JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

PERANCANGAN DOMAIN SPECIFIC LANGUAGE PADA PEMBUATAN APLIKASI FRAMEWORK REPORTING DENGAN MENGGUNAKAN PYTHON FLASK

Budi Ariyansa*[1)], Nina Setiyawati[2)]
1. Universitas Kristen Satya Wacana, Salatiga, Indonesia
2. Universitas Kristen Satya Wacana, Salatiga, Indonesia

Article Info
Kata Kunci: *Domain Specific Language;* Aplikasi Pelaporan; *General Purpose Language*; File Konfigurasi

Keywords: *Domain Specific Language; Reporting Application; General Purpose Language; Configuration File*

Article history:
Received 2 June 2023
Revised 16 May 2023
Accepted 30 June 2023
Available online 1 December 2023

DOI :
https://doi.org/10.29100/jipi.v8i4.4043

* Corresponding author.
Budi Ariyansa
E-mail address:
672019076@student.uksw.edu

ABSTRAK
Pertambahan jumlah data harus diikuti dengan sistem pengelolaan data yang lebih baik. Hal ini akan berdampak pada pembuatan laporan perusahaan yang semakin kompleks. Tingkat kompleksitas akan bertambah dengan pembuatan sistem yang pada umumnya menggunakan *general purpose language* yang memiliki kesulitan jika diterapkan dengan alasan memiliki sifat yang umum dan juga membutuhkan suatu keahlian khusus dalam pemrograman. Untuk mengatasi hal ini dibutuhkan sistem yang dapat mengelola data laporan yang lebih baik. Oleh karena itu tujuan dari penelitian ini adalah merancang dan membangun sistem aplikasi pembuat laporan dengan menggunakan *domain specific language* sebagai perantara penulisan perintah–perintah dalam pembuatan laporan sesuai dengan kebutuhan. Sistem ini akan mempermudah pembuatan laporan dengan membaca *file* konfigurasi yang diberikan sesuai dengan format dan tata bahasa yang telah ditentukan. Pengujian dilakukan dengan menggunakan metode *Black Box Testing* dan metode *System Usability Scale* dengan hasil *testing* yang baik dan sesuai dengan harapan. Dengan demikian sistem aplikasi pembuatan laporan dengan menggunakan *domain specific language* dapat digunakan untuk membuat laporan dengan lebih baik dan lebih cepat.

ABSTRACT
The increase in the amount of data must be followed by a better data management system. This will have an impact on making increasingly complex company reports. The level of complexity will increase with the creation of systems that generally use general purpose languages that have difficulties if applied on the grounds that they have a general nature and also require special skills in programming. To overcome this, a system that can better manage report data is needed. Therefore, the purpose of this research is to design and build a report-making application system using a specific language domain as an intermediary for writing commands in making reports according to needs. This system will make it easier to create reports by reading the given configuration files according to the predetermined format and grammar. Testing is carried out using the Black Box Testing method with good testing results and following expectations. This the report creation application system using a specific language domain can be used to make reports better and faster.

I. PENDAHULUAN

SETIAP bidang usaha pada umumnya membuat laporan dengan tujuan memberikan informasi secara lengkap dan juga menjadi media pengawasan terhadap kinerja yang telah dilakukan selama periode tertentu. Laporan ini juga dapat digunakan sebagai tolak ukur kinerja suatu usaha mengalami kemunduran atau kemajuan contohnya laporan perusahaan. Laporan harus dibuat secara tepat dan akurat agar terhindar dari kesalahan dalam pengolahan data dan perhitungan yang dapat berakibat pada perkembang bidang usaha atau perusahaan. Pembuatan laporan dapat menerapkan teknologi seiring perkembangan teknologi dan informasi yang semakin pesat. Teknologi informasi sangat berperan besar dalam perkembangan perusahaan dan kemajuan teknologi informasi memungkinkan pengambilan dan pengolahan data dalam jumlah besar dalam waktu singkat dan terus menerus.

Namun, keputusan dalam pengembangan teknologi informasi sering didasarkan pada kemampuan dan kecanggihan perangkat lunak serta tidak melihat kesesuaian dengan kebutuhan organisasi dalam jangka panjang

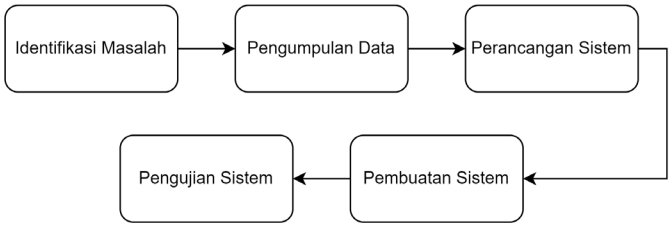
JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

[1]. Oleh karena itu, teknologi yang digunakan harus dievaluasi dan dikembangkan agar fungsionalitas yang diberikan sesuai dan menjadi solusi yang tepat guna pada permasalahan tertentu. Penerapan teknologi dalam pembuatan laporan pada umumnya dengan membuat sebuah sistem yang dapat membuat laporan secara otomatis dan cepat. Namun sebagian besar sistem yang dibuat masih menggunakan *general purpose language* atau bahasa pemrograman umum yang memiliki tujuan dan sifat yang umum sehingga jika diterapkan dalam permasalahan spesifik membutuhkan waktu dan tenaga yang lebih. Penggunaan *domain specific language* dapat menyelesaikan permasalahan tersebut. *Domain specific language* menjadi bahasa pemrograman yang sangat ekpresif atau dapat memberikan gambaran terhadap suatu masalah di dalam sebuah *domain* [2]. *Domain specific language* juga memungkinkan peningkatan proses pengembangan aplikasi secara teknis dengan produktivitas yang lebih tinggi dan lebih baik dibanding *general purpose language* [3].

Beberapa penelitian terdahulu yang diambil sebagai acuan dan pertimbangan dilakukannya penelitian ini. Dalam penelitian berjudul "**Pengembangan *Domain Specific Langauge* Untuk Aplikasi CRUD Berbasis *Web***" menjelaskan bahwa dengan menggunakan DSL, dapat menentukan fungsionalitas dan secara otomatis menghasilkan sebuah kode sumber dan aplikasi CRUD sederhana untuk menyimpan data siswa [4]. Dalam penelitian yang berjudul "***The Development of Domain Specific Language For Email Sorting***" menjelaskan bahwa dengan penggunaan DSL dapat meningkatkan produktivitas perangkat lunak dengan mengabstraksi kode *boilerplate* (kode yang sering digunakan berulang–ulang diberbagai tempat) untuk menghasilkan metode kueri *email* yang ramah pengguna [5]. Dalam penelitian yang berjudul "**Analisis dan Perancangan *Domain Specific Language* untuk Data *Generator* pada *Relational Database***" membahas tentang pembuatan alat *generator* data dengan menggunakan DSL dengan alasan agar proses pembuatan data *dummy* yang biasanya digunakan dalam pembelajaran basis data, pengujian perangkat lunak maupun praktek lainnya akan lebih cepat [6].

Menurut penelitian–penelitian yang telah dilangsungkan sebelumnya, diperoleh bahwa penggunaan *domain specific language* dapat membantu dan mempercepat penyelesaian masalah yang spesifik dan juga dapat meningkatkan produktivitas perangkat lunak. Peningkatan akan lebih baik lagi dengan penggunaan *python flask* dalam membangun aplikasi yang cepat dan ringan. *Flask* sendiri adalah kerangka kerja mikro yang ditulis dengan bahasa *Python*. Biasanya digunakan untuk penyusunan aplikasi dengan cepat karena telah menyediakan komponen-komponen lengkap yang dapat digunakan dalam membangun sebuah *web app* tanpa harus membuatnya dari awal [7]. Sehingga akan dilakukan penelitian untuk merancang dan membangun *domain specific language* pada pembuatan aplikasi *reporting* dengan menggunakan *python flask*. Penelitian ini juga bertujuan untuk merancang dan membangun *domain specific language* pada aplikasi *reporting* sehingga pembuatan laporan akan menjadi lebih baik dan lebih cepat dan juga memberikan informasi kepada pembaca mengenai *domain specific language* dalam bentuk sistem aplikasi pembuat laporan.

II. METODE PENELITIAN



Gambar. 1. Metode Penelitian

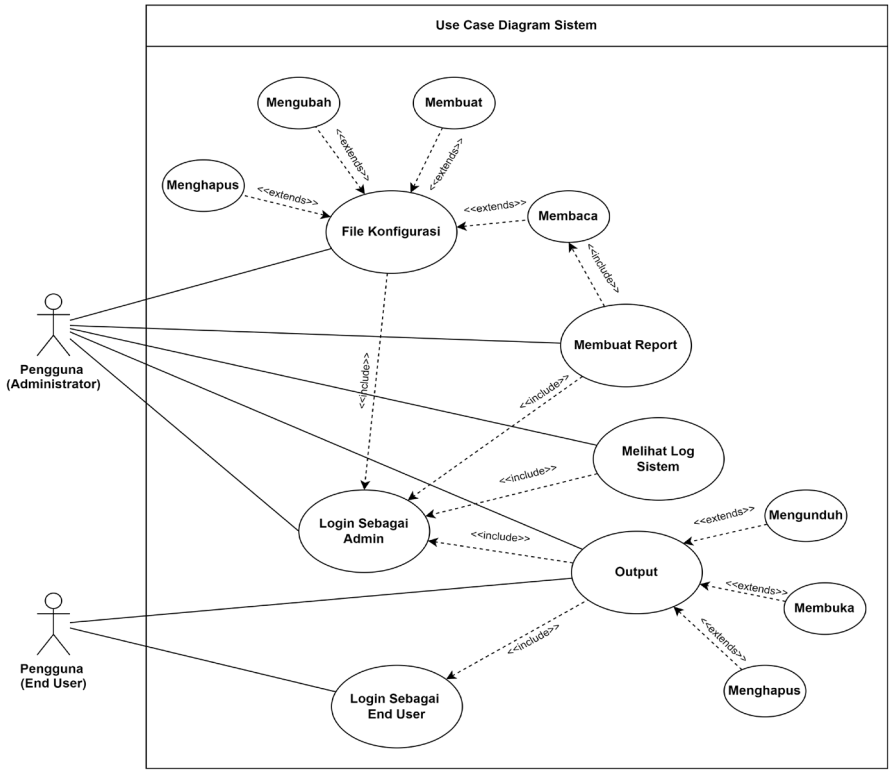
Berdasarkan Gambar 1 menggambarkan tentang tahapan penelitian yang dilangsungkan. Pada tahap pertama yaitu mengidentifikasi permasalahan dalam kasus ini adalah permasalahan sistem aplikasi *reporting* yang umumnya sistem dibuat menggunakan *general purpose language* atau bahasa pemrograman umum yang memiliki tujuan dan sifat yang umum sehingga jika diterapkan dalam permasalahan spesifik membutuhkan waktu dan tenaga yang lebih. Sehingga dibutuhkannya sistem yang dapat menyelesaikan permasalahan tersebut dengan tingkat kesulitan yang lebih rendah dan lebih baik yaitu dengan menggunakan *domain specific language*. *Domain specific language* (*DSL*) merupakan suatu bahasa pemrograman yang didesain untuk menyelesaikan masalah yang spesifik dari suatu domain dan mempunyai sintaks yang terbatas. *DSL* berbeda dengan bahasa pemrograman yang dibuat untuk tujuan misalnya *CSS* (*Cascading Style Sheet*), *HTML* (*Hypertext Markup Language*), dan lainnya. *Domain specific language* juga bersifat *reusable* dan berguna untuk mempercepat pengaturan suatu perkejaan [4].

Setelah melakukan identifikasi masalah, lanjut ke tahap pengumpulan data. Pada tahap ini data yang dikumpulkan berasal dari observasi kebutuhan sistem mengenai proses pembuatan sistem aplikasi *reporting* dan juga ringkasan dokumen seperti artikel jurnal, report, buku dan dokumen lainnya yang dijadikan sebagai acuan

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

dalam penelitian. Sumber – sumber yang digunakan dalam penelitian ini berkaitan dengan *Domain Specific Language, Flask, Bootstrap* dan *SQLite*.

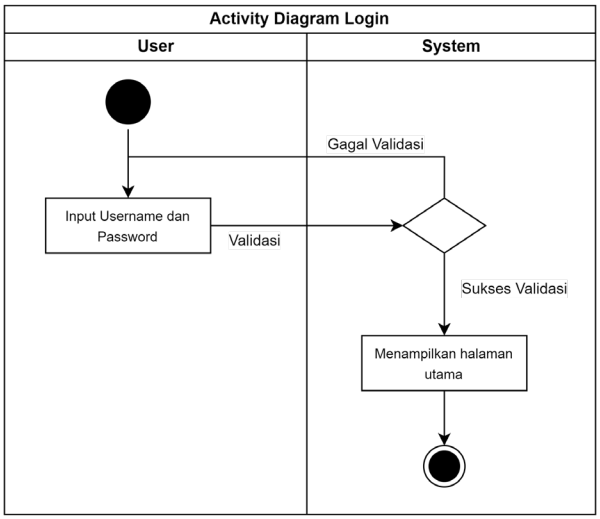
Pada tahap perancangan sistem ini yang akan menjadi penentu proses suatu sistem akan berjalan dengan melakukan penggambaran, pembuatan ilustrasi dan perencanaan dari beberapa bagian seperti struktur sistem, *database, user interface* dan elemen lain yang dibutuhkan dalam kelengkapan fungsi [7]. Perancangan sistem akan dibuat dengan menggunakan UML (*Unified Modeling Language*). *Unified Modeling Language* dibuat sebagai penyedia perangkat yang dibutuhkan oleh pengembang perangkat lunak dalam melakukan analisis, perancangan dan implementasi sistem berbasis perangkat lunak [8]. Pertama yaitu *use case diagram* yang digunakan untuk menggambarkan dan juga pemodelan setiap interaksi atau kelakuan dari sistem [9].



Gambar. 2. Use Case Diagram

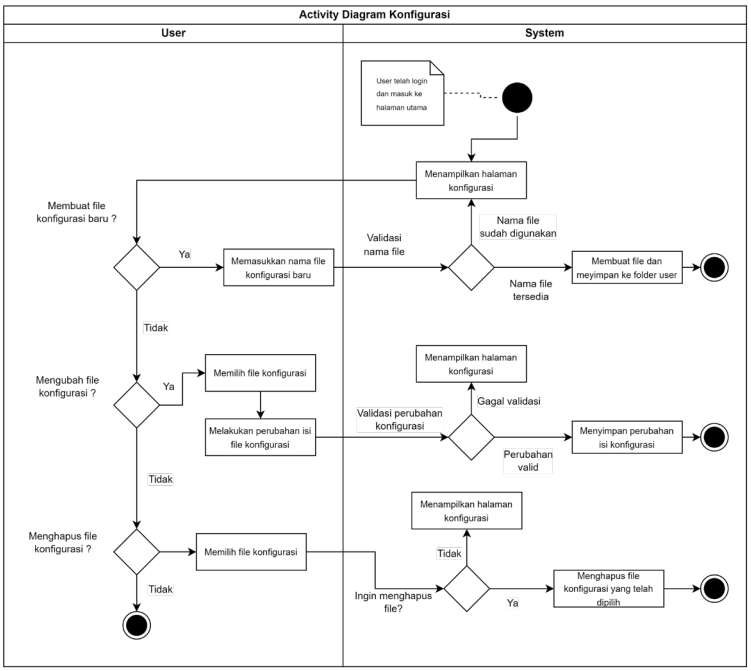
Berdasarkan Gambar 2 menggambarkan bahwa ada dua jenis pengguna yaitu pengguna sebagai *admin* maupun juga pengguna sebagai *end user* yang dimana masing – masing mempunyai perbedaan aktivitas pada sistem. Pada pengguna *admin* dapat melakukan CRUD *file* konfigurasi dan dapat membuat *output report* berdasarkan konfigurasi yang telah dipilih. Pengguna *admin* juga dapat melihat *Log System* yang akan berisi riwayat jika terjadi masalah atau ada aktivitas pada sistem seperti *error*, membuat *output* dan lainnya. Sedangkan untuk pengguna *end user* hanya dapat mengakses *output* yang sudah dibuat. Selanjunya yaitu *activity diagram* yang digunakan untuk menggambarkan aktivitas atau alur kerja dari sebuah sistem berdasarkan menu atau fitur yang diberikan [9].

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492



Gambar. 3. Activity Diagram – Login

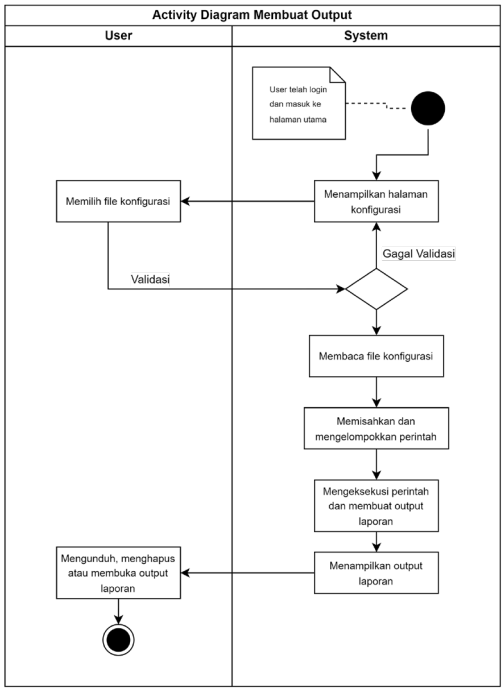
Pada Gambar 3 merupakan *activity diagram login*. *User* memasukkan *username* dan *password* dan kemudian akan divalidasi oleh sistem. Jika *username* dan *password* sesuai, maka sistem akan menampilkan halaman utama. Apabila validasi gagal, maka akan diarahkan kembali ke *input username* dan *password*.



Gambar. 4. Activity Diagram – Konfigurasi

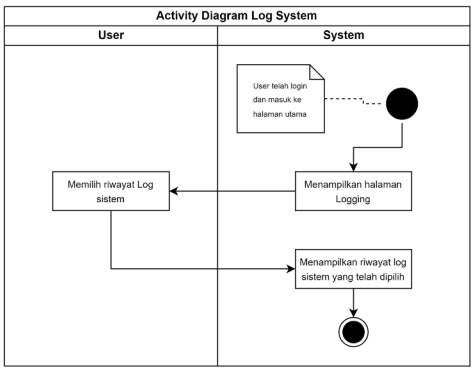
Pada Gambar 4 merupakan *activity diagram konfigurasi*. Setelah *user* telah melakukan *login* dan sistem menampilkan halaman utama, selanjutnya membuat *file* konfigurasi baru dengan memasukkan nama *file* baru. Setelah itu sistem akan divalidasi nama *file* tersebut sudah digunakan atau belum. Jika sudah maka akan diarahkan kembali ke pembuatan *file* baru dan memasukkan nama kembali dan jika nama *file* belum digunakan maka sistem akan membuat dan menyimpan *file* ke dalam *folder user* tersebut. Selanjutnya *user* melakukan perubahan *file* konfigurasi berdasarkan *file* yang telah dipilih, kemudian *user* mengubah isi dari konfigurasi dan sistem akan melakukan pengecekan perubahan konfigurasi. Jika valid maka perubahan akan disimpan dan jika tidak maka akan kembali ke halaman konfigurasi. Kemudian jika *user* memilih untuk menghapus *file* konfigurasi, maka *user* harus memilih terlebih dahulu *file* yang dingin dihapus dan kemudian sistem akan meminta konfirmasi penghapusan *file*. Jika setuju maka sistem akan menghapus *file* tersebut dan jika tidak sistem akan mengarahkan kembali ke halaman utama.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492



Gambar. 5. Activity Diagram - Output Report

Pada Gambar 5 merupakan *activity diagram* membuat *output* laporan. *User* telah melakukan *login* dan sistem akan menampilkan halaman konfigurasi. Selanjutnya *user* akan memillih *file* konfigurasi yang akan digunakan dalam pembuatan laporan dan akan validasi *file* tersebut oleh sistem. Sistem kemudian akan memisahkan dan mengelompokkan setiap perintah berdasarkan masing – masing fungsi. Berdasarkan perintah yang sudah dipisah akan dieksekusi untuk menjalankan proses pembuatan laporan mulai dari pengambilan data hingga penulisan *output* dengan jenis *file* yang telah ditentukan. *Output* yang telah jadi akan disimpan ke dalam *folder user* yang membuat *output* tersebut. Hasilnya dapat diunduh, dihapus maupun dibuka oleh semua *user* termasuk *end user*.



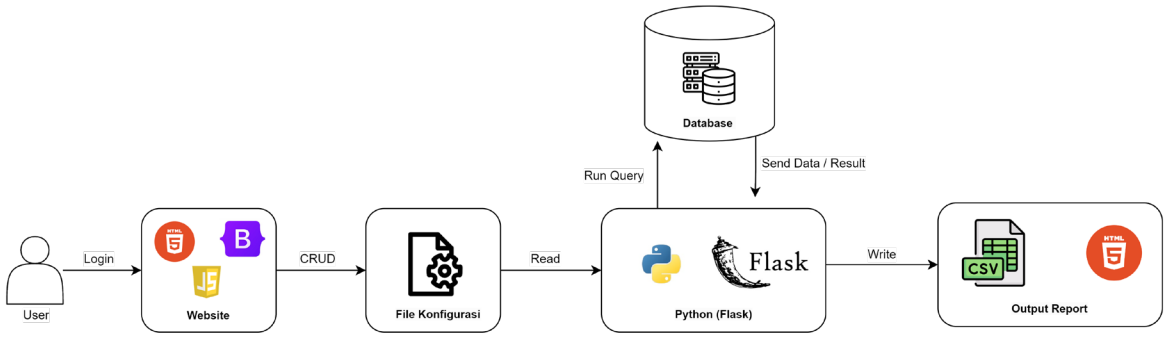
Gambar. 6. Activity Diagram - Log System

Pada Gambar 6 merupakan *activity diagram* untuk melihat *log system*. *Log* sistem ini berisi riwayat aktivitas yang terjadi pada sistem mulai dari *error*, pembuatan konfigurasi hingga ke pembuatan laporan oleh *user* yang sedang *login* saat itu juga. *User* dapat memilih riwayat *log* sesuai dengan yang tersedia pada tampilan. Setelah memilih, sistem akan menampilkan riwayat *log* pada tanggal yang telah dipilih. Tujuan lain dari *log* sistem ini digunakan untuk *monitoring* kegiatan *user* selama berada di sistem. Hal ini dapat mencegah pengguna yang akan selalu memiliki peluang membuat, memodifikasi, menghilangkan data atau informasi pada sistem secara tidak bertanggung jawab [10].

Masuk ke tahap selanjutnya yaitu tahap pembuatan sistem. Sistem dibuat dengan menggunakan *Python Flask* untuk melakukan proses pembacaan konfigurasi hingga pembuatan *output*. Tampilan berbasis *web* yang dibangun dengan menggunakan *HTML, Bootstrap* dan *Javascript*. *Bootstrap* merupakan *framework* desain yang biasa digunakan pada *web* dengan fitur tambahan dan dibuat untuk menyederhanakan proses desain situs *web* dari semua tingkatan, mulai dari pemula hingga ke berpengalaman. Dengan *bootstrap* juga dapat membangun sebuah *website* menjadi lebih *responsive* tanpa membuatnya dari awal karena telah disediakan oleh *bootstrap* [11]. Kemudian

penggunaan *database SQLite* untuk menyimpan data *user*. *SQLite* adalah basis data yang menerapkan sifat mandiri, tanpa konfigurasi, tanpa server dan juga mesin transaksional. *SQLite* juga merupakan basis data yang salah satunya paling banyak digunakan di dunia karena mendukung jenis data seperti teks, bilangan bulat, dan *real* seperti yanag dimiliki di bahasa pemrograman *Java* [12]. Lanjut ke tahap pengujian yang dimana pada tahap ini, pengujian sistem akan menggunakan metode *Black Box Testing* dan metode *SUS* (*System Usability Scale*). Metode *Black Box Testing* yaitu metode yang melakukan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan [13]. Cara ini juga cukup dengan mengamati masukkan dan keluaran dari sistem aplikasi yang dibangun tanpa harus memahami desain kontrol dalam aplikasi [7]. Dan metode SUS merupakan pemberian nilai yang sifatnya global terhadap aspek *usability* seperti efektivitas, efisiensi, dan kepuasaan secara subjektif yang dirasakan oleh pengguna [14].

III. HASIL DAN PEMBAHASAN



Gambar. 7. Desain Arsitektur Sistem

Gambar 2 merupakan gambaran mengenai arsitektur sistem yang akan dibuat. Dimulai dari pengguna (*user*) *login* terlebih dahulu di *website* yang dijadikan sebagai tampilan sistem yang dibentuk dengan menggunakan *HTML*, *Javascript* dan *Bootstrap*. Setelah *login* pengguna bisa membuat, mengubah, membaca, atau menghapus *file* konfigurasi. Jika pengguna telah memilih *file* konfigurasi, maka pengguna dapat membuat output dan nantinya konfigurasi akan dibaca oleh sistem menggunakan *Python* (*Flask*). Dari *flask* akan membaca dan menjalankan setiap perintah yang diberikan di konfigurasi dan akan diproses lebih lanjut untuk menghasilkan *output* yang berupa *CSV* atau *HTML* sesuai yang diberikan di konfigurasi.

Untuk perancangan *domain specific language* berdasarkan kebutuhan pada pembuatan *reporting* seperti pengambilan data, pemrosesan data, format data dan juga jenis *report* yang akan dihasilkan. Proses ini akan diuraikan ke dalam perintah – perintah yang ditulis ke dalam sebuah *file* konfigurasi dan kemudian akan dibaca oleh sistem. *File* konfigurasi merupakan berkas kode yang digunakan untuk melakukan pemilihan berbagai fitur dan pengaturan seperti menentukan preferensi, parameter, maupun opsi alternatif lainnya [15]. Para pengembang biasanya menggunakan *configuration file* dalam mengatur *parameter* dan kondisi awal untuk sistem komputer mereka [16].

Kemudian terdapat perintah yang sifatnya *mandatory* (wajib) dan *optional* (tidak wajib) tergantung pada kondisi konfigurasi yang telah diatur dan kebutuhan pengguna.

TABEL I
PERANCANGAN GRAMMAR DOMAIN SPECIFIC LANGUAGE

```
DB = nama_database
SELECT = select column1, column2, column3
 From table_name
PARAM_SOURCE = [int]paramName
OUTPUT = html
TITLENAME = LAPORAN PEGAWAI PT. TESTING JAYA
USER_ACCESS = user01, user02
[FORMATTING]
  "column1" = color:red;
  "column2" = color:green; number:curr; total:true;
```

Pada Tabel I merupakan format penulisan *domain specific language*. Perintah – perintah dibuat berdasarkan kebutuhan dalam pembuatan laporan seperti *DB* yang akan berisi nama *database* yang akan diakses, *SELECT* merupakan *select statement* yang akan dieksekusi sebagai *query database* untuk pengambilan data, *PARAM_SOURCE* merupakan parameter yang dibutuhkan pada perintah *select, OUTPUT* sebagai jenis *file* keluaran laporan berupa *HTML* atau *CSV*, *TITLENAME* akan menjadi judul pada laporan, *USER_ACCESS* berisi

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

nama pengguna yang bisa mengakses *file* konfigurasi, *FORMATTING* sebagai format tabel dan hanya jika dibutuhkan. Untuk perintah yang sifatnya *optional* yaitu *PARAM_SOURCE, USER_ACCESS* dan juga *FORMATTING*.

Format penulisan *domain specific language* telah disusun dengan aturan penulisan sebagai berikut:

1) Letak perintah tidak harus urut sehingga posisi perintah dapat disesuaikan dengan keinginan, contohnya seperti *select* ditulis terlebih dahulu kemudian lanjutkan *db* dan seterusnya.
2) Untuk *multiline select* harus dimulai dengan satu spasi agar sistem dapat mendeteksi bahwa baris selanjutnya merupakan lanjutan dari perintah *select* sebelumnya. *Select* disini sama dengan *select statement* yang biasa digunakan pada *query database* umum.
3) Penulisan *param_source* harus sesuai format yaitu [tipe_data]nama_param. Nama *param* harus disesuaikan dengan *parameter* yang dibutuhkan pada *select statement* untuk menghindari *error*.
4) Penulisan *user_access* diisi dengan *username* pengguna yang ingin ditambahkan aksesnya ke dalam file konfigurasi.
5) Penulisan *formatting* harus sesuai dengan aturan penulisan yaitu "nama_kolom" = jenis_format:nilai. Dan jika format yang dibutuhkan lebih dari satu dapat dipisah dengan menggunakan simbol titik koma. Untuk format yang disediakan yaitu *color, number* dan total. *Color* merupakan warna yang ingin digunakan pada kolom tersebut dan dapat ditulis dengan menggunakan kata maupun *hexadecimal*. *Number* merupakan format data *numeric* yang ditampilkan dan hanya bisa menggunakan format mata uang (*currency*). Total digunakan jika membutuhkan akumulasi total pada kolom tersebut.

TABEL II
KODE PROGRAM FUNGSI SPLIT_FILE_CONFIG

Kode Program

```
def split_file_config(config_data):
    result = {}
    try:
        for data in config_data:
            if ' ' == data[0]:
                try:
                    update_select = result.get('select') + data
                    result.update({'select': update_select})
                except Exception as e:
                    log(f'Error filehelper.splitConfig, Error menggabungkan select statement :
({e})')
            elif '[formatting]' in data.replace(' ', '').strip().lower():
                result[data.replace(' ', '').strip().lower()] = {}
            elif '\t' == data[0]:
                data_split = data.split('=')
                key = data_split[0].strip().replace('"', '')
                result['[formatting]'].update({key : {}})

                format_split = data_split[1].strip().replace(' ', '').split(';')
                fix_split = list(filter(None, format_split))

                for format in fix_split:
                    temp = format.split(':')
                    result['[formatting]'][key].update({
                        temp[0]: temp[1]
                    })
            else:
                data_split = data.split('=')
                key = data_split[0].strip().lower()
                result[key] = data_split[1]

    except Exception as e:
        log(f'Error filehelper.splitConfig, Terdapat error di file config : ({e})')
    return result
```

Pada Tabel II merupakan fungsi *split_file_config* yang digunakan untuk memisahkan dan mengelomppokkan kode – kode atau perintah yang terdapat di *file* konfigurasi dengan acuan pemisah yaitu tanda sama dengan (=). Pada variabel *result* merupakan variabel yang digunakan untuk menampung konfigurasi yang telah dipisahkan. Pemisahan perintah *select* dilakukan dengan mengecek apakah *index* pertama pada data konfigurasi merupakan spasi atau bukan. Jika spasi maka baris tersebut merupakan lanjutan dari *select* sebelumnya dan diubah. Pemisahan juga dilakukan pada perintah *formatting* untuk mengambil nilai dari format yang dituliskan.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

TABEL III
KODE PROGRAM PEMBUATAN OUTPUT CSV

Kode Program

```
with xlsxwriter.Workbook(file_name) as workbook:
            worksheet = workbook.add_worksheet()
            title_format = workbook.add_format({'valign':'vcenter','align': 'left', 'bold':
True })
            header_table_format = workbook.add_format({'bold': True,'border': 2,'bg_color':
'#0051de','font_color': 'white'})
            table_format = workbook.add_format({'border': 2,'align': 'left'})
            worksheet.merge_range(0, 0, 1, len(data_result[0].keys())-1, f'{title_name}', ti-
tle_format)
            worksheet.merge_range(2, 0, 4, len(data_result[0].keys())-1, f'{format_date_re-
port}', title_format)
            row = 5
            for column_index, column_name in enumerate(data_result[0].keys()):
                worksheet.write(row, column_index, column_name.upper(), header_table_format)
            row = 6
            sum_total = dict()
            number_format = list()
            for row_data in data_result:
                for column_index,(column_name, column_value) in enumerate(row_data.items()):
                    type_format = output_format.get(column_name, {}) if output_format else {}
                    temp_data = temp_format = None
                    if type_format:
                        if type_format.get('number', '') == 'curr':
                            temp_data = locale.currency(column_value, grouping=True)
                            number_format.append(column_name)
                        if type_format.get('color', ''):
                            temp_format = workbook.add_format({'font_color': type_for-
mat.get('color', 'black'), 'border': 2,'align': 'left'})
                        if type_format.get('total', 'false') == 'true':
                            sum_total[column_name] = sum_total.get(column_name, 0) + col-
umn_value
                    worksheet.write(row, column_index, temp_data if temp_data else col-
umn_value, temp_format if temp_format else table_format)
                row += 1
            row = 5
            last_column = worksheet.dim_colmax + 2
            for sum_name, sum_value in sum_total.items():
                worksheet.write(row, last_column, sum_name, header_table_format)
                if sum_name in number_format:
                    worksheet.write(row, last_column+1, locale.currency(sum_value, group-
ing=True), table_format)
                else:
                    worksheet.write(row, last_column+1, sum_value, table_format)
                row += 1
            worksheet.autofit()
```

Pada Tabel III merupakan kode program yang digunakan untuk membaca *dictionary* hasil pengelompokkan perintah *file* konfigurasi dan juga membuat *output* laporan dengan tipe *file CSV*. Variabel *title_format*, *header_table_format* dan *table_format* merupakan format *default* dari tabel yang akan dibuat. Pada *merge_range* pertama dan kedua digunakan untuk mencetak judul dan tanggal pada *output*. Kemudian pada perulangan *for* pertama digunakan untuk mencetak *header* tabel yang berisi nama kolom. Pada perulangan *for* kedua digunakan untuk mencetak data utama pada tabel. Di dalam perulangan kedua terdapat pembacaan *format* jika terdapat *format* yang dituliskan pada konfigurasi. Dan pada perulangan *for* terakhir digunakan untuk mencetak informasi tambahan seperti total nilai pada kolom yang telah ditentukan.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

TABEL IV
KODE PROGRAM PEMBUATAN OUTPUT HTML

Kode Program

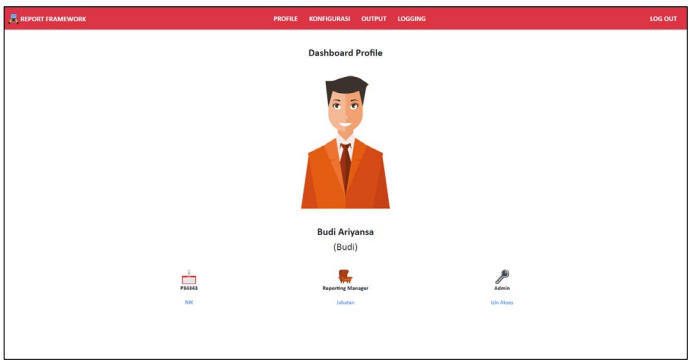
```
with open(file_name, "w") as file:
            html_container     =     f'<h2     style="padding:     10px;     margin:0;"><b>{ti-
tle_name}</b></h2>\n'
            html_container   +=   f'<h3   style="margin:0;padding:   10px;"><b>{format_date_re-
port}</b></h3>\n'
            html_container   +=   '<table><tr><td>   table-1   </td><td   valign="top">   table-2
</td></tr></table>'
            html_container += '<style>*{font-family:"Calibri"}</style>'

            first_table = '<table border="1">'
            first_table += '<tr>'
            for column_name in data_result[0].keys():
                first_table   +=   f'<th   style="background-color:#0051de;   color:white;">{col-
umn_name.upper()}</th>'
            first_table += '</tr>'

            sum_total = dict()
            number_format = list()
            for row_data in data_result:
                first_table += '<tr>'
                for column_name, column_value in row_data.items():
                    type_format = output_format.get(column_name, {}) if output_format else {}
                    temp_data = temp_format = None
                    if type_format:
                        if type_format.get('number', '') == 'curr':
                            temp_data = locale.currency(column_value, grouping=True)
                            number_format.append(column_name)
                        if type_format.get('color', ''):
                            color = type_format.get('color', 'black')
                            temp_format = f'color: {color}'
                        if type_format.get('total', 'false') == 'true':
                            sum_total[column_name]   =   sum_total.get(column_name,   0)   +   col-
umn_value
                    first_table    +=    f'<td    style="{temp_format    if    temp_format    else
""}">{temp_data if temp_data else column_value}</td>'
                first_table += '</tr>'
            first_table += '</table>'

            second_table = '<table border="1">'
            for sum_name, sum_value in sum_total.items():
                second_table += '<tr>'
                if sum_name in number_format:
                    second_table        +=        f'<td        style="background-color:#0051de;
color:white;"><b>{sum_name.upper()}</b></td>'
                    second_table += f'<td>{locale.currency(sum_value, grouping=True)}</td>'
                second_table += '</tr>'
            second_table += '</table>'
            html_container = html_container.replace('table-1', first_table)
            html_container = html_container.replace('table-2', second_table)
            file.write(html_container)
```

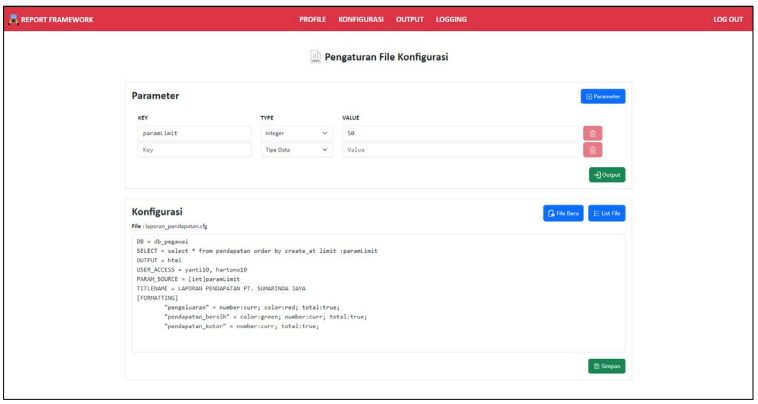
Pada Tabel IV memiliki kegunaan yang sama dengan Tabel III yaitu digunakan untuk membuat *output* laporan namun dengan tipe *file* yang berbeda yaitu *HTML*.



Gambar. 8. Tampilan Profile

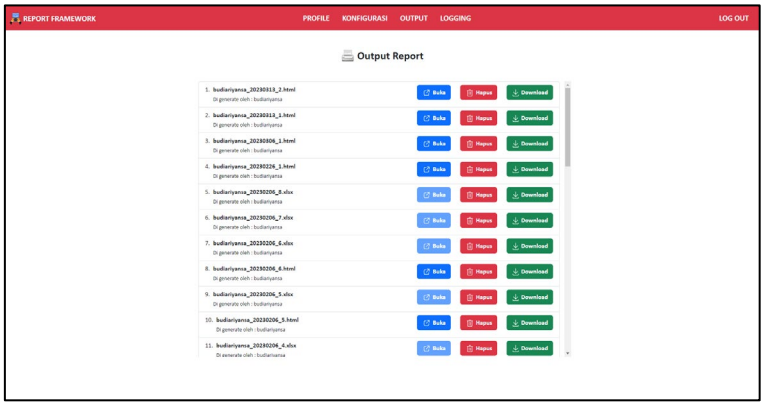
Pada Gambar 8 merupakan tampilan profile dan menjadi halam utama setelah *login* dilakukan oleh *user*.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492



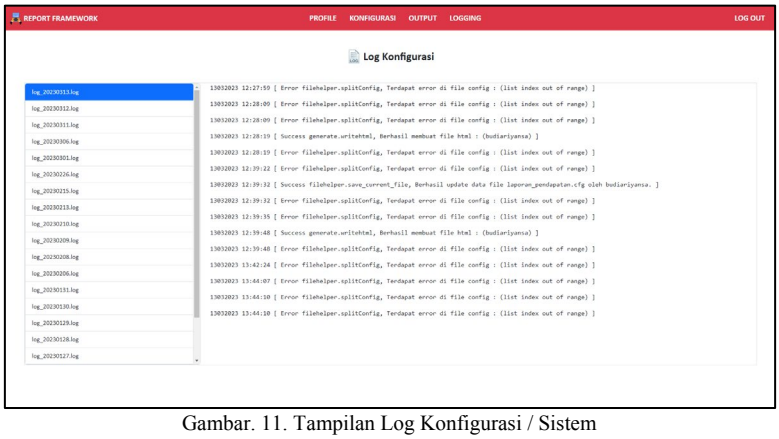
Gambar. 9. Tampilan Konfigurasi

Pada Gambar 9 merupakan tampilan halaman konfigurasi. Halaman ini adalah tempat *user* membuat *file* konfigurasi, membaca, menghapus, mengubah isi konfigurasi dan juga membuat *file output report* berdasarkan konfigurasi yang sedang dibuka. *Parameter* yang dibutuhkan juga dikirim melalui halaman ini yaitu di bagian *input field parameter*.



Gambar. 10. Tampilan Output

Pada Gambar 10 merupakan tampilan *output* yang memberikan *list* dari semua laporan yang telah dibuat dari konfigurasi. *File output* bisa diunduh, dihapus dan juga dibuka. *File output* yang bisa dibuka secara langsung hanya *file HTML*. Setiap *user* dapat menghapus *output* yang telah dibuat.



Gambar. 11. Tampilan Log Konfigurasi / Sistem

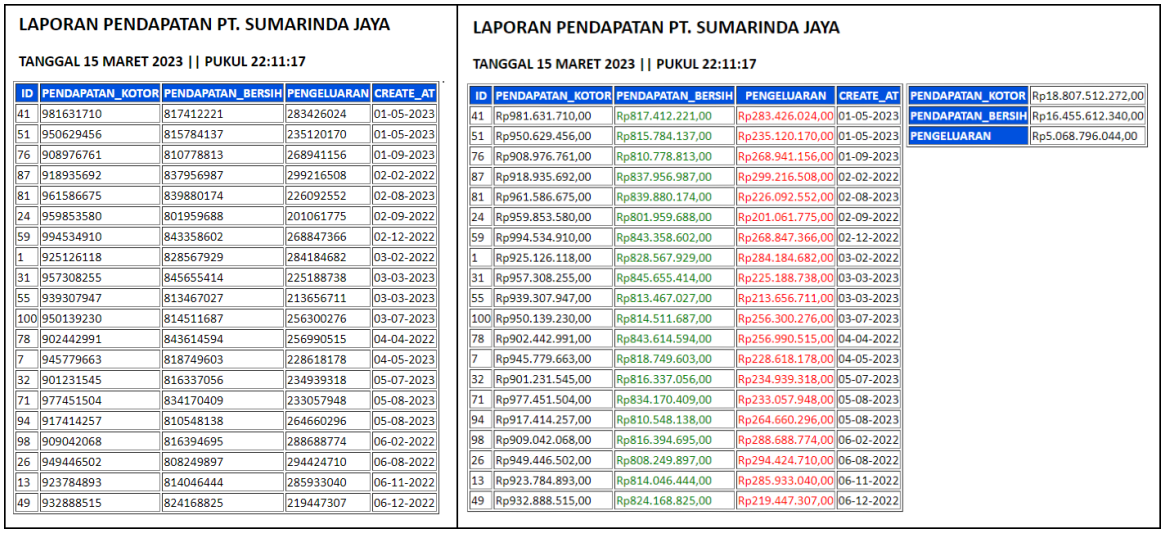
Pada Gambar 11 merupakan tampilan *Log* konfigurasi atau sistem. *Log* ini berisi informasi *error* maupun aktivitas yang dilakukan oleh *user* seperti membuat *file* konfigurasi baru hingga ke proses pembuatan *output* laporan. *Log* yang ditampilkan hanya mulai dari 20 hari sebelum tanggal hari ini.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492

TABEL V
CONTOH KONFIGURASI REPORTING

```
DB = db_pegawai
SELECT = select * from pendapatan order by create_at
 limit :paramLimit
OUTPUT = html
USER_ACCESS = yanti10, hartono10
PARAM_SOURCE = [int]paramLimit
TITLENAME = LAPORAN PENDAPATAN PT. SUMARINDA JAYA
[FORMATTING]
  "pengeluaran" = number:curr; color:red; total:true;
  "pendapatan_bersih" = color:green; number:curr; total:true;
  "pendapatan_kotor" = number:curr; total:true;
```

Pada Tabel II merupakan contoh penulisan konfigurasi yang digunakan dalam membuat *file report* nantinya. Dan hasil dari konfigurasi ini dapat dilihat pada Gambar 12 dan Gambar 13



LAPORAN PENDAPATAN PT. SUMARINDA JAYA

TANGGAL 15 MARET 2023 || PUKUL 22:11:17

ID	PENDAPATAN_KOTOR	PENDAPATAN_BERSIH	PENGELUARAN	CREATE_AT
41	981631710	817412221	283426024	01-05-2023
51	950629456	815784137	235120170	01-05-2023
76	908976761	810778813	268941156	01-09-2023
87	918935692	837956987	299216508	02-02-2022
81	961586675	839880174	226092552	02-08-2023
24	959853580	801959688	201061775	02-09-2022
59	994534910	843358602	268847366	02-12-2022
1	925126118	828567929	284184682	03-02-2022
31	957308255	845655414	225188738	03-03-2023
55	939307947	813467027	213656711	03-03-2023
100	950139230	814511687	256300276	03-07-2023
78	902442991	843614594	256990515	04-04-2022
7	945779663	818749603	228618178	04-05-2023
32	901231545	816337056	234939318	05-07-2023
71	977451504	834170409	233057948	05-08-2023
94	917414257	810548138	264660296	05-08-2023
98	909042068	816394695	288688774	06-02-2022
26	949446502	808249897	294424710	06-08-2022
13	923784893	814046444	285933040	06-11-2022
49	932888515	824168825	219447307	06-12-2022

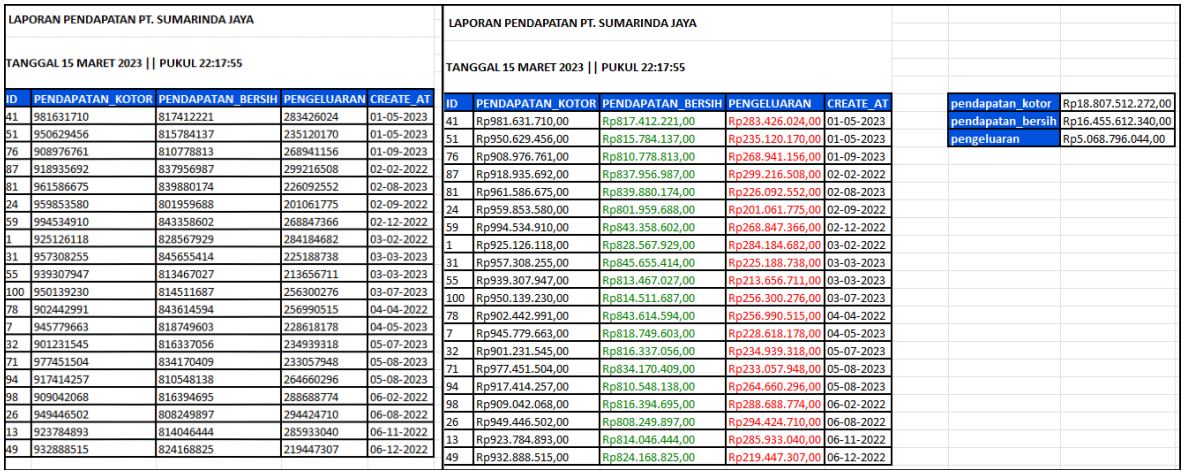
LAPORAN PENDAPATAN PT. SUMARINDA JAYA

TANGGAL 15 MARET 2023 || PUKUL 22:11:17

ID	PENDAPATAN_KOTOR	PENDAPATAN_BERSIH	PENGELUARAN	CREATE_AT
41	Rp981.631.710,00	Rp817.412.221,00	Rp283.426.024,00	01-05-2023
51	Rp950.629.456,00	Rp815.784.137,00	Rp235.120.170,00	01-05-2023
76	Rp908.976.761,00	Rp810.778.813,00	Rp268.941.156,00	01-09-2023
87	Rp918.935.692,00	Rp837.956.987,00	Rp299.216.508,00	02-02-2022
81	Rp961.586.675,00	Rp839.880.174,00	Rp226.092.552,00	02-08-2023
24	Rp959.853.580,00	Rp801.959.688,00	Rp201.061.775,00	02-09-2022
59	Rp994.534.910,00	Rp843.358.602,00	Rp268.847.366,00	02-12-2022
1	Rp925.126.118,00	Rp828.567.929,00	Rp284.184.682,00	03-02-2022
31	Rp957.308.255,00	Rp845.655.414,00	Rp225.188.738,00	03-03-2023
55	Rp939.307.947,00	Rp813.467.027,00	Rp213.656.711,00	03-03-2023
100	Rp950.139.230,00	Rp814.511.687,00	Rp256.300.276,00	03-07-2023
78	Rp902.442.991,00	Rp843.614.594,00	Rp256.990.515,00	04-04-2022
7	Rp945.779.663,00	Rp818.749.603,00	Rp228.618.178,00	04-05-2023
32	Rp901.231.545,00	Rp816.337.056,00	Rp234.939.318,00	05-07-2023
71	Rp977.451.504,00	Rp834.170.409,00	Rp233.057.948,00	05-08-2023
94	Rp917.414.257,00	Rp810.548.138,00	Rp264.660.296,00	05-08-2023
98	Rp909.042.068,00	Rp816.394.695,00	Rp288.688.774,00	06-02-2022
26	Rp949.446.502,00	Rp808.249.897,00	Rp294.424.710,00	06-08-2022
13	Rp923.784.893,00	Rp814.046.444,00	Rp285.933.040,00	06-11-2022
49	Rp932.888.515,00	Rp824.168.825,00	Rp219.447.307,00	06-12-2022

PENDAPATAN_KOTOR	Rp18.807.512.272,00
PENDAPATAN_BERSIH	Rp16.455.612.340,00
PENGELUARAN	Rp5.068.796.044,00

Gambar. 12. Hasil Output Report HTML



LAPORAN PENDAPATAN PT. SUMARINDA JAYA

TANGGAL 15 MARET 2023 || PUKUL 22:17:55

ID	PENDAPATAN_KOTOR	PENDAPATAN_BERSIH	PENGELUARAN	CREATE_AT
41	981631710	817412221	283426024	01-05-2023
51	950629456	815784137	235120170	01-05-2023
76	908976761	810778813	268941156	01-09-2023
87	918935692	837956987	299216508	02-02-2022
81	961586675	839880174	226092552	02-08-2023
24	959853580	801959688	201061775	02-09-2022
59	994534910	843358602	268847366	02-12-2022
1	925126118	828567929	284184682	03-02-2022
31	957308255	845655414	225188738	03-03-2023
55	939307947	813467027	213656711	03-03-2023
100	950139230	814511687	256300276	03-07-2023
78	902442991	843614594	256990515	04-04-2022
7	945779663	818749603	228618178	04-05-2023
32	901231545	816337056	234939318	05-07-2023
71	977451504	834170409	233057948	05-08-2023
94	917414257	810548138	264660296	05-08-2023
98	909042068	816394695	288688774	06-02-2022
26	949446502	808249897	294424710	06-08-2022
13	923784893	814046444	285933040	06-11-2022
49	932888515	824168825	219447307	06-12-2022

LAPORAN PENDAPATAN PT. SUMARINDA JAYA

TANGGAL 15 MARET 2023 || PUKUL 22:17:55

ID	PENDAPATAN_KOTOR	PENDAPATAN_BERSIH	PENGELUARAN	CREATE_AT
41	Rp981.631.710,00	Rp817.412.221,00	Rp283.426.024,00	01-05-2023
51	Rp950.629.456,00	Rp815.784.137,00	Rp235.120.170,00	01-05-2023
76	Rp908.976.761,00	Rp810.778.813,00	Rp268.941.156,00	01-09-2023
87	Rp918.935.692,00	Rp837.956.987,00	Rp299.216.508,00	02-02-2022
81	Rp961.586.675,00	Rp839.880.174,00	Rp226.092.552,00	02-08-2023
24	Rp959.853.580,00	Rp801.959.688,00	Rp201.061.775,00	02-09-2022
59	Rp994.534.910,00	Rp843.358.602,00	Rp268.847.366,00	02-12-2022
1	Rp925.126.118,00	Rp828.567.929,00	Rp284.184.682,00	03-02-2022
31	Rp957.308.255,00	Rp845.655.414,00	Rp225.188.738,00	03-03-2023
55	Rp939.307.947,00	Rp813.467.027,00	Rp213.656.711,00	03-03-2023
100	Rp950.139.230,00	Rp814.511.687,00	Rp256.300.276,00	03-07-2023
78	Rp902.442.991,00	Rp843.614.594,00	Rp256.990.515,00	04-04-2022
7	Rp945.779.663,00	Rp818.749.603,00	Rp228.618.178,00	04-05-2023
32	Rp901.231.545,00	Rp816.337.056,00	Rp234.939.318,00	05-07-2023
71	Rp977.451.504,00	Rp834.170.409,00	Rp233.057.948,00	05-08-2023
94	Rp917.414.257,00	Rp810.548.138,00	Rp264.660.296,00	05-08-2023
98	Rp909.042.068,00	Rp816.394.695,00	Rp288.688.774,00	06-02-2022
26	Rp949.446.502,00	Rp808.249.897,00	Rp294.424.710,00	06-08-2022
13	Rp923.784.893,00	Rp814.046.444,00	Rp285.933.040,00	06-11-2022
49	Rp932.888.515,00	Rp824.168.825,00	Rp219.447.307,00	06-12-2022

pendapatan_kotor	Rp18.807.512.272,00
pendapatan_bersih	Rp16.455.612.340,00
pengeluaran	Rp5.068.796.044,00

Gambar. 13. Hasil Output Report CSV

Pada Gambar 12 dan Gambar 13 merupakan hasil *output report* yang dibuat dengan konfigurasi. Pada Gambar 12 maupun Gambar 13 juga menunjukkan perbedaan hasil jika menggunakan *formatting* dan tidak. Terlihat perbedaan pada kolom yang diberikan format, ada yang berwarna merah dan hijau dan juga setiap kolom ditampilkan sebagai mata uang. Pemberian total juga ditambahkan untuk setiap kolom yang dituliskan. Dengan demikian pembuatan dan pemberian format pada laporan akan lebih baik dan lebih cepat karena pengambilan data, pemberian format dan juga penghitungan total pada kolom yang dipilih tidak dilakukan secara manual namun secara otomatis oleh sistem.

TABEL VI
TABEL PENGUJIAN BLACK BOX

No	Langkah Percobaan	Hasil Pengujian Yang Diharapkan	Keputusan Akhir
1	*User* mengisi *username* dan *password*	*Login* berhasil dan masuk ke halaman utama	Berhasil
2	*User* membuat *file* konfigurasi baru	*File* konfigurasi berhasil dibuat dan akan disimpan ke dalam *folder user* tersebut dengan nama *file* yang berbeda dengan *file* lainnya.	Berhasil
3	*User* membuka *file* konfigurasi yang dipilih	*File* konfigurasi yang dipilih akan dibuka dan ditampilkan datanya pada *website*	Berhasil
4	*User* mengubah isi *file* konfigurasi yang dipilih	*File* konfigurasi yang dipilih dan diubah isinya akan disimpan jika valid dan isi terbaru akan ditampilkan pada *website*	Berhasil
5	*User* mengubah isi *file* konfigurasi dengan menambahkan *user access*	*File* konfigurasi yang diubah akan bisa diakses oleh *user* yang telah dituliskan pada *user access* tersebut	Berhasil
6	*User* menghapus *file* konfigurasi yang dipilih	*File* konfigurasi yang dipilih akan dihapus dari *folder user* tersebut	Berhasil
7	*User* membuat *output* laporan dengan tanpa perintah yang sifatnya optional	Berhasil membuat *output* laporan dengan *file* konfigurasi yang telah dipilih dan file output disimpan ke *folder output user* tersebut	Berhasil
8	*User* membuat *output* laporan dengan penambahan perintah *formatting*	Berhasil membuat *output* laporan dengan *format* yang telah diberikan dari *file* konfigurasi yang telah dipilih dan *file output* disimpan ke *folder output user* tersebut	Berhasil
9	*User* mengunduh *output* laporan	*File* laporan akan langsung diunduh.	Berhasil
10	*User* menghapus *output* laporan	*File* laporan akan dihapus dari *folder user* yang telah membuat *output* tersebut	Berhasil
11	*User* memilih dan membuka *Log* sistem	Muncul *Log* sistem berdasarkan *Log* yang telah dipilih.	Berhasil

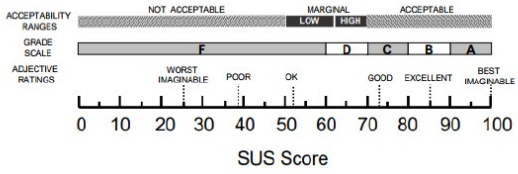
Tabel III merupakan hasil pengujian sistem dengan 11 langkah pengujian yang dilakukan oleh pihak pengembang sebagai pengguna. Dan berdasarkan hasil 11 langkah tersebut dapat berjalan sukses dan sesuai dengan harapan. Dan tahap selanjutnya yaitu pengujian dengan metode *System Usability Scale*. Pada tabel VII merupakan beberapa pertanyaan yang akan digunakan pada penelitian ini.

TABEL VII
PERTANYAAN METODE SYSTEM USABILITY SCALE (SUS)

No	Pertanyaan
1	Saya berpikir akan menggunakan sistem ini lagi
2	Saya merasa sistem ini rumit untuk digunakan
3	Saya merasa sistem ini mudah digunakan
4	Saya membutuhkan bantuan orang lain atau teknisi dalam menggunakan sistem ini
5	Saya merasa fitur – fitur sistem ini berjalan dengan semestinya
6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini)
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat
8	Saya merasa sistem ini mebingungkan
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini

Terdapat tiga buah sudut pandang pada metode *System Usability Scale* yaitu *acceptability, grade, adjective rating*. Sudut pandang *acceptability* dengan tiga tingkatan yaitu *not acceptable, marginal (low and high),* dan *acceptable*. Kemudian pada *grade* memiliki lima tingkatan A, B, C, D, dan F. Lanjut yang terakhir yaitu *adjective rating* memiliki banyak tingkatan meliputi *worst imaginable, poor, ok, good, excellent,* dan *best imaginable* [17]. Untuk interpretasi nilai *system usability scale* dapat dilihat pada Gambar 14.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 4, Desember 2023, Pp. 1479-1492



Gambar. 14. Interpretasi Nilai System Usability Scale [17]

TABEL VIII
RATA - RATA HASIL SYSTEM USABILITY SCALE (SUS)

Responden	Usia	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai (Jumlah x 2,5
R1	21 Tahun	5	2	4	3	5	2	4	2	4	3	34	85
R2	21 Tahun	5	1	5	1	5	1	5	1	5	1	30	75
R3	22 Tahun	4	2	4	3	5	2	2	1	2	5	30	75
R4	21 Tahun	3	2	3	4	4	3	2	5	4	5	35	87,5
R5	22 Tahun	5	1	5	1	5	2	4	1	5	1	30	75
R6	22 Tahun	4	2	4	3	4	4	3	4	4	4	36	90
R7	22 Tahun	4	2	4	3	4	2	4	2	4	4	33	82,5
R8	21 Tahun	4	2	4	3	4	2	4	2	2	4	31	77,5
R9	22 Tahun	5	1	5	1	5	1	5	1	5	1	30	75
R10	21 Tahun	5	2	4	3	5	2	4	2	4	3	34	85
											Rata - Rata		80,75

Berdasarkan hasil kuesioner yang telah disebar kepada beberapa responden yaitu pihak pengembang lainnya sebagai pengguna dan dapat dianalisis maupun dilakukan perhitungan terhadap jawaban yang telah diterima berdasarkan ketentuan SUS. Skor yang dihasilkan yaitu sebesar 80,75 sehingga masuk ke kategori *good* dengan *grade* B dan artinya aplikasi dapat digunakan dengan sangat baik dan layak.

IV. KESIMPULAN DAN SARAN

Pada penelitian ini yaitu Perancangan *Domain Specific Language* Pada Pembuatan Aplikasi *Framework Reporting* Dengan Menggunakan *Python Flask*. Berdasarkan hasil dan pembahasan diambil kesimpulan bahwa *domain specific language* dapat membantu dan mempercepat maupun meningkatkan efisiensi dan efektivitas proses pembuatan laporan karena proses pembuatan mulai dari pengambilan data, pemberian format hingga ke perhitungan yang biasanya dilakukan manual menjadi otomatis dilakukan oleh sistem dengan membaca konfigurasi yang telah diberikan.

Pada pembuatan sistem aplikasi masih memiliki banyak kekurangan yaitu *user interface* masih kurang responsive sehingga ketika dibuka di *smartphone*, tampilan akan rusak dan tidak teratur membuat sistem sulit untuk digunakan. Pada pengembangan selanjutnya diharapkan dapat memperbaiki dan meningkatkan *responsive* pada aplikasi.

DAFTAR PUSTAKA

[1] N. Ashshidiqy and H. Ali, "PENYELARASAN TEKNOLOGI INFORMASIDENGAN STRATEGI BISNIS," *JURNAL EKONOMI DAN MANAJEMEN SISTEM INFORMASI*, vol. 1, pp. 2686–4916, 2019, doi: 10.31933/JEMSI.

[2] A. Suljkanović, B. Milosavljević, V. Indić, and I. Dejanović, "Developing Microservice-Based Applications Using the Silvera Domain-Specific Language," *Applied Sciences (Switzerland)*, vol. 12, no. 13, Jul. 2022, doi: 10.3390/app12136679.

[3] E. Negm, S. Makady, and A. Salah, "Survey on Domain Specific Languages Implementation Aspects," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 10, no. 11, 2019, [Online]. Available: http://www.intentsoft.com/

[4] M. Adrian Halomoan and A. Putra Kharisma, "Pengembangan Domain Specific Language Untuk Aplikasi CRUD Berbasis Web," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 1, pp. 34–41, 2021, [Online]. Available: http://j-ptiik.ub.ac.id

[5] A.-C. Ciobanu, C. Cătălin Miron, D. Streleț, R.-V. Aramă, and A. Bîrsan, "THE DEVELOPMENT OF A DOMAIN SPECIFIC LANGUAGE FOR EMAIL SORTING," *Technical-Scientific Conference of Undergraduate, Master and Phd Students*, pp. 23–25, 2021.

[6] V. Natali and P. Alfadian, "Analisis dan Perancangan Domain Specific Language untuk Data Generator pada Relational Database," *Jurnal Masyarakat Informatika Unjani (JUMANJI)*, vol. 3, no. 1, pp. 64–73, 2019.

[7] C. Wijayanto and Y. A. Susetyo, "IMPLEMENTASI FLASK FRAMEWORK PADA PEMBANGUNAN APLIKASI SISTEM INFORMASI HELPDESK (SIH)," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, pp. 858–868, 2022.

[8] T. A. Kurniawan, "Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 1, p. 77, Mar. 2018, doi: 10.25126/jtiik.201851610.

[9] M. Syarif and W. Nugraha, "PEMODELAN DIAGRAM UML SISTEM PEMBAYARAN TUNAI PADA TRANSAKSI E-COMMERCE," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 4, no. 1, 2020.

[10] A. Pamuji and H. S. Setiawan, "KLUSTERISASI LOG AKTIFITAS PADA SISTEM KEAMANAN FILE DENGAN PENDEKATAN DATA MINING," *Jurnal Teknik Informatika*, vol. 14, no. 2, 2022.
[11] D. Pratama Putra *et al.*, "Rancang Bangun Sistem Informasi Booking Antrian pada Klinik Berbasis Website," *Jurnal Riset Sistem Informasi (RESI)*, vol. 1, 2023.
[12] R. B. D. Putra, E. S. Budi, and A. R. Kadafi, "Perbandingan Antara SQLite, Room, dan RBDLiTe Dalam Pembuatan Basis Data pada Aplikasi Android," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 3, p. 376, Jun. 2020, doi: 10.30865/jurikom.v7i3.2161.
[13] A. Fahrezi, F. N. Salam, G. M. Ibrahim, R. R. Syaiful, and A. Saifudin, "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia," *Jurnal Ilmu Komputer dan Pendidikan*, vol. 1, pp. 1–5, 2022, [Online]. Available: https://journal.mediapub-likasi.id/index.php/logic
[14] E. Kurniawan, A. Nata, and S. Royal, "PENERAPAN SYSTEM USABILITY SCALE (SUS) DALAM PENGUKURAN KEBERGUNAAN WEBSITE PROGRAM STUDI DI STMIK ROYAL," *Journal of Science and Social Research*, no. 1, pp. 43–49, 2022, [Online]. Available: http://jurnal.goretanpena.com/index.php/JSSR
[15] Red Hat, "What is a configuration file?," Nov. 28, 2022. https://www.redhat.com/en/topics/management/what-are-configuration-files (accessed Mar. 22, 2023).
[16] R. Vidmar and N. Creati, "QCOBJ a Python package to handle quantity-aware configuration files," *SoftwareX*, vol. 7, pp. 347–351, Jan. 2018, doi: 10.1016/j.softx.2018.10.003.
[17] A. Muhammad Nur Fauzi *et al.*, "MENGUKUR TINGKAT KEPUASAN PENGGUNA APLIKASI KEARSIPAN MENGGUNAKAN SYSTEM USABILITY SCALE DAN PIECES FRAMEWORK," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 7, no. 1, pp. 231–239, 2022.