

IMPLEMENTASI FLASK FRAMEWORK PADA PEMBANGUNAN APLIKASI SISTEM INFORMASI HELPDESK (SIH)

Candra Wijayanto¹⁾, Yeremia Alfa Susetyo²⁾

Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Dr. O. Notohamidjojo, Salatiga 50715, Indonesia
e-mail: 672018259@student.uksw.edu¹⁾, yeremia.alfa@uksw.edu²⁾

ABSTRAK

Helpdesk menjadi peranan yang penting dalam mempertahankan dan mengawasi kelancaran proses bisnis. PT. XYZ yang merupakan salah satu perusahaan retail terbesar di-Indonesia dalam menjalankan proses bisnisnya juga memiliki divisi helpdesk, namun proses pelaporan dan pencatatan kendala di PT. XYZ masih dilakukan secara manual sehingga tidak efisien. Oleh karena itu pada penelitian ini dibangunlah aplikasi sistem informasi helpdesk dengan harapan untuk meningkatkan efisiensi dan efektivitas proses bisnis di PT. XYZ. Pembangunan aplikasi dilakukan pada platform web dengan mengimplementasikan Python Flask Framework, Selain menggunakan flask, dalam pembangunan aplikasi ini juga menggunakan bootstrap sebagai UI library pada bagian antarmuka aplikasi, JavaScript untuk membuat tampilan yang lebih atraktif, serta PostgreSQL sebagai sistem penorganisasian dan sistem pengolahan database. Hasil dari penelitian ini adalah sebuah aplikasi sistem informasi helpdesk berbasis web yang dapat digunakan sebagai langkah awal dalam pemecahan masalah jika terjadi kendala pada operasional perusahaan, selain itu melalui aplikasi ini setiap kendala yang dilaporkan juga dapat dimonitoring dengan mudah oleh user. Aplikasi yang dihasilkan juga telah melalui tahap pengujian dengan metode blackbox testing dengan hasil test yang baik atau aplikasi dapat berjalan sesuai dengan yang diharapkan.

Kata Kunci: Helpdesk, Flask framework, Python, Bootstrap, postgresSQL

ABSTRACT

The helpdesk plays an important role in maintaining and supervising the smooth running of business processes. PT. XYZ which is one of the largest retail companies in Indonesia in carrying out its business processes also has a helpdesk division, but the reporting and recording process of problems at PT. XYZ is still done manually so it is not efficient. Therefore, in this study, a helpdesk information system application was built in the hope of increasing the efficiency and effectiveness of business processes at PT. XYZ. Application development is carried out on a web platform by implementing the Python Flask Framework. In addition to using flask, in the development of this application, bootstrap is also used as a UI library in the application interface, JavaScript to create a more attractive appearance, and PostgreSQL as an organizational system and database processing system. The result of this research is a web-based helpdesk information system application that can be used as a first step in solving problems if there are problems with the company's operations. In addition, any reported problems can also be monitored easily by the user.

Keywords: Helpdesk, Flask framework, Python, Bootstrap, postgresSQL

I. PENDAHULUAN

PT. XYZ merupakan salah satu perusahaan retail terbesar di Indonesia yang bergerak di bidang retail. Sebagai perusahaan retail yang memiliki ribuan transaksi per hari, membuat PT XYZ memiliki berbagai macam interaksi dalam proses bisnisnya. Pelaporan kendala atau helpdesk pada barang/inventaris kantor merupakan salah satu contoh interaksi yang dilakukan oleh karyawan jika terjadi kendala pada barang/inventaris kantor. Pelaporan kendala atau helpdesk pada dasarnya adalah titik sentral di mana kendala atau masalah dilaporkan dan diatur secara teratur dan terorganisir [1]□. Sistem helpdesk menjadi peranan yang penting dalam mempertahankan dan mengawasi kelancaran dalam bisnis. Karena dengan adanya helpdesk diharapkan dapat memberikan manfaat, diantaranya: riwayat kendala dicatat sehingga mempercepat pemrosesan jika kendala yang sama terulang, memberikan insight mengenai kinerja suatu divisi dalam menyelesaikan kendala, kendala yang terpusat membuat pembagian tugas tepat dan merata pada divisi yang bersangkutan [2]□.

Dalam menjalankan proses bisnisnya, PT. XYZ juga sering mendapat berbagai laporan kendala mengenai barang/inventaris kantor oleh karyawannya. Namun proses pencatatan laporan kendala di PT. XYZ masih dilakukan secara manual. Tim helpdesk akan mencatat detail kendala dalam *form excel*. Jika dalam proses penanganan dibutuhkan pengecekan lebih lanjut, maka kendala akan di alihkan kepada divisi terkait melalui email.

Tim helpdesk juga perlu mencatat riwayat setiap menangani kendala. Proses pengisian formulir *Excel* dan penggunaan email secara manual ini tentunya tidak efisien [3]□. Hal ini menimbulkan masalah seperti: resiko pencatatan yang duplikat, adanya masalah/kendala yang tidak tercatat karena operator sibuk, atasan yang harus menunggu untuk mendapatkan laporan karena belum adanya rekap bulanan dari keluhan yang telah diproses, dan perhitungan masalah menjadi tidak akurat karena dilakukan secara manual [1]□.

Untuk mengatasi permasalahan tersebut, PT. XYZ melakukan pengembangan aplikasi manajemen helpdesk yang diberi nama Sistem Informasi Helpdesk (SIH) dengan tujuan untuk meningkatkan efisiensi dan efektivitas proses bisnis di PT. XYZ. Pembangunan aplikasi dilakukan pada platform web dengan menggunakan Python Flask *Framework*. Flask merupakan *micro web framework* yang menggunakan bahasa python. Penggunaan *framework* ini bertujuan untuk mempercepat pembangunan aplikasi karena dalam flask sudah disediakan *library* dan kumpulan kode program yang siap digunakan untuk membangun sebuah *webapp* tanpa harus membuatnya dari awal [4]□. Selain itu jika menggunakan flask maka akan mengurangi penggunaan *resource memory* karena flask tergolong dalam *micro-framework* [5]□. Selain menggunakan flask, dalam pembangunan aplikasi ini juga menggunakan bootstrap sebagai *UI library* pada bagian antarmuka aplikasi, serta PostgreSQL sebagai database. Pembangunan aplikasi SIH dengan mengimplementasikan Flask *Framework* ini bertujuan untuk menjadi solusi dalam meningkatkan efisiensi dan efektivitas proses bisnis di PT. XYZ.

Penelitian mengenai *helpdesk* pernah dilakukan sebelumnya seperti pada penelitian dengan judul “**Analisis Problem Management pada IT Helpdesk dengan implementasi ITSM dan SLA (Studi Kasus : Citigroup Indonesia)**”. Dalam penelitian tersebut peneliti membahas mengenai pentingnya *Helpdesk* yang diperlukan untuk mempertahankan dan selalu mengawasi teknologi dalam bisnis. Hasil penelitian tersebut menunjukkan bahwa sistem *Helpdesk* pada bisnis merupakan hal yang penting karena *helpdesk* merupakan saluran untuk memberikan layanan IT kepada pengguna, selain itu juga menjadi sumber untuk mengedukasi pengguna terhadap kendala yang ditemui pada sistem yang berjalan [6]□.

Penelitian terdahulu yang memanfaatkan flask *framework* juga pernah dilakukan, yakni penelitian dengan judul “**Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request**” yang dilakukan oleh Dinda & Nina. Penelitian tersebut dilakukan dengan tujuan untuk mempermudah karyawan dalam melakukan *purchasing request*, serta meningkatkan efektivitas waktu dalam kegiatan *purchasing request & approval*. Pembangunan aplikasi dilakukan menggunakan metode *Rapid Application Development* menggunakan flask *framework*. Luaran dari penelitian tersebut berupa aplikasi “*Purchasing Approval Request*” yang dapat mempermudah user dalam melakukan *purchasing request* [7]□. Dalam penelitian tersebut peneliti berhasil mengimplementasikan flask *framework* untuk membuat aplikasi PAR yang memang menyelesaikan masalah pada peneliti, sehingga menjadi acuan pada penelitian ini untuk mengimplementasikan flask *framework*.

Penelitian lain mengenai pembangunan aplikasi Helpdesk berbasis web juga sudah pernah dilakukan sebelumnya, seperti penelitian yang dilakukan oleh Fauzi & Asep dengan judul “**Perancangan Aplikasi IT Helpdesk Berbasis Web Di Pt. Panca Abadi Nan Jaya**”. Penelitian ini dilakukan karena sistem helpdesk IT pada PT. Panca Abadi Nan Jaya masih menggunakan cara manual sehingga menimbulkan masalah tidak teratasinya masalah, sehingga peneliti membangun aplikasi Helpdesk. *Output* dari penelitian tersebut berupa aplikasi Helpdesk yang dapat mengatasi permasalahan yang ada, serta membantu dalam mengontrol pelaporan masalah yang dilaporkan oleh *user* [8]□. Pada penelitian tersebut hanya membuat sistem *helpdesk* yang sederhana, jika terdapat masalah yang perlu tindakan lebih lanjut maka masalah perlu dialihkan, namun belum ada fitur pengalihan pada aplikasi tersebut, sehingga pada penelitian kali ini akan dibuat sistem helpdesk dengan fitur penyelesaian kendala ke pihak lain.

Berdasarkan beberapa penelitian yang telah dilakukan sebelumnya, didapatkan kesimpulan bahwa sistem *helpdesk* dapat membantu dalam memperlancar proses bisnis, juga dapat meningkatkan efisiensi dalam menangani masalah. Penggunaan flask *framework* menjadi solusi dalam pembangunan aplikasi yang cepat dan ringan. Sehingga akan dilakukan penelitian terkait pembangunan aplikasi sistem informasi *helpdesk* dengan mengimplementasikan flask *framework* yang diberi judul “**Implementasi Flask Framework Pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH)**”.

II. METODE PENELITIAN



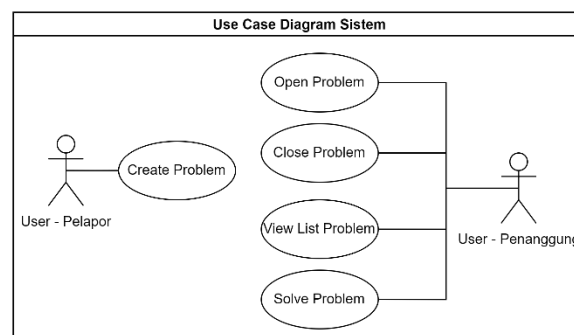
Gambar 1 Metode penelitian

Pada Gambar 1 menjelaskan setiap tahapan yang dilakukan dalam penelitian ini. Tahapan awal yaitu mengidentifikasi permasalahan yang ada, permasalahan yang ada pada PT. XYZ yaitu kurang efektifnya sistem helpdesk yang ada karena masih dilakukan secara manual. Sehingga akan dibangun sistem helpdesk yang lebih baik dengan mengimplementasikan Python Flask Framework.

Setelah permasalahan teridentifikasi, masuk pada tahapan studi literatur, studi literatur merupakan teknik pengumpulan informasi dan data yang berasal dari ringkasan tertulis mengenai artikel jurnal, buku, dan dokumen lain untuk mendapatkan informasi yang dibutuhkan sebagai bahan pendukung penelitian ini [9]. Beberapa sumber literatur yang dipakai dalam penelitian ini adalah literatur yang berhubungan dengan Flask *Framework*, Bootstrap, dan PostgreSQL.

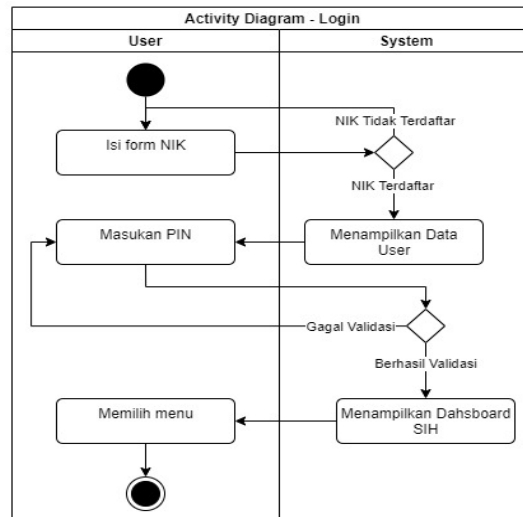
Tahapan selanjutnya melakukan analisa kebutuhan sistem, yaitu analisa terhadap kebutuhan sistem yang akan dibangun dengan cara melakukan observasi mengenai proses penanganan kendala pada PT. XYZ yang nantinya digunakan sebagai acuan dalam perancangan sistem.

Masuk pada tahap perancangan sistem, yaitu tahapan untuk menentukan bagaimana suatu sistem akan beroperasi dengan melakukan penggambaran, perencanaan dan pembuatan sketsa dari berbagai elemen seperti arsitektur, *user interface*, *database*, dan hal lainnya yang diperlukan ke dalam satu kesatuan yang utuh dan berfungsi [10]. Pada penelitian ini rancangan sistem yang dibuat meliputi *Unified Modeling Language* (UML) diagram. UML adalah suatu metode pemodelan secara visual untuk mendokumentasikan, menspesifikasikan, dan membangun suatu sistem perangkat lunak [11]. Pada Gambar 2 menggambarkan diagram *use case* pada aplikasi yang dibuat. *Use Case Diagram* merupakan diagram yang menggambarkan interaksi antara sistem dengan pelaku atau aktor [12].



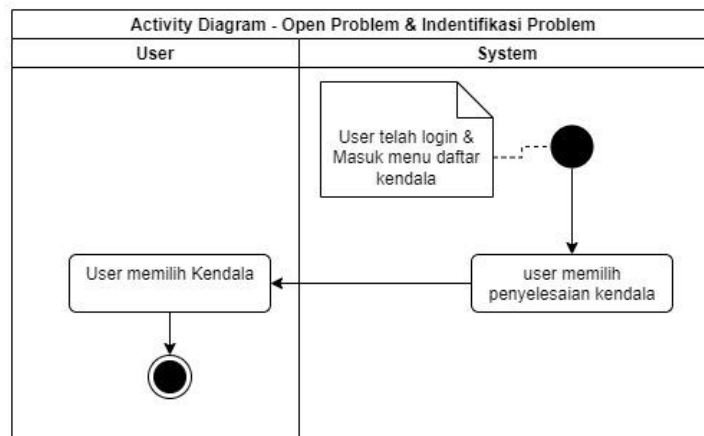
Gambar 2 Use Case Diagram

Diagram *use case* pada Gambar 2 menjelaskan terdapat dua aktor yaitu aktor Pelapor, dan Penanggung. Tiap aktor memiliki hak akses yang berbeda-beda. Aktor Pelapor dapat melakukan aktivitas *create problem*, yaitu proses untuk membuat laporan kendala kepada tim helpdesk atau aktor penanggung. Aktor Penanggung dapat melakukan aktivitas *open problem* untuk menampilkan seluruh *problem* yang dilaporkan oleh user pelapor, aktivitas *close problem* untuk menyatakan bahwa kendala telah selesai ditangani, aktivitas *View List Problem* untuk menampilkan seluruh *problem* yang dilaporkan oleh *user* pelapor, dan aktivitas *Solve Problem* untuk melempar tanggung jawab penanggung kendala kepada pihak yang lebih berwenang. Pada Gambar 3, 4, 5, dan 6 menggambarkan *activity diagram* dari aplikasi yang dibuat. *Activity Diagram* merupakan sebuah teknik untuk menggambarkan logika prosedural, proses bisnis dan jalur kerja sistem [13]



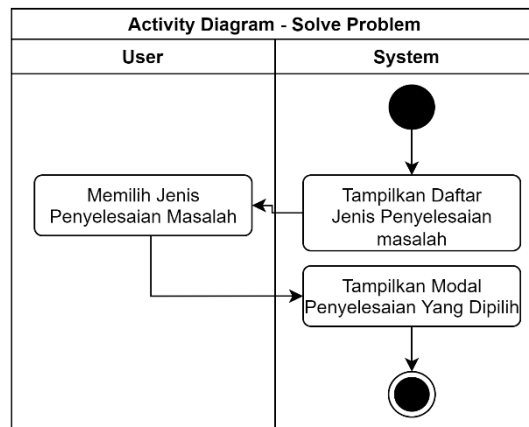
Gambar 3 Activity Diagram - Login

Activity login digambarkan pada Gambar 3 diatas. *User* memasukkan NIK (Nomor Induk Karyawan) yang kemudian diproses oleh sistem, jika NIK terdaftar maka akan menampilkan detail data user berupa nama dan lokasi kerja tempat NIK tersebut terdaftar, kemudian *user* harus memasukkan PIN untuk dapat menyelesaikan *activity login*, jika NIK atau PIN salah, maka *user* diminta untuk mengulanginya.



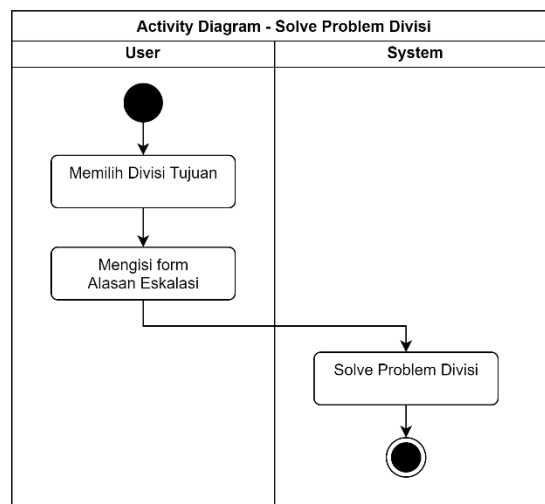
Gambar 4 Activity Diagram - Open Problem & Identifikasi Masalah

Setelah berhasil *login*, *user* akan diarahkan pada menu daftar kendala, pada menu ini *user* bisa melihat seluruh daftar kendala yang dilaporkan, kemudian penanggung dapat memilih penyelesaian kendala yang tersedia.



Gambar 5 Activity Diagram – Penyelesaian masalah

Gambar 5 merupakan *activity* penyelesaian masalah, dimana *user* memilih jenis penyelesaian masalah yang tersedia, kemudian sistem akan menampilkan modal sesuai dengan jenis penyelesaian yang dipilih oleh user, karena tiap jenis penyelesaian memiliki *activity* yang berbeda.

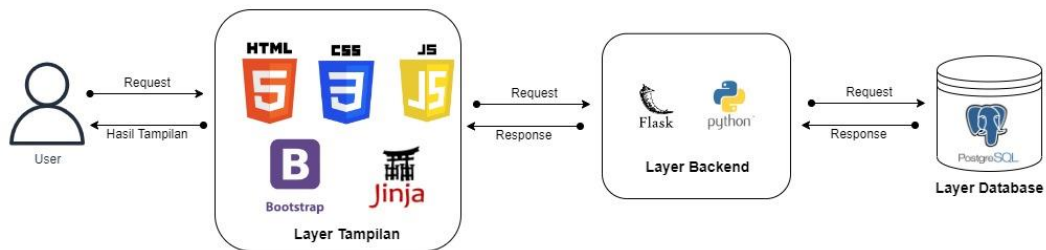


Gambar 6 Activity Diagram – Solve Problem Divisi

Jika *user* memilih jenis penyelesaian divisi, maka alur *activity*-nya seperti yang terpapar pada **Gambar 6**. *User* memilih divisi tujuan penanggung untuk menyelesaikan masalah, kemudian user diminta untuk mengisi form alasan, setelah selesai mengisi form yang diperlukan, maka sistem akan melakukan proses penyelesaian kendala. Tahap selanjutnya adalah tahap pembangunan aplikasi berdasarkan rancangan sistem yang telah dibuat. Pembangunan aplikasi ini dilakukan dengan mengimplementasikan Flask *framework*, selain itu juga menggunakan Bootstrap untuk pembuatan halaman antarmuka *webapp* dan PostgreSQL untuk pembuatan *database*. Tahapan terakhir adalah pengujian sistem, yaitu tahap pengujian terhadap aplikasi yang telah dibangun untuk melihat apakah aplikasi yang dihasilkan sudah dapat berjalan sesuai dengan yang diharapkan atau tidak [14]. Metode pengujian yang digunakan pada penelitian ini adalah metode *black box testing*, yaitu pengujian yang dilakukan sepenuhnya dengan hanya berfokus pada kebutuhan dan spesifikasi *software*. Metode ini hanya cukup meninjau input dan output dari sistem perangkat lunak yang ada tanpa perlu mengetahui struktur kontrol internal perangkat lunak [15].

III. HASIL DAN PEMBAHASAN

Berdasarkan perancangan sistem dan pembangunan aplikasi yang telah dibuat, maka penelitian ini menghasilkan *output* berupa aplikasi yang diberi nama Sistem Informasi Helpdesk (SIH) berbasis web yang dibangun menggunakan Flask *Framework*, JavaScript, Bootstrap dan PostgreSQL sehingga memungkinkan untuk dijalankan pada beragam sistem operasi yang memiliki support untuk *web browser* seperti Windows, Android, GNU/Linux, DLL.



Gambar 7 Arsitektur Aplikasi

Gambar 7 merupakan arsitektur aplikasi. Pada *layer tampilan* ada beberapa teknologi yang digunakan diantaranya Javascript yang berfungsi untuk menyalurkan *inputan* atau permintaan *user* ke-*layer backend*, juga mengolah dan menampilkan data yang diterima dari *layer backend* dengan menggunakan protokol transfer hiperteks, selain itu Javascript juga digunakan untuk memanipulasi tampilan aplikasi sehingga menjadi atraktif. Teknologi HTML, CSS dan Bootstrap digunakan untuk mempercantik tampilan aplikasi, teknologi lain yang digunakan pada *layer tampilan* ini adalah Jinja yang membantu dalam mengolah komponen tampilan yang sama untuk dapat digunakan berulang kali. *Layer backend* bertugas untuk memproses dan mengolah permintaan *user*. Alur kerja yang terjadi pada *layer* ini adalah *user* melakukan *request*, kemudian *request* yang diterima di-sanitasi untuk tujuan keamanan dan juga diolah sebelum akhirnya diteruskan ke *layer database* untuk menyimpan atau mendapatkan data yang diperlukan. *Response* atau balikan yang diterima dari *layer database* kemudian diolah lagi sesuai kebutuhan sebelum akhirnya dikembalikan ke *layer tampilan*. Teknologi yang digunakan pada *layer backend* ini adalah Python dan Flask Framework. *Layer database* menggunakan teknologi PostgreSQL yang berperan dalam semua hal yang berurusan dengan data meliputi DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*). *Layer database* ini dikontrol langsung oleh *layer backend*.

```

1   from flask import Flask
2   from flask_login import login_manager
3
4   app = Flask(__name__)
5   app.config.from_pyfile('setting.py')
6
7   login_manager = LoginManager(app)
8   login_manager.login_view = 'login'
9
10  from .controllers.* import *
  
```

Kode Program 1 __init__.py

Kode Program 1 merupakan kode awal dalam pembuatan aplikasi menggunakan flask *framework*. Baris satu dan dua merupakan proses *import modul* flask *framework*. Pada baris empat merupakan tahap pembuatan *instance* flask yang dimasukkan dalam *variable* app. Pada baris lima merupakan proses konfigurasi informasi kredensial yang akan digunakan pada aplikasi.

```

1   import os
2   SECRET_KEY = os.getenv('secret_key')
3   USER_POSTGRES_DB= os.getenv('db_user_postgres')
4   PASSWORD_POSTGRES_DB=os.getenv('db_password_postgres')
5   HOST_POSTGRES_DB=os.getenv('db_host_postgres')
6   PORT_POSTGRES_DB=os.getenv('db_port_postgres')
7   DATABASE_POSTGRES_DB=os.getenv('db_sid_postgres')
  
```

Kode Program 2 settings.py

Untuk alasan keamanan, data penting seperti info *login* tidak langsung diletakkan atau *hardcode* pada kode program, melainkan disimpan pada *environment variable*

sistem operasi. Kode Program 2 digunakan untuk mendapatkan informasi kredensial yang disimpan dalam *environment variable* sistem operasi. Informasi kredensial ini nantinya digunakan untuk koneksi dengan *database postgresSQL*.

```

1      from . import app
2
3      if __name__ == '__main__':
4          app.run(host='localhost', port=8080)

```

Kode Program 3 main.py

Kode Program 3 digunakan untuk mulai menjalankan aplikasi flask yang telah dibuat. Baris satu merupakan kode untuk *import instance* flask yang telah dibuat pada Kode Program 1 sebelumnya, kemudian pada baris tiga hingga empat merupakan kode untuk mengeksekusi kode *instance* flask yang telah di-*import*.

```

1      from . import app
2      import psycopg2
3
4      class Query(object):
5          def __init__(self):
6              self._db_con =
7                  psycopg2.connect(user=app.config["USER_POSTGRES_DB"],
8                                  password=app.config["PASSWORD_POSTGRES_DB"],
9                                  host=app.config["HOST_POSTGRES_DB"],
10                                 port=app.config["PORT_POSTGRES_DB"],
11
12                                 database=app.config["DATABASE_POSTGRES_DB"])
13              self._db_cursor = self._db_connection.cursor()
14
15          def __del__(self):
16              self._db_cursor.close()
17              self._db_connection.close()
18
19          def response(status, flag, message, result):
20              return {
21                  'status_code': status,
22                  'status': flag,
23                  'message': message,
24                  'result': result
25              }
26
27          def select(self, query, param):
28              try:
29                  result = self._db_cursor.execute(query, param)
30                  return response(200, 'T', 'Sukses', result)
31              except Exception as e:
32                  return (200, 'F', str(e), None)
33
34          def execute(self, query, param):
35              try:
36                  self._db_cursor.execute(query, param)
37                  self._db_connection.commit()
38                  return response(200, 'T', 'Sukses', None)
39              except Exception as e:
40                  return (200, 'F', str(e), None)

```

Kode Program 4 Query.py

Kode Program 4 merupakan kode pembuatan *class* yang menangani pertukaran data antara *layer database* dengan *layer backend*. Baris lima hingga sepuluh merupakan *class constructor* yang akan membuat koneksi *database*. Baris ke-27 hingga ke-32 merupakan *method* untuk melakukan *select statement*. Baris ke-34 hingga ke-40 merupakan *method* untuk melakukan *execute statement*. Sedangkan baris ke-15 hingga ke-17 merupakan *class destructor* yang akan menutup koneksi *database* tepat setelah program selesai melakukan operasi *query database*,

penutupan koneksi perlu dilakukan untuk mengurangi beban kinerja *server* karena melayani banyak *session* dalam satu waktu.

```

1  @app.route("/searchUserAccount", methods=['POST'])
2  def searchUserAccount():
3      try:
4          v_nik = request.form.get('nik', None)
5
6          v_hasil = getDataUserLogin(v_nik)
7
8          if(v_hasil['status'] == 'T'):
9              ms= 'Maaf akun ini telah dinonaktifkan oleh admin!'
10
11             if len(v_hasil['result']) != 0:
12                 return jsonify(v_hasil['result'][0]), 200
13             elif len(v_hasil['result']) == 0:
14                 return ajaxError(False, "NIK Belum Terdaftar!")
15
16             if(v_hasil['result'][0]['aktive'] != 'T'):
17                 return ajaxError(False, ms)
18
19             raise Exception(v_hasil)
20         except Exception as error:
21             return ajaxError(False, "Gagal Mendapatkan Data User!")

```

Kode Program 5 loginController.py

Kode Program 5 merupakan kode untuk proses pengambilan data *user*, pengambilan data dilakukan pada baris ke-6 yang dijabarkan pada Kode Program 6. Baris ke-13 hingga ke-14 merupakan proses pengecekan bilamana user tidak terdaftar maka program akan memunculkan pesan bahwa *user* belum terdaftar. Baris ke-16 hingga baris ke-17 adalah proses pengecekan ulang, bilamana *user* terdaftar namun status akunya dibekukan oleh admin maka akan menampilkan pesan info tidak bisa *login*. Baris ke-11 hingga ke-12 merupakan proses menampilkan data nama dan lokasi kerja *user* jika user terdaftar dalam sistem dan statusnya tidak dibekukan oleh admin.

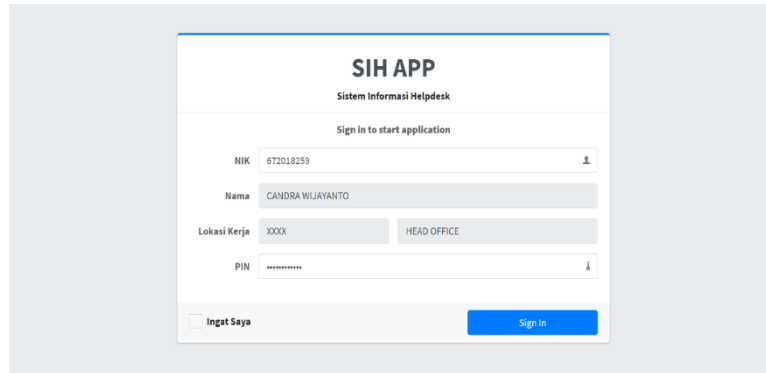
```

1  def getDataUserLogin(nik):
2      db = Query()
3      query = ''' SELECT
4                  k.nik AS nik_karyawan,
5                  k.nama AS nama_karyawan,
6                  k.status,
7                  t.kode AS kode_toko
8                  t.nama AS nama_toko
9      FROM
10         tbl_karyawan k
11         INNER JOIN tbl_toko
12             ON k.kode_toko = t.kode
13     WHERE
14         k.nik = %(nik)s
15     '''
16     return db.select(query, {'nik': nik})

```

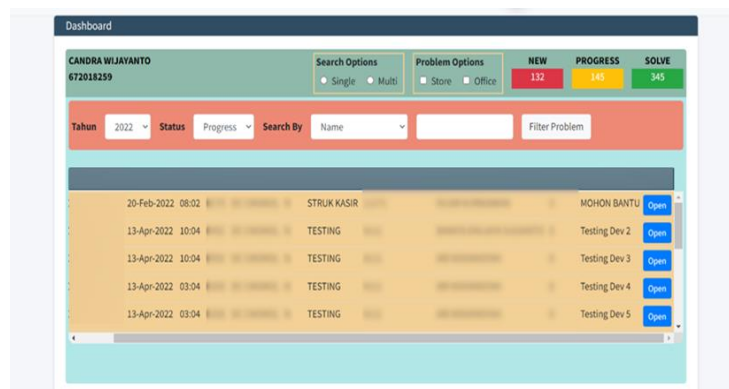
Kode Program 6 loginDAO.py

Kode Program 6 merupakan kode untuk melakukan eksekusi *select query* pada *database*, dimana parameter *nik* digunakan sebagai *primary key* dalam pencarian data *user* atau karyawan. Hasil dari eksekusi kode program ini akan digunakan sebagai bahan validasi *login* yang ada pada Kode Program 5.



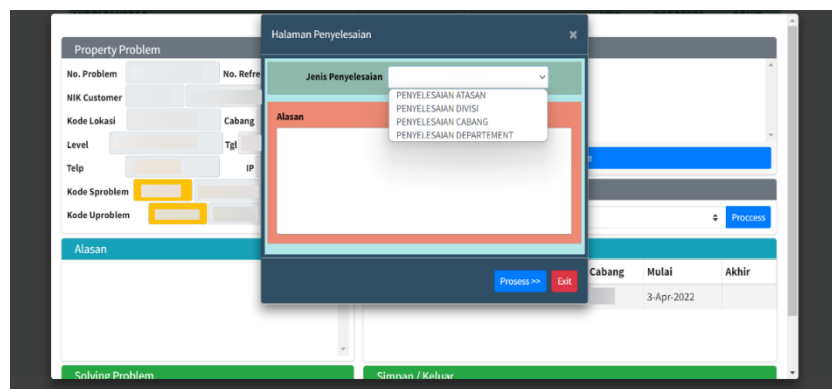
Gambar 8 Tampilan Login

Gambar 8 merupakan tampilan halaman login setelah setelah user mengisi *form* NIK & pin.



Gambar 9 Tampilan View List Problem

Gambar 9 merupakan tampilan pada menu *View List Problem*, tampilan ini akan muncul tepat setelah user berhasil melakukan proses *login*.



Gambar 10 Tampilan modal penyelesaian kendala

Gambar 10 merupakan tampilan untuk *activity* Penyelesaian Masalah. Penyelesaian masalah dilakukan bilamana penanggung merasa bahwa problem yang sedang ditangani di luar tanggung jawabnya sehingga dilakukan pelemperan tanggung jawab kepada penanggung lain.

TABEL I
TABEL PENGUJIAN

No	Langkah Pengujian	Yang diharapkan	Kesimpulan
1	User isi NIK, kemudian mengisi PIN	Login berhasil dan masuk ke halaman utama	Sukses
2	User melakukan penyelesaian divisi IT	Muncul modal penyelesaian divisi dengan tambahan input divisi tujuan	Sukses
3	User mengganti dari jenis penyelesaian divisi IT ke divisi selain IT	Hanya muncul form alasan.	Sukses
4	User melakukan penyelesaian departemen, kemudian klik "Penyelesaian>>"	Muncul modal tambahan untuk memilih PIC	Sukses
5	User tidak mengisi form penyelesaian dengan lengkap	Muncul pesan error "Harap isi form kosong!"	Sukses
6	User melakukan Open Problem menyimpan problem tanpa melakukan identifikasi masalah	Muncul pesan error "Harap isi form dengan benar!"	Sukses
7	Mengganti Filter tahun ke tahun lain	Program menampilkan problem pada tahun yang dipilih	Sukses
8	Login dengan akun lain	Tampilan NIK dan nama berubah sesuai dengan data login	Sukses

Tabel 1 merupakan hasil pengujian sistem yang dilakukan oleh tim *quality assurance* PT. XYZ. Metode yang digunakan adalah *blackbox testing*. Berdasarkan hasil pengujian delapan dari delapan fungsi yang dibuat berjalan sesuai dengan yang diharapkan.

IV. KESIMPULAN DAN SARAN

Pada penelitian ini dilakukan pembangunan aplikasi Sistem Informasi Helpdesk dengan mengimplementasikan Flask *Framework*. Berdasarkan hasil dan pembahasan yang ada dapat disimpulkan bahwa aplikasi Sistem Informasi Helpdesk ini dapat digunakan sebagai langkah awal dalam pemecahan masalah jika terjadi kendala pada operasional perusahaan, Selain itu setiap kendala yang dilaporkan juga dapat dimonitoring dengan mudah oleh user, sehingga dapat meningkatkan efisiensi dan efektivitas proses bisnis di PT. XYZ.

Pada aplikasi yang dibangun masih terdapat kekurangan berupa tampilan aplikasi yang tidak responsif jika dibuka menggunakan perangkat yang memiliki ukuran layar kecil seperti Android dan iOS, sehingga tampilan program menjadi berantakan dan sulit digunakan. Pada pengembangan berikutnya diharapkan dapat memperhatikan pada bagian responsivitas tampilan aplikasi.

DAFTAR PUSTAKA

- [1] W. Likhar and H. Purwanto, "ANALISA DAN PERANCANGAN SISTEM INFORMASI TICKETING HELPDESK ONLINE BERBASIS WEB: STUDI KASUS PT XYZ Weli Likhar 1, Hari Purwanto 2 1," 2021.
- [2] N. L. G. P. Suwirmayanti, I. K. A. A. Aryanto, I. G. A. N. W. Putra, N. K. Sukerti, and R. Hadi, "Penerapan Helpdesk System dengan Pengujian Blackbox Testing," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 2, no. 02, 2020, doi: 10.46772/intech.v2i02.290.
- [3] M. Ardiansyah, "Penerapan Model Rapid Application Development pada Aplikasi Helpdesk Trouble Ticket PT. Satkomindo Mediyasa," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 2, p. 43, 2019, doi: 10.32493/jtsi.v2i2.2759.
- [4] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," no. May, pp. 1–40, 2020, [Online]. Available: <http://www.theseus.fi/handle/10024/339796>.
- [5] K. Relan, *Building REST APIs with Flask*, 1st ed. Berkeley, CA: Apress, 2019.
- [6] L. Magdalena, "Analisis Problem Management pada IT Helpdesk dengan implementasi ITSM dan SLA (Studi Kasus : Citigroup Indonesia)," *J. Digit.*, vol. 1, no. 2, pp. 97–112, 2017, doi: <https://doi.org/10.51920/jd.v1i2.2>.
- [7] D. F. Ningtyas and N. Setiyawati, "Implementasi Flask Framework pada Pembangunan Aplikasi Purchasing Approval Request," *J. Janitra Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 19–34, 2021, doi: 10.25008/janitra.v1i1.120.
- [8] A. Fauzi and A. T. Suryadi, "Perancangan Aplikasi IT Helpdesk Berbasis Web di PT. Panca Abadi Nan Jaya," *J. RESPONSIF Ris. Sains Inform.*, vol. 2, no. 1, pp. 99–105, 2020.
- [9] A. H. Fitri, "PENGARUH VARIASI LATIHAN TERHADAP KEMAMPUAN DRIBBLING DALAM PERMAINAN SEPAK BOLA : STUDI LITERATUR Alchonity Harika Fitri .," no. 1, pp. 1–5, 2021.
- [10] A. Dennis, B. H. Wixom, and R. M. Roth, *System Analysis and Design 5th Edition*, vol. 54, no. SUPPL. 3, 2012.
- [11] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 1, p. 77, 2018, doi: 10.25126/jtiik.201851610.
- [12] I. Abdulruof, O. Ahmed, M. Elamin, and E. Daleel, "ISSN : 2249-0868 Foundation of Computer Science FCS," *Int. J. Appl. Inf. Syst.*, vol. 12, no.



- 34, pp. 1–4, 2020, [Online]. Available: www.ijais.org.
- [13] Z. H. Muhamad, D. A. Abdulmonim, and B. Alathari, “An integration of uml use case diagram and activity diagram with Z language for formalization of library management system,” *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 3069–3076, 2019, doi: 10.11591/ijece.v9i4.pp3069-3076.
- [14] L. C. Hermawan, M. R. Mubarak, H. Mairudin, A. Mahdiyan, and Y. Yulianti, “Penguujian Black Box pada Aplikasi Verifikasi Data Nasabah dengan Menggunakan Metode Boundary Value Analysis,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 3, p. 119, 2020, doi: 10.32493/jtsi.v3i3.5331.
- [15] T. S. Jaya, “Penguujian Aplikasi Dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung),” *J. Inform. J. Pengemb. IT*, vol. 3, no. 2, pp. 45–48, 2018.