BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kedudukan atau posisi yang diberikan selama magang di PT Mitra Jasa Sarana Teknologi adalah sebagai IT *Intern* di bagian *web development*. Tugas utamanya adalah untuk membangun aplikasi sistem faktur berbasis web. Bapak Hadi Suryadi selaku *IT Consultant*, bertindak sebagai *supervisor*. Komunikasi dengan *supervisor* dilakukan secara komunikasi jarak jauh melalui pertemuan virtual yang diadakan secara berkala pada setiap hari jumat untuk mendiskusikan tugas dan memantau perkembangan proyek.

3.2 Tugas yang Dilakukan

Pada pelaksanaan magang di PT Mitra Jasa Sarana, diberikan proyek yaitu membangun aplikasi sistem faktur berbasis web, yang akan digunakan oleh internal perusahaan bertujuan untuk membantu bagian keuangan dalam mengelola dan mendata *invoice* secara efisien. Beberapa tugas yang dilakukan selama magang ini adalah sebagai berikut:

1. Membuat halaman input *client* dan *client list* untuk memasukkan dan melihat data *client*.
2. Membuat halaman *create invoice* dan *invoice list* untuk memasukkan dan melihat data *invoice*.
3. Membuat fitur konversi data *invoice* ke dalam bentuk PDF.
4. Membuat halaman *setting* untuk mengedit data dari perusahaan.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang pada PT Mitra Jasa Sarana Teknologi diuraikan seperti pada Tabel 3.1.

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

Minggu Ke-	Pekerjaan yang dilakukan
1	Penjelasan proyek, pengumpulan *user requirement*, dan *training*
2	Membuat *flowchart* dan merancang relasi *database*
3	Membuat dan melakukan perbaikan *client page* dan *invoice page*
4	Membuat dan melakukan perbaikan *log in logic* dan *generate* PDF pada *invoice list page*
5	Membuat *settings page* dan *create new user page*
6	Menerapkan *session token* setiap kali *log in*
7	Melakukan perbaikan pada bagian *settings page* dan *generate* PDF
8	Membuat fitur *edit* dan *delete* pada *client list page*, *user list page*, dan *setting page*

3.3.1 User Requirement

Dalam pengembangan aplikasi sistem faktur, terdapat beberapa *requirement* yang perlu diperhatikan.

1. Halaman input *client* dan *client list* untuk memasukkan dan melihat data *client*.
2. Halaman *create invoice* dan *invoice list* untuk memasukkan dan melihat data *invoice*.
3. Fitur konversi data *invoice* ke dalam bentuk PDF.
4. Halaman *setting* untuk mengedit data dari perusahaan.

3.3.2 Perancangan Sistem

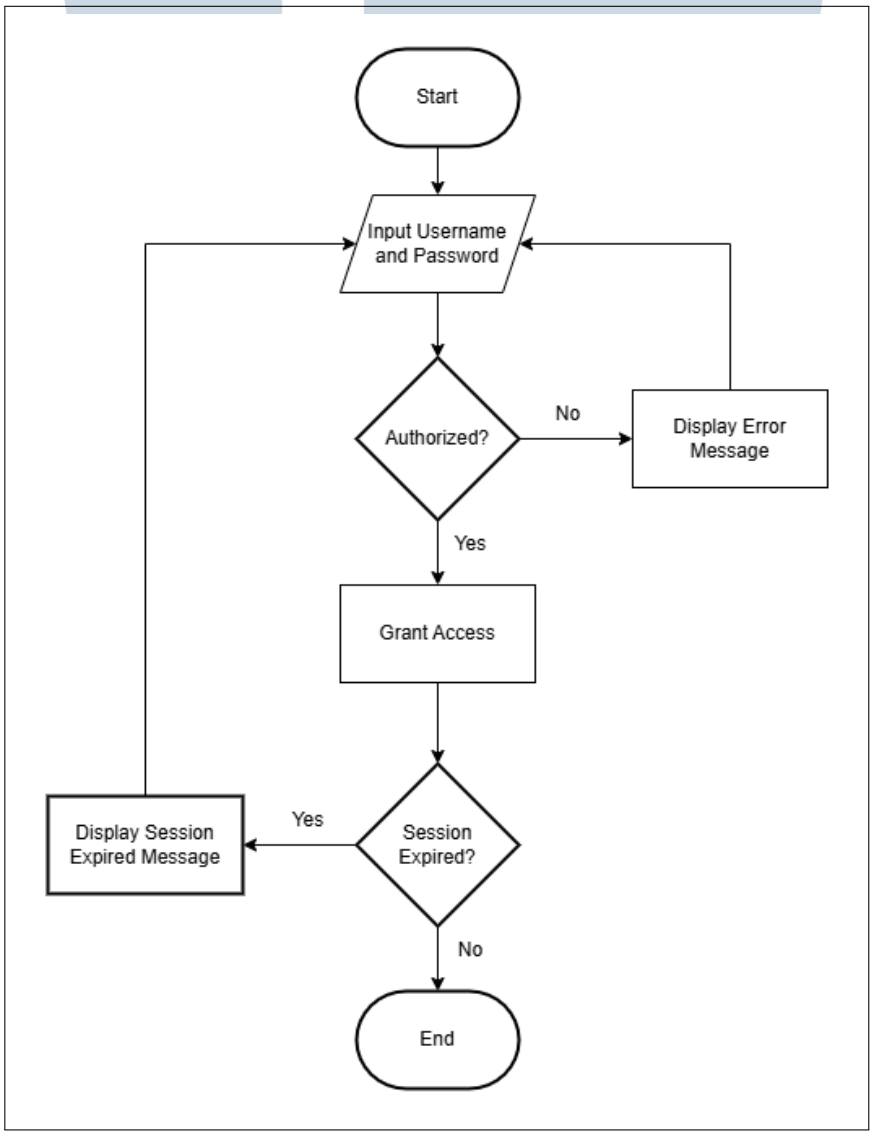
Perancangan sistem dalam aplikasi sistem faktur bertujuan untuk memastikan alur kerja dan fungsi-fungsi yang ada dapat berjalan dengan baik sesuai kebutuhan pengguna. Proses ini melibatkan pembuatan *flowchart*, *use case diagram* dan desain *database relation* yang mendukung pengelolaan data secara efisien, termasuk data *client*, *invoice*, dan perusahaan.

A. Flowchart

Flowchart digunakan untuk menggambarkan alur proses dari setiap fitur yang ada pada aplikasi, mulai dari *log in*, pengelolaan data *client*, hingga pembuatan *invoice*. Diagram ini membantu dalam memvisualisasikan langkah-langkah yang akan dijalankan oleh sistem, sehingga dapat memahami kebutuhan dan memastikan setiap fungsi bekerja sesuai dengan rencana.

A.1 Log in

Flowchart log in digambarkan seperti pada Gambar 3.1.



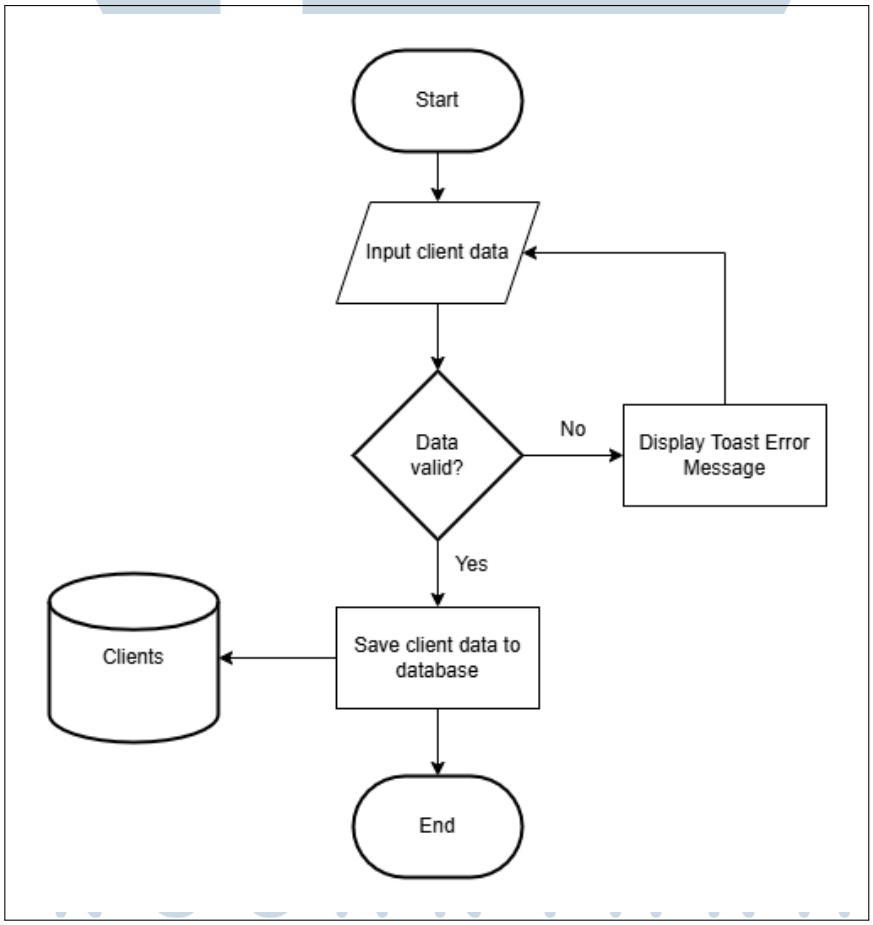
Gambar 3.1. *Flowchart log in*

Ketika *user* membuka aplikasi, halaman pertama yang akan ditampilkan adalah *log in page*. Pada halaman ini, *user* diminta untuk memasukkan *username* dan *password* untuk masuk ke dalam aplikasi. Setelah *user* memasukkan *username* dan *password*, data yang diinput akan diperiksa apakah *authorized* atau tidak.

Jika tidak *authorized*, aplikasi akan menampilkan *error message*, dan user diminta untuk memasukkan *username* dan *password* kembali. Namun, jika *authorized*, *user* akan diarahkan ke halaman *dashboard*. Setelah itu, akan diperiksa apakah *session* sudah *expired* atau belum. Jika sudah, *user* akan kembali ke halaman *log in*. Namun, jika belum *user* tidak akan kembali ke halaman *log in*.

A.2 Input Client

Flowchart input *client* digambarkan seperti pada Gambar 3.2.



Gambar 3.2. *Flowchart* input *client*

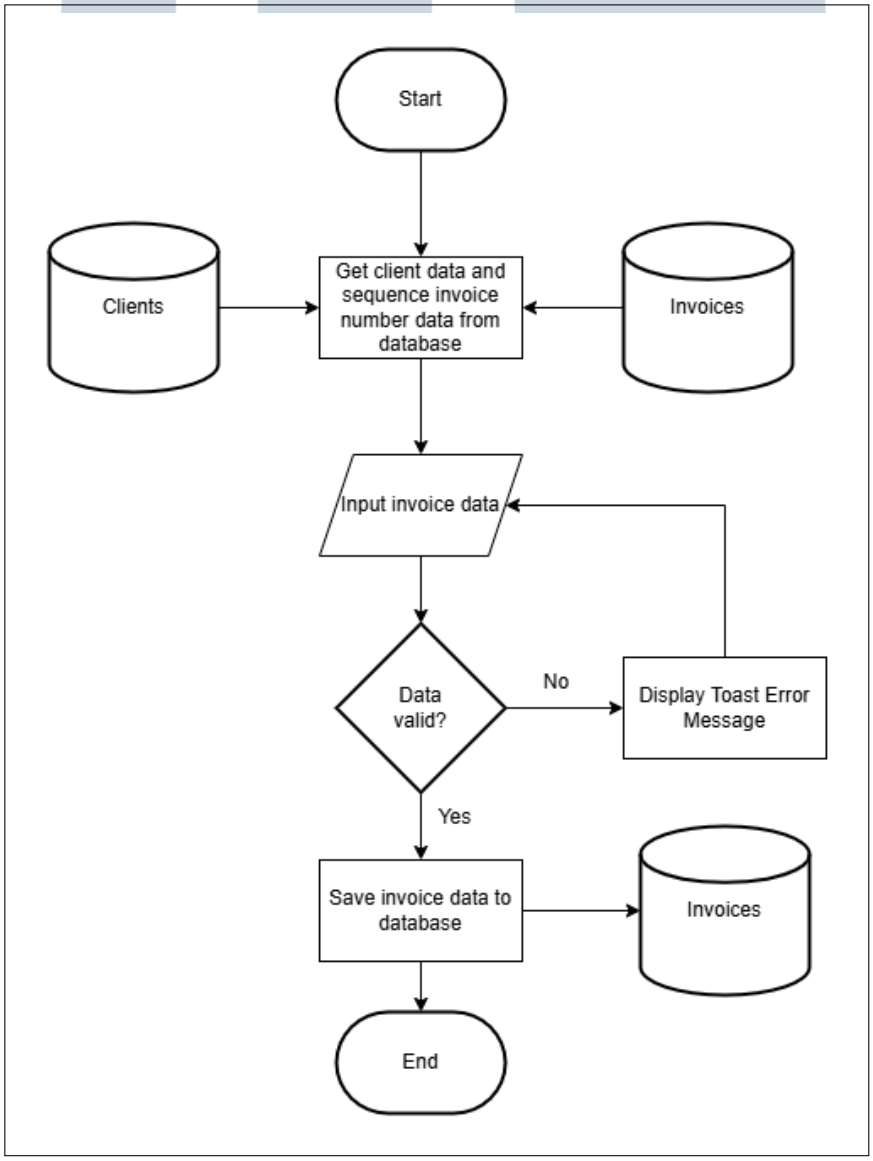
Pada halaman input *client*, *user* diminta untuk memasukkan data *client*, meliputi nama *client*, alamat *client*, nomor telepon *client*, *PIC client*, *title PIC*

client, dan *business sector client*. Setelah data tersebut dimasukkan, sistem akan memeriksa apakah data yang diinput valid atau tidak.

Jika data tidak valid, sistem akan menampilkan *error message*, dan *user* diminta untuk memperbaiki atau memasukkan kembali data *client*. Namun, jika data sudah valid, data tersebut akan disimpan ke dalam *database client*.

A.3 Create Invoice

Flowchart create invoice digambarkan seperti pada Gambar 3.3.



Gambar 3.3. *Flowchart create invoice*

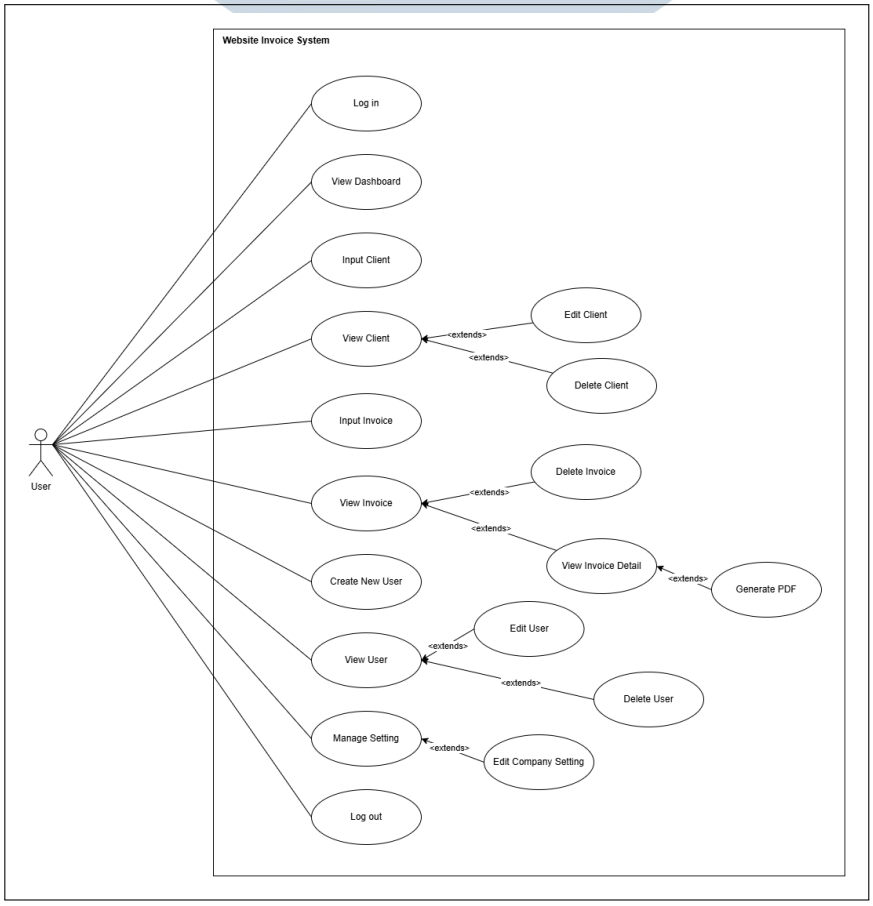
Pada halaman *create invoice*, sistem akan mengambil data *client* dan

nomor urut *invoice* per hari untuk ditampilkan pada formulir pembuatan *invoice*. Setelah itu, *user* akan diminta untuk memasukkan *contract number*, deskripsi produk, *quantity* produk, harga produk serta memilih apakah produk tersebut akan dikenakan *Value-added Tax* (VAT) atau tidak. Setelah data tersebut dimasukkan, sistem akan memeriksa apakah data yang diinput valid atau tidak.

Jika data tidak valid, sistem akan menampilkan *error message*, dan *user* akan diminta untuk memperbaiki atau memasukkan kembali data yang diinput. Namun, jika data sudah valid data akan disimpan ke dalam *database invoice*.

B. Use Case Diagram

Use case diagram digunakan untuk memodelkan interaksi antara *user* dengan sistem faktur. Diagram ini menggambarkan semua fungsi utama yang dapat diakses oleh *user*, seperti autentikasi (*Log in*), pengelolaan data *client*, pembuatan dan pengelolaan *invoice*, manajemen *user*, serta pengaturan perusahaan. *use case diagram website* sistem faktur digambarkan seperti pada Gambar 3.4.



Gambar 3.4. *Use case diagram website* sistem faktur

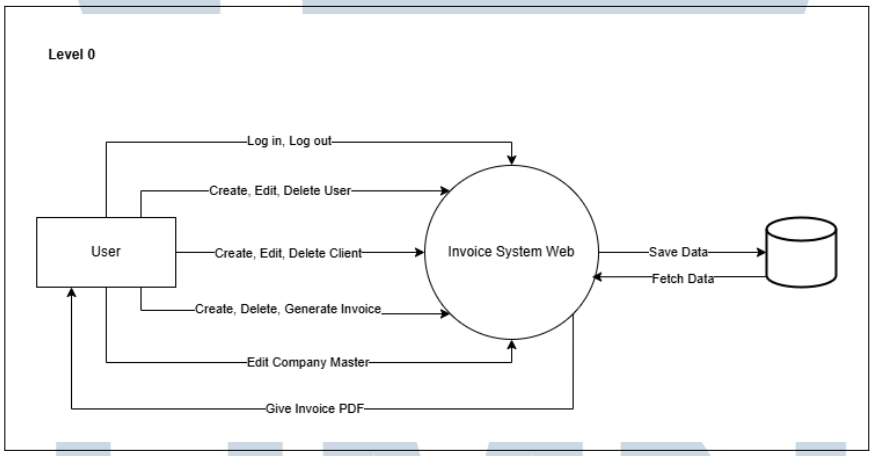
Use case diagram ini terdiri dari beberapa *use case* utama yang dapat dilakukan oleh *user*, yaitu:

1. *Log in* – Proses autentikasi pengguna untuk mengakses sistem.
2. *View Dashboard* – Menampilkan informasi ringkasan yang relevan terkait *invoice*.
3. *Input Client* – Memasukkan data *client* baru ke dalam sistem.
4. *View Client* – Menampilkan daftar *client* yang tersimpan dalam sistem.
    - *Edit Client* (*extends*) – Mengedit informasi *client* tertentu yang sudah ada.
    - *Delete Client* (*extends*) – Menghapus data *client* tertentu yang sudah ada dari sistem.
5. *Create Invoice* – Membuat data *invoice* baru dalam sistem.
6. *View Invoice* – Menampilkan daftar *invoice* yang telah dibuat.
    - *Delete Invoice* (*extends*) – Menghapus *invoice* tertentu.
    - *View Invoice Detail* (extends) – Menampilkan detail dari sebuah *invoice*.
        - *Generate* PDF (*extends*) – Menghasilkan *file* PDF dari detail *invoice*.
7. *Create New User* – Membuat akun pengguna baru untuk sistem.
8. *View User* – Menampilkan daftar pengguna yang ada.
    - *Edit User* (*extends*) – Mengedit informasi pengguna tertentu yang sudah ada.
    - *Delete User* (*extends*) – Menghapus data pengguna tertentu dari sistem.
9. *Manage Setting* – Mengatur informasi perusahaan.
    - *Edit Company Setting* (*extends*) – Mengubah pengaturan terkait informasi perusahaan.
10. *Log out* – Keluar dari sistem.

Relasi *extends* pada diagram digunakan untuk menunjukkan bahwa fitur tambahan seperti *edit*, *delete*, dan *generate* PDF hanya dapat dilakukan setelah fungsi utama terkait dipanggil terlebih dahulu. Dengan adanya *use case diagram*, sistem menjadi lebih terstruktur dalam menggambarkan kebutuhan fungsional dari perspektif *user*, memastikan setiap proses telah didefinisikan dengan jelas.

C. Data Flow Diagram

Data flow diagram (DFD) *level* 0 berfungsi untuk menggambarkan arus data utama dalam web sistem faktur. Diagram ini menunjukkan bagaimana *user* berinteraksi dengan sistem, dan bagaimana sistem memproses serta menyimpan data. Berikut penjelasan dari DFD *level* 0 pada Gambar 3.5:



Gambar 3.5. *Data flow diagram website* sistem faktur *level* 0

1. *User* berinteraksi dengan web sistem faktur untuk melakukan berbagai aktivitas, seperti:
    - *Log in, log out* – *User* melakukan autentikasi masuk dan keluar dari sistem.
    - *Create, edit, delete user* – Mengelola data *user* yang dapat mengakses sistem.
    - *Create, edit, delete client* – Mengelola data klien yang akan dihubungkan dengan *invoice*.
    - *Create, delete, generate invoice* – Membuat, menghapus, dan menghasilkan PDF *invoice*.
    - *Edit company master* – Mengedit informasi terkait perusahaan.

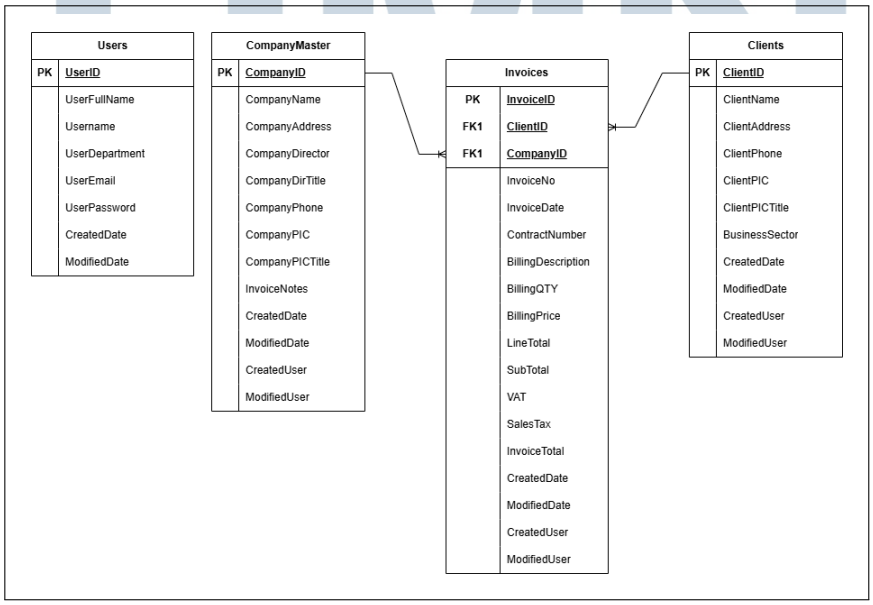
- *Give Invoice* PDF – Menghasilkan dan memberikan *file* PDF dari *invoice* kepada *user*.

2. Web sistem faktur atau *Invoice system web* berfungsi sebagai pusat pemrosesan data. Semua permintaan dari *user* diproses oleh sistem dan diteruskan ke *database* untuk penyimpanan atau pengambilan data.
3. *Database* berfungsi untuk menyimpan dan menyediakan data sesuai kebutuhan sistem. Aktivitas ini melibatkan:
    - *Save Data* – Menyimpan data seperti *user*, *client*, *invoice*, dan data perusahaan.
    - *Fetch Data* – Mengambil data yang tersimpan untuk ditampilkan kembali ke dalam sistem.

DFD *level* 0 ini menunjukkan alur data sederhana antara entitas *user*, *invoice system web*, dan *Database*. Diagram ini memudahkan pemahaman proses operasional yang terjadi di dalam web sistem faktur.

D. Database

Struktur *database* yang ditunjukkan dalam Gambar 3.6 merupakan *relational database* yang dirancang untuk mengelola informasi terkait *user*, *invoice*, *client*, dan perusahaan.



Gambar 3.6. Skema *database* sistem faktur

Pada tabel *user* digunakan untuk menyimpan informasi terkait pengguna sistem, seperti ID *user*, nama lengkap, (*username*), departemen, alamat email, dan *password*. Tabel ini juga menyimpan data terkait waktu pembuatan dan pembaruan informasi *user*. Data *user* digunakan untuk keperluan otentikasi dan otorisasi untuk mengakses sistem dan menjalankan fungsionalitas sesuai dengan peran dan izin yang diberikan.

Selanjutnya pada tabel *company*, digunakan untuk menyimpan informasi terkait perusahaan, seperti ID perusahaan, nama perusahaan, alamat, nama direktur, nomor telepon, email, *company PIC*, *company PIC title*, dan *invoice notes*. Data ini berfungsi untuk memberikan konteks dan identitas pada dokumen-dokumen yang berkaitan dengan perusahaan, seperti *invoice* dan *setting page*.

Berikutnya, pada tabel *Client*, digunakan untuk menyimpan informasi mengenai *client* perusahaan, seperti ID *client*, nama, alamat, nomor telepon, *client PIC* dan *client PIC title*. Data ini sangat penting untuk mengelola pemrosesan *invoice*, pemrosesan pesanan, serta menjaga kontak dan informasi *client* untuk keperluan komunikasi dan pengelolaan hubungan bisnis.

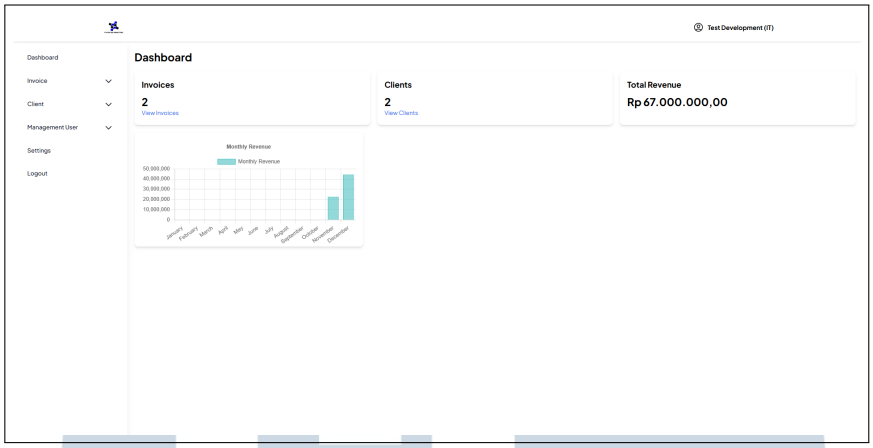
Dan yang terakhir, pada tabel *Invoice*, digunakan untuk menyimpan informasi terkait transaksi keuangan antara perusahaan dan *client*, seperti ID *invoice*, ID *client*, nomor *invoice*, tanggal *invoice*, serta rincian penagihan (deskripsi, jumlah, harga), total per item, *subtotal*, pajak penjualan, dan jumlah total *invoice*. Data ini penting untuk keperluan akuntansi, pelacakan, dan pelaporan, serta untuk memastikan bahwa semua transaksi keuangan tercatat dengan benar dan dapat dipertanggungjawabkan.

E. Implementasi

Aplikasi sistem faktur memiliki empat halaman utama, yaitu halaman *invoice*, halaman *client*, halaman *management user*, dan halaman *setting*. Berikut ini adalah penjelasan dari setiap halaman tersebut.

E.1 Modul Dashboard

Halaman dashboard adalah halaman pertama yang akan muncul ketika *user* dapat berhasil *log in* ke dalam aplikasi. Tampilan halaman *dashboard* dapat dilihat pada Gambar 3.7.

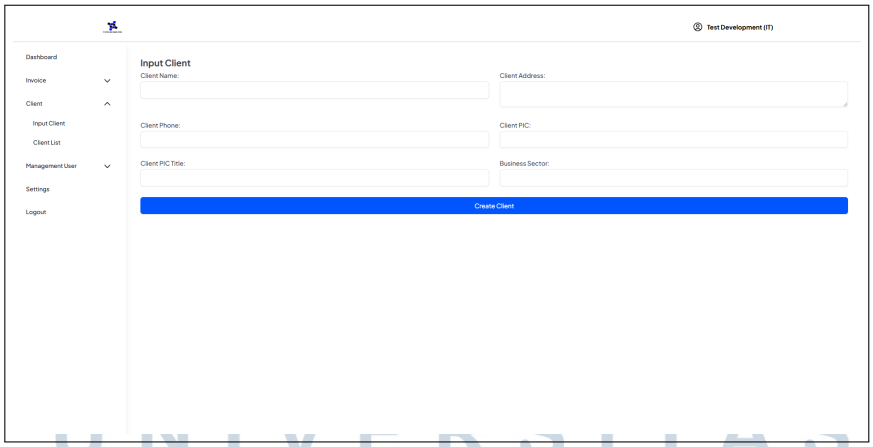


Gambar 3.7. Halaman *dashboard*

Pada halaman ini ditampilkan jumlah *invoice*, *client*, total *revenue*, serta grafik *revenue* per bulan.

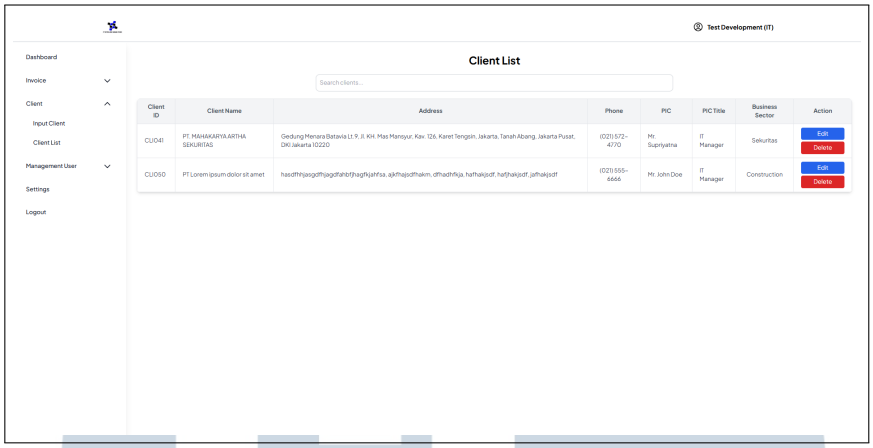
E.2 Modul Client

Pada modul *client*, terdapat dua halaman utama, yaitu halaman input *client* dan halaman *client list*. Tampilan halaman input *client* dapat dilihat pada Gambar 3.8.



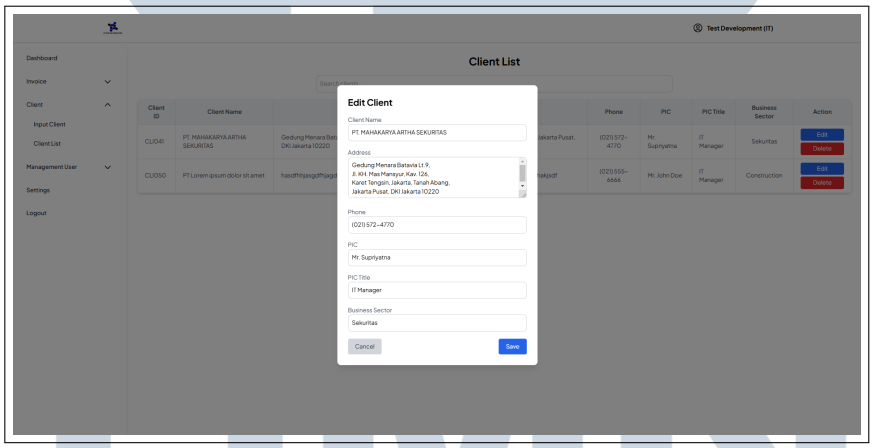
Gambar 3.8. Halaman input *client*

Pada halaman ini, *user* dapat memasukkan data *client* yang meliputi nama *client*, alamat *client*, nomor telepon *client*, *client PIC*, *client PIC title*, dan *business sector*. Data ini diperlukan jika ingin menambahkan *client* yang akan dibuatkan *invoice*. Selanjutnya, untuk tampilan *client list* dapat dilihat pada Gambar 3.9.



Gambar 3.9. Halaman *client list*

Pada halaman ini, *user* dapat mengubah data *client* serta menghapus data *client* yang sudah ada. Sementara itu, pada Gambar 3.10 menunjukkan modal untuk mengubah data *client* yang sudah ada.

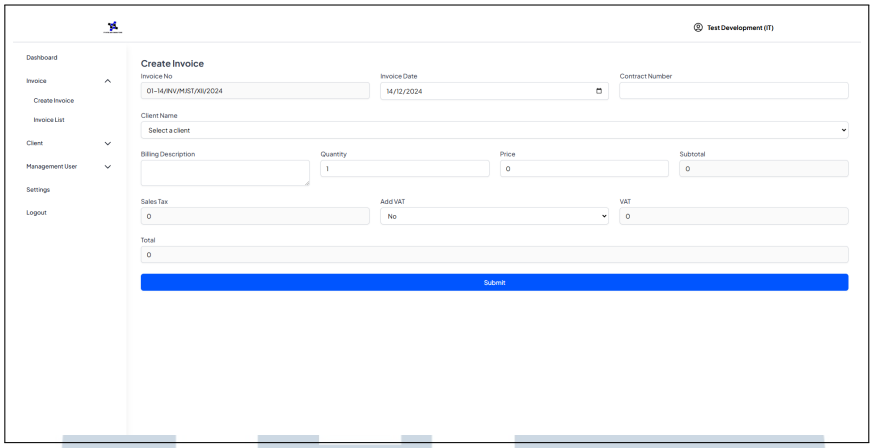


Gambar 3.10. Modal *edit client list*

Pada modal *edit client*, terdapat dua *button* yaitu, *cancel* untuk membatalkan perubahan data dan *save* untuk menyimpan data yang telah diubah.

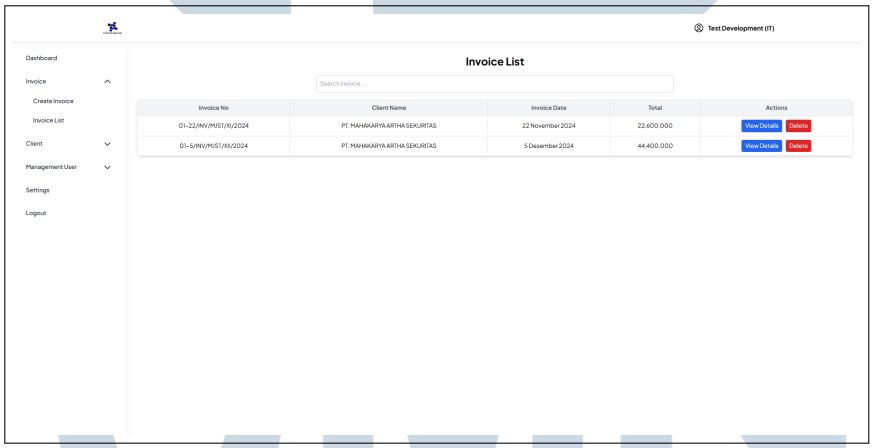
E.3 Modul Invoice

Pada modul *invoice*, terdapat dua halaman utama, yaitu halaman *create invoice* dan halaman *invoice list*. Tampilan halaman *create invoice* dapat dilihat pada Gambar 3.11.



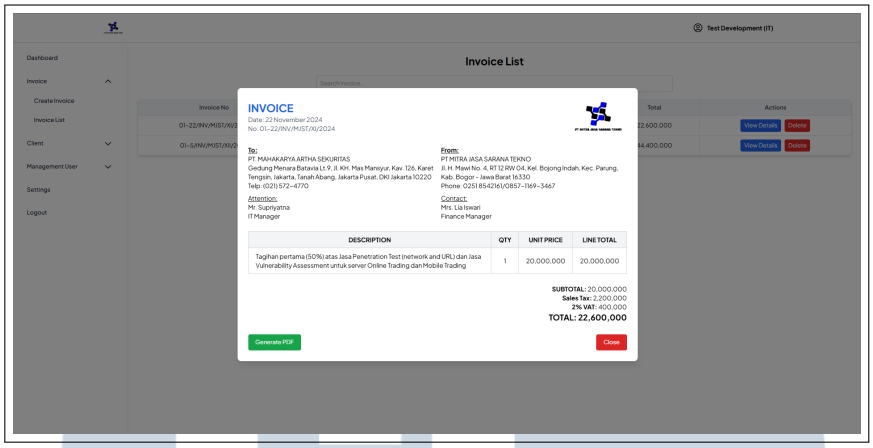
Gambar 3.11. Halaman *create invoice*

Pada halaman ini, *user* dapat memasukkan data yang diperlukan untuk membuat *invoice*, seperti tanggal *invoice*, *contract number*, nama *client*, deskripsi produk, *quantity* produk, harga produk, dan *value-added tax* (VAT). Selanjutnya, untuk tampilan halaman *invoice list* dapat dilihat pada Gambar 3.12.



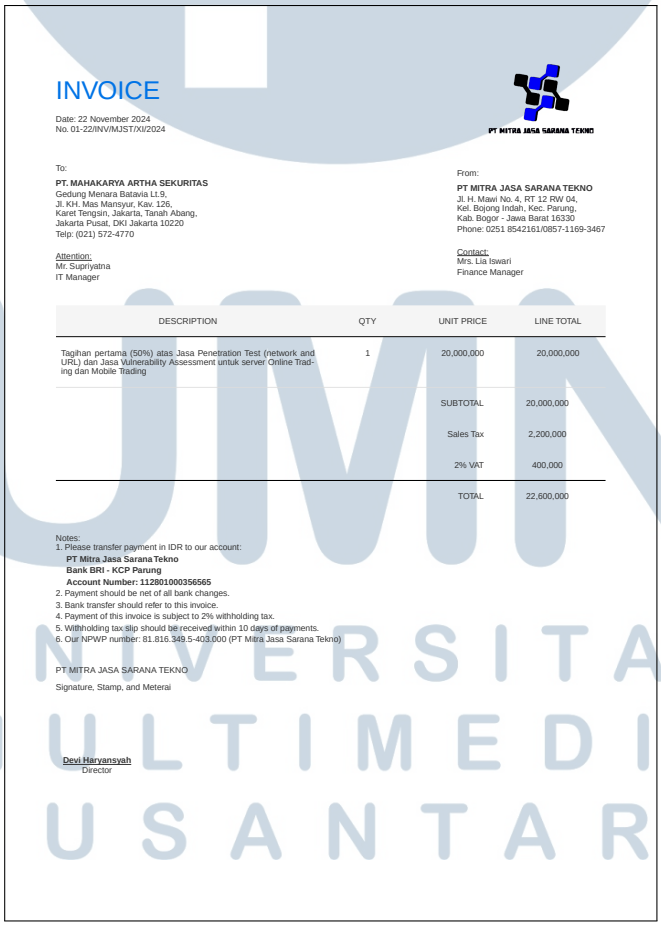
Gambar 3.12. Halaman *invoice list*

Pada halaman ini, *user* dapat menghapus dan melihat *detail* dari data *invoice* yang sudah ada. Sementara itu, modal dari *view detail* dapat dilihat pada Gambar 3.13.



Gambar 3.13. Modal *detail invoice*

Pada modal *detail invoice* ini terdapat fitur untuk mengkonversi *detail invoice* ke dalam bentuk PDF. Tampilan hasil dari konversi *detail invoice* ke dalam bentuk PDF dapat dilihat pada Gambar 3.14.



INVOICE

Date: 22 November 2024
No. 01-22/INV/MJST/XI/2024

PT MITRA JASA SARANA TEKNO

To:

PT. MAHAKARYA ARTHA SEKURITAS
Gedung Menara Batavia Lt.9,
Jl. KH. Mas Mansyur, Kav. 126,
Karet Tengsin, Jakarta, Tanah Abang,
Jakarta Pusat, DKI Jakarta 10220
Telp: (021) 572-4770

Attention:
Mr. Supriyatna
IT Manager

From:

PT MITRA JASA SARANA TEKNO
Jl. H. Mawi No. 4, RT 12 RW 04,
Kel. Bojong Indah, Kec. Parung,
Kab. Bogor - Jawa Barat 16330
Phone: 0251 8542161/0857-1169-3467

Contact:
Mrs. Lia Iswari
Finance Manager

DESCRIPTION	QTY	UNIT PRICE	LINE TOTAL
Tagihan pertama (50%) atas Jasa Penetration Test (network and URL) dan Jasa Vulnerability Assessment untuk server Online Trading dan Mobile Trading	1	20,000,000	20,000,000
		SUBTOTAL	20,000,000
		Sales Tax	2,200,000
		2% VAT	400,000
		TOTAL	22,600,000

Notes:
1. Please transfer payment in IDR to our account:
   PT Mitra Jasa Sarana Tekno
   Bank BRI - KCP Parung
   Account Number: 112801000356565
2. Payment should be net of all bank changes.
3. Bank transfer should refer to this invoice.
4. Payment of this invoice is subject to 2% withholding tax.
5. Withholding tax slip should be received within 10 days of payments.
6. Our NPWP number: 81.816.349.5-403.000 (PT Mitra Jasa Sarana Tekno)

PT MITRA JASA SARANA TEKNO

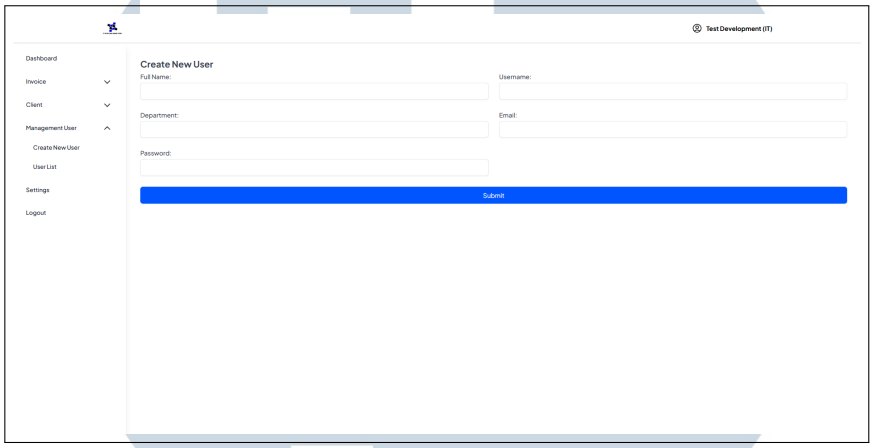
Signature, Stamp, and Meterai

Devi Haryansyah
Director

Gambar 3.14. *Invoice* PDF

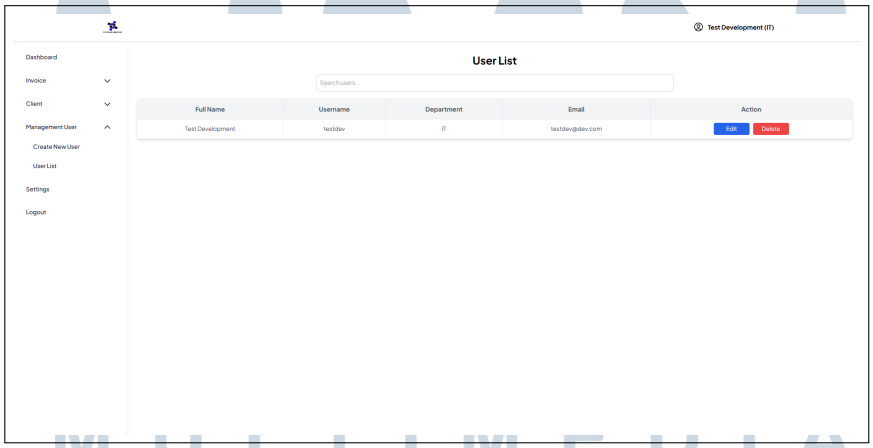
E.4 Modal Management User

Pada modul *management user*, terdapat dua halaman utama, yaitu halaman *create new user* dan halaman *user list*. Tampilan halaman *create new user* dapat dilihat pada Gambar 3.15.



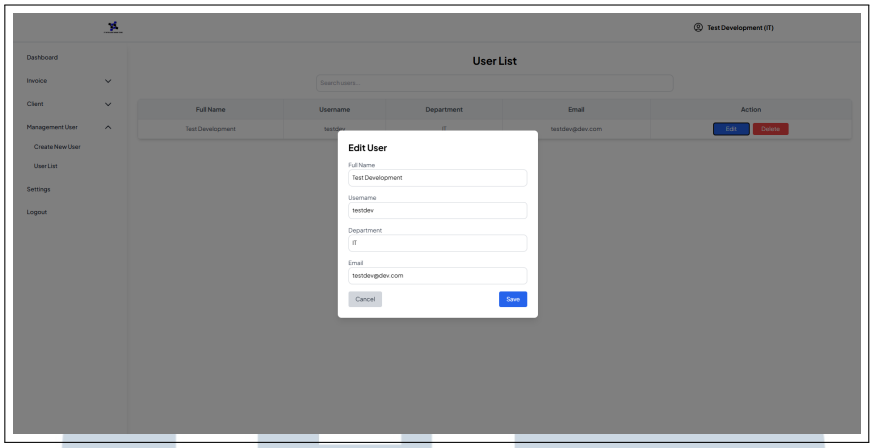
Gambar 3.15. Halaman *create new user*

Pada halaman ini, *user* dapat memasukkan data dari *user*, seperti nama panjang *user*, *username*, *department user*, *email user*, dan *password* untuk membuatkan akun kepada *user* baru. Selanjutnya, untuk halaman *user list* dapat dilihat pada Gambar 3.16.



Gambar 3.16. Halaman *user list*

Pada halaman ini *user* dapat mengubah dan menghapus data dari *user* yang sudah ada. Sementara itu, modal untuk mengubah data *user* dapat dilihat pada Gambar 3.17.

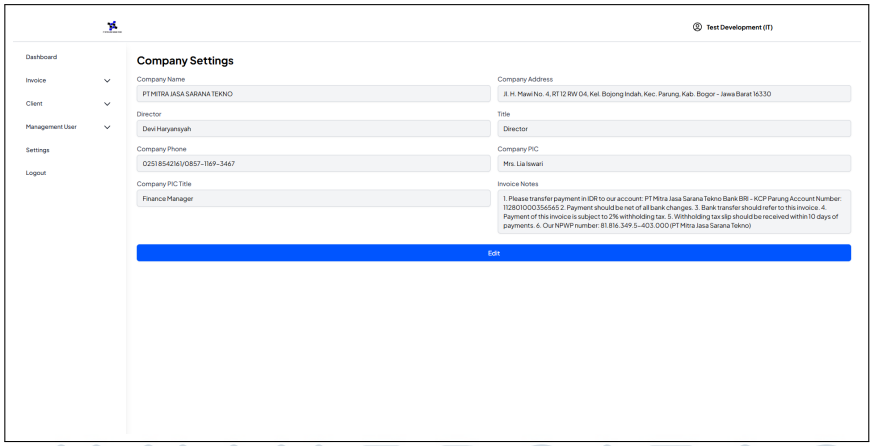


Gambar 3.17. Modal *edit user*

Pada modal *edit user*, terdapat dua *button* yaitu, *cancel* untuk membatalkan perubahan data dan *save* untuk menyimpan data yang telah diubah.

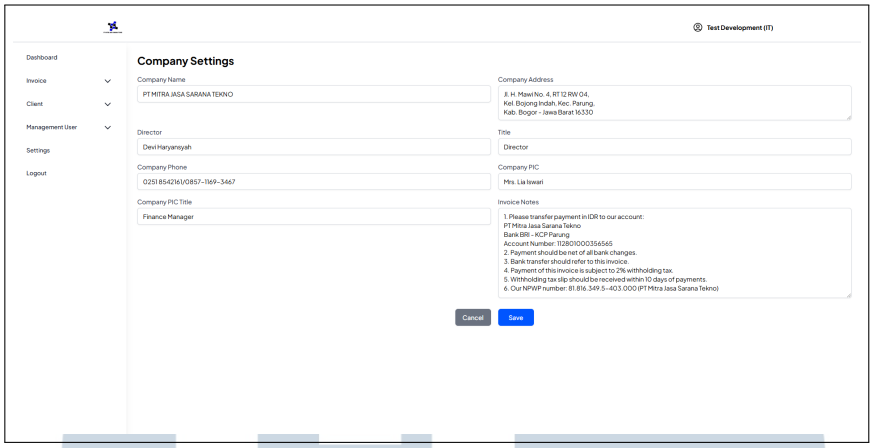
E.5 Modul Setting

Halaman *setting* merupakan halaman untuk melihat dan mengubah data dari perusahaan. Tampilan dari halaman *setting* dapat dilihat Gambar 3.18.



Gambar 3.18. Halaman *setting*

Pada halaman ini terdapat data dari perusahaan dan *edit button* untuk mengubah data dari perusahaan yang sudah ada. Tampilan dari modal *edit* data perusahaan daspat dilihat pada Gambar 3.19.



Gambar 3.19. Halaman *edit setting*

Pada halaman *edit setting*, terdapat dua *button* yaitu, *cancel* untuk membatalkan perubahan data dan *save* untuk menyimpan data yang telah diubah.

E.6 Express.js

Express merupakan *framework* yang cepat, fleksibel, minimalis, *user-friendly* dan dilengkapi dengan berbagai pilihan *middleware* yang menyederhanakan pembuatan aplikasi Node. Express menggunakan JavaScript sebagai bahasa pemrogramannya dan menawarkan metode efektif untuk *developing* aplikasi web dan API. Dengan Express, pengelolaan rute, *requests*, dan respons menjadi lebih sederhana, sehingga memudahkan untuk memfasilitasi *development* dalam pembuatan aplikasi yang kuat dan *scalable* [8].

E.7 React.js

React merupakan *Library* JavaScript yang dirancang untuk membangun antarmuka pengguna (UI) di web dengan cara deklaratif dan berbasis komponen. Pustaka ini memungkinkan pengembang untuk membuat komponen UI yang dapat dipakai ulang. React menggunakan pendekatan Virtual DOM, yang bertujuan untuk meningkatkan kinerja *rendering* dengan mengurangi pembaruan DOM. React sangat efisien dan dapat terintegrasi dengan berbagai alat serta pustaka lainnya [9].

E.8 TailwindCSS

Tailwind CSS merupakan kerangka kerja CSS berbasis utilitas yang mempermudah pengembangan web dengan memberikan berbagai kelas utilitas

yang telah disiapkan sebelumnya. Dengan kelas-kelas ini, Anda dapat membuat desain *custom* tanpa perlu menulis CSS secara manual, yang mendukung konsistensi, skalabilitas, dan efisiensi dalam pengembangan. Tailwind memfokuskan pada penggunaan kelas utilitas fungsional, memungkinkan pembuatan antarmuka yang responsif dan menarik dengan cepat dan efisien [10].

E.9 @react-pdf/renderer

Library @react-pdf/renderer digunakan untuk menghasilkan dokumen PDF dalam aplikasi React. Dengan *library* ini, *developer* dapat merancang tata letak PDF yang rumit menggunakan komponen React, sehingga memudahkan pembuatan dokumen yang dinamis dan menarik secara estetika [11].

3.4 Kendala dan Solusi yang Ditemukan

Terdapat sejumlah kendala yang ditemukan beserta solusi yang diterapkan selama pelaksanaan proyek magang pada PT Mitra Jasa Sarana Teknologi, yaitu:

3.4.1 Kendala

1. Pengumpulan *user requirement* untuk kebutuhan aplikasi tidak dijelaskan secara rinci di awal, sehingga memerlukan waktu tambahan untuk berkomunikasi dengan supervisor guna memahami kebutuhan pengguna secara keseluruhan.
2. Teknologi *library* @react-pdf/renderer belum pernah digunakan sebelumnya, sehingga memerlukan waktu untuk mempelajarinya.

3.4.2 Solusi

1. Mengadakan sesi diskusi tambahan secara virtual untuk memperoleh pemahaman yang lebih mendalam mengenai kebutuhan aplikasi.
2. Melakukan riset dan membaca dokumentasi terkait *library* @react-pdf/renderer, serta mencoba contoh implementasi sederhana untuk memahami cara penggunaannya sebelum diintegrasikan ke dalam aplikasi.