

Perancangan Sistem *Inventory* Berbasis *Web* Menggunakan *Framework Flask* : PT. Gagas Mitra Jaya (Area Salatiga)

Oktaviano Miftahqul Ma'arif ^{1*}, Tatit Kurniasih ²

^{1*,2}Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

Corresponding Email: 672020091@student.uksw.edu ^{1*}

Histori Artikel:

Dikirim 17 April 2024; *Diterima dalam bentuk revisi* 23 April 2024; *Diterima* 5 Mei 2024; *Diterbitkan* 20 Mei 2024. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STMIK Indonesia Banda Aceh.

Abstrak

Penelitian dilakukan untuk membangun sistem inventory berbasis web di PT. Gagas Mitra Jaya area Salatiga sebagai solusi bagi tantangan dalam proses pencatatan material yang dimiliki oleh setiap teknisi. Metode penelitian yang digunakan adalah hybrid waterfall-agile. Perancangan sistem menggunakan use case diagram dan activity diagram dalam UML untuk menggambarkan alur pembuatan sistem, yang melibatkan proses pengumpulan data melalui wawancara dan kuesioner. Proses pembangunan menggunakan bahasa Python, Flask, VS Code IDE, dan PostgreSQL. Pemilihan ini berdasarkan pertimbangan fleksibilitas, aksesibilitas, kemudahan integrasi, dan kemampuan pembaruan yang lebih cepat. Arsitektur kode yang digunakan adalah MVC (Model-View-Controller), sebuah konsep pengembangan yang membagi struktur data, logika sistem, dan interface pada website. Sistem yang dirancang bernama GNS (Gagas Network Supplies) memiliki menu utama yang mencakup pengambilan, pemakaian, dan pengembalian material untuk mencatat penggunaan material, serta menu total barang merangkum stok material. Dengan sistem GNS diharapkan memberikan proses pelayanan yang lebih baik, pemantauan stok lebih efektif, akurasi inventaris yang lebih tinggi, dan fleksibilitas dalam pertumbuhan bisnis.

Kata Kunci: Flask; Hybrid Waterfall-Agile; MVC; Postgresql; Python.

Abstract

Research was conducted to build web-based inventory system at PT. Gagas Mitra Jaya in Salatiga as solution to the challenges in process of recording materials owned by technician. The research method used hybrid waterfall-agile. System design uses use case and activity diagrams in UML to describe flow of system creation, which involves the process of collecting data through interviews and questionnaires. The development uses Python, Flask, VS Code IDE, and PostgreSQL. This selection based on considerations of flexibility, accessibility, ease of integration, and faster update capabilities. The code architecture is MVC (Model-View-Controller), a development concept that divides data structures, system logic and interfaces on websites. The system designed called GNS (Gagas Network Supplies) has main menu that includes picking up, using and returning materials to record material usage, as well total item menu that summarizes material stock. The GNS system is expected to provide better service processes, more effective stock monitoring, higher inventory accuracy, and flexibility in business growth.

Keyword: Flask; Hybrid Waterfall-Agile; MVC; Postgresql; Python.

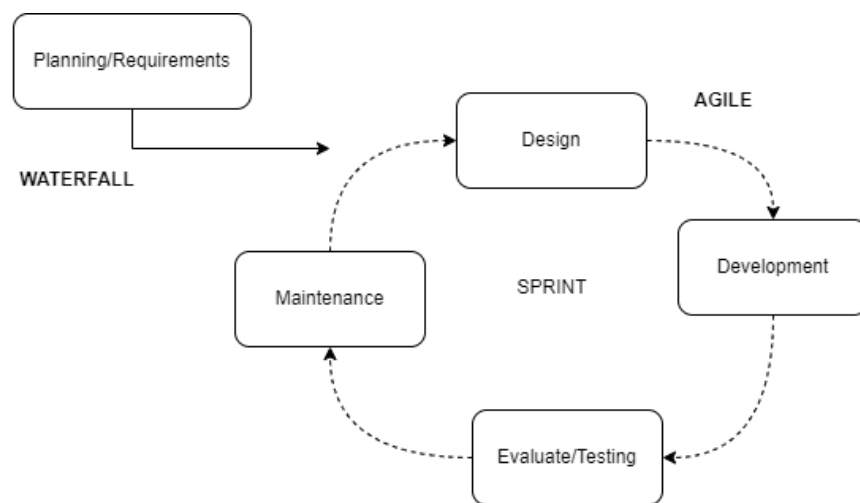
1. Pendahuluan

Di era globalisasi dan kemajuan teknologi informasi, sektor telekomunikasi menjadi kunci untuk konektivitas dan pertumbuhan ekonomi. Terutama bagi perusahaan yang berfokus pada pembangunan infrastruktur jaringan (Bagaskara, 2023). Seiring dengan kebutuhan, menjadi sangat penting bagi perusahaan untuk mengadopsi teknologi informasi guna meningkatkan efisiensi operasional. Sistem manajemen yang terstruktur diperlukan untuk mengatur dan mengelola data guna mencapai hasil maksimal. Dengan sinergi yang terencana, sistem ini berfungsi sebagai kesatuan terorganisir yang memiliki informasi dengan tujuan spesifik (Pratiwi *et al.*, 2024). Sistem memberikan informasi berupa kumpulan data, prosedur, dan sumber daya yang saling terintegrasi dengan penerapan sistem inventori untuk peningkatan operasional (Kesuma *et al.*, 2022). Sistem inventori mengelola data persediaan barang untuk mendukung pengambilan keputusan strategis (Handayani *et al.*, 2023). PT. Gagas Mitra Jaya (GMJ) adalah perusahaan konstruksi jaringan komunikasi yang dimiliki oleh Koperasi Pegawai (KOPEGTEL) Yogyakarta. Untuk konstruksi jaringan dimulai dengan pemesanan oleh mitra kerjasama, yang kemudian diproses melalui aplikasi bernama MyGMJ. Melalui data disistem, membantu supervisor mendapatkan informasi untuk membuat keputusan dalam menentukan material yang harus diambil oleh teknisi. Walaupun begitu, terdapat masalah dalam pengelolaan material saat sudah berada di tangan teknisi selama operasional, karena masih mengandalkan pencatatan manual yang kurang efisien. Selain itu, jika terdapat kendala dilapangan yang memerlukan pertukaran material juga menyulitkan untuk monitoring sumber daya. Sehubungan dengan masalah tersebut, maka diperlukan sebuah sistem inventori yang dirancang dengan tujuan mempermudah proses pencatatan dan pengkoordinasian penggunaan material oleh teknisi. Sistem tersebut akan dirancang dalam bentuk *website* dengan memanfaatkan *Flask*, *Python*, dan *database PostgreSQL* (Saifudin & Kautsar, 2024). Pemilihan ini berdasarkan pada pertimbangan yaitu kemampuan adaptasi sistem terhadap lingkungan yang kompleks dengan kemampuan proses integrasi terhadap sistem lain yang mudah, proses pembaruan yang dapat dilakukan dengan cepat, kemampuan pengolahan data yang besar dan penyediaan fungsi maupun *library* yang dapat mempermudah pengembangan sistem (Alifiansyah Saleh *et al.*, 2023). Kebutuhan akan data yang besar dan keamanan data yang tinggi sangat diperlukan pada sebuah sistem aplikasi. Dengan memanfaatkan sistem basis data yang terdistribusi seperti pemakaian data bersama (*share*), kehandalan (*reliability*), ketersediaan (*availability*), dan kecepatan pemrosesan query yang mampu mengolah data secara akurat sangat diperlukan dalam proses pengolahan data. Memastikan data secara akurat dan diolah secara efisien. Website yang dibangun bernama *Gagas Network Supplies* (Muliawan, 2023).

Beberapa penelitian yang telah dilakukan mengenai penggunaan Flask untuk mengembangkan sistem inventori seperti penelitian oleh Andre Josiah Budianto yang membahas pengembangan modul inventori web untuk mengontrol stok dan melakukan pembaruan terhadap sistem *Oracle Form* (Budianto & Saian, 2023). Penelitian oleh Alpha Adarrani Wibowo yang membahas penggunaan *database Flask*, *PostgreSQL*, dan *Pycharm* untuk aplikasi pengecekan stok pada *distribution center* dan toko (Wibowo Putri & Susetyo, 2022). Penelitian oleh Hana Louise Walingkas pada implementasi Flask untuk aplikasi inventori yang berfokus pada supplier dan penelitian oleh Eben Haezer Yena Kristianto yang berfokus pada aplikasi *Virtual Inventory System* (VIS) berbasis *web* (Walingkas & Saian, 2023) (Eben Haezer & Setiyawati, 2022). Perbedaan pada penelitian ini adalah metode yang dipakai dengan memisahkan file dalam kelompok berdasarkan fungsi dengan MVC serta penggunaan *library pandas* untuk mengolah *metadata* dari database dibuat menjadi struktur yang sesuai untuk format *xlsx* (*excel*). Serta tahapan penelitian yang menggunakan metode hybrid untuk mempercepat waktu pembuatan website. Penelitian ini menggunakan *Visual Studio Code IDE* yang lebih fleksibel. Dengan dibangunnya website ini, diharapkan dapat mempermudah pengelolaan material yang digunakan teknisi sehingga mengurangi adanya kebutuhan biaya yang lebih besar ketika diperlukan tambahan material.

2. Metode Penelitian

Tahapan penelitian yang akan dilakukan menggunakan model *hybrid waterfall-agile*. Metode hybrid bertujuan untuk memperoleh keunggulan yang dimiliki oleh kedua metode tersebut. Metode *waterfall* merupakan metode yang sering digunakan karena sesuai untuk proyek kecil dengan model tahapan yang sederhana dan mudah dimengerti, serta setiap prosesnya dapat didefinisikan dengan terstruktur. Sedangkan untuk metode *agile* memiliki proses yang lebih cepat karena proses pengembangan sistem dapat diuraikan menjadi bagian-bagian kecil yang kemudian dapat dikembangkan secara bertahap (Ramadhan *et al.*, 2023) (Marques *et al.*, 2023).



Gambar 1. Model Proses Penelitian

Adapun tahapan-tahapan penelitian metode hybrid yang diterapkan pada perancangan seperti yang ditunjukkan pada Gambar 1 adalah sebagai berikut :

1) *Requirements/Planning*

Tahap awal melibatkan identifikasi masalah dan kebutuhan yang diperlukan melalui interaksi dengan pengguna. Untuk pengumpulan data dilakukan melalui wawancara dan diskusi. Seluruh data yang didapatkan akan digunakan sebagai bahan perancangan sistem sehingga proses kerja dari sistem akan sesuai dengan apa yang dibutuhkan. Kebutuhan seperti : data material konstruksi, data karyawan, gambaran alur dari sistem dan kebutuhan perangkat seperti *Python*, *Visual Studio Code IDE*, *Framework Flask*, *Database PostgreSQL*, *OS Windows 10*, dan *Library Pandas*..

2) *Design*

UML (*Unified Modeling Language*) merupakan sekumpulan diagram yang digunakan untuk mendefinisikan, menganalisis, dan mendesain arsitektur program yang berorientasi objek secara rinci dan efisien (Rambe *et al.*, 2020). Perancangan sistem akan menggunakan *Use Case Diagram* dan *Activity Diagram*. Pemilihan diagram ini karena *Use Case* dapat menjelaskan interaksi antara sistem dan entitas eksternal atau aktor serta memaparkan fungsionalitas sistem dari sudut pandang pengguna. Sedangkan *Activity* mampu menggambarkan alur kerja atau proses bisnis dalam sistem, menangkap urutan aktivitas dan keputusan yang diambil selama proses. Hasil dari tahap ini diagram UML dan relasi antara tabel yang dibuat pada database Postgres (Suriya & S., 2023).

3) *Development*

Setelah tahap perancangan dilakukan, langkah selanjutnya adalah pengembangan terhadap alur kerja yang telah dibuat untuk dijadikan sebuah program. Pada bagian *frontend* menggunakan HTML dan CSS bersama dengan JavaScript untuk meningkatkan interaktif dari antarmuka. Untuk penggunaan fungsi, route, dan template menggunakan komponen atau *library* yang tersedia

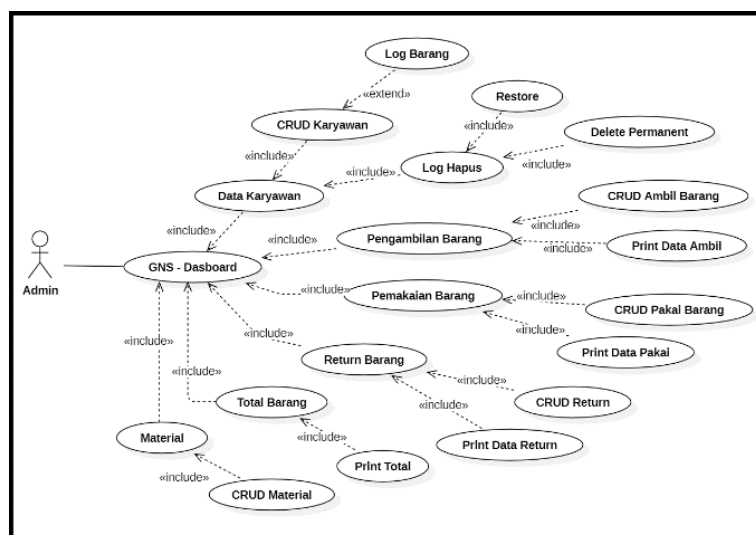
dalam bahasa pemrograman Python dan framework Flask. Proses pengembangan akan dilakukan menggunakan Visual Studio Code IDE dengan konfigurasi yang menghubungkan pada *database Postgres*.

4) *Testing*

Langkah berikutnya adalah pengujian untuk menilai sejauh mana sistem tersebut memenuhi persyaratan. Pengujian dilakukan melalui dua metode, yaitu metode blackbox dan kuesioner. Pengujian blackbox digunakan untuk mengevaluasi fungsionalitas dari sistem, memastikan bahwa sistem berjalan sesuai dengan kebutuhan pengguna. Pada proses pengujian melibatkan analisis input dan output dari sistem. Selain itu, pengujian kuesioner juga dilakukan untuk mengevaluasi pengalaman dari pengguna atau persepsi mereka terhadap sistem. (Fahrezi *et al.*, 2022) (Uminingsih *et al.*, 2022).

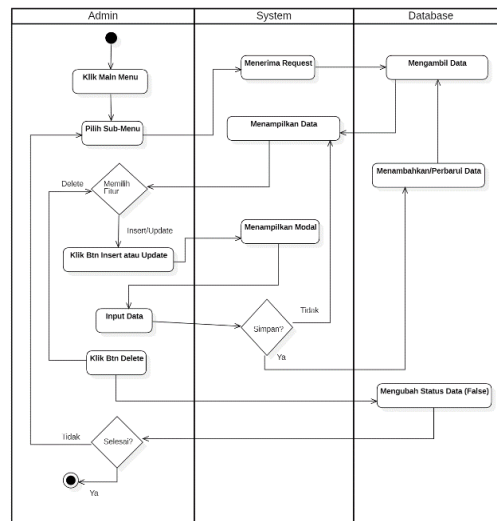
5) *Maintenance*

Setelah sistem berhasil melewati pengujian dengan baik, langkah selanjutnya melakukan pemeliharaan yang melibatkan pengecekan dan perbaikan terhadap keseluruhan sistem secara berkala sesuai dengan keadaan. Tujuannya adalah untuk menjaga kondisi sistem supaya tetap optimal. Proses pemeliharaan akan dilakukan dengan melakukan perawatan secara rutin seperti memperbarui konten, memperbaiki logika sistem, dan tautan yang bermasalah. Lalu melakukan pembaruan teknologi untuk menjaga keamanan data serta kinerja dari sistem.

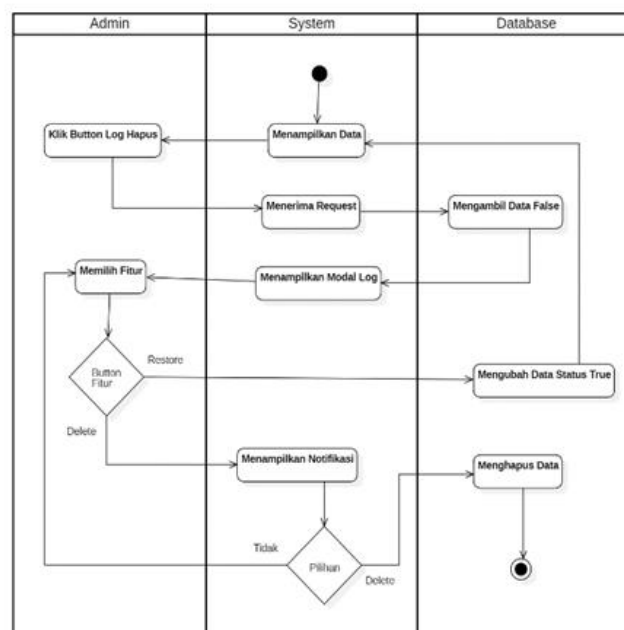


Gambar 2. Use Case Diagram

Gambar 2 menampilkan ketika admin, yang berperan sebagai pengguna utama dalam sistem, telah melakukan Login. Setelah login, maka akan ditampilkan *dashboard* sebagai halaman utama. Terdapat enam menu yang dapat diakses yaitu menu Pengambilan barang, Pemakaian Barang, *return* barang, total barang, Data Karyawan, dan Data Material. Pada menu Data Karyawan, *user* dapat melakukan pengolahan data karyawan, menggunakan fungsi Log Barang dan fungsi Log Hapus. Menu pengambilan barang, pemakaian barang, dan return barang terdapat fungsi pengolahan data serta fungsi print untuk menu total barang hanya terdapat fungsi print.

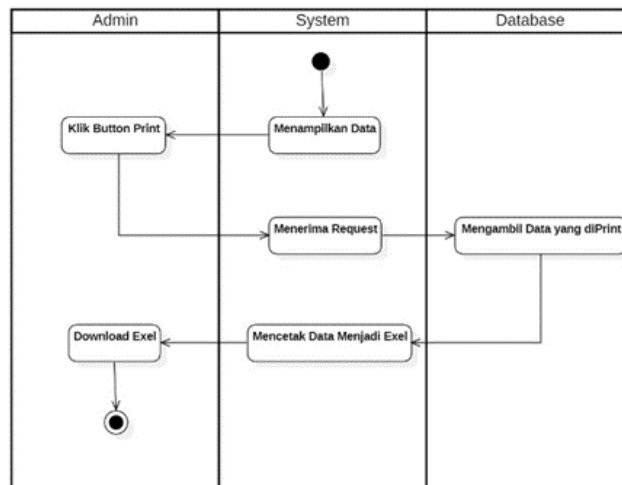


Gambar 3. Activity Diagram Create, Update, Delete



Gambar 4. Activity Diagram Log Hapus

Pada gambar 3, ketika di menu pengguna diminta untuk memilih sub-menu, ketika sub-menu tersebut dipilih, sistem akan menerima *request* dan mengirimkan ke database untuk mengambil data. Database akan mengirimkan data yang memiliki status true atau aktif sesuai dengan menu yang dipilih, dan menampilkan data tersebut pada sistem. Setelah data ditampilkan, maka pengguna dapat mengolah data yang ada dengan fitur yang tersedia. Untuk insert dan update akan ditampilkan modal untuk mempermudah pengguna. Sedangkan untuk fitur delete akan mengubah data menjadi berstatus false. Pada gambar 4 menggambarkan Log hapus pada menu data karyawan. Pengguna akan memilih fitur Log Hapus, dan akan menampilkan modal yang berisi tabel data yang berstatus false. Pada modal terdapat fitur tambahan yaitu *restore* dan *delete* (permanen).



Gambar 5. Activity Diagram Cetak File Xlsx

Pada diagram gambar 5 merupakan gambaran fitur cetak yang terdapat dalam menu pengambilan, pemakaian, dan return barang. Ketika sub-menu ditampilkan, maka terdapat salah satu fitur yaitu print untuk mencetak data yang ada di menu tersebut. Ketika print dipilih, sistem akan menerima request dan mengirimkannya ke database untuk mengambil data sesuai dengan parameter yang dikirimkan oleh sistem. Setelah itu, sistem menghasilkan response berupa file dengan format Xlsx.

3. Hasil dan Pembahasan

Arsitektur pembuatan program adalah MVC (*Model-View-Controller*) yang merupakan sebuah konsep pengembangan website dengan membagi struktur data, logika sistem, dan interface yang akan ditampilkan (Ardiansyah & Munawaroh, 2023). *Model* adalah bagian yang mengelola database atau sering disebut sebagai DAO yang bertugas untuk mengenkapsulasi data-data serta pemrosesannya. *View* merupakan bagian interface yang biasanya berisi file HTML pada folder template. Sedangkan *Controller* adalah bagian yang menampung semua rute antara user dengan tampilan aplikasi ataupun dengan database. Konsep ini lebih mudah dilakukan dalam melakukan pengembangan sistem karena struktur yang tersusun berdasarkan fungsi dan kegunaannya (Rahmatika, Ardhi Dinullah Baihaqie, 2022).

Kode Program 1. File `_init_.py` berisi konfigurasi app

```

# Flask app
app = Flask(__name__)

# PostgreSQL database configuration
PG_USER = "postgres"
PG_PW = "vdummypw"
PG_DB = "dbdummy"
PG_HOST = "localhost"
PG_PORT = "5432"

# Konektor
def dbConnectPG():
    db_uri = f"postgresql://{PG_USER}:{PG_PW}@{PG_HOST}:{PG_PORT}/{PG_DB}"
    connection = psycopg2.connect(db_uri)
    return connection

# import controller / route

```

```
from website.controller import
(r_control,controllerproject1,controllerPengambilan,controllerPemakaian,contro
llerMaterial,controllerReturn,controllerTotalBrg)
```

Pada kode program 1 baris `app = Flask(__name__)` membuat objek aplikasi Flask bernama `app`. Variabel `__name__` adalah variabel Python yang mewakili nama modul saat ini. Saat Flask diinisialisasi, ia menggunakan informasi ini untuk menemukan sumber daya seperti *template* dan file statis. Bagian kedua fungsi `dbconnectPG()` untuk konektor ke database berdasarkan konfigurasi. Selain itu mengimpor berbagai control atau rute yang menangani fungsionalitas seperti *request*, memproses data, dan menghasilkan respons. Hal ini memudahkan melakukan kontrol terhadap seluruh aplikasi (Santoso & Saian, 2023).

Kode program 2. *Flask Controller*

```
@app.route("/DataKaryawan", methods=['GET', 'POST'])
def getdatakaryawan():
    dataKaryawan = getdata()
    datalog = loghapusdata()
    return render_template("VwKaryawan.html", title='Data Karyawan',
dataKaryawan =dataKaryawan, datalog=datalog)

@app.route("/DataKaryawan/DataLogKaryawan", methods=['GET', 'POST'])
def getdataLog():
    datalog = loghapusdata()
    datalog = datalog['result']
    return jsonify(datalog)

@app.route("/DataKaryawan/LogTotalBrg", methods=['GET', 'POST'])
def getDataLOGTtlBrg():
    ID_Karyawan = request.form.get('ID_Teknisi')
    dataLogBrg = QLogDataBarang (ID_Karyawan)
    dataLogBrg = dataLogBrg['result']
    return jsonify(dataLogBrg))
```

Kode program 2 file `controllerproject` untuk memetakan alamat URL dari aplikasi melalui decorator `@app.route` serta fungsi yang akan dijalankan ketika alamat URL diakses. Pada rute yang pertama yaitu `/DataKaryawan` akan dijalankan pertama, fungsi ini mengambil hasil dari `getdata()` yang berasal dari file `daodatakaryawan` dan mengirimkan ke `template`. Bagian kedua dengan alamat URL `/DataKaryawan/DataLogKaryawan` bertujuan untuk mengembalikan hasil response dari file `Dao` pada fungsi `loghapusdata()` yang diakses ketika modal Log Hapus dijalankan.

Kode program 3. Fungsi untuk membuat data menjadi file *xlsx*

```
def cetakLapPakaiBrg():
    #Bagian Pertama
    Daerah_STO = 'SALATIGA'
    nm_STO = 'TAMAN SARI'
    tglCetak = datetime.now(pytz. timezone('Asia/Jakarta')).strftime('%d-%m-%Y')
    nm_file = 'Lap_Pemakaian_Barang'

    selectedColumn =["id_pemakaian", "id_karyawan", "nama_karyawan",
"barangid", "namabarang", "quantity", "tanggal_pakai"]
    renameColumn={"id_pemakaian": "NO. STRUK", "id_karyawan": "NIK KARYAWAN",
"nama_karyawan": "NAMA KARYAWAN", "barangid": "KODE BARANG", "namabarang": "NAMA BARANG",
"quantity": "JUMLAH BARANG", "tanggal_pakai": "TANGGAL PAKAI"}

    # Bagian Kedua
    dataPKBrg = getdata()
    dataPKBrg = dataPKBrg['result']
```

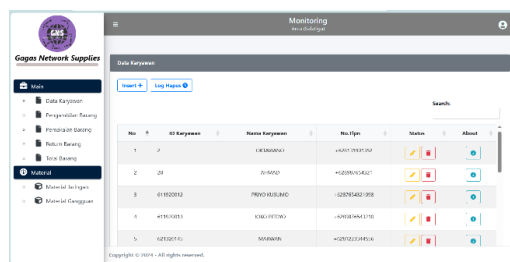
```
# Bagian Ketiga
if len(dataPKBrg) == 0:
    dataPKBrg = [{"id_pemakaian" : "", "id_karyawan": "", "nama_karyawan": "",
"barangid": "", "namabarang": "", "quantity": "", "tanggal_pakai": ""}]
    filename = '{}_{}'.format (nm_file, str(tglCetak))+".xls"
    file_excel = CreateLaporanPakai (dataPKBrg, selectedColumn, renameColumn,
nm_file, 'Laporan Pemakaian Barang', tglCetak, '{}-{}'.format(Daerah_STO,
nm_STO))

    else:
        filename= '{}_{}'.format (nm_file, str(tglCetak))+".xls"
        file_excel = CreateLaporanPakai (dataPKBrg, selectedColumn, renameColumn,
nm_file, 'Laporan Pemakaian Barang', tglCetak, '{}-{}'.format(Daerah_STO,
nm_STO))

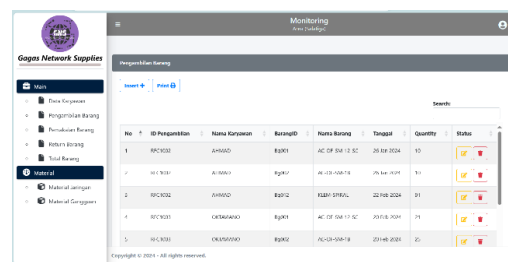
# Bagian KeEmpat
response = Response(file_excel,
content_type='application/vnd.openxmlformats-officedocument.spreadsheetml.sheet')
response.headers['Content-Type'] = 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet'
response.headers['Content-Disposition'] = 'attachment; filename = %s' % filename

return response
```

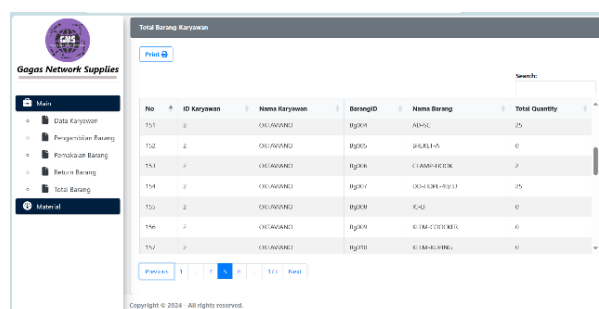
Kode program 3 merupakan fungsi untuk menangani request mencetak *file excel*. Pada bagian pertama, mendefinisikan *metadata* yang akan digunakan dan menentukan list kolom yang diambil dari *dataset* untuk memetakan kolom awal dengan nama kolom baru agar sesuai ketika dibuat dalam *file excel*. Bagian ketiga menangani hasil dari data pada bagian kedua, dan memanggil fungsi *CreateLaporanPakai()* untuk memproses metadata dan hasil tersebut. Bagian terakhir adalah konstruksi *respons*, dimana kode *Response (file excel, content type)* yang akan digunakan untuk membuat objek *Flask Response* dan hasil dari *response* akan dikembalikan untuk mentrigger *browser* agar menampilkan *prompt download*.



Gambar 6. Antar muka DataKaryawan



Gambar 7. Antar muka Pengambilan Barang



Gambar 8. Antar muka Total Barang

Pada gambar 6, merupakan tampilan dari menu Data Karyawan untuk mengelola data seperti menambahkan dengan tombol insert yang akan memunculkan modal LOV berisi *form*. Untuk melihat material yang dimiliki karyawan dapat menekan tombol pada kolom about yang akan menampilkan modal Log Barang. Dapat juga melihat history data yang dihapus dengan tombol Log Hapus yang akan menampilkan data status false. Pada gambar 7 menu untuk Pengambilan Barang. Berbeda dengan menu Data Karyawan, menu ini terdapat pilihan untuk mencetak data yang ditampilkan dengan tombol print agar dapat mendownload *file excel*. Tampilan menu ini sama dengan menu pemakaian barang dan return barang, yang membedakan hanya kode transaksinya. Terdapat kode khusus seperti SC untuk pemakaian, RFR untuk return, dan RFC untuk pengambilan. Dalam gambar 8, menunjukan tampilan menu dari total barang yang tersedia. Menu ini bertujuan untuk menghitung jumlah material yang dimiliki oleh setiap karyawan untuk setiap item, berdasarkan transaksi yang tercatat dalam menu pengambilan, pemakaian, dan pengembalian. Untuk pengumpulan data secara efisien, digunakan *query VIEW* untuk menciptakan virtual tabel dalam *database*. Proses ini memungkinkan pengambilan data secara dinamis tanpa harus menyimpannya diberbagai tabel dengan tujuan untuk menghindari duplikasi data dan untuk memastikan data yang dihasilkan selalu terbaru.

Tabel 1. Tabel pengujian *Blackbox*

No	Skenario Uji	Hasil yang diharapkan	Hasil Uji	Hasil
1.	<i>Username</i> dan <i>Password</i> tidak diisi, atau tidak sesuai	Tidak berhasil login, sistem akan menampilkan pesan notifikasi peringatan	Login gagal, muncul pesan notifikasi	<i>Valid</i>
2.	<i>Username</i> dan <i>Password</i> sesuai	Berhasil masuk, dan sistem menampilkan dashboard	Berhasil login, dan masuk ke halaman dashboard	<i>Valid</i>
3.	Input dan <i>update</i> Menu Data Karyawan	Berhasil Insert atau Update dan akan mengubah data pada tabel	Data pada tabel pengambilan berhasil berubah	<i>Valid</i>
4.	Menghapus data pada DataKaryawan	Akan menghapus data pada tabel	Data dihapus tidak ditampilkan pada tabel pengambilan	<i>Valid</i>
5.	Menampilkan Log Hapus	Menampilkan Modal Log Hapus berisi data yang dihapus (Berstatus False)	Berhasil menampilkan modal Log Hapus (Data False)	<i>Valid</i>
6.	Menambahkan dan <i>update</i> data barang pada menu penambilan, pemakaian, dan pengembalian	Berhasil Input atau Update dan memperbarui tabel	Berhasil mengubah data	<i>Valid</i>
7.	Mengubah data dengan kode transaksi tidak sesuai	Muncul notifikasi dan data tidak berubah	Muncul notifikasi dan data tabel tidak berubah	<i>Valid</i>
8.	Menghapus data barang	Behasil menghapus data	Data barang berhasil dihapus	<i>Valid</i>
9.	<i>Update</i> otomatis data Menu Total Baang	Berhasil mengubah data (dengan mengubah pengambilan)	Data berhasil update otomatis	<i>Valid</i>
10.	Menekan tombol Print	Berhasil mencetak file. <i>xlsx</i> berisikan data	Berhasil download file <i>.xlsx</i> berisikan data	<i>Valid</i>

Setelah melakukan pengujian untuk menilai fungsionalitas dari sistem, maka selanjutnya akan dilakukan *User Acceptance Test* melalui kuesioner untuk menentukan apakah aplikasi yang dirancang sudah sesuai dengan kebutuhan dan keinginan dari pengguna. Pengujian ini melibatkan 15 responden dan diukur menggunakan skala likert.

Tabel 2. Tabel pertanyaan kuesioner

No	Pertanyaan	Skor					Total Skor
		1	2	3	4	5	
1.	Apakah navigasi pada website mudah untuk dijalankan serta dipahami?	-	-	3	12	-	57
2.	Apakah fitur yang disediakan efektif dalam membantu melakukan proses pencatatan data material?	-	-	7	7	1	54
3.	Apakah data yang ditampilkan mampu memberikan informasi yang jelas mengenai pemakaian material?	-	-	4	11	-	56
4.	Apakah website ini dapat meningkatkan kinerja operasional konstruksi jaringan?	-	2	11	2	-	45
5.	Apakah sistem yang ada sangat intuitif dalam mengatur dan mengelompokkan informasi?	-	-	10	5	-	50
6.	Apakah fungsi yang disediakan sudah sesuai dengan kebutuhan monitoring material?	-	-	8	6	1	53
7.	Apakah hasil pelaporan penggunaan material dapat dengan mudah dipahami?	-	-	5	10	-	55

Tabel 3. Tabel Hasil Validitas

Tingkat Validitas	Nilai Korelasi	Nilai t-tabel (pearson)	Kriteria	Interpretasi	Kategori
Pertanyaan 1	0.9982954346	0.514	VALID	0.76	Suka
Pertanyaan 2	0.9980122454	0.514	VALID	0.72	Suka
Pertanyaan 3	0.998221321	0.514	VALID	0.7466666667	Suka
Pertanyaan 4	0.9985215903	0.514	VALID	0.6	Suka
Pertanyaan 5	0.9985532238	0.514	VALID	0.6666666667	Suka
Pertanyaan 5	0.7066666667	0.514	VALID	0.7066666667	Suka
Pertanyaan 6	0.7333333333	0.514	VALID	0.7333333333	Suka
Pertanyaan 7	0.9982954346	0.514	VALID	0.76	Suka

Tabel 4. Tabel Skor Likert

Skor	Keterangan	Presentase
1	Sangat Tidak Suka	0% - 19,99%
2	Tidak Suka	20% - 39,99%
3	Netral	40% - 59,99%
4	Suka	60% - 79,99%
5	Sangat Suka	80% - 100%

$$\text{Skor Likert} = T \times P_n$$

$$\text{Interpretasi Skor : } \begin{aligned} Y &= \text{skala likert tertinggi} \times \text{jumlah responden} \\ X &= \text{skala likert terendah} \times \text{jumlah responden} \end{aligned}$$

$$\text{Rumus Indeks} = \text{Total Skor} / Y \times 100 \%$$

Tabel 2 menunjukkan skor dari setiap jawaban responden untuk 7 pertanyaan yang diberikan. Untuk perhitungan total skor berdasarkan rumus 1 dengan T merupakan jumlah responden yang memilih dan P_n merupakan pilihan angka skor pada likert. Pada tabel 5 untuk pertanyaan pertama $(3 \times 3) + (12 \times 4) = 57$, yang berarti total skor adalah 57. Setelah mendapatkan total skor untuk setiap pertanyaan, selanjutnya menghitung interpretasi skor pada rumus 2 dengan Y = 75 dan X = 15 untuk digunakan pada rumus 3. Hasil interpretasi tertera pada tabel 3 menunjukkan angka berkisar antara 0.6 – 0.76, dengan nilai rata-rata 0.7048 (70.48%). Berdasarkan tabel 4, persentase berada pada skor 4 atau suka dalam rentang 60% - 79,99%. Berdasarkan hasil tersebut sistem yang dibangun memiliki keunggulan dalam peningkatan proses pencatatan material dengan interface yang mudah dijalankan. Namun terdapat kekurangan dalam peningkatan kinerja operasional karena sistem belum mampu diakses secara fleksibel.

4. Kesimpulan

Berdasarkan hasil pembahasan penggunaan *framework Flask* dan *PostgreSQL* dapat mempermudah pembangunan *website* GNS untuk monitoring inventory material. Perusahaan GMJ diharapkan dapat menggunakan *framework* yang memberikan fleksibilitas dengan fitur serta library yang sudah disediakan, sehingga lebih mudah untuk diimplementasikan dan dimengerti. Pemilihan *PostgreSQL* sebagai database juga mendukung untuk transaksi beban kerja yang besar. Dengan mengumpulkan semua konfigurasi serta layanan dalam satu file, mempermudah untuk mengatur maupun mengubah alur pada sistem jika diperlukan sewaktu-waktu. Penggunaan fungsi Flask *render_template* dan *redirect* sangat memudahkan ketika menjalankan file HTML dengan memanggil template engine Jinja2. Sistem yang dihasilkan dapat membantu memberikan solusi permasalahan dalam proses monitoring material konstruksi jaringan dengan memberikan informasi total barang secara *up-to-date*. Berdasarkan hasil pengujian dengan persentase 70.48% menjadikan sistem telah memenuhi standar kebutuhan untuk monitoring material serta membantu pelaksanaan operasional. Namun, sistem ini masih memiliki kekurangan, yaitu terbatasnya penggunaan yang hanya bisa diakses pada *server internal*. Diperlukan adanya sebuah penyesuaian secara terus-menerus untuk mendapatkan hasil yang lebih baik.

5. Daftar Pustaka

- Ardiansyah, F. (2023). Pengembangan Sistem Informasi Keanggotaan Online Berbasis Web menggunakan Framework Laravel dengan Metode Prototype pada Asosiasi Inkindo. *Journal of Research and Publication Innovation*, 1(2), 266-271.
- Budianto, A. J., & Saian, P. O. N. (2023). Pengembangan Modul Inventory Management pada Aplikasi Master Distribution Centre System Menggunakan Framework Flask di PT XYZ. *Jurnal JTik (Jurnal Teknologi Informasi dan Komunikasi)*, 7(2), 201-207. DOI: <https://doi.org/10.35870/jtik.v7i2.714>.
- Fahrezi, A., Salam, F. N., Ibrahim, G. M., Syaiful, R. R., & Saifudin, A. (2022). Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia. *LOGIC: Jurnal Ilmu Komputer dan Pendidikan*, 1(1), 1-5.
- Haezer, E. H. Y. K. E., & Setiyawati, N. (2022). Pembangunan Aplikasi Virtual Inventory System (VIS) Berbasis Web Menggunakan Flask Framework. *Jurnal Mnemonic*, 5(2), 128-135. DOI: <https://doi.org/10.36040/mnemonic.v5i2.4799>.

- Handayani, H., Ayulya, A. M., Faizah, K. U., Wulan, D., Rozan, M. F., & Hamzah, M. L. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 29-40.
- Handayani, H., Faizah, K. U., Mutiara Ayulya, A., Rozan, M. F., Wulan, D., & Hamzah, M. L. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development Designing a Web-Based Inventory Information System Using the Agile Software Development Method. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 29
- Haniva, D. T., Ramadhan, J. A., & Suharso, A. (2023). Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid. *JIEET (Journal of Information Engineering and Educational Technology)*, 7(1), 36-42. DOI: <https://doi.org/10.26740/jieet.v7n1.p36-42>.
- Ichsanudin, M. N., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, 1(2), 1-8. DOI: <https://doi.org/10.55123/storage.v1i2.270>.
- Kesuma, C., & Handayani, V. R. (2022). Sistem Informasi Inventory Alat Kesehatan Menggunakan Metode Fefo. *SPEED-Sentra Penelitian Engineering dan Edukasi*, 14(1). DOI: <http://dx.doi.org/10.55181/speed.v14i1.749>.
- Marques, J. A. L., dos Reis Morais, J. J. B., Alves, J., & Gonçalves, M. (2023). Effectiveness analysis of waterfall and agile project management methodologies—a case study from Macau's construction industry. *Revista Gestão em Análise*, 12(1), 23-38. DOI: <https://doi.org/10.12662/2359-618xregea.v12i1.p23-38.2023>.
- Muliawan, D. (2023). PENERAPAN REPLIKASI SYNCHRONOUS PADA APLIKASI DATA KRS ONLINE PADA UNIVERSITAS JABAL GHAFUR MENGGUNAKAN POSTGRESQL. *Future Academia: The Journal of Multidisciplinary Research on Scientific and Advanced*, 1(1), 31-45. DOI: <https://doi.org/10.61579/future.v1i1.4>.
- Munthe, I. R., Rambe, B. H., Pane, R., Irmayani, D., & Nasution, M. (2020). UML Modeling and Black Box Testing Methods in the School Payment Information System. *Jurnal Mantik*, 4(3), 1634-1640. DOI: <https://doi.org/10.35335/mantik.Vol4.2020.969.pp1634-1640>.
- Pratiwi, A. J., Sukarno, H. B., & Kusumadiarti, R. S. (2024). Perancangan Sistem Informasi Penjualan Obat Pembersih Kerak Lantai Berbasis Web pada PT Duta Jaya Fritzama. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 5(1), 8-19. DOI: <https://doi.org/10.35870/jimik.v5i1.422>.
- Rahmatika, R., Baihaqie, A. D., & Isnain, N. (2022). Design and Build a Human Resource Information System Application at PT XYZ. *JISICOM (Journal of Information System, Informatics and Computing)*, 6(2), 600-610. DOI: <https://doi.org/10.52362/jisicom.v6i2.966>.
- Saifudin, M., & Kautsar, I. A. (2024). Perancangan Aplikasi Inventory E-Canteen Berbasis Web Service. *Journal of Internet and Software Engineering*, 1(1), 18-18. DOI: <https://doi.org/10.47134/pjise.v1i1.2245>.

- Saleh, M. A., Sugiarta, I. G. R. A., & Purwanto, A. (2023). Sistem Informasi Control Inventory di CV. AgungSeafood Berbasis Website Menggunakan Framework Codeigniter. *Jurnal Informasi dan Teknologi*, 234-239. DOI: <https://doi.org/10.37034/jidt.v5i1.300>.
- Santoso, B. B., & Saian, P. O. N. (2023). Implementasi Flask Framework pada Development Modul Reporting Aplikasi Sistem Informasi Helpdesk di PT. XYZ. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 7(2), 217-226. DOI: <https://doi.org/10.35870/jtik.v7i2.718>.
- Suseno, T. A. P., Moh, A. L., Prayoga, R. A. S., & Bagaskara, D. B. (2023, July). Pemerataan Infrastruktur Telekomunikasi untuk Kesejahteraan Digital. In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 7, No. 1, pp. 9-17). DOI: <https://doi.org/10.29407/inotek.v7i1.3399>.
- Susetyo, Y. A. (2022). IMPLEMENTATION OF FLASK FOR STOCK CHECKING IN DISTRIBUTION CENTER & STORE ON MONITORING STOCK APPLICATION IN PT. XYZ. *Jurnal Teknik Informatika (Jutif)*, 3(5), 1265-1274. DOI: <https://doi.org/10.20884/1.jutif.2022.3.5.334>.
- Walingkas, H. L., & Saian, P. O. N. (2023). Penerapan Framework Flask pada Pembangunan Sistem Informasi Pemasok Barang. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 7(2), 227-234. DOI: <https://doi.org/10.35870/jtik.v7i2.729>.