

# PERANCANGAN SISTEM INFORMASI MANAJEMEN INVOICE MENGUNAKAN GENERATOR FREAMWORK DJANGO-PYTHON BERBASIS WEBSITE PADA PT. LAMPUIND TEKNO ELEKTRIK

Fitria Risyda<sup>a</sup>, Yamin Nuryamin<sup>b</sup>

<sup>a</sup> frisyda@unsurya.ac.id, <sup>b</sup> yamin.yny@nusamandiri.co.id

<sup>a</sup> Universitas Dirgantara Marsekal Suryadarma, <sup>b</sup> Universitas Nusa Mandiri

## **Abstract**

*Rapid technological advances encourage every company to keep abreast of developments and continue to improve its ability to manage data and information. Lampuind Tekno Electric companies currently still manages financial data and invoices by manually. This is prone to causing errors in typing invoices and loss of customer transaction data. For this reason, we need a system that helps in managing invoice data so that invoice data processing becomes faster and more accurate. The method used in the development of this application uses the waterfall model, the author makes the application design with the django Python framework and the modeling language uses the Unified Modeling Language (UML). With the existence of a web-based invoice management information system, it is hoped that it can assist employees in making invoices more quickly and accurately as well as making it easier to search for data and present information.*

*Keywords: Information System, Invoice Management, Django Framework*

## **Abstrak**

Kemajuan teknologi sangat cepat mendorong setiap perusahaan untuk tetap mengikuti perkembangan dan terus meningkatkan kemampuannya dalam mengelola data-data dan informasi. PT. Lampuind Tekno Elektrik dalam mengelola data keuangan dan invoice yang berjalan saat ini masih secara manual. Hal ini rentan menimbulkan kesalahan dalam pengetikan pembuatan invoice dan hilangnya data transaksi pelanggan. Untuk itu diperlukan sebuah sistem yang membantu dalam pengelolaan data invoice sehingga pengolahan data invoice menjadi lebih cepat dan akurat. Metode yang digunakan pada pengembangan aplikasi ini menggunakan model waterfall, perancangan aplikasinya penulis membuat dengan *framework* django Phyton dan bahasa pemodelannya menggunakan *Unified Modeling Language* (UML). Dengan adanya sistem informasi manajemen invoice berbasis web ini diharapkan dapat membantu karyawan dalam pembuatan invoice lebih cepat dan akurat serta kemudahan pencarian data dan penyajian informasi.

Kata kunci: Sistem Informasi, Manajemen invoice, Django framework

## **I. PENDAHULUAN**

Kemajuan teknologi sangat cepat mendorong setiap perusahaan untuk tetap mengikuti perkembangan dan terus meningkatkan kemampuannya dalam mengelola data-data dan informasi yang dimiliki dan yang dibutuhkan-kannya untuk mencapai tujuan tersebut, suatu perusahaan memerlukan informasi tepat dan efisien juga cepat. Untuk itu suatu perusahaan membutuhkan suatu sistem informasi yang mendukung kebutuhan

perusahaan yang akan sangat membantu sebuah manajemen perusahaan baik dalam menciptakan efisiensi dan efektivitas kerja perusahaan itu sendiri, maupun dalam meningkatkan pelayanan perusahaan terhadap pelanggan. Aplikasi berbasis website ini sangat berdampak besar terhadap proses berjalannya kegiatan pembayaran, selain sebagai media penyampaian informasi, aplikasi ini juga sebagai contoh dalam penerapan teknologi yang semakin canggih.

PT. Lampuind Tekno Elektrik ingin mengembangkan usahanya menjadi lebih maju dan menjadi lebih baik lagi dalam pendataan keuangan dan pembuatan invoice yang berjalan saat ini masih dilakukan secara manual. Sistem yang berjalan saat ini masih belum berjalan dengan baik dikarenakan sering terjadinya kesalahan pencatatan dan tagihan customer yang hilang menjadikan laporan yang dihasilkan datanya terkadang tidak sesuai dengan total tagihan yang masuk.

Untuk meningkatkan kualitas dari usaha ini maka dibutuhkan sistem yang dapat membantu bagian keuangan dalam mendata invoice. Sehingga invoice yang masuk sesuai dengan jumlah uang yang dikeluarkan. Perancangan aplikasi ini dimaksudkan untuk membuat sistem yang dapat membantu bagian keuangan dalam mengelola atau membuat invoice dengan berbasis website sehingga akan mengurangi terjadinya kesalahan pada saat pendataan invoice dan kemudahan pencarian data serta pembuatan laporan yang cepat dan akurat.

## II. LANDASAN TEORI

Dalam merancang sistem informasi manajemen invoice berbasis web ini, diperlukan perlengkapan pendukung. Berikut adalah teori yang berkaitan dengan *tools system* dan *tool program* pendukung perancangan aplikasi

### 2.1 Pengertian Aplikasi Berbasis Web

Aplikasi berbasis website ialah aplikasi yang bisa diakses lewat website browser dikala terhubung dengan jaringan internet ataupun intranet

Dikala ini banyak industri memakai aplikasi berbasis website dalam merancang program untuk mengelola industri, sampai pengolahan informasi persediaan serta informasi karyawan. Dalam bidang keuangan, tipe aplikasi ini bisa pula

digunakan buat membuat invoice harian keuangan industri.

### 2.2 HTML

*Hypertext Markup Language* (HTML) merupakan bahasa *website scripting* bersifat *client side* yang membolehkan menampilkan data dalam wujud bacaan, grafik, dan multimedia serta pula buat menghubungkan antar tampilan website *page* ataupun hyperlink.

Elemen HTML ditafsirkan oleh tag, ditulis memakai ciri sudut. Tag semacam langsung perkenalkan konten ke dalam taman (Awaludin & Amelia, 2022). Tag lain semacam mengelilingi serta membagikan data tentang bacaan dokumen serta bisa jadi menyertakan tag lain selaku sub-elemen. Peramban tidak menunjukkan tag HTML, namun memakainya buat menafsirkan konten taman.

HTML bisa menyematkan program yang ditulis dalam bahasa scripting semacam JavaScript, yang mempengaruhi sikap serta konten taman website. Dimasukkannya CSS mendefinisikan tampilan serta tata etak konten. World Wide Website Consortium (W3C), mantan pengelola HTML serta pemelihara standar CSS dikala ini, sudah mendorong pada HTML presentasi eksplisit semenjak 1997.

### 2.3 PYTHON

Python telah menjadi bahasa yang banyak digunakan untuk aplikasi data science. Bahasa python merupakan kombinasi dari general purpose programming languages dengan kemudahan penggunaan bahasa domain specific scripting seperti MATLAB atau R. Python memiliki library untuk data loading, visualisasi, statistic, dan natural language processing, image processing dan lainnya.

Menurut Müller & Guido, python masih digunakan sebagai bahasa untuk pemrograman/pengolahan data statistik seperti machine learning karena bersifat open source, powerful, dan didukung banyak library machine learning yang bervariasi. Anaconda merupakan distribusi python untuk memproses data dengan skala yang lebih besar, predictive analytics, dan scientific computing. Anaconda sudah terdiri dari beberapa paket seperti Numpy, Scipy, matplotlib, pandas, IPython, Jupyter Notebook, dan scikitlearn. Anaconda tersedia untuk semua sistem operasi, merupakan solusi yang sangat mudah dan sangat disarankan karena tidak perlu menginstal paket python satu-persatu.

#### 2.4 Framework Django

Django adalah sebuah framework full-stack untuk membuat aplikasi web dengan bahasa pemrograman Python. Framework akan membantu membuat web lebih cepat, dibandingkan menulis kode dari nol. Full-stack artinya, django meliputi sisi front-end dan juga back-end. Front-end adalah sisi depan yang akan dilihat oleh pengguna, sedangkan back-end adalah sisi belakang yang berhubungan dengan database dan logika bisnis. Django adalah high-level Python Web framework tingkat tinggi yang mendorong pengembangan atau development dengan cepat dan desain pragmatis yang bersih.

Django adalah kerangka kerja Python Web tingkat tinggi yang mendorong pengembangan cepat dan desain pragmatis yang bersih. Kerangka kerja Web adalah seperangkat komponen yang menyediakan cara standar untuk mengembangkan situs web dengan cepat dan mudah. Tujuan utama Django adalah untuk memudahkan pembuatan situs web yang digerakkan oleh basis data yang kompleks. Beberapa situs terkenal yang

menggunakan Django termasuk PBS, Instagram, Disqus, Washington Times, Bitbucket dan Mozilla.

Kelebihan dari Django dibandingkan framework lain diantaranya:

- (a) Merepresentasikan data model dari basis data yang dimuat dalam bentuk objek, sesuai dengan konsep pemrograman kekinian yang berbasis object - oriented programming (OOP).
- (b) Django memahami situasi ketika kita harus menjaga kerahasiaan dan integritas data, Django menyediakan admin interface secara otomatis, tanpa perlu repot membuatnya. Halaman admin ini tidak boleh diakses oleh sembarang orang dan membutuhkan waktu yang relatif lama jika membuatnya sendiri.
- (c) Pembuatan URL yang lebih mudah dan fleksibel.
- (d) Sistem template yang powerful dan mudah dipelajari baik itu untuk web developer maupun web design serta mendukung template hierarchy (pewarisan).
- (e) Sistem cache dapat digunakan dengan mudah.
- (f) Mempermudah web developer dalam membuat web multibahasa.
- (g) Tidak perlu menginstall web server selama proses development, cukup dijalankan saja.

#### 2.5 Bootstrap CSS

Bootstrap adalah framework HTML, CSS, dan JavaScript yang berfungsi untuk mendesain website *responsive* dengan cepat dan mudah. Framework open source ini diciptakan pada tahun 2011 oleh Mark Otto dan Jacob Thornton dari Twitter. Itulah kenapa dulunya Bootstrap dinamakan Twitter Blueprint. Bootstrap dengan cepat meraih popularitas digunakan oleh 27% website di seluruh dunia. Hal itu karena kese-

derhanaan dan konsistensi yang ditawarkan Bootstrap dibanding framework lainnya saat itu. Kemudahan yang ditawarkan oleh Bootstrap adalah Anda tak perlu *coding* komponen website dari nol. Framework ini tersusun dari kumpulan file CSS dan JavaScript berbentuk *class* yang tinggal pakai.

*Class* yang disediakan Bootstrap juga cukup lengkap. Mulai dari *class* untuk layout halaman, *class* menu navigasi, *class* animasi, dan masih banyak lainnya. Menariknya lagi, Bootstrap bersifat responsive berkat grid system yang digunakan. Sistem grid pada bootstrap menggunakan rangkaian *containers*, baris, dan kolom untuk menyesuaikan bentuk layout dan konten website Anda. Dengan kata lain, Bootstrap menjamin tampilan website Anda akan tetap rapi dan konsisten di berbagai perangkat pengunjung. Baik melalui smartphone, tablet, atau laptop.

## 2.6 SQLite

SQLite adalah perpustakaan perangkat lunak yang menerapkan engine database SQL secara mandiri, tanpa memerlukan server, tanpa perlu melakukan konfigurasi, dan bersifat transaksional. SQLite adalah engine database SQL yang paling banyak digunakan di dunia. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D.Richard Hipp.

SQLite adalah sebuah engine database SQL yang langsung tertanam atau pada aplikasi. Tidak seperti kebanyakan database SQL lainnya, SQLite tidak memiliki server yang terpisah dari aplikasi. SQLite membaca dan menulis langsung ke file disk biasa. Database SQLite memiliki fitur lengkap dengan banyak tabel, indexes, trigger, dan tampilan, serta tersimpan pada satu file tunggal dalam hard-disk. Format file

databasenya bersifat cross-platform. Sehingga dapat dengan bebas menyalin database antara sistem 32-bit dan 64-bit atau antara arsitektur yang berbeda platform. Fitur-fitur ini membuat SQLite menjadi pilihan populer sebagai Application File Format.

## 2.6 UML (United Modelling Language)

“Unified Modeling Language (UML) merupakan “Salah satu standar bahasa yang banyak digunakan di dunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” (Rosa dan Shalahuddin, 2016).

UML terdiri dari beberapa diagram diantaranya:

- a) Use case Diagram: Diagram ini memperlihatkan himpunan use-case dan aktor- aktor (suatu jenis khusus dari kelas).
- b) Activity Diagram: Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem.
- c) Sequence Diagram: diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu
- d) Class Diagram: Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi.

## 2.7 Metode Pengembangan Sistem

Model proses yang digunakan pada penelitian ini yaitu, Model Sekuensial Linier atau Waterfall Development Model. Model ini mengusulkan sebuah pendekatan perkembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat dan kemajuan sistem pada seluruh tahapan analisis, desain, kode, pengujian, dan pemeliharaan.

- a) Rekayasa dan pemodelan sistem/informasi.  
Langkah pertama dimulai dengan membangun keseluruhan elemen sistem dan memilah bagian-bagian mana yang akan dijadikan bahan pengembangan perangkat lunak, dengan memperhatikan hubungannya dengan *Hardware*, *User*, dan *Database*.
- b) Analisis kebutuhan perangkat lunak  
Pada proses ini, dilakukan penganalisaan dan pengumpulan kebutuhan sistem. Hasil penganalisaan dan pengumpulan tersebut didokumentasikan dan diperlihatkan kembali kepada pelanggan.
- c) Desain  
Pada proses desain, dilakukan penjerjemahan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuatnya proses pengkodean (coding). Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi interface, dan detail algoritma prosedural.
- d) Pengkodean  
Pengkodean merupakan proses menerjemahkan perancangan desain ke bentuk yang dapat dimengerti oleh mesin, dengan menggunakan bahasa pemrograman.
- e) Pengujian  
Setelah proses pengkodean selesai, dilanjutkan dengan proses pengujian pada program perangkat lunak, baik pengujian logika internal, maupun pengujian eksternal fungsional untuk memeriksa segala kemungkinan terjadinya kesalahan dan memeriksa apakah hasil dari pengembangan tersebut sesuai dengan hasil yang diinginkan.
- f) Pemeliharaan  
Proses pemeliharaan merupakan bagian paling akhir dari siklus

pengembangan dan dilakukan setelah perangkat lunak dipergunakan. Kegiatan yang dilakukan pada proses pemeliharaan antara lain:

- 1) Corrective Maintenance: yaitu mengoreksi apabila terdapat kesalahan pada perangkat lunak, yang baru terdeteksi pada saat perangkat lunak dipergunakan.
- 2) Adaptive Maintenance : yaitu dilakukannya penyesuaian/perubahan sesuai dengan lingkungan yang baru, misalnya hardware, periperal, sistem operasi baru, atau sebagai tuntutan atas perkembangan sistem komputer, misalnya penambahan driver, dll.
- 3) Perfektive Maintenance: Bila perangkat lunak sukses dipergunakan oleh pemakai. Pemeliharaan ditujukan untuk menambah kemampuannya seperti memberikan fungsi-fungsi tambahan, peningkatan kinerja dan sebagainya.

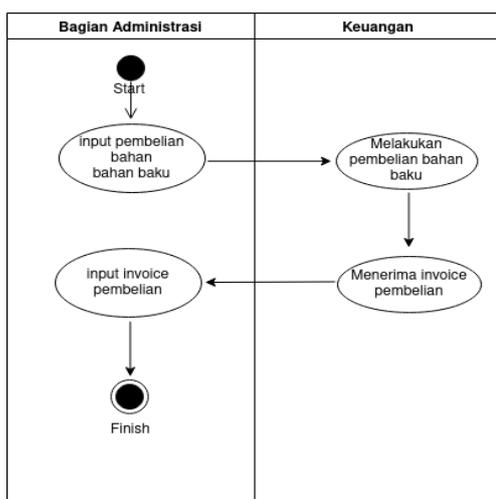
### III. ANALISA SISTEM BERJALAN

#### 3.1 Tinjauan institusi/Perusahaan

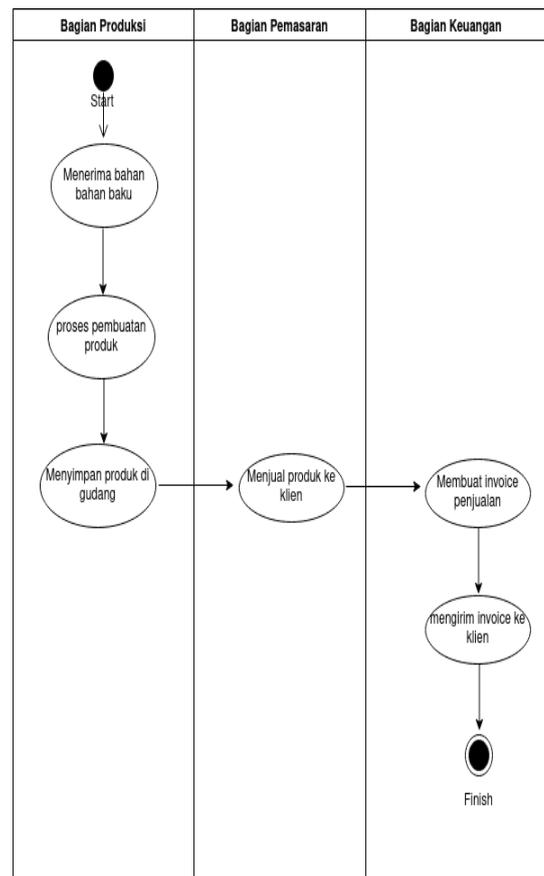
Badan usaha Lampuind Tekno Elektrik berawal dari kepedulian terhadap lingkungan di Indonesia yang masih terdapat daerah dan desa yang masih belum memiliki penerangan listrik. Terdapat 2500 desa dari 82.395 desa di Indonesia yang masih belum teraliri listrik. Di sisi lain di Indonesia sendiri memiliki potensi yang sangat tinggi akan energi matahari. Energy matahari yang dapat dihasilkan di Indonesia memiliki nilai rata-rata 4 Kw / m<sup>2</sup> yang setara dengan 10 kali energy matahari di Jerman. Tomy Abuzairi bersama rekan pendiri awalnya Kresna Devara yang sekaligus sebagai kepala divisi pengembangan alat sebelumnya sudah berpengalaman membuat model purwa yang didesain untuk daerah terpencil. Pada tahun 2017 mereka memutuskan

untuk mengembangkan produk awal mereka dari model purwa rupa tersebut dengan system power conserve. Produk awal Lampuind adalah lampu jalan yang ditujukan untuk lokasi terpencil yang sulit dijangkau listrik secara konvensional, lampu tersebut memiliki system yang system penerangannya dipengaruhi oleh intensitas cahaya yang ada pada lingkungan sehingga daya yang digunakan akan menyesuaikan dengan terangnya kondisi sekitar yang kemudian sistem produk ini dipatenkan. Pada tahun 2018 untuk memperbesar bisnis mereka lampuind kemudian mengikuti program inkubasi bisnis start-up yang diadakan oleh Direktorat Inovasi dan Inkubasi Bisnis Universitas Indonesia (DIIB UI) yang kemudian mendapat pendanaan. Ditahun yang sama sedang melakukan pengembangan produk untuk produk keduanya. Produk keduanya adalah pengembangan dari produk pertama mereka namun dibuat lebih kompak dan mudah dibawa.

### 3.2 Proses Bisnis Sistem



Gambar 3.1 Proses bisnis system pembelian bahan baku produk



Gambar 3.2 Proses bisnis sistem pembuatan produk dan pemasaran produk

## IV PEMBAHASAN

### 4.1 Analisa Kebutuhan Software

Analisa kebutuhan perangkat lunak ini berguna untuk menentukan rancangan sistem yang akan dibuat sesuai dengan permasalahan yang ditemukan di dalam sistem berjalan sangat lambat, guna memperbaiki sistem tersebut.

#### A) Tahapan Analisis

Sistem invoice generator adalah sistem pembuatan invoice secara online berbasis web dimana user dapat membuat invoice secara langsung. User membuat melalui media browser. Berikut ini spesifikasi kebutuhan (system requirement) dari sistem invoice generator;

Halaman Front-page:

A1. User bisa melakukan login.

A2. User bisa melihat jumlah total invoice, invoice dibayar.

- A3. User bisa melihat jumlah total klien.
- A4. User bisa menambahkan invoice baru.
- A5. User bisa menambahkan produk baru.
- A6. User bisa menambahkan klien baru
- A7. User bisa melihat profil perusahaan.

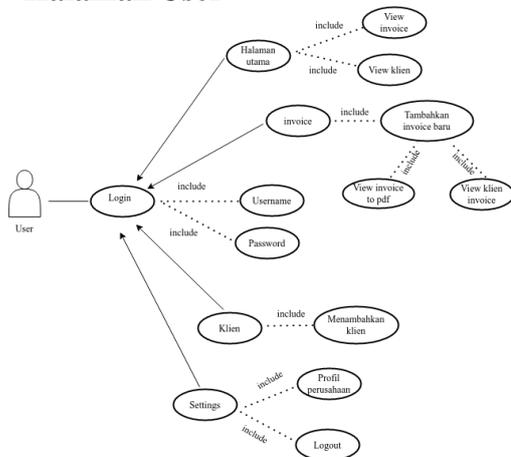
Halaman administrasi:

- B1. Admin dapat mengelola izin dari user.
- B2. Admin dapat mengelola klien.
- B3. Admin dapat mengelola invoice.
- B4. Admin dapat mengelola produk.

**B) Use Case Diagram**

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk diagram yang dapat menjelaskan suatu alur proses sistem yang akan di bangun. Dalam penulisan skripsi ini penulis menggunakan metode UML yang dalam metode itu penulis menerapkan diagram use case. Maka digambarlah suatu bentuk diagram use case yang dilihat pada Gambar IV.4

**1. Use Case Diagram Invoice Online Halaman User**



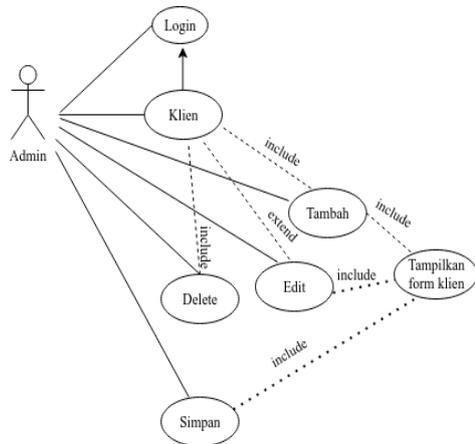
Gambar 4.1 Use Case Diagram Invoice Online Halaman User

Deskripsi *use case diagram* invoice online halaman user:

Tabel 4.1 Use case diagram halaman user

<b>Use Case Name</b>	<b>Menambahkan invoice baru</b>
<b>Requirements</b>	A4, A5, A6
<b>Goal</b>	User dapat membuat invoice secara online melalui website
<b>Pre-conditions</b>	User telah memilih invoice
<b>Post-conditions</b>	System mengkonvert invoice via pdf
<b>Failed and conditions</b>	User menghapus invoice. Kemungkinan user salah menginput produk, tidak memasukan nama klien, atau merubah status pembayaran
<b>Primary actors</b>	User
<b>Main flow / Basic path</b>	<ol style="list-style-type: none"> <li>1. User melakukan login</li> <li>2. User memilih menu invoice</li> <li>3. User klik tombol Tambahkan invoice baru</li> <li>4. System menampilkan template membuat invoice</li> <li>5. User memasukan produk</li> <li>6. User memasukan nama klien</li> <li>7. User memasukan nama invoice</li> <li>8. User memasukan tanggal jatuh tempo</li> <li>9. User memasukan syarat pembayaran</li> <li>10. User memasukan catatan pada invoice, atau merubah status pembayaran</li> </ol>
<b>Invariant A :</b>	<p>A8. User klik view klien invoice</p> <p>A9. System menampilkan invoice yang telah dibuatkan dalam format pdf</p> <p>A10. User klik pdf invoice</p> <p>A11. System akan menampilkan invoice dan dapat di download dalam format pdf</p>

a) **Use case diagram Mengelola Data Klien**



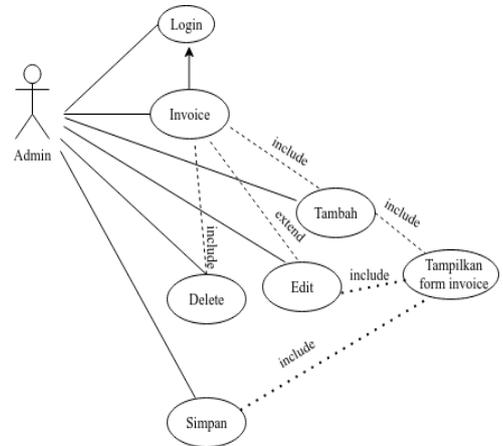
Gambar 4.2 Use case diagram mengelola data klien

Deskripsi use case diagram mengelola data klien

Tabel 4.2 Use case diagram mengelola data klien

<b>Use Case Name</b>	Menambahkan invoice baru
<b>Requirements</b>	B3
<b>Goal</b>	Admin dapat menambah, mengedit dan menghapus data barang.
<b>Pre-conditions</b>	Admin telah login
<b>Post-conditions</b>	Data invoice tersimpan, terhapus atau terupdate
<b>Failed and conditions</b>	Gagal menyimpan, mengupdate atau menghapus
<b>Primary actors</b>	Administrator
<b>Main flow / Basic path</b>	1. Admin melihat daftar invoice. 2. Admin menambah data invoice 3. Admin menyimpan data invoice.
<b>Invariant A :</b>	2a. Admin mengedit data invoice. 2b. Admin menghapus data invoice

b) **Use case diagram Mengelola Data Invoice**



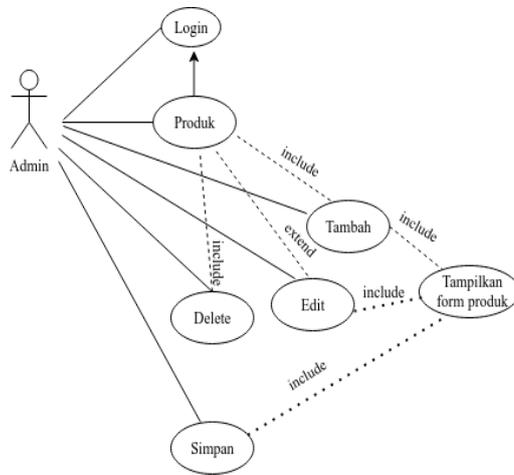
Gambar 4.3 Use case diagram mengelola data invoice

Deskripsi use case diagram mengelola data invoice

Tabel 4.3 Use case diagram mengelola data invoice

<b>Use Case Name</b>	Menambahkan klien baru
<b>Requirements</b>	B2
<b>Goal</b>	Admin dapat menambah, mengedit dan menghapus data barang.
<b>Pre-conditions</b>	Admin telah login
<b>Post-conditions</b>	Data klien tersimpan, terhapus atau terupdate
<b>Failed and conditions</b>	Gagal menyimpan, mengupdate atau menghapus
<b>Primary actors</b>	Administrator
<b>Main flow / Basic path</b>	1. Admin melihat daftar klien. 2. Admin menambah data klien 3. Admin menyimpan data klien.
<b>Invariant A :</b>	2a. Admin mengedit data klien. 2b. Admin menghapus data klien

**c) Use case diagram Mengelola Data Produk**



Gambar 4.4 Use case diagram mengelola data produk

Deskripsi use case diagram mengelola data produk.

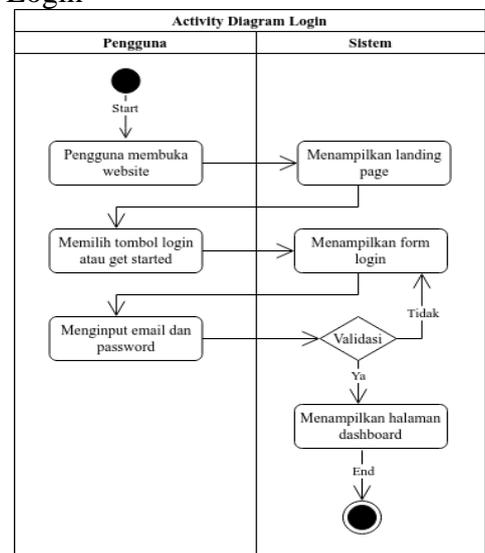
Tabel 4.4 Use case diagram mengelola data produk

<b>Use Case Name</b>	<b>Menambahkan produk baru</b>
<b>Requirements</b>	B4
<b>Goal</b>	Admin dapat menambah, mengedit dan menghapus data barang.
<b>Pre-conditions</b>	Admin telah login
<b>Post-conditions</b>	Data produk tersimpan, terhapus atau terupdate
<b>Failed and conditions</b>	Gagal menyimpan, mengupdate atau menghapus
<b>Primary actors</b>	Administrator
<b>Main flow / Basic path</b>	1. Admin melihat daftar produk. 2. Admin menambah data produk. 3. Admin menyimpan data produk.
<b>Invariant A :</b>	2a. Admin mengedit data produk. 2b. Admin menghapus data produk

**C) Activity Diagram**

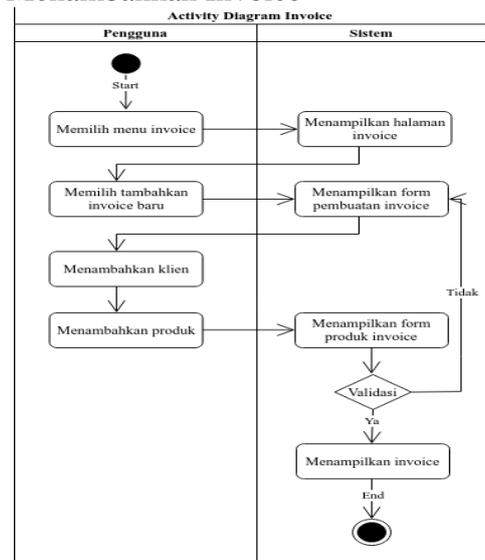
Activity Diagram adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu di ketahui adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun keterangan dari Activity Diagram dari beberapa sistem adalah sebagai berikut:

1. Login



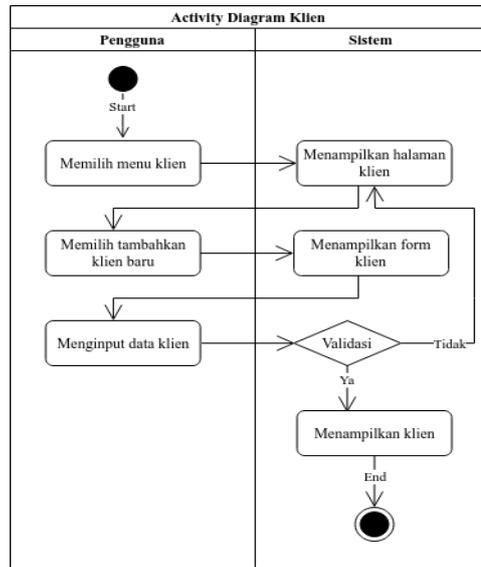
Gambar 4.5 Activity diagram login

2. Menambahkan Invoice



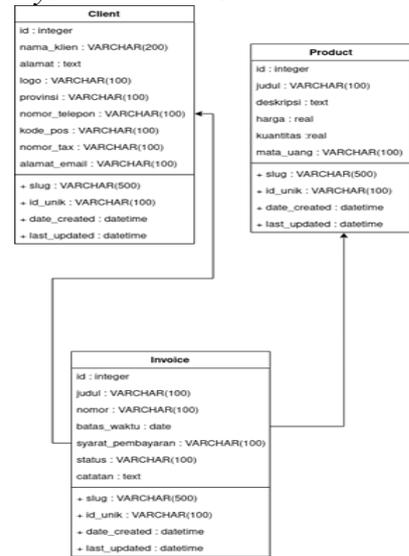
Gambar 4.6 Activity menambahkan invoice baru

### 3. Menambahkan Klien



Gambar 4.7 Activity diagram menambahkan klien baru

### b) Physical Data Model



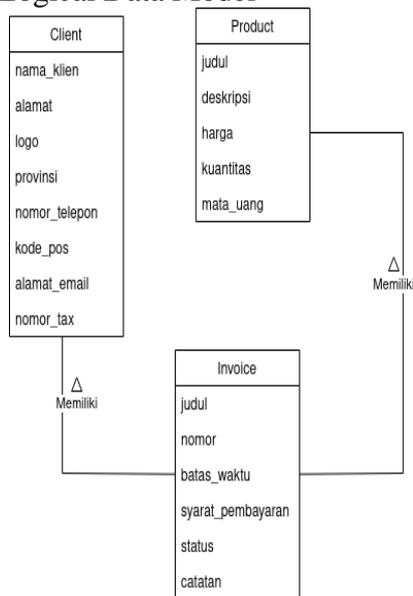
Gambar 4.9 Physical data model invoice online

## 4.2 Desain

Desain sistem tidak hanya meliputi desain atau apa saja yang ditampilkan didalam. Pada tahapan ini akan menjelaskan tentang desain database, desain software architecture dan desain interface dari sistem yang sedang dibuat.

### 1. Database

#### a) Logical Data Model



Gambar 4.8 Logical data model invoice online

## 2. Software Architecture

### a) Class Diagram

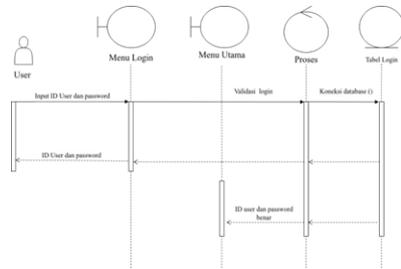
Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, dan relasi-relasi. Berikut ini adalah objek-objek yang teridentifikasi dari sistem invoice online:

1. Invoice
2. Client
3. Product
4. Setting

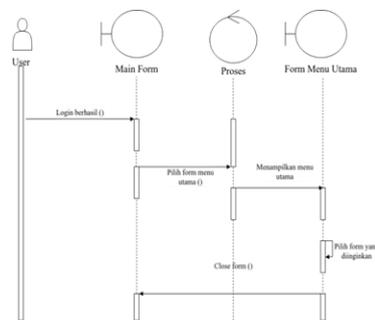


Gambar 4.10 Class diagram invoice online

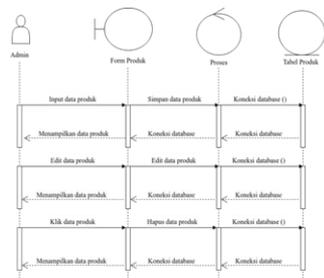
- b) **Sequence Diagram**  
Diagram yang menggambarkan rincian bagaimana suatu operasi teknis dilakukan.



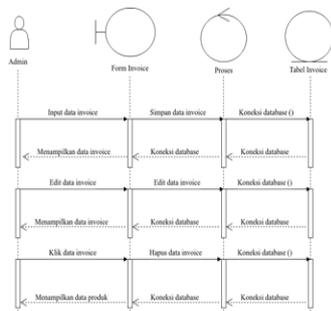
Gambar 4.11 Sequence diagram login



Gambar 4.12 Sequence Halaman Utama



Gambar 4.14 Sequence Diagram Data Produk



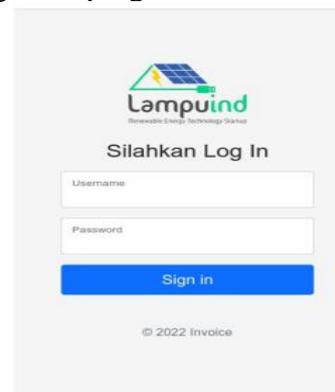
Gambar 4.15 Sequence Diagram Data Invoice

### 3. Desain Tampilan Program

Berikut beberapa Rancangan tampilan program untuk sistem informasi manajemen invoice pada PT. Lampuind Tekno Elektrik

- a) **Tampilan Login User**

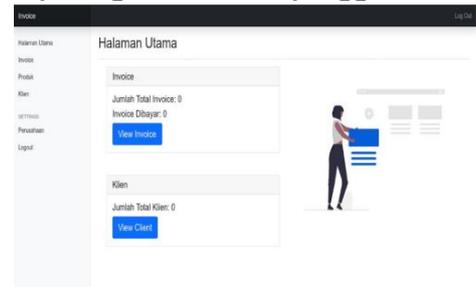
Form ini digunakan untuk akses masuk ke aplikasi manajemen invoice. Form ini merupakan salah satu bentuk keamanan karena hanya user yang mempunyai akun yang dapat mengakses program



Gambar 4.16 Tampilan Login User

- b) **Tampilan Halaman Utama**

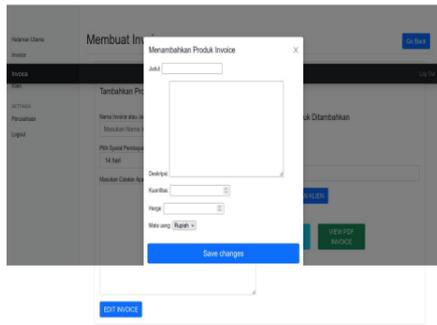
Form ini menampilkan pilihan Menu yang tersedia di dalam aplikasi yang dapat digunakan oleh pengguna.



Gambar 4.17 Tampilan Halaman Utama

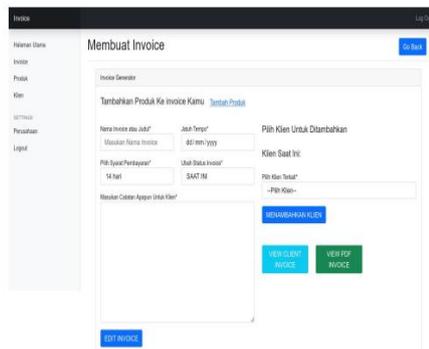
- c) **Tampilan Menambah Data Produk**

Form ini digunakan untuk menambah data produk



Gambar 4.18 Tampilan Tambah Data Produk

- d) Tampilan Menambah Data Invoice Form ini digunakan untuk menambahkan data invoice



Gambar 4.19 Tampilan Tambah Data Invoice

## V, KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Dalam ulasan kesimpulan menanggapi kasus yang terdapat serta berisi tentang kelebihan serta kekurangan sistem yang di buat. Cocok dengan apa yang sudah dibahas pada bab-bab lebih

dahulu, hingga penulis mengambil kesimpulan selaku berikut:

1. Sistem manajemen berbasis website ini dirancang buat mempermudah pembuatan invoice.
2. Pendataan serta pengaturan posisi invoice pengaruhi proses mencari serta menciptakan segala dokumen yang dibutuhkan.
3. Penyajian informasi serta data dikira user friendly, memudahkan pekerjaan yang sepanjang ini dicoba dengan metode manual.

### 5.2 Saran

Anjuran dalam sistem ini disadari masih banyak kekurangan serta kelemahan, oleh sebab itu buat pengembangan berikutnya, penulis membagikan anjuran:

1. Buat tingkatkan kinerja sistem website ini, hendaknya pengembangan memaintaince mulai dari sesi tampilan taman website hingga dengan penjaannya supaya tidak terkesan monoton serta lebih efisien.
2. Sistem data pembuatan invoice ini ialah sesuatu sistem data yang belum sempurna, sehingga butuh dicoba pengembangan serta revisi sistem yang cocok untuk kebutuhan pengguna.
3. Butuh terdapatnya penyesuaian terhadap sistem manajemen yang terdapat pada industri.

## DAFTAR PUSTAKA

- [1] Subianto, “Perancangan Sistem Pengelolaan Dokumen Invoice Berbasis Web,” *Inf. Komput. Akunt. Dan Manaj.*, vol. 16, no. 2, pp. 951–952, 2021.
- [2] Mirza M. Haekal, “Bootstrap: Pengertian, Kegunaan, Kelebihan, dan Kekurangannya,” *www.niagahoster.co.id*, 2021. <https://www.niagahoster.co.id/blog/bootstrap-adalah/> (accessed Aug. 16, 2021).
- [3] WebFrameworks, “Web Frameworks for Python,” *wiki.python.org*, 2020. <https://wiki.python.org/moin/WebFramework> (accessed May 04, 2020).
- [4] Real Python, “Django Tutorials,” *realpython.com*, 2020. <https://realpython.com/tutorials/django/> (accessed Jul. 15, 2020).
- [5] MDN, “Django Web Framework (Python),” *developer.mozilla.org*, 2020. <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction> (accessed Jul. 15, 2020).
- [6] Yudana, “SQLite Sistem Manajemen Basis Data Yang Berukuran Kecil,” *yudana.id*, 2017. <https://www.yudana.id/sqlite-sistem-manajemen-basis-data-berukuran-kecil/> (accessed Jun. 03, 2017).
- [7] Django, “Meet Django,” *www.djangoprojects.com*, 2020. <https://ww16.djangoprojects.com/?sub1=20220628-0147-58ff-8826-fd848d71d0fa> (accessed Feb. 13, 2020).
- [8] Kezia Rafinska, “Aplikasi Berbasis Web dan Keunggulannya yang Bisa Anda Ketahui,” *www.online-pajak.com*, 2020. <https://www.online-pajak.com/seputar-efiling/aplikasi-berbasis-web> (accessed Feb. 12, 2020).
- [9] J. Sovia, Rini dan Febio, “MEMBANGUN APLIKASI E-LIBRARY MENGGUNAKAN HTML, PHP SCRIPT, DAN MYSQL DATABASE Rini Sovia dan Jimmy Febio,” *Processor*, vol. 6, no. 2, pp. 38–54, 2011.
- [10] A. Pajankar and A. Joshi, *Introduction to Machine Learning with Scikit-learn*. 2022. doi: 10.1007/978-1-4842-7921-2\_5.
- [11] A. Musyafa *et al.*, “Faktur Berbasis Web Pada Pt Indah Jaya Textile Design Information System Invoices Receipt,” vol. 1, pp. 133–142, 2017.
- [12] M. Rahmayu, R. K. Serli, and S. A. Hanawi, “Sistem Pelayanan E-Invoicing Bulanan Pada Apartemen Permata Surya 1 Jakarta,” *JUSIM (Jurnal Sist. Inf. Musirawas)*, vol. 4, no. 1, pp. 18–26, 2019, doi: 10.32767/jusim.v4i1.407.
- [13] Awaludin, M., & Amelia, L. V. (2022). Penerapan Structural Equation Modeling (Sem) Dengan Lisrel Terhadap Perbedaan Tarif Penerbangan Pada Penumpang Domestik Di Bandara Halim Perdanakusuma. *Jurnal Sistem Informasi Universitas Suryadarma*, 9(1). <https://doi.org/10.35968/jsi.v9i1.855>

