

**PENGEMBANGAN APLIKASI *PRIOR ADJUSTMENT* BERBASIS *WEB*
MENGUNAKAN *FRAMEWORK ANGULAR* PADA PT. ADICIPTA
INOVASI TEKNOLOGI**

PRAKTIK KERJA LAPANGAN



**SIUSIU TRIFENA HALIM
NIM : 312110013**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG
MALANG
2025**

**LEMBAR PENGESAHAN
PRAKTEK KERJA LAPANGAN**

**PENGEMBANGAN APLIKASI *PRIOR ADJUSTMENT* BERBASIS *WEB*
MENGUNAKAN *FRAMEWORK ANGULAR* PADA PT. ADICIPTA
INOVASI TEKNOLOGI**

Oleh:

Siusiu Trifena Halim

NIM. 312110013

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

Dosen Pembimbing,



Paulus Lucky Tirma Irawan, S.Kom., MT.

NIP. 20100005

Dekan Fakultas Sains dan Teknologi



Prof. Dr.Eng. Romv Budhi, ST., MT., M.Pd.

NIP. 20070035

Kata Pengantar

Puji syukur penulis ucapkan kepada Tuhan yang Maha Esa karena berkat dan rahmat yang diberikan selama kegiatan Praktik Kerja Lapangan berlangsung, sehingga penulis dapat membuat dan menyelesaikan laporan Praktik Kerja Lapangan dengan judul “PENGEMBANGAN APLIKASI *PRIOR ADJUSTMENT* BERBASIS *WEB* MENGGUNAKAN *FRAMEWORK ANGULAR* PADA PT. ADICIPTA INOVASI TEKNOLOGI” yang merupakan salah satu prasyarat untuk memperoleh gelar Sarjana Komputer Universitas Ma Chung. Terdapat banyak sekali dukungan mulai dari awal hingga selesainya Praktik Kerja Lapangan ini, penulis mengucapkan banyak terima kasih kepada pihak-pihak yang membantu penulis hingga Praktik Kerja Lapangan terselesaikan dengan baik. Pihak-pihak tersebut diantaranya adalah:

1. Tuhan yang Maha Esa atas berkat dan karunia-Nya selama proses Praktik Kerja Lapangan hingga laporan ini terselesaikan.
2. Bapak Hendry Setiawan, ST., M.Kom. selaku Kepala Prodi Teknik Informatika Universitas Ma Chung.
3. Bapak Paulus Lucky Tirma Irawan, S.Kom., MT. selaku Dosen Pembimbing Praktik Kerja Lapangan yang telah memberi banyak bantuan, saran, dan semangat dalam proses penyelesaian laporan ini.
4. Bapak Welly Satria Nugraha selaku *Team Leader* divisi EGL2 yang membantu dan membimbing selama Praktik Kerja Lapangan.
5. Segenap karyawan PT Adicipta Inovasi Teknologi yang mendukung penulis dalam menyelesaikan pekerjaan yang diberikan dan memberikan motivasi kepada penulis dalam kegiatan Praktik Kerja Lapangan.
6. Universitas Ma Chung Malang yang telah memberikan kesempatan kepada Penulis untuk dapat memperoleh ilmu pengetahuan serta pengalaman dalam dunia kerja melalui kegiatan Praktik Kerja Lapangan.
7. Orang tua yang telah memberikan doa dan motivasi penulis dalam menyelesaikan Laporan Praktik Kerja Lapangan.
8. Teman-teman Teknik Informatika 2021 yang membantu selama proses pembelajaran di program studi Teknik Informatika.

Penulis menyadari bahwa laporan ini jauh dari sempurna yang disebabkan oleh hal seperti keterbatasan ilmu pengetahuan dari penulis. Oleh karena itu, penulis ingin mengucapkan permohonan maaf yang sebesar-besarnya apabila terdapat kesalahan baik dari ejaan ataupun kalimat. Penulis juga mengharapkan saran dan kritik yang positif dan membangun dari pembaca. Akhir kata, penulis berharap agar laporan ini bisa bermanfaat dan berguna untuk semua pihak dari berbagai kalangan yang membutuhkan.

Malang, 16 Januari 2025



Siusiu Trifena Halim



UNIVERSITAS
MA CHUNG

Daftar Isi

Kata Pengantar	i
Daftar Isi	iii
Daftar Tabel	v
Daftar Gambar	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Batasan Masalah	2
1.3 Rumusan Masalah	2
1.4 Tujuan Praktik Kerja Lapangan	2
1.5 Manfaat Praktik Kerja Lapangan	2
BAB II GAMBARAN UMUM PERUSAHAAN	4
2.1 Jenis Usaha Perusahaan	4
2.1.1 Jenis Produk dan Layanan PT Adicipta Inovasi Teknologi	5
2.2 Struktur Organisasi	7
2.3 Operasional Perusahaan	7
2.3.1 Visi	7
2.3.2 Misi	8
2.4 Tim EGL (<i>Electric General Ledger</i>)	8
BAB III TINJAUAN PUSTAKA	9
3.1 Dokumentasi Keuangan	9
3.2 <i>Chart of Account (COA)</i>	9
3.3 <i>Framework Angular</i>	10

3.4 .NET Core 6	11
3.5 C#	11
3.6 Visual Studio Code	12
3.7 Microsoft Visual Studio	13
3.8 Swagger	13
3.9 Microsoft SQL Server 2017	14
3.10 Rapid Application Development (RAD)	14
3.11 Flowchart	15
BAB IV HASIL PRAKTIK KERJA LAPANGAN	16
4.1 Prosedur Pelaksanaan Praktik Kerja Lapangan di PT Adicipta Inovasi Teknologi	16
4.2 Proses Pengerjaan Aplikasi	16
4.3.1 Perencanaan Kebutuhan	19
4.3.2 Persiapan Database	20
4.3.3 Pembuatan fungsi	24
4.3.4 Hasil <i>Front end</i>	25
BAB V KESIMPULAN DAN SARAN	39
5.1 Kesimpulan	39
5.2 Saran	40
Daftar Pustaka	42
Lampiran A Biodata Mahasiswa	45
Lampiran B Form Bimbingan Dosen Pembimbing	46
Lampiran C Lembar Penilaian Perusahaan	47
Lampiran D Lembar Partisipasi Seminar	48

Daftar Tabel

Table 4. 1 Table yang digunakan	20
Table 4. 2 Fungsi Back End Prior Adjustment	24



UNIVERSITAS
MA CHUNG

Daftar Gambar

Gambar 2. 1 Logo PT Adicipta Inovasi Teknonlogi.....	4
Gambar 2. 2 Logo CONFINS	5
Gambar 2. 3 Stuktur Organisasi PT Adicipta Inovasi Teknologi	7
Gambar 2. 4 Logo EGL.....	8
Gambar 3. 1 Pola Program C#	12
Gambar 3. 2 Tahap Pengembangan RAD	14
Gambar 4. 1 Alur Proses Pengerjaan Aplikasi <i>Prior Adjustment</i> 1	16
Gambar 4. 2 Alur Proses Pengerjaan Aplikasi <i>Prior Adjustment</i> 2	17
Gambar 4. 3 Alur <i>Prior Adjustment</i>	21
Gambar 4. 4 Alur <i>Prior Adjustment</i>	22
Gambar 4. 5 Tampilan Awal <i>Prior Adjustment</i>	25
Gambar 4. 6 <i>View Batch Journal</i>	26
Gambar 4. 7 Halaman <i>Edit Batch Prior Adjustment</i>	27
Gambar 4. 8 <i>Delete Batch</i>	28
Gambar 4. 9 Tampilan Halaman <i>Add Batch</i>	28
Gambar 4. 10 Tampilan Halaman Transaksi <i>Prior Adjustment</i>	29
Gambar 4. 11 Tampilan Halaman <i>Add Voucher</i>	30
Gambar 4. 12 Tampilan setelah data terisi	31
Gambar 4. 13 Tampilan <i>Add Detail Voucher</i>	32
Gambar 4. 14 Tampilan <i>Voucher</i> yang sudah memiliki <i>Detail</i>	33
Gambar 4. 15 Tampilan <i>Voucher Unbalanced</i>	34
Gambar 4. 16 Tampilan <i>Voucher Balanced</i>	34
Gambar 4. 17 Halaman Transaksi dengan <i>Voucher</i>	35
Gambar 4. 18 Halaman <i>Voucher Posted</i>	36
Gambar 4. 19 <i>Report Journal List</i>	37
Gambar 4. 20 <i>Report Journal Transaction</i>	38

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dokumentasi keuangan sangat penting untuk mengukur kinerja dan perkembangan bisnis perusahaan yang berkelanjutan, Fahmi (2012). Tujuan utama dari dokumentasi keuangan adalah untuk memberikan informasi yang relevan untuk membuat keputusan bisnis dan ekonomi, sebagaimana dinyatakan dalam dokumen Pernyataan Konsep Akuntansi Keuangan (SFAC) No. 1. Penting untuk memiliki sistem yang dapat melakukan koreksi dan penyesuaian apabila terdapat perubahan nominal suatu pengeluaran atau pemasukan.

PT. Adicipta Inovasi Teknologi, melalui proyek EGL, menghadirkan solusi inovatif berupa aplikasi *Prior Adjustment*. Fitur ini dirancang untuk mempermudah perusahaan dalam mengelola perubahan nominal yang berkaitan dengan pemasukan dan pengeluaran, serta menghasilkan jurnal keuangan yang terintegrasi. Proses yang ada di *Prior Adjustment* ini, memungkinkan perusahaan untuk memasukkan jurnal mutasi perubahan meliputi pemasukan dan pengeluaran, lalu sistem akan otomatis melakukan pembaruan saldo dan membentuk jurnal laporan baru terkait perubahan yang dilakukan. Dengan begitu, perubahan mutasi keuangan lebih mudah disampaikan

Dengan adanya aplikasi ini, perusahaan dapat memaksimalkan akurasi dalam menyimpan data keuangan, meminimalkan kesalahan pencatatan laporan keuangan, meningkatkan efisiensi pengelolaan keuangan, serta menyampaikan mutasi perubahan saldo secara lebih terstruktur dan mudah dibaca pengguna.

Untuk mendukung implementasi yang efektif, penulis mengembangkan aplikasi berbasis *website* yang memfasilitasi dokumentasi mutasi perubahan. Aplikasi ini dirancang untuk memproses perubahan saldo dan menghasilkan jurnal keuangan secara otomatis, sehingga perusahaan dapat dengan mudah mendata dan melacak saldo secara real-time.

1.2 Batasan Masalah

Batasan masalah dari pembuatan aplikasi *Prior Adjustment* berbasis *website* adalah sebagai berikut.

1. *Website* yang dikerjakan penulis meliputi penginputan mutasi berbentuk jurnal *credit* dan *debit*, fitur untuk *create edit Delete Batch*, dan *view Voucher*.
2. *Website* dikerjakan menggunakan framework *Angular*.
3. *Database Management System* (DBMS) yang digunakan adalah *Microsoft SQL Server 2007*.

1.3 Rumusan Masalah

Bagaimana cara membuat aplikasi yang dapat digunakan untuk memudahkan pendokumentasian laporan keuangan sesuai dengan ketentuan produk di proyek EGL di PT. Adicipta Inovasi Teknologi.

1.4 Tujuan Praktik Kerja Lapangan

Tujuan dari pengembangan aplikasi *Prior Adjustment* di PT Adicipta Inovasi Teknologi adalah:

1. Mengetahui tahapan pengembangan sebuah aplikasi di PT Adicipta Inovasi Teknologi.
2. Memudahkan pendokumentasian meliputi pencatatan dan pembentukan jurnal dari adanya pergerakan keuangan.

1.5 Manfaat Praktik Kerja Lapangan

Manfaat dari praktik kerja lapangan di PT Adicipta Inovasi Teknologi adalah:

- a. Bagi Mahasiswa
 1. Mendapat pengalaman dan gambaran kerja secara nyata.
 2. Menambah wawasan mengenai prospek kerja sebagai alumni Teknik Informatika.

3. Sarana untuk menerapkan teori yang diperoleh selama masa perkuliahan sesuai kebutuhan perusahaan.
 4. Memperoleh bekal untuk masuk baik dalam kehidupan pekerjaan dan kehidupan sosial masyarakat.
 5. Melatih kemampuan analisis masalah secara langsung berdasarkan keadaan sesungguhnya di perusahaan.
 6. Memahami proses pengembangan aplikasi di PT Adicipta Inovasi Teknologi.
- b. Bagi Perusahaan
1. Sarana untuk menjalin hubungan kerja sama dengan Prodi Teknik Informatika Universitas Ma Chung baik secara akademis maupun organisasi.
 2. Sarana untuk mengetahui kualitas pendidikan di Program Studi Teknik Informatika.
 3. Memperoleh masukan/usulan untuk meningkatkan kinerja perusahaan.

UNIVERSITAS
MA CHUNG

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Jenis Usaha Perusahaan

PT. Adicipta Inovasi Teknologi (AdIns) adalah perusahaan yang menyediakan solusi bisnis keuangan yang bergerak di bidang teknologi informasi. PT. Adicipta Inovasi Teknologi (AdIns) pertama kali didirikan pada tanggal 21 April 2000 oleh Bapak Guntur Gozali selaku CEO dan salah satu pendiri AdIns. Perusahaan ini memiliki kantor pusat yang berada di Jakarta dan memiliki kantor cabang yang berada di Malang. Untuk kantor pusat yang berada di Jakarta terletak di Jalan Kebon Jeruk Raya No. 80, Jakarta Barat sedangkan kantor cabang yang berada di Malang terletak di Jalan Lembah Dieng No. 7, Sumberjo, Kalisongo, Dau, Malang.



Gambar 2. 1 Logo PT Adicipta Inovasi Teknonologi

PT. Adicipta Inovasi Teknologi (AdIns) telah menyediakan solusi untuk berbagai bisnis keuangan mulai dari pembiayaan klien hingga solusi *leasing*. Melihat peran pentingnya dalam industri keuangan, tidak mengherankan jika banyak pendiri PT. Adicipta Inovasi Teknologi pernah bekerja di perusahaan pembiayaan seperti *Astra Credit Company* (ACC). Sebelum memutuskan membangun perusahaan ini, Bapak Guntur Gozali sekaligus CEO dan salah satu pendiri AdIns pernah bekerja di ACC dan dua perusahaan lainnya.

Pada tahun 2000, Bapak Guntur Gozali yang memiliki pengalaman dan pendidikan di bidang teknologi informasi bersama dengan adik iparnya memutuskan untuk mendirikan AdIns. Seiring waktu, AdIns berhasil memberdayakan para profesional muda dan AdIns berupaya mengoptimalkan

produktivitas seluruh karyawan. Untuk memberikan layanan terbaik kepada klien, AdIns juga menawarkan berbagai layanan produk, salah satunya adalah CONFINS (Consumer Finance and Leasing Solution). Layanan produk CONFINS ini disediakan untuk *multifinance* yang berguna untuk mengelola data pribadi pelanggan. Layanan produk CONFINS juga merupakan layanan produk paling awal yang di produksi oleh AdIns.



Gambar 2. 2 Logo CONFINS

Keberhasilan produk CONFINS dan produk lainnya dari AdIns telah mencapai keberhasilan yang memuaskan. Produk-produk ini telah mendapatkan kepercayaan dari banyak pelanggan. Selain itu, AdIns telah menekankan pada pengembangan dan peningkatan layanan sehingga banyak orang yang mendapat manfaat dari layanan tersebut dan puas dengan kinerja CONFINS serta produk lainnya karena AdIns menjaga kualitas yang baik dalam memenuhi kebutuhan pelanggan.

Pada akhirnya, AdIns tidak pernah berhenti meningkatkan dan memperbarui layanannya untuk klien lama dan pelanggan baru. AdIns selalu menjaga komitmennya terhadap integritas dalam hal mengembangkan teknologi informasi yang lebih baik di Indonesia. Oleh karena itu, AdIns akan selalu menjaga kepercayaannya untuk menciptakan pasar pengembangan IT yang kompetitif untuk kepentingan klien.

2.1.1 Jenis Produk dan Layanan PT Adicipta Inovasi Teknologi

PT Adicipta Inovasi Teknologi menyediakan berbagai product *software*. Berikut beberapa produk dan layanan yang PT Adicipta Inovasi Teknologi berikan:

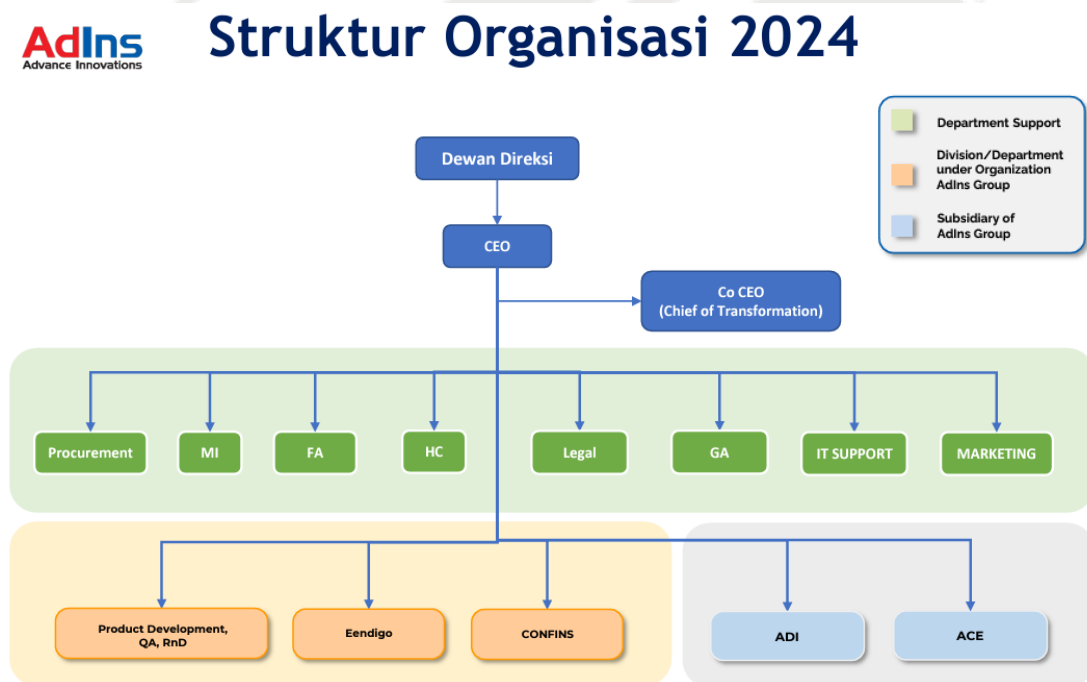
1. *Aplikasi Core Systems - CONFINS*: *CONFINS* adalah sistem inti yang dirancang untuk mengelola perusahaan pembiayaan dan data pribadi klien mereka. Sebagai salah satu layanan produk utama dari AdIns, *CONFINS* penting bagi perusahaan pembiayaan untuk memastikan pertumbuhan bisnis yang berkelanjutan dan produktivitas kerja yang optimal.
2. *Digital Signature*: Layanan Tanda Tangan Digital dari AdIns memungkinkan digitalisasi tanda tangan Anda, membantu mempercepat proses dokumen dengan lebih cepat dan aman, sehingga mendukung pertumbuhan bisnis.
3. *Optical Character Recognition (OCR)*: Aplikasi OCR berbasis Artificial Intelligence dari AdIns membantu Anda mengisi dokumen digital dengan mudah dan praktis, meningkatkan efisiensi dalam pemrosesan dokumen seperti KTP, NPWP, Kartu Keluarga, STNK, dan BPKB.
4. *Aplikasi Mobile Multifinance - AdIns Mobile*: AdIns Mobile adalah aplikasi multifinance berbasis mobile yang dirancang untuk memastikan operasional yang lebih lancar, pengawasan yang lebih baik, dan meningkatkan produktivitas karyawan.
5. *Aplikasi DMS (Document Management System) - Lite DMS*: Lite DMS adalah sistem manajemen dokumen yang memudahkan pencarian dokumen secara spesifik dan terenkripsi, memastikan keamanan dan kemudahan akses terhadap dokumen penting perusahaan.
6. *EKYC (Know Your Customer) System - PROFIND*: PROFIND adalah aplikasi yang membantu mencari profil calon debitur dalam proses kredit dengan lebih cepat, mudah, dan akurat, mendukung proses penilaian kredit yang lebih efisien.
7. *Monitor Network IT - SUPRBOARD*: SUPRBOARD adalah solusi pengelolaan jaringan dan kinerja aplikasi yang memungkinkan upaya pencegahan fluktuasi infrastruktur jaringan, memastikan stabilitas dan keandalan sistem IT perusahaan.
8. *Aplikasi Business Intelligence (ARS)*: ARS adalah sistem monitoring bisnis yang menyediakan dashboard untuk memberikan wawasan internal.

perusahaan, membantu dalam pengambilan keputusan yang lebih informed.

9. *IT Service*: AdIns menawarkan berbagai layanan seperti konsultasi strategi infrastruktur IT, manajemen IT, Service Desk, dan layanan keamanan jaringan, membantu bisnis Anda menjadi lebih efisien dan efektif.
10. *Jasa Data Entry - DOCUPRO*: DOCUPRO adalah layanan oleh tim profesional dengan kemampuan data entry dan verifikasi data dalam jumlah besar, memastikan akurasi dan efisiensi dalam pengelolaan data perusahaan

2.2 Struktur Organisasi

Berikut adalah struktur organisasi dari PT Adicipta Inovasi Teknologi:



Gambar 2. 3 Stuktur Organisasi PT Adicipta Inovasi Teknologi

2.3 Operasional Perusahaan

2.3.1 Visi

PT Adicipta Inovasi Teknologi memiliki visi untuk menjadi berkat bagi masyarakat melalui penerapan inovasi di bidang Teknologi Informasi dan

Komunikasi. Visi ini diwujudkan tidak hanya melalui pengembangan produk dan layanan bagi klien, tetapi juga dengan membangun budaya kerja yang saling mendukung dan kolaboratif di antara karyawan. Dengan berkomitmen pada solusi inovatif dan berkelanjutan, perusahaan terus berupaya menciptakan produk yang meningkatkan efisiensi kerja dan memberikan dampak positif yang nyata bagi masyarakat luas.

2.3.2 Misi

1. Mendorong dan mendukung karyawan untuk bertumbuh
2. Menjadi mitra andalan dengan solusi terbukti menggunakan inovasi kelas dunia di bidang teknologi *informasi* dan komunikasi
3. Menjadi aset negara berdasarkan Pancasila
4. Memberikan imbal balik terbaik kepada pemegang saham

2.4 Tim EGL (*Electric General Ledger*)

Tim EGL merupakan salah satu tim yang berada di bawah divisi *Product Development Core System*. Tim EGL ini bertugas untuk menjaga produk CONFINS R3 *Core* dengan melakukan *bug fixing*, *enhancement*, dan penambahan fitur-fitur untuk produk tersebut.



Gambar 2. 4 Logo EGL

BAB III

TINJAUAN PUSTAKA

3.1 Dokumentasi Keuangan

Dokumentasi keuangan merupakan suatu informasi yang menggambarkan kondisi keuangan suatu perusahaan. Isi dari dokumentasi keuangan tersebut dapat dijadikan sebagai gambaran kinerja keuangan perusahaan tersebut. Dokumentasi keuangan digunakan untuk mengukur hasil usaha dan perkembangan perusahaan, serta mengukur posisi perusahaan dalam mencapai tujuan (Fahmi, 2012).

3.2 *Chart of Account (COA)*

Bagan akun atau yang biasa disebut *chart of account* adalah kumpulan kode dan nama akun yang disusun menggunakan simbol huruf, angka, atau gabungan keduanya secara sistematis dengan tujuan memudahkan pencatatan dan pemrosesan data agar lebih mudah dalam pelaporan (Habibi dan Supriatna, 2021). *Chart of account* memiliki struktur dengan tingkat -tingkat kode akun yang dapat dibagi menjadi beberapa bagian yaitu lima Tingkat kodifikasi dengan angka-angka yang dipisahkan oleh titik (Ribeiro dan Prativiera, 2014).

Simkin et al. (2014) membahas struktur *chart of account* yang baik membantu dalam mengelompokkan transaksi keuangan secara logis, sehingga mempermudah pelacakan informasi akuntansi dan pengambilan keputusan strategis. Dengan pendekatan yang terorganisasi, setiap entitas keuangan dapat dikelola secara akurat, mencakup pendapatan, pengeluaran, aset, kewajiban, dan ekuitas.

Selain itu, *Chart of Account* juga berfungsi sebagai alat kontrol internal untuk memastikan bahwa setiap transaksi dicatat dalam kategori yang benar (Romney dan Steinbart, 2018). Hal ini sangat penting bagi organisasi dengan sistem keuangan yang kompleks, karena bagan akun yang dirancang dengan baik dapat meningkatkan efisiensi dalam pelaporan keuangan dan mengurangi potensi kesalahan.

3.3 *Framework Angular*

Istilah *framework* pada umumnya merujuk ke kumpulan standar, konvensi, dan daftar (*library*) yang membantu *developer* untuk mengembangkan aplikasi berbasis *web* dengan lebih efisien (Bielak dkk., 2022). *Angular* merupakan salah satu *framework* yang memudahkan *developer* untuk mengembangkan aplikasi *web*. Salah satu fitur utama *Angular* yaitu penggunaan HTML sebagai *template* dengan fungsionalitas khusus. Sistem modul di *Angular* memungkinkan *developer* untuk memecah code menjadi modul terpisah, sehingga mempermudah pemeliharaan aplikasi yang besar. Modul seperti *NgModule* dalam *Angular* dapat digunakan untuk mengimpor fungsi dan komponen yang diperlukan dalam pengembangan aplikasi (Kirana dan Zukri, 2023).

Angular merupakan generasi terbaru dari *Angular JS* dengan berbagai peningkatan. *Angular JS* mendorong pemisahan data dengan antarmuka menggunakan bahasa JavaScript, sedangkan *Angular* dibangun menggunakan bahasa *TypeScript* yang berbasis komponen sehingga memungkinkan adanya penempelan data atau data *binding*. Performa *Angular* diukur 7 kali lebih cepat dibandingkan *Angular JS* karena dapat memanipulasi struktur *DOM* (*Document Object Model*) secara langsung tanpa membuat dokumen HTML terlebih dahulu (Bielak dkk., 2022).

Angular menggunakan konsep interpolasi untuk menampilkan properti-properti komponen dalam *template-view*. Misalkan atribut nama yang akan ditampilkan harus ada antara kurung ganda ‘{}’ atau disebut sintaks *moustache*. Selain itu, data binding dalam *Angular* ada dua jenis: satu arah dan dua arah. Jenis dua arah memungkinkan nilai *property* komponen diperbarui secara otomatis (Sahani dkk., 2020). Kelebihan menggunakan *Angular* mencakup kemampuan pembuatannya yang lebih mudah karena memberikan metode yang sistematis dan efektif untuk membuat aplikasi *web* interaktif. Selain itu, *Angular* juga didukung oleh koleksi *library* yang luas sehingga dapat mejangkau berbagai kebutuhan fitur dalam pengembangan aplikasi *web*. Untuk mengguakan *Angular*, *developer* perlu melakukan persiapan lingkungan lokal untuk *CLI* (*Comand Line Interface*)

Angular. Ini termasuk instalasi CLI, penciptaan *workspace*, hingga *setting-up* prasyarat dasarnya (Purohit dkk., 2023).

3.4 .NET Core 6

Arora dan Dash (2018) mendefinisikan *.NET Core* sebagai *platform* untuk membuat aplikasi *web*, *micro service*, *libraries*, dan aplikasi yang berjalan di *Windows*, *Linux*, dan *macOS*. *ASP.NET Core* adalah *framework open-source* yang merupakan generasi berikutnya dari *ASP.NET*. *Framework* ini adalah rangkaian kerja modular yang terdiri dari package dari komponen *framework* yang berjalan pada *.NET Framework*, *Windows*, dan *.NET Core cross-platform*.

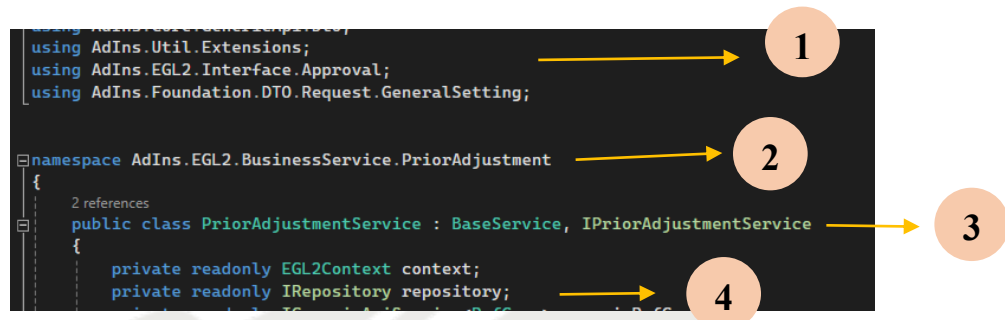
ASP.NET Core menyatukan *ASP.NET MVC* (Model View Controller). dan *ASP.NET Web API* yang sebelumnya terpisah menjadi satu model pemrograman. *ASP.NET Web API* dibuat untuk memetakan model pemrograman *web/HTTP* ke model pemrograman *.NET Framework*. API ini menggunakan konstruksi yang sudah dikenal, seperti *Controller*, *Action*, *Filter*, dan sebagainya, yang digunakan dalam *ASP.NET MVC* (Model View Controller).

Framework .Net Core 6.0 merupakan *framework* yang menggunakan bahasa pemrograman C# menggunakan aplikasi *Visual Studio*. *Framework* ini memudahkan sisi pengembang untuk membangun suatu sistem dan memiliki fitur tambahan yaitu penggunaan *library global* yang dilakukan secara implisit. Sehingga, selain memudahkan pembuatan program, juga dapat mempermudah proses *debugging* untuk mencari *error* (Griffiths, 2022).

3.5 C#

Bahasa C# adalah sebuah bahasa pemrograman *modern* yang bersifat general, berorientasi objek, yang dapat digunakan untuk membuat program di atas arsitektur *Microsoft .NET Framework*. Bahasa C# ini memiliki kemiripan dengan bahasa *Java*, C dan C++. Bahasa pemrograman ini dikembangkan oleh sebuah tim pengembang di *Microsoft* yang dipimpin oleh Anders Hejlsberg, seorang yang mengembangkan bahasa pemrograman *Borland Turbo Pascal*, *Borland Delphi*, dan juga *Microsoft J++* (Sanjaya dkk., 2021).

Menurut Miles (2016) tentang kemiripan bahasa pemrograman C# dapat disimpulkan bahwa bahasa C# adalah bahasa gabungan yang kuat antara C++ dan *Java*. Kelebihan utama bahasa ini adalah sintaks yang mirip bahasa pemrograman C, namun lebih mudah dan bersih (Rachmatullah., 2002).



Gambar 3. 1 Pola Program C#

Gambar diatas merupakan pola program aplikasi C#. Bagian 1 merupakan kumpulan *library* yang digunakan. Lalu pada bagian 2, adalah *namespace* untuk mengorganisasi program agar teratur. Bagian nomor 3 adalah kelas untuk menyimpan informasi sifat sifat kode program. Bagian 4 yaitu kata *Main()* merupakan kata kunci dimana aplikasi C# dipanggil melalui *method* ini.

3.6 Visual Studio Code

Menurut Faisal (2017), *Visual Studio Code* adalah *source code editor* yang dapat digunakan di berbagai platform seperti *Windows*, *Linux*, dan *Mac OS*. Aplikasi *Visual Studio* muncul dengan built-in untuk bahasa pemrograman *JavaScript*, *TypeScript* dan *Node.JS* dan memiliki ekosistem ekstensi yang beragam untuk bahasa lain (seperti C++, C#, *Java*, *Python*, *PHP*) dan runtime (seperti *.NET* dan *Unity*) (Romzi dan Kurniawan, 2020).

Fitur utama dari aplikasi ini yaitu *Intelligent Code Completion* dimana fitur ini dapat membantu pengembang untuk melengkapi kode, metode, fungsi, nama *variable*, dan *framework* yang digunakan. Terdapat juga fitur *Streamlined debugging* yaitu fitur untuk melakukan *debugging* dengan menambahkan *breakpoint*. Fitur lainnya yaitu *Linters*, *Multi-cursor Editing*, *Parameter Hints*, *Code Navigation*, *Refactoring*, dan dukungan akses *Git* (Faisal, 2017).

3.7 *Microsoft Visual Studio*

Microsoft Visual Studio merupakan *Integrated Development Environment* (IDE) yang dikembangkan oleh *Microsoft*. Aplikasi ini dapat digunakan untuk mengembangkan sebuah aplikasi, baik aplikasi bisnis, personal, atau komponen aplikasi berbentuk *console*, *mobile*, dan atau *web* (Faisal, 2017). Kompiler di dalam *Microsoft Visual Studio* antara lain *C++*, *C#*, *Visual Basic*, *Visual Basic .NET*, *Visual InterDev*, *Visual J++*, *Visual J#*, *Visual FoxPro*, dan *Visual SourceSafe*. *Microsoft Visual Studio* dapat digunakan dalam bentuk *native code* (bahasa yang berjalan diatas *Microsoft*) ataupun *managed code* (bahasa yang berjalan di atas *.NET Framework*) (Ruli, 2017).

Visual Studio tersedia dalam beberapa versi, versi gratis yaitu *community edition* yang ditujukan untuk individu, pengembang *open-source*, dan tim kecil. Lalu ada versi berbayar yaitu *professional* dan *enterprise editions* yang menawarkan banyak fitur termasuk alat analisis dan dukungan tambahan untuk tim besar (Amann dkk., 2016)

3.8 *Swagger*

Swagger adalah alat yang dapat digunakan untuk keperluan dokumentasi, dan menyediakan semua informasi mengenai *end-point* API pada satu layar, tempat Anda dapat memvisualisasikan API dan mengujinya dengan mengirimkan *parameter* juga. *Swagger* digunakan untuk mendokumentasikan REST API. *Swagger* bersifat *real time* sehingga klien dan sistem dokumentasi bergerak dengan kecepatan yang sama dengan *server*. Deskripsi metode, *parameter*, dan model terintegrasi erat ke dalam kode *server*, sehingga sinkronisasi API dan dokumentasinya dapat terjaga (Sari dan Hidayat, 2022).

Selain itu, *Swagger* memungkinkan pengembang untuk mempercepat proses pengembangan dengan menyediakan antarmuka yang interaktif untuk pengujian langsung. Fitur-fitur seperti *Swagger UI* membantu dalam eksplorasi *endpoint* API dengan mudah, sementara dokumentasi yang dihasilkan secara otomatis memastikan bahwa setiap perubahan pada API langsung tercermin pada

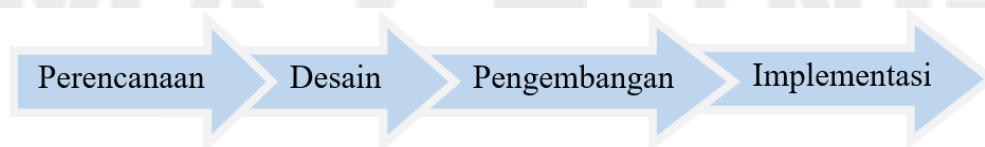
dokumentasi. Hal ini menjadikan *Swagger* sebagai solusi yang andal untuk pengelolaan API modern (Bermbach dkk., 2017)

3.9 *Microsoft SQL Server 2017*

Microsoft SQL Server adalah sebuah Sistem Manajemen Basis Data yang dibuat *Microsoft*. Bahasa query yang digunakan pada aplikasi ini adalah *Transact-SQL* yang berasal dari implementasi *SQL* standar ANSI/ISO. Aplikasi *Microsoft SQL Server 2017* memiliki beberapa kelebihan yaitu sistem basis data pada *level enterprise* yang lebih mudah digunakan dan memiliki banyak fitur termasuk integrasi dengan *.NET framework*. Aplikasi ini juga memiliki fitur restorasi dan pemulihan data untuk data yang *corrupt*. Dibalik kelebihannya, aplikasi ini memiliki kekurangan yaitu biaya lisensi yang cukup mahal dan kompatibilitas yang terbatas dengan server *Windows* (Subagia, 2017).

3.10 *Rapid Application Development (RAD)*

Metode pengembangan aplikasi yang digunakan yaitu *RAD (Rapid Application Development)*. Wahyuningrum dan Januarita (2014) menyatakan bahwa metode ini adalah sebuah model proses perkembangan perangkat lunak yang menekankan siklus perkembangan yang sangat pendek. Dalam sistem RAD, menggunakan metode perulangan dalam mengembangkan sistem dimana model bekerja dikonstruksikan di awal tahap pengembangan dengan tujuan menetapkan kebutuhan pengguna. Berikut tahapan pengembangan metode *Rapid Application Development (RAD)*:



Gambar 3. 2 Tahap Pengembangan RAD

1. Perencanaan Kebutuhan

Pengguna dan *analyst* mengidentifikasi tujuan sistem dan kelengkapan informasi untuk mencapai tujuan.

2. Desain Sistem

Pada tahaan desain sistem, dilakukan penyesuaian terhadap dokumentasi desain yang sudah ada. Pada tahapan ini, menghasilkan spesifikasi software yang meliputi organisasi sistem dan struktur data

3. Pengembangan dan Implementasi.

Tahap ini merupakan tahap dimana *developer* mengembangkan program yang sudah direncanakan. Sebelum di aplikasikan, dilakukan pengujian terhadap program sesuai ketentuan yang ada pada desain sistem.

3.11 *Flowchart*

Flowchart, atau diagram alir, adalah representasi grafis yang menggambarkan langkah-langkah dan urutan prosedur dalam suatu proses atau sistem. *Flowchart* membantu analis dan programmer memecah masalah menjadi bagian-bagian yang lebih kecil, memudahkan analisis alternatif operasional, dan menyederhanakan pemahaman alur kerja. *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program (Zalukhu dkk., 2023)

Fungsi utama *flowchart* adalah memberikan gambaran visual tentang proses yang memudahkan pemahaman dan analisis. Malabay (2016) menyatakan bahwa *flowchart* sangat membantu dalam memberikan gambaran suatu proses produksi agar mudah dipahami dan dilihat berdasarkan urutan langkahnya dari satu proses ke proses lainnya.

Secara keseluruhan, *flowchart* adalah alat yang efektif dalam perancangan dan analisis sistem, membantu dalam memvisualisasikan proses, mengidentifikasi potensi perbaikan, dan memastikan bahwa semua pihak memiliki pemahaman yang sama tentang alur kerja yang sedang dianalisis atau dikembangkan.

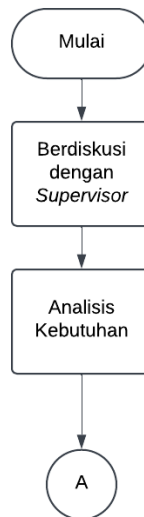
BAB IV

HASIL PRAKTIK KERJA LAPANGAN

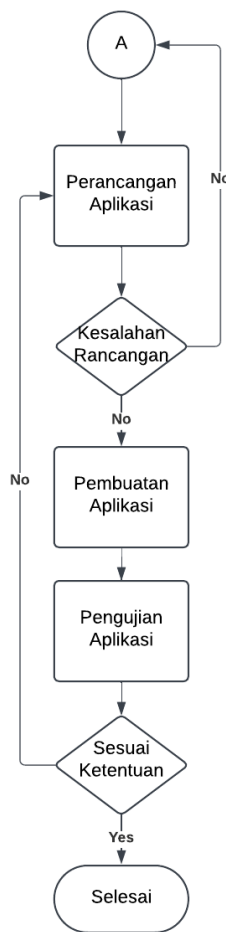
4.1 Prosedur Pelaksanaan Praktik Kerja Lapangan di PT Adicipta Inovasi Teknologi

Praktik Kerja Lapangan (PKL) dilaksanakan di PT Adicipta Inovasi Teknologi yang terletak di Jl. Lembah Dieng No.A1-7, Sumberjo, Kalisongo, Kec. Dau, Kabupaten Malang, Jawa Timur 65151. Praktik Kerja Lapangan (PKL) dilakukan dengan 5 hari kerja dari Senin hingga Jumat selama 6 bulan. Kegiatan Praktik Kerja Lapangan (PKL) dimulai pada tanggal 1 Juli 2024 dan berakhir pada 20 Februari 2025 dengan jam kerja selama delapan jam dimulai pukul 08.30 WIB dan berakhir pukul 17.30 WIB dengan istirahat pada pukul 12.00 sampai 13.00 WIB. Departemen tempat dilaksanakannya Praktik Kerja Lapangan (PKL) adalah *Project Development*.

4.2 Proses Pengerjaan Aplikasi



Gambar 4. 1 Alur Proses Pengerjaan Aplikasi *Prior Adjustment 1*



Gambar 4. 2 Alur Proses Pengerjaan Aplikasi *Prior Adjustment 2*

Gambar 4.1 dan Gambar 4.2 menunjukkan proses pengerjaan aplikasi *Prior Adjustment*. Pengerjaan dimulai dengan tahapan perencanaan awal. Langkah pertama adalah mengidentifikasi kebutuhan dan tujuan utama dari aplikasi yang akan dibuat. Untuk itu, dilakukan diskusi bersama pembimbing lapangan guna mendapatkan pemahaman yang jelas mengenai fitur-fitur yang diperlukan dan ketentuan yang harus dipenuhi oleh aplikasi.

Selanjutnya, hasil diskusi tersebut menjadi dasar untuk melakukan analisis kebutuhan. Pada tahap ini, kebutuhan fungsional dan non-fungsional dari aplikasi dirumuskan secara rinci, termasuk identifikasi fitur seperti *Add Batch*, *Add Voucher*, *Add Voucher Detail*, *Edit Voucher Detail*, *Add Prior Adjustment*, serta mekanisme pengelolaan jurnal seperti jurnal *Tax*, *Forex*, dan *Prior Adjustment*.

Selain itu, dilakukan juga penentuan alur kerja sistem untuk memastikan efisiensi dalam proses pencatatan dan pelaporan data.

Setelah analisis kebutuhan selesai, langkah berikutnya adalah perancangan aplikasi. Tahap ini mencakup pembuatan desain *interface* (tampilan pengguna), perancangan alur logika sistem, serta struktur database yang diperlukan. Proses ini dilakukan secara teliti untuk memastikan bahwa rancangan memenuhi kebutuhan bisnis yang telah dianalisis sebelumnya.

Setelah rancangan selesai, dilakukan pengecekan kesalahan terhadap rancangan yang telah dibuat. Jika ditemukan kesalahan atau rancangan belum memenuhi kriteria yang ditentukan, proses kembali ke tahap perancangan aplikasi untuk dilakukan revisi. Tahap ini bersifat iteratif, di mana pengecekan dan revisi dilakukan hingga rancangan benar-benar sesuai dengan spesifikasi yang diinginkan.

Jika rancangan telah disetujui, proses berlanjut ke pembuatan aplikasi. Pada tahap ini, aplikasi mulai dikembangkan berdasarkan rancangan yang telah disetujui, dengan melibatkan pengkodean, integrasi database, serta implementasi fitur yang telah dirancang sebelumnya.

Setelah aplikasi selesai dibuat, dilakukan pengujian aplikasi untuk memastikan bahwa semua fitur bekerja sesuai dengan ketentuan. Pada tahap pengujian, aplikasi diuji dari berbagai aspek, seperti fungsionalitas, keamanan, dan kehandalan sistem. Jika ditemukan bahwa aplikasi tidak sesuai dengan ketentuan, proses akan kembali ke tahap perancangan aplikasi untuk dilakukan perbaikan dan penyesuaian.

Apabila hasil pengujian menunjukkan bahwa aplikasi telah sesuai dengan kebutuhan dan memenuhi semua ketentuan yang berlaku, maka proses pengerjaan dianggap selesai. Aplikasi siap untuk digunakan, dan dokumentasi lengkap mengenai fitur serta alur kerja aplikasi diserahkan sebagai referensi untuk pengguna dan pengelola sistem.

4.3.1 Perencanaan Kebutuhan

Langkah pertama yang dilakukan penulis sebelum memulai pembuatan aplikasi *Prior Adjustment* selama Praktik Kerja Lapangan di PT Adicipta Inovasi Teknologi adalah dengan melaksanakan analisis kebutuhan secara menyeluruh. Proses ini dimulai dengan melakukan konsultasi bersama pembimbing lapangan untuk memahami secara mendalam fitur-fitur apa saja yang diperlukan dalam aplikasi. Konsultasi ini bertujuan untuk memastikan bahwa aplikasi yang dibuat nantinya tidak hanya memenuhi kebutuhan bisnis tetapi juga mematuhi standar dan ketentuan yang berlaku di perusahaan. Penulis bersama supervisi mendiskusikan alur kerja, proses bisnis, serta skenario penggunaan yang relevan untuk aplikasi *Prior Adjustment*.

Setelah proses konsultasi selesai dilakukan, penulis menyimpulkan beberapa kebutuhan utama dan langkah yang harus ditempuh untuk pengembangan aplikasi. Salah satu poin penting yang diidentifikasi adalah perlunya pengembangan aplikasi dengan beberapa fitur utama, yaitu fitur *form*, fitur *Delete*, fitur *view*, dan fitur *report*. Setiap fitur ini dirancang untuk mendukung fungsionalitas utama aplikasi. Misalnya, fitur *form* digunakan dalam berbagai tahapan proses, termasuk *Add Batch*, *Add Voucher*, *Add Voucher Detail*, dan *Edit Voucher Detail*. Fitur ini memungkinkan pengguna untuk menambahkan atau memperbarui data dengan mudah sesuai kebutuhan.

Selanjutnya, fitur *Delete* dirancang untuk memberikan fleksibilitas kepada pengguna dalam menghapus data *batch* yang sudah tidak diperlukan. Proses penghapusan ini juga dilengkapi dengan validasi atau konfirmasi untuk mencegah penghapusan data secara tidak sengaja. Sementara itu, fitur *view* memungkinkan pengguna untuk melihat *detail batch*, seperti informasi status, *detail voucher*, dan jurnal yang terkait dengan transaksi *Prior Adjustment*. Fitur ini sangat penting dalam memberikan transparansi dan kemudahan akses informasi kepada pengguna.

Terakhir, fitur *report* menjadi bagian yang tidak kalah penting dalam aplikasi. Fitur ini mendukung proses pencetakan laporan, seperti *Print Voucher* dan *Print Voucher Detail*, yang berfungsi untuk mendokumentasikan transaksi

secara resmi. Laporan yang dihasilkan akan mencakup detail transaksi, *balance*, dan jurnal yang telah terbentuk, sehingga dapat digunakan sebagai referensi atau arsip untuk keperluan audit dan evaluasi. Dengan perencanaan yang matang dan fitur yang dirancang sesuai kebutuhan, aplikasi *Prior Adjustment* diharapkan dapat menjadi solusi yang efektif dan efisien bagi perusahaan.

4.3.2 Persiapan Database

Pengembangan aplikasi dimulai dengan persiapan *database* untuk menampung data yang digunakan saat mengoperasikan menu *Prior Adjustment*. Pembuatan *database* dilakukan dengan *script Base Entity* di *ASP.NET* menggunakan aplikasi *Visual Studio 2022*.

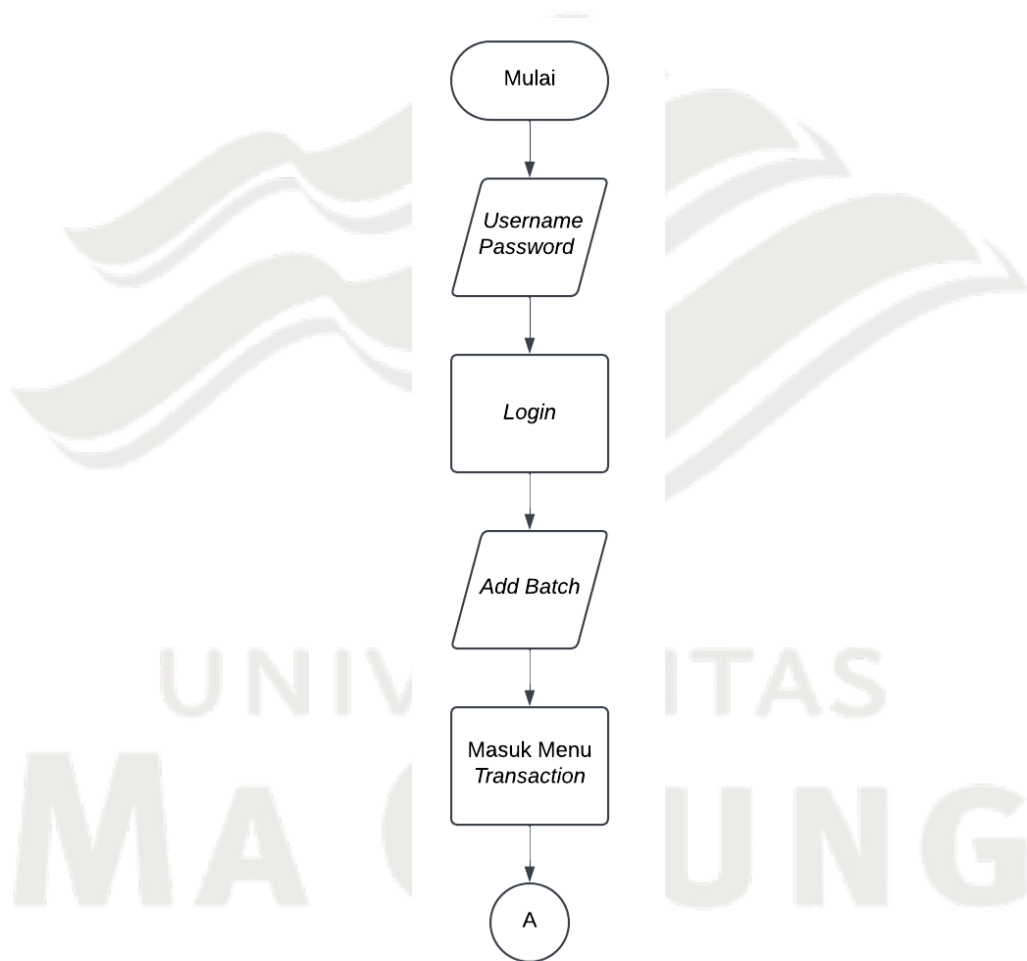
Table 4. 1 *Table* yang digunakan

Table	Keterangan
REF_COA	Berisi list COA yang digunakan
COA_BALANCE	Berisi saldo dari setiap COA
COA_BALANCE_MAINTENANCE	Berisi mutasi dari transaksi sebelumnya
JR_BATCH	Berisi seluruh transaksi jurnal
JR_VOUCHER_H	Berisi data jurnal secara umum
JR_VOUCHER_D	Berisi detail jurnal beserta mutasi

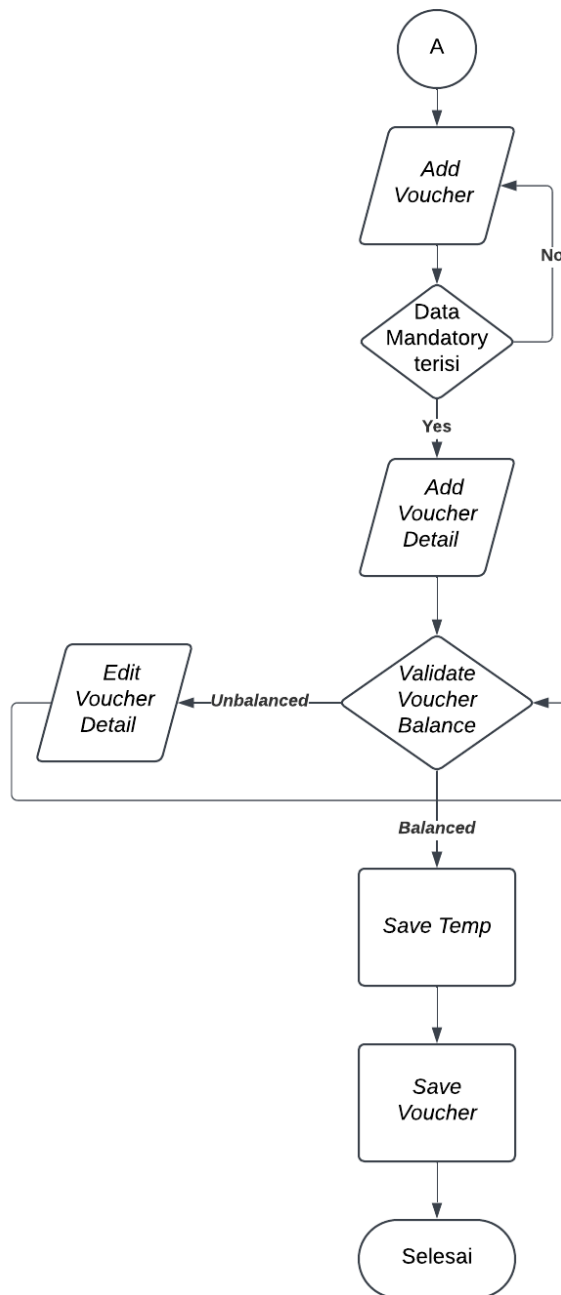
Tabel REF_COA berfungsi untuk menyimpan data bagan *Chart of Accounts (COA)* yang memiliki pengaturan tertentu. Tabel ini menjadi acuan utama dalam sistem isi, di mana setiap akun dalam *COA* didefinisikan dengan atribut yang relevan, seperti kode *COA*, nama *COA*, tipe *COA* (debit atau kredit), serta status aktif atau tidaknya akun tersebut. Dengan adanya tabel ini, pengguna dapat dengan mudah mengelola dan mengatur struktur keuangan perusahaan.

Tabel COA_BALANCE menyimpan informasi mengenai saldo dari masing-masing akun dalam *COA*, baik untuk transaksi debit maupun kredit. Setiap entri dalam tabel ini mencerminkan posisi keuangan terkini dari setiap akun pada periode tertentu. Informasi yang terdapat dalam tabel ini sangat penting untuk analisis keuangan dan pelaporan, karena memberikan gambaran jelas tentang kesehatan finansial perusahaan.

Tabel COA_BALANCE_MAINTENANCE digunakan untuk mendokumentasikan jumlah (*amount*) sebelumnya dari setiap akun berdasarkan COA_BALANCE_ID. Tabel ini berfungsi sebagai catatan historis yang memungkinkan pengguna untuk melacak perubahan saldo dari waktu ke waktu. Dengan adanya dokumentasi ini, perusahaan dapat melakukan audit dan analisis lebih mendalam mengenai fluktuasi saldo akun serta memastikan transparansi dan akurasi dalam laporan keuangan



Gambar 4. 3 Alur *Prior Adjustment*



Gambar 4. 4 Alur *Prior Adjustment*

Alur melakukan *Prior Adjustment* diawali dengan pengguna yang perlu membuat *Batch* terlebih dahulu. *Batch* yang baru dibuat akan terdapat pada table JR_BATCH dengan status *open*. Setelah *batch* sudah terbuat, pengguna dapat melakukan transaksi *Prior Adjustment*.

Selanjutnya pengguna dapat masuk ke halaman transaksi, lalu menambahkan *voucher Prior Adjustment*. Setelah mengisi *form voucher*, akan muncul tombol *Add Voucher Detail*. Pada *voucher detail* terdapat Prinsip Keseimbangan Dua Sisi (*Double-Entry Accounting*). Prinsip ini menyatakan bahwa setiap transaksi yang tercatat dalam sistem akuntansi harus selalu memiliki keseimbangan antara debit dan kredit. Dengan kata lain, untuk setiap debit yang tercatat, harus ada kredit yang sesuai dan sebaliknya (Horngren dkk., 2006). Hal ini memastikan bahwa laporan keuangan yang dihasilkan mencerminkan posisi keuangan perusahaan yang akurat dan tidak terjadi kesalahan dalam pencatatan ini tetap berlaku.

Setiap penyesuaian yang dilakukan harus memastikan bahwa jumlah debit dan kredit pada setiap *voucher detail* adalah seimbang. Ketidakseimbangan dalam jumlah ini dapat menyebabkan kesalahan dalam perhitungan saldo dan menghasilkan laporan keuangan yang tidak akurat. Selain keseimbangan antara debit dan kredit, penting juga untuk memastikan bahwa *balance amount* pada setiap *voucher detail* yang diinput sesuai dengan total transaksi yang tercatat. Proses ini menjaga integritas laporan keuangan dengan memastikan bahwa jumlah total yang tercatat untuk setiap akun dalam *voucher detail* tidak mengalami kesalahan perhitungan.

Setelah pengguna menambahkan *voucher detail*, dapat dilakukan pengecekan kembali dengan klik tombol *Validate* untuk memastikan keseimbangan *voucher* sesuai. Setelah berhasil melakukan *validate*, akan muncul tombol *Save Temp* untuk kembali ke halaman awal transaksi. Setelah itu pengguna bisa melakukan *Save Voucher* untuk submit transaksi *Prior Adjustment*. Setelah berhasil melakukan *Prior Adjustment*, status dari *batch* di JR_BATCH akan menjadi *posted*. Setelah status menjadi *posted*, akan terbentuk jurnal baru seperti jurnal *Tax*, jurnal *Forex*, dan jurnal *Prior Adjustment*. Jurnal yang terbentuk akan masuk ke table JR_VOUCHER_H dan JR_VOUCHER_D.

Setelah *batch* berstatus *posted*, pengguna tidak dapat melakukan transaksi perubahan atau penyesuaian lebih lanjut pada *batch* tersebut. Hal ini bertujuan untuk menjaga integritas dan keakuratan data akuntansi. Dengan status *posted*,

transaksi telah tercatat secara final dan resmi, sehingga memastikan bahwa seluruh proses pencatatan keuangan telah selesai dengan baik. Kebijakan ini juga mencegah potensi kesalahan yang dapat terjadi jika transaksi masih dapat diubah setelah proses *posting* selesai, menjaga kestabilan dan keamanan dalam sistem akuntansi perusahaan.

4.3.3 Pembuatan fungsi

Setelah tabel *database* siap digunakan, selanjutnya dilakukan pembuatan fungsi fungsi yang digunakan. Berikut fungsi fungsi dari setiap tabel:

Table 4. 2 Fungsi *Back End Prior Adjustment*

No	Function	Keterangan
1	<i>Add Prior Adjustment</i>	Validasi data ke <i>table</i> master Perubahan saldo pada <i>table</i> COA_BALANCE Perubahan saldo pada <i>table</i> COA_BALANCE_MAINTENANCE Pembuatan jurnal <i>Prior Adjustment</i> Pembuatan jurnal <i>Forex</i> Penambahan saldo ke <i>COA Forex</i> Pembuatan jurnal <i>Tax</i> Penambahan saldo ke <i>COA Tax</i>
2	<i>Add Batch</i>	<i>Add Batch</i> untuk melakukan transaksi <i>Prior Adjustment</i>
3	<i>Edit Batch</i>	Mengubah data pada <i>Batch</i> yang dipilih
4	<i>Delete Batch</i>	Mengubah status <i>Batch</i> menjadi <i>Closed</i>
5	<i>View Batch</i>	Menampilkan <i>detail</i> data <i>Batch</i>
6	<i>Validate</i>	Validasi <i>Post code</i> dan <i>Balance</i>

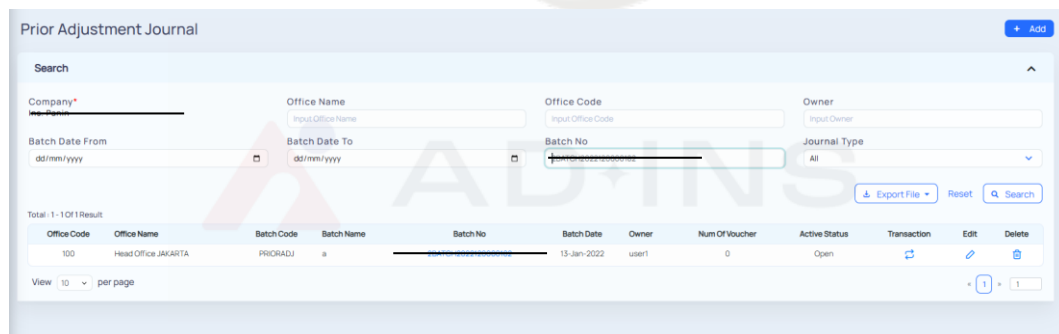
Pada fungsi *back end Add Prior Adjustment*, menerima beberapa *request object* dari *form* di tampilan *front end*. Pada fungsi *Add Prior Adjustment*, terdapat beberapa tahapan. Pada tahapan awal, dilakukan validasi data *request* terhadap

tabel *master*. Lalu dilakukan perubahan terhadap saldo *base amount* dan *origin amount* di tabel COA_BALANCE berdasarkan inputan COA dan waktu yang dipilih.

Terdapat juga perubahan pada table COA_BALANCE MAINTENANCE bagian *base prior* dan *origin prior*. Setelah itu terbentuk jurnal *Prior Adjustment*, jurnal *Tax* dan jurnal *Forex*. Fungsi *Add Prior Adjustment* dikembangkan menggunakan bahasa pemrograman C# dan LINQ dalam aplikasi *Visual Studio 2022*

Fungsi *Validate* bertujuan untuk memeriksa keseimbangan antara *post code* dan *balance* pada *voucher detail* yang ditambahkan ke dalam *voucher prior*. Fungsi ini bekerja dengan mengambil data *post code* dan *balance* dari halaman formulir di *front-end*, kemudian membandingkan jumlah *post code* dan *balance* tersebut untuk memastikan bahwa keduanya seimbang. Proses validasi ini penting untuk menjaga integritas data dan memastikan bahwa setiap *voucher prior* yang dibuat telah memenuhi aturan akuntansi yang berlaku.

4.3.4 Hasil Front end



Gambar 4. 5 Tampilan Awal *Prior Adjustment*

Gambar 4.5 merupakan tampilan awal halaman *Prior Adjustment*, yang dirancang untuk memudahkan pengguna dalam mencari dan mengelola *batch Prior Adjustment* yang tersedia. Pada halaman ini, pengguna dapat melakukan pencarian (*search*) untuk *batch Prior Adjustment* yang sudah terbentuk jurnal maupun yang belum memiliki transaksi. Halaman ini menyediakan berbagai opsi filter pencarian yang dapat digunakan untuk mempersempit hasil pencarian

Selain fungsi pencarian, halaman ini juga dilengkapi dengan tombol *Add*. Tombol ini berfungsi sebagai akses untuk menuju halaman *Add Batch Prior Adjustment*, di mana pengguna dapat membuat *batch Prior Adjustment* baru. Halaman ini dirancang untuk memberikan antarmuka yang intuitif, sehingga mempermudah pengguna dalam mengelola proses penyesuaian saldo. Kombinasi fitur pencarian yang komprehensif dan akses cepat ke fitur tambah *batch* menjadikan halaman ini sebagai titik awal yang efektif untuk pengelolaan *Prior Adjustment* dalam sistem.

Gambar 4.6 menunjukkan halaman *View Batch Journal*, yang muncul saat

26

Edit Prior Adjustment Journal

Main Information

Company Name

Ind. Pabrik

Batch No

BATCH-2022120600164

Batch Name *

PRIORADJ

Office Name

Head Office JAKARTA

Batch Date *

13/01/2022

Description *

test untuk prior

Cancel

Submit

Gambar 4.7 merupakan halaman Edit *Batch*, yang muncul saat pengguna mengklik ikon *edit* pada kolom *Edit* di halaman awal *Prior Adjustment*. Halaman ini memberikan akses bagi pengguna untuk memperbarui informasi *Batch Date* dan *Description* yang sebelumnya telah diinput. Fitur ini sangat berguna jika terdapat kesalahan data atau jika diperlukan penyesuaian setelah *batch* dibuat, sehingga fleksibilitas dalam pengelolaan *batch* dapat terjamin.

Selain itu, halaman ini dirancang dengan antarmuka yang sederhana dan intuitif, sehingga memudahkan pengguna dalam melakukan pembaruan tanpa risiko kebingungan. Setiap perubahan yang dilakukan pada halaman *Edit Batch* akan tersimpan secara otomatis dalam sistem, memastikan data tetap akurat dan *update*. Dengan adanya fitur ini, sistem menjadi lebih dinamis dan mampu mendukung kebutuhan pengguna dalam mengelola data *batch*.

Gambar 4.9 menunjukkan halaman *Add Batch*, yang memungkinkan pengguna untuk menambahkan *batch* baru ke dalam sistem. Pada halaman ini, pengguna diminta untuk mengisi informasi seperti *Office Name*, *Batch Date*, dan *Description* sebagai data wajib yang diperlukan untuk membuat *batch*. Setelah data diinput dengan lengkap, *batch* tersebut akan tersimpan dalam sistem dan muncul pada halaman awal *Prior Adjustment*, memberikan langkah awal yang jelas untuk melanjutkan proses transaksi.

Halaman *Add Batch* ini dirancang untuk memberikan kemudahan dan kenyamanan bagi pengguna dalam memulai proses penyesuaian saldo. Pengguna dapat menambahkan data *batch* secara efisien tanpa perlu melakukan langkah yang kompleks. *Batch* yang berhasil dibuat kemudian dapat digunakan untuk transaksi lebih lanjut, termasuk pembuatan jurnal *Prior Adjustment* yang diperlukan dalam sistem keuangan.

Gambar 4. 10 Tampilan Halaman Transaksi *Prior Adjustment*

Gambar 4.10 adalah tampilan awal dari halaman transaksi *Prior Adjustment* yang bisa diakses saat pengguna mengklik tombol *Transaction* pada halaman awal *Prior Adjustment*. Pada tahap ini, pengguna akan melihat bahwa belum terdapat transaksi *Prior Adjustment* yang terdaftar. Tampilan ini menunjukkan bahwa halaman masih kosong dan belum ada data *voucher* yang diinputkan atau diproses. Kondisi ini memberikan kesempatan bagi pengguna untuk memulai transaksi baru dengan menambahkan *voucher*.

Untuk memulai, pengguna harus mengklik tombol *Add Voucher* yang terletak di halaman tersebut. Tombol ini akan membawa pengguna ke *formulir input data voucher*, di mana mereka dapat memasukkan informasi yang dibutuhkan, seperti *Transaction*, *Voucher Date*, *Description*, *Notes*, *Reference No*, dan *Reference Date*. Proses ini adalah langkah awal yang wajib dilakukan karena, tanpa adanya *voucher* yang ditambahkan, sistem tidak akan mengaktifkan tombol *Save Voucher*.

Hal ini dirancang untuk mencegah pengguna menyimpan transaksi yang tidak lengkap atau kosong, sehingga menjaga integritas data yang tercatat dalam sistem. Dengan demikian, pengguna harus memastikan bahwa setidaknya satu *voucher* telah dibuat dan divalidasi sebelum melanjutkan ke langkah penyimpanan. Desain ini bertujuan untuk memberikan kontrol penuh kepada pengguna dan memastikan bahwa setiap langkah transaksi tercatat secara sistematis sesuai prosedur.

Gambar 4. 11 Tampilan Halaman *Add Voucher*

Gambar 4.11 menunjukkan tampilan formulir isian untuk menambahkan data *Voucher* dalam transaksi *Prior Adjustment*. Pada halaman ini, pengguna diminta untuk mengisi beberapa data wajib, antara lain *Transaction*, *Voucher Date*, *Description*, *Notes*, *Reference No*, dan *Reference Date*. Pengisian data yang tepat dan lengkap sangat penting untuk memastikan bahwa transaksi yang

dilakukan tercatat dengan baik dan sesuai dengan standar akuntansi yang berlaku. Data-data ini akan membentuk dasar dari transaksi yang akan diproses lebih lanjut dalam sistem.

Setelah seluruh data wajib berhasil diisi, sistem secara otomatis akan menampilkan tombol *Add Detail Voucher* di bagian kanan bawah layar. Tombol ini memungkinkan pengguna untuk melanjutkan ke tahap pengisian detail *voucher* yang lebih rinci. Halaman ini dirancang agar pengguna dapat melengkapi setiap transaksi secara menyeluruh, memastikan tidak ada informasi penting yang terlewatkan, dan memberikan kontrol penuh bagi pengguna untuk melakukan pencatatan yang akurat dan sesuai dengan kebutuhan perusahaan.

The screenshot shows a web application interface for adding a voucher. The top section, 'Voucher Information', contains several input fields: 'Company Name' (dropdown), 'Office Name' (text), 'Transaction' (dropdown), 'Voucher Date' (calendar), 'Description' (text), 'Notes' (text), 'Batch Code' (text), 'Batch Date' (calendar), 'Batch Name' (text), 'Reference No' (text), and 'Reference Date' (calendar). The bottom section, 'Detail Voucher List', features a table with the following columns: 'Coy/Office/COA', 'COA Name', 'Description', 'Currency', 'Rate', 'D/C', 'Base Amount', 'Origin Amount', 'Edit', and 'Delete'. The table is currently empty, with a 'Total: 0 - 0 Of 0 Result' message and a 'No Data' status. A '+ Add Detail' button is located at the bottom right of the table.

Gambar 4. 12 Tampilan setelah data terisi

Gambar 4.12 memperlihatkan tombol *Add Detail*, yang digunakan untuk menambahkan detail *voucher* pada transaksi *Prior Adjustment*. Dalam sistem ini, setiap *voucher* idealnya harus memiliki minimal dua detail *voucher* yang terdiri dari *post code credit* (C) dan *debit* (D). Struktur ini penting untuk menjaga keseimbangan transaksi akuntansi dan memastikan bahwa semua data yang dimasukkan mencerminkan situasi keuangan yang akurat. Selain itu, nilai *Base Amount* dan *Origin Amount* dari kedua detail *voucher* yang dimasukkan juga harus seimbang (*balanced*), yang berarti total kredit dan debit harus memiliki nilai yang sama.

Proses ini dirancang untuk memastikan bahwa tidak ada data yang timpang (*unbalanced*) yang dapat memengaruhi akurasi pencatatan transaksi dalam buku besar. Jika jumlah debit dan kredit tidak seimbang, sistem akan memberikan peringatan atau *error message* yang menginstruksikan pengguna untuk memperbaiki data sebelum melanjutkan. Dengan cara ini, sistem membantu mencegah kesalahan dalam pencatatan yang bisa berdampak pada laporan keuangan perusahaan.

Gambar 4. 13 Tampilan *Add Detail Voucher*

Gambar 4.13 menampilkan formulir untuk menambahkan data *Voucher Detail*. Di sini, pengguna dapat mengisi berbagai *field*, seperti *Company Name*, *Office Name*, *COA Code*, *Currency Rate*, *Description*, *Post (C/D)*, dan *Base Amount*. Untuk *field Office Name*, pilihan yang tersedia akan menyesuaikan dengan *Company Name* yang dipilih sebelumnya, menjamin bahwa informasi yang dimasukkan relevan dengan konteks yang ada. Selain itu, sistem akan otomatis menampilkan pilihan untuk *COA Code* yang hanya relevan dengan *Company* dan *Office* yang telah dipilih, sehingga mengurangi potensi kesalahan input data.

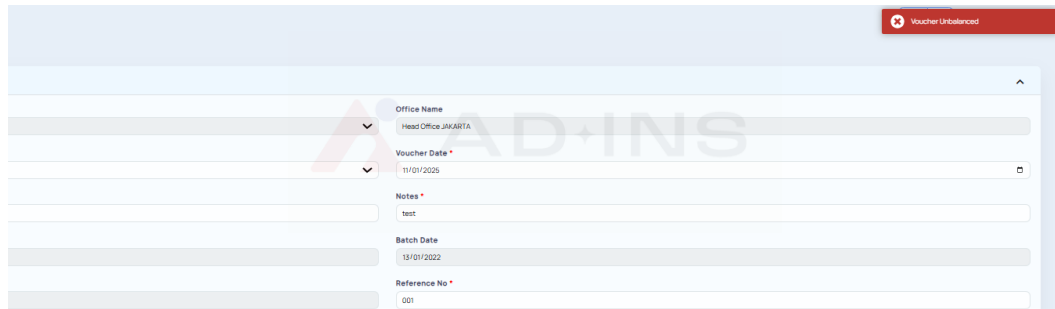
Setelah memilih *COA Code*, *field* tambahan seperti *COA Name*, *Currency*, dan *Coy/Office/COA* akan terisi secara otomatis berdasarkan data yang ada pada tabel di *database*. Dengan cara ini, pengguna tidak perlu lagi mengisi secara manual informasi yang telah ada, mengurangi risiko kesalahan dan meningkatkan efisiensi proses input. Terakhir, nilai pada *field Origin Amount* akan dihitung

otomatis berdasarkan rumus: $Origin Amount = Base Amount / Currency Rate$, memastikan bahwa konversi mata uang dilakukan dengan benar sesuai dengan kurs yang berlaku.

Gambar 4. 14 Tampilan *Voucher* yang sudah memiliki *Detail*

Gambar 4.14 menunjukkan tampilan *voucher* yang telah memiliki dua detail *voucher*. Pada kondisi ini, kedua *post code* dalam detail *voucher* tersebut sama-sama berstatus debit (D), sehingga *voucher* menjadi *unbalanced* atau tidak seimbang. Ketidakseimbangan ini melanggar prinsip dasar akuntansi, yaitu keseimbangan antara debit dan kredit. Saat pengguna mencoba melakukan validasi dengan menekan tombol *Validate*, sistem secara otomatis menjalankan fungsi untuk memeriksa kesesuaian antara *post code* dan *balance* pada detail *voucher*. Ketika sistem mendeteksi ketidakseimbangan, muncul *pop-up error message* yang menyatakan bahwa *voucher* tidak seimbang (*Voucher Unbalanced*).

Validasi ini berfungsi sebagai pengingat bagi pengguna bahwa data *voucher* belum memenuhi aturan keseimbangan dan harus diperbaiki sebelum transaksi dapat dilanjutkan. Dalam kasus dua detail *voucher*, salah satunya harus menggunakan *post code* kredit (C), sementara yang lainnya debit (D). Jika jumlah detail *voucher* lebih dari dua, total nilai debit dan kredit harus genap dan saling seimbang. Dengan adanya validasi ini, sistem memastikan integritas pencatatan keuangan tetap terjaga dan menghindari kesalahan yang dapat berdampak pada laporan keuangan.



Voucher Unbalanced

Office Name
Head Office JAKARTA

Voucher Date
11/01/2025

Notes
test

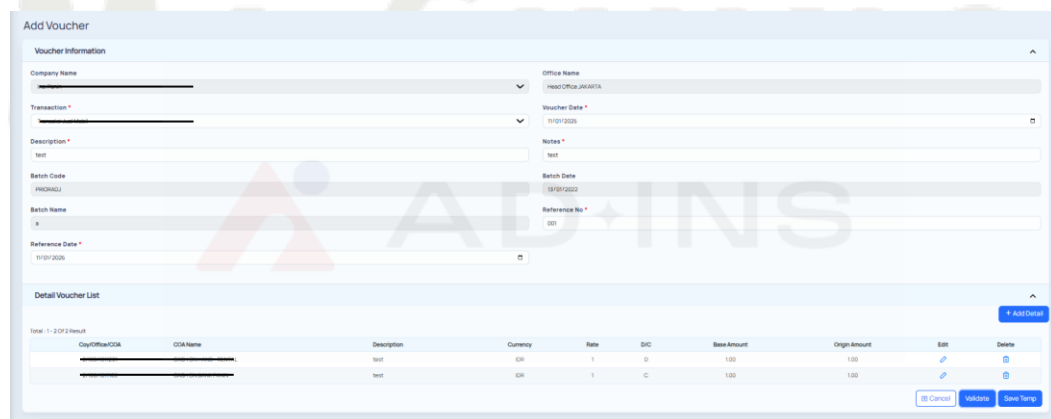
Batch Date
13/01/2022

Reference No
001

Gambar 4. 15 Tampilan *Voucher Unbalanced*

Gambar 4.15 memperlihatkan *pop-up error message* yang muncul ketika sistem mendeteksi ketidakseimbangan pada *post code voucher*. Pesan ini muncul akibat kedua *post code* pada detail *voucher* sama-sama memiliki status debit (D). Situasi ini tidak memenuhi prinsip keseimbangan akuntansi, yang mengharuskan setiap transaksi memiliki proporsi debit dan kredit yang seimbang. Validasi sistem ini bertujuan untuk mencegah pengguna menyimpan data yang tidak sesuai dengan aturan akuntansi.

Selain itu, *pop-up error* ini memberikan panduan bagi pengguna untuk memperbaiki detail *voucher* agar sesuai dengan aturan. Ketika jumlah detail *voucher* lebih dari dua, jumlah *post code* debit dan kredit harus genap dan nilai totalnya saling berimbang. Proses validasi ini penting untuk menjaga konsistensi data dalam sistem dan memastikan bahwa setiap transaksi yang dicatat telah memenuhi standar akuntansi yang berlaku.



Add Voucher

Voucher Information

Company Name
Head Office JAKARTA

Transaction
11/01/2025

Description
test

Batch Date
13/01/2022

Batch Name
001

Reference Date
11/01/2025

Office Name
Head Office JAKARTA

Voucher Date
11/01/2025

Notes
test

Batch Date
13/01/2022

Reference No
001

Detail Voucher List

Total: 1 - 2 Of 2 Result

Copy/Office/COA	COA Name	Description	Currency	Rate	SDC	Base Amount	Origin Amount	Edit	Delete
		test	IDR	1	D	100	100		
		test	IDR	1	C	100	100		

Buttons: Cancel, Validate, Save Temp, Apply/Save

Gambar 4. 16 Tampilan *Voucher Balanced*

Gambar 4.16 menunjukkan hasil validasi yang sukses setelah post code pada dua detail *voucher* telah diatur sesuai aturan. Pada tampilan ini, terlihat bahwa jumlah debit dan kredit telah seimbang, dengan masing-masing post code memiliki nilai yang sesuai. Setelah validasi berhasil, sistem secara otomatis menampilkan tombol *Save Temp*, yang memungkinkan pengguna untuk menyimpan sementara data *voucher* yang telah divalidasi. Penyimpanan sementara ini memberikan fleksibilitas kepada pengguna untuk meninjau ulang data atau melanjutkan proses transaksi tanpa kehilangan data.

Tombol *Save Temp* ini juga memastikan bahwa data yang telah divalidasi tetap terjaga dengan baik dalam sistem. Setelah proses validasi selesai, pengguna dapat melanjutkan langkah berikutnya dalam pembuatan transaksi *Prior Adjustment*. Dengan memastikan bahwa *voucher* telah memenuhi semua aturan akuntansi sebelum transaksi dilanjutkan, sistem mendukung pencatatan yang akurat dan efisien.

The screenshot shows a web-based form titled "Prior Adjustment Journal Transaction". It contains several input fields and a table. The fields include "Company Name" (PT. PRIMA), "Office Name" (Head Office JAKARTA), "Branch Code" (PRIORADJ), "Branch Date" (13-Jan-2022), "Description" (Prior Adjustment), and "Status" (Open). Below these fields is a "Voucher List" table with columns for "Transaction", "Voucher No", "Date", "Description", and "Balance". The table currently shows one entry with "Transaction" as "13-Jan-2022" and "Description" as "Test". At the bottom right of the form, there are buttons for "Back", "Detail Print Voucher", and "Save Voucher".

Gambar 4. 17 Halaman Transaksi dengan *Voucher*

Gambar 4.17 menunjukkan halaman awal transaksi yang telah memiliki minimal satu *voucher*. Setelah pengguna berhasil menambahkan dan memvalidasi *voucher*, tombol *Save Voucher* akan muncul di halaman ini. Tombol ini digunakan untuk menjalankan fungsi utama *Add Prior Adjustment*, yang mencakup pembuatan jurnal dan pencatatan transaksi keuangan dalam sistem. Setelah transaksi selesai, status *batch* akan berubah menjadi *Posted*, yang menandakan bahwa transaksi telah berhasil tercatat secara resmi.

Prior Adjustment Journal Transaction

Prior Adjustment Journal Transaction

Company Name

PT. OFF Inigrate

Office Name

Head Office

Batch Code

PRIORADJ

Batch Date

01-Mar-2023

Batch Name

Testing Data SIT

Description

Prior Adjustment

Owner

User1

Close Date

-

Post By

User1

Post Date

01-Mar-2023

Status

Posted

Voucher List

Print Voucher

Total : 1 - 6 Of 6 Result

Transaction	Voucher No	Date	Description	Edit	Delete
FADJ	00000000000000000000	30-Nov-2022	testing sit		
FADJ	00000000000000000000	31-Dec-2022	testing sit		
FADJ	00000000000000000000	31-Jan-2023	testing sit		
FADJ	00000000000000000000	28-Feb-2023	testing sit		
TAX	00000000000000000000	30-Nov-2022	testing sit		
TRJB	00000000000000000000	01-Nov-2022	testing sit		

Back

Detail Print Voucher

Gambar 4. 18 Halaman *Voucher Posted*

Gambar 4.18 memperlihatkan contoh *voucher* dengan status *Posted*, yang menandakan bahwa transaksi *Prior Adjustment* pada *batch* tersebut telah selesai dan tercatat secara resmi di dalam sistem. Pada tahap ini, sistem akan secara otomatis menghasilkan beberapa jurnal penting, seperti jurnal *Tax*, jurnal *Forex*, dan jurnal *Prior Adjustment*, yang semuanya mencerminkan transaksi yang telah dilakukan. Jurnal-jurnal ini terbentuk berdasarkan data *voucher* dan detail *voucher* yang telah divalidasi sebelumnya.

Setelah status *voucher* berubah menjadi *Posted*, tidak ada lagi perubahan atau penyesuaian yang dapat dilakukan terhadap transaksi tersebut. Hal ini bertujuan untuk menjaga integritas data yang tercatat dalam sistem, serta memastikan bahwa tidak ada modifikasi yang dapat menyebabkan inkonsistensi pada laporan keuangan atau jurnal yang telah terbentuk. Sebagai tambahan, status *Posted* juga menandakan bahwa seluruh proses, mulai dari input data hingga validasi dan pembuatan jurnal, telah selesai dengan benar sesuai prosedur yang ditetapkan.

Pada tahap ini, pengguna hanya dapat mengakses fitur-fitur seperti *Print Voucher* atau *Detail Print Voucher* untuk menghasilkan laporan dari transaksi yang telah selesai. Laporan tersebut memuat informasi lengkap mengenai transaksi *Prior Adjustment*, termasuk daftar jurnal yang terbentuk, nilai *balance*, dan *detail voucher* yang terlibat. Dengan laporan ini, pengguna dapat

mendokumentasikan transaksi untuk keperluan audit, arsip, atau analisis lebih lanjut.

Pentingnya perubahan status menjadi *Posted* tidak hanya untuk menandai transaksi selesai tetapi juga berfungsi sebagai pengunci data, sehingga *batch* tersebut tidak dapat digunakan kembali untuk transaksi lainnya. Desain ini memastikan bahwa setiap transaksi *Prior Adjustment* dikelola secara terorganisir, akurat, dan aman dari potensi kesalahan atau manipulasi data di masa mendatang.

PRIOR ADJUSTMENT JOURNAL LIST

02 Dec 2022

Company - Office : PT. SPTMigrate
Sub System : CONFINS
Batch Code : PRIORADJ
Date : 01/03/2023
Batch Name : Prior Adjustment
Description : testing data SIT

Transaction	Voucher Code	Date	Description	Reference Code
FADJ	2120001000	11/30/2022 12:00:00 AM	testing sit	
FADJ	2120001000	12/31/2022 12:00:00 AM	testing sit	
FADJ	2120001000	1/31/2023 12:00:00 AM	testing sit	
FADJ	2120001000	2/28/2023 12:00:00 AM	testing sit	
TAX	2120001000	11/30/2022 12:00:00 AM	testing sit	
TRJB	2120001000	11/1/2022 12:00:00 AM	testing sit	

Gambar 4. 19 *Report Journal List*

Gambar 4.19 menunjukkan hasil *generate report* yang dihasilkan setelah pengguna mengklik tombol *Print Voucher*. Pada laporan ini, pengguna dapat melihat daftar jurnal yang terbentuk dalam transaksi *Prior Adjustment* pada *batch* tertentu. Informasi yang ditampilkan mencakup detail jurnal, seperti kode transaksi, kode voucher, tanggal, deskripsi, dan kode referensi. Laporan ini memberikan gambaran umum yang sangat berguna bagi pengguna untuk mengevaluasi transaksi yang telah dilakukan.

Laporan ini dirancang untuk memberikan transparansi dan kemudahan bagi pengguna dalam meninjau hasil transaksi. Dengan adanya laporan ini,

pengguna dapat melakukan audit internal atau memberikan informasi kepada pihak manajemen sebagai bentuk tanggung jawab dalam pelaporan keuangan. Laporan ini juga dapat dijadikan referensi untuk pengambilan keputusan yang berkaitan dengan transaksi keuangan di masa mendatang.

Prior Adjustment Journal Transaction

Batch No

: 00000000000000000000

Voucher No	COA	COA Description	Currency	Rate	Origin Amount	Base Amount	
						Debit	Credit
00000000000000000000		BELOW GAIN	IDR	1.00	0.00	0.00	1,000.00
		EXCHANGE OF					
			USD	1.00	0.00	1,000.00	0.00
		LEASING USD					
00000000000000000000			IDR	1.00	0.00	0.00	500.00
		EXCHANGE OF					

Gambar 4. 20 Report Journal Transaction

Gambar 4.20 menampilkan hasil *generate report* yang lebih detail saat pengguna mengklik tombol *Detail Print Voucher*. Laporan ini berisi informasi rinci mengenai *balance* dan transaksi *Prior Adjustment* dari setiap jurnal yang terbentuk. Setiap elemen transaksi dijelaskan secara mendalam dalam laporan ini, termasuk nilai-nilai saldo, akun-akun yang terlibat, dan jenis transaksi yang dilakukan.

Laporan ini memberikan tingkat transparansi yang tinggi dan memungkinkan pengguna untuk melacak setiap elemen transaksi secara akurat. Hal ini penting untuk memastikan bahwa semua data telah tercatat dengan benar dan sesuai dengan aturan akuntansi yang berlaku. Selain itu, laporan ini dapat digunakan sebagai dokumen pendukung untuk audit atau kebutuhan pelaporan keuangan lainnya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil Praktik Kerja Lapangan (PKL) yang telah dilakukan di PT Adicipta Inovasi Teknologi, dapat disimpulkan bahwa proses pengembangan aplikasi untuk proyek EGL2 berjalan dengan baik dan sesuai dengan perencanaan. Aplikasi ini dirancang untuk memudahkan pendokumentasian laporan keuangan sesuai dengan ketentuan produk yang berlaku dalam proyek EGL.

Dalam pengembangannya, aplikasi ini menggunakan teknologi ASP.NET untuk pengembangan back-end dan Angular untuk front-end, yang memungkinkan sistem memiliki performa yang optimal, skalabilitas yang baik, serta antarmuka pengguna yang intuitif dan responsif.

Fitur utama yang dikembangkan dalam aplikasi ini mencakup:

1. Perubahan saldo secara otomatis ketika terjadi transaksi, memastikan data selalu akurat dan terkini.
2. Pembentukan jurnal keuangan yang berjalan dengan baik saat ada mutasi pemasukan dan pengeluaran, membantu dalam pencatatan transaksi sesuai dengan standar keuangan perusahaan.
3. Fitur *Add Prior*, yang berfungsi untuk memberikan prioritas pada transaksi tertentu, sehingga dapat dikelola dengan lebih efisien.
4. Fitur *Create*, *Edit*, dan *Delete Batch*, yang memungkinkan pengguna untuk mengelola kumpulan transaksi dengan fleksibilitas tinggi, mendukung efisiensi dalam proses input dan pengolahan data keuangan.
5. Fitur *View Voucher*, yang memungkinkan pengguna untuk melihat detail transaksi keuangan dalam bentuk voucher, membantu dalam verifikasi dan audit keuangan.

Agar pengembangan dapat berjalan dengan cepat dan sesuai kebutuhan pengguna, metode *Rapid Application Development* (RAD) diterapkan dalam proyek ini. Metode ini terbukti efektif dalam mempercepat siklus pengembangan melalui tahapan iteratif yang memungkinkan pengembang untuk terus beradaptasi dengan kebutuhan bisnis dan ketentuan produk. Dengan pendekatan ini, setiap

perubahan atau pembaruan fitur dapat dilakukan dengan cepat berdasarkan umpan balik dari pengguna, sehingga aplikasi dapat dikembangkan dengan lebih fleksibel dan sesuai dengan standar yang ditetapkan.

Selain itu, proses kolaborasi yang baik antara tim pengembang dan tim pengujian memastikan bahwa setiap fitur diuji secara menyeluruh sebelum dirilis, sehingga dapat meminimalisir kesalahan dan meningkatkan keandalan sistem. Hambatan-hambatan teknis yang muncul selama pengembangan dapat diatasi dengan cepat melalui diskusi dan penyempurnaan berkelanjutan.

Dengan diterapkannya pendekatan ini, aplikasi yang dikembangkan berhasil memenuhi kebutuhan perusahaan dalam memudahkan pendokumentasian laporan keuangan. Aplikasi ini memungkinkan pencatatan transaksi keuangan menjadi lebih sistematis, akurat, dan terdokumentasi dengan baik sesuai dengan ketentuan produk dalam proyek EGL2 di PT Adicipta Inovasi Teknologi.

5.2 Saran

Berdasarkan hasil analisis dan evaluasi terhadap proses pengembangan aplikasi di perusahaan ini, terdapat beberapa aspek yang dapat ditingkatkan untuk meningkatkan efisiensi dan mengurangi revisi yang berulang. Salah satu tantangan utama yang dihadapi adalah kompleksitas aplikasi yang tinggi dan proses pengujian yang memakan waktu. Oleh karena itu, disarankan agar perusahaan menerapkan pendekatan pemecahan tugas menjadi unit yang lebih kecil (*modular development*).

Dengan membagi pengembangan menjadi tahapan yang lebih kecil dan terstruktur, beberapa manfaat yang dapat diperoleh antara lain:

1. Pengurangan Beban Revisi

Dengan memecah pengembangan menjadi modul-modul kecil, setiap bagian dapat diuji dan diperbaiki lebih awal sebelum masuk ke tahap berikutnya. Hal ini mengurangi kemungkinan kesalahan besar yang sulit diperbaiki di tahap akhir.

2. Peningkatan Efisiensi Pengujian

Proses testing dapat dilakukan per modul sebelum digabungkan dengan

sistem utama. Ini akan mengurangi kompleksitas pengujian sekaligus memastikan setiap bagian sudah berjalan dengan baik sebelum integrasi penuh dilakukan.

3. Penyelesaian yang Lebih Terukur

Dengan pendekatan bertahap, pengembangan dapat dipantau lebih jelas, dan estimasi waktu untuk setiap tahap dapat lebih akurat dibandingkan langsung mengerjakan sistem dalam skala besar sekaligus.

4. Fleksibilitas dalam Perubahan

Jika terjadi perubahan kebutuhan atau penyesuaian sistem, hanya modul terkait yang perlu direvisi, tanpa harus melakukan perubahan besar yang mempengaruhi seluruh aplikasi.

Implementasi strategi ini diharapkan dapat membantu perusahaan dalam mengoptimalkan waktu pengembangan, mengurangi revisi yang tidak perlu, serta meningkatkan kualitas aplikasi yang dihasilkan.

UNIVERSITAS
MA CHUNG

Daftar Pustaka

- Bielak, K., Borek, B., & Plechawska-Wójcik, M. (2022). JCSI (Journal Computer Science Institute), 23, 77–83.
- Kirana, W. K. C., & Zuhri, Z. (2023). Pemanfaatan Directive pada Framework Angular untuk Pengembangan Website Penerimaan Mahasiswa Baru. *Jurnal Universitas Islam Indonesia*.
- Aroraa, G., & Dash, T. (2018). Building RESTful Web Services with .NET Core: Developing Distributed Web Services with .NET Core. Packt Publishing.
- Griffiths, I. (2022). Programming C# 10: Build Cloud, Web, and Desktop Applications. O'Reilly Media, Inc.
- Sanjaya, D., Abdurachman, H., Wicaksono, A. A., & Masya, F. (2021). Sistem Informasi Pengendalian Asset Kendaraan di Perusahaan Transportasi/Ekspedisi Berbasis Web (REMINDsset). *RABIT: Jurnal Teknologi dan Sistem Informasi Univrab*, 6(1), 24–32.
- Purnama, R. A., & Putra, A. T. L. (2018). Aplikasi Web Server Berbasis Bahasa C#. *Jurnal Teknik Komputer*, 4(1), 22.
- Dharma, B., Ramadhani, Y., & Reitandi. (2024). Pentingnya Laporan Keuangan untuk Menilai Kinerja Suatu Perusahaan. *El Mujtama Jurnal Pengabdian Masyarakat*, 4(1), 46–52.
- Kurniawan, A. (2013). Seri Belajar Mandiri: Pemrograman C# Untuk Pemula. Ilmu Data Publisher.
- Rachmatullah, A. (2022). Mempelajari C#: Bahasa Pemrograman Modern.

- Sari, A. S., & Hidayat, R. (2022). Designing Website Vaccine Booking System Using Golang Programming Language and Framework React JS. *Journal of Information System, Informatics and Computing*, 6(1), 26.
- Faisal, M. R. (2017). *Seri Belajar ASP.NET: ASP.NET Core MVC & MySQL dengan Visual Studio Code*.
- Romzi, M., & Kurniawan, B. (2020). Implementasi Pemrograman Python Menggunakan Visual Studio Code. *Jurnal Informatika dan Komputer*, 11(2).
- Rais, A. R. (2017). Implementasi Aplikasi Pendaftaran dan Pembayaran Kontrak Ahmad Rais Berbasis Desktop VB.Net dan Microsoft Access. *Paradigma*, 19(1), Maret.
- Wahyuningrum, T., & Januarita, D. (2014). Perancangan WEB e-Commerce dengan Metode Rapid Application Development (RAD) untuk Produk Unggulan Desa. *Seminar Nasional Teknologi Informasi & Komunikasi Terapan (SEMANTIK 2014)*, 81–82.
- Fahmi, I. (2012). *Analisis Laporan Keuangan*. Alfabeta. Bandung.
- Habibi, L. H., & Supriatna, I. (2021). Penerapan Aplikasi Keuangan Berbasis Android SI APIK Dalam Penyusunan Laporan Keuangan Berdasarkan SAK EMKM (Studi Kasus Qaya Laundry). *Indonesian Accounting Literacy Journal*, 1(3), 659–670.
- Ribeiro, E. M. S., & Prativiera, G. A. (2014). Information Theoretic Approach for Accounting Classification. *Departamento de Administração, FEARP, Universidade de São Paulo, Ribeirão Preto, Brazil*.

- Sahani, A. K., & Singh, P. (2020). Web Development Using Angular: A Case Study. *Journal of Informatics Electrical and Electronics Engineering*, 1(2), 1–7.
- Purohit, R., Jain, S., & Gupta, S. (2023). Role of Angular Framework in Web Development. *International Journal of Advanced Research in Science, Communication and Technology*, 10(2), 103–115.
- Amann, S., Proksch, S., & Mezini, M. (2016). A Study of Visual Studio Usage in Practice. *IEEE International Conference*.
- Subagia, A. (2017). *Membangun Aplikasi dengan CodeIgniter dan Database SQL Server*. Elex Media Komputindo.
- Zalukhu, S., Situmeang, R. O., & Marpaung, F. R. (2023). Implementasi Flowchart dalam Perancangan Sistem Informasi. *Jurnal Teknologi dan Informatika*, 4(2), 1-10.
- Malabay, M. (2016). Analisis Penggunaan Flowchart dalam Proses Produksi. *Jurnal Ilmu Komputer*, 8(1), 15-22.

Lampiran A

Biodata Mahasiswa

Data Pribadi

Nama Lengkap	:	Siusiu Trifena Halim
Nama Panggilan	:	Siusiu
NIM	:	312110013
Tempat Tanggal Lahir	:	Malang, 2 Februari 2003
Fakultas/Prodi	:	Teknologi dan Desain/Teknik Informatika
Universitas	:	Universitas Ma Chung
Alamat	:	Jl. Jaksa Agung Suprpto no 53 Malang
No. Ponsel	:	081999717402
Email Address	:	siu.siutrifena@gmail.com
IPK terakhir	:	3,74

Pendidikan

Tingkat Pendidikan	Nama Sekolah	Kota
Sekolah Dasar (SD)	SD Taman Harapan	Malang
Sekolah Menengah Pertama (SMP)	SMPK Cor Jesu	Malang
Sekolah Menengah Atas (SMA)	SMAK Cor Jesu	Malang

Pengalaman Organisasi

Organisasi	Tahun	Posisi
BEMU	2022-2024	Anggota Home Affair

Organisasi	Tahun	Posisi
Ma Chung Festivsl	2022	Anggota Kesehatan
Ma Chung Festival	2023	Koordinator Kesehatan

Lampiran B

Form Bimbingan Dosen Pembimbing



**FAKULTAS
TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

FORM PKL_FTD04

Universitas Ma Chung
Soegeng Hendarto Bhakti Persada Building
Villa Puncak Tidar N-01
Malang 65151, Indonesia
ftd@machung.ac.id (Mail)
+62 341 550171 (Phone)
+62 341 550175 (Fax)

LEMBAR BIMBINGAN PRAKTIK KERJA LAPANGAN (PKL)

Nama Mahasiswa	:	Siusiu Trifena Halim
NIM	:	312110013
Program Studi	:	Teknik Informatika
Judul Laporan PKL	:	Pengembangan Aplikasi Prior Adjustment Berbasis Web Menggunakan Framework Angular pada PT Adicpta Inovasi Teknologi

No	Hari, tanggal	Topik Bimbingan	TTD Dosen Pembimbing
1	Jumat, 8/11/2024	Konsultasi Awal Pelaksanaan Magang	
2	Senin, 30/12/2024	Konsultasi Bab 1 dan 2	
3	Jumat, 10/1/2025	Konsultasi Bab 3	
4	Kamis, 16/1/2025	Konsultasi Materi Magang di Adi Cipta Inovasi teknologi	
5	Jumat, 24/1/2025	Konsultasi Laporan Magang di Adi Cipta Inovasi teknologi	



"EXCELLENCE THROUGH COMPETENCY"

Lampiran C

Lembar Penilaian Perusahaan



**FAKULTAS
TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

Universitas Ma Chung
Soegeng Hendarto Bhakti Persada Building
Villa Puncak Tidar N-01
Malang 65151, Indonesia
ftd@machung.ac.id (Mail)
+62 341 550171 (Phone)
+62 341 550175 (Fax)

FORM PKL_FTD03

LEMBAR PENILAIAN EKSTERNAL PRAKTIK KERJA LAPANGAN (PKL)

Nama Mahasiswa	: Siusiu Trifena Halim
NIM	: 312110013
Program Studi	: Teknik Informatika
Pembimbing Lapangan	: Welly Satria Nugraha
Tanggal Pelaksanaan	: 24 Januari 2025
Tempat PKL	: PT Adicipta Inovasi Teknologi
Topik PKL	: PENGEMBANGAN APLIKASI <i>PRIOR ADJUSTMENT</i> BERBASIS <i>WEB</i> MENGGUNAKAN <i>FRAMEWORK ANGULAR</i> PADA PT. ADICIPTA INOVASI TEKNOLOGI

Komponen Penilaian :

No	Komponen Penilaian	Nilai Angka (0-100)
1	Disiplin dalam kehadiran (presensi)	92
2	Kesungguhan dalam melakukan praktik kerja	97
3	Disiplin dalam pekerjaan	92
4	Kemampuan memecahkan masalah	94
5	Tanggung jawab	98
6	Kemauan untuk mengetahui hal-hal yang ada di tempat praktik kerja	95
7	Pengetahuan tentang ilmu yang dilaksanakan	97
8	Keterampilan	95
9	Kemampuan menyampaikan pendapat	95
10	Kemampuan dalam bekerja sama	95
Rerata (Jumlah Nilai : 10 komponen)		95

Rentang Nilai:

MUTU	RENTANG	BOBOT NILAI
A	95,00 s/d 100	4,00
AB	90,00 s/d 94,99	3,70
BA	82,00 s/d 89,99	3,30
B	73,00 s/d 81,99	3,00
BC	65,00 s/d 72,99	2,70
CB	60,00 s/d 64,99	2,30
C	56,00 s/d 59,99	2,00
CD	50,00 s/d 55,99	1,50
D	40,00 s/d 49,99	1,00
E	00,00 s/d 39,99	0,00

Jumat, 24 Januari 2024

Pembimbing Lapangan

Welly Satria Nugraha
(Team Leader)



"EXCELLENCE THROUGH COMPETENCY"

Lampiran D

Lembar Partisipasi Seminar



**FAKULTAS
TEKNOLOGI DAN DESAIN
UNIVERSITAS MA CHUNG**

Universitas Ma Chung
Soegeng Hendarto Bhakti Persada Building
Villa Puncak Tidar N-01
Malang 65151, Indonesia
ftd@machung.ac.id (Mail)
+62 341 550171 (Phone)
+62 341 550175 (Fax)

FORM PKL_FTD06

LEMBAR PARTISIPASI SEMINAR PRAKTIK KERJA LAPANGAN (PKL)

Nama Mahasiswa	:	Siusiu Trifena Halim
NIM	:	312110013
Program Studi	:	Teknik Informatika

No	Hari, tanggal	Judul PKL	Pemateri PKL	TTD Dosen Pembimbing Pemateri
1	Rabu, 5 Juli 2023	Pembuatan Software Berbasis Python untuk Mengontrol I2C Dongle STM32 dengan Protokol Komunikasi UART dan USB	Yohanes Vianney Dani	Dr.Eng. Romy Budhi., ST., MT., M.Pd
2	Rabu, 7 Agustus 2024	Adjustment dan Rollout Project Box Counter dengan Computer Vision di PT .XYZ	Louis Axel Tjokro Atmodjo	Dr. Kestrillia Rega Prilianti., M.Si
3	Jumat, 16 Agustus 2024	Implementasi Teknologi Front End dalam Proyek Pengembangan Website	Vincent Dwi Hartanto	Paulus Lucky Tirma Irawan, S.Kom., MT
4	Selasa, 10 September 2024	Pengembangan Front-End Aplikasi Desktop untuk Konfigurasi Touch IC pada Layar Smartphone di Viewtrix Technology Limited	Stella Maureen Ignacia Santoso	Muhamad Subianto, S.Kom., M.Cs
5	Selasa, 17 September 2024	Pengembangan Back-End Aplikasi Desktop untuk Konfigurasi Touch IC pada Layar Smartphone di Viewtrix Technology Limited	David Yusaku Setiyono	Muhamad Subianto, S.Kom., M.Cs



"EXCELLENCE THROUGH COMPETENCY"