**SISTEM INFORMASI MANAJEMEN PROYEK MENGGUNAKAN FRAMEWORK LARAVEL**

**11 DI PT. ESA DATA TEKNIKA BOGOR**

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Jenjang Strata Satu (S1)

Pada Program Sistem Informasi

Tahun Akademik 2024-2025



**DISUSUN OLEH :**

**VERIA RAJA TUNGGAL**

**109210940174**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER (STMIK) PRANATA INDONESIA**

**2025**

**LEMBAR MOTTO DAN PERSEMBAHAN**

**MOTTO :**

* Belajar bukan untuk jadi yang terbaik, tapi untuk menjadi bermanfaat.
* Kegagalan adalah langkah menuju keberhasilan.
* Syukuri yang ada, perjuangkan yang belum ada.

**Kerya tulis ini saya persembahkan untuk :**

Ibu dan Ayah tercinta Tri Sugiyanti dan Mersis serta Adikku Aji Sultan Khasuro yang telah banyak berkorban dan memotivasi dalam penyelesaian skripsi ini.

**LEMBAR PERSETUJUAN PEMBIMBING**

Mahasiswa dengan identitas di bawah ini :

Nama : VERIA RAJA TUNGGAL

NPM : 109210940174

Jurusan : Sistem Informasi

Jenjang Pendidikan : Strata-satu (S1)

Tempat Penelitian : PT. Esa Data Teknika

Judul : Sistem Informasi Manajemen Proyek menggunakan

framework laravel 11 di PT. Esa Data Teknika Bogor

Dinyatakan telah SELESAI / BELUM SELESAI melaksanakan penulisan Skripsi/Tugas Akhir. Sehingga yang bersangkutan DIIZINKAN / TIDAK DIIZINKAN mengikuti sidang Ujian Skripsi/Tugas Akhir.

Bogor, 15 Agustus 2025

Menyetujui,

Pembimbing I

(Imam Zaenudin, S.Kom., M.Kom)

NUPTK: 7053744645130063

Pembimbing II

(Drs. Harry Subagja. MM. M.Pd)

NUPTK: 4736761662137132

Mengetahui,

Ketua Program Studi Sistem Informasi

(Nurhadi, S.Kom., M.Kom)

NUPTK: 745875667131093

**LEMBAR PENGESAHAN SIDANG YUDISIUM**

Skripsi / Tugas Akhir dengan judul Sistem Informasi Manajemen Proyek menggunakan framework laravel 11 di PT. Esa Data Teknika Bogor, telah diujikan dalam sidang yudisium, dihadapan Dewan Penguji pada tanggal 23 bulan Agustus tahun 2025. Dan dinyatakan : LULUS / TIDAK LULUS / LULUS BERSYARAT atau dengan Catatan :

........................................................................................................................................................................................................................................................................

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Penguji : | Tanda Tangan : | Tanggal : | Keterangan |
| 1. | NUPTK : |  |  | Ketua Penguji |
| 2. | NUPTK : |  |  | Anggota Penguji |
| 3. | NUPTK : |  |  | Anggota Penguji |

Mengetahui,

Ketua STMIK Pranata Indonesia Ketua DBPTA

(Drs Harry Subagjo, M.M.,M.Pd) (Drs Harry Subagjo, M.M.,M.Pd)

NUPTK : 7053744645130063 NUPTK : 745875667131093

**PERNYATAAN KEASLIAN SKRIPSI/TUGAS AKHIR**

Saya yang bertanda tangan dibawah ini :

Nama : VERIA RAJA TUNGGAL

NPM : 109210940174

Program Studi : Sistem Informasi

Judul Skripsi : Sistem Informasi Manajemen Proyek menggunakan

framework laravel 11 di PT. Esa Data Teknika Bogor

Menyatakan dengan sesungguhnya bahwa saya telah mengerjakan dan menyusun skripsi dengan judul “Sistem Informasi Manajemen Proyek menggunakan framework laravel 11 di PT. Esa Data Teknika Bogor” dimana dalam skripsi ini :

1. Tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi.
2. Sepanjang sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
3. Disusun dan dikerjakan sendiri tanpa menyuruh orang lain untuk mengerjakannya.

Bila dikemudian hari ternyata terbukti bahwa skripsi ini adalah plagiat ataupun bukan saya yang mengerjakannya Pranata Indonesia yakni **pencabutan ijazah yang telah saya terima dan ijazah tersebut dinyatakan tidak sah.**

Demikian pernyataan ini saya perbuat dengan sungguh-sungguh, dalam keadaan sadar dan tanpa ada tekanan dari pihak manapun.

Bogor, ..... , ................... 2025

Veria Raja Tunggal

Veria Raja Tunggal, 109210940174, Sistem Informasi

Sistem Informasi Manajemen Proyek menggunakan framework laravel 11 di PT. Esa Data Teknika Bogor, dibawah bimbingan Imam Zaenudin, S.Kom., M.Kom, dan Drs Harry Subagjo, M.M.,M.Pd

90 + xiv halaman/ 28 tabel/ 54 gambar/ 35 lampiran

**ABSTRAKSI**

Penelitian ini bertujuan untuk mengembangkan sistem informasi manajemen proyek menggunakan framework laravel 11 di PT. Esa Data Teknika Bogor, dengan fokus meningkatkan *efisiensi* dan akurasi dalam manajemen proyek.

Metodologi yang digunakan dalam penelitian ini adalah metode *framework*, yang merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi**.**

Perancangan sistem dilakukan menggunakan *Unified Modeling Language* (UML) dengan mencakup diagram *Use Case Diagram, Activity Diagram, Class Diagram* dan *Sequence Diagram* untuk mendefinisikan struktur dan alur kerja sistem.

Hasil penelitian ini adalah sebuah sistem informasi yang dapat memberikan dukungan yang baik untuk kebutuhan operasional PT. Esa Data Teknika dalam manajemen proyek. Sistem ini terintegrasi dengan baik dalam hal manajemen data dan pengambilan keputusan berdasarkan laporan yang dihasilkan.

**Kata Kunci** : Sistem Informasi, manajemen proyek, framework laravel

Veria Raja Tunggal, 109210940174, *Information Systems*

*Project Management Information System Using* Laravel 11 *Framework* *at* PT. Esa Data Teknika, Bogor, *under the supervision of* Imam Zaenudin, S.Kom., M.Kom, *and* Drs Harry Subagjo, M.M.,M.Pd

90 + xiv pages/ 28 *tables*/ 54 *images*/ 35 appendices

**ABSTRACTION**

This research aims to develop a project management information system using the Laravel 11 framework at PT. Esa Data Teknika, Bogor, with a focus on improving efficiency and accuracy in project management.  
The methodology employed in this study is the framework method, which serves as a structured approach designed to facilitate the development and maintenance of applications.  
The system design was carried out using Unified Modeling Language (UML), which includes the Use Case Diagram, Activity Diagram, Class Diagram, and Sequence Diagram to define the system’s structure and workflow.  
The results of this study are an information system that provides strong support for the operational needs of PT. Esa Data Teknika in project management. The system is well-integrated in terms of data management and decision-making based on the generated reports.

***Keywords****: Information System, Project Management, Laravel Framework*

**KATA PENGANTAR**

Segala puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan berkat, anugrah, dan karunia yang melimpah, sehingga penulis dapat menyelesaikan Skripsi/Tugas akhir ini

STMIK Pranata Indonesia sebagai sebuah lembaga pendidikan tinggi yang tumbuh dan berkembang, memiliki kumpulan data dan informasi penting yang harus dikelola dengan benar, dijaga kerahasian, integritas dan ketersediaannya, agar data atau informasi hanya dapat diakses oleh yang berwenang, tidak diubah oleh siapapun yang tidak berhak. Informasi harus akurat, dan tersedia saat dibutuhkan.

Pengelolaan sistem informasi menjadi penting ketika terkait dengan kelangsungan hidup orang banyak. Ketika perusahaan menempatkan informasi sebagai infrastruktur penting, maka pengelolaan system informasi yang dimiliki menjadi prioritas utama demi kelangsungan hidup dan perkembangan perusahaan

Dengan melihat kondisi ini penulis menyusun sebuah Skripsi Tugas Akhir yang berjudul "Sistem Informasi Manajemen Proyek menggunakan framework laravel 11 di PT. Esa Data Teknika Bogor".

Skripsi/Tugas Akhir tersebut melengkapi salah satu persyaratan yang diajukan dalam rangka menempuh ujian akhir untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Sarjana (S1), Sekolah Tinggi Manajemen Informatika dan Komputer Pranata Indonesia

Penulis sungguh sangat menyadari, bahwa penulisan Skripsi/Tugas Akhir ini tidak akan terwujud tanpa adanya dukungan dan bantuan dari perbagai pihak. Maka, dalam kesempatan ini penulis menghaturkan penghargaan dan ucapan terima kasih yang sebesar besarnya kepada yang terhormat :

1. Drs Harry Subagjo MM., M.Pd, selaku Ketua STMIK Pranata Indonesia
2. Imam Zaenudin S.Kom., M.Kom, selaku Wakil Ketua I Bidang Akademik
3. Nurhadi, S.Kom., M.Kom, selaku Ketua Program Studi Sistem Informatika
4. Imam Zaendin S.Kom M.Kom sebagai Dosen Pembimbing I (materi) yang telah memberi arahan dan waktunya kepada penulis selama penulisan ini berlangsung sampai selesai
5. Drs Harry Subagjo MM., M.Pd sebagai pembimbing II (penulisan) yang telah memberi arahan dan waktunya kepada penulis selama penulisan ini berlangsung sampai selesai
6. Bapak dan Ibu Dosen STMIK Pranata Indonesia yang telah memberikan ilmu pengetahuan kepada penulis
7. Bapak Robi Alfian Muhtar sebagai Direktur PT. Esa Data Teknika observasi penelitian untuk Skripsi/Tugas Akhir tempat penulis mengadakan
8. Orang tua tercinta yang telah memberikan kasih sayang, semangat, dan doa sehingga penulisan ini berjalan dengan baik.
9. Guru-guru yang telah menjadi seperti orang tua sendiri, yaitu Ibu Setiani, Ibu Alpita Susanti, dan Bapak Andi Muhamad Yusup, yang senantiasa memberikan bimbingan, motivasi, serta teladan berharga bagi penulis.
10. Teman satu pembimbing, Muhammad Maulana, Andi Sulaeman, Satrio Budiyantoro, Reza Abdul Nazir, Dicko Andrian Sumardiyansah, Rio Adrian, Egitia Setiawan serta teman-teman lain yang selalu memberikan semangat, masukan positif, dan dukungan kepada penulis selama penyusunan Skripsi/Tugas Akhir ini.
11. Semua pihak yang tidak dapat disebutkan satu persatu, terimakasih atas kerja samanya sehingga penulisan ini dapat diselesaikan dengan tepat waktu.

Akhir kata penulis mohon maaf atas kekeliruan dan kesalahan yang terdapat dalam Skripsi/Tugas Akhir dan berharap semoga Skripsi/Tugas Akhir ini dapat bermanfaat dan menambah wawasan bagi khasanah pengetahuan Teknologi Informasi di Indonesia.

Bekasi, 20 Agustus 2025

Penulis

DAFTAR ISI

COVER i

LEMBAR MOTTO & PERSEMBAHAN ii

LEMBAR PERSETUJUAN PEMBIMBING iii

LEMBAR PENGESAHAN SIDANG YUDISIUM iv

LEMBAR PERNYATAAN KEASLIAN SKRIPSI/TA v

ABSTRAKSI vi

ABSTRACTION vii

KATA PENGANTAR viii

DAFTAR ISI x

DAFTAR TABEL xii

DAFTAR GAMBAR xiii

**BAB I PENDAHULUAN 1**

1. Latar Belakang Masalah 1
2. Identifikasi Masalah 2
3. Pembatasan Masalah 3
4. Perumusan Masalah 3
5. Tujuan dan Manfaat Penelitian 4
6. Sistematika Penulisan 3

**BAB II TINJAUAN PUSTAKA 5**

1. Tinjauan Pustaka 5
2. Landasan Teori 6
3. Kerangka Pemikiran 33

**BAB III METODOLOGI PENELITIAN 37**

1. Analisa Kebutuhan 37
2. Perancangan Penelitian 38
3. Jadwal Kegiatan dan Waktu Penelitian 40

**BAB IV PEMBAHASAN DAN HASIL 37**

1. Pembahasan 37
2. Hasil 38

**BAB V PENUTUP 37**

1. Kesimpulan 37
2. Saran 38

**DAFTAR PUSTAKA**

**DAFTAR RIWAYAT HIDUP**

**LAMPIRAN-LAMPIRAN**

**DAFTAR TABEL**

[Tabel 2.1 Penelitian yang relevan](#_Toc125725644) 7

Tabel 2.2 *Class Diagram* 10

Tabel 2.3 *Object Diagram* 12

Tabel 2.4 *Component Diagram* 13

Tabel 2.5 *Composite Structure Diagram* 14

Tabel 2.6 *Deployment Diagram* 16

Tabel 2.7 *Package Diagram* 17

Tabel 2.8 *Use Case Diagram* 18

Tabel 2.9 *Activity Diagram* 20

Tabel 2.10 *State Machine Diagram* 21

Tabel 2.11 *Sequence Diagram* 22

Tabel 2.12 *Communication Diagram* 25

Tabel 2.13 Simbol *flowchart* dokument 27

Tabel 3.1 Struktur Pengkodean 52

Tabel 3.2 Rencana Kegiatan Penelitian 75

Tabel 4.1 Tabel Users 78

Tabel 4.2 Tabel Companies 79

Tabel 4.3 Tabel Third Parties 79

Tabel 4.4 Tabel Projects 80

Tabel 4.5 Tabel Purchase Orders 80

Tabel 4.6 Tabel Item Orders 81

Tabel 4.7 Tabel File Orders 81

Tabel 4.8 Tabel Invoices 82

Tabel 4.9 Tabel Delivery Orders 82

Tabel 4.10 Tabel Delivery Items 83

Tabel 4.11 Tabel Unexpected Expenses 83

Tabel 4.12 Tabel Tasks 83

Tabel 4.13 Tabel hasil pengujian 88

**DAFTAR GAMBAR**

[Gambar 2.1. Diagram UML](#_Toc125725458) 9

Gambar 2.2 *Timing Diagram* 27

Gambar 2.3 *Interaction Overview Diagram* 28

Gambar 2.4 *Contoh Flowchart* Dokumen *3*1

Gambar 2.5 Kerangka Pemikiran 34

Gambar 3.1 Struktur Organisasi PT. Esa Data Teknika 39

Gambar 3.2 *Flowchart* Dokumen Sistem Berjalan Manajemen Proyek 45

Gambar 3.3 Dokumentasi *input* formulir pemesaran proyek 49

Gambar 3.4 Dokumentasi *output* formulir BAST 50

Gambar 3.5 Sistem usulan 55

Gambar 3.6 *Use Case* Sistem Manajemen Proyek 56

Gambar 3.7 *Activity* Diagram *Login* 57

Gambar 3.8 *Activity* Diagram *Company* 58

Gambar 3.9 *Activity* Diagram *Users* 58

Gambar 3.10 *Activity* Diagram *Third Party* 59

Gambar 3.11 *Activity* Diagram *Projects* 59

Gambar 3.12 *Activity* Diagram *Purchase Orders* 60

Gambar 3.13 *Activity* Diagram *Delivery Orders* 60

Gambar 3.14 *Class* Diagram PT. Esa Data Teknika 61

Gambar 3.15 *Sequence* Diagram proses login 62

Gambar 3.16 *Sequence* Diagram proses logout 62

Gambar 3.17 *Sequence* Diagram proses mengelola data company 63

Gambar 3.18 *Sequence* Diagram proses tambah data *users* 63

Gambar 3.19 *Sequence* Diagram proses *edit* data *users* 64

Gambar 3.20 *Sequence* Diagram proses hapus data *users* 64

Gambar 3.21 *Sequence* Diagram proses tambah data *third party* 65

Gambar 3.22 *Sequence* Diagram proses *edit* data *third party* 65

Gambar 3.23 *Sequence* Diagram proses hapus data *third party* 66

Gambar 3.24 *Sequence* Diagram proses tambah data *project* 66

Gambar 3.25 *Sequence* Diagram proses *edit* data *project* 67

Gambar 3.26 *Sequence* Diagram proses hapus data *project* 68

Gambar 3.27 *Sequence* Diagram proses tambah data *purchase order* 68

Gambar 3.28 *Sequence* Diagram proses *edit* data *purchase order* 69

Gambar 3.29 *Sequence* Diagram proses hapus data *purchase order* 69

Gambar 3.30 *Sequence* Diagram proses tambah data *delivery order* 70

Gambar 3.31 *Sequence* Diagram proses *edit* data *delivery order* 70

Gambar 3.32 *Sequence* Diagram proses hapus data *delivery order* 71

Gambar 3.33 Rancangan *form login* 71

Gambar 3.34 Rancangan *dashboard* 72

Gambar 3.35 Rancangan *form project* 72

Gambar 3.36 Rancangan *form purchase order* 73

Gambar 3.37 Rancangan *form delivery order* 73

Gambar 3.38 Rancangan halaman *sales report* 74

Gambar 4.1 Hubungan antar tabel 78

Gambar 4.2 Tampilan halaman *login* 84

Gambar 4.3 Tampilan halaman *dashboard* 85

Gambar 4.4 Tampilan halaman companies 85

Gambar 4.5 Tampilanhalaman *users* 85

Gambar 4.6 Tampilan halaman *third parties* 86

Gambar 4.7 Tampilan halaman *projects* 86

Gambar 4.8 Tampilan halaman *purchase orders* 86

Gambar 4.9 Tampilan halaman *delivery orders* 87

Gambar 4.10 Tampilan halaman *roles* 87

Gambar 4.11 Tampilan halaman *sales reports* 87

BAB I

PENDAHULUAN

1. **Latar Belakang Masalah**

Manajemen proyek merupakan salah satu elemen penting dalam memastikan keberhasilan operasional perusahaan di berbagai sektor industri. Pengelolaan proyek yang baik tidak hanya berkaitan dengan pelacakan progres dan alokasi sumber daya, tetapi juga mencakup kecepatan akses informasi, transparansi data, serta kemampuan untuk melakukan koordinasi lintas divisi secara *real-time*. Oleh karena itu, keberadaan sistem manajemen proyek yang terdigitalisasi menjadi kebutuhan mendesak bagi perusahaan yang ingin tetap kompetitif.

Saat ini, PT. Esa Data Teknika masih mengandalkan Microsoft Excel sebagai alat utama dalam pengelolaan informasi proyek, mulai dari pencatatan data proyek, pelacakan progres, hingga pembuatan laporan. Meskipun Excel cukup fleksibel dan mudah digunakan, pendekatan ini memiliki sejumlah keterbatasan, seperti kurangnya sistem validasi data yang kuat, tidak adanya update data secara real-time yang terintegrasi, serta risiko hilangnya data akibat human error atau kerusakan file. Selain itu, pengelolaan data proyek yang tersebar di banyak file Excel menyulitkan perusahaan untuk mendapatkan gambaran utuh dan terkini mengenai kondisi proyek secara menyeluruh.

Kondisi serupa banyak dijumpai di perusahaan lain yang masih bergantung pada pengelolaan proyek secara manual atau semi-manual. Kurangnya sistem informasi terpusat, tidak adanya pelacakan otomatis terhadap aktivitas proyek, hingga minimnya dokumentasi progres menjadi kendala utama dalam menjaga keberlangsungan dan kualitas proyek. Dalam situasi ini, perusahaan rawan mengalami keterlambatan penyelesaian proyek, meningkatnya beban kerja administratif, serta sulitnya mengukur kinerja tim secara objektif. Jika tidak segera ditangani, hal ini dapat berdampak pada menurunnya reputasi perusahaan dan hilangnya peluang bisnis di tengah persaingan industri yang semakin ketat.

Sebagai solusi strategis, Maka penulis menentukan skripsi dengan judul **Sistem Informasi Manajemen Proyek Menggunakan Framework Laravel 11 di PT. Esa Data Teknika Bogor** yang akan menjadi langkah penting untuk mentransformasikan proses manajemen proyek ke arah yang lebih modern dan efisien. Sistem ini akan mengintegrasikan seluruh data proyek dalam satu platform yang terpusat, memungkinkan akses real-time, pelacakan progres secara sistematis, serta peningkatan akurasi dan keamanan data. Dengan demikian, perusahaan dapat meningkatkan efektivitas operasional, mempercepat koordinasi tim, dan memperkuat kemampuan dalam menyelesaikan proyek secara tepat waktu dan terukur.

1. **Identifikasi Masalah**

Berdasarkan latar belakang yang sudah dijelaskan, penulis merumuskan beberapa masalah sebagai berikut :

1. Proses pengelolaan data proyek masih dilakukan secara manual menggunakan Microsoft Excel, yang mengakibatkan potensi duplikasi, kesalahan input, serta sulitnya integrasi antar file. Ketidakterpaduan ini menyulitkan tim dalam mendapatkan data yang konsisten dan akurat.
2. Data proyek disimpan secara lokal di masing-masing perangkat tanpa sistem backup dan kontrol akses yang baik. Hal ini berisiko terhadap kehilangan data penting serta menyulitkan tim dalam mengakses informasi proyek secara fleksibel dan aman, terutama saat dibutuhkan secara mendadak.
3. Tidak adanya sistem monitoring yang mendukung pelacakan progres proyek secara real-time membuat tim manajemen kesulitan dalam mengetahui status terkini dari setiap tahapan pekerjaan. Hal ini menghambat identifikasi dini terhadap keterlambatan atau kendala teknis di lapangan.
4. Komunikasi antar tim yang tidak sinkron menyebabkan banyak pekerjaan dilakukan berulang dan manual, seperti penginputan ulang data yang akibatnya, efisiensi kerja menurun dan waktu penyelesaian proyek menjadi lebih lama.
5. Kesulitan dalam mengakses data historis dan membandingkan progres proyek antar periode membuat proses evaluasi menjadi kurang maksimal. Tanpa analisis berbasis data, perusahaan kesulitan dalam menyusun strategi perbaikan dan pengambilan keputusan jangka panjang.
6. Pembuatan laporan proyek dilakukan secara terpisah dan tidak terstruktur, sehingga memperlambat proses penyusunan laporan berkala. Selain itu, laporan yang dihasilkan cenderung statis dan tidak mudah diperbarui sesuai perkembangan proyek.
7. **Pembatasan Masalah**

Dapat diketahui bahwa batasan masalah penulisan ini yaitu:

* 1. Sistem informasi yang dikembangkan hanya akan berbasis web, tidak mencakup pengembangan aplikasi mobile atau sistem berbasis desktop.

1. **Perumusan Masalah**

Berdasarkan latar belakang, identifikasi masalah, dan pembatasan masalah yang telah dijelaskan, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengembangkan sistem informasi manajemen proyek berbasis web yang dapat mempermudah pencatatan data proyek, pembuatan laporan, dan pemantauan progres proyek?
2. Bagaimana merancang dan membangun basis data yang terpusat dan terstruktur untuk menyimpan, mengelola, dan mengamankan data proyek secara efisien agar dapat diakses secara real-time oleh seluruh tim terkait?
3. **Tujuan dan Manfaat Penelitian**

Berdasarkan latar belakang yang telah disampaikan, tujuan dan manfaat penelitian ini adalah sebagai berikut:

1. Tujuan
2. Mengembangkan sistem informasi manajemen proyek berbasis web yang terintegrasi guna meningkatkan efisiensi operasional, pencatatan data proyek, pembuatan laporan, dan pemantauan progres proyek secara digital di PT. Esa Data Teknika.
3. Menyediakan alat bantu sistem yang mampu mendukung pencatatan proyek, pemantauan progres, dan koordinasi antar tim secara *real-time*.
4. Memenuhi salah satu syarat sidang Yudisium Strata-1 di STMIK Pranata Indonesia.
5. Manfaat
6. Meningkatkan efisiensi pengelolaan proyek, meminimalkan risiko kesalahan pencatatan, serta mempercepat proses pemantauan dan pelaporan proyek secara terpusat dan realtime bagi PT. Esa Data Teknika.
7. Mempermudah akses data, memperlancar koordinasi antar divisi, dan mendukung proses evaluasi serta pengambilan keputusan.
8. Menambah wawasan dan pengalaman penulis dalam menerapkan ilmu perkuliahan ke dunia kerja, serta menjadi referensi untuk penelitian atau pengembangan sistem manajemen proyek.
9. **Sistematika Penulisan**

**BAB I PENDAHULUAN**

Bab ini memuat latar belakang masalah, identifikasi masalah, pembatasan masalah, perumusan masalah, tujuan dan manfaat penelitian, sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Bab ini membahas tentang landasan teori, pengertian-pengertian, teori tentang materi pembahasan, teori tentang program aplikasi, kerangka pemikiran yang merupakan sebuah bagan alir kerja dalam memecahkan permasalahan penelitian.

**BAB III METODOLOGI PENELITIAN**

Bab ini menyajikan secara sederhana langkah-langkah penelitian yang dilakukan. Mulai dari analisa kebnutuhan, perancangan penelitian sampai jadwal kegiatan dan waktu penelitian.

**BAB IV PEMBAHASAN DAN HASIL**

Bab ini berisi pembahasan dan hasil penelitian yang diolah, dianalisis, ditafsirkan. Mulai dari analisa kebutuhan, perancangan penelitian sampai jadwal kegiatan dan waktu penelitian.

**BAB V PENUTUP**

Bab ini merupakan hasil penulisan skripsi yang berisi kesimpulan beserta saran-saran terhadap sistem untuk pengembangan dengan harapan bisa bermanfaat bagi semua pihak.

**BAB II**

**TINJAUAN PUSTAKA**

* 1. **Tinjauan Pustaka**

Berikut 5 penelitian yang relevan digunakan sebagai perbandingan dan acuan untuk membuat sistem seperti terlihat pada Tabel 2.1 di bawah ini.

TABEL: 2.1. Penelitian yang relevan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Penulis / Tahun | Topik Penelitian | Metode | Hasil | Perbedaan |
| 1 | Shila Kartika Barus & Yahfizham (2022) | Sistem Informasi Manajemen Proyek pada perusahaan kontraktor | Waterfall | Meningkatkan efisiensi operasional, transparansi, dan monitoring progres proyek di perusahaan kontraktor | Sistem berfokus pada proyek konstruksi skala besar. |
| 2 | Hambali, Lalu Delsi & Panji Wijayanto (2022) | Pengembangan sistem manajemen proyek berbasis web (CodeIgniter) | Waterfall | Sistem mendukung multi-role user (admin, CEO, client, divisi) dan pelaporan terintegrasi | Menggunakan CodeIgniter dan mendukung multi-role user. |
| 3 | Desi Ardila & Yahfizham (2024) | Sistem manajemen proyek jasa menjahit berbasis web | Waterfall | Meningkatkan efisiensi dalam penjadwalan dan monitoring layanan jahit | Sistem sederhana untuk usaha jasa jahit. |
| 4 | Tamaulina Barus (2024) | SIMPRO untuk Bengkel Las Listrik Sri Rezeky | Waterfall | Mempermudah pengelolaan jadwal, pengerjaan, dan pelaporan proyek las | Hanya untuk pelacakan proyek bengkel las. |
| 5 | Larasati & Yahfizham (2024) | Sistem Informasi Manajemen Proyek untuk kontraktor | Waterfall | Mengatur distribusi tugas, kontrol progres, dan pelaporan harian proyek | Digunakan pada proyek lapangan tanpa integrasi lintas divisi. |

Kesimpulan :

Berdasarkan pembahasan pada Tabel 2.1 dapat disimpulkan bahwa penelitian yang dikembangkan dari beberapa penelitian sebelumnya dengan penelitian yang dibuat penulis memiliki perbedaan dalam desain dan rancangan aplikasi.

* 1. **Landasan Teori**

1. **Pengertian**
   * 1. Pengertian Sistem

“Dalam konsep dasar sistem, istilah “sistem” merujuk pada suatu entitas yang terdiri dari elemen atau komponen yang saling berhubungan dan berinteraksi secara kontinu atau berkesinambungan dengan lingkungan sekitarnya.” (Widarti, E., Sari, F. P., & Mardian, R. 2024).

* + 1. Pengertian Informasi

“Informasi adalah kumpulan informasi atau kenyataan yang diolah dengan cara khusus sehingga mempunyai tujuan bagi penerima informasi tersebut.” (Widarti, E., Sari, F.P., & Mardian, R. 2024).

* + 1. Pengertian Sistem Informasi

“Sistem informasi dirancang untuk mencapai tujuan tertentu dalam organisasi. Tujuan ini dapat berkaitan dengan efisiensi operasional, pengambilan keputusan yang lebih baik, peningkatan produktivitas, maupun dukungan terhadap strategi bisnis.” (Widarti, E., Sari, F.P., & Mardian, R. 2024).

* + 1. Pengertian Manajemen Proyek

”Proyek adalah urutan kegiatan yang unik, kompleks, dan saling terkait, memiliki satu tujuan, dan tujuan harus diselesaikan dalam waktu tertentu, sesuai anggaran, dan memenuhi spesifikasi. Manajemen/pengelolaan proyek perangkat lunak bertujuan agar perangkat lunak yang dibuat sampai ke tangan pelanggan (*customer*) tepat waktu dan sesuai dengan harapan pelanggan (*customer*).” (A.S., Rosa dan Shalahuddin, 2019).

* + 1. Pengertian Web

“Website merupakan kumpulan halaman digital yang berisi informasi berupa teks, animasi, gambar, suara dan video atau gabungan dari semuanya yang terkoneksi oleh internet, sehingga dapat dilihat oleh seluruh siapapun yang terkoneksi jaringan internet.” (Sari, A. O., Abdilah, A., & Sunarti. 2019).

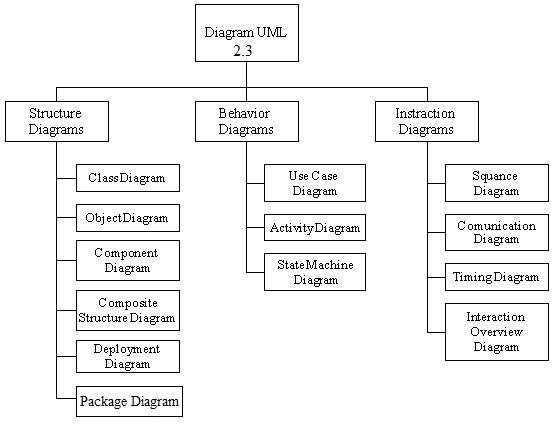
1. **Teori Tentang Materi Pembahasan**
   * 1. UML (*Unified Modelling Language*)

“UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.” (A.S., Rosa & Shalahuddin, M. 2019).

* + 1. Diagram-diagram UML (*Unified Modelling Language*)

“Pada 1996, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September tahun 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusi nya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya.” (A.S., Rosa & Shalahuddin, M. 2019).

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini.



GAMBAR: 2.1. Diagram UML (A.S., Rosa dan Shalahuddin, 2019:140)

* + - 1. *Class* Diagram

“Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.” (A.S., Rosa dan Shalahuddin, 2019)

Seperti terlihat pada contoh tabel 2.2 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Class* Diagram.

TABEL: 2.2. *Class* Diagram (A.S., Rosa dan Shalahuddin, 2019:146)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Kelas | Kelas pada struktur sistem |
| Antarmuka / *interface*  **nama\_*interface*** | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| Asosiasi / *association* | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity* |
| Asosiasi berarah / *directed association* | Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity* |
| Generalisasi | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus) |
| Kebergantungan / *dependency* | Relasi antarkelas dengan makna kebergantungan antarkelas |
| Agregasi / *aggregation* | Relasi antarkelas dengan makna semua-bagian (*whole-part*) |

* + - 1. *Object* Diagram

“*Object* diagram atau diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan”. (A.S., Rosa dan Shalahuddin, 2019).

Seperti terlihat pada contoh tabel 2.3 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada Object Diagram.

TABEL: 2.3. *Object* Diagram (A.S., Rosa dan Shalahuddin, 2019:147)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| Objek | Objek dari kelas yang berjalan saat sistem dijalankan |
| *Link* | Relasi antar objek |

* + - 1. *Component* Diagram

Diagram komponen atau *Component* Diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut:

* + - * 1. *Source code* program perangkat lunak
        2. Komponen *executable* yang dilepas ke *user*
        3. Basis data secara fisik
        4. Sistem yang harus beradaptasi dengan sistem lain
        5. *Framework* sistem, *framework* pada perangkat lunak merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi, contohnya seperti *Struts* dari *Apache* yang menggunakan prinsip desain Model-*View-Controller* (MVC) dimana *source code* program dikelompokkan berdasarkan fungsinya.

Seperti terlihat pada contoh tabel 2.4 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Component* Diagram.

TABEL: 2.4. *Component* Diagram (A.S., Rosa dan Shalahuddin, 2019:149)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Package*    package | *Package* merupakan sebuah bungkusan dari satu atau lebih komponen |
| Komponen  **nama\_komponen** | Komponen sistem |
| Kebergantungan *dependency* | Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai |
| *Interface*  **nama\_interface** | Sama dengan konsep *interface* pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen |
| *Link* | Relasi antar komponen |

* + - 1. *Composite Structure* Diagram

“*Composite Structure* Diagram baru dimulai ada pada UML versi 2.0, pada versi 1.x diagram ini belum muncul. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (runtime) dari instance yang saling terhubung. Dapat menggambarkan struktur didalam kelas atau kolaborasi. Contoh penggunaan diagram ini misalnya untuk menggambarkan deskripsi dari setiap bagian mesin yang saling terkait untuk menjalankan fungsi mesin tersebut, menggambarkan aliran data router pada jaringan komputer, dan lain-lain.” (A.S., Rosa dan Shalahuddin, 2019).

Seperti terlihat pada contoh tabel 2.5 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada Composite Structure Diagram.

TABEL: 2.5. *Composite Structure* Diagram (A.S., Rosa dan Shalahuddin, 2019:151)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Property*  roleName: TypeNama | *Property* adalah satu set dari suatu *instance*.  roleName: peran / nama / identitas dari *property* (opsional)  TypeName : tipe kelas dari *property* (harus ada) |
| *Connector*  {multiplicity1} {multiplicity2}  {roleName1} {roleName2}  nama *connector* : tipe *connector* | *Connector* adalah cara komunikasi dari dua buah *instance*  ConnName : nama *connector* (opsional)  ConnType : tipe *connector* (opsional) |
| *Port* | *Port* merupakan cara yang digunakan dalam diagram *composite structure* tanpa menampilkan detail internal dari suatu sistem. |
| *Property Property* | *Port* digambarkan dalam bentuk kotak kecil yang menempel atau di dalam suatu *property.* |
|  | *Port* digambarkan menempel property jika fungsi tersebut dapat diakses public. Sedangkan *port* digambarkan di dalam suatu property jika fungsi tersebut bersifat *protected* |
| *Class*  *Class*  *assclass* | Kelas; jika yang akan dijabarkan strukturnya adalah sebuah kelas |

* + - 1. *Deployment* Diagram

Diagram *deployment* atau *deployment* diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram *deployment* juga dapat digunakan untuk memodelkan hal-hal berikut:

* + - * 1. Sistem tambahan (*embedded system*) yang menggambarkan rancangan *device, node,* dan *hardware*.
        2. Sistem *client* *server*.
        3. Sistem terdistribusi murni.
        4. Rekayasa ulang aplikasi.

Seperti terlihat pada contoh tabel 2.6 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Deployment* Diagram.

TABEL: 2.6. *Deployment* Diagram (A.S., Rosa dan Shalahuddin, 2019:154)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Package* | *Package* merupakan sebuah bungkusan dari satu atau lebih *node* |
| *Node*  Nama\_Node | Biasanya mengacu pada perangkat keras (*hardware*), perangkat lunak yang tidak dibuat sendiri (*software*), jika didalam *node* disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen |
| Kebergantungan / *Dependency* | Kebergantungan antar *node*, arah panah mengarah pada *node* yang dipakai |
| Link | Relasi antar *node* |

* + - 1. *Package* Diagram

“*Package* Diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan package diagram. Berikut adalah simbol-simbol *Package* Diagram.” (A.S., Rosa dan Shalahuddin. 2019).

Seperti terlihat pada contoh tabel 2.7 dibawah ini (A.S., Rosa dan Shalahuddin. 2019), merupakan simbol-simbol yang ada pada *Package* Diagram.

TABEL: 2.7. *Package* Diagram (A.S., Rosa dan Shalahuddin, 2019:153)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Package* | *Package* merupakan sebuah bungkusan dari satu atau lebih kelas atau elemen UML lainnya. |
| *Elemen Package*  Perpustakaan  Pustaka Anggota Peminjaman | Elemen dalam *package* digambarkan di dalam *package*. |
| perpustakaann  Metode  Atribute  Pustaka  Metode  Atribute  Pustaka  Metode  Atribute  Pustaka | Elemen dalam *package* digambarkan diluar *package* |

* + - 1. *Use Case* Diagram

“*Use* *Case* Diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use* *case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan di buat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.” (A.S., Rosa dan Shalahuddin, 2019).

Seperti terlihat pada contoh tabel 2.8 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Use* *Case* Diagram.

TABEL: 2.8. *Use Case* Diagram (A.S., Rosa dan Shalahuddin, 2019:156)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Aktor / *actor* | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor |
|  | *Use Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case* |
|  | *Asosiasi / association* | Komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* memiliki interaksi dengan aktor |
| <<*include*>> | *Include Relationship* | Relasi cakupan memungkinkan suatu use untuk menggunakan fungsionalitas yang disediakan oleh use case yang lainnya. |
| <<*extended*>> | *Extends Relationship* | Memungkinkan suatu use case memiliki kemungkinan untuk memperluas fungsional yang disediakan *use case* yang lainnya. |

* + - 1. *Activity* Diagram

“Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.” (A.S., Rosa dan Shalahuddin, 2019).

Seperti terlihat pada contoh tabel 2.9 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Activity* Diagram.

TABEL: 2.9. *Activity* Diagram (A.S., Rosa dan Shalahuddin, 2019:162)

|  |  |  |
| --- | --- | --- |
| Simbol | Nama | Keterangan |
|  | Status Awal | Sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Percabangan / *decision* | Percabangan dimana ada pilihan aktivitas yang lebih dari satu. |
|  | Penggabungan / *join* | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu |
|  | Status | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
|  | *Swimlane* | *Swimlane* memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

* + - 1. *State Machine* Diagram

“*State Machine* Diagram atau *statechart* diagram atau dalam bahasa Indonesia disebut diagram mesin status atau sering juga disebut diagram status digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek. Jika diagram sekuen digunakan untuk interaksi antar objek maka diagram status dapat digunakan untuk interaksi didalam sebuah objek.” (A.S., Rosa dan Shalahuddin, 2019).

Seperti terlihat pada contoh tabel 2.10 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *State Machine* Diagram.

TABEL: 2.10. *State Machine* Diagram (A.S., Rosa dan Shalahuddin, 2019:164)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *End* / status akhir (*final state*) | *End* atau *final state* adalah *state* keadaan akhir dari daur hidup suatu sistem. |
| *Event* | *Event* adalah kegiatan yang menyebabkan berubahnya status mesin. |
| *State*  *state* | Sistem pada waktu tertentu. *State* dapat berubah jika ada *event* tertentu yang memicu perubahan tersebut. |

* + - 1. *Sequence* Diagram

“Diagram sekuen atau *Sequence* diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.” (A.S., Rosa dan Shalahuddin, 2019).

Seperti terlihat pada contoh tabel 2.11 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Sequence* Diagram.

TABEL: 2.11. *Sequence* Diagram (A.S., Rosa dan Shalahuddin, 2019:165)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Actor*    Nama aktor  Atau  Nama aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat  itu sendiri, jadi walaupun simbol dari aktor merupakan orang, biasanya dinyatakan menggunakan kata benda. |
| Garis hidup / *lifeline* | Menyatakan kehidupan suatu objek. |
| Objek  Nama objek : nama | Menyatakan objek yang berinteraksi pesan. |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya  1 : login ()  2: Cek status login ()  3 : open ()  Maka cekStatusLogin () dan open () dilakukan didalam metode login () aktor tidak memiliki waktu aktif. |
| <<*create*>> | Objek yang lain, arah panah mengarah pada objek yang dibuat. |
| Pesan tipe *call*  1 : nama\_metode() | Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri,  1: Nama\_metode() |
|  | Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| Pesan tipe *send*  1: masukan | Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek yang lainnya. Arah panah mengarah pada objek yang dikirim. |
| Pesan tipe *return* 1: keluaran | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau ke objek tertentu, arah panah mengarah pada objek yang menerima kembali. |
| Pesan tipe *destroy*  <<*destroy*>> | Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada crate maka ada destroy. |

* + - 1. *Communication* Diagram

*Communication* diagram atau diagram komunikasi pada UML versi 2.x merupakan penyederhanaan dari diagram kolaborasi (*collaboration* diagram) pada UML versi 1.x. *Collaboration* diagram sudah tidak muncul lagi pada UML versi 2.x diagram komunikasi sebenarnya adalah diagram kolaborasi tetapi dibuat untuk tiap sekuen.

Diagram komunikasi menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang telah diperoleh dari diagram kelas, diagram sekuen, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

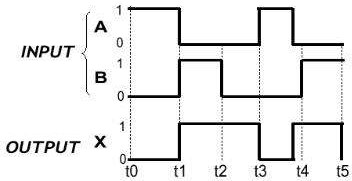
Seperti terlihat pada contoh tabel 2.12 dibawah ini (A.S., Rosa dan Shalahuddin, 2019), merupakan simbol-simbol yang ada pada *Communication* Diagram.

TABEL: 2.12. *Communication* Diagram (A.S., Rosa dan Shalahuddin, 2019:168)

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| *Object* | Merupakan *instance* dari sebuah class yang dituliskan dan disusun secara horizontal. |
| *Message* | Merupakan suatu garis penghubung yang menunjukan adanya komunikasi antar objek satu dengan objek lainnya. |
| *Link* | Merupakan suatu garis yang berfungsi menghubungkan objek satu dengan objek lain |

* + - 1. *Timing* Diagram

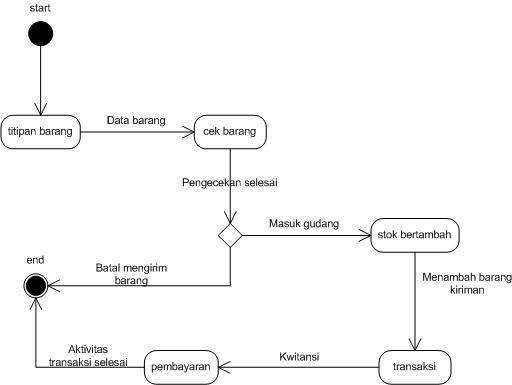
*“Timing* Diagram merupakan diagram yang fokus pada penggambaran terkait atas waktu. *Timing* diagram digunakan untuk menggambarkan tingkah laku sistem dalam periode waktu tertentu., biasanya digunakan mendeskripsikan operasi dari alat digital karena penggambarannya secara visual akan lebih mudah dipahami.”(A.S., Rosa dan Shalahuddin, 2019).

Berikut ini merupakan contoh gambar pemanfaatan dari *Timing* Diagram.

GAMBAR: 2.2. *Timing* Diagram (A.S., Rosa dan Shalahuddin, 2019:170)

* + - 1. *Interaction Overview* Diagram

*“Interaction Overview* Diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas. Interaksi diagram dapat meliputi diagram sekuen, diagram komunikasi, *interaction over view diargam*. Sebagai contoh *initial,* dan *nodes* sama seperti pada diagram aktivitas.” (A.S., Rosa dan Shalahuddin, 2019).

GAMBAR: 2.3. *Interaction Overview* Diagram (A.S., Rosa dan Shalahuddin, 2019:173)

* + 1. *Document Flowchart*

“Diagram alur atau *Flowchart* adalah representasi diagram dari langkah-langkah suatu algoritma.” (Irawan, Muhammad Dedi. 2022).

Simbol dalam *flowchart* dokumen dapat dilihat pada tabel 2.13 berikut:

TABEL: 2.13. Simbol *flowchart* dokumen (Irawan, Muhammad Dedi. 2022:18)

|  |  |  |
| --- | --- | --- |
| Simbol | Arti | Keterangan |
|  | Dokumen | Menunjukkan dokumen *input* dan *output.* |
|  | Kegiatan *manual* | Menunjukkan pekerjaan manual |
|  | Pengurutan *offline* | Menunjukkan proses pengurutan data diluar proses komputer. |
|  | Simpanan *offline* | File yang diarsip baik urut angka, *alphabed,* dan tanggal |
|  | Kartu Plong | Menunjukkan *input* dan *Output* yang menggunakan kartu plong *(punched card)* |
|  | Proses | Menunjukkan kegiatan proses dari operasi komputer |
|  | Operasi luar | Menunjukkan operasi yang dilakukan diluar proses operasi komputer |
|  | Pita Magnetik | Menunjukkan *input*/*output* menggunakan peta *magnetic* |
|  | *Hardisc* | Menunjukkan *input/output* menggunakan hardisc |
|  | *Diskette* | Menunjukkan *input/output* menggunakan *diskette* |
|  | Drum Magnetik | Menunjukkan *input/output* menggunakan drum *magnetic* |
|  | Pita kertas berlubang | Menunjukkan *input/output* menggunakan pita kertas berlubang |
|  | Keyboard | Menunjukkan *input* menggunakan keyboard |
|  | Display | Menunjukkan *output* yang ditampilkan di monitor |
|  | Arus data | Menunjukkan arus dari sebuah proses |
|  | Penjelasan | Menunjukkan penjelasan dari sebuah proses |
|  | Penghubung | Menunjukkan penghubung ke halaman yang masih sama atau halaman baru |
|  | Mulai dan akhir | Menunjukkan mulai dan akhir dari suatu proses |
|  | Pita kontrol | Menunjukkan penggunaan pita control untuk pencocokan proses |
|  | Hubungan komunikasi | Menunjukkan proses transmisi data melalui *channel* komunikasi |

* 1. **Teori Tentang Program Aplikasi**

Untuk menunjang pengembangan sistem informasi layanan pelanggan berbasis *web*, dibutuhkan pemahaman terhadap berbagai komponen teknis yang membentuk dasar dari program aplikasi. Oleh karena itu, pada bagian ini akan dibahas teori-teori terkait perangkat lunak pendukung seperti *web* *server*, *web* *browser*, bahasa pemrograman *web*, basis data, serta alat bantu (*tools*) yang digunakan dalam proses pengembangan aplikasi.

* + 1. *Web* *Server*

*“Web* *Server* merupakan sebuah *software* yang memberikan layanan berbasis data dan berfungsi untuk menerima permintaan dari HTTP atau HTTPS pada klien yang dikenal sebagai *web* *browser.”* (Wijayanti, H. 2020).

*Web* *server* sendiri memiliki beberapa jenis, salah satunya yang paling banyak digunakan dalam pengembangan aplikasi berbasis *web* adalah ***Apache***. *Web* *server* ini dikenal karena fleksibilitas dan kompatibilitasnya di berbagai sistem operasi.

*“Apache* merupakan *web* *server* antar-*platform* yang dapat berjalan di beberapa *platform*, seperti *Linux* dan *Windows*.” (Wijayanti, H. 2020).

* + 1. *Web* *Browser*

“*Web* *browser* merupakan tool utama yang digunakan untuk menampilkan halaman *web* (*homepage*).” (Muthohir, M. 2020).

Terdapat berbagai jenis *web* *browser* yang dapat digunakan oleh pengguna maupun *developer*, dan masing-masing memiliki kelebihan tersendiri. Salah satu *browser* yang cukup populer di kalangan pengembang *web* adalah ***Mozilla Firefox****.*

“*Firefox* merupakan salah satu *browser* yang digemari oleh para *developer* *web* karena *Add-on* yang beragam.“ (Muthohir, M. 2020).

* + 1. *Web* *Programming*

Dalam proses pengembangan aplikasi berbasis *web*, pemilihan *framework* yang tepat sangat berpengaruh terhadap efisiensi dan kemudahan dalam membangun sistem. Salah satu *framework* PHP yang banyak digunakan oleh pengembang karena kemudahan sintaksis dan dukungannya terhadap arsitektur MVC adalah ***Laravel***.

“*Laravel* adalah *framework* *web* PHP gratis *open source* yang dikembangkan oleh Taylor Otwell. *Laravel* ditujukan untuk pengembangan aplikasi berbasis *web*, khususnya bkagi mereka yang mengikuti pola arsitektur MVC (*Model View Controller*).” (Sabaruddin, R., Murni, S., Lisnawanty, & Nugraha, W. 2024).

* + 1. Basis Data

“Basis data atau *database* adalah sekumpulan data yang saling terkait yang disimpan di dalam komputer dan dapat diolah menggunakan perangkat lunak sehingga memudahkan aktivitas untuk memperoleh informasi.” (Sulistyoningrum, T. (2020).

Dalam pengelolaan basis data, dibutuhkan sistem manajemen basis data yang handal untuk mengatur, menyimpan, dan mengakses data secara efisien. Salah satu *database* yang paling banyak digunakan dalam pengembangan aplikasi *web* adalah **MySQL** karena sifatnya yang *open source* dan kompatibel dengan berbagai *platform*.

“MySQL (*My Structured Query Language)* adalah sebuah perangkat lunak sistem manajemen basis data atau *Database* *Management System* (DBMS) yang memiliki karakteristik *multithread* dan *multiuser* serta bersifat *open source* atau gratis di bawah lisensi GNU *General Public License* (GPL).” (Sulistyoningrum, T. (2020).

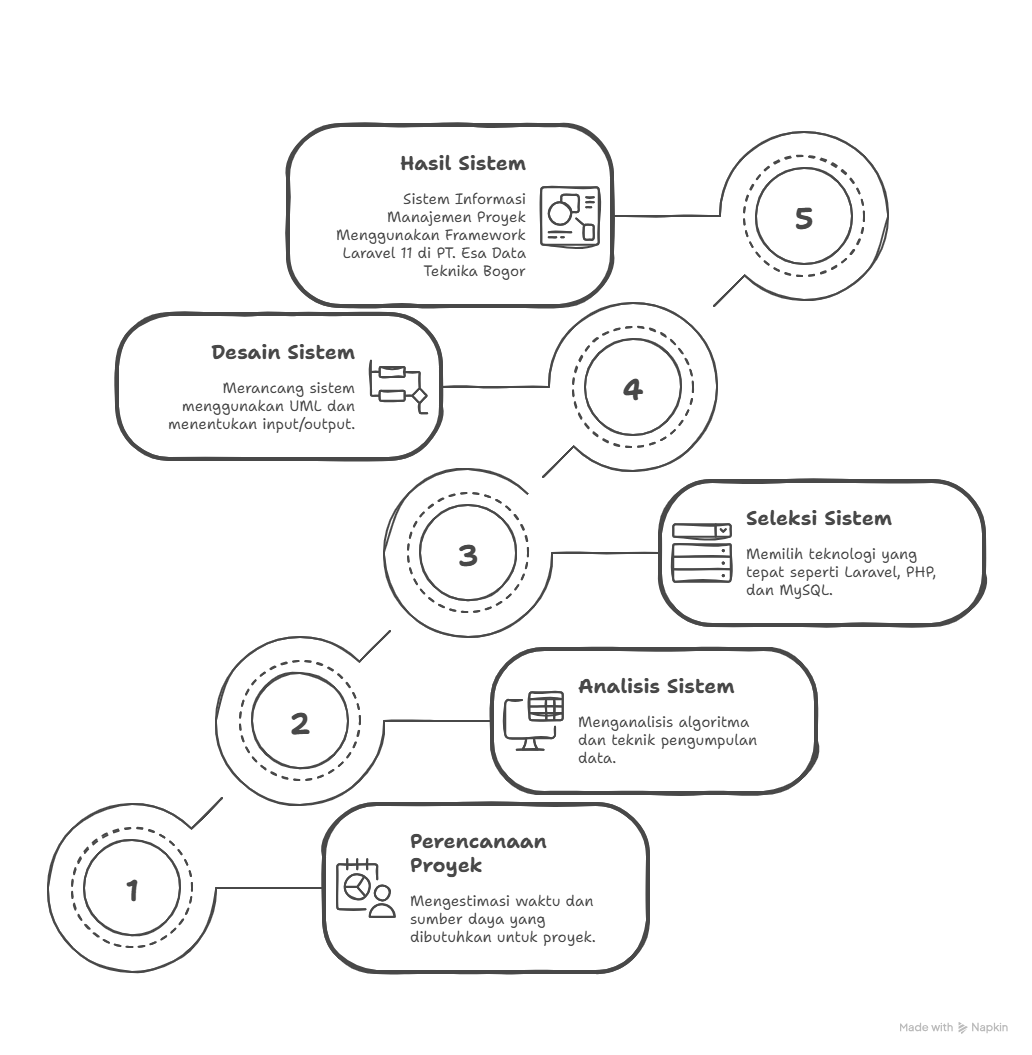
* + 1. *Tools*

Untuk mendukung proses pengembangan sistem informasi, dibutuhkan perangkat bantu (*tools*) yang memudahkan pengembang dalam menulis kode, melakukan *debugging*, dan menjalankan aplikasi secara efisien. Salah satu editor kode yang paling banyak digunakan oleh *developer* adalah ***Visual Studio Code***, karena fleksibilitasnya dan dukungan terhadap berbagai bahasa pemrograman.

*“Visual Studio Code is a cross-platform editor. In this instance, cross-platform means that a version is available to run on Windows (7, 8, and 10), macOS, and Linux.”* (Johnson, B. 2019).

1. **Kerangka Pemikiran**

Kerangka pemikiran merupakan suatu penjelasan tentang kerangka berpikir pengembangan sistem untuk memecahkan masalah yang sedang diteliti, termasuk menguraikan objek penelitian.

****

GAMBAR: 2.5. Kerangka Pemikiran Sistem Informasi Manajemen Proyek Menggunakan Framework Laravel 11 di PT. Esa Data Teknika Bogor

Uraian objek penelitian dalam kerangka pemikiran sebagai berikut :

1. Pengumpulan Data

Pengumpulan data yang dilakukan dalam sistem informasi manajemen proyek menggunakan framework laravel di PT. Esa Data Teknika Bogor sebagai berikut:

* + 1. Observasi, melihat secara langsung proses kerja manual di PT. Esa Data Teknika.
    2. Wawancara untuk menggali informasi dari pihak terkait mengenai kendala sistem yang ada.
    3. Studi Pustaka mencari referensi literatur tentang sistem informasi manajemen proyek.

1. Analisis sistem
2. Algoritma yang merupakan logika serta dokunentasi dari alur berjalan sistem ini.
3. Teknik pengumpulan data yang dilakukan pada penelitian di PT. Esa Data Teknika
4. Desain dan Perancangan sistem

Perancagan adalah proses yang dilakukan setelah melakukan kegiatan tahapan analisis sistem tersebut :

1. Langkah awal yang menjadi acuan dalam perancangan aplikasi adalah dengan membuatnya UML.
2. Perancangan database merupakan penataan pada proses penyimpanan data untuk proses penggalian, penggabungan, pencarian serta manipulasi data.
3. Merancang input/output sebagai sarana komunikasi sistem dangan pengguna.
4. Seleksi Sistem

Dalam seleksi sistem bahasa pemograman yang digunakan dan manipulasi database menggunakan Laravel, PHP dan MYSQL.

1. Hasil Program Aplikasi

Hasil program aplikasi atau sistem baru yang diciptakan dari sistem informasi manajemen proyek berbasis web pada PT. Esa Data Teknika. Sistem informasi manajemen proyek berbasi web mempercepat mendapatkan informasi proyek dan pelaporan rekapan proyek setiap bulannya.

**BAB III**

**METODOLOGI PENELITIAN**

* 1. **Analisa Kebutuhan**
  2. **Gambaran Umum Objek Penelitian**
     1. **Sejarah PT. Esa Data Teknika**

PT. Esa Data Teknika merupakan perusahaan yang menyediakan layanan perencanaan, pembangunan, dan pengelolaan pusat data (data center), sistem kabel terstruktur, fiber optic, sistem keamanan (CCTV dan access control), hingga sistem kelistrikan dan HVAC yang mendukung operasional teknologi secara optimal. Melalui pendekatan profesional dan efisiensi biaya, PT. Esa Data Teknika melayani berbagai skala pelanggan, mulai dari instansi pemerintah, perusahaan swasta, hingga mitra strategis lainnya. Sejak didirikan pada tahun 2016, PT. Esa Data Teknika telah memantapkan diri sebagai perusahaan yang bergerak di bidang solusi infrastruktur IT dan Data Center yang berfokus pada pelayanan berkualitas tinggi dan inovasi berkelanjutan.

PT. Esa Data Teknika hadir sebagai jawaban atas kebutuhan akan infrastruktur teknologi informasi yang handal dan terpercaya di era transformasi digital yang kian pesat di Indonesia. Dengan dukungan tim muda yang kompeten, berdedikasi tinggi, serta memiliki sertifikasi di bidangnya, PT. Esa Data Teknika terus membuktikan komitmennya dalam membantu perusahaan dan organisasi di seluruh Indonesia membangun fondasi teknologi yang kokoh. Perusahaan ini juga dikenal karena kemampuannya menjangkau wilayah terpencil dan menghadirkan solusi yang tepat guna dalam kondisi dan kebutuhan lapangan yang beragam.

Di tengah dinamika perkembangan teknologi dan persaingan industri yang terbuka, PT. Esa Data Teknika terus melangkah maju dengan mengedepankan integritas, profesionalisme, dan semangat kolaborasi dengan mitra kerja. Perpaduan antara budaya kerja yang adaptif dan layanan yang berorientasi pada kepuasan pelanggan menjadi kunci utama dalam perjalanannya menuju perusahaan penyedia solusi infrastruktur IT terdepan di Indonesia.

* + 1. **Visi dan Misi PT. Esa Data Teknika**
       1. Visi

Menjadi penyedia solusi infrastruktur IT dan Data Center terbaik di Indonesia, yang inovatif, handal, dan terpercaya di semua tingkatan pasar dari kota besar hingga wilayah terpencil.

* + - 1. Misi

1. Memberikan layanan berkualitas tinggi dalam bidang perencanaan, instalasi, dan pengelolaan Data Center, structured cabling, sistem keamanan, HVAC, dan UPS.
2. Menjamin solusi yang efisien dan terjangkau, sesuai anggaran pelanggan, tanpa mengorbankan keandalan teknis.
3. Mengembangkan tim yang profesional dan bersertifikasi, dengan semangat kerja tinggi dan responsif terhadap kebutuhan pelanggan.
4. Menjangkau dan melayani wilayah terpencil, memperluas akses teknologi bagi perusahaan dan organisasi di seluruh Indonesia.
5. Menjalin kemitraan strategis yang etis dan kolaboratif, bersama klien dan stakeholder untuk keberhasilan proyek jangka panjang.
6. Selalu berinovasi dengan teknologi mutakhir, mendukung perkembangan digitalisasi nasional dan transformasi global.
   * 1. **Struktur Organisasi**

Struktur organisasi PT. Esa Data Teknika

* + - 1. Direktur Utama : Robi A. Muhtar
      2. Supervisor : Muhammad Maulana
      3. Keuangan & Administrasi : Reza
      4. Logistik & Administrasi : Ilham Waluyo
      5. Marketing : Robi & Reska Oktaria
      6. Proyek Manajer : Harja Wiharja
      7. Asisten PM & Pelaporan : Nurul Januar
      8. Koordinator Teknisi 1 : Edi Waluyo
      9. Koordinator Teknisi 2 : Denny Roy



GAMBAR: 3.1 Struktur Organisasi PT. Esa Data Teknika

* + 1. **Uraian Tugas, Wewenang dan Tanggung Jawab**

1. Direktur Utama
   * + - 1. Tugas
2. Memimpin dan mengarahkan seluruh operasional perusahaan secara strategis.
   * + - 1. Wewenang

Mengambil keputusan penting terkait keuangan, SDM, dan arah kebijakan perusahaan.

Menyetujui atau menolak rencana dan laporan dari setiap divisi.

* + - * 1. Tanggung Jawab

Menjamin pencapaian visi dan misi perusahaan.

Bertanggung jawab terhadap seluruh kinerja perusahaan.

1. Supervisor
2. Tugas
3. Mengawasi dan mengarahkan pelaksanaan pekerjaan harian tim sesuai standar operasional prosedur (SOP).
4. Menyusun jadwal kerja dan memastikan target tercapai.
5. Wewenang
6. Memberikan instruksi langsung kepada tim kerja.
7. Menegur atau memberi tindakan korektif terhadap pelanggaran disiplin di tingkat bawahannya.
8. Mengusulkan kebutuhan bahan/material, peralatan, atau sumber daya manusia kepada atasan.
9. Menyetujui pekerjaan yang sudah selesai sebelum diserahkan ke klien atau tahap selanjutnya.
10. Menjadi penghubung antara manajemen dan staf pelaksana.
11. Tanggung Jawab
12. Menjamin mutu, ketepatan waktu, dan keselamatan kerja.
13. Melaporkan progres dan kendala kepada atasan.
14. Keuangan & Administrasi
    * + - 1. Tugas

Mengelola keuangan perusahaan termasuk anggaran, pengeluaran, dan pelaporan.

Menangani administrasi umum dan dokumentasi perusahaan.

* + - * 1. Wewenang

Menyusun dan mengontrol anggaran setiap divisi.

Mengelola sistem akuntansi dan laporan keuangan.

* + - * 1. Tanggung Jawab

Menjamin keteraturan administrasi dan pelaporan keuangan yang akurat dan tepat waktu.

1. Logistik & Administrasi
   * + - 1. Tugas

Mengatur pengadaan, penyimpanan, dan distribusi barang serta kelengkapan proyek.

Mendukung administrasi logistik dan kebutuhan operasional lainnya.

* + - * 1. Wewenang

Memilih vendor dan supplier terkait logistik.

Mengelola jadwal pengiriman dan ketersediaan alat/barang.

* + - * 1. Tanggung Jawab

Menjamin kelancaran logistik dan administrasi pendukung proyek.

1. Marketing
   * + - 1. Tugas

Menyusun dan melaksanakan strategi pemasaran produk/jasa perusahaan.

Menjalin hubungan dengan klien dan pihak ketiga.

* + - * 1. Wewenang

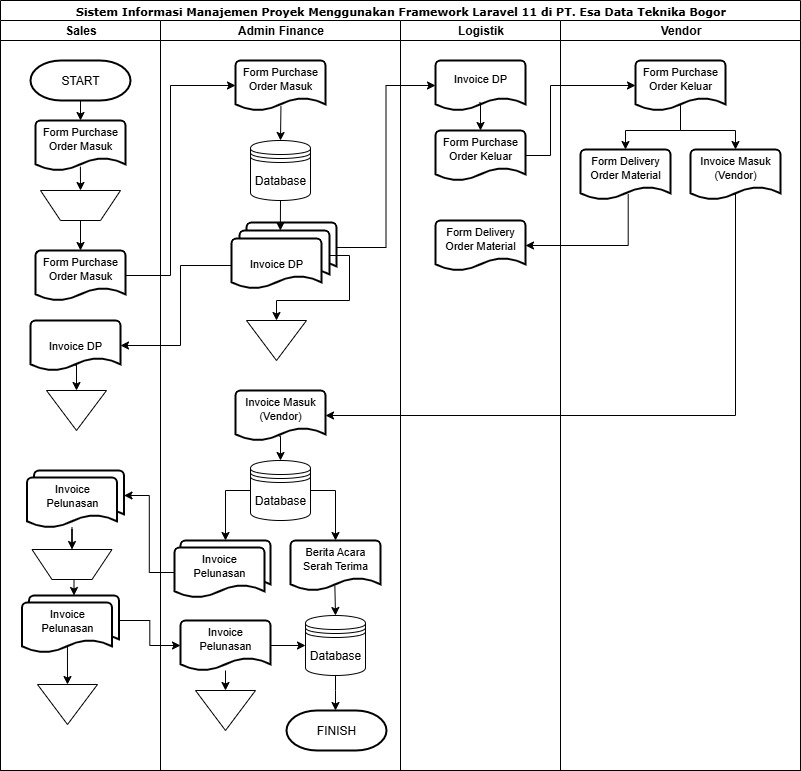
1. Menentukan strategi promosi dan branding.
2. Melakukan negosiasi dengan calon klien.
   * + - 1. Tanggung Jawab

(1) Meningkatkan pendapatan perusahaan melalui pemasaran.

1. Proyek Manajer
   * + - 1. Tugas
2. Merencanakan, mengelola, dan mengevaluasi pelaksanaan proyek.
   * + - 1. Wewenang
3. Mengatur sumber daya, penjadwalan, dan pembagian tugas tim proyek.
4. Mengambil keputusan teknis dan operasional dalam proyek.
   * + - 1. Tanggung Jawab
5. Menjamin proyek berjalan sesuai waktu, kualitas, dan anggaran yang telah ditetapkan
6. Asisten PM & Pelaporan
   * + - 1. Tugas
7. Membantu Project Manager dalam pelaksanaan tugas harian.
8. Menyusun laporan perkembangan proyek secara periodik.
   * + - 1. Wewenang
9. Mengumpulkan data dari tim teknis dan menyusun laporan untuk PM dan stakeholder.
10. Memberikan masukan administratif kepada PM.
    * + - 1. Tanggung Jawab
11. Menyediakan dokumentasi dan pelaporan yang akurat serta mendukung kelancaran proyek.
12. Koordinator Teknisi 1
    * + - 1. Tugas
13. Mengelola tim teknisi dan memantau pelaksanaan pekerjaan di lokasi proyek.
    * + - 1. Wewenang
14. Memberikan instruksi langsung kepada teknisi.
15. Melakukan evaluasi teknis dan pelaporan harian.
    * + - 1. Tanggung Jawab
16. Menjaga kualitas dan efisiensi pelaksanaan tugas teknis di lapangan.
17. Koordinator Teknisi 2
    * + - 1. Tugas
18. Mengelola tim teknisi dan memantau pelaksanaan pekerjaan di lokasi proyek.
    * + - 1. Wewenang
19. Memberikan instruksi langsung kepada teknisi.
20. Melakukan evaluasi teknis dan pelaporan harian.
    * + - 1. Tanggung Jawab
21. Menjaga kualitas dan efisiensi pelaksanaan tugas teknis di lapangan.
    1. **Analisa Sistem Berjalan**

Sistem berjalan manajemen proyek pada PT. Esa Data Teknika :

* 1. *Sales* menyerahkan data proyek dan *purchase order* dari *client* kepada admin finance untuk diinput kedalam sistem.
  2. Admin *finance* membuat *invoice* DP (*Down Payment*)
  3. Admin *logistik* membuat po keluar (kepada vendor) untuk persiapan material
  4. Admin *finance* membuat *invoice* masuk untuk pembayaran kepada vendor atas pembelian material
  5. Admin *logistik* membuat DO (*Delivery Order*) untuk pengiriman material untuk *project* tersebut
  6. Admin *finance* membuat BAST (Berita Acara Serah Terima) ketika pekerjaan sudah selesai.
  7. Admin Finance membuat invoice pelunasan ketika BAST sudah di ttd oleh customer
  8. Admin finance memasukan semua dokumen terkait project ke dalam sistem sesuai dengan nomor project



Gambar 3.2. *Flowchart* Dokumen Sistem Berjalan Manajemen Proyek

* 1. **Analisa Kebutuhan Sistem**

Pengembangan sistem informasi layanan pelanggan berbasis web pada PT. Semangat Pagi Diginet di Bogor ini didasarkan pada analisa terhadap kendala operasional yang dihadapi perusahaan sebagaimana diuraikan pada BAB I. Sistem yang akan dibuat menggunakan framework laravel ini bertujuan untuk mengatasi masalah-masalah tersebut dan memenuhi kebutuhan fungsional serta non-fungsional yang teridentifikasi.

* + 1. Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan fitur dan kemampuan spesifik yang harus dimiliki oleh sistem untuk menjawab permasalahan yang telah diidentifikasi dan mendukung tujuan penelitian:

* + - 1. Pengelolaan Data Proyek Terpusat
         1. Sistem mampu mencatat dan mengelola data proyek (nama proyek, deskripsi, klien, tanggal mulai & selesai, status) secara terpusat.
         2. Admin dapat melakukan input, edit, dan penghapusan data proyek.
         3. Sistem mendukung pencarian dan filter proyek berdasarkan status atau waktu.
      2. Monitoring Progress Proyek
         1. Sistem menampilkan diagram atau indikator visual untuk menunjukkan perkembangan proyek secara real-time.
         2. Admin dapat memperbarui status proyek dan setiap tugas.
      3. Pengelolaan Klien ataupun Vendor
         1. Sistem menyimpan data klien ataupun vendor (nama, perusahaan, kontak, alamat) yang terlibat dalam proyek.
         2. Admin dapat menambah, mengubah, dan mencari data klien.
      4. Manajemen Dokumen Proyek
         1. Sistem mendukung unggah dokumen proyek (proposal, laporan, desain, dsb).
         2. Dokumen dapat dikategorikan berdasarkan proyek terkait.
      5. Hak Akses Pengguna
         1. Sistem memiliki role-based access control (RBAC),
         2. Setiap peran hanya bisa mengakses dan memodifikasi data sesuai wewenangnya.
      6. Pelaporan Proyek
         1. Sistem mampu menghasilkan laporan proyek secara otomatis dalam format PDF atau Excel.
         2. Admin dapat memfilter laporan berdasarkan waktu atau proyek tertentu.
    1. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional berfokus pada kualitas sistem yang dikembangkan untuk menjamin keandalan dan keberlangsungan pemanfaatan sistem di lingkungan kerja:

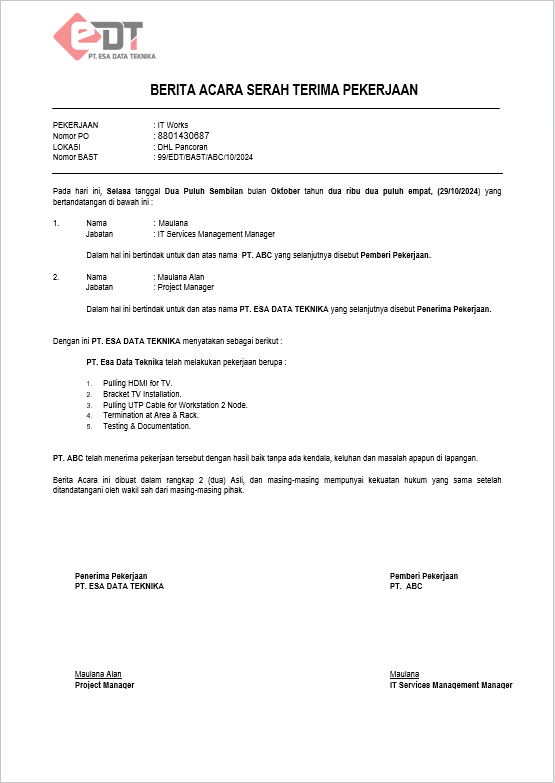
* + - 1. Keamanan (Security)

1. Sistem harus melindungi informasi proyek dan data pengguna dari akses tidak sah.
2. Sistem menggunakan autentikasi login yang aman (enkripsi password, CSRF protection, dll).
3. Setiap akses data dicatat dalam sistem log (activity log).
   * + 1. Kemudahan Penggunaan (Usability)
4. Antarmuka harus intuitif dan mudah dipahami oleh pengguna dari berbagai latar belakang teknis.
5. Tata letak menu dan navigasi dibuat sederhana dan responsif.
   * + 1. Kinerja (Performance)
6. Performa tetap stabil meskipun jumlah data proyek dan pengguna bertambah.
   * + 1. Keandalan (Reliability)
7. Sistem dapat berjalan stabil tanpa crash saat digunakan oleh banyak pengguna.
8. Diperlukan fitur auto-backup data secara berkala.
   * + 1. Skalabilitas (Scalability)
9. Sistem dirancang modular sehingga memungkinkan pengembangan fitur baru seperti API mobile, integrasi payment gateway, dan lainnya.
   * + 1. Pemeliharaan (Maintainability)
10. Sistem dikembangkan menggunakan Laravel untuk mendukung struktur kode yang rapi dan mudah dikelola.
11. Dokumentasi kode dan struktur sistem disiapkan dengan baik.
    * + 1. Dokumentasi Sistem
12. Sistem didokumentasikan menggunakan diagram UML (Use Case, Activity, Sequence) dan flowchart untuk mempermudah pengembangan lanjutan dan pemeliharaan.
    1. **Dokumen Input & Output**
       1. Dokumentasi *input*



Gambar 3.3. Dokumentasi input formulir pemesanan proyek

* + 1. Dokumentasi *output*



Gambar 3.4. Dokumentasi *output* formulir berita acara serah terima pekerjaan

* 1. **Metode Pengumpulan Data**
     1. **Studi Lapangan**
        1. Observasi

Metode yang dilakukan untuk mengidentifikasi data yang dilakukan secara sistematis baik dengan cara memperhatikan secara langsung atau tidak langsung objek yang diteliti serta mengambil data visual sesuai kebutuhan penelitian sehingga tidak ada data yang terlewat dalam pengembangan sistem informasi manajemen proyek pada PT. Esa Data Teknika Bogor menggunakan Laravel.

* + - 1. Wawancara

Merupakan teknik pengumpulan data, berita, fakta dan informasi dilapangan yang prosesnya bisa dilakukan dengan cara menanyakan langsung kepada pihak-pihak yang bisa memberikan informasi mengenai masalah yang sedang diteliti atau secara tidak langsung seperti melakukan telepon, email dan surat (wawancara tertulis).

* + 1. **Studi Pustaka**

Usaha yang dilakukan untuk menghimpun informasi yang relevan dengan topik dan masalah yang akan atau sedang diteliti. Informasi itu dapat diperoleh dari buku-buku ilmiah, laporan penelitian, karangan-karangan ilmiah, tesis dan disertasi, peraturan-peraturan, ketetapan-ketetapan, buku-buku tahunan, ensiklopedia dan sumber-sumber tertulis baik tercetak maupun elektronik serta internet.

* + 1. **Metode Framework**

Metode Framework pada Website adalah pendekatan dalam pengembangan website dengan menggunakan kerangka kerja (framework) yang telah disiapkan sebelumnya. Framework menyediakan struktur dan komponen siap pakai sehingga pengembang tidak perlu membangun semuanya dari nol. Ini mempercepat pengembangan, meningkatkan keamanan, dan menjaga standar penulisan kode.

* 1. **Struktur Pengkodean**

Struktur pengkodean yang akan digunakan untuk sistem yang diajukan adalah sebagai berikut:

TABEL: 3.1. Struktur Pengkodean

|  |  |  |  |
| --- | --- | --- | --- |
| Database | Objek | Kode | Keterangan |
| *Project* | *Project* | EP2507-0001 | EP – Kode untuk Project  2507 – Tahun dan Bulan  0001 – Urutan ke-1 |
| *Purchase Orders* | PO Out | POO2506-0001 | POO – Kode untuk Purchase Order Out  25 – Tahun 2025  06 – Bulan Juni  0001 – Urutan ke-1 |
| PO In | POI2506-0001 | POI – Kode untuk Purchase Order In  25 – Tahun 2025  06 – Bulan Juni  0001 – Urutan ke-1 |
| *Invoices* | *Invoices* | INV\_202507060001 | INV – Kode untuk Invoice  20250706 – Tanggal 6 Juli 2025  0001 – Urutan ke-1 pada tanggal tersebut |
| *Third Parties* | *Vendor* | EV250500001 | EV – Kode untuk Vendor  25 – Tahun 2025  05 – Bulan Mei  00001 – Urutan ke-1 |
| *Client* | EC250600001 | EC – Kode untuk Client  25 – Tahun 2025  06 – Bulan Juni  00001 – Urutan ke-1 |
| *Delivery Orders* | *Delivery* *Orders* | 01/EDT/DO/Maulana/07/2025 | 01 – Urutan ke-1  EDT – Kode internal atau unit/departemen pengiriman  DO – Delivery Order  Maulana – Nama pengirim atau penanggung jawab  07 – Bulan Juli  2025 – Tahun |

* 1. **Perencanaan Penelitian**

1. **Prosedur Sistem Usulan**
   * 1. **Sistem *Login***

Admin dan karyawan perusahaan melakukan login ke sistem dengan memasukkan username dan password. Sistem akan memverifikasi kredensial tersebut; jika valid, pengguna diarahkan ke dashboard sesuai perannya. Jika tidak valid, sistem akan menampilkan pesan kesalahan untuk mencegah akses tidak sah.

* + 1. **Pengelolaan Data Perusahaan**

Admin dapat mencatat, mengedit, dan mencari data perusahaan klien (nama, alamat, kontak, bidang usaha) secara terpusat. Validasi input dilakukan untuk menghindari duplikasi dan memastikan informasi perusahaan tetap konsisten dan akurat.

* + 1. **Pengelolaan Data *Third Party***

Admin dapat mengelola data vendor/third party yang terlibat dalam proyek, termasuk nama perusahaan, kontak, dan jenis layanan yang diberikan. Data ini berguna dalam proses pemilihan vendor dan manajemen hubungan kerja sama.

* + 1. **Pengelolaan *Purchase Order***

Admin dapat membuat dan mengelola purchase order (PO) untuk kebutuhan proyek. PO mencakup informasi barang/jasa, vendor, harga, dan termin pembayaran. Sistem juga akan menghasilkan dokumen PO secara otomatis untuk didistribusikan kepada vendor.

* + 1. **Monitoring Status *Purchase Order***

Sistem memungkinkan admin untuk memantau status setiap purchase order, seperti *draft*, *submitted*, *approved*, *rejected*, atau *completed*. Fitur ini membantu manajemen proyek dalam pelacakan dan kontrol pengadaan barang/jasa.

* + 1. **Pengelolaan *Delivery Order***

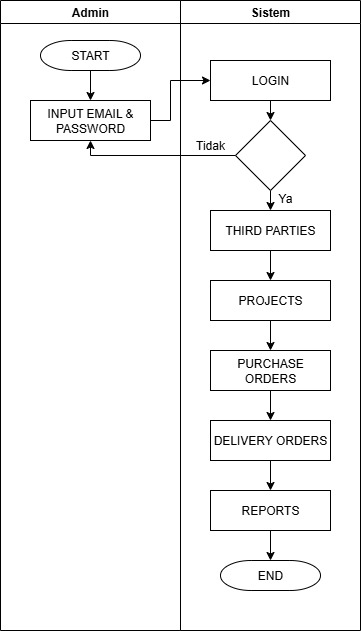
Admin dapat mengelola pengiriman barang/jasa melalui modul delivery order (DO). DO dapat ditautkan ke PO yang relevan dan mencakup informasi detail pengiriman seperti tanggal kirim, status barang, dan penerima. Sistem juga mencatat bukti pengiriman untuk dokumentasi proyek.

* + 1. **Pengelolaan Pengguna dan Hak Akses**

Sistem menerapkan hak akses berdasarkan peran (admin, karyawan perusahaan). Admin memiliki akses penuh terhadap semua modul, sedangkan karyawan perusahaan hanya dapat mengakses fitur tertentu sesuai dengan kebutuhannya. Ini menjaga keamanan dan integritas data.

* + 1. **Logout**

Fitur logout memastikan setiap pengguna dapat keluar dari sistem secara aman. Ini mencegah penggunaan tidak sah pada perangkat bersama serta melindungi data sensitif dari pihak yang tidak berkepentingan.



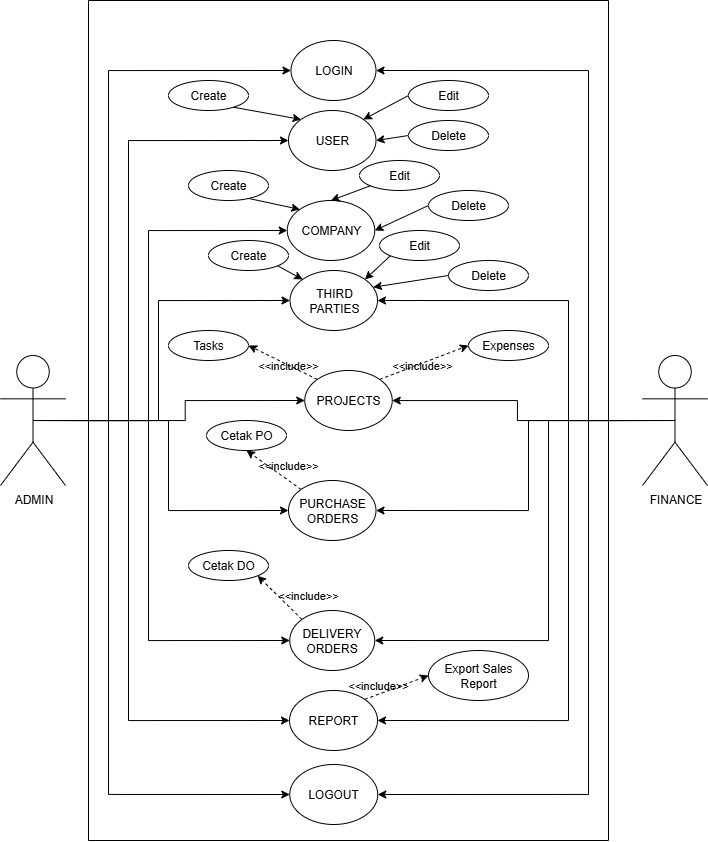
GAMBAR: 3.5. Sistem Usulan

1. **Prosedur Sistem Usulan Menggunakan UML**

Alur prosedur sistem usulan berbasiis web pada layanan Manajemen Proyek PT. Esa Data Teknika digambarkan dengan menggunakan diagram UML.

* + 1. UML (*Unified Modelling Language*)
       1. *Use Case* Diagram

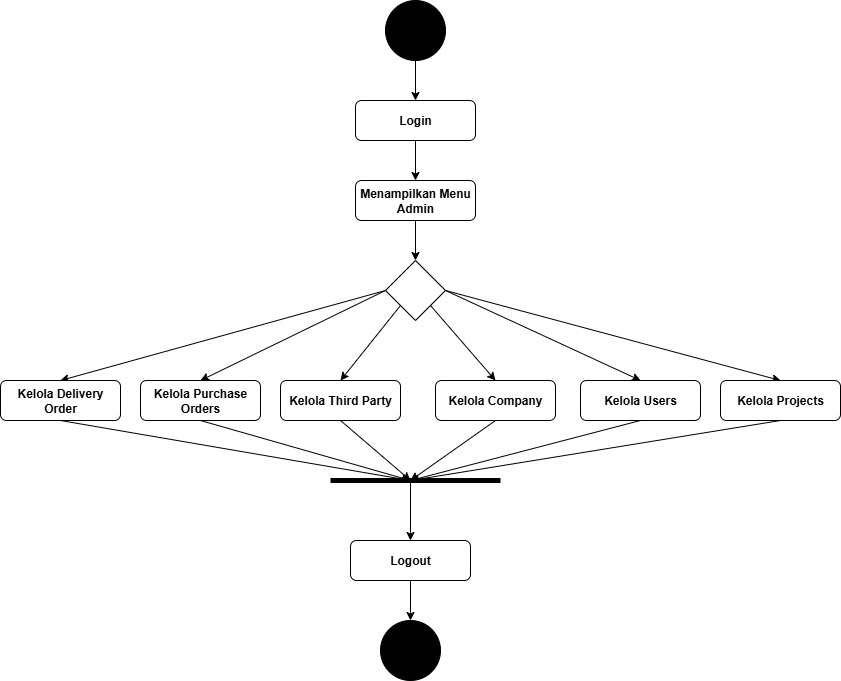
Berikut adalah gambaran Use Case Diagram sistem informasi layanan pelanggan dapat dilihat dibawah ini:



GAMBAR: 3.6. *Use Case* Sistem Manajemen Proyek

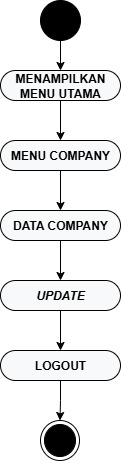
* + - 1. *Activity* Diagram

Berikut gambaran proses *activity* diagram dibawah sebagai berikut:

* + - * 1. *Activity* diagram *authentication*

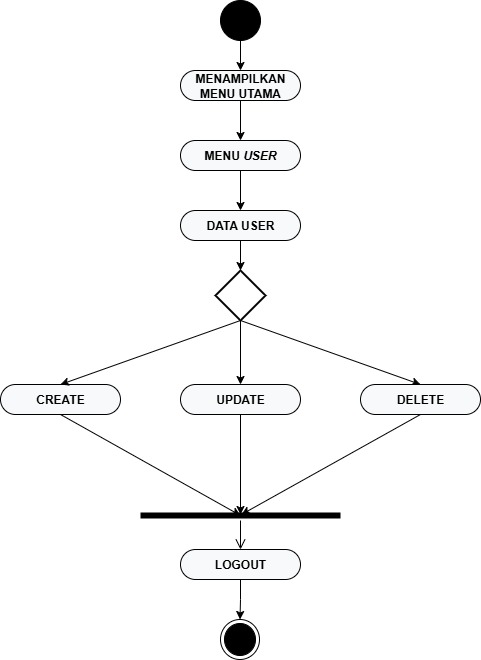
GAMBAR: 3.7. *Activity* Diagram *Login*

* + - * 1. *Activity* diagram *company*



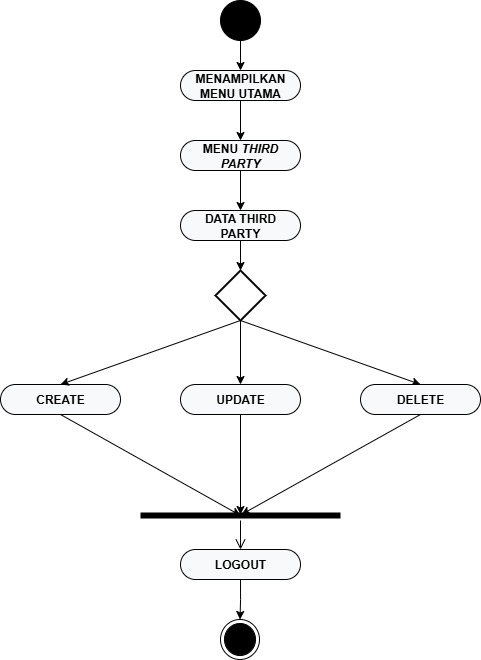
GAMBAR: 3.8. *Activity* Diagram *Company*

* + - * 1. *Activity* diagram *users*



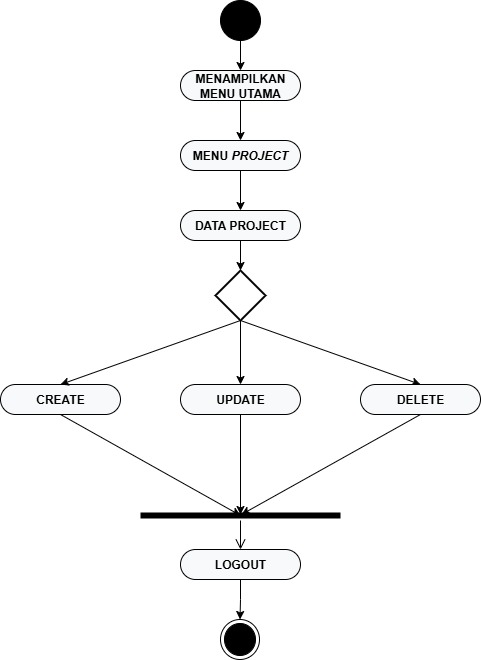
GAMBAR: 3.9. *Activity* Diagram *Users*

* + - * 1. *Activity* diagram *third party*

*.* 

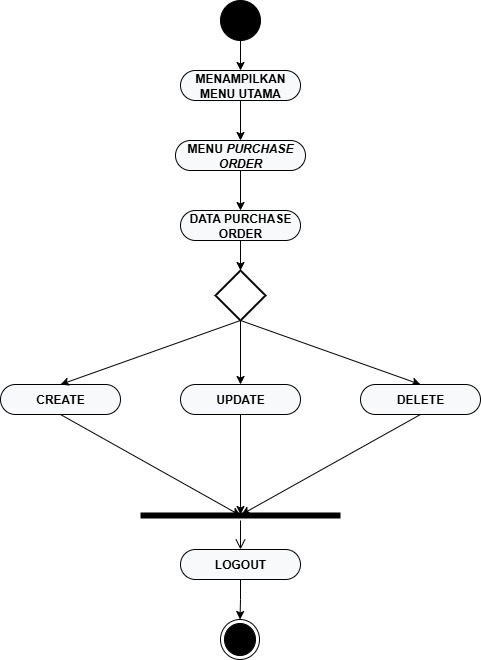
GAMBAR: 3.10. *Activity* Diagram *Third Party*

* + - * 1. *Activity* diagram *projects*



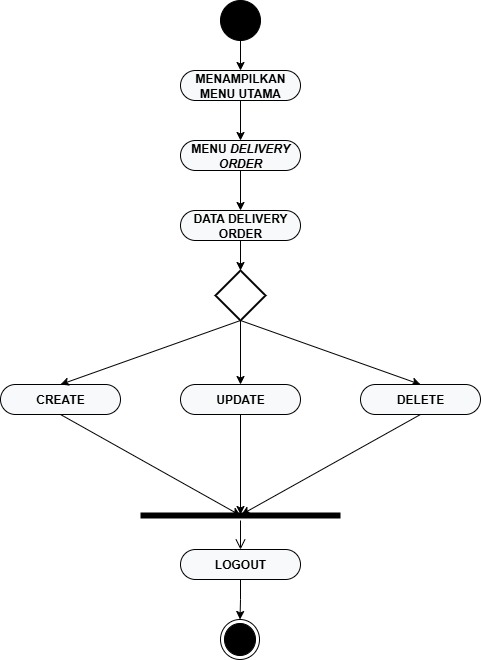
GAMBAR: 3.11. *Activity* Diagram *Project*

* + - * 1. *Activity* diagram *purchase order*



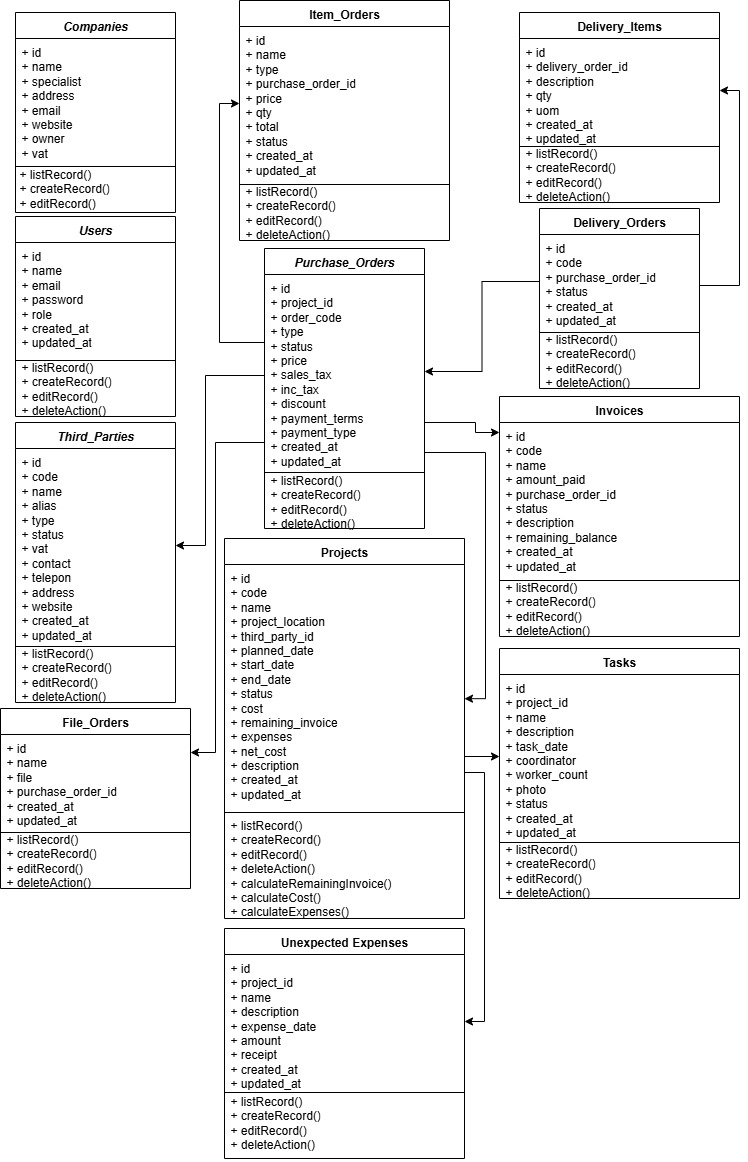
GAMBAR: 3.12. *Activity* Diagram *Purchase Order*

* + - * 1. *Activity* diagram *delivery order*



GAMBAR: 3.13. *Activity* Diagram *Delivery Order*

* + - 1. *Class* Diagram

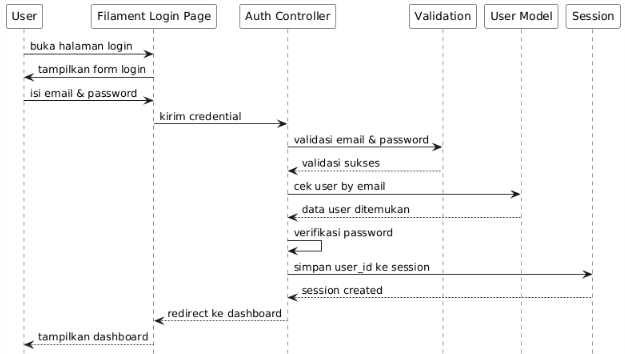


GAMBAR: 3.14 *Class* Diagram PT. Esa Data Teknika

* + - 1. *Sequence* Diagram

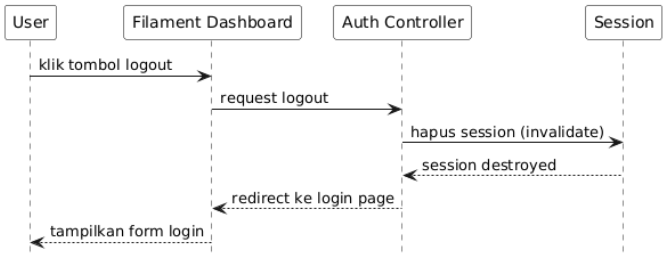
Berikut gambaran proses *sequence* diagram:

* + - * 1. *Sequence* diagram proses login



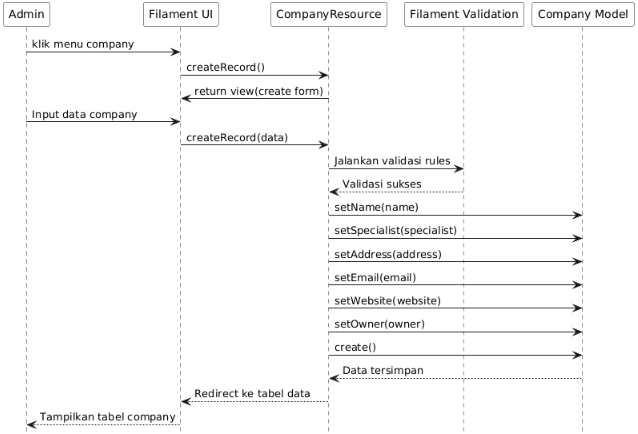
GAMBAR: 3.15. *Sequence* Diagram proses login

* + - * 1. *Sequence* diagram proses logout

**

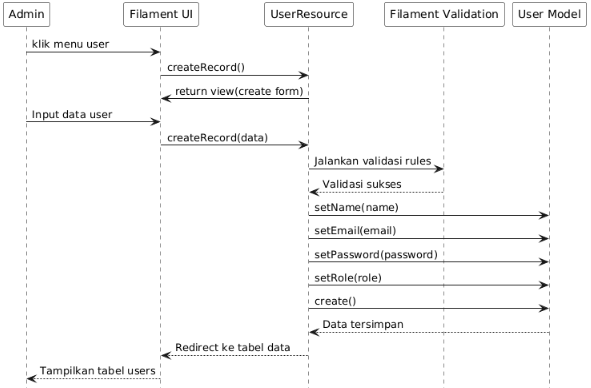
GAMBAR: 3.16. *Sequence* Diagram proses logout

* + - * 1. *Sequence* diagram proses mengelola data *company.*

**

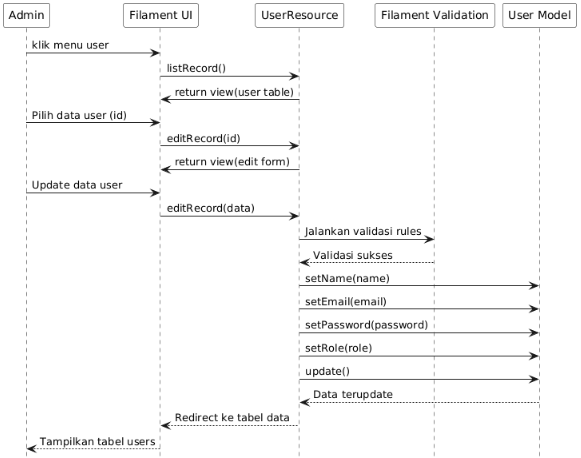
GAMBAR: 3.17. *Sequence* Diagram mengelola data *company*

* + - * 1. *Sequence* diagram proses tambah data *users*

**

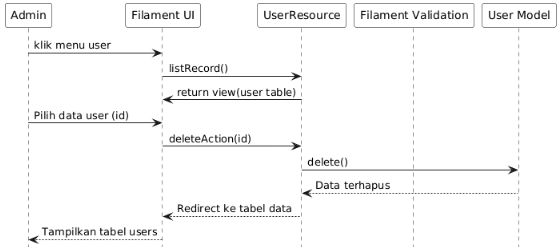
GAMBAR: 3.18. *Sequence* Diagram proses tambah data *users*

* + - * 1. *Sequence* diagram proses edit data *users*

**

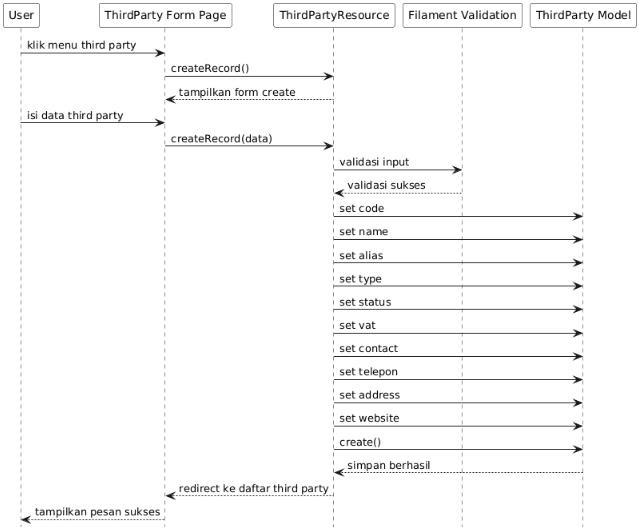
GAMBAR: 3.19. *Sequence* Diagram proses edit data *users*

* + - * 1. *Sequence* diagram proses hapus data *users*

**

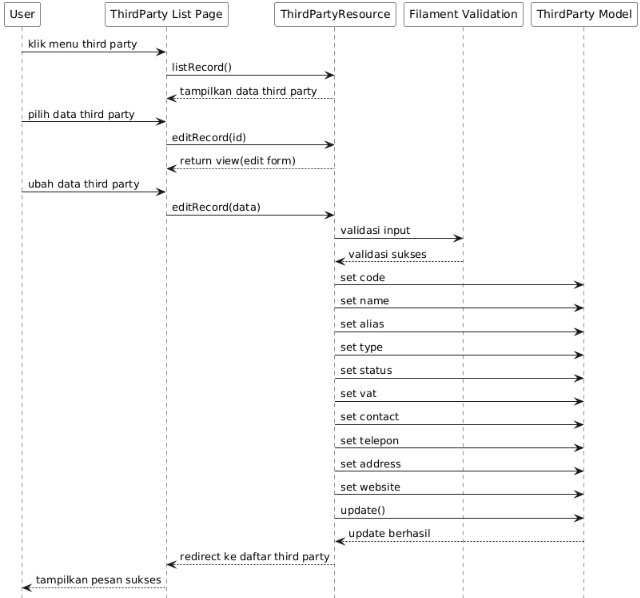
GAMBAR: 3.20. *Sequence* Diagram hapus data *users*

* + - * 1. *Sequence* diagram proses tambah data *third party*



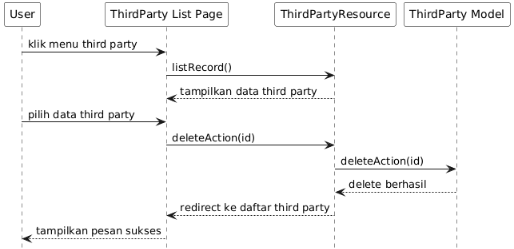
GAMBAR: 3.21. *Sequence* Diagram tambah data *third party*

* + - * 1. *Sequence* diagram proses edit data *third party*

**

*GAMBAR:* 3*.*22*. Sequence Diagram* edit data *third party*

* + - * 1. *Sequence* diagram proses hapus data *third party*

**

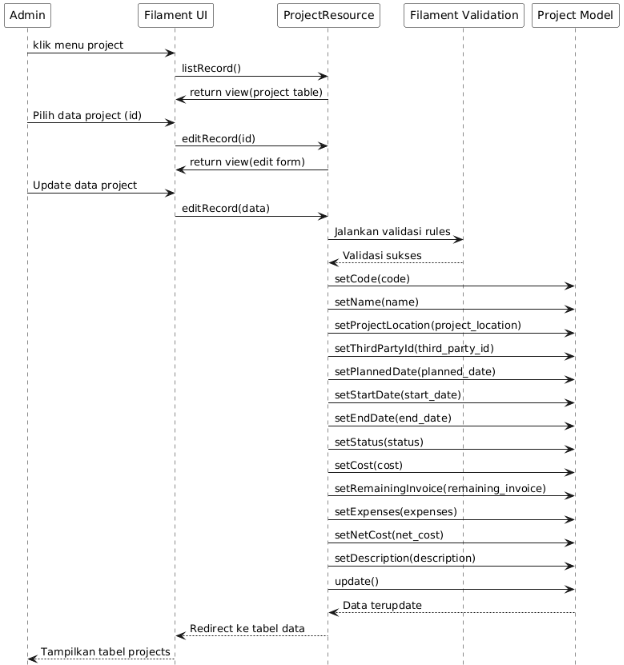
GAMBAR: 3.23. *Sequence Diagram* hapusdata *third party*

* + - * 1. *Sequence* diagram proses tambah data *project*

**

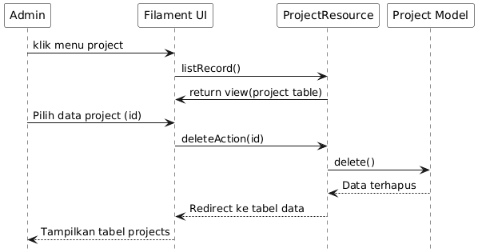
GAMBAR: 3.24. *Sequence* Diagram tambah data *project*

* + - * 1. *Sequence* diagram proses *edit* data *project*

**

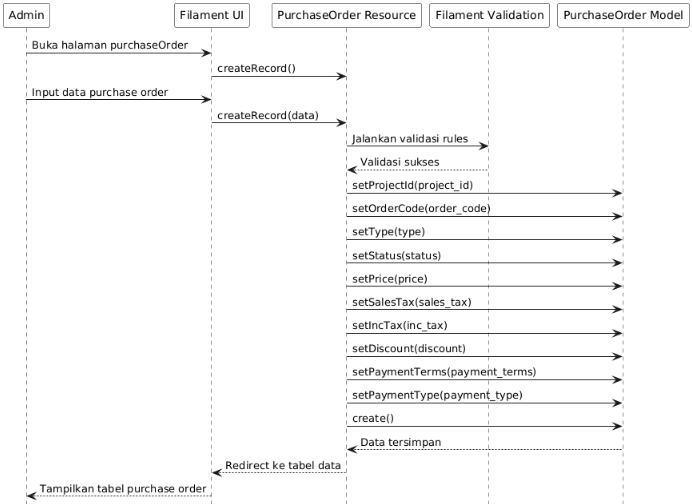
*GAMBAR: 3.25. Sequence Diagram* edit *data project*

* + - * 1. *Sequence* diagram proses hapus data *project*

**

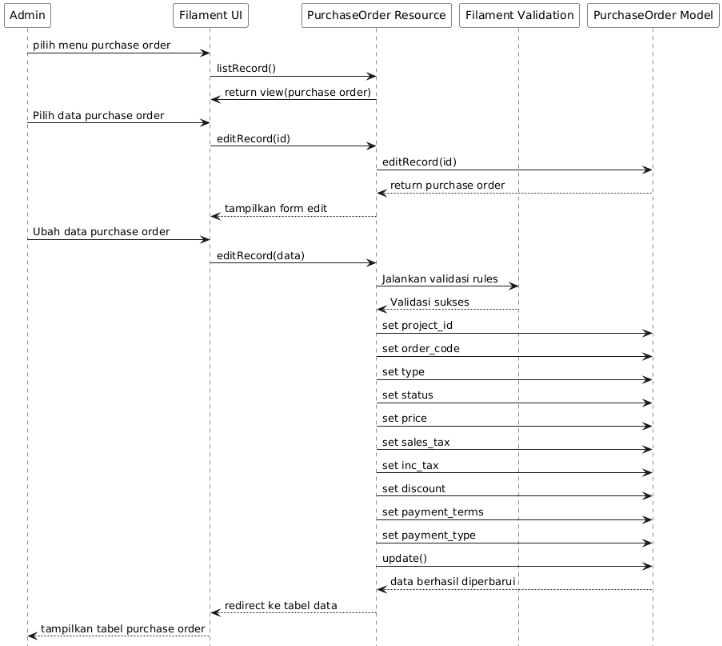
*GAMBAR: 3.26. Sequence Diagram* hapus *data project*

* + - * 1. *Sequence* diagram proses tambah data *purchase order*

**

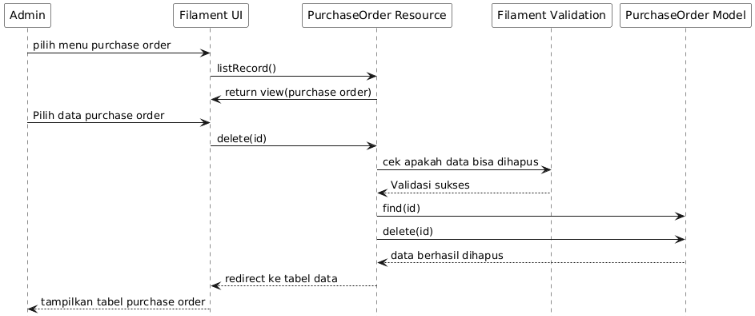
GAMBAR: 3.27. *Sequence* Diagram tambah data *purchase order*

* + - * 1. *Sequence* diagram proses edit data *purchase order*

**

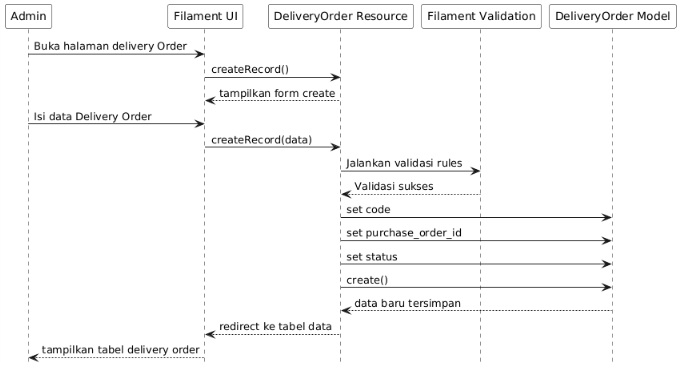
*GAMBAR: 3.28. Sequence Diagram* edit data *purchase order*

* + - * 1. *Sequence* diagram proses hapus data *purchase order*

**

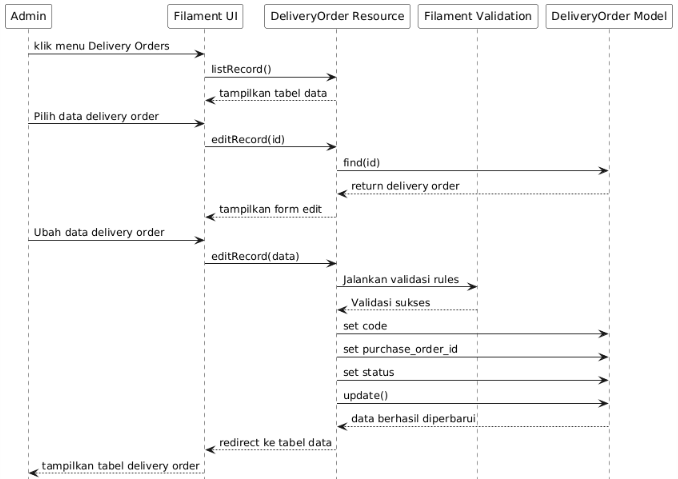
*GAMBAR: 3.29. Sequence Diagram* hapus data *purchase order*

* + - * 1. *Sequence* diagram proses tambah data *delivery order*

**

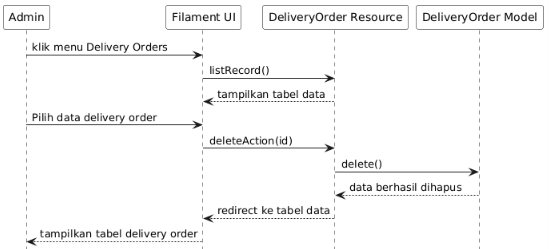
GAMBAR: 3.30. *Sequence* Diagram tambah data *delivery order*

* + - * 1. *Sequence* diagram proses *edit* data *delivery order*

**

*GAMBAR: 3.31. Sequence Diagram* edit *data delivery order*

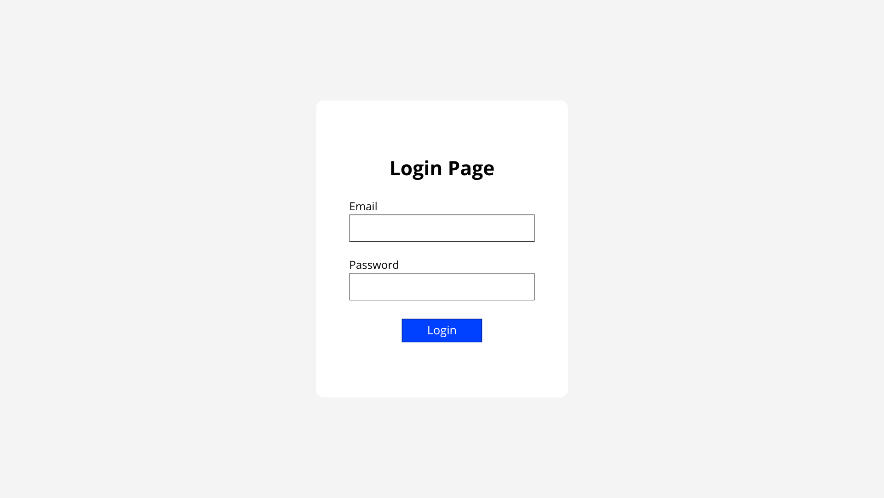
* + - * 1. *Sequence* diagram proses hapus data *delivery order*

**

*GAMBAR: 3.32. Sequence Diagram* hapus *data delivery order*

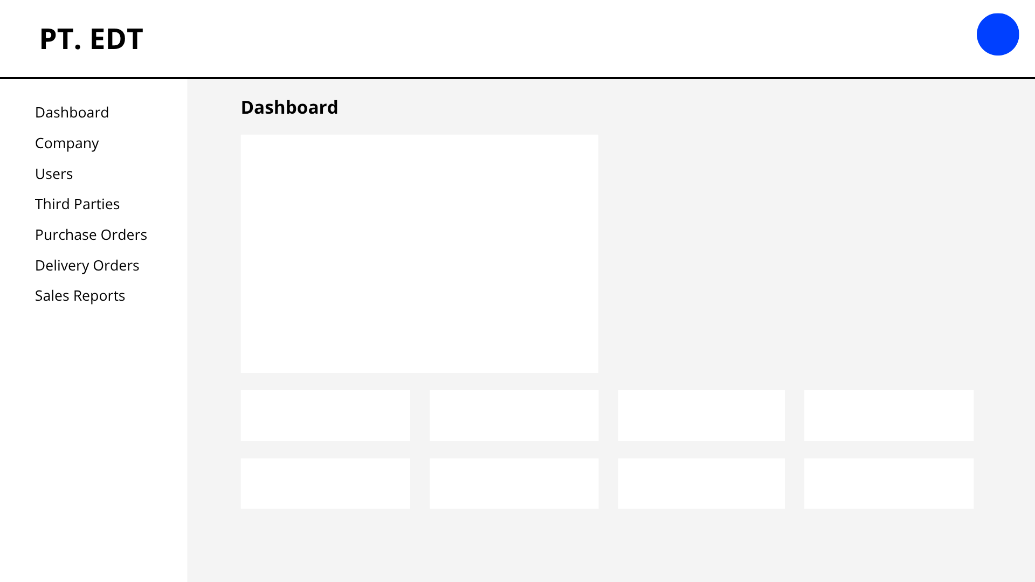
1. **Perancangan *Interface***
   * 1. *Form Login*

*Form login* ini dilakukan untuk *login* admin sebelum mengoperasikan sistem informasi manajemen proyek. Seorang admin harus memasukkan email dan password agar dapat mengoperasikan sistem informasi manajemen proyek.

  
GAMBAR: 3.33. Rancangan *form login*

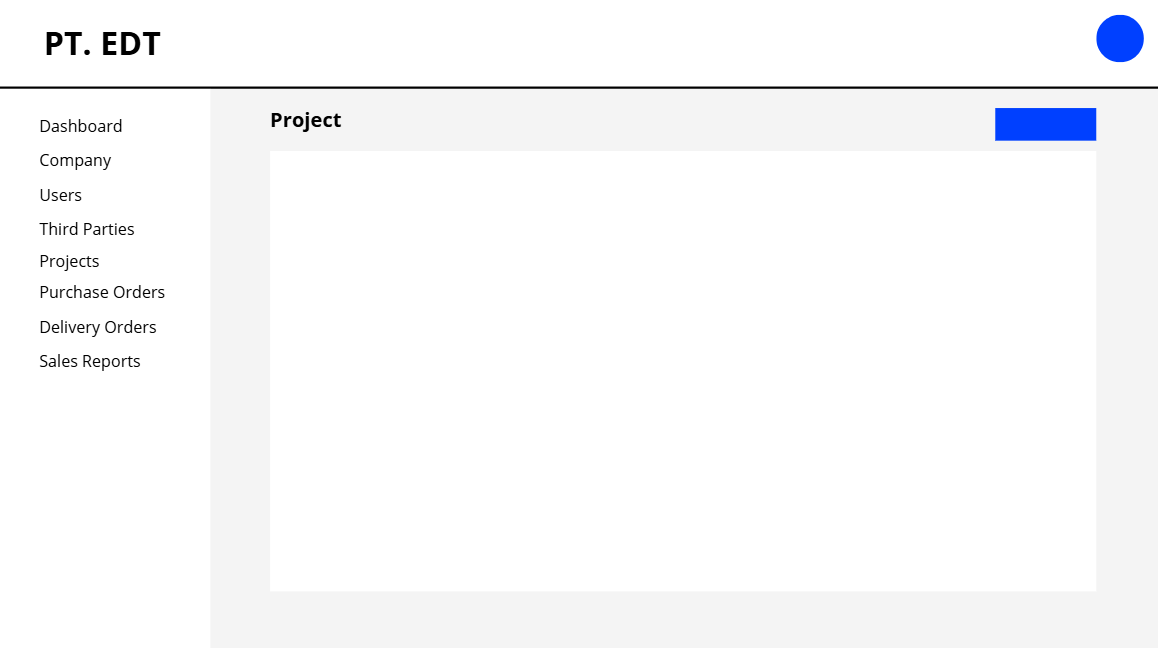
* + 1. *Dashboard*

*Dashboard* digunakan sebagai *form* *navigator* untuk menuju ke form lainnya.



GAMBAR: 3.34. Rancangan *dashboard*

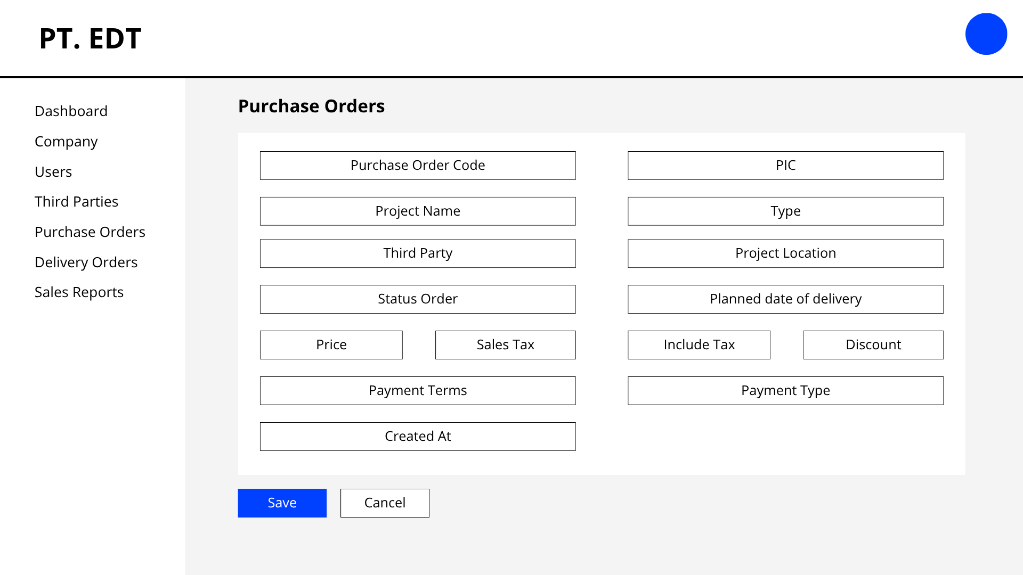
* + 1. *Form project*



GAMBAR: 3.35. Rancangan *form project*

* + 1. *Form purchase order*

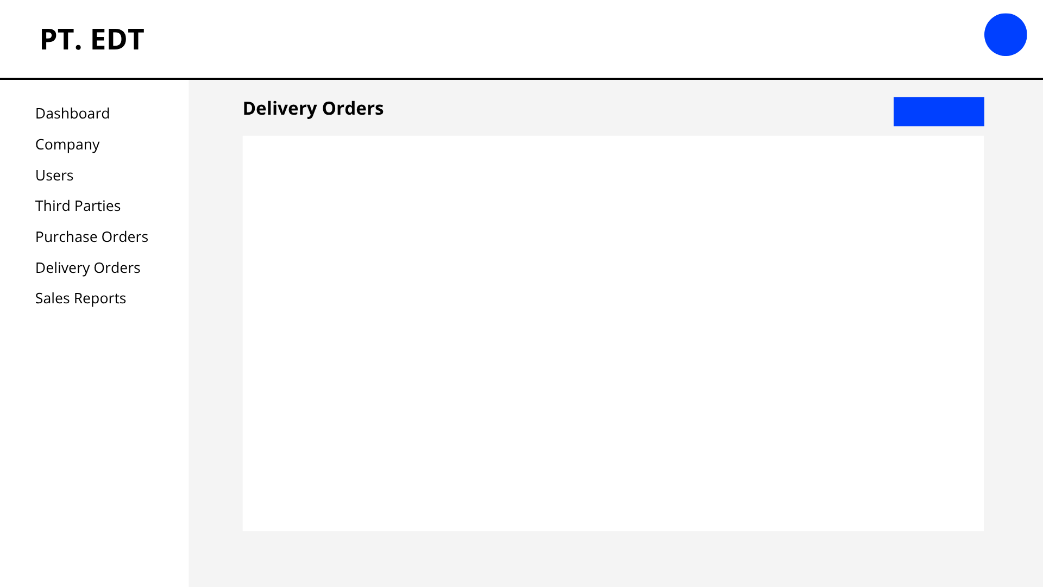
*Form purchase order* adalah formulir yang digunakan oleh admin untuk mendata setiap proyek pada *purchase order.*



GAMBAR: 3.36. Rancangan *form purchase order*

* + 1. *Delivery order*

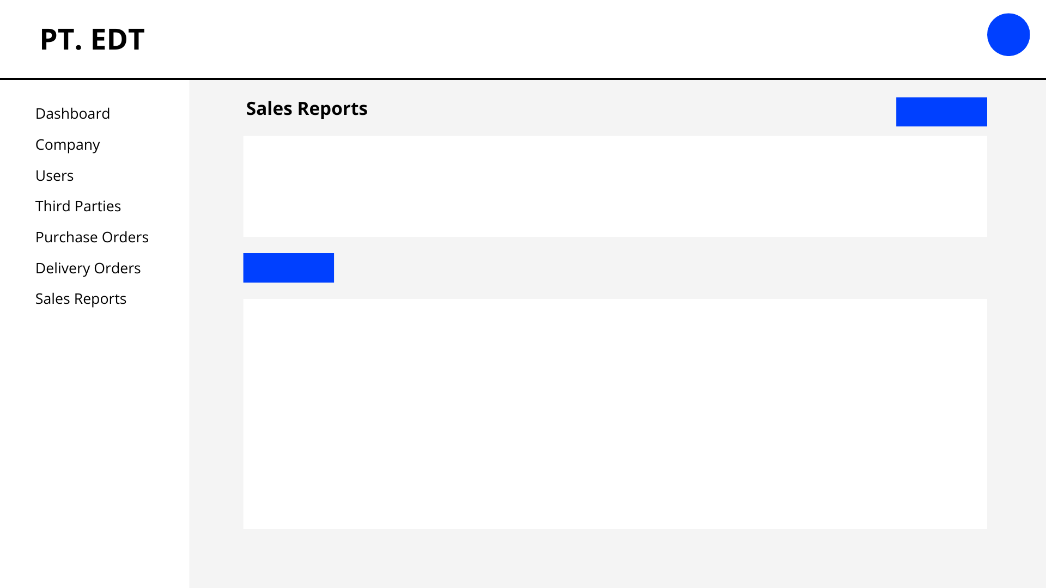
Halaman *delivery order* digunakan oleh admin untuk mendata *delivery order* terhadap setiap *item* pada *purchase order* (proyek).



GAMBAR: 3.37. Rancangan halaman *delivery order*

* + 1. *Sales Report*

Halaman *sales report* digunakan olehbagian *finance* untuk memastikan semua transaksi berjalan dengan seharusnya yang bertujuan sebagai bahan pelaporan proyek disetiap waktu yang telah ditentukan.



GAMBAR: 3.38. Rancangan halaman *sales reports*

* 1. **Rencana Waktu dan Lokasi Penelitian**

1. Lokasi penelitian di PT. Esa Data Teknika yang berlokasi di Cileungsi. Jl. Alternative Cibubur No. 60, Kabupaten Bogor, Jawa Barat 16820
2. Rencana waktu penelitian terhitung 6 bulan terhitung setelah disahkan judul proposal.

Jadwal kegiatan :

TABEL: 3.2. Rencana Kegiatan Penelitian

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No | Rencana  Kegiatan | November  2024 | | | | Desember  2024 | | | | Januari  2025 | | | | Februari s.d Juli 2025 | | | | Agustus 2025 | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | Pencarian Bahan Penelitian |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Pembuatan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Seminar Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Revisi Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Perancangan Sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Uji Coba dan implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Monitoring dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Sidang Yudisium |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Revisi Hasil Sidang |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Pengumpulan Skripsi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**BAB IV**

**PEMBAHASAN DAN HASIL**

* 1. **Pembahasan**

1. **Kebutuhan perangkat keras dan perangkat lunak**
   * 1. Perangkat keras

Spesifikasi perangkat keras (*hardware*) di bawah ini dapat menjadi acuan untuk implementasi sistem informasi manajemen proyek pada PT. Esa Data Teknika.

1. Laptop / komputer processor intel core I3 (generasi ke-5 atau diatasnya) atau AMD Ryzen 3.
2. Ram minimal 8 (disarankan 16GB untuk kinerja lebih lancar).
3. Penyimpanan SSD 256GB (untuk kecepatan akses database dan kompilasi kode).
4. Sistem Operasi Windows 10/11 (64-bit), Linux, atau macOS.
   * 1. Perangkat lunak

Untuk mendukung sistem yang diusulkan berjalan dengan optimal, dibutuhkan *software* pengelolaan data yang digunakan untuk mendukung pembuatan program sistem informasi manajemen proyek pada PT. Esa Data Teknika sebagai berikut:

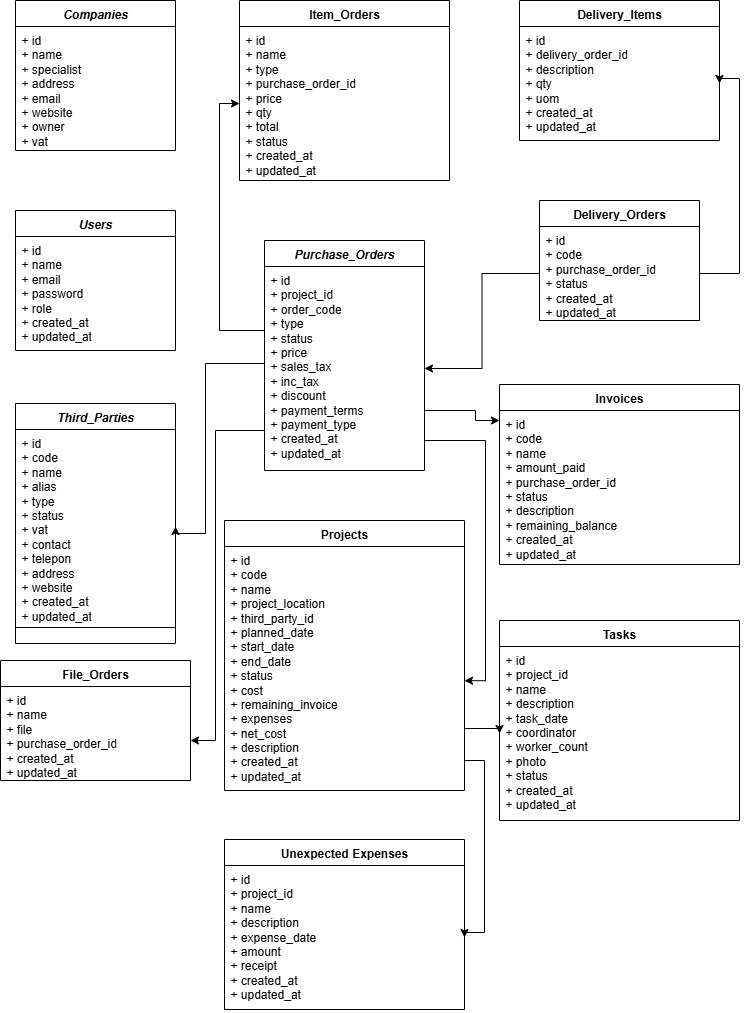
1. XAMPP
2. Composer
3. Laravel
4. Browser
5. Database mengunakan MySQL
   * 1. Indentifikasi sistem

Pembangunan Sistem Informasi Manajemen Proyek pada PT. Esa Data Teknika difokuskan untuk mendukung kebutuhan perusahaan, khususnya bagian keuangan (*finance*), dalam mengelola data proyek secara lebih efisien, akurat, dan terdokumentasi. Sistem ini mencakup fitur pengelolaan data perusahaan, vendor (*third party*), pembuatan dan pelacakan *Purchase* *Order* (PO) serta *Delivery Order* (DO), serta monitoring status proyek secara real-time.

Dengan antarmuka sederhana dan sistem hak akses berbasis peran, informasi sensitif dapat diamankan dan hanya dapat diakses oleh pihak yang berwenang. Sistem ini diharapkan mampu mempercepat alur kerja administrasi proyek, meminimalkan kesalahan pencatatan manual, dan mendukung pengambilan keputusan keuangan dengan lebih tepat dan cepat.

1. **Hubungan antar tabel**

Hubungan antar table menggambarkan hubungan tabel-table yang terdapat pada database dimana di dalam tabel tersebut terdapat field kunci (primary key) dan terdapat kunci tamu (foreign key).



GAMBAR: 4.1. Hubungan antar tabel

1. **Struktur database/file**
   * 1. Tabel *Users*

Tabel 4.1: Tabel *Users*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | name | VARCHAR | 35 |
| 3 | Unique | email | VARCHAR | 35 |
| 4 | - | password | VARCHAR | 75 |
| 5 | - | role | ENUM | - |
| 6 | - | remember\_token | VARCHAR | 75 |
| 7 | - | created\_at | TIMESTAMP | - |
| 8 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Companies*

Tabel 4.2: Tabel *Companies*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | name | VARCHAR | 75 |
| 3 | - | specialist | VARCHAR | 100 |
| 4 | - | address | VARCHAR | 100 |
| 5 | - | email | VARCHAR | 50 |
| 6 | - | website | VARCHAR | 50 |
| 7 | - | owner | VARCHAR | 50 |
| 8 | - | vat | VARCHAR | 25 |
| 9 | - | created\_at | TIMESTAMP | - |
| 10 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Third Parties*

Tabel 4.3: Tabel *Third Party*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | name | VARCHAR | 50 |
| 3 | - | specialist | VARCHAR | 100 |
| 4 | - | address | VARCHAR | 100 |
| 5 | - | email | VARCHAR | 50 |
| 6 | - | website | VARCHAR | 50 |
| 7 | - | owner | VARCHAR | 50 |
| 8 | - | vat | VARCHAR | 25 |
| 9 | - | created\_at | TIMESTAMP | - |
| 10 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Projects*

Tabel 4.4: Tabel *Projects*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | Unique | code | VARCHAR | 25 |
| 3 | - | name | VARCHAR | 50 |
| 4 | - | project\_location | VARCHAR | 100 |
| 5 | Foreign Key | third\_party\_id | INT | - |
| 6 | - | planned\_date | INT | - |
| 7 | - | start\_date | DATE | - |
| 8 | - | end\_date | DATE | - |
| 9 | - | status | ENUM | - |
| 10 | - | cost | INT | - |
| 11 | - | remaining\_invoice | INT | - |
| 12 | - | expenses | INT | - |
| 13 | - | net\_cost | INT | - |
| 14 | - | description | VARCHAR | - |
| 15 | - | created\_at | TIMESTAMP | - |
| 16 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Purchase Orders*

Tabel 4.5: Tabel *Purchase Orders*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | Unique | order\_code | VARCHAR | 25 |
| 3 | Foreign Key | project\_id | INT | - |
| 4 | - | type | ENUM | - |
| 5 | - | status | ENUM | - |
| 6 | - | price | INT | - |
| 7 | - | sales\_tax | INT | - |
| 8 | - | inc\_tax | INT | - |
| 9 | - | discount | INT | - |
| 10 | - | payment\_terms | ENUM | - |
| 11 | - | payment\_type | ENUM | - |
| 12 | - | created\_at | TIMESTAMP | - |
| 13 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Item Orders*

Tabel 4.6: Tabel *Item Orders*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | name | VARCHAR | 50 |
| 3 | - | type | ENUM | - |
| 4 | Foreign Key | purchase\_order\_id | INT | - |
| 5 | - | price | INT | - |
| 6 | - | qty | INT | - |
| 7 | - | total | INT | - |
| 8 | - | status | ENUM | - |
| 9 | - | created\_at | TIMESTAMP | - |
| 10 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *File Orders*

Tabel 4.7S: Tabel *Item Orders*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | name | VARCHAR | 75 |
| 3 | - | file | VARCHAR | 255 |
| 4 | Foreign Key | purchase\_order\_id | INT | - |
| 5 | - | created\_at | TIMESTAMP | - |
| 6 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Invoices*

Tabel 4.8: Tabel *Invoices*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | code | VARCHAR | 25 |
| 3 | - | name | VARCHAR | 50 |
| 4 | - | amount\_paid | INT | - |
| 5 | Foreign Key | purchase\_order\_id | BIGINT | 20 |
| 6 | - | status | ENUM | - |
| 7 | - | description | VARCHAR | 255 |
| 8 | - | remaining\_balance | INT | - |
| 9 | - | created\_at | TIMESTAMP | - |
| 10 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Delivery Orders*

Tabel 4.9: Tabel *Delivery Orders*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | - | code | VARCHAR | 25 |
| 3 | - | purchase\_order\_id | INT | - |
| 4 | - | status | ENUM | - |
| 5 | - | created\_at | TIMESTAMP | - |
| 6 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Delivery Items*

Tabel 4.10: Tabel *Delivery Items*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | Foreign Key | delivery\_order\_id | BIGINT | 20 |
| 3 | - | description | TEXT | - |
| 4 | - | qty | INT | - |
| 5 | - | uom | VARCHAR | 20 |
| 6 | - | created\_at | TIMESTAMP | - |
| 7 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Unexpected Expenses*

Tabel 4.11: Tabel *Unexpected Expenses*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | Foreign Key | project\_id | BIGINT | 20 |
| 3 | - | name | VARCHAR | 100 |
| 4 | - | description | TEXT | - |
| 5 | - | expense\_date | DATE | - |
| 6 | - | amount | INT | - |
| 7 | - | receipt | VARCHAR | 255 |
| 8 | - | created\_at | TIMESTAMP | - |
| 9 | - | updated\_at | TIMESTAMP | - |

* + 1. Tabel *Tasks*

Tabel 4.12: Tabel *Tasks*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Constraint** | **Field** | **Type** | **Width** |
| 1 | Primary Key | id | BIGINT | 20 |
| 2 | Foreign Key | project\_id | BIGINT | 20 |
| 3 | - | name | TEXT | - |
| 4 | - | Description | INT | - |
| 5 | - | task\_date | VARCHAR | 20 |
| 6 | - | coordinator | VARCHAR | 50 |
| 7 | - | worker\_count | INT | - |
| 8 | - | photo | VARCHAR | 255 |
| 9 | - | status | ENUM | - |
| 10 | - | created\_at | TIMESTAMP | - |
| 11 | - | updated\_at | TIMESTAMP | - |

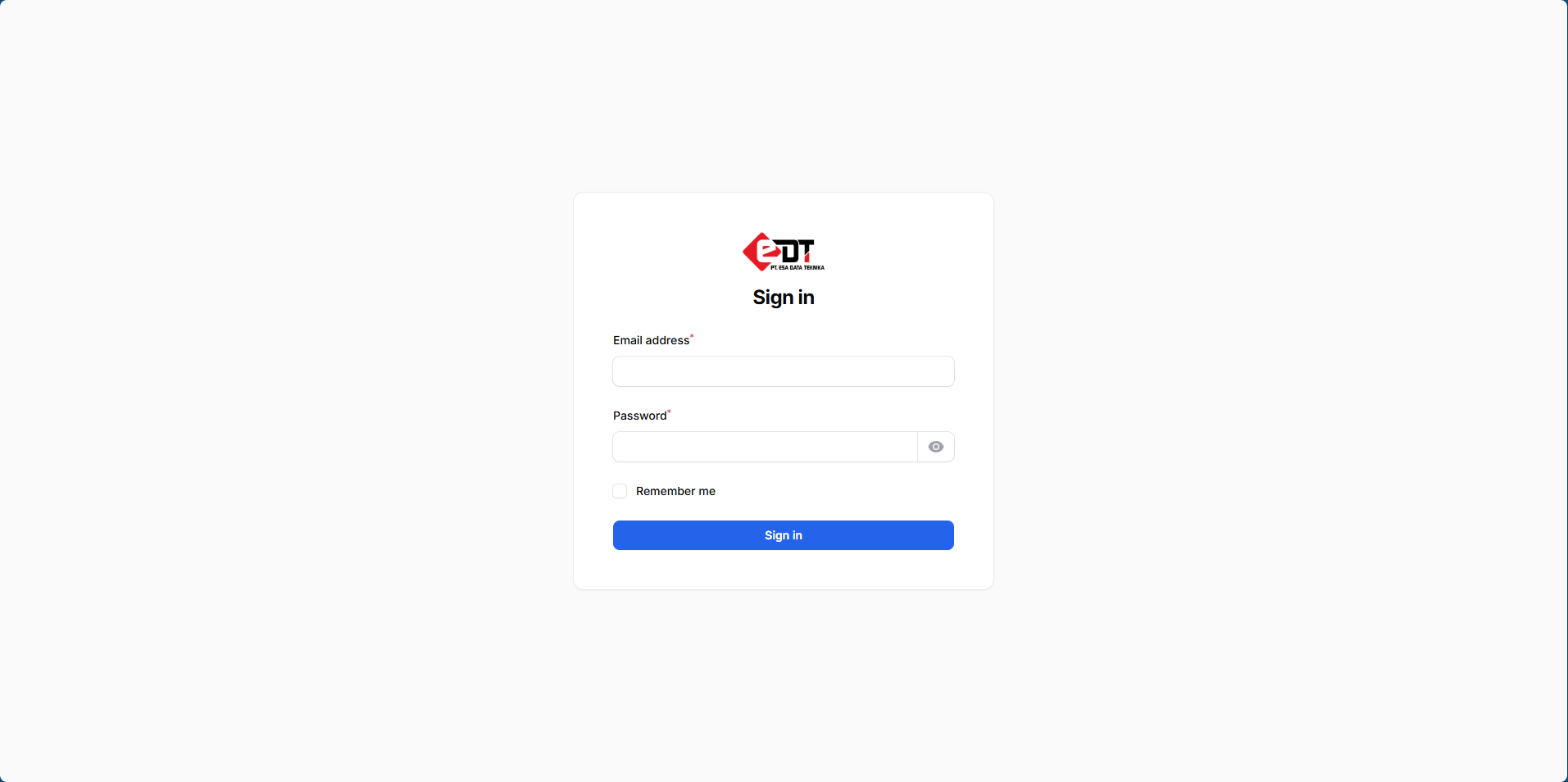
* 1. **Hasil**

Berikut ini tampilan hasil dari sistem informasi manajemen proyek menggunakan laravel 11 pada PT. Esa Data Teknika Bogor.

* + - 1. **Tampilan *Interface***

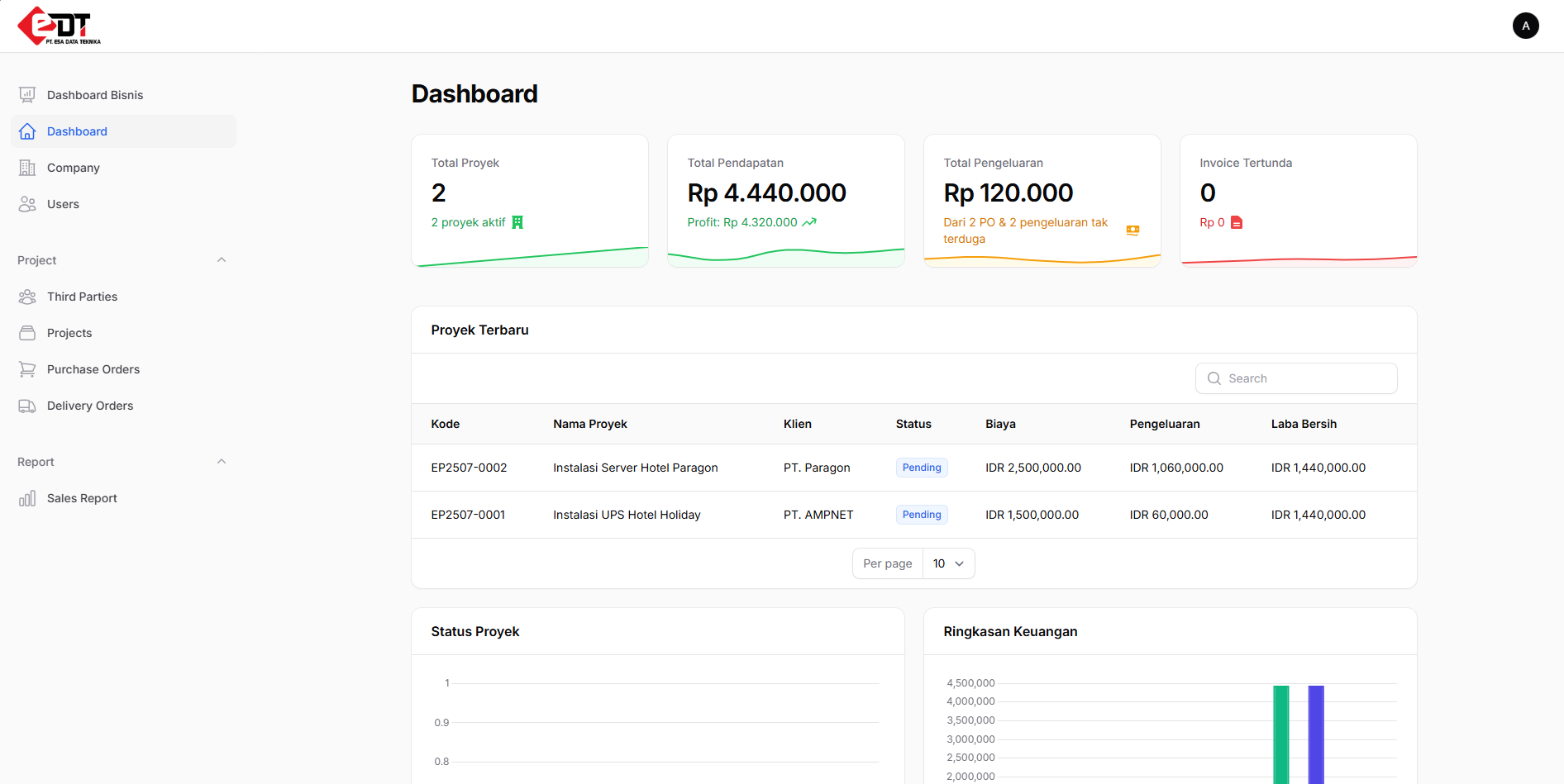
Berikut Berikut ini tampilan hasil dari sistem informasi manajemen proyek menggunakan laravel 11 di PT. Esa Data Teknika Bogor.

* + 1. Halaman *Login*



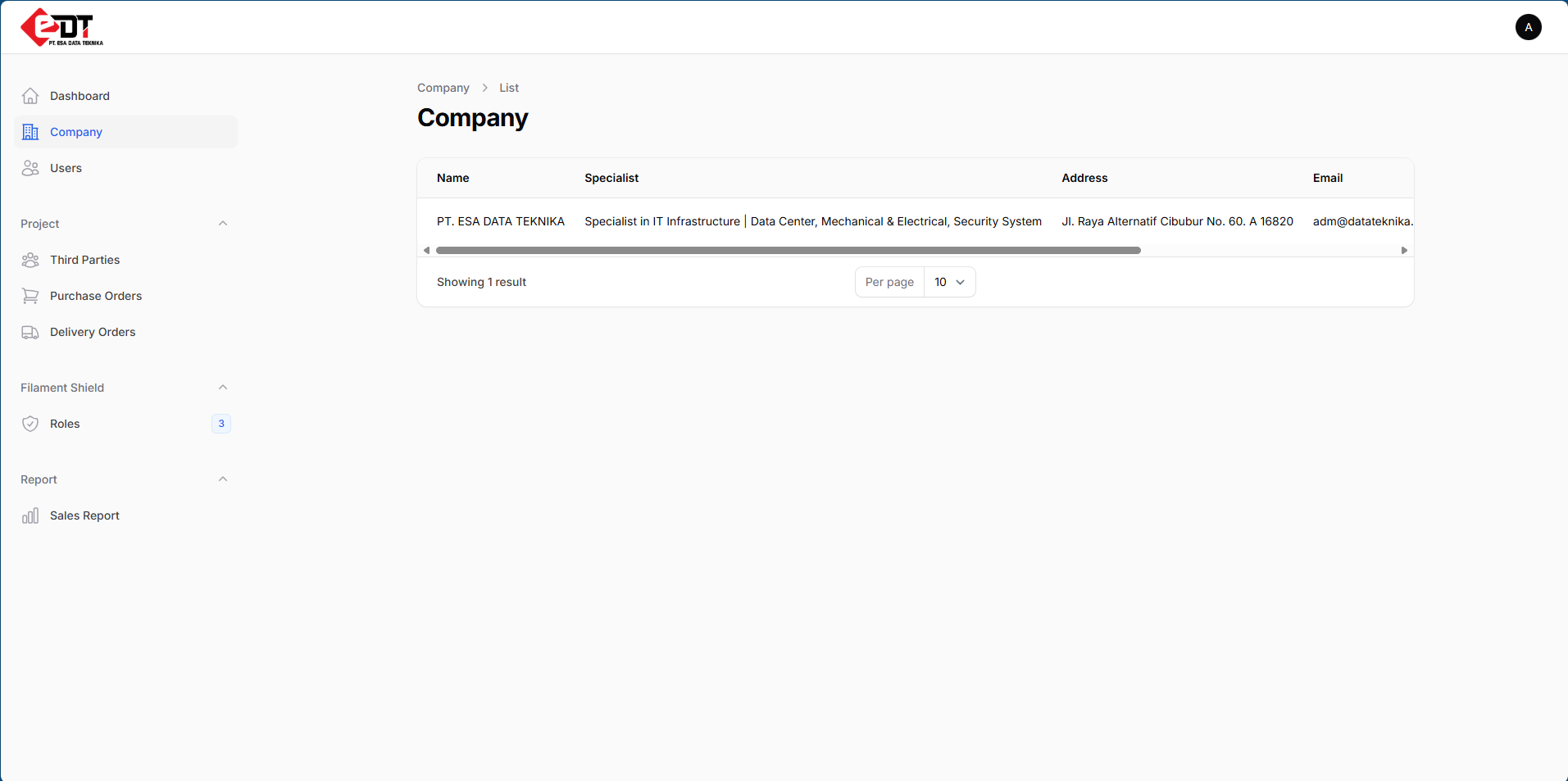
GAMBAR: 4.2. Tampilan halaman *login*

* + 1. Halaman *Dashboard*



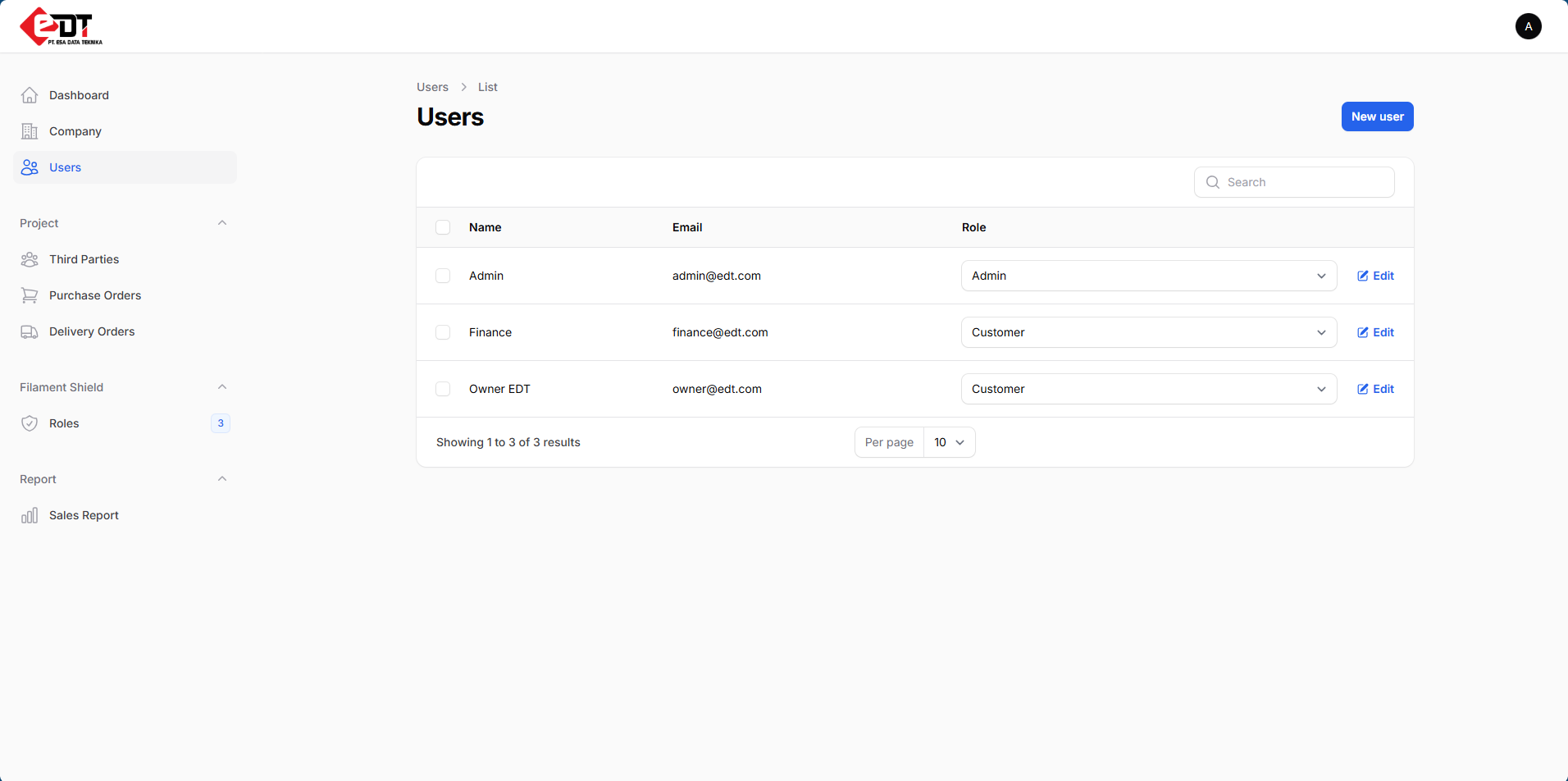
GAMBAR: 4.3. Tampilan halaman *dashboard*

* + 1. Halaman *Companies*



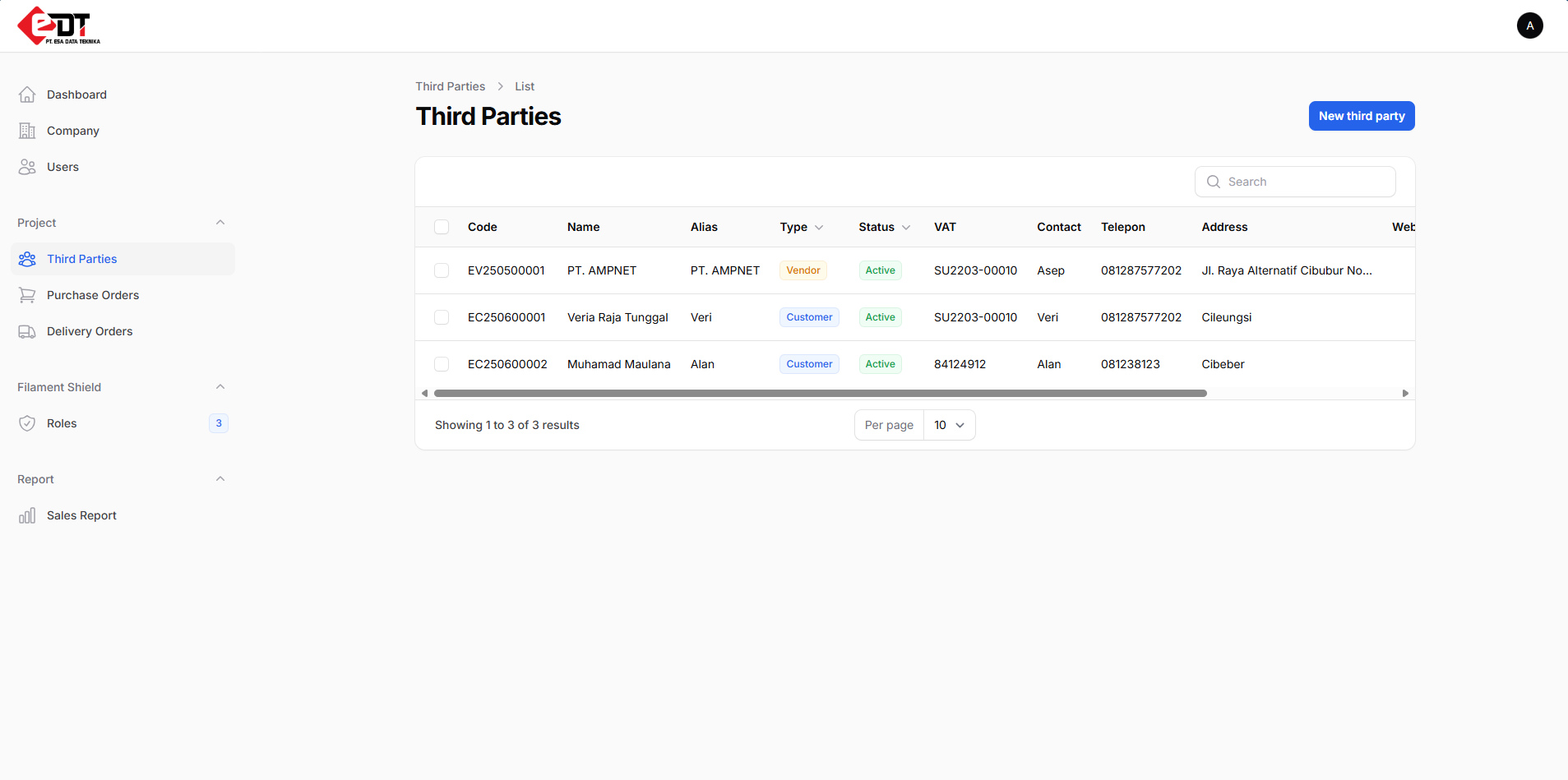
GAMBAR: 4.4. Tampilan halaman *companies*

* + 1. Halaman *Users*



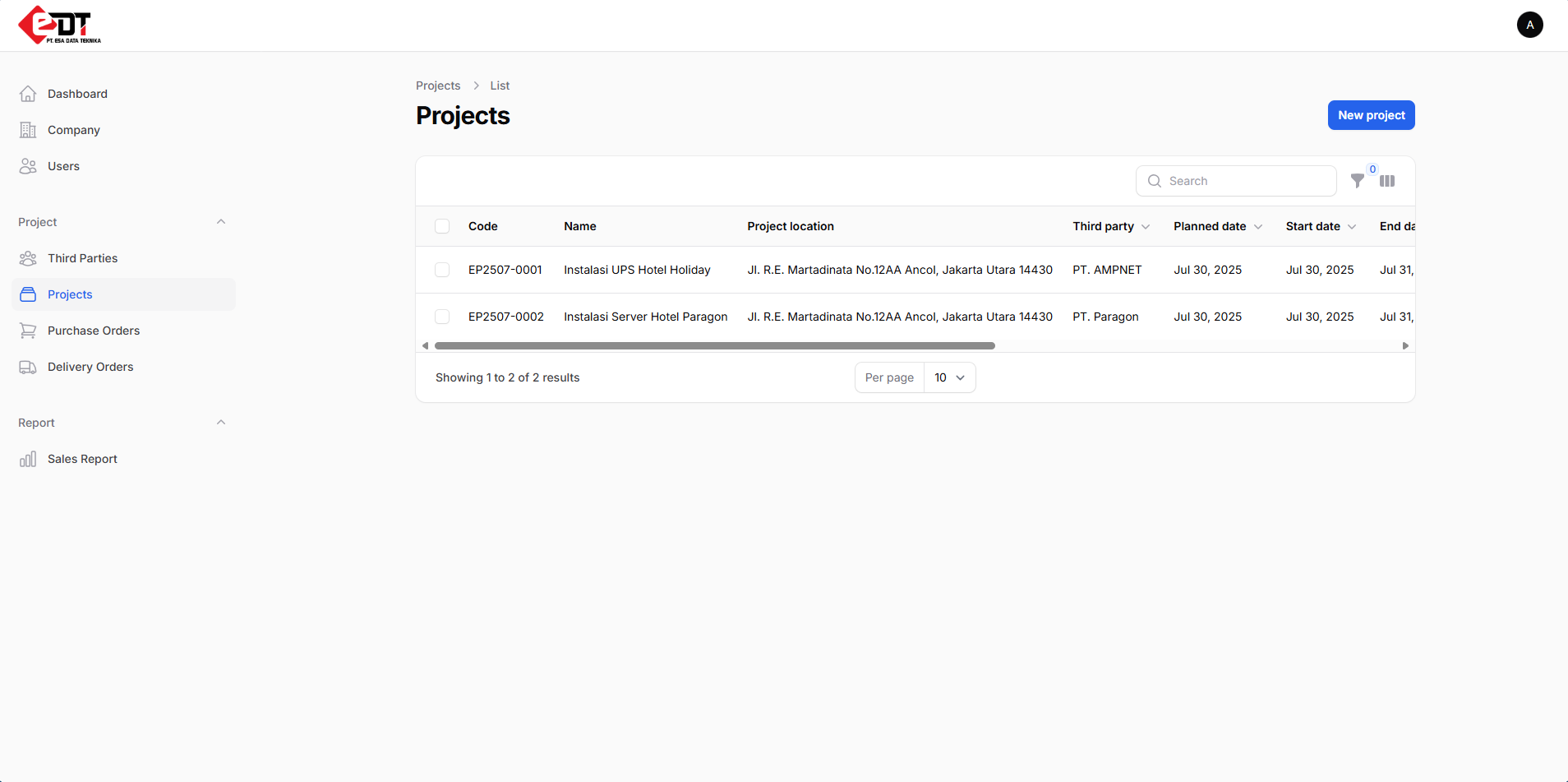
GAMBAR: 4.5. Tampilan halaman *users*

* + 1. Halaman *third parties*



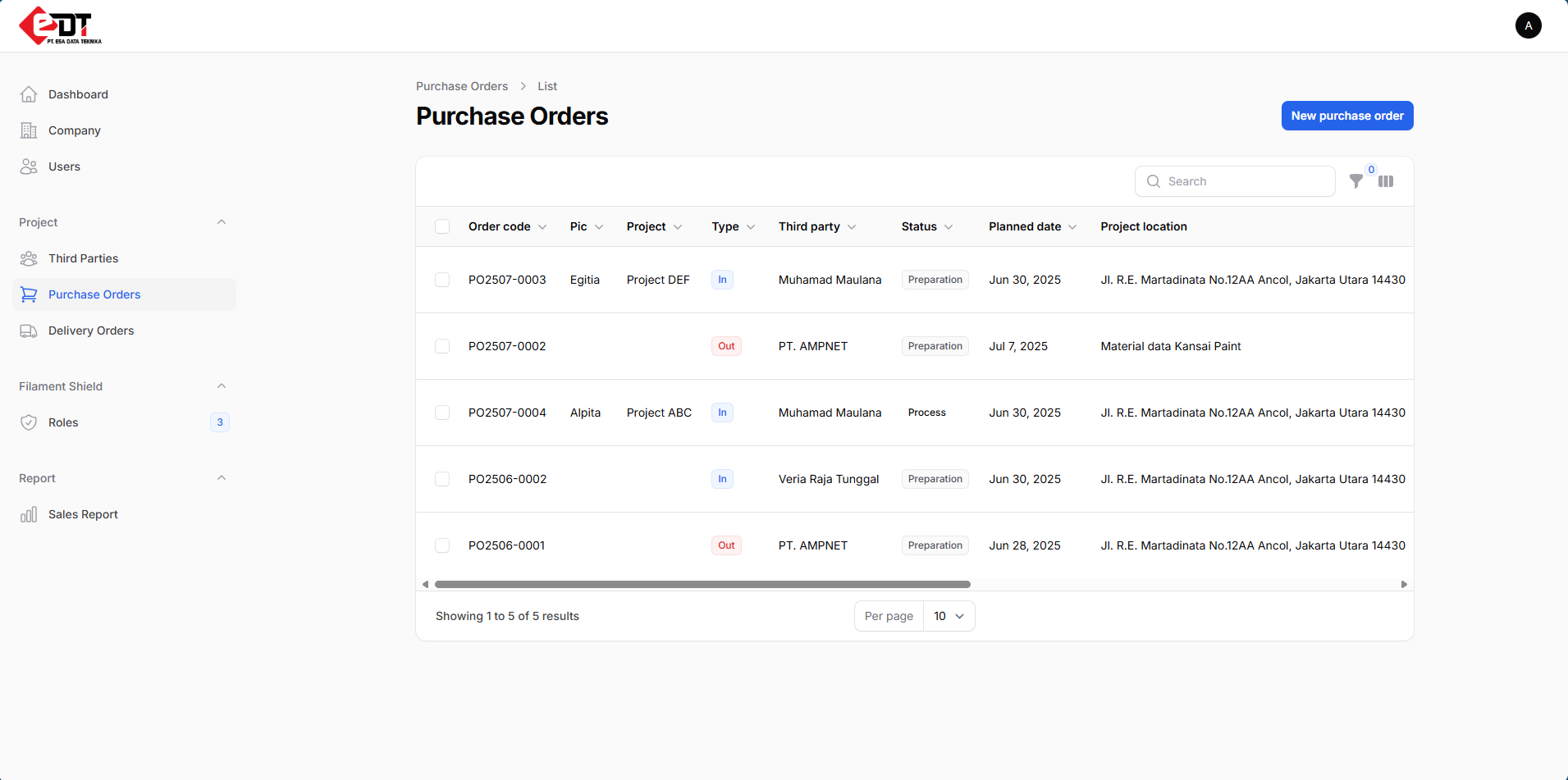
GAMBAR: 4.6. Tampilan halaman *third parties*

* + 1. Halaman *projects*



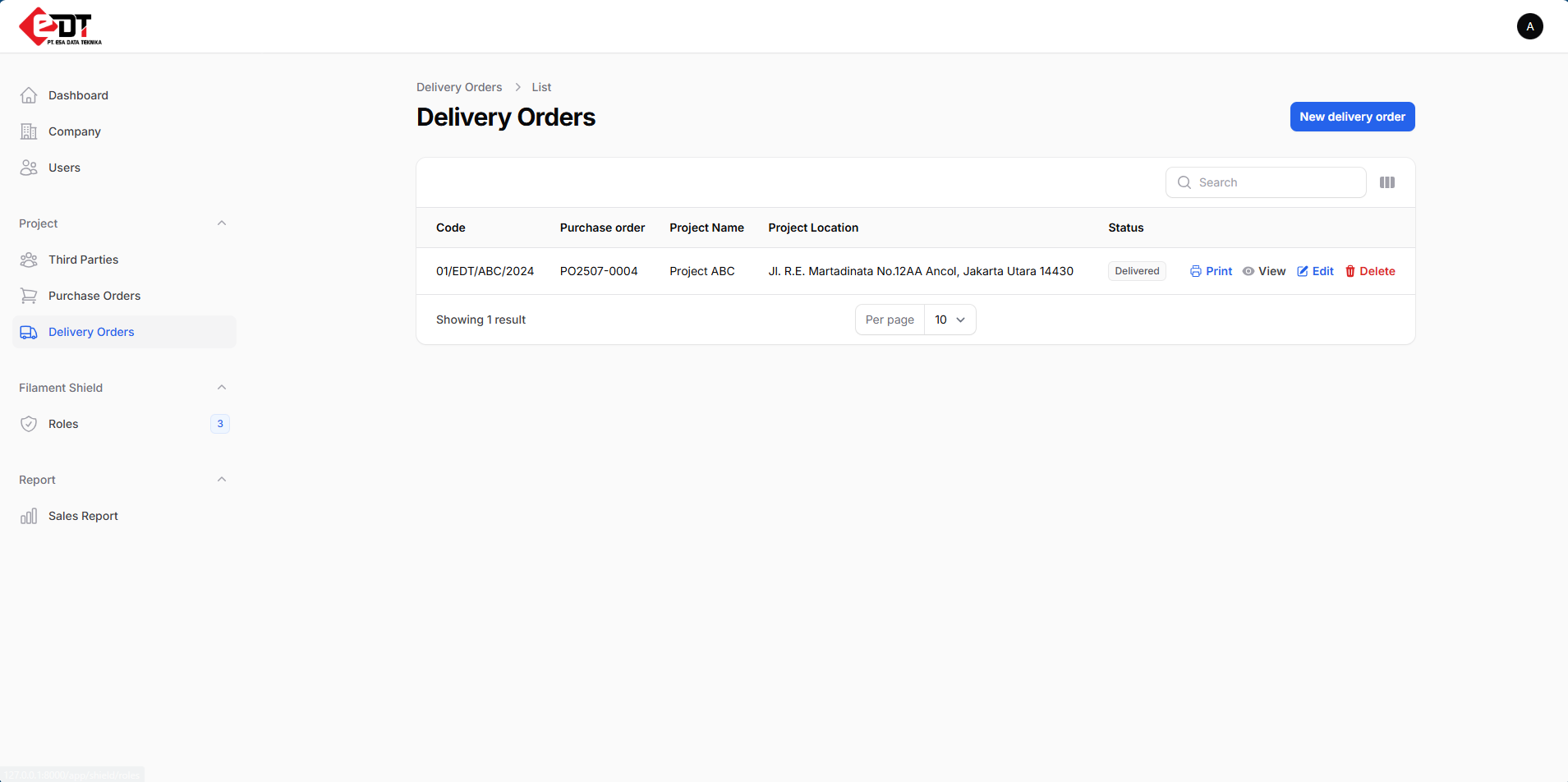
GAMBAR: 4.7. Tampilan halaman *projects*

* + 1. Halaman *purchase orders*



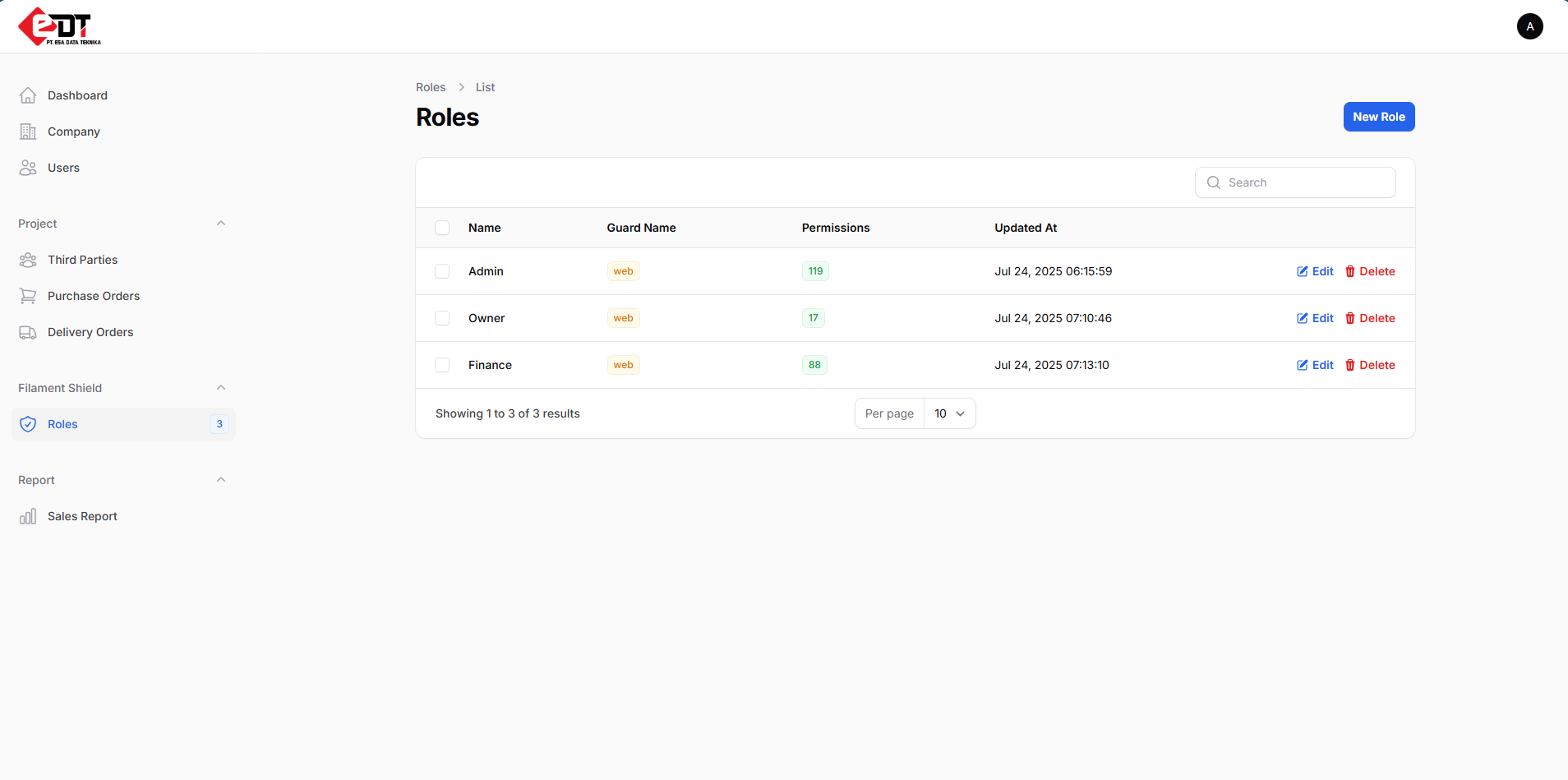
GAMBAR: 4.8. Tampilan halaman *purchase orders*

* + 1. Halaman *delivery orders*



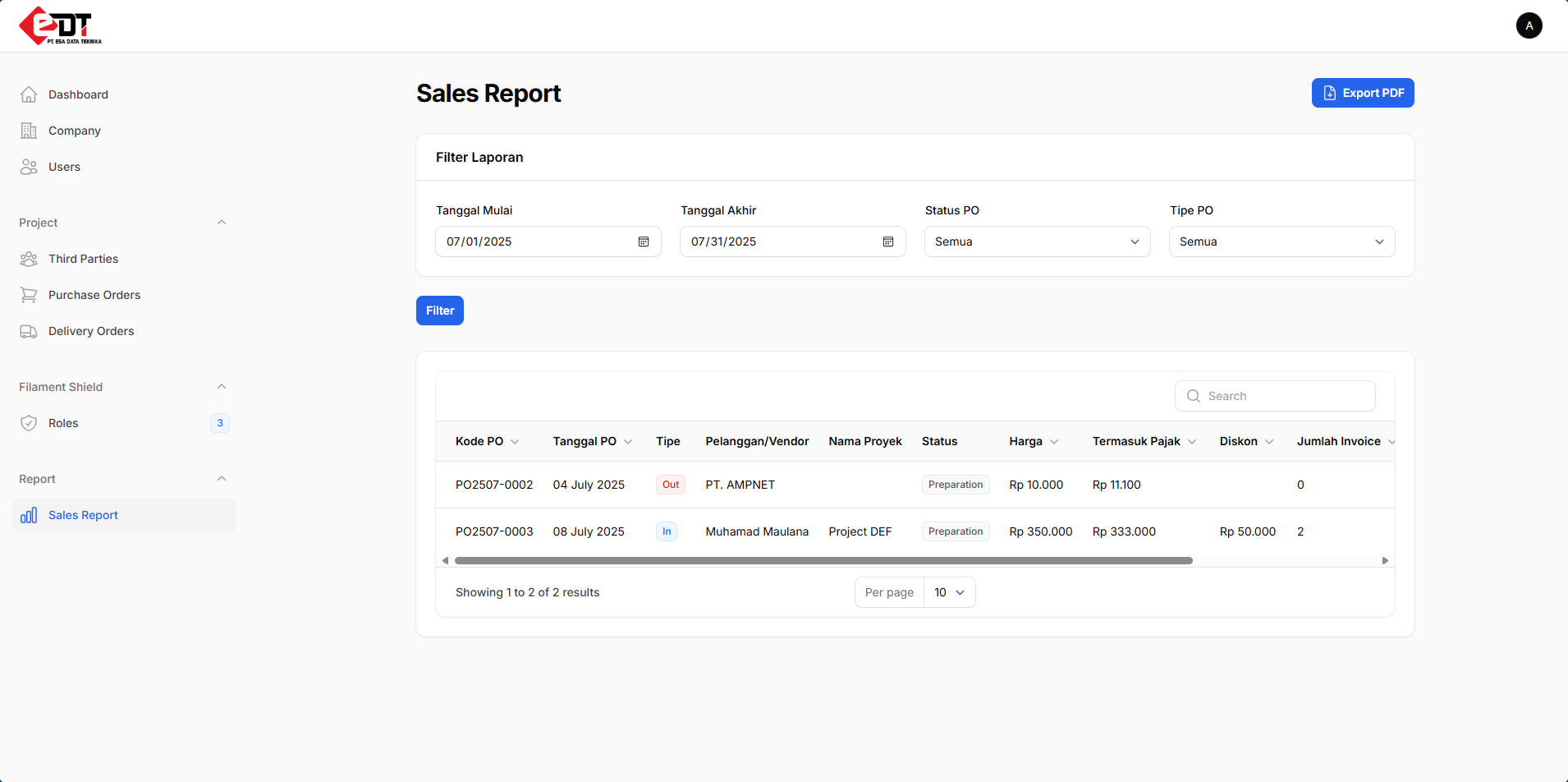
GAMBAR: 4.9. Tampilan halaman *delivery orders*

* + 1. Halaman *roles*



GAMBAR: 4.10. Tampilan halaman *roles*

* + - 1. **Tampilan Laporan**
    1. Berikut adalah halaman *sales reports:*

GAMBAR: 4.11. Tampilan halaman *sales reports*

* + - 1. **Hasil Pengujian**

Dalam penyusunan skripsi, hasil pengujian dilakukan dengan menggunakan teknik *Black Box Testing*. Teknik ini digunakan untuk menguji seluruh tampilan (form atau halaman) pada aplikasi yang dibangun telah berfungsi dengan baik atau tidak.

Tabel 4.13: Tabel hasil pengujian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nama Pengujian** | **Tase Case** | **Hasil Penguji** | **Ket** |
| 1 | *Form Login* |  | Sistem berhasil memverifikasi kredensial dan mengarahkan pengguna ke halaman dashboard. | Valid |
| 2 | *Dashboard* |  | Halaman dashboard tampil dengan informasi yang lengkap dan sesuai. | Valid |
| 3 | *Form* *Project* |  | Sistem berhasil meyimpan data *project* dan menampilkannya dalam daftar. | Valid |
| 4 | *Form* *Purchase Order* |  | Sistem berhasil meyimpan data *purchase order* dan menampilkannya dalam daftar. | Valid |
| 5 | *Form* *Delivery Order* |  | Sistem berhasil meyimpan data *delivery order* dan menampilkannya dalam daftar. | Valid |
| 6 | *Sales Reports* |  | Data *sales reports* muncul dengan benar dan lengkap, serta dapat difilter dan dicari. | Valid |

**BAB V**

**PENUTUP**

* 1. **Kesimpulan**

Berdasarkan hasil penelitian dan pengembangan Sistem Informasi Manajemen Proyek berbasis web pada PT. Esa Data Teknika, maka dapat disimpulkan hal-hal berikut:

1. Sistem informasi yang dikembangkan mampu mengintegrasikan proses pencatatan, pelaporan, dan pemantauan proyek secara digital dan real-time sehingga mengurangi risiko kesalahan pencatatan dan duplikasi data.
2. Sistem ini diharapkan dapat memudahkan pengguna dalam mengakses informasi proyek secara terpusat, mempercepat proses pelacakan progres, dan meningkatkan akurasi data proyek yang dibutuhkan oleh berbagai divisi.
3. Pengembangan sistem ini memberikan pengalaman langsung bagi penulis dalam menerapkan teori dan keterampilan yang telah diperoleh selama perkuliahan ke dalam permasalahan nyata di dunia kerja.
   1. **Saran**

Penulis menyadari bahwa dalam penulisan laporan ini belum sempurna, saran yang dapat penulis kemukakan adalah :

* 1. Disarankan untuk memberikan pelatihan dan sosialisasi kepada seluruh karyawan yang berkaitan dengan penggunaan sistem agar implementasinya berjalan optimal dan sesuai kebutuhan *operasional*.
  2. Perlunya mem-*backup database* secara berkala demi keamanan.
  3. Disarankan untuk melakukan evaluasi dan pemeliharaan sistem secara berkala guna memastikan bahwa sistem tetap berjalan dengan baik, serta dapat menyesuaikan kebutuhan dan perkembangan proyek di masa depan.

**DAFTAR PUSTAKA**

Abdillah, R., Kuncoro, A., & Kurniawan, I. (2019). Analisis Aplikasi Pembelajaran Matematika Berbasis Android dan Desain Sistem Menggunakan UML 2.0. Jurnal Theorems, 4(1), 301752.

Arifin, N. Y., Kom, S., Kom, M., Tyas, S. S., Sulistiani, H., Kom, M., Alim

Asmara, Y. (2019). Pembelajaran Sejarah Menjadi Bermakna dengan Pendekatan Kontektual. Kaganga: Jurnal Pendidikan Sejarah Dan Riset Sosial Humaniora, 2(2), 105–120.

Bratha, W. G. E. (2022). Literature Review Komponen Sistem Informasi Manajemen: Software, Database Dan Brainware. Jurnal Ekonomi Manajemen Sistem Informasi, 3(3), 344–360.

Budiman, I., Saori, S., Anwar, R. N., Fitriani, F., & Pangestu, M. Y. (2021). Analisis Pengendalian Mutu Di Bidang Industri Makanan (Studi Kasus: Umkm Mochi Kaswari Lampion Kota Sukabumi). Jurnal Inovasi Penelitian, 1(10), 2185–2190.

Darmawan, W. D., Junaedi, I., & Sianipar, A. Z. (2022). Perancangan aplikasi pembayaran produk distro brainnacle dengan berbasis web. Jurnal Sains Dan Teknologi Widyaloka (JSTekWid), 1(1), 26–48.

DR Prehanto, S.Kom, M.Kom (2020). Konsep Sistem Informasi.

Effendy, E., Siregar, E. A., Fitri, P. C., & Damanik, I. A. S. (2023). mengenal sistem informasi manajemen dakwah (pengertian sistem, karakteristik sistem). Jurnal Pendidikan Dan Konseling (JPDK), 5(2), 4343–4349.

Rosa A.S M. Shalahuddin (2019) . *Rekayasa Perangkat Lunak.* INFORMATIKA.

Roza, R., Fauzan, M. N., & Rahayu, W. I. (2020). Tutorial sistem informasi prediksi jumlah pelanggan menggunakan metode *regresi linier* berganda *berbasis web* menggunakan *framework codeigniter*. *Kreatif*.

Remawati, D., & Wijayanto, H. (2021). Buku Ajar*. Web JSP* dengan *database MySQL*.

Supardi, Y. (2019). Semua Bisa Menjadi Programmer *Laravel Basic*. Elex Media Komputindo.

Saputro, A. K., Wibisono, K. A., Alfita, R., Nahari, R. V., Ibadillah, A. F., & Anwar, M. N. (2024). Pemrogramandasar *web dengan html*. Penerbit kbm indonesia.

Tjandra, W. R., & SH, M. (2022). *Hukum Pengadaan Barang dan Jasa*. Prenada Media.

Elisabeth, D. M. (2019). Kajian terhadap peranan teknologi informasi dalam perkembangan audit komputerisasi (studi kajian teoritis). METHOMIKA: Jurnal Manajemen Informatika \& Komputerisasi Akuntansi, 3(1), 40–53.

Ependi, U. (2018). Pemodelan Sistem Informasi Monitoring Inventory Sekretariat Daerah Kabupaten Musi Banyuasin. Klik-Kumpulan Jurnal Ilmu Komputer, 5(1), 49–60.

Fikriyani, D. N., & Herdi, H. (2021). Perencanaan program bimbingan karir dalam meningkatkan eksplorasi karir siswa. JURNAL EDUKASI: Jurnal Bimbingan Konseling, 7(1), 1–14.

Firmansyah, H. I., Purnowidodo, A., & Setyabudi, S. A. (2018). Pengaruh mechanical bonding pada aluminium dengan serat karbon terhadap kekuatan tarik fiber metal laminates. Jurnal Rekayasa Mesin, 9(2), 127–134.

Frisdayanti, A. (2019). Peranan Brainware Dalam Sistem Informasi Manajemen. Jurnal Ekonomi Manajemen Sistem Informasi, 1(1), 60–69.

Habibi, R., Masruro, D. A., & Khonsa, N. H. (2020). Aplikasi Inventory barang menggunakan QR code (Vol. 1). Kreatif.

Hardiansyah, S. T., Kom, M., Suri, G. P., Kom, M., & others. (2022). Analisa Perancangan Sistem Informasi. Cendikia Mulia Mandiri.

Harismawan, A. F., Kharisma, A. P., & Afirianto, T. (2018). Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 2(1), 237–245.

Widia, D. M., & Asriningtias, S. R. (2021). *Cara Cepat dan Praktis Membangun Web Dinamis dengan PHP dan MySQL*. Universitas Brawijaya Press.

Hazimah, M., & Rizki, M. (2020). Perancangan sistem informasi administrasi rawat jalan pada klinik Insan Permata berbasis web. ADI Bisnis Digital Interdisiplin Jurnal, 1(2 Desember), 71–80.

**DAFTAR RIWAYAT HIDUP**

* + 1. Data Pribadi

NPM : 109210940174

Nama : Veria Raja Tunggal

Tempat / Tgl. Lahir : Wonogiri / 25 November 2002

Jenis Kelamin : Laki-Laki

Agama : Islam

Status Perkawinan : Belum Menikah

Jenjang : Strata-satu (S1)

Program Studi : Sistem Informasi

Alamat Rumah : Perum. Graha Mustika, RT/RW 07/09, Desa Bojong, Kecamatan Klapanunggal, 16710. Kabupaten Bogor

HP : 081287577202

Email : [veriarajatunggal@gmail.com](mailto:veriarajatunggal@gmail.com)

Pekerjaan : SMK Pariwisata Metland School

Jabatan : Guru Produktif PPLG

Alamat Kantor : Jl. Kota Taman Metropolitan, Cileungsi Kidul, Kec. Cileungsi, Kabupaten Bogor, Jawa Barat

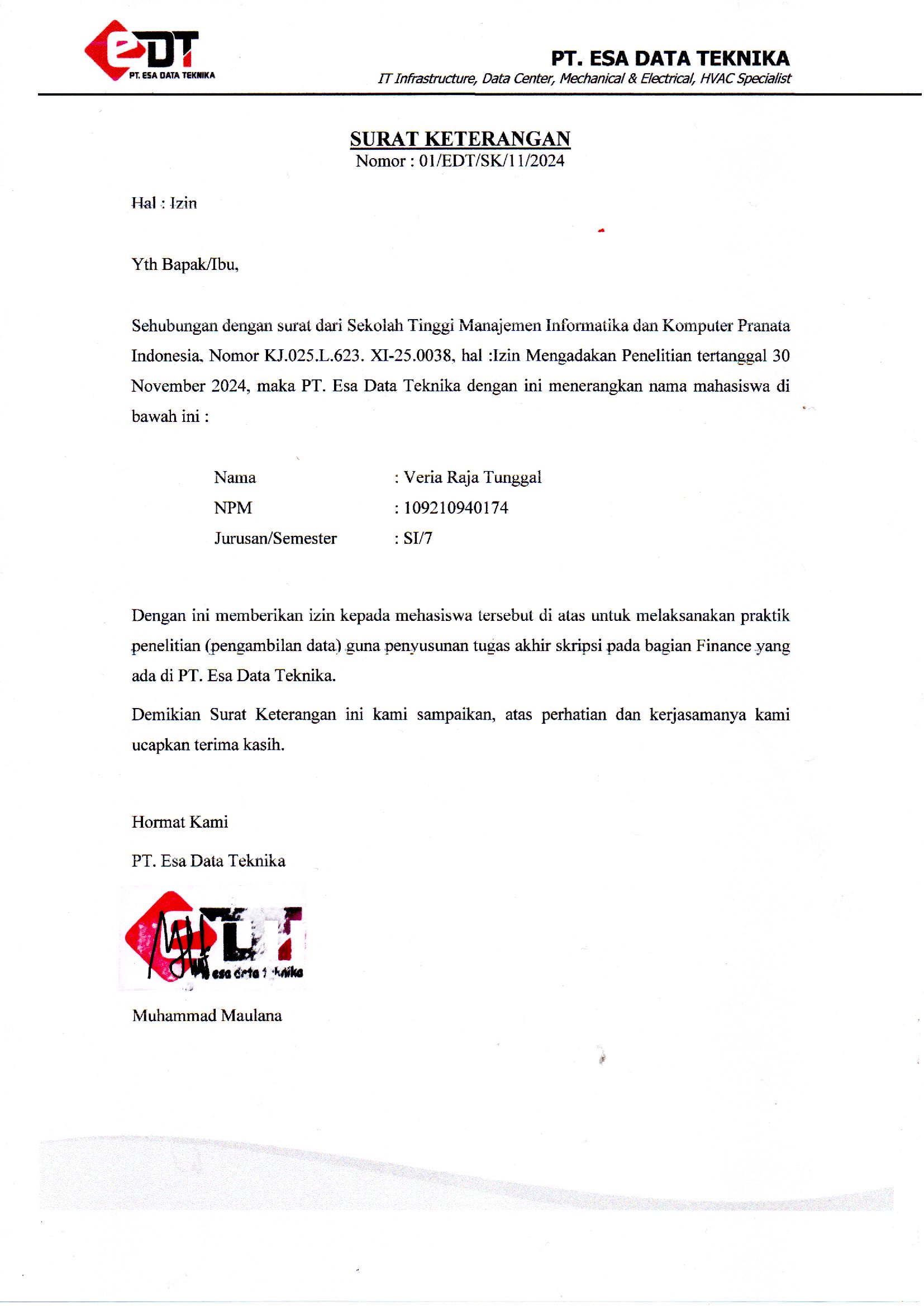
* + 1. Pendidikan

|  |  |  |  |
| --- | --- | --- | --- |
| Jenjang | Nama Lembaga | Jurusan | Tahun Lulus |
| SD | SDN Nyalindung |  | 2015 |
| SMP | SMPN 1 Klapanunggal |  | 2018 |
| SMK | SMK Bina Mandiri Multimedia | RPL | 2021 |
| PT | STMIK Pranata Indonesia | Strata-satu (S1) Sistem Informasi | 2025 |

Demikianlah daftar riwayat hidup ini dibuat dengan sebenarnya.

Bekasi, 20 Agustus 2025

Veria Raja Tunggal



**LAMPIRAN**

**SOURCE CODE**

* **Project**

<?php

namespace App\Filament\Resources;

use Filament\Forms;

use Filament\Tables;

use App\Models\Project;

use Filament\Forms\Form;

use App\Models\ThirdParty;

use Filament\Tables\Table;

use App\Models\PurchaseOrder;

use Filament\Resources\Resource;

use Filament\Forms\Components\Select;

use Filament\Resources\Components\Tab;

use Filament\Forms\Components\Textarea;

use Filament\Navigation\NavigationItem;

use Filament\Forms\Components\TextInput;

use Filament\Forms\Components\DatePicker;

use Illuminate\Database\Eloquent\Builder;

use App\Filament\Resources\ProjectResource\Pages;

use Illuminate\Database\Eloquent\SoftDeletingScope;

use App\Filament\Resources\ProjectResource\RelationManagers;

use App\Filament\Resources\ProjectResource\RelationManagers\TasksRelationManager;

use App\Filament\Resources\ProjectResource\RelationManagers\PurchaseOrdersInRelationManager;

use App\Filament\Resources\ProjectResource\RelationManagers\PurchaseOrdersOutRelationManager;

use App\Filament\Resources\ProjectResource\RelationManagers\UnexpectedExpensesRelationManager;

class ProjectResource extends Resource

{

protected static ?string $navigationGroup = 'Project';

protected static ?string $model = Project::class;

protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

public static function form(Form $form): Form

{

return $form

->schema([

TextInput::make('code')

->label('Code Project')

->readOnly()

->required()

->afterStateHydrated(function (callable $set) {

// Get year (2 digits) and month (2 digits)

$year = date('y');

$month = date('m');

// Count data for current month and year

$orderCount = Project::whereYear('created\_at', date('Y'))

->whereMonth('created\_at', date('m'))

->count();

// Create new sequence number

$newNumber = str\_pad($orderCount + 1, 4, '0', STR\_PAD\_LEFT);

// Combine into PO format

$poCode = "EP{$year}{$month}-{$newNumber}";

// Check for duplicates and increment if needed

while (Project::where('code', $poCode)->exists()) {

$newNumber = str\_pad((int)$newNumber + 1, 4, '0', STR\_PAD\_LEFT);

$poCode = "EP{$year}{$month}-{$newNumber}";

}

// Set form state

$set('code', $poCode);

}),

TextInput::make('name')

->label('Project Name')

->required()

->maxLength(255),

TextInput::make('project\_location')

->required()

->maxLength(255),

Select::make('third\_party\_id')

->relationship('thirdParty', 'name')

->required()

->createOptionForm([

TextInput::make('code')

->label('Code')

->readOnly()

->dehydrated()

->required()

->reactive(),

TextInput::make('name')->required(),

TextInput::make('alias')->required(),

Select::make('type')

->label('Type')

->options([

'Vendor' => 'Vendor',

'Customer' => 'Customer',

])

->required()

->reactive()

->afterStateUpdated(function ($state, callable $set) {

$count = ThirdParty::where('type', $state)->count() + 1;

$prefix = $state === 'Vendor' ? 'EV' : 'EC';

$code = $prefix . date('ym') . str\_pad($count, 5, '0', STR\_PAD\_LEFT);

$set('code', $code);

}),

Select::make('status')

->options([

'Active' => 'Active',

'Non Active' => 'Non Active',

])

->required(),

TextInput::make('vat')->required(),

TextInput::make('contact')->required(),

TextInput::make('telepon')->required(),

TextInput::make('address')->required(),

TextInput::make('website'),

]),

DatePicker::make('planned\_date')

->required(),

DatePicker::make('start\_date'),

DatePicker::make('end\_date'),

Select::make('status')

->options([

'Pending' => 'Pending',

'Preparation' => 'Preparation',

'Process' => 'Process',

'BAST' => 'BAST',

'Success' => 'Success',

'Cancel' => 'Cancel'

])

->default('Pending')

->required(),

Forms\Components\Grid::make(4)

->schema([

TextInput::make('cost')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

TextInput::make('remaining\_invoice')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

TextInput::make('expenses')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

TextInput::make('net\_cost')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

]),

Textarea::make('description')

->maxLength(65535)

->columnSpanFull(),

]);

}

public static function table(Table $table): Table

{

return $table

->columns([

Tables\Columns\TextColumn::make('code')

->searchable(),

Tables\Columns\TextColumn::make('name')

->searchable(),

Tables\Columns\TextColumn::make('project\_location')

->searchable(),

Tables\Columns\TextColumn::make('thirdParty.name')

->sortable(),

Tables\Columns\TextColumn::make('planned\_date')

->date()

->sortable(),

Tables\Columns\TextColumn::make('start\_date')

->date()

->sortable(),

Tables\Columns\TextColumn::make('end\_date')

->date()

->sortable(),

Tables\Columns\TextColumn::make('status')

->searchable(),

Tables\Columns\TextColumn::make('cost')

->money('idr')

->sortable(),

Tables\Columns\TextColumn::make('remaining\_invoice')

->money('idr')

->sortable(),

Tables\Columns\TextColumn::make('expenses')

->money('idr')

->sortable(),

Tables\Columns\TextColumn::make('net\_cost')

->money('idr')

->sortable(),

Tables\Columns\TextColumn::make('created\_at')

->dateTime()

->sortable()

->toggleable(isToggledHiddenByDefault: true),

Tables\Columns\TextColumn::make('updated\_at')

->dateTime()

->sortable()

->toggleable(isToggledHiddenByDefault: true),

])

->filters([

Tables\Filters\SelectFilter::make('status')

->options([

'Pending' => 'Pending',

'Preparation' => 'Preparation',

'Process' => 'Process',

'BAST' => 'BAST',

'Success' => 'Success',

'Cancel' => 'Cancel'

]),

Tables\Filters\SelectFilter::make('third\_party\_id')

->relationship('thirdParty', 'name'),

Tables\Filters\Filter::make('planned\_date')

->form([

DatePicker::make('planned\_from'),

DatePicker::make('planned\_until'),

])

->query(function (Builder $query, array $data): Builder {

return $query

->when(

$data['planned\_from'],

fn(Builder $query, $date): Builder => $query->whereDate('planned\_date', '>=', $date),

)

->when(

$data['planned\_until'],

fn(Builder $query, $date): Builder => $query->whereDate('planned\_date', '<=', $date),

);

}),

Tables\Filters\Filter::make('date\_range')

->form([

DatePicker::make('start\_from'),

DatePicker::make('start\_until'),

])

->query(function (Builder $query, array $data): Builder {

return $query

->when(

$data['start\_from'],

fn(Builder $query, $date): Builder => $query->whereDate('start\_date', '>=', $date),

)

->when(

$data['start\_until'],

fn(Builder $query, $date): Builder => $query->whereDate('start\_date', '<=', $date),

);

}),

])

->actions([

Tables\Actions\EditAction::make(),

Tables\Actions\Action::make('print')

->label('Cetak')

->icon('heroicon-o-printer')

->url(fn($record) => route('purchase-order.print', ['record' => $record->id]))

->openUrlInNewTab(),

])

->bulkActions([

Tables\Actions\BulkActionGroup::make([

Tables\Actions\DeleteBulkAction::make(),

]),

]);

}

public static function getRelations(): array

{

return [

PurchaseOrdersInRelationManager::class,

PurchaseOrdersOutRelationManager::class,

TasksRelationManager::class,

UnexpectedExpensesRelationManager::class,

];

}

public static function getPages(): array

{

return [

'index' => Pages\ListProjects::route('/'),

'create' => Pages\CreateProject::route('/create'),

'edit' => Pages\EditProject::route('/{record}/edit'),

];

}

}

* **Purchase Order**

<?php

namespace App\Filament\Resources;

use Filament\Forms;

use Filament\Tables;

use App\Models\Project;

use Filament\Forms\Form;

use App\Models\ThirdParty;

use Filament\Tables\Table;

use App\Models\PurchaseOrder;

use Ramsey\Uuid\Type\Integer;

use Filament\Resources\Resource;

use Filament\Forms\Components\Mask;

use Filament\Forms\Components\Select;

use Filament\Forms\Components\Textarea;

use Filament\Tables\Columns\TextColumn;

use Filament\Forms\Components\TextInput;

use Filament\Forms\Components\DatePicker;

use Filament\Tables\Columns\SelectColumn;

use Illuminate\Database\Eloquent\Builder;

use Filament\Forms\Components\Actions\Action;

use Illuminate\Database\Eloquent\SoftDeletingScope;

use App\Filament\Resources\PurchaseOrderResource\Pages;

use Pelmered\FilamentMoneyField\Forms\Components\MoneyInput;

use App\Filament\Resources\PurchaseOrderResource\RelationManagers;

use App\Filament\Resources\PurchaseOrderResource\RelationManagers\InvoicesRelationManager;

use App\Filament\Resources\PurchaseOrderResource\RelationManagers\FileOrdersRelationManager;

use App\Filament\Resources\PurchaseOrderResource\RelationManagers\ItemOrdersRelationManager;

class PurchaseOrderResource extends Resource

{

protected static ?string $model = PurchaseOrder::class;

protected static ?string $navigationGroup = 'Project';

protected static ?int $navigationSort = 1;

protected static ?string $navigationIcon = 'heroicon-o-shopping-cart';

public static function form(Form $form): Form

{

return $form

->schema([

Forms\Components\Select::make('project\_id')

->relationship('project', 'name')

->label('Project Name')

->required()

->searchable()

->preload()

->createOptionForm([

TextInput::make('code')

->label('Code Project')

->readOnly()

->required()

->afterStateHydrated(function (callable $set) {

// Get year (2 digits) and month (2 digits)

$year = date('y');

$month = date('m');

// Count data for current month and year

$orderCount = Project::whereYear('created\_at', date('Y'))

->whereMonth('created\_at', date('m'))

->count();

// Create new sequence number

$newNumber = str\_pad($orderCount + 1, 4, '0', STR\_PAD\_LEFT);

// Combine into PO format

$poCode = "EP{$year}{$month}-{$newNumber}";

// Check for duplicates and increment if needed

while (Project::where('code', $poCode)->exists()) {

$newNumber = str\_pad((int)$newNumber + 1, 4, '0', STR\_PAD\_LEFT);

$poCode = "EP{$year}{$month}-{$newNumber}";

}

// Set form state

$set('code', $poCode);

}),

TextInput::make('name')

->label('Project Name')

->required()

->maxLength(255),

TextInput::make('project\_location')

->required()

->maxLength(255),

Select::make('third\_party\_id')

->relationship('thirdParty', 'name')

->required()

->createOptionForm([

TextInput::make('code')

->label('Code')

->readOnly()

->dehydrated()

->required()

->reactive(),

TextInput::make('name')->required(),

TextInput::make('alias')->required(),

Select::make('type')

->label('Type')

->options([

'Vendor' => 'Vendor',

'Customer' => 'Customer',

])

->required()

->reactive()

->afterStateUpdated(function ($state, callable $set) {

$count = ThirdParty::where('type', $state)->count() + 1;

$prefix = $state === 'Vendor' ? 'EV' : 'EC';

$code = $prefix . date('ym') . str\_pad($count, 5, '0', STR\_PAD\_LEFT);

$set('code', $code);

}),

Select::make('status')

->options([

'Active' => 'Active',

'Non Active' => 'Non Active',

])

->required(),

TextInput::make('vat')->required(),

TextInput::make('contact')->required(),

TextInput::make('telepon')->required(),

TextInput::make('address')->required(),

TextInput::make('website'),

]),

DatePicker::make('planned\_date')

->required(),

DatePicker::make('start\_date'),

DatePicker::make('end\_date'),

Select::make('status')

->options([

'Pending' => 'Pending',

'Preparation' => 'Preparation',

'Process' => 'Process',

'BAST' => 'BAST',

'Success' => 'Success',

'Cancel' => 'Cancel'

])

->default('Pending')

->required(),

Forms\Components\Grid::make(2)

->schema([

TextInput::make('cost')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

TextInput::make('remaining\_invoice')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

]),

Forms\Components\Grid::make(2)

->schema([

TextInput::make('expenses')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

TextInput::make('net\_cost')

->numeric()

->prefix('Rp')

->default(0)

->readOnly(),

]),

Textarea::make('description')

->maxLength(65535)

->columnSpanFull(),

])

->createOptionAction(function (Action $action) {

return $action

->modalHeading('Create new project')

->modalButton('Create project')

->modalWidth('lg');

}),

Forms\Components\TextInput::make('order\_code')

->label('Purchase Order')

->readOnly()

->required()

->afterStateHydrated(function (callable $set) {

// Get year and month in 2 digits format

$year = date('y');

$month = date('m');

// Count orders for current month/year

$orderCount = PurchaseOrder::whereYear('created\_at', date('Y'))

->whereMonth('created\_at', date('m'))

->count();

// Generate new sequence number

$newNumber = str\_pad($orderCount + 1, 4, '0', STR\_PAD\_LEFT);

// Generate PO code

$poCode = "POO{$year}{$month}-{$newNumber}";

// Check for duplicates and increment if needed

while (PurchaseOrder::where('order\_code', $poCode)->exists()) {

$newNumber = str\_pad((int)$newNumber + 1, 4, '0', STR\_PAD\_LEFT);

$poCode = "POO{$year}{$month}-{$newNumber}";

}

$set('order\_code', $poCode);

$set('type', 'Out');

}),

Forms\Components\Hidden::make('type')

->default('Out'),

Forms\Components\Select::make('status')

->label('Status Order')

->options([

'Pending' => 'Pending',

'Preparation' => 'Preparation',

'Process' => 'Process',

'BAST' => 'BAST',

'Success' => 'Success',

'Cancel' => 'Cancel',

])

->default('Preparation')

->required(),

Forms\Components\Grid::make(4)

->schema([

Forms\Components\TextInput::make('price')

->label('Price')

->numeric()

->prefix('Rp ')

->readOnly()

->default(0)

->dehydrated(),

Forms\Components\TextInput::make('sales\_tax')

->label('Sales Tax')

->numeric()

->suffix('%')

->default(11)

->required(),

Forms\Components\TextInput::make('inc\_tax')

->label('Include Tax')

->numeric()

->readOnly()

->prefix('Rp ')

->default(0)

->dehydrated(),

Forms\Components\TextInput::make('discount')

->label('Discount')

->numeric()

->prefix('Rp ')

->default(0),

]),

Forms\Components\Select::make('payment\_terms')

->options([

'DP 30%, Progress 30%, After BAST 40%' => 'DP 30%, Progress 30%, After BAST 40%',

'DP 50%, Progress 20%, After BAST 25%, Retensi 1 Bulan 5%' => 'DP 50%, Progress 20%, After BAST 25%, Retensi 1 Bulan 5%',

'30% Down Payment, 60% Before Delivery, 10% After TesComm' => '30% Down Payment, 60% Before Delivery, 10% After TesComm',

'Cash On Delivery' => 'Cash On Delivery',

'50% DP, 50% After BAST' => '50% DP, 50% After BAST',

'50% DP, 50% Before Delivery' => '50% DP, 50% Before Delivery',

'Invoice/Maintenance Visit' => 'Invoice/Maintenance Visit',

'No DP, 100% After Completion' => 'No DP, 100% After Completion',

'40% Before Delivery, 55% After BAST, 5% Retention' => '40% Before Delivery, 55% After BAST, 5% Retention',

'100% Before Delivery' => '100% Before Delivery',

'DP 30%, Progress 30%, After Completion 35%, 5% Retention' => 'DP 30%, Progress 30%, After Completion 35%, 5% Retention',

'Due Upon Receipt' => 'Due Upon Receipt'

])

->required(),

Forms\Components\Select::make('payment\_type')

->options([

'Bank Transfer' => 'Bank Transfer',

'Cash' => 'Cash',

'Check' => 'Check',

'Credit Card' => 'Credit Card',

'Debit Payment Order' => 'Debit Payment Order'

])

->required(),

]);

}

public static function table(Table $table): Table

{

return $table

->defaultSort('created\_at', 'desc') // Sort by newest first

->columns([

Tables\Columns\TextColumn::make('order\_code')

->searchable()

->sortable(),

Tables\Columns\TextColumn::make('project.name')

->searchable()

->sortable(),

Tables\Columns\TextColumn::make('type')

->searchable()

->sortable()

->badge()

->color(fn(string $state): string => match ($state) {

'In' => 'info',

'Out' => 'danger',

}),

Tables\Columns\TextColumn::make('status')

->searchable()

->sortable()

->badge()

->color(fn(string $state): string => match ($state) {

'Pending' => 'warning',

'Preparation' => 'gray',

'Process' => 'puple',

'BAST' => 'info',

'Success' => 'success',

'Cancel' => 'danger',

}),

Tables\Columns\TextColumn::make('price')

->formatStateUsing(fn($state) => 'Rp ' . number\_format($state, 0, ',', '.'))

->sortable(),

Tables\Columns\TextColumn::make('sales\_tax')

->numeric()

->suffix('%')

->sortable(),

Tables\Columns\TextColumn::make('inc\_tax')

->formatStateUsing(fn($state) => 'Rp ' . number\_format($state, 0, ',', '.'))

->sortable(),

Tables\Columns\TextColumn::make('payment\_terms')

->searchable()

->wrap(),

Tables\Columns\TextColumn::make('payment\_type')

->searchable(),

Tables\Columns\TextColumn::make('created\_at')

->date('d F Y')

->sortable()

->badge()

->formatStateUsing(function ($state) {

$created = \Carbon\Carbon::parse($state);

$now = \Carbon\Carbon::now();

$diff = $created->diffInDays($now);

$diff = ceil($diff);

return $created->format('d F Y') . ' (' . $diff . ' days ago)';

})

->color(function ($state) {

$created = \Carbon\Carbon::parse($state);

$now = \Carbon\Carbon::now();

$diff = ceil($created->diffInDays($now));

return match (true) {

$diff <= 7 => 'success', // Less than a week

$diff <= 30 => 'warning', // Less than a month

default => 'danger', // More than a month

};

}),

// ->toggleable(isToggledHiddenByDefault: true),

Tables\Columns\TextColumn::make('updated\_at')

->dateTime()

->sortable()

->toggleable(isToggledHiddenByDefault: true)

])

->filters([

Tables\Filters\SelectFilter::make('type')

->options([

'In' => 'In',

'Out' => 'Out',

]),

Tables\Filters\SelectFilter::make('status')

->options([

'Pending' => 'Pending',

'Preparation' => 'Preparation',

'Process' => 'Process',

'BAST' => 'BAST',

'Success' => 'Success',

'Cancel' => 'Cancel',

]),

Tables\Filters\SelectFilter::make('third\_party')

->relationship('thirdParty', 'name'),

Tables\Filters\Filter::make('planned\_date')

->form([

Forms\Components\DatePicker::make('from'),

Forms\Components\DatePicker::make('until'),

])

->query(function (Builder $query, array $data): Builder {

return $query

->when(

$data['from'],

fn(Builder $query, $date): Builder => $query->whereDate('planned\_date', '>=', $date),

)

->when(

$data['until'],

fn(Builder $query, $date): Builder => $query->whereDate('planned\_date', '<=', $date),

);

}),

Tables\Filters\Filter::make('price\_range')

->form([

Forms\Components\TextInput::make('min\_price')

->numeric()

->prefix('Rp'),

Forms\Components\TextInput::make('max\_price')

->numeric()

->prefix('Rp'),

])

->query(function (Builder $query, array $data): Builder {

return $query

->when(

$data['min\_price'],

fn(Builder $query, $min): Builder => $query->where('price', '>=', $min),

)

->when(

$data['max\_price'],

fn(Builder $query, $max): Builder => $query->where('price', '<=', $max),

);

}),

])

->actions([

Tables\Actions\Action::make('export\_pdf')

->label('Export PDF')

->icon('heroicon-o-document-arrow-down')

->color('success')

->url(function () {

$params = [

'start\_date' => request('tableFilters.planned\_date.from') ?? now()->startOfMonth()->format('Y-m-d'),

'end\_date' => request('tableFilters.planned\_date.until') ?? now()->endOfMonth()->format('Y-m-d'),

'status' => request('tableFilters.status') ?? 'all',

'type' => request('tableFilters.type') ?? 'all',

'third\_party\_id' => request('tableFilters.third\_party') ?? 'all',

];

return route('purchase-order.export-pdf', $params);

})

->openUrlInNewTab(),

Tables\Actions\Action::make('export\_excel')

->label('Export Excel')

->icon('heroicon-o-table')

->color('primary')

->url(function () {

$params = [

'start\_date' => request('tableFilters.planned\_date.from') ?? now()->startOfMonth()->format('Y-m-d'),

'end\_date' => request('tableFilters.planned\_date.until') ?? now()->endOfMonth()->format('Y-m-d'),

'status' => request('tableFilters.status') ?? 'all',

'type' => request('tableFilters.type') ?? 'all',

'third\_party\_id' => request('tableFilters.third\_party') ?? 'all',

];

return route('purchase-order.export-excel', $params);

})

->openUrlInNewTab(),

])

->actions([

Tables\Actions\EditAction::make(),

Tables\Actions\Action::make('print')

->label('Cetak PDF')

->icon('heroicon-o-document-arrow-down')

->color('success')

->url(fn($record) => route('purchase-order.print', ['record' => $record->id]))

->openUrlInNewTab(),

])

->bulkActions([

Tables\Actions\BulkActionGroup::make([

Tables\Actions\DeleteBulkAction::make(),

]),

]);

}

public static function getRelations(): array

{

return [

ItemOrdersRelationManager::class,

FileOrdersRelationManager::class,

InvoicesRelationManager::class,

];

}

public static function getPages(): array

{

return [

'index' => Pages\ListPurchaseOrders::route('/'),

'create' => Pages\CreatePurchaseOrder::route('/create'),

'edit' => Pages\EditPurchaseOrder::route('/{record}/edit'),

];

}

}

* **Third Party**

<?php

namespace App\Filament\Resources;

use Filament\Forms;

use Filament\Tables;

use Filament\Forms\Form;

use App\Models\ThirdParty;

use Filament\Tables\Table;

use Filament\Resources\Resource;

use Filament\Forms\Components\Select;

use Filament\Tables\Columns\TextColumn;

use Filament\Forms\Components\TextInput;

use Illuminate\Database\Eloquent\Builder;

use Illuminate\Database\Eloquent\SoftDeletingScope;

use App\Filament\Resources\ThirdPartyResource\Pages;

use App\Filament\Resources\ThirdPartyResource\RelationManagers;

class ThirdPartyResource extends Resource

{

protected static ?string $model = ThirdParty::class;

protected static ?string $navigationGroup = 'Project';

protected static ?int $navigationSort = -2;

protected static ?string $navigationIcon = 'heroicon-o-user-group';

public static function form(Form $form): Form

{

return $form

->schema([

TextInput::make('code')

->label('Code')

->readOnly()

->dehydrated()

->required()

->reactive(),

TextInput::make('name')->required(),

TextInput::make('alias')->required(),

Select::make('type')

->label('Type')

->options([

'Vendor' => 'Vendor',

'Customer' => 'Customer',

])

->required()

->reactive()

->afterStateUpdated(function ($state, callable $set) {

// Hitung jumlah berdasarkan tipe

$count = ThirdParty::where('type', $state)->count() + 1;

// Tentukan awalan berdasarkan tipe

$prefix = $state === 'Vendor' ? 'EV' : 'EC';

// Format final kode

$code = $prefix . date('ym') . str\_pad($count, 5, '0', STR\_PAD\_LEFT);

// Set ke input code

$set('code', $code);

}),

Select::make('status')

->options([

'Active' => 'Active',

'Non Active' => 'Non Active',

])

->required(),

TextInput::make('vat')->required(),

TextInput::make('contact')->required(),

TextInput::make('telepon')->required(),

TextInput::make('address')->required(),

TextInput::make('website'),

]);

}

public static function table(Table $table): Table

{

return $table

->columns([

TextColumn::make('code')->label('Code')->searchable(),

TextColumn::make('name')->label('Name')->searchable(),

TextColumn::make('alias')->label('Alias')->searchable(),

TextColumn::make('type')

->label('Type')

->badge()

->color(fn(string $state): string => match ($state) {

'Vendor' => 'warning',

'Customer' => 'info',

default => 'gray',

})

->sortable(),

TextColumn::make('status')

->label('Status')

->badge()

->color(fn(string $state): string => match ($state) {

'Active' => 'success',

'Non Active' => 'danger',

default => 'gray',

})

->sortable(),

TextColumn::make('vat')->label('VAT'),

TextColumn::make('contact')->label('Contact'),

TextColumn::make('telepon')->label('Telepon'),

TextColumn::make('address')->label('Address')->limit(30),

TextColumn::make('website')->label('Website')->limit(30),

TextColumn::make('created\_at')->label('Created')->dateTime()->sortable(),

])

->filters([

//

])

->actions([

Tables\Actions\EditAction::make(),

])

->bulkActions([

Tables\Actions\BulkActionGroup::make([

Tables\Actions\DeleteBulkAction::make(),

]),

]);

}

public static function getRelations(): array

{

return [

//

];

}

public static function getPages(): array

{

return [

'index' => Pages\ListThirdParties::route('/'),

'create' => Pages\CreateThirdParty::route('/create'),

'edit' => Pages\EditThirdParty::route('/{record}/edit'),

];

}

}

* **Sales Report**

<?php

namespace App\Filament\Pages;

use App\Models\PurchaseOrder;

use App\Models\Invoice;

use App\Models\DeliveryOrder;

use App\Models\DeliveryItem;

use Filament\Pages\Page;

use Filament\Forms\Components\DatePicker;

use Filament\Forms\Components\Select;

use Filament\Forms\Components\Section;

use Filament\Forms\Form;

use Filament\Tables\Columns\TextColumn;

use Filament\Tables\Concerns\InteractsWithTable;

use Filament\Tables\Contracts\HasTable;

use Filament\Tables\Table;

use Illuminate\Database\Eloquent\Builder;

use Illuminate\Support\Carbon;

use Filament\Actions\Action;

use Illuminate\Support\Facades\Route;

class SalesReport extends Page implements HasTable

{

use InteractsWithTable;

protected static ?string $navigationIcon = 'heroicon-o-chart-bar';

protected static ?string $navigationLabel = 'Sales Report';

protected static ?string $title = 'Sales Report';

protected static ?string $navigationGroup = 'Report';

protected static ?int $navigationSort = -1;

protected static string $view = 'filament.pages.sales-report';

public ?array $data = [];

public function mount(): void

{

$this->form->fill();

}

protected function getHeaderActions(): array

{

return [

Action::make('export\_pdf')

->label('Export PDF')

->icon('heroicon-o-document-arrow-down')

->url(function () {

$params = [

'start\_date' => $this->data['start\_date'] ?? now()->startOfMonth()->format('Y-m-d'),

'end\_date' => $this->data['end\_date'] ?? now()->endOfMonth()->format('Y-m-d'),

'status' => $this->data['status'] ?? 'all',

'type' => $this->data['type'] ?? 'all',

];

return route('sales-report.export-pdf', $params);

})

->openUrlInNewTab(),

];

}

public function form(Form $form): Form

{

return $form

->schema([

Section::make('Filter Laporan')

->schema([

DatePicker::make('start\_date')

->label('Tanggal Mulai')

->default(now()->startOfMonth())

->reactive(),

DatePicker::make('end\_date')

->label('Tanggal Akhir')

->default(now()->endOfMonth())

->reactive(),

Select::make('status')

->label('Status PO')

->options([

'all' => 'Semua',

'Preparation' => 'Preparation',

'Process' => 'Process',

'BAST' => 'BAST',

'Success' => 'Success',

'Cancel' => 'Cancel',

])

->default('all')

->reactive(),

Select::make('type')

->label('Tipe PO')

->options([

'all' => 'Semua',

'In' => 'In',

'Out' => 'Out',

])

->default('all')

->reactive(),

])

->columns(4),

])

->statePath('data');

}

public function table(Table $table): Table

{

return $table

->query($this->getTableQuery())

->columns([

TextColumn::make('order\_code')

->label('Kode PO')

->searchable()

->sortable(),

TextColumn::make('created\_at')

->label('Tanggal PO')

->date('d F Y')

->sortable(),

TextColumn::make('type')

->label('Tipe')

->badge()

->color(fn(string $state): string => match ($state) {

'In' => 'info',

'Out' => 'danger',

}),

TextColumn::make('project.thirdParty.name')

->label('Pelanggan/Vendor')

->searchable(),

TextColumn::make('project.name')

->label('Nama Proyek')

->searchable(),

TextColumn::make('status')

->label('Status')

->badge()

->color(fn(string $state): string => match ($state) {

'Pending' => 'warning',

'Preparation' => 'gray',

'Process' => 'purple',

'BAST' => 'info',

'Success' => 'success',

'Cancel' => 'danger',

}),

TextColumn::make('price')

->label('Harga')

->formatStateUsing(fn($state) => 'Rp ' . number\_format($state, 0, ',', '.'))

->sortable(),

TextColumn::make('inc\_tax')

->label('Termasuk Pajak')

->formatStateUsing(fn($state) => 'Rp ' . number\_format($state, 0, ',', '.'))

->sortable(),

TextColumn::make('discount')

->label('Diskon')

->formatStateUsing(fn($state) => 'Rp ' . number\_format($state, 0, ',', '.'))

->sortable(),

TextColumn::make('invoices\_count')

->label('Jumlah Invoice')

->counts('invoices')

->sortable(),

TextColumn::make('invoices\_sum\_amount\_paid')

->label('Total Dibayar')

->sum('invoices', 'amount\_paid')

->formatStateUsing(fn($state) => 'Rp ' . number\_format($state, 0, ',', '.'))

->sortable(),

TextColumn::make('delivery\_orders\_count')

->label('Jumlah DO')

->counts('deliveryOrders')

->sortable(),

])

->filters([])

->actions([])

->bulkActions([])

->emptyStateHeading('Belum ada data penjualan')

->emptyStateDescription('Data penjualan akan muncul di sini setelah Anda membuat Purchase Order.');

}

protected function getTableQuery(): Builder

{

$query = PurchaseOrder::query()

->with(['thirdParty', 'invoices', 'deliveryOrders'])

->when(

isset($this->data['start\_date']) && $this->data['start\_date'],

fn(Builder $query) => $query->whereDate('created\_at', '>=', $this->data['start\_date'])

)

->when(

isset($this->data['end\_date']) && $this->data['end\_date'],

fn(Builder $query) => $query->whereDate('created\_at', '<=', $this->data['end\_date'])

)

->when(

isset($this->data['status']) && $this->data['status'] !== 'all',

fn(Builder $query) => $query->where('status', $this->data['status'])

)

->when(

isset($this->data['type']) && $this->data['type'] !== 'all',

fn(Builder $query) => $query->where('type', $this->data['type'])

);

return $query;

}

public function filter(): void

{

// Metode ini akan dipanggil saat form disubmit

// Tidak perlu melakukan apa-apa karena getTableQuery() sudah menggunakan data filter

}

}

**SISTEM INFORMASI MANAJEMEN PROYEK MENGGUNAKAN FRAMEWORK LARAVEL 11 DI PT. ESA DATA TEKNIKA BOGOR**

**Imam Zaenudin,** Sistem Informasi, STMIK Pranata Indonesia, Email: [imamzaenuddin@gmail.com](mailto:imamzaenuddin@gmail.com1)

**Harry Subagjo,** Sistem Informasi, STMIK Pranata Indonesia, Email: harrysb@rocketmail.com

**Veria Raja Tunggal,** Sistem Informasi, STMIK Pranata Indonesia, Email : veriarajatunggal@gmail.com

***Abstract***

*This research aims to develop a project management information system using the Laravel 11 framework at PT. Esa Data Teknika, Bogor, with a focus on improving efficiency and accuracy in project management. The methodology employed in this study is the framework method, which serves as a structured approach designed to facilitate the development and maintenance of applications.  
The system design was carried out using Unified Modeling Language (UML), which includes the Use Case Diagram, Activity Diagram, Class Diagram, and Sequence Diagram to define the system’s structure and workflow.   
The results of this study are an information system that provides strong support for the operational needs of PT. Esa Data Teknika in project management. The system is well-integrated in terms of data management and decision-making based on the generated reports.*

***Keywords****: Information System, Project Management, Laravel Framework*

***Abstrak***

*Penelitian ini bertujuan untuk mengembangkan sistem informasi manajemen proyek menggunakan framework laravel 11 di PT. Esa Data Teknika Bogor, dengan fokus meningkatkan efisiensi dan akurasi dalam manajemen proyek.*

*Metodologi yang digunakan dalam penelitian ini adalah metode framework, yang merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi****.***

*Perancangan sistem dilakukan menggunakan Unified Modeling Language (UML) dengan mencakup diagram Use Case Diagram, Activity Diagram, Class Diagram dan Sequence Diagram untuk mendefinisikan struktur dan alur kerja sistem.*

*Hasil penelitian ini adalah sebuah sistem informasi yang dapat memberikan dukungan yang baik untuk kebutuhan operasional PT. Esa Data Teknika dalam manajemen proyek. Sistem ini terintegrasi dengan baik dalam hal manajemen data dan pengambilan keputusan berdasarkan laporan yang dihasilkan.*

***Kata Kunci*** *: Sistem Informasi, manajemen proyek, framework laravel*

1. **PENDAHULUAN**

Manajemen proyek memegang peranan penting dalam keberhasilan operasional perusahaan di berbagai sektor industri. Pengelolaan proyek yang efektif tidak hanya mencakup pelacakan progres dan alokasi sumber daya, tetapi juga membutuhkan akses informasi cepat, transparansi data, dan koordinasi lintas divisi secara real-time.

PT. Esa Data Teknika saat ini masih mengandalkan Microsoft Excel untuk mengelola data proyek, mulai dari pencatatan, pelacakan progres, hingga pembuatan laporan. Meskipun fleksibel, metode ini memiliki keterbatasan seperti validasi data yang lemah, tidak adanya pembaruan data secara real-time, serta risiko kehilangan data. Akibatnya, perusahaan mengalami kesulitan mendapatkan gambaran proyek secara utuh dan terkini.

Oleh karena itu, penelitian ini bertujuan mengembangkan **Sistem Informasi Manajemen Proyek berbasis web** menggunakan framework Laravel 11. Sistem ini diharapkan dapat mengintegrasikan data proyek dalam satu platform terpusat, menyediakan pelacakan progres yang sistematis, serta meningkatkan akurasi dan keamanan data.

**Tujuan Penelitian**:

1. Mengembangkan sistem informasi manajemen proyek berbasis web terintegrasi guna meningkatkan efisiensi operasional di PT. Esa Data Teknika.
2. Menyediakan alat bantu pencatatan, pemantauan progres, dan koordinasi tim secara real-time.
3. **METODE PENELITIAN**

Penelitian ini menggunakan **metode framework**, yaitu pendekatan pengembangan aplikasi berbasis kerangka kerja yang mempermudah proses coding, meningkatkan keamanan, dan menjaga standar kode.

**Objek Penelitian**: PT. Esa Data Teknika Bogor.

**Tahapan Penelitian**:

1. **Pengumpulan Data**
   * **Observasi**: Mengamati proses kerja manual.
   * **Wawancara**: Menggali kendala dari pihak terkait.
   * **Studi Pustaka**: Mengkaji literatur terkait manajemen proyek dan Laravel.
2. **Analisis Sistem Berjalan**
   * Mengidentifikasi proses manual seperti pembuatan invoice, purchase order, dan laporan proyek.
3. **Perancangan Sistem**
   * Menggunakan **UML** (Use Case, Activity, Class, Sequence Diagram).
   * Merancang basis data terpusat dan terstruktur.
4. **Pengembangan Sistem**
   * Menggunakan Laravel 11, PHP, MySQL.
   * Penerapan **Role-Based Access Control (RBAC)**.
5. **Pengujian Sistem**
   * Menggunakan uji coba internal untuk memastikan fungsi berjalan sesuai kebutuhan.
6. **HASIL DAN PEMBAHASAN**

Hasil penelitian adalah sistem informasi manajemen proyek berbasis web yang memiliki fitur:

1. **Pengelolaan Data Proyek Terpusat** – mencatat dan mengelola data proyek secara terstruktur.
2. **Monitoring Progress Proyek** – menampilkan status dan perkembangan proyek secara real-time.
3. **Manajemen Klien & Vendor** – menyimpan data pihak terlibat dengan pencarian cepat.
4. **Manajemen Dokumen** – unggah dan kategorisasi dokumen proyek.
5. **Pelaporan Otomatis** – menghasilkan laporan proyek dalam format PDF/Excel.

**Pembahasan**  
Sistem yang dikembangkan mampu mengatasi permasalahan sebelumnya:

1. Data proyek kini tersimpan di database terpusat sehingga mengurangi risiko duplikasi.
2. Monitoring progres menjadi lebih akurat dengan indikator visual.
3. Proses pelaporan menjadi lebih cepat dan terintegrasi.

Dibandingkan penelitian terdahulu, sistem ini memiliki keunggulan pada integrasi antar-divisi, pelaporan otomatis, serta fleksibilitas pengelolaan dokumen proyek.

1. **KESIMPULAN**

Pengembangan sistem informasi manajemen proyek berbasis Laravel 11 di PT. Esa Data Teknika mampu meningkatkan efisiensi operasional, mempercepat koordinasi antar tim, dan mempermudah proses pelaporan. Sistem ini dapat menjadi solusi jangka panjang bagi perusahaan dalam menghadapi tantangan pengelolaan proyek yang kompleks.

**Saran**

Pengembangan selanjutnya dapat mencakup integrasi dengan aplikasi mobile, modul notifikasi otomatis, serta integrasi dengan API pihak ketiga untuk mendukung manajemen proyek yang lebih luas.

1. **REFERENSI**
2. Rosa, A.S. & Shalahuddin, M. (2019). Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika.
3. Widarti, E., Sari, F.P., & Mardian, R. (2024). Sistem Informasi dan Teknologi.
4. Sabaruddin, R., Murni, S., Lisnawanty, & Nugraha, W. (2024). Laravel Framework Development.
5. Sulistyoningrum, T. (2020). Pengantar Basis Data.
6. Wijayanti, H. (2020). Web Server Technology.
7. Muthohir, M. (2020). Web Browser & Development Tools.