FORMAT: Jurnal Ilmiah Teknik Informatika
Akreditasi Sinta 5 RISTEKBRIN No. 164/E/KPT/2021; E-ISSN 2722-7162 | P-ISSN 2089-5615

Vol. 14, No. 1,
January 2025,
90-111. 2025

Implementasi Algoritma Quick Sort Pada Aplikasi Pemrograman Berorientasi Objek Berbasis Python Dan Mysql Workbench

(Studi Kasus: Aplikasi Sistem Manajemen Piutang Jangka Panjang Lainnya)

Mohamad Yusuf [1] ; Faaza Naimah [2]

[1,2] *Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Mercu Buana, Jakarta, Indonesia*

[1] mhd.yusuf@mercubuana.ac.id, [2] ifafaaza@gmail.com

Kata kunci:
Python, Tkinter, Incremental,, Quick Sort, White-box Testing (Unittest)

Abstract

This study aims to develop another long-term receivables management system based on Python, using Tkinter for the GUI, and MySQL Workbench as the database administrator. The system development method employs an incremental approach with two main stages: CRUD for basic data management operations and general information presented through data visualization. This study focuses on implementing the performance of the Quick Sort algorithm within an Object-Oriented Programming (OOP) application. The results of system testing with the white-box testing method, using Python Unittest, demonstrate the consistency of functions and logic as expected. The system was tested on 108 cases of other long-term receivables information, successfully providing ease of use, data accuracy, speed of access, and improved information visualization.

Pendahuluan

Dalam konteks modernisasi pengelolaan keuangan sektor publik, peran teknologi informasi menjadi sangat penting sebagai pendorong utama perubahan dan peningkatan efisiensi operasional. Fokus utama hal ini adalah pada kemajuan teknologi informasi sebagai katalisator utama dalam membentuk paradigma baru dalam tata kelola keuangan organisasi publik, khususnya dalam pengelolaan piutang jangka panjang lainnya.

Penerapan teknologi informasi dalam pengelolaan piutang jangka panjang di sektor publik bertujuan untuk memberikan kontribusi positif terhadap peningkatan efisiensi dan efektivitas di dalam organisasi. Hal ini selaras dengan Hadi [16] bahwa perkembangan TIK serta penerapan konektivitas internet ke dalam tata kelola pemerintah diharapkan mampu mengatasi berbagai macam persoalan melalui peningkatan efisiensi, inovasi, produktivitas, perluasan jangkauan, dan penghematan biaya. Inovasi digitalisasi pada penelitian ini mewakili kebaruan bahwa inovasi berkaitan dengan sesuatu yang baru bagi orang, organisasi, masyarakat atau situasi tertentu [1].

Tujuan dari penelitian ini, diharapkan proses identifikasi, pengumpulan, verifikasi, penetapan, pemantauan, evaluasi, dan pemantauan data pendukung laporan keuangan terkait piutang jangka panjang lainnya dapat dilakukan dengan lebih efisien, konsisten, dan akurat. Adapun penerapan teknologi informasi berupa pengimplementasian algoritma quick sort dan pemrograman berorientasi objek yang dibangun dengan Bahasa pemrograman Python diartikulasikan sebagai langkah proaktif untuk mencapai tata kelola keuangan yang transparan, efisien, dan sesuai dengan standar manajemen organisasi sektor publik.

Tinjauan Pustaka

A. Incremental Method

Pressman [5] mengatakan bahwa metode inkremental merupakan salah satu metode pengembangan perangkat lunak yang mampu meminimalisir ketidaksesuaian dalam proses pengembangan perangkat lunak. Pada metode ini, tahapan dalam metodologi terdapat masukan dan keluaran. Keluaran dari proses sebelumnya akan menjadi masukan pada proses inkrement selanjutnya [11].

Pada metode inkremental, setiap tahapan yang ada dalam metodologi terdapat masukan (input) dan keluaran (output). Metode ini juga merupakan perbaikan dari metode waterfall dengan ide dasar pengembangan software secara meningkat (increment) atau bertahap berdasarkan kemampuan fungsional [9].

B. Object-Oriented Programming

Pemrograman Berorientasi Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya [8].

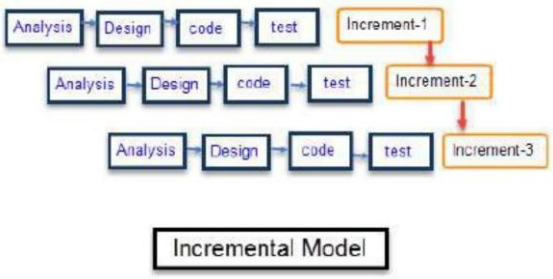
Pemrograman Berorientasi Objek merupakan pemrograman yang menggunakan object dan class sebagai representasi dari permasalahan dan sistem dari dunia nyata. Sistem yang berkonsep pemrograman berorientasi objek memiliki sebuah fungsi serta kumpulan data yang dikelompokkan ke dalam komponen yang dienkapsulasi ke dalam suatu bentuk objek. Sehingga dalam hal ini, setiap objek dapat mewariskan sifatnya atau setiap objek yang berbeda. Berdasarkan hal tersebut, kumpulan objek yang berinteraksi satu sama lain akan menghasilkan output seperti yang diinginkan.

C. Algoritma Quick Sort

Algoritma quick sort merupakan metode yang membandingkan suatu elemen (pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen lain yang lebih kecil dari pivot tersebut terletak di sebelah kirinya dan elemen-elemen lain yang lebih besar daripada pivot tersebut terletak di sebelah kanan. Sehingga dengan demikian terbentuk dua sublist yaitu yang terletak disebelah kiri pivot dan sebelah kanan pivot [4].

Metode Penelitian

Pada penelitian ini digunakan *incremental model* dengan pertimbangan bahwa metode ini dapat meminalisir ketidaksesuaian dalam pengembangan perangkat lunak [5]. Berikut merupakan tahapan dalam perancangan sistem menggunakan pendekatan metode incremental.



Gambar 1 Tahapan Incremental Model

Setiap tahapan penelitian pada metode inkremental terdapat masukan (input) dan keluaran (output), sebagaimana tabel berikut.

No	Aktivitas	Input	Output
1	Studi Literatur	• Latar Belakang • Permasalahan • Tujuan Penelitian	• Informasi UML • Informasi metode pengembangan aplikasi
2	Analisa Kebutuhan	Pengumpulan data dan fakta	• Data Excel berupa Matriks Informasi Piutang Jangka Panjang Lainnya (Kerugian Negara) • Data Excel berupa Matriks Piutang Jangka Panjang Lainnya (Kerugian Negara) • Data Excel berupa Matriks Piutang Negara
3	Desain dan Pengkodean	Informasi Proses Bisnis	Model Proses Bisnis
		Model Proses Bisnis	Implementasi Pengkodean
4	Testing	*Source code* kelas-kelas yang belum teruji	*Source code* telah teruji dengan Metode White Box Testing (Unittest)
5	Deployment	Realease *source code* telah teruji	Aplikasi yang sudah terpasang di dalam server local
6	Uji Coba: Studi Kasus	Aplikasi yang sudah terpasang di dalam server lokal	Implementasi Tambah 108 Kasus Informasi Kerugian Negara melalui GUI pada database MySQL Workbench

Hasil Dan Pembahasan

Proses bisnis pada penyelesaian tindak lanjut atas temuan pemeriksaan Pengawas Eksternal dan pengawasan Pengawas Internal, terdapat 2 jenis temuan, yaitu Sistem Pengendalian Internal dan Kepatuhan. Adapun temuan Kepatuhan merupakan merupakan salah satu pemeriksaan dengan tujuan tertentu (PDTT) untuk melakukan pemeriksaan terhadap ketentuan Peraturan Perundang-Undangan. Lebih lanjut, terhadap hal tersebut terdapat kekurangan keuangan negara yang harus dikembalikan ke rekening kas umum negara (RKUN) melalui mekanisme tertentu yang diatur di dalam peraturan yang berlaku.

Pada sistem berjalan, terdapat beberapa pencatatan terhadap informasi kekurangan keuangan negara yang selanjutnya disebut informasi kerugian, sebelum ditetapkan sebagai piutang jangka panjang lainnya, ke dalam beberapa matriks. Berikut merupakan matriks-matriks terkait piutang jangka panjang lainnya pada instansi per tahun 2023:

a. Matriks Informasi Kerugian Negara Kementerian Semester II Tahun 2023

   Matriks berisi data **informasi** kerugian berupa Nomor, ID, Eselon I, Penanggung Jawab, Jabatan, Sumber Data, Tahun, Data Informasi (Jenis, Dokumen TP/TGR), Jumlah Nilai, Jumlah Angsuran, Jumlah Sisa, dan Keterangan. Pada matriks ini data masih berstatus sebagai "**informasi**" yang belum ditetapkan statusnya menjadi piutang jangka panjang lainnya. Di terdapat berbagai *worksheet* per eselon I yang diolah menjadi rekap per jenis temuan, yaitu pemeriksa eksternal (BPK) dan pemeriksa internal (Itjen) dan per jenis kerugian, yaitu Temuan Pemeriksaan (TP), Tuntutan Ganti Rugi (TGR), PIII (Pihak Ketiga).

Temuan	Jenis Kerugian	Jumlah Kasus	Progres	Sisa
Temuan BPK	TGR	1		1
	PIII	-		-
Temuan ITJEN	TP	-		-
	TGR	-		-
	PIII	1		1
Jumlah Informa	**TP**	**0**	**0**	**0**
	TGR	**1**	**0**	**1**
	PIII	**1**	**0**	**1**
Total		**2**	**0**	**2**

Gambar 2 Rekapitulasi Per Jenis Temuan dan Jenis Kerugian

Pada matriks juga terdapat rekapitulasi seluruh Eselon I per jenis temuan dengan spesifikasi total temuan dan sisa temuan. Data matriks ini akan menjadi bahan pemutakhiran data untuk ditetapkan menjadi piutang jangka panjang lainnya oleh pemeriksa eksternal.

No.	Unit Eselon I	Temuan BPK			Temuan Itjen			Total	Total	Sisa
		Jumlah	Progres	Sisa	Jumlah	Progres	Sisa	Temuan	Progres	
1	[redacted]	1	-	1	1	-	1	2	-	2
2	[redacted]	21	-	21	17	-	17	38	-	38
3	[redacted]	9	-	9	4	-	4	13	-	13
4	[redacted]	3			3					
5	[redacted]	8	-	8	-	-	-	8	-	8
6	[redacted]	14		14	21		21	35		35
7	[redacted]	8	-	8	8	-	8	16	-	16
8	[redacted]	-	-	-	8	-	8	8	-	8
	Jumlah	56	-	56	52	-	52	108	-	108

Gambar 3 Rekapitulasi Per Eselon I

a. Matriks Kerugian Negara Kementerian Semester II Tahun 2023

Matriks berisi data kerugian berupa

1) ID
2) Data Penanggung Jawab berupa Nama, NIP, Satuan Kerja
3) Data Kerugian berupa Jenis Kasus, Jenis Kerugian, Surat Pemberitahuan, SKTJM/SKP
4) Data Hasil Pemantauan berupa Jumlah Nilai, Total Angsuran, Total Sisa, dan Keterangan

Pada matriks ini data sudah berstatus "**penetapan**" yang selanjutnya dicatat pada piutang jangka panjang lainnya pada Catatan atas Laporan Keuangan. Di dalamnya juga terdapat histori piutang jangka lainnya baik yang telah lunas atau yang masih dalam proses pelunasan.

b. Matriks Data Piutang Kementerian per bulan September Tahun 2023

Pada matriks ini berisi data piutang baik jangka panjang maupun jangka pendek instansi yang dikelola oleh tim kerja yang berbeda. Sehingga diperlukan aktivitas rekonsiliasi sebelum dilakukan pemutakhiran data.

Berdasarkan data di atas, metode perancangan sistem menggunakan Metode Incremental sebagai pendekatan pengembangan sistem yang membagi sistem ke dalam beberapa tahap modul. Dalam hal ini, pembagian modul CRUD pada fase pertama dan modul Visualisasi pada fase kedua. Selanjutnya, aktivitas utama dari sistem yaitu untuk melakukan *update* dan monitoring Informasi Kerugian Negara pada Microsoft Excel berjudul Matriks Informasi Kerugian Negara dan Matriks Kerugian Negara, serta bahan rekonsiliasi data terhadap Matriks Piutang Negara pada instansi. Hal ini sesuai tujuan penelitian proses identifikasi, pengumpulan, verifikasi, penetapan, pemantauan, evaluasi, dan pemantauan data yang lebih efektif dan efisien.

A. Fase Pertama: CRUD

1) Identifikasi Kebutuhan

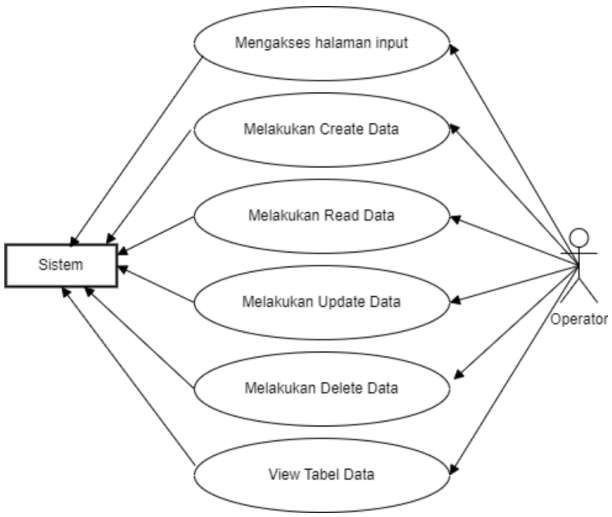
Identifikasi kebutuhan merupakan tahap pengumpulan data dan identifikasi kebutuhan fungsional dan non fungsional untuk mendukung kemudahan perancangan Sistem Manajemen. Berikut merupakan hasil identifikasi kebutuhan fungsional dan non fungsional Sistem Manajemen Piutang Jangka Panjang Lainnya Fase Pertama.

Tabel 1. Spesifikasi Kebutuhan Fase 1

Fungsional	
No	**Analisa Kebutuhan**
1	Sistem dapat menampilkan tampilan awal Sistem Manajemen
2	Sistem dapat melakukan unggah file dalam bentuk pdf, excel, dan word
3	Sistem dapat melakukan operasi matematika sederhana untuk menentukan Sisa
4	Sistem dapat melakukan Tambah data

5	Sistem dapat melakukan Edit data
6	Sistem dapat melakukan Hapus data
7	Sistem dapat melakukan Cari data
8	Sistem dapat melakukan Clear label input
9	Sistem dapat menampilkan tabel data tertentu dari database
10	Sistem dapat membuka jendela Informasi Umum saat ditekan button Info
11	Sistem dapat melakukan Keluar aplikasi
Non Fungsional	
11	Menggunakan pustaka/modul Tkinter sebagai antarmuka pengguna grafis GUI
12	Menggunakan MySQL Workbench sebagai database
13	Tampilan sistem *user-friendly*

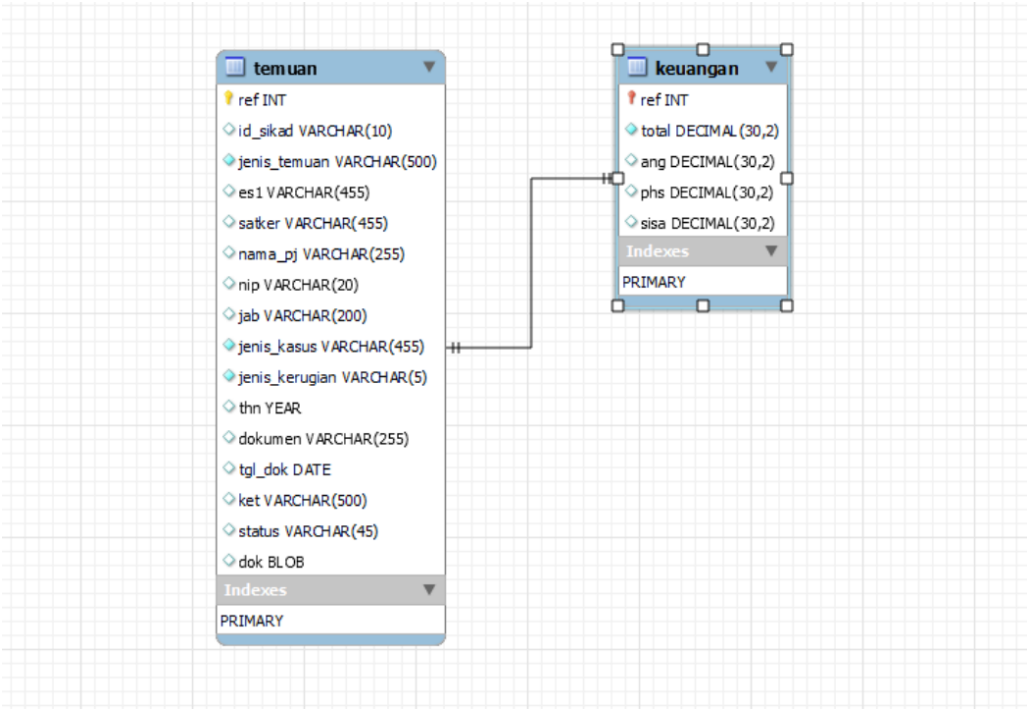
Berdasarkan Tabel di atas, berikut spesifikasi kebutuhan sistem yang dimodelkan dalam desain UML (*Unifed Modeling Language*) Fase 1.



Gambar 4 *Use Case Tahap 1*

Pada *use case diagram* terdapat operator sebagai aktor yang melakukan kegiatan CRUD sebagaimana pada gambar, yaitu mengakses halaman input, *create, read, update, delete,* dan *view table.*

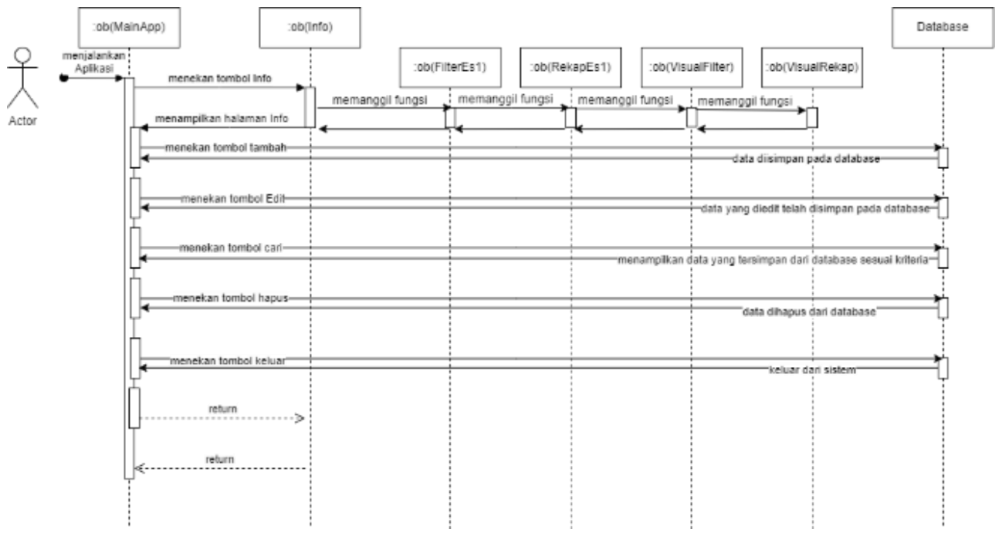
Berdasarkan identifikasi hubungan antar objek dalam perancangan sistem yang digambarkan melalui ERR (*Entity-Relationship Requirements*).



Gambar 5 *Entity-Relationship Requirements*

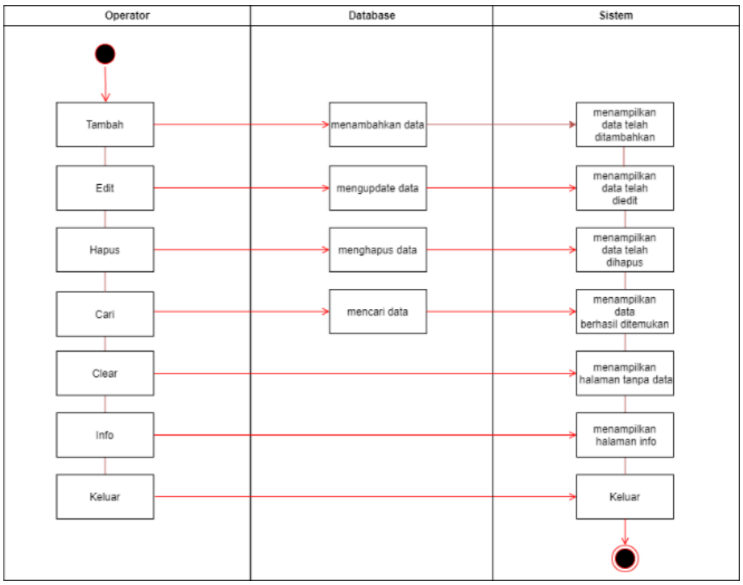
Terdapat 2 tabel, yaitu tabel temuan yang berisi informasi deksripsi dari kasus dan tabel keuangan yang berisi data nilai keuangan pada kasus. Dalam rangka efektifvitas dan efisiensi data, maka ditambahkan atribut status guna menjadi penghubung antar matriks sehingga dapat menjadi suatu database yang saling terhubung dan memiliki histori yang dinamis sesuai dengan kondisi terkini.

Berdasarkan use case, berikut merupakan interaksi antar entitas dalam scenario mengakses halaman input, *create*, *read*, *update*, *delete*, dan *view table.*



Gambar 6 Sequence Diagram Fase 1

Sedangkan aktivitas pada sistem yaitu terdapat aktivitas-aktivitas sebagaimana diagram berikut:

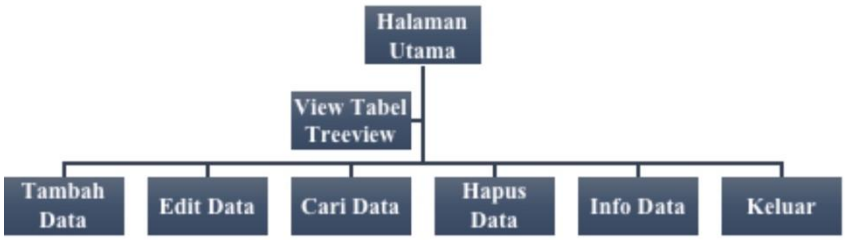


Gambar 7 *Activity Diagram* Fase 1

Desain dan Implementasi Pengkodean

1) Desain

Berikut merupakan desain navigasi Fase 2 yang menampilkan beberapa navigasi program.



Gambar 8 Desain Navigasi Fase 1

2) Implementasi Pengkodean

Pada proses coding, digunakan bahasa pemrograman Python, sistem database MySQL Workbench, source code editor Visual Studio Code dan MySQL Server.

Aplikasi menggunakan tkinter sebagai pustaka untuk membuat antarmuka pengguna (GUI) utama. Selain itu digunakan juga beberapa pustaka sebagai berikut untuk pengembangan sistem:

1. **ttk** (dari tkinter) untuk mengimplementasikan widget tematik yang lebih modern di dalam tkinter.
2. **mysql.connector** untuk berinteraksi dengan database MySQL Workbench
3. matplotlib.pyplot untuk membuat visualisasi, termasuk diagram batang (bar chart), FigureCanvasTkAgg dari matplotlib.backends.backend tkagg digunakan untuk menanamkan objek Figure ke dalam GUI tkinter.

4. numpy dan pandas untuk manipulasi dan analisis data terstruktur pada pengolahan data sebelum ditampilkan dalam diagram batang atau *sunburst chart.*

Paradigma pemrograman yang digunakan yaitu *object-oriented programming* dengan pembagian *class* menjadi 6 bagian, yang mana pada fase ini terdapat 1 *class*, yaitu:

1. MainApp

Berupa kode aplikasi halaman utama yang berisi CRUD, berupa

a. Tambah : Terdapat operasi pemasukan data ke dalam database keruneg melalui query INSERT. Pada aplikasi, data dimasukkan ke dalam tabel temuan dan keuangan dengan pemanggilan metode Tambah.

b. Read : Terdapat operasi pengambilan data dari database keruneg melalui query SELECT. Data diambil dari tabel temuan dan keuangan. Hal ini juga diimplementasikan pada metode Cari.

c. Edit : Terdapat operasi edit data dari database keruneg melalui query UPDATE. Pada aplikasi, data yang tersedia diubah dari tabel temuan dan keuanga dengan pemanggilan metode Edit.

d. Hapus : Terdapat operasi penghapusan data dari database keruneg melalui query DELETE. Pada aplikasi, data dihapus dari tabel temuan dan keuangan dengan pemanggilan metode Hapus.

Selain CRUD di atas, terdapat pula pendefinisian metode berikut:

a. Menampilkan data tersedia menggunakan modul ttk.Treeview dengan pemanggilan metode RefreshTable. Di dalam metode ini, dilakukan query_join berupa LEFT JOIN keuangan k ON t.ref = k.ref. Hal ini dikarenakan pada sistem basis data, k.ref merupakan foreign key sekaligus primary key dari tabel keuangan yang juga primary key pada tabel temuan.

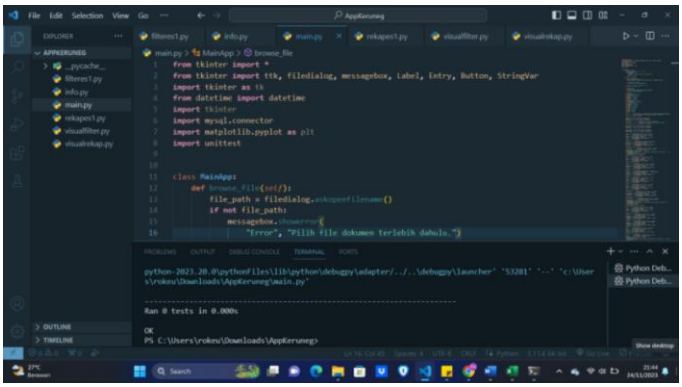
b. Menampilkan jendela tkinter baru berisi Informasi Umum dengan pemanggilan metode Show Info pada button Info.

c. Fungsi Clear untuk membersihkan semua kolom formulir, sehingga pengguna dapat memulai pengisian data baru.

d. Fungsi untuk keluar dari aplikasi berupa button Keluar yang memanggil metode Keluar

3) Pengujian dengan White-Box Testing

Pengujian dilakukan menggunakan framework unittest white-box testing dengan mengimpor unittest pada file Python. Pengujian ini meliputi unit-unit kecil dari kode, seperti fungsi atau metode, dengan memeriksa implementasi internal, termasuk juga pengujian sistem, integrasi, dan pengujian penerimaan yang melibatkan pengetahuan tentang struktur dan logika internal aplikasi. Berikut merupakan hasil dari pengujian white-box testing unittest pada Class Main App.



Gambar 9 *White Box Testing* Kelas MainApp

B. Fase Kedua: Visualisasi Informasi Umum

Berikut merupakan hasil identifikasi kebutuhan fungsional dan non fungsional Sistem Manajemen Piutang Jangka Panjang Lainnya Fase Kedua Bagian Visualisasi Informasi Umum.

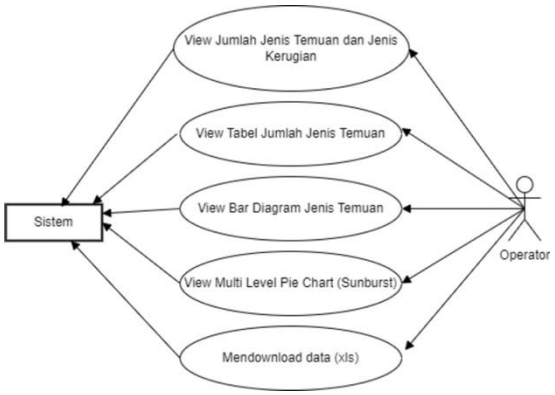
a) Identifikasi Kebutuhan

Tabel 3 Spesifikasi Kebutuhan Fase 2

Fungsional	
No	**Analisa Kebutuhan**
1	Sistem dapat menampilkan tampilan Informasi Umum
2	Sistem dapat menampilkan filter Rekap Jumlah Jenis Temuan dan Jenis Kerugian per Eselon I
3	Sistem dapat menampilkan Rekapitulasi Jumlah Jenis Temuan dan Jenis Kerugian per Eselon I
4	Sistem dapat menampilkan Rekapitulasi Jumlah Jenis Temuan dan Jenis Kerugian per Eselon I berupa diagram ke samping
5	Sistem dapat menampilkan Filter Rekapitulasi Jumlah Jenis Temuan dan Jenis Kerugian per Eselon I berupa Multi Level Pie Chart
6	Sistem dapat melakukan unduh excel dari database
7	Sistem dapat melakukan Keluar jendela Informasi Umum
Non Fungsional	
8	Menggunakan pustaka/modul Tkinter sebagai antarmuka pengguna grafis GUI
9	Menggunakan MySQL Workbench sebagai database
10	Tampilan sistem *user-friendly*

Tabel 3 Spesifikasi Kebutuhan Fase 2

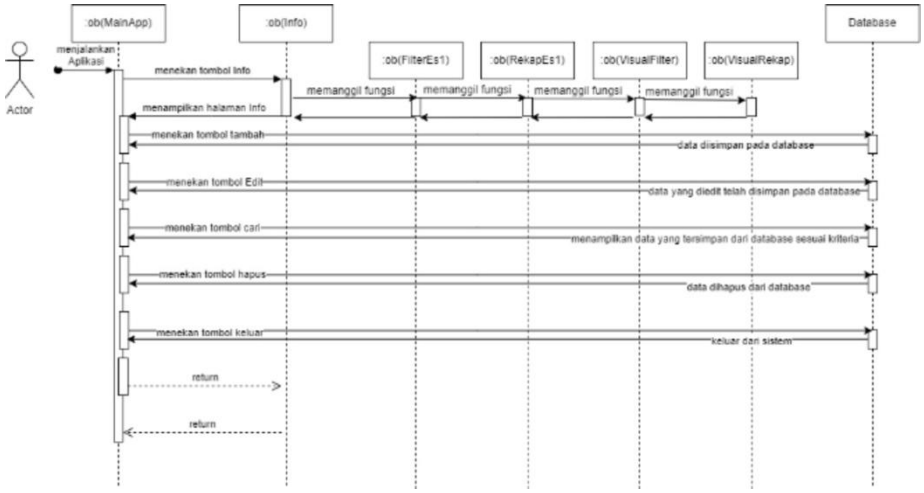
Berdasarkan Tabel di atas, berikut spesifikasi kebutuhan sistem yang dimodelkan dalam desain UML (Unifed Modeling Language) Fase 2.



Gambar 10 *Use Case* Fase 2

Pada *use case diagram* terdapat operator sebagai aktor yang melakukan kegiatan view sebagaimana pada gambar.

Berdasarkan use case, berikut merupakan interaksi antar entitas dalam scenario mengakses halaman info untuk menampilkan visualisasi jumlah jenis temuan dan jenis kerugian, tabel jumlah jenis temuan, bar diagram jenis temuan, dan *multi level pie chart.*



Gambar 11 *Sequence Diagram* Fase 2

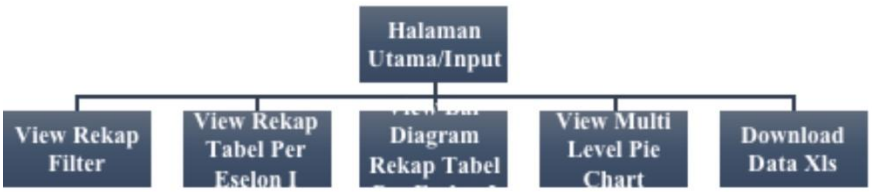
Sedangkan aktivitas pada sistem yaitu terdapat aktivitas-aktivitas sebagaimana diagram berikut:

Gambar 12 *Activity Diagram* Fase 2

Desain dan Implementasi

1) Desain

Berikut merupakan desain navigasi Fase 2 yang menampilkan beberapa navigasi program.



Gambar 13 Desain Navigasi Fase 2

Implementasi Pengkodean

Pada pengkodean Fase Kedua berupa class-class yang berisi metode untuk menampilkan visualisasi sebagai berikut:

1.Info

Pada kelas ini terdapat beberapa hal yaitu:

a. Button "Excel" dan "Keluar" memberikan fungsionalitas tambahan. Button "Excel" yang memanggil fungsi Xls yang mana dilakukan proses pengambilan data dari database dan menyimpannya dalam file Excel. Kemudian button "Keluar" untuk menutup aplikasi Tkinter.

a. Terdapat struktur proyek dengan kode terorganisir dalam kelas Info, yang membentuk struktur proyek dengan penggunaan beberapa file terpisah seperti filteres1.py, rekapes1.py, visualrekap.py, dan visualfilter.py.

Adapun masing-masing dipanggil dan ditampilkan pada dataframe berikut:

```
self.filter_frame = Filteres1(DataframeLeft1)
self.filter_frame = Visualfilter(DataframeLeft2)
self.filter_frame = Rekapes1(DataframeRight1)
self.filter_frame = Visualrekap(DataframeRight2)
```

2. Filteres1

Class berfungsi untuk menampilkan visual dan menghitung total kasus dan menampilkannya di sebuah label dengan metode:

a. populate_table yang berfungsi mengambil data es1 dari tabel temuan untuk diisi ke dalam combobox

b. filter_data yang berfungsi mengambil data dari database berdasarkan filter es1 yang dipilih.

c. update_special_counts yang berfungsi menghitung hitungan khusus untuk "TP", "TGR", dan "PIII" dan menampilkan hitungan khusus di bagian bawah kolom jenis_kerugian

Algoritm *quick sort* yang diimplementasikan pada kelas ini bertindak sebagai komponen dalam pengelolaan dan penyajian data pada aplikasi. Algoritma pengurutan yang digunakan dalam fungsi ini beroperasi dengan memilih elemen sebagai pivot, yang kemudian menjadi landasan untuk pembagian data menjadi tiga kelompok utama, yaitu elemen yang lebih kecil dari pivot, elemen yang sama dengan pivot, dan elemen lebih besar dari pivot. Fungsi *quick sort* pada kelas ini diaplikasikan pada fase sebelum menampilkan data tabel *treeview* melalui fungsi *populate_table.* Implementasi ini

memastikan bahwa data yang dtampilkan telah diurutkan sesuai dengan aturan algoritma *quick sort*

```
def quick_sort(self, data):
        if len(data) <= 1:
            return data

        pivot = data[len(data) // 2]
        left = [x for x in data if x < pivot]
        middle = [x for x in data if x == pivot]
        right = [x for x in data if x > pivot]

       return self.quick_sort(left) + middle + self.quick_sort(right)
```

3. Rekapes1

   Class ini berfungsi untuk menampilkan tabel dengan Pustaka tk.Treeview dari pemanggilan metode:

   a. Metode get_jenis_temuan_count yang mengambil data dari database untuk menghitung jumlah kasus berdasarkan es1 dan jenis temuan. Digunakan untuk mengisi nilai kolom "BPK (Eksternal)", "Itjen (Internal)", dan menghitung total kasus.
   b. Metode sort_column yang mengurutkan data pada kolom tertentu saat heading kolom diklik.
   c. Mengisi Data ke Treeview dengan mengambil data es1 dari tabel temuan, melakukan iterasi melalui nilai es1 dan menghitung jumlah kasus berdasarkan jenis temuan, dan menambahkan total pada bagian bawah ttk.Treeview.

Implementasi algoritma *quick sort* pada kelas Rekapes1 adalah pada proses pengurutan data sebelum tersaji pada tabel Rekapitulasi Data Per Eselon 1. Setiap kolom dari tabel, yaitu es1, BPK (Eksternal), Itjen (Internal), dan Total Kasus, dapat diurutkan dengan lebih efisien oleh user melalui fungsi klik pada *header* pada kolom terkait. Adapun sorting dilakukan berdasarkan analisa pada data temuan dan kasus kerugian per Eselon I.

```
def quick_sort(self, items, col):
        if len(items) <= 1:
            return items
        else:
            pivot = self.tree.set(items[0], col)
            lesser = [item for item in items[1:]
               if self.tree.set(item, col) < pivot]
            greater = [item for item in items[1:]
                if self.tree.set(item, col) >= pivot]
        return self.quick_sort(lesser, col) + [items[0]] + self.quick_sort(greater, col)
def sort_column(self, col):
        items = self.tree.get_children('')
        sorted_items = self.quick_sort(items, col)
        self.refresh_treeview(sorted_items)
```

4. **Visualfilter**

Pada class ini terdapat kode untuk menampilkan diagram multi-level pie chart interaktif *(sunburst chart)* dengan kemampuan filter berdasarkan kategori "es1" dari data yang diambil dari database MySQL. Diagram dibuat menggunakan matplotlib dan menambahkan label pada sektor-sektor tertentu dengan warna untuk membedakan sub kategori. Selain itu juga digunakan Event Binding dan Konfigurasi untuk menangani perubahan ukuran canvas dan mengkonfigurasi ulang scroll region.

Pada class ini juga diimplementasikan algoritma quick sort yang memiliki peran penting dalam menyajikan data secara terstruktur pada visualisasi sunburst chart. Sehingga pada visualisasi ini dimungkinkan untuk pengguna melakukan aktivitas memilih dan sistem akan melakukan filterisasi pada data temuan berdasarkan Eselon 1 dengan bantuan tombol Combobox yang disediakan. Setiap perubaan pada pemilihan Eselon 1 akan mengaktifkan fungsi filter yang akan mengimplementasikan algoritma *quick sort* untuk mengurutkan dan memperbaharui visualisasi *sunburst chart.*

```
def quick_sort(self, data, col):
        if len(data) <= 1:
            return data
        else:
            pivot = data[col].iloc[0]
            lesser = data[data[col] < pivot]
            equal = data[data[col] == pivot]
            greater = data[data[col] > pivot]
            return pd.concat([self.quick_sort(lesser, col), equal, self.quick_sort(greater, col)],
 ignore_index=True)
```

5. Visualrekap

Pada class ini terdapat kode untuk menampilkan diagram batang (bar chart) yang memanggil Metode show_bar_chart untuk mengambil data dari database (es1_values dan total_kasus_values) dan membuat diagram batang dengan menggunakan matplotlib. Kemudian memanggil Metode get_jenis_temuan_count untuk mengambil jumlah kasus berdasarkan es1 dan jenis temuan dari database.

Agoritma *quick sort* berkontribusi signifikan pada kelas ini dalam menyajikan informasi rekapitulasi data temuan per Eselon 1 dengan visualisasi berupa diagram batang (*bar chart*). Setiap nilai pada Eselon 1 dan total kasus dirutkan secara efisien untuk ditampilkan pada visualisasi dalam diagram.

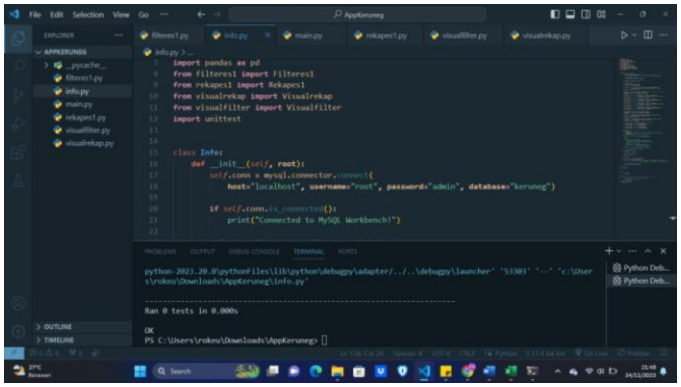
Proses *quick sort* dimulai dengan pemilihan elemen pivot dari nilai Eselon 1. Setelah itu, data dibagi menjadi tiga kelompok: nilai Eselon 1 yang lebih kecil dari pivot, nilai yang sama dengan pivot, dan nilai yang lebih besar dari pivot. Proses ini berulang secara rekursif pada setiap kelompok hingga seluruh data terurut.

```
def quick_sort(self, es1_values, total_kasus_values):
        if len(es1_values) <= 1:
            return es1_values, total_kasus_values
        else:
            pivot = es1_values[0]
            lesser_indices = [i for i, x in enumerate(es1_values) if x < pivot]
            equal_indices = [i for i, x in enumerate(es1_values) if x == pivot]
            greater_indices = [
                i for i, x in enumerate(es1_values) if x > pivot]
               lesser_es1, lesser_total_kasus = self.quick_sort( [es1_values[i] for i in lesser_indices],
[total_kasus_values[i] for i in lesser_indices])
 equal_es1, equal_total_kasus = self.quick_sort( [es1_values[i] for i in equal_indices], [total_kasus_values[i]
for i in equal_indices]) greater_es1, greater_total_kasus = self.quick_sort( [es1_values[i] for i in
greater_indices], [total_kasus_values[i] for i in greater_indices]) return lesser_es1 + equal_es1 +
greater_es1, lesser_total_kasus + equal_total_kasus + greater_total_kasus
```

Pengujian dengan White-Box Testing

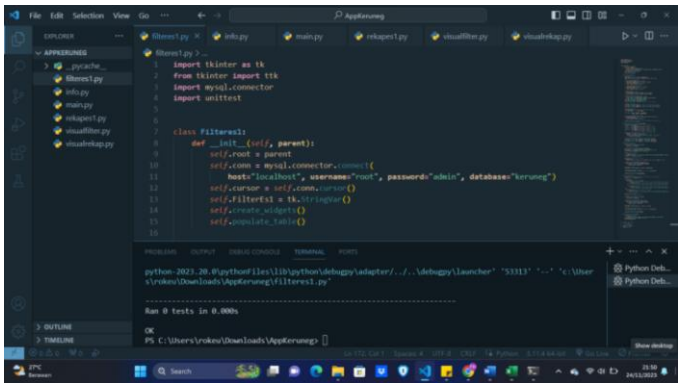
Berikut merupakan hasil dari pengujian *white-box testing unittest* pada Fase Kedua:

a) Class Info



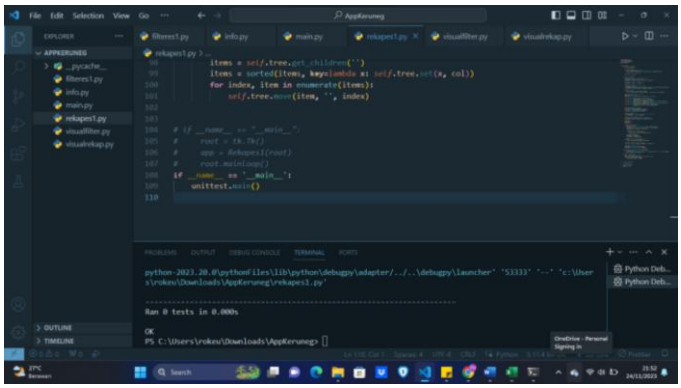
Gambar 14 *White Box Testing* Kelas info

b) Class Filteres1



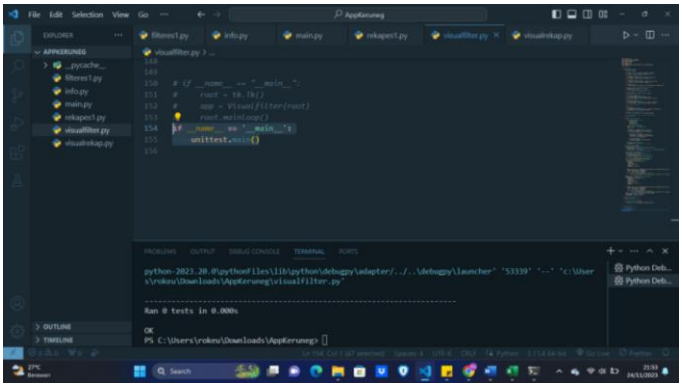
Gambar 15 *White Box Testing* Kelas Filteres1

c) Class Rekapes1



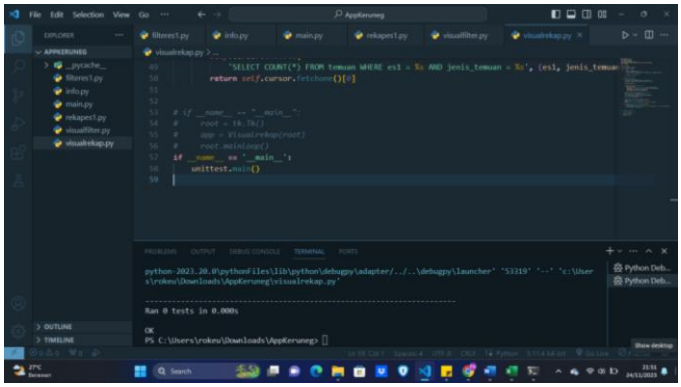
Gambar 16 white box testing kelas Rekapes1

d) Class Visualfilter



Gambar 17 *White Box Testing* Kelas Visualfilter

e) Class Visualrekap

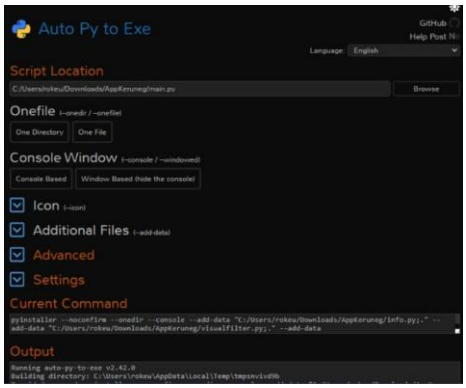


Gambar 18 *White Box Testing* Kelas Visualrekap

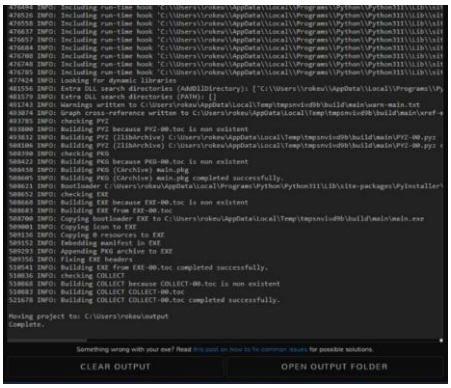
Hasil pengujian terhadap sistem dengan *Metode White-box Testing* dengan menggunakan *Unittest* Python pada 6 *class* sebagaimana pada gambar di atas menunjukkan menunjukkan value "OK" yang berarti fungsi dan logika berperilaku sesuai yang diharapkan untuk skenario uji.

Deployment

Pada tahap ini dilakukan pemindahan atau pengimplementasian source code ke lingkungan produksi dengan utilitas auto-py-to-exe yang berfungsi mengonversi skrip Python menjadi file yang dapat dijalankan tanpa menginstal Python di sistem pengguna. Langkah-langkahnya yaitu memilih skrip Python, menyesuaikan opsi dengan file atau direktori tambahan, lalu menekan CONVERT .PY TO .EXE" untuk memulai proses build. Setelah tools menunjukkan aktivitas complete, maka skrip Python sudah menjadi file eksekutif (.exe).



Gambar 19 Tampilan Konfigurasi *Auto-Py-To-Exe*



Gambar 20 Tampilan Complete Konversi pada Auto-Py-To-Ex

Gambar 21 AppKeruneg.exe

Pembahasan

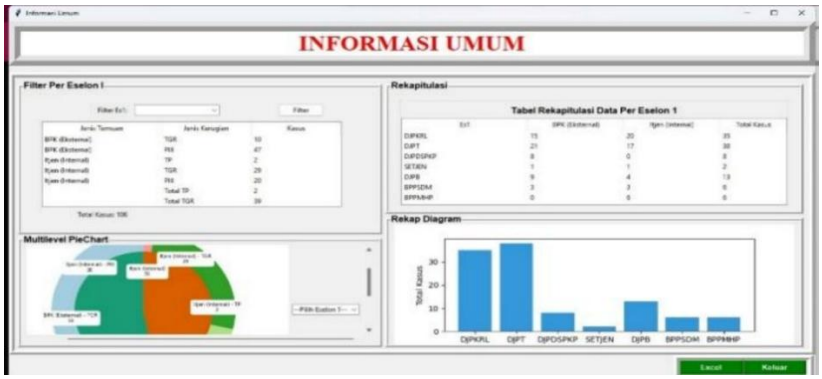
Source code aplikasi Sistem Manajemen Piutang Jangka Panjang Lainnya telah di uji menggunakan pengujian *White Box Testing (Unittest)* dan berhasil dilakukan *deployment* pada sistem operasi Windows. Sistem juga telah berhasil dilakukan uji coba pada 108 kasus dari temuan baik bersumber dari pemeriksa eksternal maupun pengawas internal yang berstatus "**informasi**".

a) Tampilan Fase 1

Gambar 22 Tampilan Fase 1 Berupa CRUD dan Treeview

Berikut merupakan Tampilan dari Fase 1 yang berisi data informasi kasus serta memiliki beberapa fitur seperti Tambah, Edit, Hapus, Cari, Clear, Info, dan Keluar. Adapun pada saat tombol Info ditekan, maka fungsi akan memanggil fungsi dari Fase Kedua sebagaimana Tampilan Fase 2 berikut.

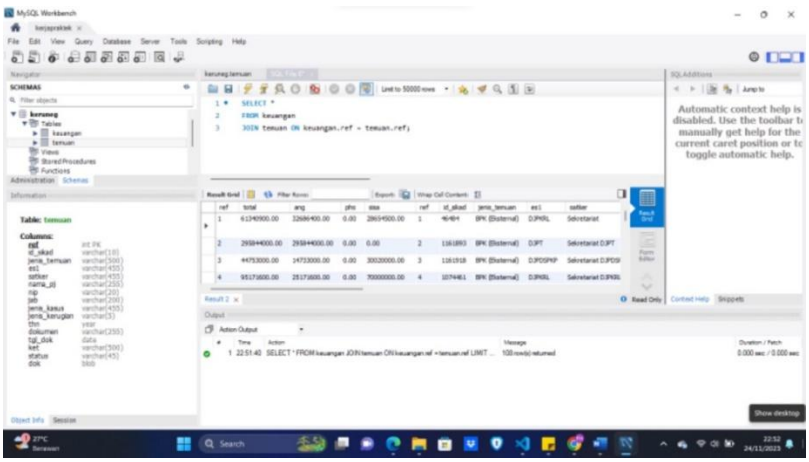
b) Tampilan Fase 2



Gambar 23 Tampilan Fase 2 berupa Visualisasi Informasi Umum

c) Hasil Implementasi Tambah 108 Kasus Informasi Kerugian Negara melalui GUI pada database MySQL Workbench

Pada database keruneg terdapat 2 tabel yang saling berinteraksi menunjukkan terdapat 108 rows dari input Aplikasi Sistem Manajemen.



Gambar 23 Implementasi Tambah Informasi Kerugian pada MySQL Workbench

Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, Peneliti dapat menyimpulkan bahwa perancangan Sistem Manajemen Piutang Jangka Panjang Lainnya dengan Pemrograman Berorientasi Objek berbasis Python dan MySQL Workbench ini telah berhasil memberikan inovasi berupa:

1) aplikasi dengan tampilan ringan dan *user-friendly*
2) mengakomodasi informasi Status kasus sehingga dapat mengintegrasikan Matriks Kerugian Negara dan Matriks Informasi Kerugian Negara ke dalam suatu sistem yang terintegrasi
3) menampilkan visualisasi yang sebelumnya dilakukan secara manual di dalam Microsoft Excel
4) dapat melakukan ekspor data dalam bentuk ekstensi .xls, sehingga dapat menyajikan data aktual dengan cepat

Hal ini merupakan bentuk solusi peningkatan efektivitas dan efisiensi waktu serta tenaga dari SDM ketika data dibutuhkan dalam waktu cepat, visualisasi untuk bahan presentasi, serta monitoring yang terintegrasi.

Referensi

[1] S. Beck dan Whistler, “Innovative Organizations: A Selective View of Current Research”, Journal of Business, vol. 40, no. 3, pp. 462–469, 1967.

[2] A. dan Trollip, “Computer Based Instruction: Method and Development,” New Jersey: Prenctice-Hall inc, 2001.

[3] Jogiyanto, “Analisis & Desain sistem informasi: Pendekatan terstruktur Teori dan Praktik,” Yogyakarta: Andi, 2005.

[4] Andri, “Algoritma & pemrograman dengan C++ edisi 2,” Yogyakarta: Graha, 2009.

[5] Presman, “Software Engineering: A Practitioner’s Approach,” New York: Mcgraw-Hill, 2010.

[6] Hermawan, “Menguasai Java 2 dan Object-Oriented Programming,” Yogyakarta: Andi, 2004.

[7] Retnoningsih E, “Pembelajaran Pemrograman Berorientasi Objek (Object Oriented Programming) Berbasis Project Based Learning,” Informatics For Educators and Professionals. 2 (1): 95 – 104, 2017.

[8] Sukamto dan Shalahuddin, “Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek,” Bandung: Informatika Bandung, 2018.

[9] E.D. Wahyuni, “Implementasi Metode Incremental Pada Sistem Informasi Administrasi Desa Jambuwer,” Jurnal Tekno Kompak, pp 158-160, 2020.

[10] I. Kalb, “Object-Oriented Python: Master OOP by Building Games and GUIs,” San Francisco: No Starch Press, 2022

[11] K. Rokoyah, “Penerapan Model Incremental dalam Merancang Aplikasi Pengenalan Bentuk dan Fungsi Gigi Pada Manusia Berbasis Web,” Jurnal Ilmiah Sikomtek, 2022.

[12] D. Love, “Tkinter GUI Programming by Example,” Birmingham: Packt Publishing Ltd, 2018.

[13] A. D. Moore, “Python GUI Programming with Tkinter,” Birmingham*:* Packt Publishing Ltd, 2018

[14] W. Nugraha, “Metode Incremental Dalam Membangun Aplikasi Identifikasi Gaya Belajar Untuk Meningkatkan Hasil Belajar Siswa,” Jurnal Sistem Komputer Musirawas, 45, 2019.

[15] M. Roseman, “Modern Tkinter for Busy Python Developers (3rd Edition),” E-book: Late Afternoon Press, 2020.

Mohamad Yusuf, S.Kom., M.C.S.

Dosen Fasilkom Program Studi Teknik Informatika Universitas Mercu Buana, Jakarta.

Faaza Naima

Mahasiswa Fasilkom Program Studi Teknik Informatika Universitas Mercu Buana, Jakarta.