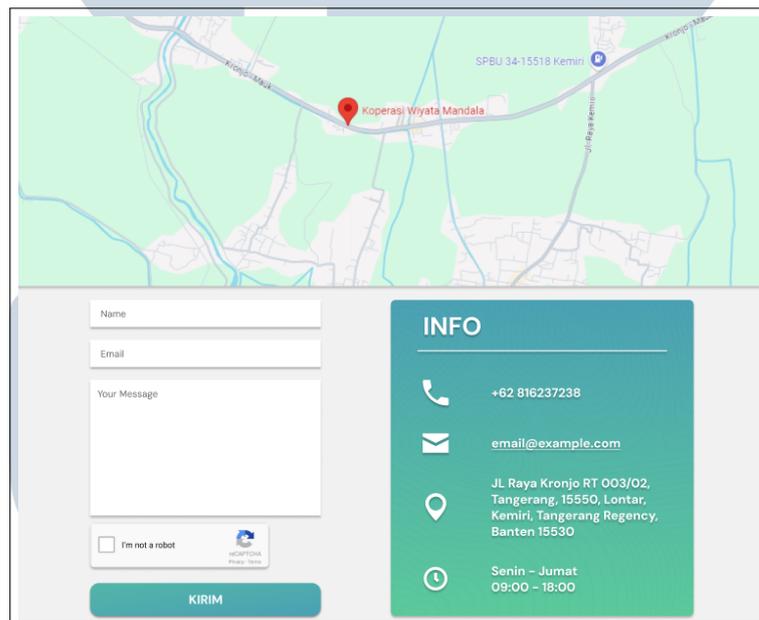
Training memberikan tampilan menjadi Konsisten (*consistency*) dengan memberikan tata letak *header* berada diatas dan *footer* berada di bawah, lalu adanya kombinasi warna dari logo Koperasi Indonesia yang berpaduan dengann warna hijau, putih dan merah. Hal tersebut memberikan hasil pada warna *background* yang berpaduan antara hijau dan putih.



Gambar 3.2. Tampilan *footer*

Tampilan *footer* ini berperan sebagai menampilkan informasi tambahan, navigasi, atau elemen penutup. Pada gambar diatas terdapat informasi yang sudah ada pada *header* seperti Tentang Kami, Produk, dan Hubungi Kami. Elemen ini membantu meningkatkan pengalaman pengguna dengan menyediakan detail yang mungkin tidak relevan untuk bagian utama, tetapi tetap penting karena *footer* menandakan halaman paling bawah pada sebuah situs web.

Terdapat tampilan Hubungi Kami yang bertujuan untuk terjadinya interaksi antar pengguna. Fitur ini dibuat agar membantu pengurus untuk mendapatkan *feedback* atau pesan dari pengguna. *Training* membuat fitur ini menggunakan *framework ReactJS* yang bernama *EmailJS*.

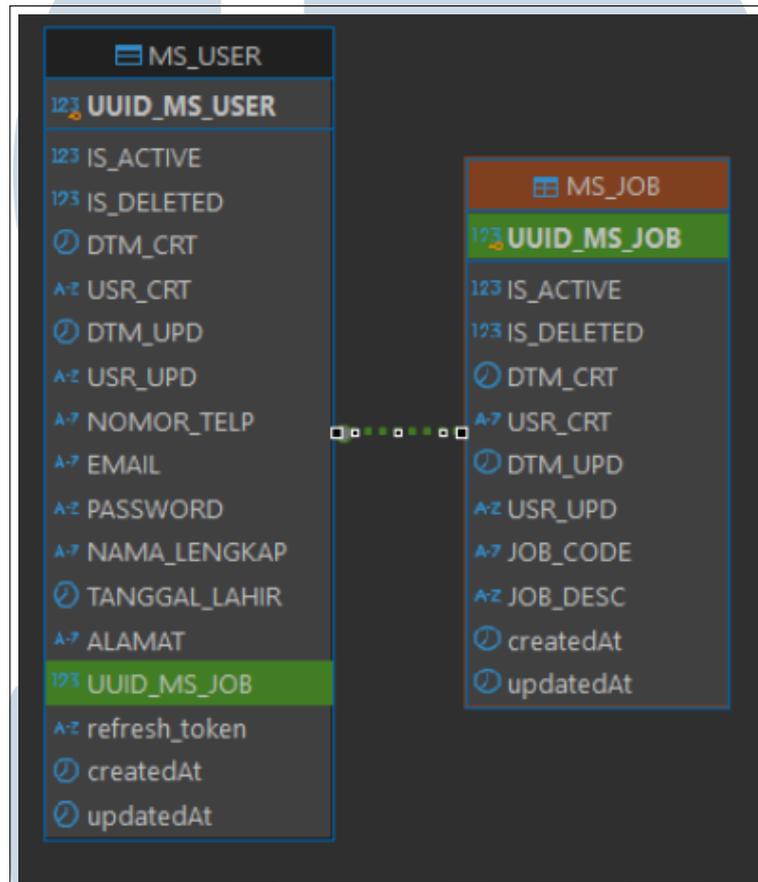


Gambar 3.3. Tampilan halaman Hubungi Kami

Pada tampilan diatas *training* memberikan informasi berupa lokasi Koperasi Wiyata Mandala, dan pada bagian bawahnya terdapat fitur kirim pesan. Fitur ini memberikan kemudahan kepada pengurus dikarenakan pesan yang diinput oleh pengguna akan dikirim langsung secara *realtime* ke *email* pengurus.

3.3.2 Fase Kedua

Pada fase Kedua ini *training* membuat fitur Autentikasi pada situs web Koperasi Wiyata Mandala. Awal mula pembuatan dimulai pada bagian *backend* dimana *training* mulai membuat model MS-USER dan MS-JOB untuk tabel di database, seperti gambar dibawah ini



Gambar 3.4. Tampilan model MS-USER dan MS-JOB yang dilihat dari DBEAVER

Pada gambar diatas terdapat tabel MS-USER yang bertujuan untuk menyimpan data *user* yang mengakses situs web tersebut. MS-JOB diberikan peran untuk menyimpan *role* pada situs web ini karena terdapat 3 *role* yaitu anggota, pengurus, dan admin. Pada anggota UUID-MS-JOB akan bernilai 1, pengurus akan bernilai 2 dan admin akan bernilai 3. Model tersebut terjadi *foreign key* pada UUID-MS-JOB pada kedua model tersebut yang bertujuan agar memudahkan pengembangan website berikutnya jika terjadi penambahan *role* dan nilainya bersifat *auto increment*.

Setelah membuat model *training* membuat *controller* untuk *register* dan

login. Pada pembuatan *controller* kolom yang ada di model MS-USER diatur agar data yang akan diinput oleh pengguna bisa masuk kedalam model MS-USER. Setelah diatur maka *training* melakukan pengujian menggunakan *REST Client* seperti gambar dibawah ini.

```
###
Send Request
POST http://localhost:5000/users
Content-Type: application/json

{
  "email": "Nikolas@gmail.com",
  "password": "12345",
  "confPassword": "12345",
  "name": "Niko",
  "role": "PENGURUS",
  "noTelp": "08123456789",
  "alamat": "Jl. Kebon Jeruk No. 1"
}
```

Gambar 3.5. Tampilan pengujian *Register* pada *REST Client*

Tampilan diatas menunjukkan skenario dimana pengguna akan melakukan *register* dimana pengguna akan menginput nama, surat elektronik (*Email*), kata sandi, *role*, nomor telepon dan alamat. Jika *training* melakukan *Send Request* maka data yang sudah diinput akan masuk ke database dan jika berhasil akan memunculkan hasil seperti dibawah ini.

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: http://localhost:5173
Vary: Origin
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: 26
ETag: W/"1a-ODfpLEMledk64/E5EuRtgf+hdiY"
Date: Wed, 18 Dec 2024 03:01:15 GMT
Connection: close

{
  "msg": "Register Success"
}
```

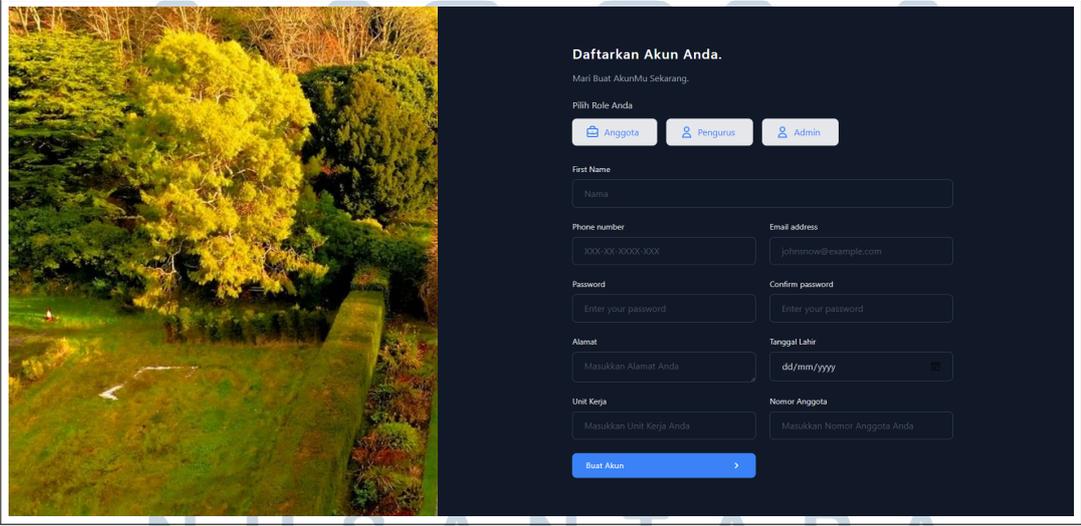
Gambar 3.6. Tampilan pengujian *Register* pada *REST Client*

Dalam pembuatan fitur registrasi ini, akses token dirancang dengan tujuan utama melindungi informasi sensitif pengguna, terutama kata sandi. Setelah proses login berhasil, kata sandi pengguna tidak perlu lagi dikirimkan dalam setiap permintaan ke server. Sebagai gantinya, akses token digunakan sebagai identitas digital yang membuktikan bahwa pengguna telah berhasil melakukan autentikasi.

Akses token ini juga memiliki peran penting dalam meningkatkan keamanan aplikasi. Resiko penyalahgunaan akses dapat diminimalkan jika token tersebut bocor. Selain itu, akses token memungkinkan server untuk membatasi akses ke sumber daya tertentu sesuai dengan hak dan peran pengguna yang telah terdaftar. Dengan demikian, penggunaan akses token tidak hanya memberikan perlindungan tambahan terhadap data sensitif pengguna, tetapi juga memastikan pengalaman pengguna yang lebih aman dan efisien dalam mengakses fitur-fitur aplikasi.

Setelah tahap registrasi selesai maka *training* membuat *controller* untuk *login* agar terbentuknya wadah bagi pengguna saat melakukan aktivitas *login*. Pada tahap ini *training* juga melakukan pengujian menggunakan *REST Client* seperti yang dilakukan pada saat registrasi.

Tahap selanjutnya *training* memulai melakukan pengembangan sisi *frontend*, yang akan menjadi antarmuka bagi pengguna untuk berinteraksi dengan sistem secara langsung. Pada tahap ini *training* membuat tampilan halaman registrasi seperti gambar dibawah ini.



The image shows a registration form on a dark background. On the left, there is a vertical image of a green landscape with trees. The form is titled "Daftarkan Akun Anda." and includes the following fields and options:

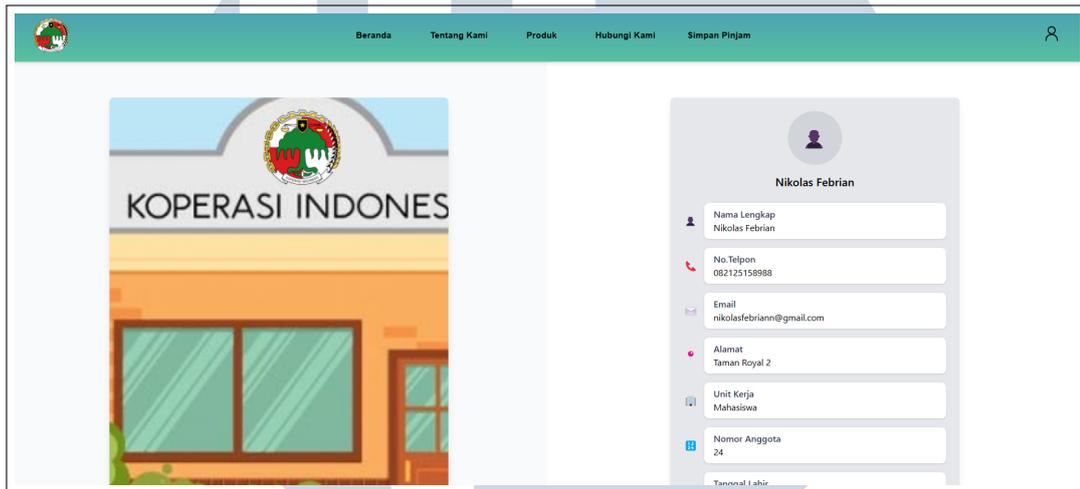
- Role selection: "Pilih Role Anda" with buttons for "Anggota", "Pengurus", and "Admin".
- First Name: "First Name" with a text input field labeled "Nama".
- Phone number: "Phone number" with a text input field labeled "XXX-XX-XXXX-XXXX".
- Email address: "Email address" with a text input field labeled "johnsnow@example.com".
- Password: "Password" with a text input field labeled "Enter your password".
- Confirm password: "Confirm password" with a text input field labeled "Enter your password".
- Alamat: "Alamat" with a text input field labeled "Masukkan Alamat Anda".
- Tanggal Lahir: "Tanggal Lahir" with a date picker labeled "dd/mm/yyyy".
- Unit Kerja: "Unit Kerja" with a text input field labeled "Masukkan Unit Kerja Anda".
- Nomor Anggota: "Nomor Anggota" with a text input field labeled "Masukkan Nomor Anggota Anda".
- A blue "Buat Akun" button with a right-pointing arrow at the bottom.

Gambar 3.7. Halaman Registrasi

Halaman tersebut jika pengguna masuk sebagai anggota atau pengurus atau admin maka pengguna harus menunggu *approve* dari pengurus terlebih dahulu, tujuan hal ini untuk mencegah orang luar untuk membuat akun karena situs web

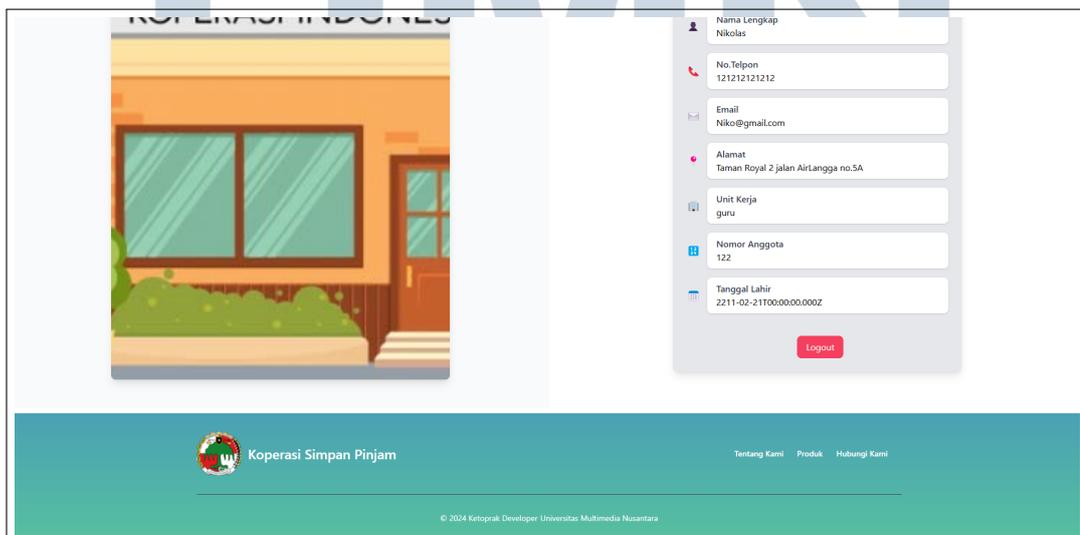
ini masih diperuntukan anggota dan pengurus Koperasi wiyata mandala.

Setelah pengguna sudah disetujui oleh pengurus maka akun yang didaftar oleh pengguna dapat melakukan *login* untuk mendapatkan akses simpan pinjam pada situs web tersebut. Pengguna dapat melihat dapat dirinya pada bagian *profile* dengan tampilan dibawah ini.



Gambar 3.8. Halaman Profil

Pada Halaman ini *entry point* yang digunakan sudah menggunakan akses token, yang dapat menampilkan data user dengan format *hash*. Setelah itu jika pengguna ingin keluar maka ada tombol logout agar sesi pengguna diakhiri sehingga tidak ada orang lain yang dapat mengakses aplikasi dari perangkat atau browser yang sama tanpa login ulang dengan tampilan dibawah ini.

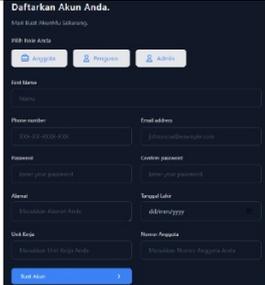
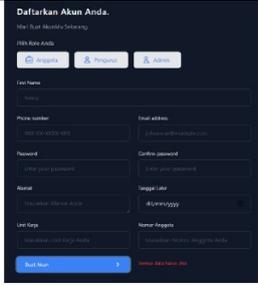
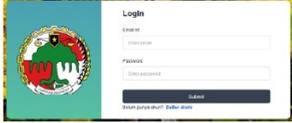


Gambar 3.9. Halaman Profil

3.4 Blackbox testing

Black box testing adalah pengujian kualitas perangkat lunak yang berfokus ada fungsionalitas perangkat lunak yang bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi [7].

Pengujian tersebut *training* lakukan dengan rekan tim dan *supervisor* pada tempat klien yang bertujuan untuk memvalidasi situs web yang sudah dirancang. Perwakilan tim yaitu *training* melakukan mengujian dengan menampilkan alur situs web yang sudah direncanakan dan mendokumentasikan berupa video.

No	Skenario Pengujian	Test Case	Hasil yang DiHarapkan	Hasil Pengujian	Kesimpulan
1	Pengguna memasukkan data dengan tidak lengkap pada halaman register		Sistem tidak akan menyimpan ketika ada yang tidak terisi, melainkan sistem akan menampilkan semua data harus diisi		Valid
2	Pengguna tidak akan bisa login jika akunnya belum terverifikasi		Sistem akan menolak karena pada database tidak ada data akun tersebut		Valid
3	Jika pengguna masuk dengan akun anggota maka hanya bisa akses fitur anggota saja		Sistem akan membaca penandaan role pengguna saat register dan login yang membuat pengguna hanya bisa akses halaman tertentu.		Valid

Gambar 3.10: Tabel *BlackBox Testing*

4	Jika pengguna masuk dengan akun pengurus maka pengguna bisa mengakses fitur pengurus		Sistem akan membaca penandaan role pengguna saat register dan login yang membuat pengguna hanya bisa akses halaman tertentu.		Valid
---	--	---	--	--	-------

Gambar 3.11: Tabel *BlackBox Testing*

3.5 Kendala dan Solusi yang Ditemukan

Kendala:

- Kurangnya pengetahuan tentang fitur *reactJS* pada bagian penempatan halaman.
- Keinginan klien yang berubah saat proses *develop*.

Solusi :

- Meluangkan waktu lebih banyak untuk mempelajari fitur-fitur yang tersedia pada *reactJS*.
- Melakukan pertemuan secara rutin dengan klien untuk mengvalidasi kebutuhan yang sudah sesuai.

