

# Implementasi Python-PPTX dan Dom-to-Image dalam Pembangunan Sistem Otomatisasi Laporan Berbasis Web di PT. XYZ

Mikha Kurnia Anggi Purbaningtyas <sup>1\*</sup>, Yeremia Alfa Susetyo <sup>2</sup>

<sup>1,2</sup>Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

Email: 672020109@student.uksw.edu <sup>1\*</sup>, yeremia.alfa@uksw.edu <sup>2</sup>

## Histori Artikel:

*Dikirim* 17 Februari 2024; *Diterima dalam bentuk revisi* 18 Maret 2024; *Diterima* 28 Maret 2024; *Diterbitkan* 10 Mei 2024. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STM IK Indonesia Banda Aceh.

## Abstrak

PT. XYZ merupakan sebuah perusahaan retail terkemuka di Indonesia yang beroperasi dalam industri yang sangat dinamis. PT. XYZ menggunakan laporan presentasi PowerPoint (PPT) sebagai alat komunikasi kunci dengan mitra bisnisnya, namun proses pembuatannya masih dilakukan secara manual. Hal tersebut menyebabkan pemborosan waktu, risiko kesalahan, dan gangguan pada efisiensi operasional. Penelitian ini bertujuan mengimplementasikan Python-pptx untuk otomatisasi penyusunan laporan dan dom-to-image untuk integrasi data visual. Tujuannya adalah mengatasi hambatan utama yang dihadapi oleh PT. XYZ. Metode penelitian mencakup identifikasi masalah, studi literatur, analisis kebutuhan sistem, perancangan sistem, pembangunan aplikasi, dan pengujian sistem. Hasil dari penelitian ini adalah implementasi berhasil dari Python-pptx dan dom-to-image dalam membangun sistem otomatisasi laporan berbasis web di PT. XYZ. Pengujian sistem menunjukkan keberhasilan teknologi ini, menjamin kepuasan pengguna, dan menyiapkan sistem untuk penggunaan yang lebih efisien.

**Kata Kunci:** Otomatisasi Laporan; Python-pptx; Dom-to-image.

## Abstract

PT. XYZ is a leading retail company in Indonesia operating in a highly dynamic industry. Despite using PowerPoint (PPT) presentation reports as a key communication tool with its business partners, PT. XYZ's report creation process is still done manually. This leads to time wastage, the risk of errors, and disruptions to operational efficiency. The purpose of this research is to implement Python-pptx for the automation of report compilation and dom-to-image for the integration of visual data. The goal is to overcome the main challenges faced by PT. XYZ. The research methodology includes problem identification, literature review, system requirements analysis, system design, application development, and system testing. The results of this research include the successful implementation of Python-pptx and dom-to-image in building a web-based automated reporting system at PT. XYZ. System testing demonstrates the success of this technology, ensuring user satisfaction, and preparing the system for more efficient use.

**Keyword:** Report Automation; Python-pptx; Dom-to-image.

## 1. Pendahuluan

PT. XYZ merupakan sebuah perusahaan retail terkemuka di Indonesia yang beroperasi dalam industri yang sangat dinamis. Kehadirannya dalam skala transaksi harian yang mencapai ribuan buah memandang pentingnya penyusunan laporan sebagai langkah strategis dalam mengidentifikasi pola dan preferensi konsumen. Penerapan visualisasi data yang efektif dalam proses pelaporan menjadi krusial dalam upaya pemahaman mendalam terhadap dinamika penjualan dan tren dalam lingkup toko, sekaligus menjadi instrumen vital dalam pengambilan keputusan di masa yang akan datang (Gelar Guntara 2023). Dalam mencapai daya saing yang tangguh di pasar, membangun keunggulan kompetitif, serta meraih tujuan perusahaan, keterlibatan yang efisien dalam kerangka kerja hubungan bisnis (B2B) menjadi faktor yang amat signifikan (Maria Kiftya & Bachtiar 2023).

Dalam konteks hubungan bisnisnya, PT. XYZ menggunakan laporan berformat presentasi PowerPoint (PPT) sebagai media berkomunikasi dan berkolaborasi dengan para mitra bisnisnya. Laporan ini harus disajikan secara jelas, memungkinkan analisis pola data, dan mendukung pengambilan keputusan berbasis data. Namun, hingga saat ini, proses pembuatan laporan masih dilakukan secara manual. Setiap diagram yang dibutuhkan diambil dari website dengan ditangkap layarnya satu per satu, kemudian dimasukkan ke dalam PowerPoint secara manual. Pendekatan ini tidak hanya mengonsumsi waktu, energi, dan pemikiran yang seharusnya bisa dialokasikan untuk kegiatan lainnya, namun juga meningkatkan risiko kesalahan yang terjadi akibat faktor manusia (Yoga Prabowo *et al.* 2022). Untuk mengatasi tantangan tersebut, diperlukan sebuah solusi yang mampu mengotomatisasi pembuatan laporan presentasi PowerPoint dengan cepat dan efisien. Pendekatan ini diharapkan dapat mengurangi kendala yang berasal dari proses manual dan meningkatkan efisiensi dalam penyusunan laporan yang menjadi landasan komunikasi dengan mitra bisnis (Ayu Megawaty, Alita & Sukma Dewi 2021).

Python-pptx adalah sebuah perpustakaan Python yang mampu untuk membuat dan mengelola dokumen berformat presentasi PowerPoint (pptx) (Borade, Jyoti G., Kiwelekar & Netak 2022). Perpustakaan ini memungkinkan pembuatan presentasi PowerPoint secara otomatis dari berbagai sumber data (Canny 2023). Dengan kemampuannya menambahkan slide, gambar, tabel, teks, serta mengelola format dan layout sesuai kebutuhan, Python-pptx menjadi alat yang kuat untuk mengotomatisasi proses pembuatan laporan. DOM, atau Model Objek Dokumen, adalah kerangka data logis yang menggambarkan susunan berbagai objek konten dalam dokumen HTML (Galahartlambang *et al.* 2021). Dengan menggunakan DOM, pengembang website dapat melakukan manipulasi pada halaman web HTML. Dom-to-image adalah sebuah perpustakaan yang ditulis dalam bahasa pemrograman JavaScript dan dapat mengonversi simpul DOM (Model Objek Dokumen) menjadi gambar vektor (SVG) atau raster (PNG atau JPEG) (Tsayan *n.d.*).

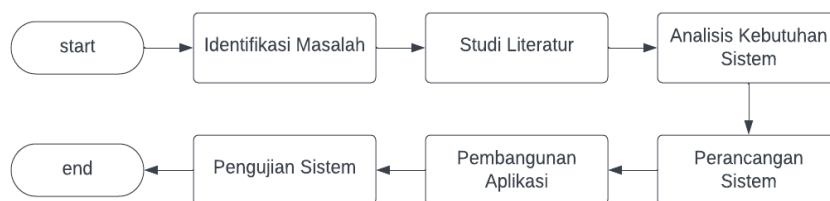
Pada penelitian yang berjudul "Automasi Pembuatan Laporan Analisis dengan R Markdown" membahas mengenai implementasi sistem otomatisasi dalam pembuatan laporan dengan penekanan pada pemanfaatan R Markdown, serta manfaat otomatisasi ini khususnya bagi ASN (Aparatur Sipil Negara). Penelitian tersebut menjelaskan bahwa Aparatur Sipil Negara (ASN) banyak membuat laporan. Namun, pembuatan laporan tersebut masih dilakukan secara manual, padahal seringkali laporan yang dibuat memiliki format dan bentuk laporan yang sama. Hal tersebut menyita banyak waktu, tenaga, dan pikiran yang dapat dialokasikan untuk kegiatan lainnya. Selain itu, pembuatan laporan secara manual rentan mengalami human error. Dalam merespon permasalahan ini, peneliti telah mengembangkan solusi dalam bentuk otomatisasi. Otomatisasi ini diwujudkan dengan memanfaatkan fungsi R Markdown yang pada akhirnya mampu meningkatkan efektivitas dan efisiensi, serta memudahkan reproduksi dalam proses penyusunan laporan analisis (Yoga Prabowo *et al.* 2022). Penelitian tersebut menyoroti keunggulan otomatisasi dalam pembuatan laporan, dengan fokus pada peningkatan efektivitas, efisiensi, dan kemudahan reproduksi dalam penyusunan laporan analisis. Meskipun metode penelitian yang diajukan berbeda, manfaat tersebut menjadi acuan untuk pengembangan otomatisasi dalam pembuatan laporan PowerPoint. Penelitian berjudul "*Automatic Grading of Student's Presentation Skills based on PowerPoint Presentation and Audio*" dijabarkan penerapan

program untuk mengekstrak fitur presentasi dan audio dari PowerPoint dengan menggunakan pustaka Python-pptx dan Librosa. Python-pptx digunakan secara spesifik untuk mengekstraksi beragam fitur terkait presentasi, seperti teks, tabel, grafik, gambar, footer, dan hyperlink. Proses ekstraksi ini dimaksudkan untuk mengenali serta mengevaluasi elemen-elemen penting yang terdapat dalam presentasi PowerPoint. Hasilnya, skor dari the Quadratic Weighted Kappa (QWK) mencapai 0.82, menandakan kesesuaian yang signifikan antara skor yang diperoleh dari sistem otomatisasi dengan skor penilaian yang diberikan oleh manusia (Borade, J. G., Kiwelekar & Netak 2022). Penelitian tersebut berhasil memanfaatkan Python-pptx secara efektif dalam memproses presentasi PowerPoint, khususnya dalam proses ekstraksi informasi dari presentasi tersebut. Hal ini menjadi referensi penting dalam konteks penelitian yang saat ini dilakukan, yang fokusnya berbeda karena lebih menekankan pada penggunaan Python-pptx untuk mengelola proses pembuatan presentasi.

Pada penelitian yang berjudul "Web Application with Built-in Remote Support Capability by Means of Voice Communication and Screen Sharing" menjelaskan tentang sebuah aplikasi web yang memungkinkan pengguna untuk meminta bantuan pendukung jarak jauh melalui komunikasi suara serta berbagi layar dalam satu aplikasi tanpa perlu alat tambahan. Penelitian ini merinci penggunaan metode Html2Canvas dalam mengambil tangkapan layar pengguna. Penelitian ini secara konkret menentukan objek yang akan di render dalam file HTML dengan memasukkannya dalam tag div. Dengan pendekatan ini, elemen-elemen khusus dari halaman web dapat diisolasi dan dianggap sebagai objek yang akan di-capture oleh Html2Canvas(). Namun, metode Html2Canvas memiliki kelemahan terkait kecepatan karena memerlukan waktu yang cukup lama. Dalam konteks ini, peneliti merujuk pada metode dom-to-image yang memerlukan waktu yang lebih singkat daripada Html2Canvas. Metode ini dijadikan sebagai pedoman dalam pengambilan tangkapan layar untuk otomatisasi pembuatan laporan PowerPoint berbasis web (Nakamura, Niibori & Kamada 2022).

Pada penelitian ini akan dilakukan penerapan Python-pptx untuk otomatisasi penyusunan laporan dan dom-to-image untuk integrasi data visual, sehingga PT. XYZ dapat mengatasi masalah utama yang mereka hadapi. Proses manual yang sebelumnya menimbulkan pemborosan waktu, risiko kesalahan, serta mengganggu efisiensi operasional dapat diminimalkan dengan signifikan. Hal ini diharapkan dapat memberikan ruang fokus yang lebih besar bagi perusahaan untuk memastikan konsistensi, meningkatkan efisiensi operasional, sekaligus memperkuat jaringan kerja bisnis yang strategis dengan mitra-mitra mereka.

## 2. Metode Penelitian

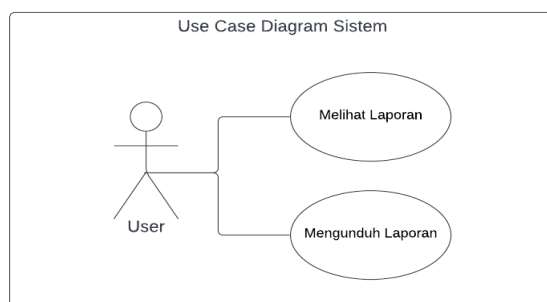


Gambar 1. Tahap Penelitian

Pada Gambar 1, dijelaskan langkah-langkah yang diterapkan dalam penelitian ini. Tahap pertama adalah mengidentifikasi masalah yang ada di PT. XYZ, yaitu proses pembuatan laporan presentasi PowerPoint yang digunakan sebagai media komunikasi dan kolaborasi dengan para mitra bisnisnya masih dilakukan secara manual. Setiap diagram yang dibutuhkan diambil dari *website* dengan ditangkap layarnya satu per satu, kemudian dimasukkan ke dalam PowerPoint secara manual. Berdasarkan masalah tersebut, akan dibangun sistem yang dapat secara cepat dan efisien mengotomatisasi pembuatan laporan presentasi PowerPoint. Setelah mengidentifikasi masalah, langkah selanjutnya adalah melakukan *studi literatur*. Ini melibatkan pengumpulan informasi dari berbagai sumber tertulis

seperti artikel jurnal, buku, dan dokumen lainnya untuk mendapatkan data yang mendukung penelitian ini (Habsy 2017). Penelitian ini menggunakan sumber-sumber literatur yang terkait dengan sistem otomatisasi laporan, pembuatan laporan secara otomatis, dan teknik menangkap layar.

Langkah berikutnya melibatkan analisis kebutuhan sistem, di mana proses ini melibatkan observasi terhadap kendala yang dihadapi di PT. XYZ. Hasil dari observasi ini akan menjadi panduan dalam merancang sistem yang dibutuhkan. Selanjutnya, masuk ke tahap perancangan sistem, di mana fokusnya adalah merencanakan bagaimana sistem akan berjalan (Permana *et al.* 2023). Ini melibatkan pembuatan sketsa dan rencana yang menggambarkan elemen-elemen kunci seperti arsitektur, antarmuka pengguna, basis data, dan komponen lainnya dalam satu kesatuan yang terintegrasi dan berfungsi dengan baik (Wijayanto & Susetyo 2022). Dalam penelitian ini, desain sistem melibatkan penggunaan *Unified Modeling Language* (UML) sebagai alat untuk membuat diagram. UML merupakan pendekatan visual untuk mendokumentasikan, menjelaskan, dan membangun sistem perangkat lunak (Kurniawan 2018). *Use Case Diagram* pada Gambar 2 mengilustrasikan interaksi antara aplikasi yang sedang dikembangkan dengan aktor atau pelaku yang terlibat dalam sistem tersebut. *Use Case Diagram* ini secara khusus memperlihatkan interaksi antara sistem dan aktor (Wijayanto & Susetyo 2022).



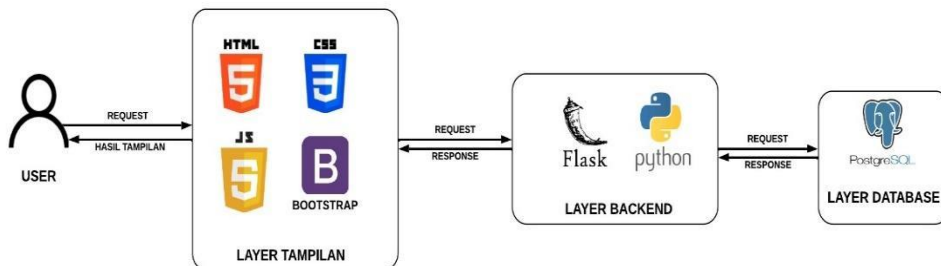
Gambar 2. *Use Case Diagram*

*Use Case Diagram* pada gambar 2 menunjukkan keberadaan satu pelaku. Pelaku ini dapat melakukan dua kegiatan utama. Pertama, kegiatan melihat laporan, yang mencakup aksi meninjau diagram-diagram yang memuat informasi tentang *performance* penjualan PT. XYZ. Kedua, kegiatan mengunduh laporan, yang merujuk pada proses mengambil laporan yang sudah tersusun dengan rapi dalam format presentasi PowerPoint (.pptx). Setelah dibuat *Use Case diagram*, lalu dijabarkan lagi dalam diagram aktivitas yang menggambarkan logika prosedural, proses bisnis, serta alur kerja sistem dalam aplikasi yang telah dibuat (Muhamad, Abdulmonim & Alathari 2019). Langkah berikutnya adalah membangun aplikasi berdasarkan desain sistem yang telah disusun sebelumnya. Pembangunan aplikasi ini melibatkan implementasi Flask *framework* sebagai kerangka kerja utama, Python-pptx untuk otomatisasi penyusunan laporan, dom-to-image untuk integrasi data visual, Bootstrap sebagai kerangka antarmuka, dan PostgreSQL untuk membuat database. Pada tahap akhir, dilakukan pengujian sistem untuk mengevaluasi kinerja aplikasi yang telah dikembangkan, dengan tujuan untuk memastikan apakah aplikasi beroperasi sesuai dengan harapan atau tidak (Wijayanto & Susetyo 2022).

### 3. Hasil dan Pembahasan

Hasil dari penelitian ini adalah implementasi Python-pptx dan dom-to-image dalam pembangunan sistem otomatisasi laporan berbasis web di PT. XYZ. Dengan penerapan kedua teknologi tersebut, perusahaan dapat mengatasi tantangan dalam penyusunan laporan presentasi PowerPoint (PPT) yang sebelumnya dilakukan secara manual. Proses otomatisasi menggunakan Python-pptx memungkinkan pembuatan slide secara dinamis berdasarkan data yang diterima melalui permintaan HTTP. Selain itu, dom-to-image digunakan untuk mengintegrasikan data visual ke dalam

laporan dengan mengonversi simpul DOM menjadi gambar vektor atau raster. Hasilnya adalah sebuah sistem yang mampu menghasilkan laporan presentasi PowerPoint secara cepat dan efisien.



Gambar 3. Arsitektur Sistem

Gambar 3 merupakan struktur sistem yang telah dikembangkan sebelumnya, yang menggunakan Python-pptx dan dom-to-image. Pada lapisan antarmuka pengguna, berbagai teknologi digunakan untuk menciptakan pengalaman pengguna yang menarik dan responsif. JavaScript bertindak sebagai perantara untuk menerima input dari pengguna dan mengirimkannya ke lapisan backend, serta untuk memanipulasi tampilan aplikasi. HTML, CSS, dan Bootstrap digunakan untuk merancang tampilan aplikasi. Lapisan backend bertanggung jawab atas pemrosesan dan pengolahan permintaan pengguna. Proses dimulai dengan pengguna mengirimkan permintaan, yang kemudian disaring dan diproses sebelum dikirim ke lapisan database untuk menyimpan atau mengambil data yang diperlukan. Respons dari lapisan database kemudian diproses lagi sebelum dikirim kembali ke lapisan antarmuka pengguna. Python dan Flask Framework digunakan sebagai teknologi backend untuk menerapkan logika bisnis dan memproses data. Lapisan database menggunakan PostgreSQL untuk manajemen data. Lapisan ini dikelola langsung oleh lapisan backend untuk memastikan keamanan dan integritas data secara keseluruhan. Dengan menggunakan kombinasi ini, aplikasi dapat memanfaatkan Python-pptx dan dom-to-image untuk membuat laporan presentasi dengan cepat dan efisien, sambil mempertahankan keamanan dan konsistensi data yang diperlukan.

Kode Program 1. Potongan fungsi mengirim data

```

function generate_into_ppt() {
  const pptButton = document.getElementById("pptButton");
  pptButton.addEventListener("click", async function() {
    const imageElements = Array.from(document.querySelectorAll(".ppt-screens"));
    const dataArray = [];
    for (const imageElement of imageElements) {
      try {
        const dataUrl = await domtoimage.toPng(imageElement);
        const titleElement = imageElement.querySelector("#ppt-title");
        const nameCategory = imageElement.querySelector("#category-name");
        const data = {
          category: nameCategory.textContent.trim(),
          title: titleElement.textContent.trim(),
          image: dataUrl
        };
        dataArray.push(data);
      }
    }
  });
}
  
```

Kode program 1 merupakan potongan fungsi untuk mengirim data yang akan dibuat ke PPT dari *frontend* ke *backend*. Diawali dengan inisialisasi tombol pada halaman web PT. XYZ dengan ID "pptButton" dan menetapkan *event listener* untuk mengaktifkan proses otomatisasi. Fungsi tersebut secara iteratif mengambil elemen gambar berdasarkan kelas "ppt-screens", menangkap gambar dari setiap elemen menggunakan dom-to-image, dan menyusun data kategori, judul, dan URL gambar. Proses kerja dom-to-image pada penelitian ini melibatkan pengambilan tangkapan layar dari elemen-elemen tertentu dalam halaman web HTML yang berisi informasi visual yang akan disertakan dalam

laporan. Metode ini memungkinkan isolasi elemen-elemen khusus dari halaman web untuk dianggap sebagai objek yang akan di-*capture* oleh dom-to-image. Kelebihan dom-to-image terlihat terutama pada kecepatan eksekusinya, yang lebih singkat daripada metode sejenis seperti Html2Canvas. Dom-to-image memerlukan waktu sekitar 1-2 detik untuk menghasilkan gambar ukuran biasa, sementara untuk gambar ukuran besar, waktu eksekusinya dapat mencapai 5 detik. Meskipun memerlukan sedikit lebih banyak waktu untuk ukuran yang lebih besar, kecepatan eksekusi yang relatif singkat ini tetap menghasilkan efisiensi dalam proses otomatisasi pembuatan laporan berbasis web. Keberhasilan dom-to-image dalam memberikan keseimbangan antara kecepatan dan kualitas tangkapan layar menjadi aspek yang penting dalam memastikan responsifitas dan performa optimal dalam penyusunan laporan presentasi PowerPoint.

### Kode program 2. Rute

```
@app.route('/index/generate_into_ppt', methods=['POST'])
def generateToPpt():
    db = pptreport()
    return db.generate_to_ppt()
```

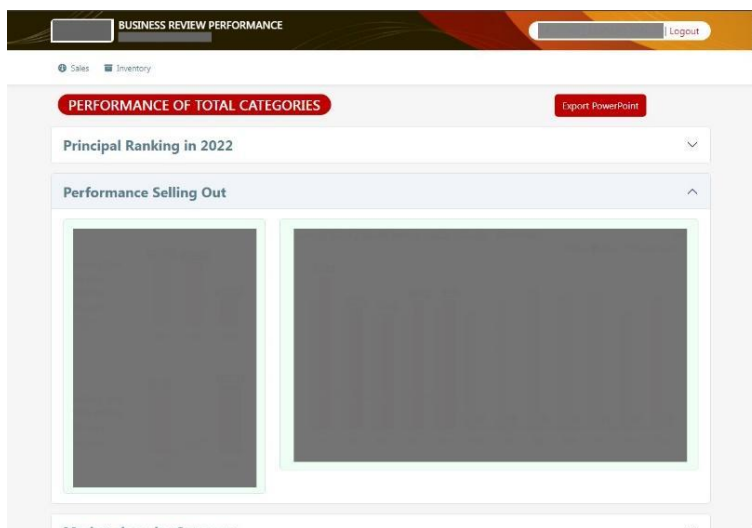
Kode program 2 merupakan rute yang diprogram untuk menangani permintaan POST. Saat diakses, fungsi tersebut membuat objek dari kelas "pptreport" dan memanggil metode "generate\_to\_ppt". Kelas tersebut menyimpan logika operasi pembuatan laporan PowerPoint. Rute ini berperan sebagai antarmuka untuk memicu proses pembuatan laporan PowerPoint melalui permintaan POST.

### Kode program 3. Potongan fungsi pembuatan powerpoint

```
def generate_into_ppt(self):
    datas = request.get_json()
    try:
        prs = Presentation()
        for data in datas:
            item_title = data.get("title", "")
            item_image = data.get("image", "")
            slide = prs.slides.add_slide(prs.slide_layouts[6])
            background = slide.background
            fill = background.fill
            fill.fore_color.rgb = RGBColor(37, 150, 190)
            textBoxTitle = slide.shapes.add_textbox(Inches(2), Inches(0.2),
            widthTitle, Inches(1))
            textFrameTitle = textBoxTitle.text_frame
            textFrameTitle.text = item_title
            for paragraph in textFrameTitle.paragraphs:
                for run in paragraph.runs:
                    run.font.bold = True
                    run.font.size = Pt(20)
                    run.font.name = "Arial"
            image_bytes = base64.b64decode(item_image.split(',')[1])
            if len(image_bytes) > 0:
                image_stream = BytesIO(image_bytes)
                pic = slide.shapes.add_picture(image_stream, Inches(0.5), Inches(2), Inches(9))
```

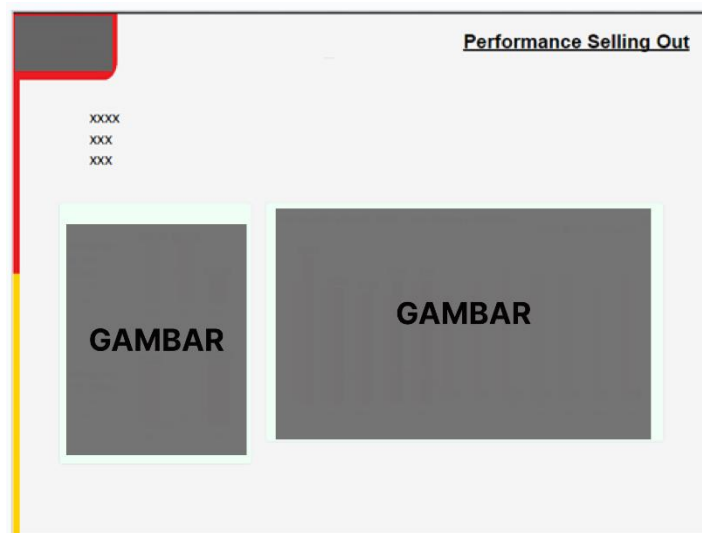
Potongan kode yang diberikan merupakan bagian krusial dari fungsi generate\_into\_ppt yang bertanggung jawab dalam membuat presentasi PowerPoint menggunakan modul Python-pptx. Fungsi ini menerima data JSON melalui permintaan HTTP di Flask, di mana setiap entri data berisi informasi tentang sebuah slide, termasuk judul (item\_title) dan gambar (item\_image). Fungsi ini menginisialisasi presentasi PowerPoint baru, menetapkan tata letak dasar slide, dan melakukan iterasi untuk setiap entri data. Untuk setiap entri data, slide baru ditambahkan ke presentasi, dan warna latar belakangnya disesuaikan. TextBox dibuat untuk menampilkan judul pada slide, dengan atribut formatting tertentu seperti ukuran font, ketebalan, dan jenis font. Fungsi ini kemudian men-*decode* data gambar dari format base64, memeriksa apakah data gambar valid, dan jika ya, menambahkan gambar ke slide pada posisi dan ukuran yang ditentukan.

Potongan kode ini menunjukkan proses otomatis untuk membuat slide PowerPoint secara dinamis berdasarkan data JSON yang diberikan, meningkatkan efisiensi pembuatan presentasi. Penggunaan pustaka Python-pptx memudahkan manipulasi elemen presentasi, memastikan format dan tata letak yang konsisten di seluruh slide. Presentasi PowerPoint yang dihasilkan adalah representasi visual dari data yang diolah, memberikan alat yang berharga untuk komunikasi dan kolaborasi yang efisien dalam konteks bisnis.



Gambar 4. Tampilan Aplikasi

Gambar 4 menampilkan antarmuka aplikasi yang memuat berbagai diagram yang mencerminkan kinerja penjualan dari PT. XYZ. Terdapat tombol "Export PowerPoint" yang jika diklik, akan memicu proses pembuatan laporan dalam format PowerPoint yang dapat diunduh oleh pengguna.



Gambar 5. Slide Laporan Powerpoint

Gambar 5 merupakan slide laporan Powerpoint yang dihasilkan dari program yang dibuat. Setiap slide dalam laporan PowerPoint memiliki struktur yang terdiri dari judul yang menjelaskan konten slide dan gambar yang merepresentasikan informasi *performance* penjualan PT. XYZ.

Tabel 1. Pengujian *Blackbox*

Pengujian	Hasil yang diharapkan	Hasil Pengujian	Status Pengujian
Keberhasilan Konversi Data DOM menjadi Gambar	Gambar hasil konversi sesuai dengan tampilan DOM	Gambar berhasil dikonversi dari simpul DOM menjadi vektor atau raster	Valid
Mengirim data ke backend	Data berupa JSON terkirim ke backend	Data terkirim dan diterima dengan baik	Valid
Membuat PowerPoint dari data yang diterima	Terbentuk powerpoint yang sesuai format yang diinginkan	PowerPoint terbentuk sesuai format, hasilnya dikirim dan dapat di-download	Valid
Kecepatan Konversi Data DOM menjadi Gambar	Proses konversi berlangsung dalam 1-2 detik untuk gambar ukuran biasa dan 5 detik untuk gambar ukuran besar	Konversi data DOM sesuai dengan waktu yang diharapkan	Valid
Kesesuaian Presentasi PowerPoint dengan Data	Presentasi PowerPoint mencerminkan data yang diterima	Presentasi sesuai dengan data, memastikan kesesuaian	Valid

Tabel 1 merupakan hasil pengujian sistem yang dilakukan oleh tim *quality assurance* PT. XYZ. Metode yang digunakan adalah *blackbox testing*. Berdasarkan hasil pengujian, seluruh tahap pengujian menunjukkan hasil yang sesuai dengan yang diharapkan, dengan status pengujian yang valid. Setelah melalui *User Accepting Test* (UAT) dengan pengguna, dokumen hasil pengujian menyatakan bahwa sistem telah memenuhi kebutuhan pengguna berdasarkan *system requirement* yang telah ditentukan. Oleh karena itu, aplikasi dianggap siap digunakan, membuktikan keberhasilannya dalam menjamin konsistensi, meningkatkan efisiensi operasional, dan memperkuat jaringan kerja bisnis yang strategis dengan para mitra.

#### 4. Kesimpulan

Dengan menerapkan Python-pptx dan dom-to-image, penelitian ini berhasil mengotomatiskan pembangunan laporan presentasi PowerPoint di PT. XYZ, sesuai dengan tujuan penelitian untuk meningkatkan efisiensi dalam proses penyusunan laporan. Hasil temuan menunjukkan bahwa sistem otomatisasi ini berhasil mengatasi tantangan manual yang sebelumnya dihadapi perusahaan, memungkinkan pembuatan slide secara dinamis berdasarkan data yang diterima melalui permintaan HTTP. Dengan adanya otomatisasi, perusahaan dapat mengurangi pemborosan waktu, mengurangi risiko kesalahan, dan meningkatkan efisiensi operasional secara signifikan. Kode program, modul Python-pptx, dan dom-to-image menjadi kunci dalam kesuksesan implementasi ini, memberikan fleksibilitas desain yang diperlukan dan kecepatan yang sesuai. Pengujian sistem, termasuk *blackbox testing* dan *User Accepting Test* (UAT), membuktikan konsistensi dan kepuasan pengguna. Kesimpulan ini mengonfirmasi bahwa penerapan Python-pptx dan dom-to-image efektif dalam mencapai tujuan

otomatisasi laporan berbasis web di PT. XYZ, membuka potensi untuk pengembangan lebih lanjut dan peningkatan lanjutan ke depannya.

## 5. Ucapan Terima Kasih

Artikel ini disusun sebagai bagian dari penyelesaian Program Studi Strata I Teknik Informatika di Universitas Kristen Satya Wacana. Meskipun menghadapi berbagai tantangan dalam proses penulisan, penulis berhasil menyelesaikan artikel ini tepat waktu berkat bimbingan dan dukungan dari berbagai pihak. Penulis ingin menyampaikan rasa terima kasih kepada mentor-mentor di PT. XYZ yang selalu memberikan bimbingan dan bantuan dalam pengembangan sistem serta penelitian ini. Juga, terima kasih kepada keluarga yang memberikan doa, semangat, dan harapan, teman-teman yang selalu mendampingi, serta semua pihak yang terlibat dalam proses penulisan ini. Rasa terima kasih penulis untuk kontribusi dan dukungan yang telah diberikan.

## 6. Daftar Pustaka

- Borade, J. G., Kiwelekar, A. W., & Netak, L. D. (2022). Automated Grading of PowerPoint Presentations Using Latent Semantic Analysis. *Revue d'Intelligence Artificielle*, 36(2), 305.
- Canny, S 2023, *Python-ptx Documentation*.
- Galahartlambang, Y., Khotiah, T., & Jumain, J. (2021, August). Analisa Performa Aplikasi Web Berbasis Manipulasi DOM dan Virtual DOM. In *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)* (Vol. 5, No. 1, pp. 036-040). <https://doi.org/10.29407/inotek.v5i1.900>.
- Guntara, R. G. (2023). Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab. *ULIL ALBAB: Jurnal Ilmiah Multidisiplin*, 2(6), 2091-2100. DOI: <https://doi.org/10.56799/jim.v2i6.1578>.
- Habsy, B. A. (2017). Seni memahami penelitian kuliitatif dalam bimbingan dan konseling: studi literatur. *Jurnal Konseling Andi Matappa*, 1(2), 90-100.
- Kiftya, A. M., & Bachtiar, E. (2023). Evaluasi Penerapan Boundary System dalam Hubungan B2B dengan Pemasok pada Perusahaan Retail PT. X. *Jurnal Manajemen dan Organisasi*, 14(3). DOI: <https://doi.org/10.29244/jmo.v14i3.47658>.
- Kiwelekar, A. W., & Netak, L. D. (2022). Automatic grading of student's presentation skills based on powerpoint presentation and audio. *U. Porto journal of engineering*, 8(2), 95-107. DOI: [https://doi.org/10.24840/2183-6493\\_008.002\\_0008](https://doi.org/10.24840/2183-6493_008.002_0008).
- Kurniawan, T. A. (2018). Pemodelan use case (UML): evaluasi terhadap beberapa kesalahan dalam praktik. *J. Teknol. Inf. dan Ilmu Komput*, 5(1), 77.
- Macklon, F., Vigiato, M., Romanova, N., Buzon, C., Paas, D., & Bezemer, C. P. (2023). A Taxonomy of Testable HTML5 Canvas Issues. *IEEE Transactions on Software Engineering*. DOI: <https://doi.org/10.1109/TSE.2023.3270740>.

- Megawaty, D. A., Alita, D., & Dewi, P. S. (2021). Penerapan Digital Library Untuk Otomatisasi Administrasi Perpustakaan. *Journal of Social Sciences and Technology for Community Service (JSSTCS)*, 2(2), 121-127. DOI: <https://doi.org/10.33365/jsstcs.v2i2.1366>.
- Muhamad, Z. H., Abdulmonim, D. A., & Alathari, B. (2019). An integration of uml use case diagram and activity diagram with Z language for formalization of library management system. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(4).
- Nakamura, S., Niibori, M., & Kamada, M. (2022, August). Web Application with Built-in Remote Support Capability by Means of Voice Communication and Screen Sharing. In *International Conference on Network-Based Information Systems* (pp. 424-429). Cham: Springer International Publishing. DOI: [https://doi.org/10.1007/978-3-031-14314-4\\_44](https://doi.org/10.1007/978-3-031-14314-4_44).
- Permana, A. A., Agustriawan, D., Johan, M. E., Fianty, M. I., Sanjaya, S. A., Sutomo, R., ... & Waworuntu, A. (2023). Memahami Software Development Life Cycle.
- Prabowo, M. Y., Firdaus, A., Damanik, D. A., & Nurtanto, T. D. (2022). Automasi Pembuatan Laporan Analisis dengan R Markdown. *Kybernan: Jurnal Studi Pemerintahan*, 5(1), 49-61. DOI: <https://doi.org/10.35326/kybernan.v5i1.2121>.
- Wijayanto, C., & Susetyo, Y. A. (2022). Implementasi Flask Framework Pada Pembangunan Aplikasi Sistem Informasi Helpdesk (SIH). *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 7(3), 858-868. DOI: <https://doi.org/10.29100/jupi.v7i3.3161>.