

IMPLEMENTASI JASPERREPORTS PADA SISTEM ONLINE REPORTING MENGUNAKAN PYREPORTJASPER DI PT XYZ

Muhamad Rizky Dhean Aryanto¹⁾, Yeremia Alfa Susetyo²⁾

^{1,2} Fakultas Teknik Informatika, Universitas Kristen Satya Wacana, Jl. Diponegoro 52-60, Salatiga
Co Responden Email: 672020289@student.uksw.edu

Abstract

Article history

Received 23 Feb 2024

Revised 14 Mar 2024

Accepted 16 Apr 2024

Available online 30 Apr 2024

Keywords

Online Reporting System,
JasperReports,
PyReportJasper,
Rapid Application
Development

PT XYZ consistently monitors sales, inventory, and performance from its 17,813 branch stores using an online reporting system. Previous technology that used in the online reporting system was HTML to PDF. However, HTML reports often encountered errors, such as displaying incorrect data values, messy layouts, and layout changes that resulted in truncated reports when printed. Therefore, this technology will be replaced by JasperReports, which is the most advanced reporting technology available today. JasperReports has a fixed nature, meaning its layout, design, and final output cannot be altered. This research is conducted swiftly yet comprehensively using the Rapid Application Development method. Implementation is carried out through a dedicated micro-service for reporting based on JasperReports, programmed in Python and utilizing the PyReportJasper package as a bridge between Python and JasperReports. The outcome of this research is an API intended for use by the Store Information System (SIS) application. The API's output is a PDF report with a consistent layout, clear presentation, and easy printing, while adhering to the established design principles of previous reports.

Abstrak

Riwayat

Diterima 23 Feb 2024

Revisi 14 Mar 2024

Disetujui 16 Apr 2024

Terbit Online 30 Apr 2024

Kata Kunci

Pelaporan Online,
JasperReports,
PyReportJasper,
Rapid Application
Development

PT XYZ selalu memantau penjualan, stok barang, dan kinerja dari 17.813 cabang toko yang tersebar di Indonesia menggunakan sebuah sistem pelaporan online. Teknologi yang digunakan dalam sistem pelaporan online sebelumnya adalah HTML to PDF. Namun, laporan HTML kerap kali mengalami error, seperti menampilkan data yang nilainya tidak sesuai, layout yang berantakan, dan perubahan layout yang membuat laporan terpotong saat dicetak. Maka dari itu, teknologi tersebut akan digantikan oleh JasperReports yang merupakan teknologi pelaporan paling mutakhir saat ini. JasperReports memiliki sifat *fixed* atau tidak dapat diubah, baik dari layout, desain, hingga output akhirnya. Penelitian ini dikerjakan dengan waktu singkat namun dilakukan secara maksimal menggunakan metode *Rapid Application Development*. Implementasi dilakukan dalam sebuah *micro-service* khusus laporan yang berbasis JasperReports dengan bahasa pemrograman Python dan menggunakan *package* PyReportJasper sebagai jembatan antara Python dan JasperReports. Hasil dari penelitian ini adalah API yang nantinya digunakan oleh aplikasi *Store Information System* (SIS). Output dari API tersebut adalah laporan PDF yang layout-nya tidak akan berubah, tampilan yang jelas, dan mudah dicetak dengan mempertahankan kaidah desain laporan sebelumnya.

PENDAHULUAN

PT XYZ merupakan salah satu perusahaan jaringan ritel minimarket modern terbesar di Indonesia dengan kurang lebih 17.813 cabang toko yang sudah beroperasi. Jaringan ritel minimarket modern adalah salah satu bidang bisnis yang selalu bertumbuh dari tahun ke tahun di Indonesia. Hal tersebut terjadi karena kebutuhan sehari-hari masyarakat yang terus

mengalami peningkatan (Ong, Sutawijaya, dan Saluy 2020).

Salah satu langkah PT XYZ dalam menghadapi pertumbuhan yang pesat tiap tahunnya adalah melakukan optimalisasi pekerjaan dengan memanfaatkan teknologi paling mutakhir. Salah satu contoh penerapan teknologinya adalah aplikasi *Store Information System* (SIS) yang dibangun dengan menggunakan bahasa pemrograman

Python dan *framework* Kivy. Sistem ini digunakan untuk mencatat, menghitung, serta merekapitulasi data-data toko. Data toko yang dimaksudkan adalah seperti transaksi toko dengan *supplier* barang, jumlah barang keluar masuk dalam gudang, sistem kepegawaian pergantian *shift*, dan sistem pelaporan (*reporting*) data toko.

Sistem pelaporan (*reporting*) pada aplikasi SIS bersifat terintegrasi dan otomatis. Sistem yang terintegrasi dan otomatis diperlukan guna pemantauan kinerja toko secara *real-time* oleh pihak yang berkepentingan (Islamy, Astuti, dan Wibowo 2020). Laporan (*report*) adalah bentuk dari transparansi dan akuntabilitas kinerja para pekerja kepada perusahaan (Yanto dan Hajawiyah 2022). Terdapat puluhan jenis laporan yang dapat dibuat. Dari mulai laporan performa toko, laporan retur, laporan persediaan toko, dan masih banyak lagi.

Laporan-laporan tersebut dapat dibuat dengan *output* tampilan langsung pada sistem, lalu dapat diunduh menjadi *file*, dan yang terakhir adalah langsung dicetak guna pelaporan kepada penanggung jawab. Laporan awalnya didesain menggunakan bahasa pemrograman HTML dan menggunakan teknologi HTML to PDF ketika diimplementasikan ke kode sistem. Fenomena yang sering terjadi terhadap teknologi HTML adalah mengalami *error* seperti berubahnya tampilan *layout* ketika diunduh atau dicetak. Data yang seharusnya ditampilkan kerap kali hilang atau terpotong. Selain itu perancangan desain laporan juga hanya dapat dilakukan secara *hard-code*, sehingga proses pembuatannya cukup memakan waktu dan hanya dapat dilakukan oleh *programmer* saja (Hasanah dan Putro 2020).

Maka dari itu perlu dilakukan perombakan sistem pelaporan di PT XYZ. Opsi terbaik untuk teknologi pelaporan yang dapat digunakan adalah JasperReports. Teknologi tersebut dipilih karena JasperReports adalah *reporting utility* yang bersifat *open-source* yang paling populer di dunia (Ruhamah, Hardiana, dan Andi Ayusiah Febriana 2022). Selain populer, JasperReports juga merupakan alat pelaporan paling efisien jika dibandingkan dengan alat pelaporan lain (Rehman dkk. 2020).

Perancangan laporan juga menggunakan *visual development tool* khusus JasperReports yaitu Jaspersoft Studio. Jaspersoft adalah salah satu penyedia solusi *Business Intelligence* terintegrasi terbesar untuk perusahaan (Rudiawan 2021). Mengingat pengkodean sistem pelaporan online menggunakan bahasa pemrograman Python, penerapan JasperReports dalam sistem tentu akan jauh lebih mudah karena sudah tersedia *package* Python bernama PyReportJasper yang dapat memasukkan data yang ada ke *file* JasperReports dan mengkovernya menjadi *file* yang diinginkan seperti PDF.

Berdasarkan uraian diatas, maka perubahan teknologi pelaporan menggunakan JasperReports dirasa sangat diperlukan. Tujuan dari perombakan teknologi ke JasperReports oleh PT XYZ adalah meminimalisir kerugian yang disebabkan oleh *error* laporan yang gagal menampilkan data dengan benar. Dengan menggunakan JasperReports, masalah masalah tersebut dapat ditanggulangi karena *page formatting* dan *fields formatting*-nya bersifat *fixed* (tidak bisa diubah), tidak seperti teknologi HTML yang *layoutnya* akan mengikuti ukuran layar.

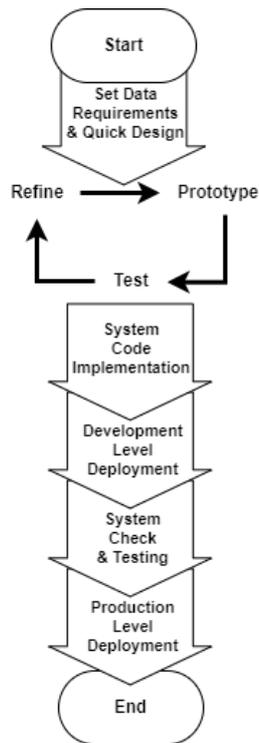
Selain menanggulangi masalah *error*, manfaat lain dari penggunaan JasperReports adalah pembuatan laporan kedepannya akan lebih mudah dan cepat karena memanfaatkan fungsi ambil *fields* dan *drag and drop* desain pada Jaspersoft Studio. Penerapan JasperReports pada sistem pelaporan online aplikasi SIS juga dipermudah dengan hadirnya *package* khusus Python yang dapat langsung digunakan yaitu PyReportJasper.

METODE PENELITIAN

Penerapan teknologi JasperReports pada sistem *online reporting* di PT XYZ ini akan dikembangkan dengan metode *Rapid Application Development* (RAD). RAD dipilih karena proyek ini berskala kecil dengan durasi yang terbilang pendek, yaitu 1 bulan saja. RAD cocok dengan proyek yang terbilang pendek karena dapat mengurangi waktu perencanaan, penerapan desain, dan penerapan teknologi dalam sistem (Fanesyah Musvina, Sri Rahmawati, dan Harkamsyah Andrianof 2022). Dengan menggunakan RAD, fokus pengembangan perangkat lunak akan menjadi cepat karena cakupan kebutuhan user yang

luas akan dilakukan dalam bagian *prototyping* (Faqih, Hikmah, dan Azizah 2022).

Tahap pertama adalah *Set Data*



Gambar 1. *Rapid Application Development*

Requirements, pada tahap ini dilakukan analisis mengenai data-data apa saja yang diperlukan oleh laporan, bagaimana kalkulasinya, *formatting* bentuk data, serta penyimpanan data dalam bentuk *temporary JSON*. *Temporary JSON* nantinya dapat diambil dan dijadikan sumber data sementara dalam melakukan *Prototyping Cycle* secara lokal. Selanjutnya proses *Quick Design* laporan akan dilakukan menggunakan aplikasi Jaspersoft Studio. Perancangan desain laporan akan dilakukan secara dasar berdasarkan layout dan elemen visualnya saja.

Setelah *Set Data Requirements* dan *Quick Design* selesai, maka dilanjutkan ke tahap *Prototyping Cycle*. Pada tahap ini dilakukan pengembangan dan penerapan dari tahapan sebelumnya. Pada tahap ini akan terdiri dari tiga sub tahapan, yang pertama adalah *Prototype*. Sub tahapan *prototype* adalah bagian di mana desain dan data *JSON* akan dihubungkan. *Fields* pada laporan juga harus sesuai penempatannya dan harus dipastikan bahwa semua datanya dapat terbaca dan masuk ke dalam laporan. Laporan yang sudah

terbentuk akan di *preview* melalui Jaspersoft Studio.

Yang kedua adalah sub tahapan *Test*, dimana pada tahapan ini, peneliti sendirilah yang akan membandingkan laporan yang sudah di *preview* menggunakan teknologi JasperReports dengan laporan yang sebelumnya yang menggunakan teknologi HTML. Sehingga dari perbandingan laporan tersebut akan terlihat bagian manakah yang belum sempurna, seperti salah *formatting* bentuk data, atau layout yang terbalik, dan masih banyak lagi.

Sub tahapan selanjutnya adalah *Refine*. Pada tahap *refine* inilah kesalahan-kesalahan yang terlihat pada tahap sebelumnya akan diperbaiki. Setelah diperbaiki kesalahannya akan kembali ke tahapan *prototype*, di mana laporan akan di-*preview* guna melihat apakah yang dibetulkan sudah benar, atau malah menambah error baru dalam laporan

Setelah *Prototyping Cycles* selesai, maka penelitian akan lanjut ke tahap *System Code Implementation*. Pada bagian ini, laporan yang sudah terbentuk dengan baik akan diimplementasikan kedalam kedalam *service* terpisah yang dikhususkan untuk laporan JasperReports. Di mana nantinya pada *service* ini akan menggunakan *package PyReportJasper* guna menghubungkan antara sistem berbasis Python dengan laporan yang menggunakan bahasa *JRXML* dan *jasper*.

Jika laporan sudah dapat diimplementasikan ke dalam sistem, maka tahap selanjutnya adalah *Development Level Deployment*. Pada bagian kode akan di-*deploy* di Google Cloud Platform milik PT XYZ. Namun tahap *deploy* ini hanya sebatas *level development*. *Level* ini digunakan oleh para *developer* di PT XYZ untuk melakukan pengembangan aplikasi atau sistem-sistemnya. Pengembangan sistem dilakukan langsung di Google Cloud Platform karena para *developer* dapat langsung merekayasa sistem dengan menggunakan data yang ada tanpa harus keluar dari ekosistem cloud (Bisong 2019).

Setelah dilakukan proses *deploy* di *level development*, langkah selanjutnya adalah *System Check & Testing*. Pada tahap ini biasanya akan dibantu atau dilakukan oleh para *Quality Assurance*, yang memiliki tugas untuk menjamin kualitas perangkat lunak.

Dengan terjaminnya kualitas sistem, perusahaan dapat meningkatkan produktivitas, mengurangi biaya *maintenance*, dan dapat memastikan semua sistem yang sedang dikembangkan dapat sukses kedepannya (Patel, Kumar, dan Sharma 2020).

Quality Assurance harus memastikan seluruh laporan sama seperti dengan yang diinginkan. Jika sistem dan laporannya sudah dirasa baik, maka dapat dilanjutkan ke tahap selanjutnya yaitu *Production Level Deployment*. Kode yang sudah dibuat sebelumnya akan di-*deploy* di *level production*. *Level production* adalah *level non-development* yang artinya sistem ini akan berjalan dan dapat digunakan oleh seluruh toko cabang PT XYZ di seluruh Indonesia.

HASIL DAN PEMBAHASAN

Pada aplikasi *Store Information System* (SIS) sendiri terdapat banyak fungsi, salah satunya adalah merkapitulasi data dan mengubahnya menjadi laporan (*report*). Terdapat puluhan jenis laporan yang dapat dibuat oleh aplikasi tersebut. Tiap – tiap laporan tentu saja memiliki perhitungan dan pengambilan data yang berbeda-beda. Maka dari itu diperlukan logika yang berbeda pula dalam pembuatan tiap laporan.

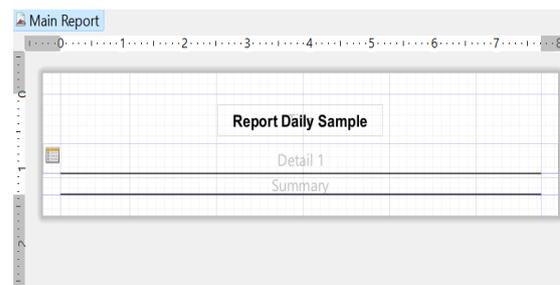
Untuk jenis data yang digunakan adalah bentuk *JSON file*. *JSON* dipilih sebagai sumber data *temporary* karena merupakan bentuk data semi-terstruktur yang ringan (Lv, Yan, dan He 2019). Nantinya *JSON file* yang berisikan data hasil kalkulasi akan digunakan oleh *file* laporan JasperReports (berupa file “.jrxml” dan file “.jasper”) sebagai sumber data. Arsitektur *JSON*-nya juga harus dikonfigurasi sesuai dengan kebutuhan laporan, dimana data-datanya harus bersifat urut dan berlapis karena kemungkinan tiap laporan memiliki subreport / anak laporan yang berupa file “.jasper” dan “.jrxml” yang berdiri sendiri.

```
json_architecture_example = {
  'main': [
    'subdata': [
      'date': '-',
      'retail_sales': '-',
      'sales_grow': '-',
      'online_transaction': '-',
      'needs_payment': '-',
      'net_retail': '-',
    ],
    'total_retail_sales': '-',
    'total_sales_grow': '-',
    'total_online_transaction': '-',
    'total_needs_payment': '-',
    'total_net_retail': '-',
  ]
}
```

Gambar 2. Kode arsitektur *JSON*

Dapat dilihat dari contoh gambar 2, dimana data yang berada di dalam ‘subdata’ adalah data yang nanti akan digunakan oleh *file subreport* / anak laporan di *file* yang berbeda. Sedangkan untuk data yang berada di *main* nantinya akan digunakan di *main report*. Selain arsitektur, nilai dalam datanya juga harus diformat bentuknya, dimana data yang didapatkan dari database bersifat mentah dan masih perlu diproses agar sesuai dengan kaidah penulisan angka dan mata uang di Indonesia.

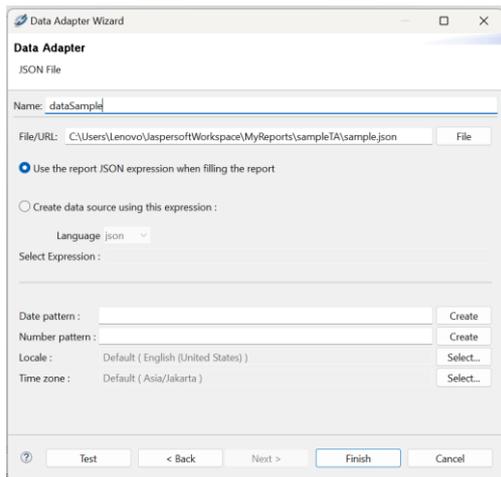
Quick Design laporan sendiri dilakukan secara lokal menggunakan aplikasi JasperSoft Studio. Untuk desain dari laporannya sendiri tinggal mengikuti kaidah laporan yang digunakan oleh perusahaan. Karena laporan sementara belum tersambung dengan sumber data, maka pendesainan laporan dapat berupa mengatur *layout*-nya, mengatur elemen desain *visual*, mengatur penempatan *fields* nantinya, menentukan tiap kolom kolom laporan seperti berapa detail, *summary*, atau *column header*, dan menentukan membuat *subreport* dan mendesain peletakkan *subreport*-nya. Contoh desain dasar-nya seperti pada gambar 3.



Gambar 3. Pembuatan *layout* dan desain sesuai kaidah perusahaan

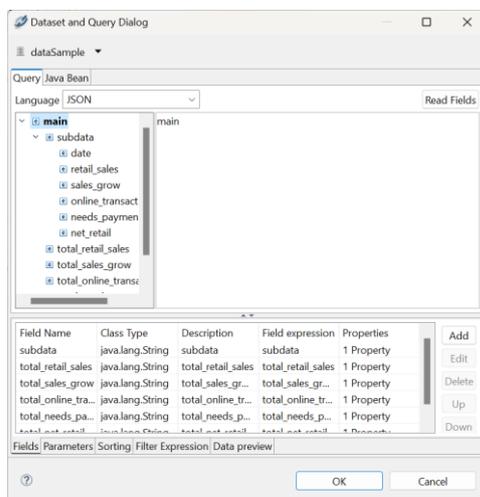
Setelah mengatur data dan desain selesai, maka langkah selanjutnya adalah melakukan penerapan dan pengembangan dari laporan yang sudah dibuat. Untuk mempersingkat dan mempermudah dalam melakukan *prototyping* / pengimplementasian antara data dan desain, maka *prototyping* akan dilakukan secara *offline* / lokal. Untuk melakukan *prototyping* secara lokal diperlukan sebuah *data adapter* untuk digunakan oleh laporan sebagai sumber datanya. Karena data yang akan diambil nanti berupa *temporary JSON*, maka diperlukan sebuah file *JSON sample* sebagai *database* dari *data adapter* yang akan dibuat. *Data temporary JSON* yang diperlukan tinggal

mengikuti bentuk arsitektur JSON yang sudah dibuat sebelumnya.



Gambar 4. Set Data Adapter

Pada gambar 4 dijelaskan pembuatan *data adapter* dengan memasukkan url lokasi file JSON. Setelah *data adapter* selesai, maka setiap *field* yang ada dalam data JSON dapat langsung dihubungkan ke dalam laporan dengan menambahkan *dataset and query*.

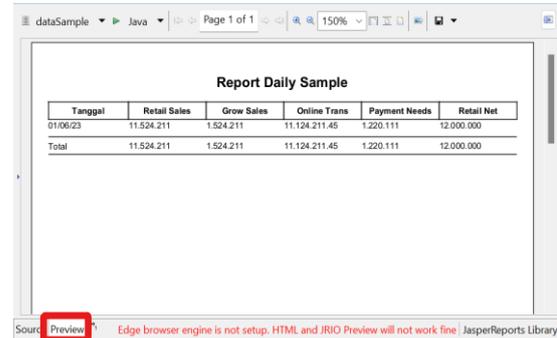


Gambar 5. Dataset and Query

Jika *dataset and query*-nya sudah diatur maka laporan dapat menarik data yang ada di dalam JSON dan mengubahnya menjadi *fields* di dalam aplikasi JasperSoft Studio. *Fields* dapat langsung dilakukan *drag and drop* sesuai dengan desain awal yang sudah ditentukan tanpa harus kebingungan mengenai mengatur nilai nilai yang ada didalam *fields*. Laporan hanya perlu dipercantik sedikit dari segi desain luarnya saja seperti pengaturan ukuran teks, penataan *fields*, dan pemilihan *font* yang sesuai.

Dalam melakukan *testing* pada *prototyping cycles* ini dapat dilakukan secara

langsung di aplikasi JasperSoft Studio. Ketika pendesainan sudah selesai maka dapat langsung saja melakukan *preview* laporan dengan menekan tombol seperti pada gambar 6.



Gambar 6. Fitur *preview*

Dengan menggunakan fitur *preview* laporan dapat dikoreksi desain dan datanya. *Error* yang sering terjadi pada saat tahap *prototyping cycles* di JasperSoft Studio adalah kesalahan dalam penampilan data. Data angka-angka kerap kali berubah format atau bahkan tidak dapat ditampilkan. Salah satu kesalahan yang sering terjadi adalah dimana *string* angka yang ada di dalam *file* JSON terbaca sebagai jenis data *Big Decimal*. Maka dari itu perlu dilakukan *refine* kode yang sebelumnya merupakan *big decimal* menjadi *string* pada file “*jrxml*” laporan.

```
<field name="avgSpdTtl"
class="java.lang.
BigDecimml">
<property
name="net.sf.jasperreports
.json.field.expression"
value="avgSpdTtl"/>
<fieldDescription><![CDATA
[avgSpdTtl]]>
</fieldDescription>
</field>
```

Menjadi:

```
<field name="avgSpdTtl"
class="java.lang.String">
<property name="net.sf.
jasperreports.json.
field.expression"
value="avgSpdTtl"/>
<fieldDescription><![
CDATA[avgSpdTtl]]>
</fieldDescription>
</field>
```

Gambar 7. Contoh *refine* kode

Setelah tahapan *prototyping cycles* selesai, maka laporan dapat langsung diimplementasikan ke *service* baru untuk aplikasi *Store Information System (SIS)* khusus laporan berbasis JasperReports. *Service* khusus ini merupakan *service* yang berbasiskan Flask dengan bahasa pemrograman Python. Sehingga dalam sistem

tersebut diperlukan *global function* yang berisikan logika dalam pembuatan laporan dengan menggunakan package PyReportJasper.

```
from pyreportjasper import PyReportJasper
class Jasper:
    def __init__(self):
        self.pyreportjasper =
            PyReportJasper()

    def generate_report(self,
        input_file='',
        output_file='',
        conn=None,
        output_formats=None,
        param=None,
        locale='id_ID'):
        if output_formats is None:
            output_formats = ["pdf"]
        if param is None:
            param = {}
        if conn is None:
            conn = {}
        self.pyreportjasper.config(
            input_file,
            output_file,
            db_connection=conn,
            output_formats=output_formats,
            parameters=param,
            locale=locale
        )
        self.pyreportjasper.process_report()
        report_path = output_file + '.pdf'
        return open(report_path, 'rb')
```

Gambar 8. *Global function* PyReportJasper

Pada gambar 8 dapat dilihat bahwa fungsi dari kelas Jasper adalah membuat laporan dengan beberapa parameter yang sudah ditentukan seperti `input_file` yang merupakan file “.jrxml-nya”, lalu `output_format` yang merupakan lokasi file outputnya, selanjutnya adalah `conn` yang merupakan koneksi ke data yang akan digunakan berupa file JSON temporary, dan parameter yang merupakan file subreport yang berupa file “.jasper”. Untuk file subreport-nya memang harus diterima sebagai sebuah parameter, dan juga harus berupa file “.jasper” bukan file “.jrxml”. Package tersebut memang menentukan bahwa file utamanya haruslah berupa file dengan tipe “.jrxml” sedangkan untuk file subreport-nya harus berupa file “.jasper”.

Karena tiap laporan memiliki kalkulasi dan pengambilan data yang berbeda-beda, maka pengimplementasian PyReportJasper-nya akan diletakkan pada satu file yang sama dengan logika pengambilan data dan kalkulasinya. Sehingga pada tiap kode model laporan memiliki dua “def” fungsi, yang pertama yaitu fungsi pengambilan dan kalkulasi data sekaligus mengubah data yang diambil menjadi JSON temporary dan yang kedua adalah fungsi pengimplementasiannya ke JasperReports.

Untuk pengambilan data dan pembentukan JSON temporary-nya dapat langsung dibuat

sesuai dengan arsitektur JSON yang sudah dibuat sebelumnya. Tidak lupa untuk mengatur *formatting* beserta dengan bentuk locale / penulisan angka dan mata uang sesuai dengan yang digunakan di Indonesia karena file JSON nantinya hanya akan menarik data yang sudah dalam bentuk string yang siap pakai, bukan bentuk decimal mentah yang belum diolah.

Diperlukan juga logika *if else* yang bertujuan untuk memastikan koneksi ke database. Jika koneksi database tidak tersambung maka akan membuat sebuah JSON yang berisikan fields kosong, namun jika koneksi tersambung maka akan melakukan pengambilan data dan menjadikan data tersebut file JSON temporary.

```
if result:
    JSON_arch = {
        'main': [{
            'subdata': [{
                'subdatadetail': detail,
            }],
            'mainDataDetail': mainDetail,
        }]
    }
    random_uid = str(uuid.uuid1())
    uid = ''.join(random_uid)+'example'
    FOLDER_OUTPUT =
    os.path.join(os.getcwd(), f"{uid}.json")
    with open(f"{FOLDER_OUTPUT}", 'w') as file:
        json.dump(JSON_arch, file)
    return 200, uid
```

Gambar 9. Kode menyimpan JSON

Pada gambar 9 dapat dijelaskan bahwa data yang diambil (“detail” dan “mainDetail”) diletakkan dalam fields JSON, lalu selanjutnya file JSON akan disimpan ke dalam file yang sudah ditentukan di FOLDER_OUTPUT dengan uuid sebagai namanya. Penggunaan uuid sebagai penanggulangan error agar nama JSON tidak terduplikasi jika misal terjadi penumpukan file JSON. Lalu logika else-nya pada dasarnya sama namun JSON-nya dibuat kosong atau string “-”.

Lalu fungsi kedua adalah logika pembuatan laporan JasperReports menggunakan JSON temporary yang sudah dibuat sebelumnya. Pada fungsi ini perlu diatur dimana direktori dari file “.jrxml” dan “.jasper”-nya berada, lalu diperlukan juga lokasi direktori dimana file PDF temporary yang dibuat nanti, dan yang terakhir adalah dimana file JSON temporary yang sudah dibuat sebelumnya disimpan. Tidak lupa untuk melakukan import fungsi jasper dari global function yang sudah dibuat sebelumnya

dengan kode “from globals.jasper import Jasper”.

```
try:
    status, result = self.generate_json(store_id)
    if status != 200:
        return 400, result, ''
    reports_dir = os.path.join(os.getcwd(), 'reportsfolder')
    input_file = os.path.join(reports_dir, 'main.jrxml')
    sub_file = os.path.join(reports_dir, 'subreport.jasper')
    output_file = os.path.join(reports_dir, 'pdf', result)
    data_file = os.path.join(reports_dir, f'{result}.json')
    conn = {
        'driver': 'json',
        'data_file': data_file,
        'json_query': 'main_data'
    }
    parameters = {'subreport': sub_file}
    jasper = Jasper()
    result = jasper.generate_report(input_file=input_file,
    output_file=output_file, conn=conn, param=parameters)

    return 200, result, output_file + '.pdf'
except Exception as e:
    error_log(f"Error Create Report : ", e)
    return 400, {'message': f"Error Create Report : {e}"}, ''
```

Gambar 10. Kode generate laporan

Dari kode pada gambar 10 dapat dilihat bahwa langkah pertamanya adalah mengatur lokasi *file-file* yang diperlukan. Setelah lokasi *file* dan direktorinya diatur maka dapat langsung membuat koneksi yang berisi jenis *driver data* yang digunakan, lokasi data *file-nya*, dan *query / nama field* pada JSON yang diinginkan. Jika laporan memiliki sub-laporan / *subreport*, maka laporan harus dalam bentuk parameter. Dimana parameter tersebut berisi nama file *subreport-nya*. Jika koneksi sudah benar maka dapat langsung mengambil fungsi *jasper* yang sudah di-*import* dan digunakan dengan memuat berbagai parameter seperti input *file-nya*, lokasi *output-nya*, koneksi, dan yang terakhir parameter untuk *subreport-nya*. Pastikan juga bahwa balikan dari fungsi tersebut adalah hasil dari perubahan laporannya, seperti pada kode tersebut mengembalikan status 200, nilai *result* dari fungsinya, dan yang terakhir adalah *file PDF*.

Setelah sistem selesai dibuat, maka satu satunya cara memastikannya adalah dengan melakukan *deploy* sistem tersebut ke Google Cloud. Dengan menggunakan *cloud*, perusahaan akan diuntungkan karena sifatnya yang fleksibel, menejemen layanan yang reliabel, dan skalabilitas yang besar (Ramsari dan Ginanjar 2022). Karena sistem masih berupa pengembangan, maka proses *Deploying* dari setiap sistem yang ada akan dibagi menjadi dua tahap yaitu tahap *staging* dan *production*. Tahap *staging* sendiri adalah tahap dimana sistem dijalankan namun ketika masih dalam pengembangan. Sehingga jika terjadi *error* dalam pengembangan tidak akan mengganggu jalannya proses bisnis dari perusahaan. Sistem yang sudah dibuat

sebelumnya dapat langsung di *deploy* di tahap *staging* melalui Cloud Run.

Setelah sistem selesai di *deploy*, maka laporan dapat diakses menggunakan API yang sudah dibuat melalui Cloud Run sebelumnya. Laporan dapat diakses dengan menambahkan beberapa *key parameter* sesuai dengan jenis laporan yang diinginkan. Pada perusahaan PT. XYZ ini, *system check* dan *testing* dibantu / dilakukan oleh karyawan yang bertugas sebagai *quality assurance*. Para *quality assurance* melakukan pengecekan dan *testing* sesuai dengan peraturan yang sudah diatur oleh perusahaan. Langkah pertama yang dilakukan seorang *quality assurance* adalah melakukan pengecekan sistem dengan melakukan *hit API* yang sudah dibuat sebelumnya. Aplikasi yang diandalkan dalam mengolah API adalah Postman, salah satu alat dokumentasi dan otomatisasi API paling besar saat ini (Kore dkk. 2022).

API akan dites dengan aplikasi Postman dengan berbagai *key parameter* yang berbeda beda guna memastikan apakah ada kesalahan dalam sistem yang membuat laporan tidak dapat tampil. Jikalau terjadi *error*, pada postman akan memunculkan *response JSON*. Dengan begitu dipastikan bahwa ada yang salah dalam pengkodean sebelumnya. Karena sistem yang berbasis Google Cloud, maka *error* tersebut dapat di tracking dengan menggunakan Logs Explorer pada Google Cloud Console.

Langkah selanjutnya adalah pengujian pada laporan. Setiap laporan harus diuji sesuai dengan standar perusahaan. Karena tiap laporan ini penting, maka pengujian laporan harus dilakukan secara teliti dan seksama oleh *quallity assurance*. Blackbox *testing* dipilih karena pengujian tidak perlu untuk memahami bahasa pemrograman didalamnya dan dilakukan dengan kacamata seorang *user* (Suwirmayanti dkk. 2020). Blackbox Testing tersebut akan dilakukan dengan 5 skema pengetesan seperti pada tabel 1.

Tabel 1. Skenario Blackbox testing

Nama Pengujian	Deskripsi	Hasil Pengujian	Status
Generate laporan	Hit API laporan dengan beberapa	Laporan muncul sesuai dengan	Valid

	Query Param	Query Param	
Aturan laporan	Desain harus sesuai aturan perusahaan	Desain laporan sudah sesuai peraturan	Valid
Pengecekan Desain	Desain tidak boleh memiliki cacat, seperti layout yang berubah dan elemen yang berantakan	Tampilan tidak memiliki kecacatan dan semua data muncul sesuai keinginan	Valid
Pengecekan Kalkulasi	Hasil kalkulasi dari <i>fields</i> laporan sudah sesuai dengan perhitungan yang benar	Kalkulasi dalam laporan sudah benar dan sesuai dengan kalkulasinya	Valid
Pengecekan Print Out Laporan	Memeriksa hasil cetak dari laporan yang dibuat dengan menggunakan <i>printer</i> khusus LX-310 dengan kertas khusus	Hasil cetak laporan sudah sesuai dan tidak ada dislokasi maupun tidak ada bagian yang terpotong	Valid

Hasil dari ke lima skema yang diterapkan pada laporan tersebut menunjukkan hasil yang *valid*. Jika memang sistem dirasa *valid* setelah pengujian, maka sistem dapat langsung dilakukan *Production Level Deployment*. Sistem yang sudah sampai ke level *Production Level Deployment* dianggap siap digunakan oleh semua karyawan toko di seluruh Indonesia. *Deploy* di level ini hanya dapat dilakukan oleh beberapa karyawan tertentu yang memang diberi akses ke *level production*. Akses yang dibatasi ini dimaksudkan agar pengelolaan dan keamanan pada *level production* ini terjaga dengan baik.

KESIMPULAN

Berdasarkan hasil pengujian Blackbox *testing* oleh *quality assurance* PT XYZ, didapatkan hasil valid di ke-lima skema pengujiannya. Lima skema Blackbox *testing* tersebut didasarkan pada *error* yang sebelumnya sering terjadi pada teknologi HTML. Dengan begitu, pengimplementasian JasperReports pada sistem pelaporan online merupakan jawaban dari masalah-masalah yang sebelumnya terjadi ketika masih menggunakan HTML. Sistem laporan JasperReports ini sudah di jalankan dengan skala nasional dan sudah digunakan di 17.813 cabang toko yang tersebar di Indonesia.

Karena waktu pengerjaan proyek yang hanya diberi waktu singkat, penggunaan metode *Rapid Application Development* juga sangat membantu penulis dalam pengerjaan sistem pelaporan online ini. Tercatat sudah 10 laporan sudah mengalami perombakan menjadi menggunakan teknologi JasperReports. Dan kesepuluh laporan tersebut merupakan laporan yang paling sering digunakan oleh karyawan toko untuk dicetak. Sepuluh laporan tersebut sudah dapat digunakan oleh karyawan di seluruh toko yang tersebar di Indonesia.

Selama proses implementasi JasperReports ini, disadari bahwa masih terdapat banyak kekurangan dalam penggunaan fitur di dalam JasperSoft Studio, seperti fitur pengolahan data dengan jenis integer dan big decimal. Hal tersebut terjadi karena waktu pengimplementasiannya yang terbatas sehingga waktu untuk riset lebih dalam fitur-fitur dalam Jaspersoft Studio masih terasa kurang. Maka dari itu seluruh pengolahan data pada implementasi kali ini hampir semuanya dilakukan pada kode Python-nya.

REFERENSI

Bisong, Ekaba. 2019. "An Overview of Google Cloud Platform Services." Dalam *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, Berkeley, CA: Apress, 7–10. doi:10.1007/978-1-4842-4470-8_2.

Faneshyah Musvina, Sri Rahmawati, dan Harkamsyah Andrianof. 2022.

- “IMPLEMENTASI METODE RAPID APPLICATION DEVELOPMENT (RAD) DALAM PERANCANGAN SISTEM INFORMASI PERPUSTAKAAN PADA SMPN 22 PADANG.” *Jurnal ilmiah Sistem Informasi dan Ilmu Komputer* 2(2): 74–90. doi:10.55606/juisik.v2i2.226.
- Faqih, Husni, Agung Baitul Hikmah, dan Wafiq Azizah. 2022. “Implementasi Metode Rapid Application Development Pada Pengembangan Aplikasi e-Fin Mosque Z.” *Indonesian Journal on Software Engineering (IJSE)* 8(1): 83–91. doi:10.31294/ijse.v8i1.13007.
- Hasanah, Amalia Nur, dan Hanson Prihantoro Putro. 2020. *Implementasi JasperReports pada Sistem Informasi Manajemen (Studi Kasus: Sistem Informasi Manajemen Penyelesaian Laporan Ombudsman)*.
- Islamy, Imam Teguh, Hanim Maria Astuti, dan Radityo Prasetyanto Wibowo. 2020. “Perancangan dan Pembuatan Sistem Pelaporan Kinerja Berbasis Online untuk Pranata Komputer.” *JUITA: Jurnal Informatika* 8(1): 91. doi:10.30595/juita.v8i1.5873.
- Kore, Pratik Prakash, Mayuresh Jaywant Lohar, Madhusudan Tanaji Surve, dan Snehal Jadhav. 2022. “API Testing Using Postman Tool.” *International Journal for Research in Applied Science and Engineering Technology* 10(12): 841–43. doi:10.22214/ijraset.2022.48030.
- Lv, Teng, Ping Yan, dan Weimin He. 2019. “On Massive JSON Data Model and Schema.” *Journal of Physics: Conference Series* 1302(2): 022031. doi:10.1088/1742-6596/1302/2/022031.
- Ong, Johan Oscar, Achmad H Sutawijaya, dan Ahmad Badawi Saluy. 2020. “STRATEGI INOVASI MODEL BISNIS RITEL MODERN DI ERA INDUSTRI 4.0.” 6(02): 201–10. <https://www.cnbcindonesia.com>.
- Patel, Shrikant, Sanjay Kumar, dan N Sharma. 2020. “Software Quality Assurance and its Requirements.” *Solid State Technology*: 7489–7506.
- Ramsari, Nopi, dan Arif Ginanjar. 2022. “Implementasi Infrastruktur server berbasis cloud computing untuk web service berbasis teknologi google cloud platform.” Dalam *Conf. Senat. STT Adisutjipto Yogyakarta*,.
- Rehman, Abdul dkk. 2020. “R-PMA: Rapid Efficient Statistical Reporting on Heterogeneous Resources.” Dalam *Proceedings - 2020 International Conference on Intelligent Computing, Automation and Systems, ICICAS 2020*, Institute of Electrical and Electronics Engineers Inc., 6–11. doi:10.1109/ICICAS51530.2020.00010.
- Rudiawan, Hendri. 2021. 23 *Jurnal Ekonomi PEMANFAATAN SISTEM BISNIS INTELIJEN (BI) DALAM PENGAMBILAN KEPUTUSAN MANAJEMEN PERUSAHAAN*.
- Ruhamah, Hardiana, dan Andi Ayusiah Febriana. 2022. “APLIKASI PERPUSTAKAAN PADA SMK HARAPAN LAMASI.” *Jurnal Ilmiah Information Technology d’Computare* 12(Juli).
- Suwirmayanti, Ni Luh Gede Pivin dkk. 2020. “Penerapan Helpdesk System dengan Pengujian Blackbox Testing.” *Jurnal Ilmiah Intech: Information Technology Journal of UMUS* 2(02). doi:10.46772/intech.v2i02.290.
- Yanto, Heri, dan Ain Hajawiyah. 2022. “The impact of risk management committee on integrated reporting disclosure in Indonesia.” *Kasetsart Journal of Social Sciences* 43(4): 999–1008. doi:10.34044/j.kjss.2022.43.4.24.