BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama melaksanakan kegiatan magang, posisi yang ditempati adalah sebagai *DCC Developer* di Tim Metrologi Digital, yang berada di bawah naungan Deputi Bidang Standar Nasional Satuan Ukuran (SNSU) - Badan Standardisasi Nasional (BSN). Tanggung jawab utama adalah mengembangkan bagian *front-end* dari sistem DCC yang sedang dibangun, di bawah supervisi dari Ibu Hayati Amalia selaku koordinator Tim Metrologi Digital.

Dalam pelaksanaan tugas, dilakukan kerja sama dengan seorang rekan magang yang bertanggung jawab pada bagian *back-end* dari sistem yang sama. Kolaborasi antara *front-end* dan *back-end* dilakukan melalui *platform* GitHub, yang digunakan untuk menyimpan *source code*, melakukan *version control*, serta mengelola *push* dan *pull request*. Penggunaan GitHub memudahkan sinkronisasi perubahan kode dan mendukung pengembangan sistem secara terstruktur dan kolaboratif.

Koordinasi dan komunikasi dengan *supervisor* dan Tim Metrologi Digital dilakukan secara daring melalui grup WhatsApp, yang biasanya digunakan untuk merencanakan waktu rapat, berbagi informasi, dan mendistribusikan dokumen atau sumber daya yang dibutuhkan. Selain itu, rapat tim rutin diadakan setiap pekan untuk menyampaikan progres pekerjaan, berdiskusi, memperoleh arahan lebih lanjut, serta menerima umpan balik dari *supervisor* dan tim pengembang terkait kualitas dan arah pengembangan sistem.

3.2 Tugas yang Dilakukan

Selama menjalani kegiatan magang, diberikan tanggung jawab untuk terlibat langsung dalam pengembangan aplikasi *website* DCC Level 2, yang dilaksanakan oleh Tim Metrologi Digital di Deputi Bidang SNSU-BSN. Tugas-tugas yang dilakukan meliputi:

1. Mengembangkan *front-end* dari *website*
   Bertanggung jawab dalam merancang dan membangun antarmuka pengguna atau *user interface* (UI). Pengembangan ini mencakup:
   - Membuat formulir digital untuk pengisian data sertifikat kalibrasi.
   - Menerapkan validasi input untuk memastikan kelengkapan data yang diisi.
   - Melakukan revisi dan perbaikan antarmuka berdasarkan umpan balik dari tim.
2. Membuat kode Python untuk pengolahan data/*file*
   Kode Python dikembangkan untuk mensimulasikan proses pengolahan data dan *file* yang akan digunakan dalam sistem *back-end* aplikasi *website*. Simulasi ini bertujuan untuk memastikan bahwa alur logika dan hasil pemrosesan sudah sesuai sebelum diintegrasikan ke dalam *back-end* aplikasi web oleh rekan magang. Fungsionalitas yang dikembangkan antara lain:
   - Memasukkan input dari formulir ke *template* sertifikat dengan format dokumen Word.
   - Mengubah input formulir menjadi format XML.
   - Menyalin tabel dari *file* Excel ke dalam *template* sertifikat.
   - Mengubah data tabel di Excel menjadi format XML.
   - Menyisipkan *file* XML sebagai lampiran dalam *file* PDF.
   - Mengubah *file* XML yang tertempel di *file* PDF menjadi tabel dalam format Excel.
   - Mengubah input formulir menjadi PDF menggunakan *template* *Hypertext Markup Language* (HTML).

Selama proses pengerjaan, juga terlibat dalam berdiskusi bersama tim terkait pengembangan sistem, melakukan pengujian fungsionalitas aplikasi, dan melaporkan progres pekerjaan.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang dilakukan mulai tanggal 10 Februari 2025 hingga 20 Juni 2025, dan diuraikan pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Perencanaan proyek dengan mempelajari kebutuhan pengguna, menentukan teknologi, dan merancang sistem.
2	Pengembangan kode Python untuk pengolahan data dan *file*.
3	Perancangan desain UI di Figma.
4	Pengembangan *front-end* formulir *generator*.
5	Implementasi kode Python untuk menyalin tabel hasil di Excel ke *template* di Word.
6	Menambahkan editor matematika LaTeX dan mengubah *card* ”Hasil”.
7	Pengembangan kode Python untuk mendeteksi isi tabel di Excel dan menjadikannya XML.
8	Integrasi iMathEQ pada kolom rumus di formulir.
9	Pengembangan *front-end* untuk *importer*.
10	Membuat *language switcher* pada *website* untuk bahasa Indonesia dan Inggris.
11	Pengembangan kode Python untuk *importer*.
12	Pengembangan halaman *login*.
13	Menambahkan tahap *comment* di formulir *generator*.
14	Revisi *front-end*.
15	Mengubah *card* kondisi lingkungan.
16	Pengembangan halaman *about*.
17	Pengembangan *template* sertifikat di HTML untuk PDF.
18	Pengembangan halaman *register*

3.3.1 Perencanaan Proyek

Sebelum proyek dieksekusi, tahap perencanaan dilakukan terlebih dahulu. Tahapan ini diawali dengan pengkajian dan analisis kebutuhan pengguna untuk memahami fitur dan fungsionalitas yang dibutuhkan. Berdasarkan hasil analisis tersebut, ditentukan teknologi yang sesuai untuk digunakan. Setelah itu, alur sistem dari aplikasi web dirancang sebagai dasar implementasi.

A Kebutuhan Pengguna

Para pengguna sistem DCC membutuhkan berbagai fitur berikut pada aplikasi web.

1. *Generator*
   Fitur ini menyediakan formulir isian serta fasilitas unggah *file* Excel. *File* Excel berisi tabel hasil kalibrasi, sementara data lainnya diinput melalui formulir yang tersedia. Tersedia juga editor matematika yang mendukung format LaTeX dan MathML. Sistem akan memproses dan mengintegrasikan data dari formulir, editor matematika, dan *file* Excel ke dalam *file* DCC, yaitu dalam format PDF dan XML. Setelah formulir diisi, pengguna mengirimnya dan sistem akan menghasilkan *file* DCC berupa PDF lengkap dengan lampiran *file* XML.

2. *Importer*
   Fitur ini menyediakan fasilitas untuk mengunggah *file* DCC dalam format PDF yang dilengkapi dengan lampiran XML. Setelah *file* diunggah, sistem akan mengekstrak dan membaca data dari *file* XML tersebut. Hasil keluaran dari proses ini adalah *file* Excel yang memuat seluruh data yang terdapat dalam XML.

3. Autentikasi Akun
   Fitur ini memungkinkan pengguna untuk mendaftarkan akun baru serta masuk ke akun tersebut untuk mengakses sistem. Dengan adanya autentikasi, hanya pengguna yang terverifikasi yang dapat menggunakan fitur-fitur pada *website*, sehingga meningkatkan keamanan dan personalisasi layanan.

4. Pengalih bahasa
   Fitur ini memungkinkan pengguna untuk mengubah bahasa antarmuka pada *website*. Bahasa yang tersedia adalah bahasa Indonesia dan bahasa Inggris. Fitur ini meningkatkan aksesibilitas sistem bagi pengguna dari dalam maupun luar negeri.

B Teknologi yang Digunakan

Dalam pengembangan aplikasi web untuk DCC Level 2, berikut adalah teknologi yang digunakan.

- Aplikasi

    1. Visual Studio Code
       Selain ringan, aplikasi *code editor* ini mendukung banyak bahasa pemrograman, termasuk bahasa-bahasa yang dibutuhkan pada pengembangan web DCC.
    2. Jupyter Notebook
       Code editor untuk bahasa Python ini dipakai untuk menulis kode Python yang berfungsi dalam pengolahan data dan *file*. Kode ini selanjutnya akan diterapkan pada *back-end*.
    3. GitHub Desktop
       Aplikasi *version control system* ini digunakan untuk menyinkronkan proyek antara anggota tim magang. Integrasinya dengan Visual Studio Code memudahkan pengelolaan versi kode secara kolaboratif.

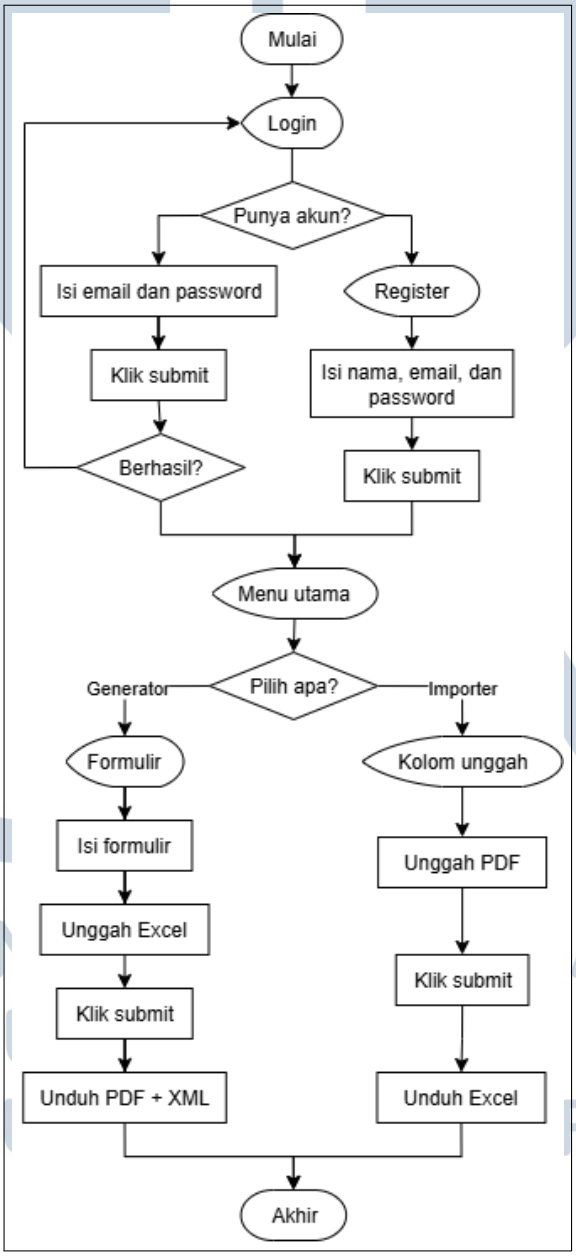
- *Front-End*

    1. Next.js
       Framework React ini mendukung pembuatan aplikasi web yang dinamis dan terstruktur. Next.js dipilih karena memiliki fitur API Routes yang memungkinkan *back-end* ringan tanpa perlu *server* terpisah.
    2. shadcn/ui
       Component library UI berbasis Tailwind CSS ini digunakan untuk mempercepat proses pengembangan antarmuka dan menjaga konsistensi desain.
    3. TypeScript
       Bahasa pemrograman yang merupakan pengembangan dari JavaScript ini digunakan untuk meningkatkan keandalan kode. TypeScript membantu mendeteksi kesalahan sejak tahap pengembangan serta mempermudah pemeliharaan kode.

- Pengolahan *file* dan data: Python
  Bahasa pemrograman ini dipilih karena memiliki banyak *library* yang mendukung manipulasi data dan *file*.

C Rancangan Sistem

Rancangan alur pengalaman dan interaksi pengguna disajikan dalam bentuk *user flow diagram* (diagram alur pengguna). Diagram ini menggambarkan tahapan yang dilalui pengguna saat berinteraksi dengan aplikasi web DCC, termasuk halaman-halaman yang dikunjungi. Visualisasi ini berfungsi sebagai panduan dalam pengembangan antarmuka pengguna (*front-end*). Diagram alur pengguna dapat dilihat pada Gambar 3.1.



Gambar 3.1. *User flow diagram* dari *front-end*

Pengalaman pengguna dimulai saat mengunjungi halaman *login*. Di halaman ini, pengguna memasukkan *email* dan kata sandinya, kemudian menekan tombol *submit*. Jika data yang dimasukkan tidak valid, proses *login* akan gagal dan pengguna tetap berada di halaman *login*. Sebaliknya, jika data valid dan proses *login* berhasil, pengguna akan diarahkan ke halaman menu utama. Apabila pengguna belum memiliki akun, maka pengguna dapat berpindah ke halaman *register* untuk mendaftarkan akunnya.

Di halaman menu utama, tersedia dua opsi, yaitu *generator* dan *importer*. Jika pengguna memilih opsi generator, maka pengguna akan diarahkan ke halaman formulir *generator*. Di halaman ini, pengguna mengisi formulir dengan data yang diperlukan dalam pembuatan sertifikat kalibrasi. Pengguna juga mengunggah *file* Excel yang berisi hasil kalibrasi. Setelah semua data terisi dan tombol *submit* ditekan, sistem akan menghasilkan *file* DCC berupa *file* PDF yang dilampirkan *file* XML.

Apabila pengguna memilih opsi *importer*, maka akan diarahkan ke halaman unggah *importer*. Di halaman ini, pengguna mengunggah *file* DCC berupa *file* PDF yang dilengkapi dengan *file* XML, kemudian menekan tombol *submit*. Jika *file* XML tidak ditemukan dalam dokumen yang diunggah, sistem akan meminta pengguna untuk mengunggah ulang *file* PDF tersebut. Jika *file* XML berhasil ditemukan, pengguna kemudian mengunggah *file* Excel yang dihasilkan.

3.3.2 Pengolahan Data dan File

Pada tahap ini, dikembangkan kode simulasi pengolahan data dan *file* dasar. Simulasi pengolahan data dan *file* dilakukan menggunakan bahasa pemrograman Python dan dijalankan pada *platform* Jupyter Notebook. Kode yang dikembangkan oleh kemudian diserahkan kepada rekan magang untuk diintegrasikan ke dalam *back-end* aplikasi web.

A Mengubah Input Formulir menjadi File XML

Karena program yang dikembangkan bersifat simulasi dan tidak terhubung langsung dengan sistem web, maka input data dibuat secara manual melalui kode program. Contoh kode input dapat dilihat pada Kode 3.1.

```
jenis = 'Digital Multimeter'
merek = 'Fluke'
tipe = '8508A'
item_issuer = 'manufacturer'
seri_item = 'MY53010340'
id_lain = '-'
```

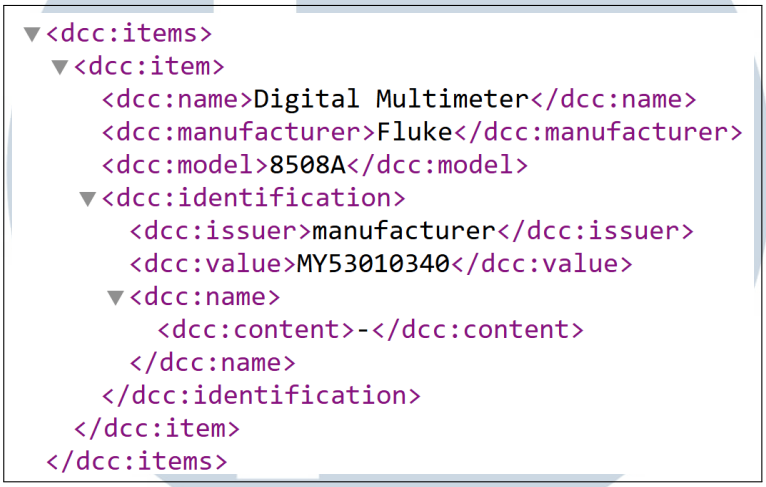
Kode 3.1: Contoh input

Program ini menggunakan *library* `yattag` untuk membangun struktur XML yang sesuai dengan standar yang ditetapkan oleh *Physikalisch-Technische Bundesanstalt* (PTB), yaitu institusi metrologi nasional di Jerman. Objek `doc`, `tag`, dan `text` diinisialisasi menggunakan `Doc().tagtext()`. Selanjutnya, struktur XML dibangun dengan membuat elemen utama `dcc:items` yang berisi satu elemen `dcc:item`. Di dalamnya, beberapa elemen anak ditambahkan, seperti `dcc:name` berisi nilai dari variabel `jenis`, `dcc:manufacturer` berisi `merek`, dan `dcc:model` berisi `tipe`. Kemudian, elemen `dcc:identification` dibuat dengan tiga sub-elemen: `dcc:issuer` berisi `item_issuer`, `dcc:value` berisi `seri_item`, serta `dcc:name`. `dcc:name` memiliki anak elemen `dcc:content` yang berisi `id_lain`. Setelah struktur XML selesai dibentuk, isinya diformat menggunakan indentasi dengan fungsi `indent` sebanyak tiga spasi. Hasil XML yang telah diformat ini kemudian ditulis ke dalam *file* bernama `Form to XML.xml`. Kode implementasi ditunjukkan pada Kode 3.2 dan cuplikan *file* XML yang dihasilkan dapat dilihat pada Gambar 3.2.

```
from yattag import Doc, indent

doc, tag, text = Doc().tagtext()

with tag('dcc:items'):
    with tag('dcc:item'):
        with tag('dcc:name'): text(jenis)
        with tag('dcc:manufacturer'): text(merek)
        with tag('dcc:model'): text(tipe)
        with tag('dcc:identification'):
            with tag('dcc:issuer'): text(item_issuer)
            with tag('dcc:value'): text(seri_item)
            with tag('dcc:name'):
                with tag('dcc:content'): text(id_lain)

result = indent(
    doc.getvalue(),
```

```
    indentation='    '
)

with open('Form to XML.xml', "w") as f:
    f.write(result)
```

Kode 3.2: Pembuatan *file* XML



Gambar 3.2. *file* XML

B Memasukkan Input Formulir ke Template Sertifikat di Word

Contoh kode input dapat dilihat pada Kode 3.1. Algoritma ini menggunakan kelas `DocxTemplate` dari *library* `docxtpl`. Kemudian, objek `doc` dibuat dari kelas `DocxTemplate` dengan memuat *file* `template DCC.docx`, yaitu *file* Word yang berfungsi sebagai *template* sertifikat. Baris kode yang memuat *template* tersebut ditunjukkan pada Kode 3.3.

```
from docxtpl import DocxTemplate
doc = DocxTemplate('template DCC.docx')
```

Kode 3.3: Memuat *template* dokumen Word

Template ini disusun dengan meniru sertifikat yang dihasilkan oleh SPARTA. Bagian untuk input dalam dokumen *template* diisi dengan *placeholder* berupa nama variabel yang ditulis di antara kurung kurawal ganda ("{{ }}"). *Template* tersebut dapat dilihat pada Gambar 3.3.

BADAN STANDARDISASI NASIONAL
Laboratorium Standar Nasional Satuan Ukuran
Laboratory of National Measurement Standards

KAN Komite Akreditasi Nasional
LK-070-IDN

Sertifikat Kalibrasi
Calibration Certificate
Certificate number: {{ certificate }}
Order number: {{ order }}

Deskripsi Objek yang Dikalibrasi/Diukur
Description of object being calibrated or measured

Jenis alat atau objek : {{ jenis }}
Type of instrument or object
Merek/pembuat dan tipe : {{ merek }} {{ tipe }}
Brand/manufacturer and type
Identifikasi alat
Instrument identification
Nomor seri : {{ seri_item }}
Serial number
Identifikasi lain : {{ id_lain }}
Other identification

Identitas Pemilik
Owner's identification

Nama : {{ nama_cust }}
Designation
Alamat : {{ jalan_cust }} {{ no_jalan_cust }}, {{ kota_cust }}, {{ state_cust }}
Address {{ pos_cust }}, {{ negara_cust }}

Pengesahan
Authorization

Pejabat yang mengesahkan : {{ peran_direktur }}
Authorizing officer
Nama : {{ nama_resp_direktur }}
Name NIP {{ nip_resp_direktur }}
Tanggal pengesahan : {{ tgl_pengesahan }}
Date of issue (dd/mm/yyyy)
Jumlah halaman (termasuk halaman ini) :
Total number of pages including this one

Dokumen ini disahkan secara elektronik sesuai peraturan yang berlaku dengan sertifikat dari Balai Sertifikasi Elektronik (BsrE) dan tidak memerlukan tanda tangan atau cap. Dokumen asli dapat diperoleh dengan memindai kode QR di samping ini.
This document is digitally signed. No signature or seal is required. The original document can be obtained by scanning the QR code on the left.

Kalibrasi atau pengukuran yang dilaporkan dalam sertifikat ini tercakup dalam lingkup akreditasi menurut SNI ISO/IEC 17025 oleh Komite Akreditasi Nasional, kecuali dinyatakan dalam badan sertifikat.
The calibration or measurement reported in the certificate is covered in the accredication scope according to SNI ISO/IEC 17025 by the National Accredication Committee of Indonesia, unless marked otherwise in the body of certificate.

Laboratorium Standar Nasional Satuan Ukuran, Badan Standarisasi Nasional (SNSU-BSN)
Laboratory of National Measurement Standards, Nasional Standardization Agency of Indonesia
Kompleks Puspiptek, Gedung 420, Setu, Tangerang Selatan, Banten 15314, Indonesia
Telephone +62-21-7560534, +62-21-7560571, Mobile +62-857-80857833, Web www.bsn.go.id, Email nmi@bsn.go.id

Gambar 3.3. Halaman pertama *template* sertifikat

Setelah *template* dimuat, dibuat sebuah *dictionary* bernama `context` yang memetakan nama-nama *placeholder* dalam dokumen *template* ke variabel-variabel input yang sesuai. *Dictionary* ini digunakan untuk mengisi *template* tersebut secara otomatis. Kode pemetaan ini ditunjukkan pada Kode 3.4.

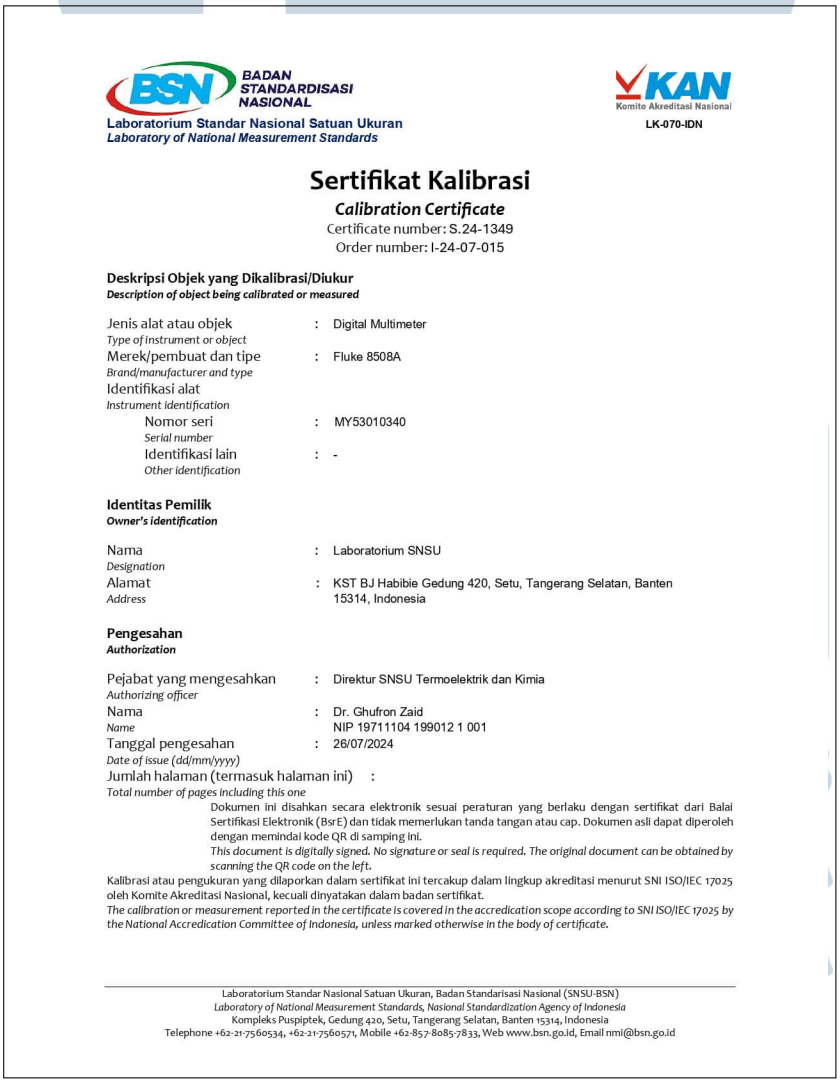
```
context = {
    'jenis': jenis,
    'merek': merek,
    'tipe': tipe,
    'seri_item': seri_item,
    'id_lain': id_lain
}
```

Kode 3.4: Pemetaan *placeholder* dengan input

Berikutnya, fungsi `render` dijalankan untuk menggantikan *placeholder* dalam *template* dengan data input, berdasarkan *dictionary* `context`. Setelah proses pengisian selesai, dokumen yang telah terisi disimpan sebagai *file* baru, misalnya bernama `'Filled Template.docx'`, menggunakan fungsi `save`. Setelah dokumen Word diisi, *file* tersebut dikonversi ke dalam format PDF. Implementasi proses ini ditunjukkan pada Kode 3.5 dan contoh dokumen hasil pengisian *template* dapat dilihat pada Gambar 3.4.

```
doc.render(context)
doc.save('Filled Template.docx')

convert("Filled Template.docx")
```

Kode 3.5: Mengisi dan menyimpan dokumen *template*



BADAN STANDARDISASI NASIONAL
Laboratorium Standar Nasional Satuan Ukuran
Laboratory of National Measurement Standards

KAN Komite Akreditasi Nasional
LK-070-IDN

Sertifikat Kalibrasi
Calibration Certificate
Certificate number: S.24-1349
Order number: I-24-07-015

Deskripsi Objek yang Dikalibrasi/Diukur
Description of object being calibrated or measured

Jenis alat atau objek : Digital Multimeter
Type of instrument or object
Merek/pembuat dan tipe : Fluke 8508A
Brand/manufacturer and type
Identifikasi alat
Instrument identification
Nomor seri : MY53010340
Serial number
Identifikasi lain : -
Other identification

Identitas Pemilik
Owner's identification

Nama : Laboratorium SNSU
Designation
Alamat : KST BJ Habibie Gedung 420, Setu, Tangerang Selatan, Banten 15314, Indonesia
Address

Pengesahan
Authorization

Pejabat yang mengesahkan : Direktur SNSU Termoelektrik dan Kimia
Authorizing officer
Nama : Dr. Ghufron Zaid
Name
NIP 19711104 199012 1 001
Tanggal pengesahan : 26/07/2024
Date of issue (dd/mm/yyyy)
Jumlah halaman (termasuk halaman ini) :
Total number of pages including this one

Dokumen ini disahkan secara elektronik sesuai peraturan yang berlaku dengan sertifikat dari Balai Sertifikasi Elektronik (BsrE) dan tidak memerlukan tanda tangan atau cap. Dokumen asli dapat diperoleh dengan memindai kode QR di samping ini.
This document is digitally signed. No signature or seal is required. The original document can be obtained by scanning the QR code on the left.

Kalibrasi atau pengukuran yang dilaporkan dalam sertifikat ini tercakup dalam lingkup akreditasi menurut SNI ISO/IEC 17025 oleh Komite Akreditasi Nasional, kecuali dinyatakan dalam badan sertifikat.
The calibration or measurement reported in the certificate is covered in the accredication scope according to SNI ISO/IEC 17025 by the National Accredication Committee of Indonesia, unless marked otherwise in the body of certificate.

Laboratorium Standar Nasional Satuan Ukuran, Badan Standarisasi Nasional (SNSU-BSN)
Laboratory of National Measurement Standards, Nasional Standardization Agency of Indonesia
Kompleks Puspiptek, Gedung 420, Setu, Tangerang Selatan, Banten 15314, Indonesia
Telephone +62-21-7560534, +62-21-7560571, Mobile +62-857-8085-7833, Web www.bsn.go.id, Email nmi@bsn.go.id

Gambar 3.4. *Template* sertifikat yang terisi

C Menyisipkan File XML sebagai Lampiran dalam File PDF

Program ini menggunakan *library* `pikepdf` modul `Pdf` dan `AttachedFileSpec` untuk menyisipkan *file* XML ke dalam *file* PDF sebagai lampiran, serta *library* `pathlib` dengan modul `Path` untuk memudahkan manipulasi *path file*. *File* PDF yang akan diberi lampiran diambil dari *path* `E:\Sertifikat.pdf`, sedangkan *file* XML yang akan dilampirkan diambil dari `E:\Form to XML.pdf`. *File* PDF kemudian dibuka menggunakan `Pdf.open()`. Objek *filespec* dibuat dari *file* XML menggunakan `AttachedFileSpec.from_filepath()`, dan ditambahkan ke dalam atribut `attachments` dari objek PDF dengan nama *file* XML sebagai kunci. PDF yang telah dimodifikasi disimpan ke *path* `E:\PDF + XML.pdf`. Kode program ini ditampilkan pada Kode 3.6.

```
from pikepdf import Pdf, AttachedFileSpec
from pathlib import Path

pdf_path = Path("E:\\Sertifikat.pdf")
xml_path = Path("E:\\Form to XML.xml")
pdf = Pdf.open(pdf_path)

filespec = AttachedFileSpec.from_filepath(pdf, xml_path)
pdf.attachments[xml_path.name] = filespec

output_pdf_path = Path("E:\\PDF + XML.pdf")
pdf.save(output_pdf_path)
```

Kode 3.6: Penyisipan *file* XML ke dalam *file* PDF

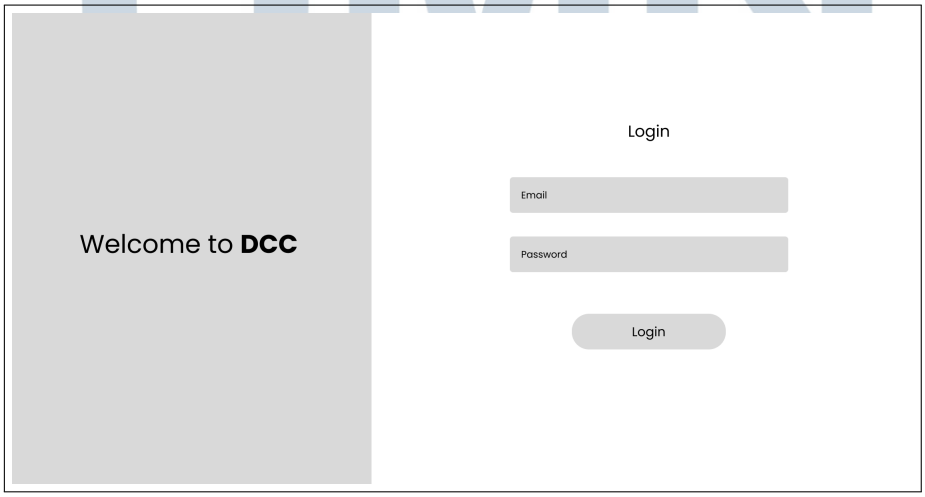
Hasil penyisipan *file* XML dalam PDF dapat dilihat pada Gambar 3.5. Gambar tersebut merupakan tangkapan layar dari aplikasi Adobe Acrobat yang digunakan untuk membuka *file* PDF tersebut. *File* lampiran dapat diakses melalui panel *Attachments*.

Gambar 3.5. Lampiran *file* XML dalam *file* PDF

3.3.3 Rancangan Desain Antarmuka

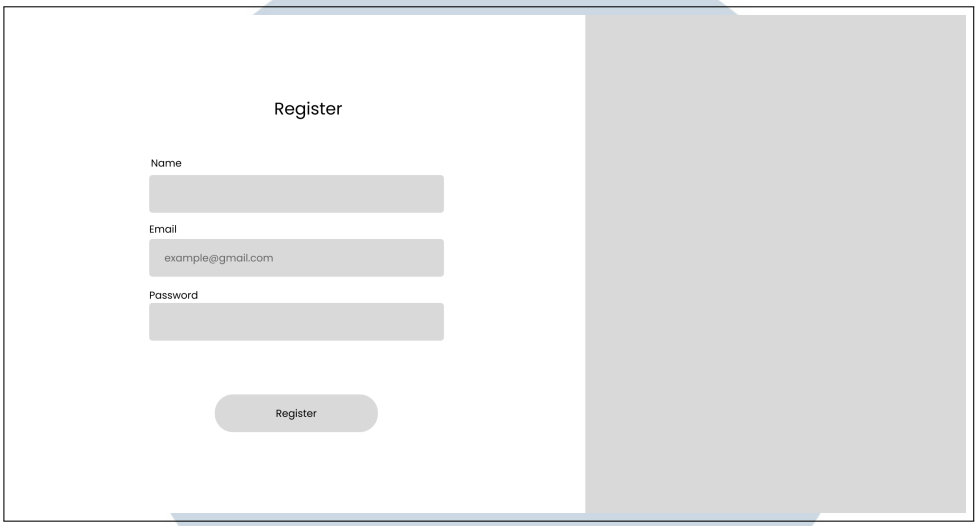
Sebelum mengembangkan *front-end* pada kode, antarmuka pengguna (UI) dirancang terlebih dahulu menggunakan Figma. Perancangan ini bertujuan untuk merencanakan tampilan, tata letak, serta komponen-komponen antarmuka agar pengembangan selanjutnya lebih terstruktur dan efisien. Tidak semua halaman dan komponen dirancang secara menyeluruh di Figma, karena perancangan ini hanya dimaksudkan sebagai gambaran awal atau prototipe kasar. Selain itu, implementasi antarmuka dalam kode tidak sepenuhnya mengikuti desain di Figma, melainkan disesuaikan kembali selama proses pengembangan.

Desain halaman *login* ditampilkan di Gambar 3.40. Halaman ini menampilkan kolom input untuk *email* dan *password*, serta tombol untuk masuk ke dalam aplikasi.



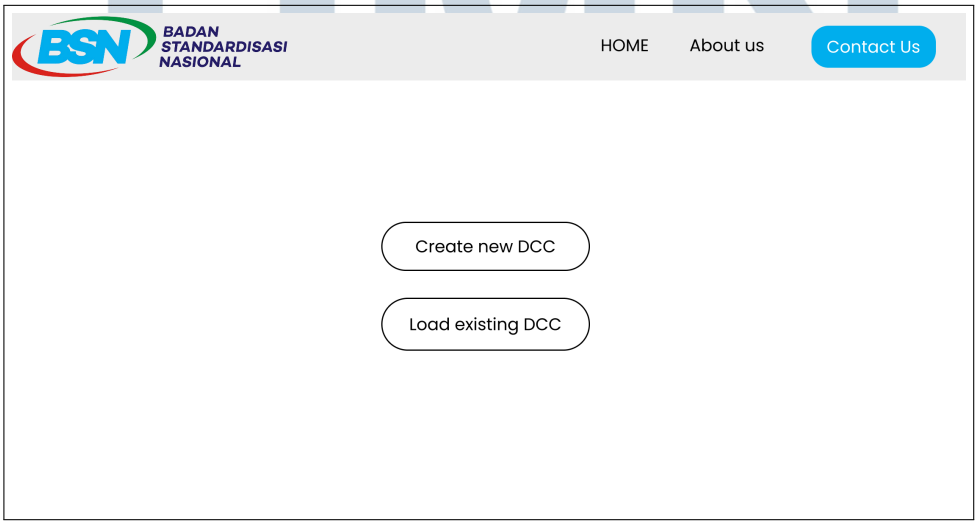
Gambar 3.6. Desain halaman *login*

Desain halaman *register* ditampilkan di Gambar 3.49. Halaman ini menampilkan kolom input untuk nama, *email*, dan *password*, serta tombol untuk mendaftarkan akun.

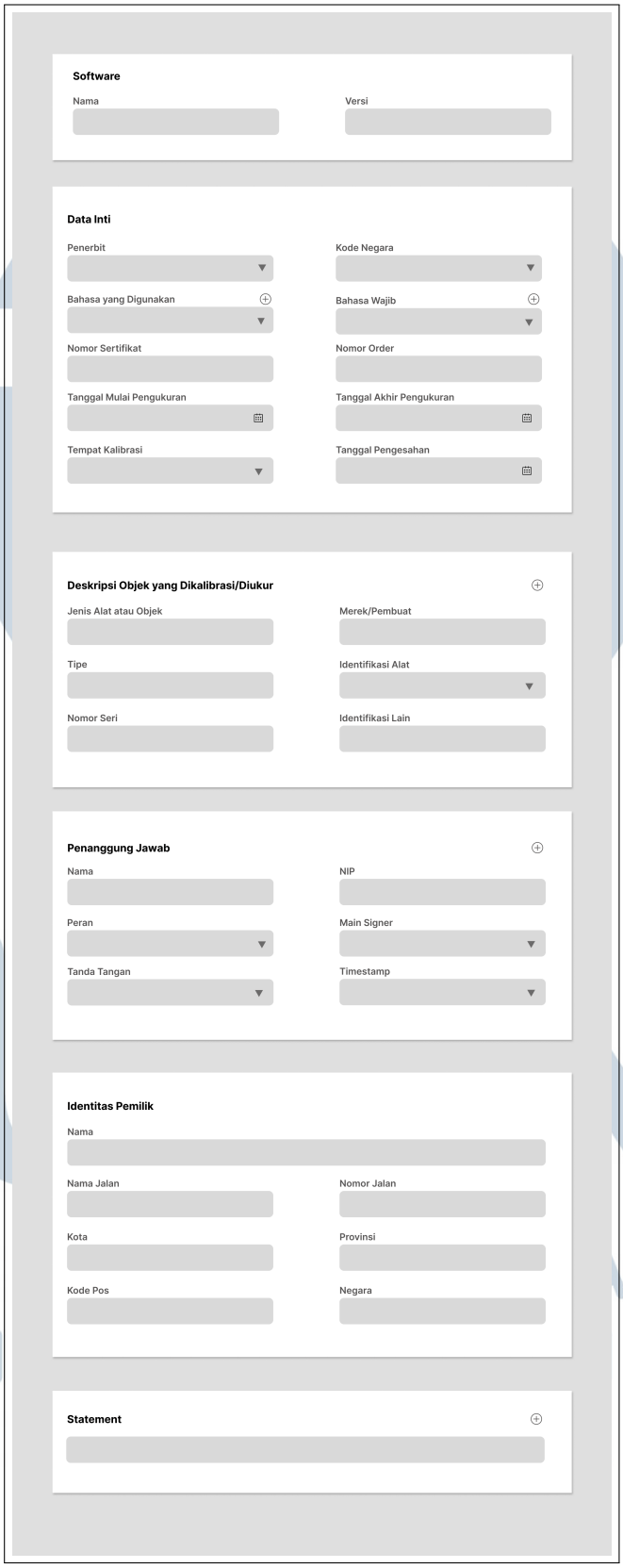


Gambar 3.7. Desain halaman *register*

Desain *navbar* dan halaman menu utama ditampilkan di Gambar 3.8. Pada rancangan awal, tombol opsi *generator* diberi tulisan ”Create new DCC”, sedangkan tombol opsi *importer* diberi tulisan ”Load existing DCC”. Pada *navbar*, tombol ”HOME” mengarahkan ke halaman menu utama, tombol ”About us” mengarahkan ke halaman *about*, dan tombol ”Contact us” mengarahkan ke halaman kontak.

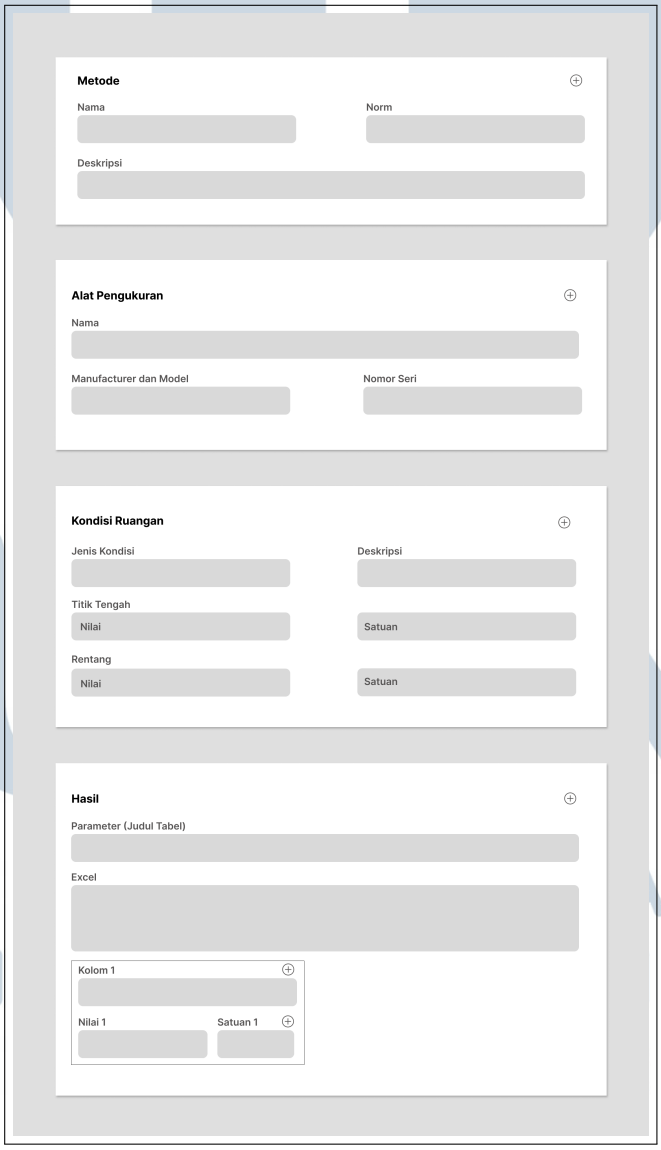


Gambar 3.8. Desain halaman menu utama beserta *navbar*



Gambar 3.9. Desain halaman formulir *generator* tahap administrasi

Cuplikan desain halaman formulir *generator* untuk tahap data administrasi ditampilkan pada Gambar 3.9, sedangkan untuk tahap hasil pengukuran ditunjukkan pada Gambar 3.10. Pada halaman formulir *generator* ini, kolom-kolom input dikategorikan berdasarkan jenis datanya dan disusun ke dalam komponen *card* untuk meningkatkan keterbacaan dan kemudahan pengisian bagi pengguna. Jenis kolom input yang digunakan pun beragam, seperti kolom teks untuk pengetikan manual, *select box* untuk pilihan tertentu, serta pemilih tanggal (*date picker*) berbentuk kalender. Selain itu, tersedia tombol tambah (+) yang memungkinkan pengguna menambahkan kolom atau *card* baru sesuai kebutuhan.



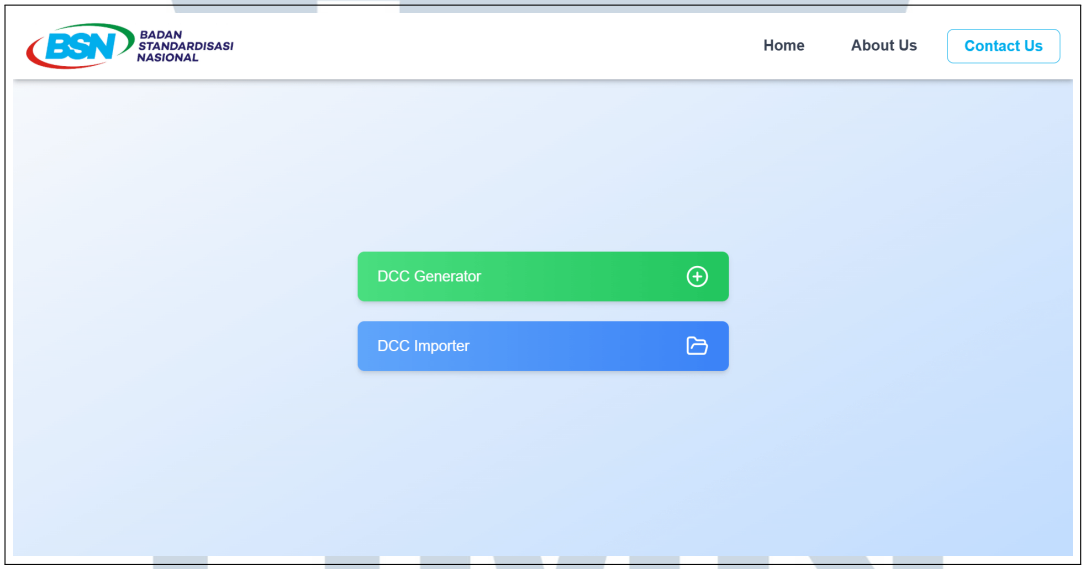
Gambar 3.10. Desain halaman formulir *generator* tahap hasil

3.3.4 Pengembangan Front-end Formulir Generator

Dalam tahap pengembangan, *front-end* dari aplikasi web dijalankan secara lokal di lingkungan pengembangan. Aplikasi ini dapat diakses melalui domain `http://localhost:3000/` setelah dijalankan pada server lokal.

A Halaman Menu Utama

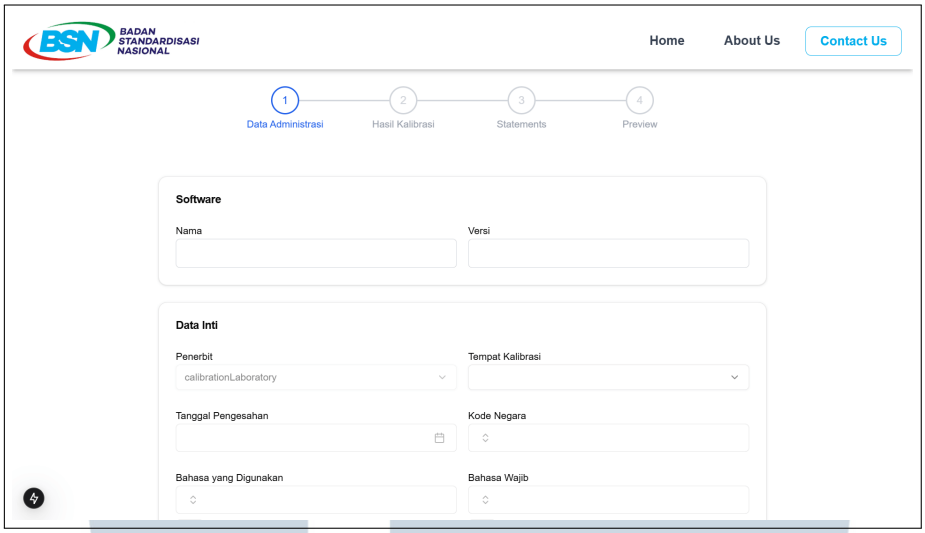
Gambar 3.11 menampilkan tampilan halaman menu utama yang berada pada *path* `/main`. Skema warna yang digunakan terinspirasi dari logo BSN, yaitu kombinasi warna biru dan hijau. Warna-warna ini diterapkan untuk memperkuat identitas visual yang konsisten dengan instansi terkait.



Gambar 3.11. Halaman menu utama

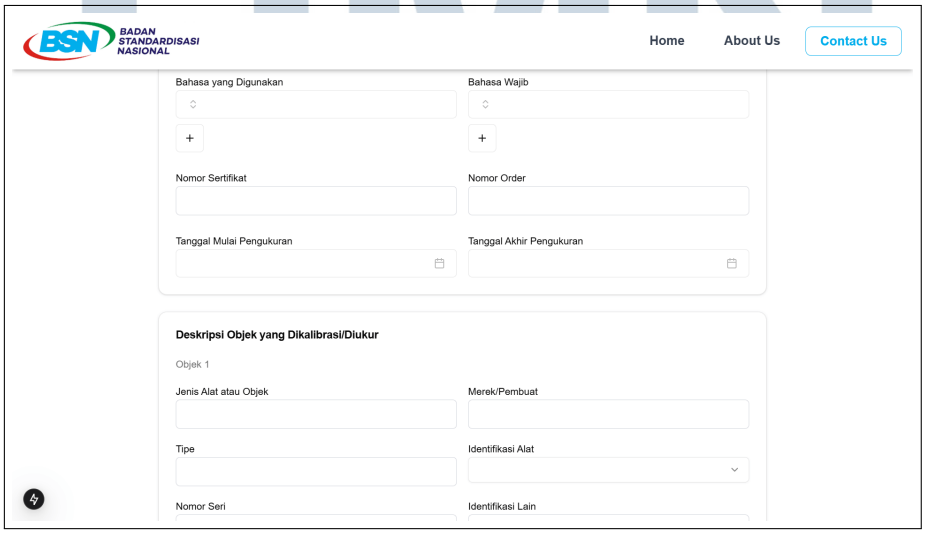
B Halaman Formulir Generator

Pada halaman formulir *generator*, yang berada pada *path* `/create`, terdapat komponen *stepper* yang berfungsi sebagai indikator tahapan pengisian sekaligus sebagai tombol navigasi untuk berpindah antar langkah secara langsung. Rangkaian tampilan pada halaman formulir *generator* tahap administrasi ditunjukkan pada Gambar 3.12 hingga 3.17.

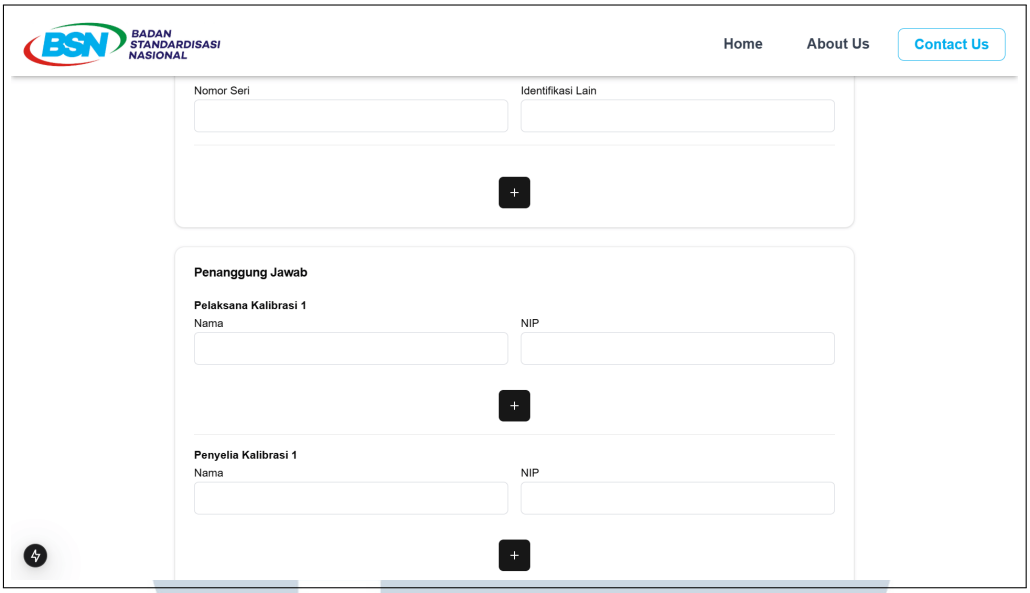


Gambar 3.12. Halaman formulir *generator* tahap administrasi bagian 1

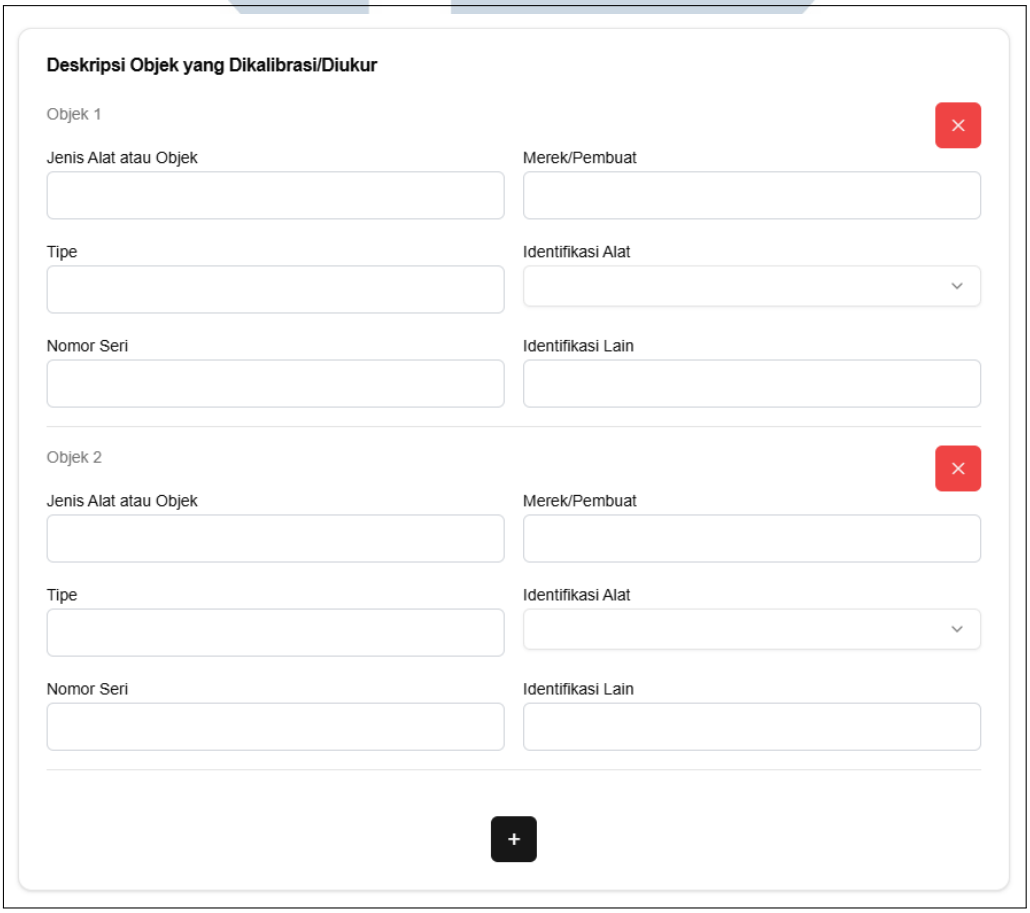
Gambar 3.12 memperlihatkan kolom *select* ”Penerbit” yang di-*disable* (dinonaktifkan) secara khusus untuk opsi ”calibrationLaboratory”, sehingga pengguna tidak dapat memilih opsi lainnya. Selain itu, terdapat kolom *combobox* ”Kode Negara” yang menggunakan *Application Programming Interface* (API) dari REST Countries [7] untuk memuat daftar lengkap negara beserta kodenya berdasarkan standar *International Organization for Standardization* (ISO) 3166-1 (dua huruf). Selain itu, pada kolom *combobox* ”Bahasa yang Digunakan” dan ”Bahasa Wajib”, digunakan API dari OpenDataSoft [8] yang memuat daftar lengkap bahasa beserta kodenya berdasarkan standar ISO 639-1 (dua huruf).



Gambar 3.13. Halaman formulir *generator* tahap administrasi bagian 2

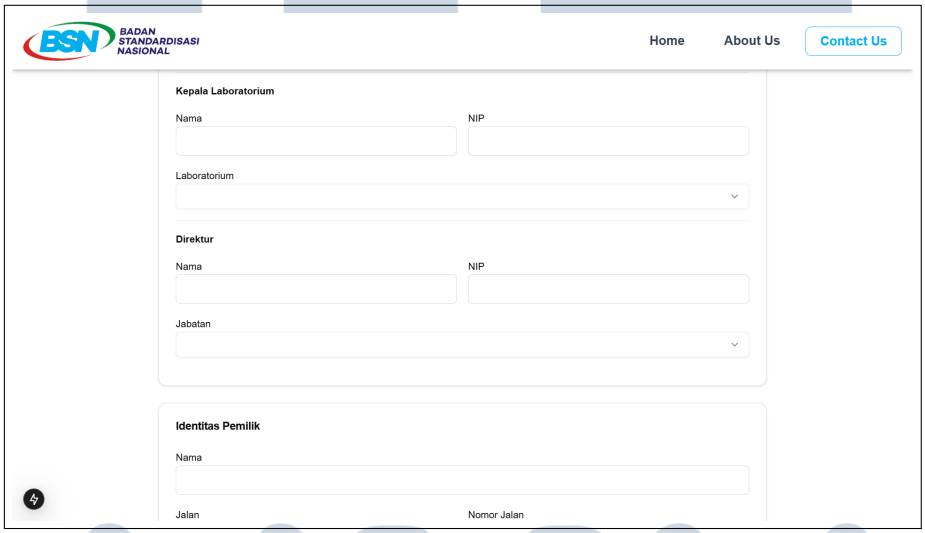


Gambar 3.14. Halaman formulir *generator* tahap administrasi bagian 3

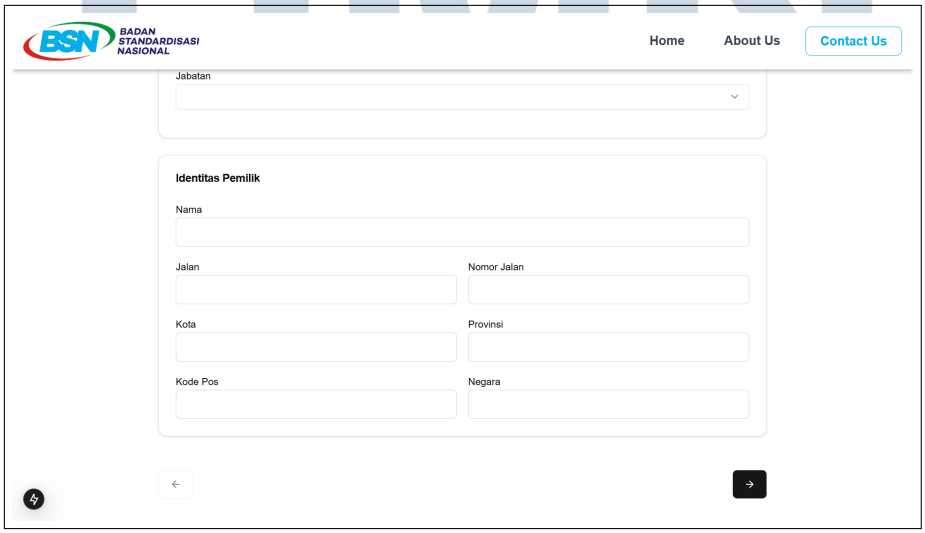


Gambar 3.15. *Card* ”Deskripsi Objek yang Dikalibrasi/Diukur” yang telah ditambah

Pada Gambar 3.13, ditampilkan tulisan ”Objek 1” pada *card* ”Deskripsi Objek yang Dikalibrasi/Diukur”. Hal ini menunjukkan bahwa *card* tersebut bersifat dinamis dan dapat ditambahkan untuk memungkinkan pengguna mengisi lebih dari satu objek. Ketika tombol tambah (+) ditekan, seperti pada Gambar 3.14, maka *card* baru akan ditambahkan dan diberi label ”Objek 2”, dan seterusnya. Jika terdapat lebih dari satu *card*, maka masing-masing *card* akan menampilkan tombol hapus (x) yang memungkinkan pengguna menghapus *card* tersebut, seperti ditunjukkan pada Gambar 3.15. Fitur ini juga berlaku pada *card* dinamis lainnya yang dapat ditambahkan oleh pengguna.

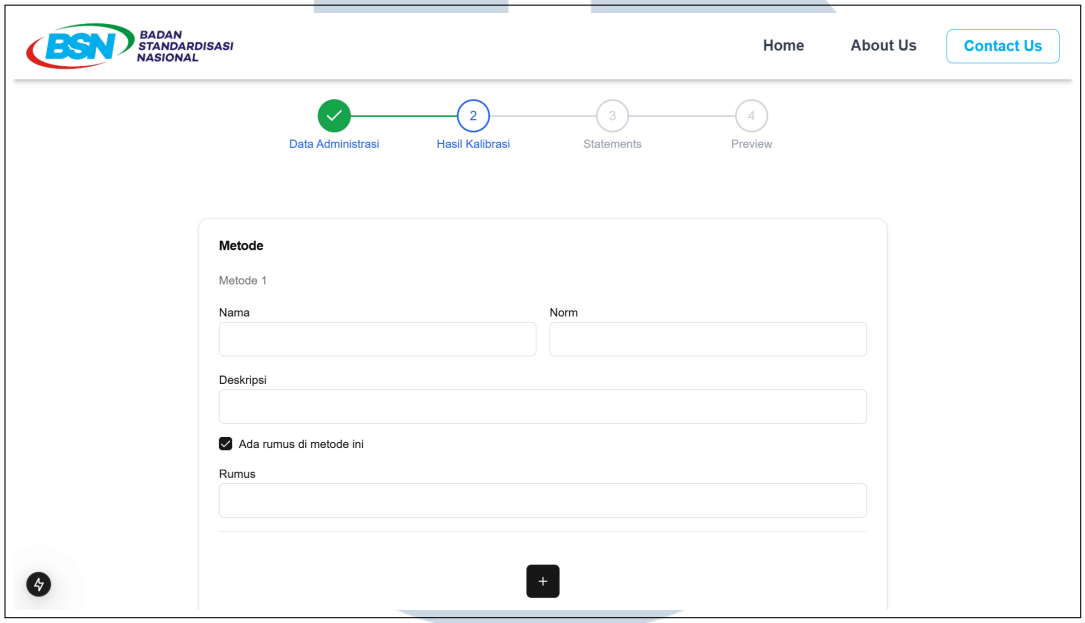


Gambar 3.16. Halaman formulir *generator* tahap administrasi bagian 4

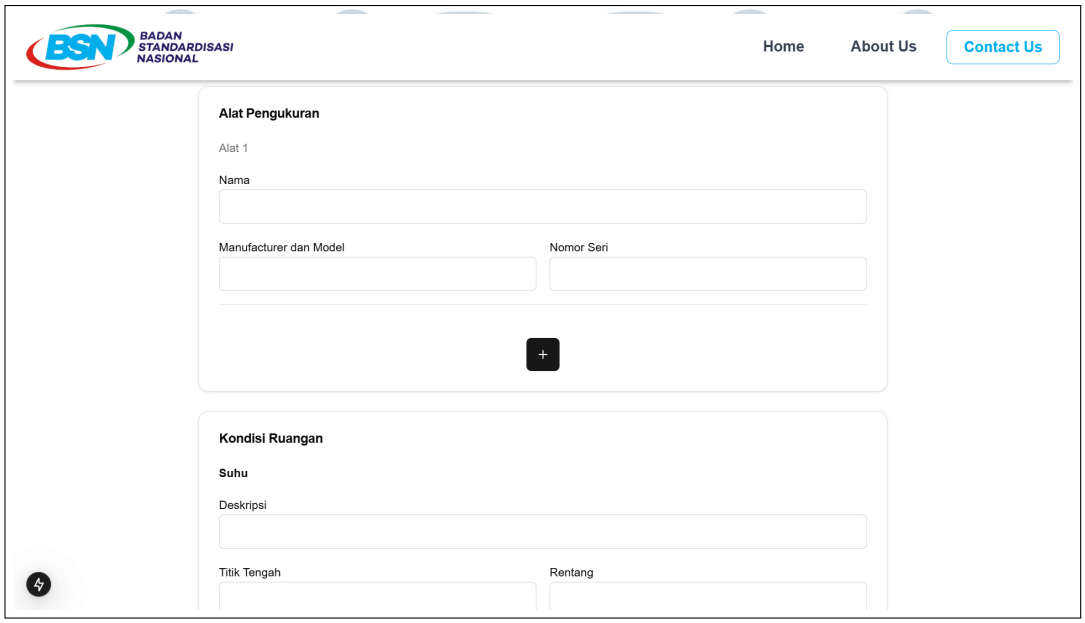


Gambar 3.17. Halaman formulir *generator* tahap administrasi bagian 5

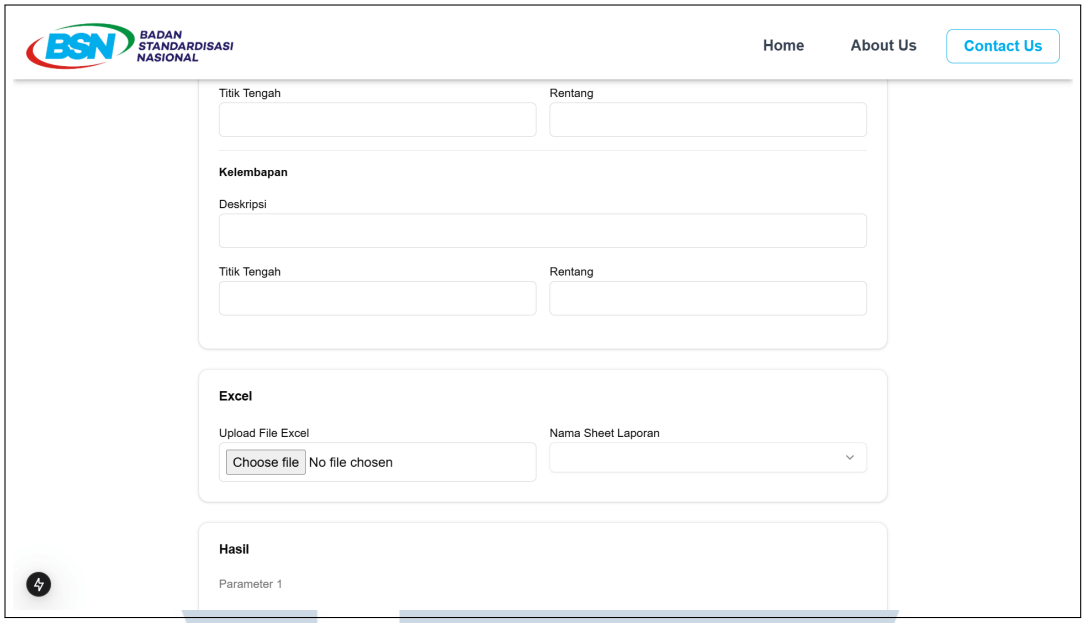
Antarmuka dari tahap hasil dalam formulir *generator* dapat dilihat pada Gambar 3.18 hingga 3.21. Seperti yang ditunjukkan pada Gambar 3.18, terdapat *checkbox* (kotak centang) di *card* ”Metode”. Jika *checkbox* dicentang, maka kolom input ”Rumus” akan muncul.



Gambar 3.18. Halaman formulir *generator* tahap administrasi bagian 1

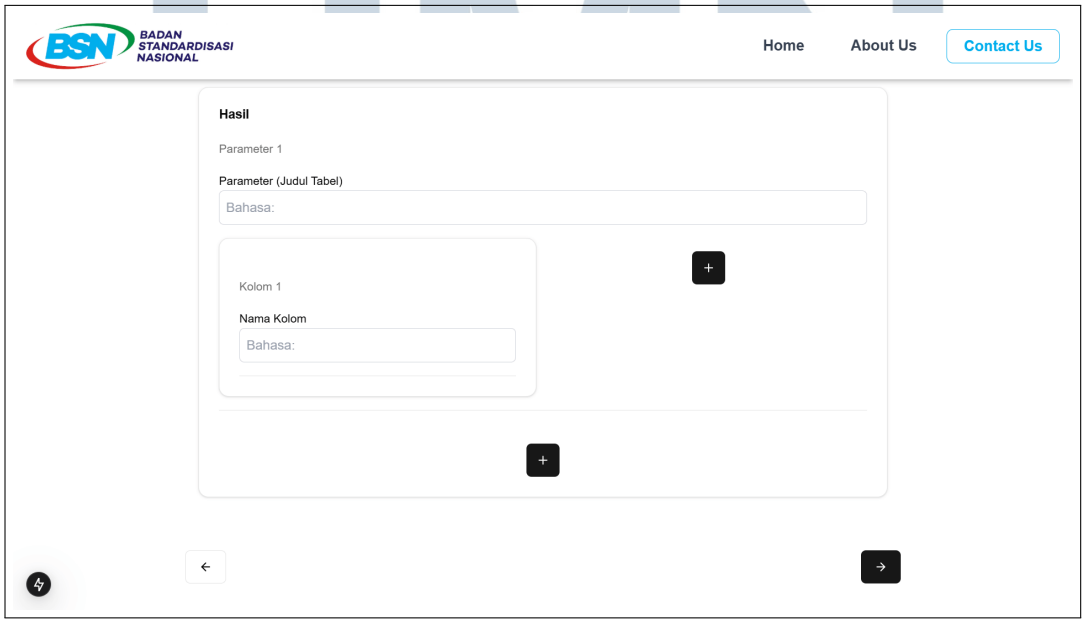


Gambar 3.19. Halaman formulir *generator* tahap administrasi bagian 2



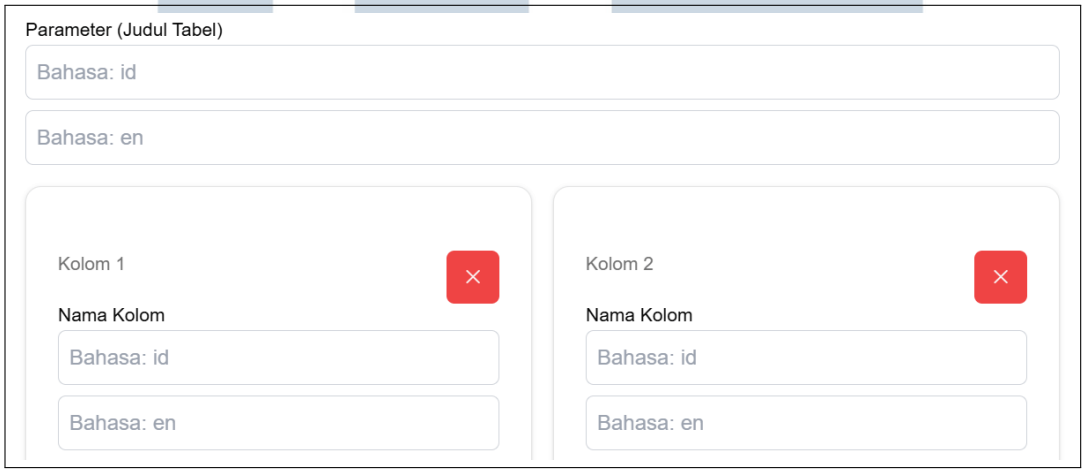
Gambar 3.20. Halaman formulir *generator* tahap administrasi bagian 3

Antarmuka kolom ”Upload File Excel” ditunjukkan pada Gambar 3.20. Kolom ini hanya menerima *file* berformat Excel dan setelah *file* diunggah, *file* tersebut langsung dikirim ke *back-end*. Nama-nama *sheet* (lembar) yang terdapat dalam *file* Excel tersebut diekstraksi dan dikirim ke *front-end* untuk ditampilkan dalam kolom *select* ”Nama Sheet Laporan”. Pengguna lalu memilih *sheet* yang mengandung tabel hasil kalibrasi yang akan diproses.



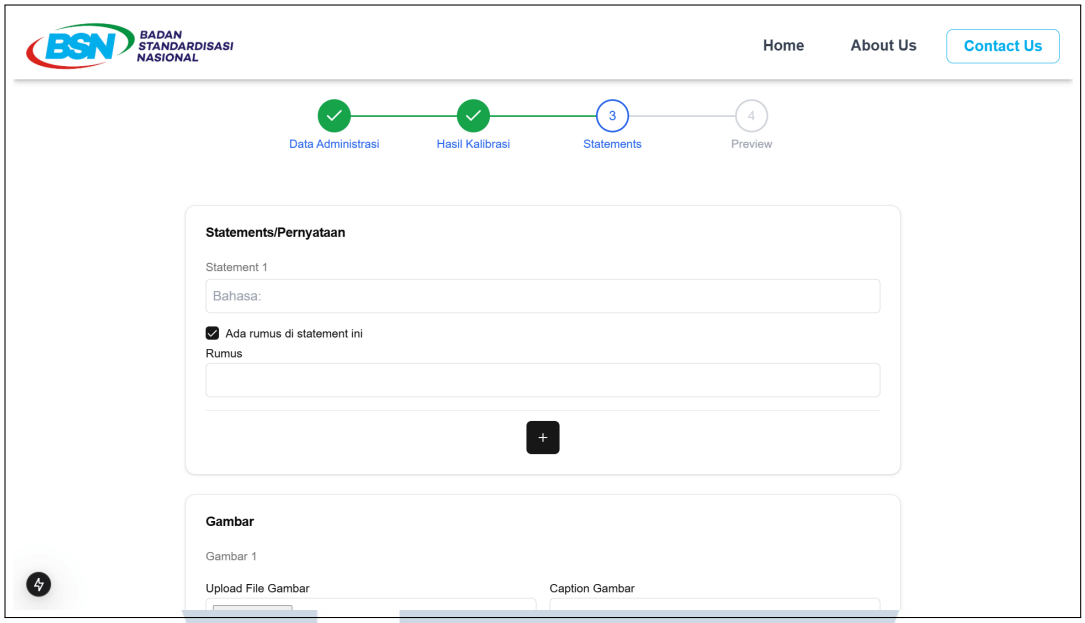
Gambar 3.21. Halaman formulir *generator* tahap administrasi bagian 4

Pada Gambar 3.21, terlihat adanya *placeholder* bertuliskan ”Bahasa: ” pada kolom input ”Parameter (Judul Tabel)” dan ”Nama Kolom”. Hal ini disebabkan oleh fleksibilitas pengisian yang disesuaikan dengan pilihan bahasa pada kolom ”Bahasa yang Digunakan”. Apabila ada bahasa yang dipilih, maka kode bahasa tersebut ditampilkan setelah teks ”Bahasa:”. Sebagai contoh, jika pengguna memilih bahasa Indonesia dan Inggris, maka akan ditampilkan dua kolom input dengan *placeholder* masing-masing bertuliskan ”Bahasa: id” dan ”Bahasa: en”, seperti pada Gambar 3.22. Gambar tersebut juga menunjukkan *card* ”Kolom” yang bersifat dinamis dan dapat ditambahkan sesuai kebutuhan.

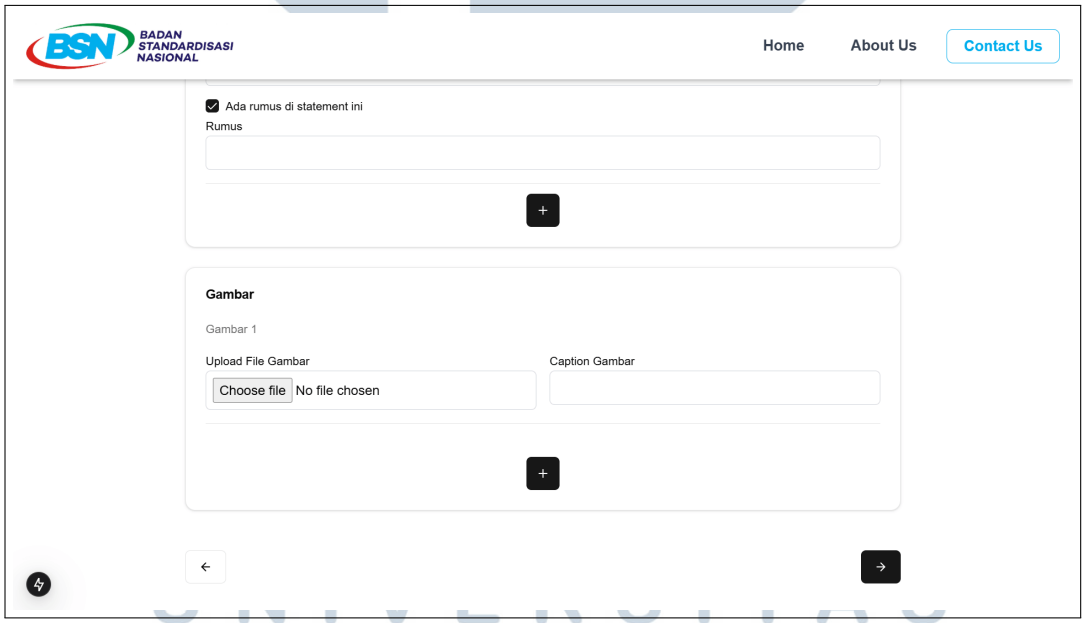


Gambar 3.22. Fitur *multi-language* (multi-bahasa)

Adapun tampilan halaman formulir *generator* pada tahap *statement* (pernyataan) diperlihatkan pada Gambar 3.23 hingga 3.24. Seperti yang ditunjukkan pada Gambar 3.24, terdapat kolom ”Upload File Gambar” yang memungkinkan pengguna untuk mengunggah *file* gambar berformat *Joint Photographic Experts Group* (JPG/JPEG) dan *Portable Network Graphics* (PNG) saja.



Gambar 3.23. Halaman formulir *generator* tahap *statement* bagian 1



Gambar 3.24. Halaman formulir *generator* tahap *statement* bagian 2

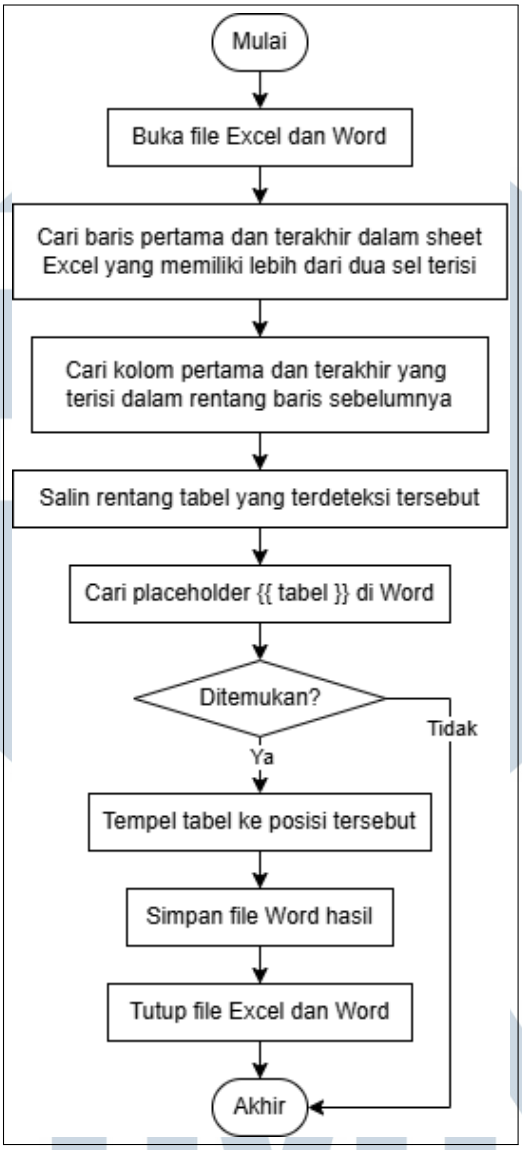
Tahap *preview* berguna untuk menampilkan pratinjau dari DCC yang akan dihasilkan, sebelum formulir dikirim. Namun, untuk tahap pengembangan saat ini, fitur tersebut belum dikembangkan dan masih hanya disediakan tempatnya.

C Pengiriman Data ke Back-end

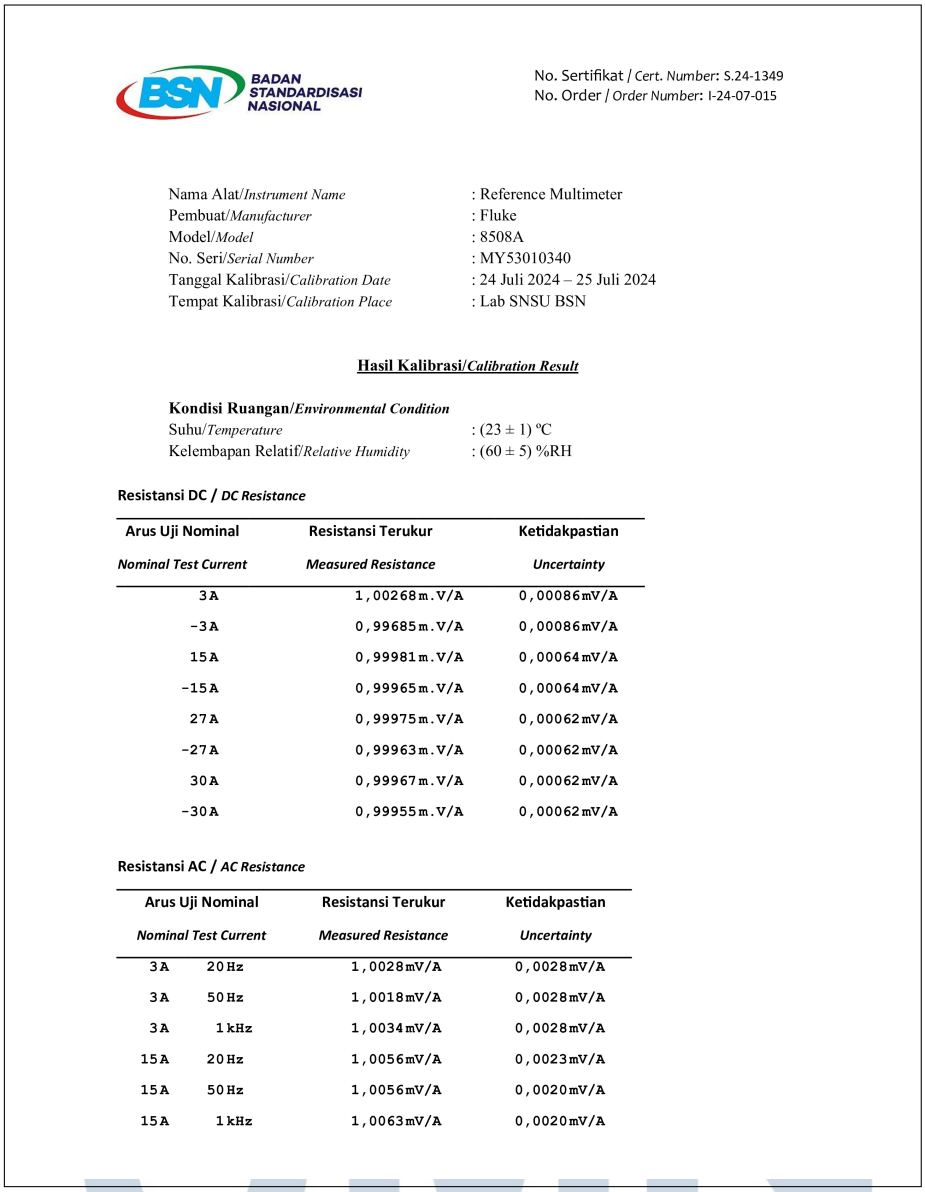
Setelah formulir diisi dan di-*submit*, *front-end* akan mengirim datanya ke *back-end* untuk diproses. Pengiriman ini dilakukan melalui metode POST ke *endpoint* (titik akhir) `http://127.0.0.1:8000/create-dcc/` dalam format *JavaScript Object Notation* (JSON). Setelah data berhasil dikirim, *back-end* akan merespons dengan informasi hasil pemrosesan. Jika respons berisi tautan unduhan, maka tautan tersebut akan ditampilkan kepada pengguna sebagai hasil dari pembuatan DCC.

3.3.5 Penyalinan Tabel di Excel ke Template Word

Alur proses penyalinan tabel di file Excel ke dalam *template* sertifikat di file Word dapat dilihat pada *flowchart* di Gambar 3.25. Proses dimulai dengan membuka aplikasi dan *file* Excel dan Word melalui *library* `win32com.client`. Program kemudian mencari baris pertama dan terakhir di *sheet* Excel yang memiliki lebih dari dua sel terisi, sebagai indikasi awal dan akhir dari tabel yang relevan. Setelah itu, program mencari kolom pertama dan terakhir yang memiliki isi dalam rentang baris yang telah ditentukan, untuk menentukan area tabel secara lengkap. Rentang tabel ini kemudian disalin dari Excel. Selanjutnya, program mencari *placeholder* berupa "{{ tabel }}" di dokumen *template* Word, dan jika ditemukan, tabel yang disalin dari Excel akan ditempel di tempat tersebut. Dokumen Word kemudian disimpan ke *path* tujuan, dan aplikasi Excel serta Word ditutup untuk mengakhiri proses. Hasil tabel di Word ditunjukkan pada Gambar 3.26.



Gambar 3.25. *Flowchart* penyalinan tabel di Excel ke *template* Word



BADAN STANDARDISASI NASIONAL

No. Sertifikat / Cert. Number: S.24-1349
No. Order / Order Number: I-24-07-015

Nama Alat/*Instrument Name* : Reference Multimeter
Pembuat/*Manufacturer* : Fluke
Model/*Model* : 8508A
No. Seri/*Serial Number* : MY53010340
Tanggal Kalibrasi/*Calibration Date* : 24 Juli 2024 – 25 Juli 2024
Tempat Kalibrasi/*Calibration Place* : Lab SNSU BSN

Hasil Kalibrasi/*Calibration Result*

Kondisi Ruangan/*Environmental Condition*
Suhu/*Temperature* : (23 ± 1) ºC
Kelembapan Relatif/*Relative Humidity* : (60 ± 5) %RH

Resistansi DC / *DC Resistance*

Arus Uji Nominal *Nominal Test Current*	Resistansi Terukur *Measured Resistance*	Ketidakpastian *Uncertainty*
3A	1,00268m.V/A	0,00086mV/A
-3A	0,99685m.V/A	0,00086mV/A
15A	0,99981m.V/A	0,00064mV/A
-15A	0,99965m.V/A	0,00064mV/A
27A	0,99975m.V/A	0,00062mV/A
-27A	0,99963m.V/A	0,00062mV/A
30A	0,99967m.V/A	0,00062mV/A
-30A	0,99955m.V/A	0,00062mV/A

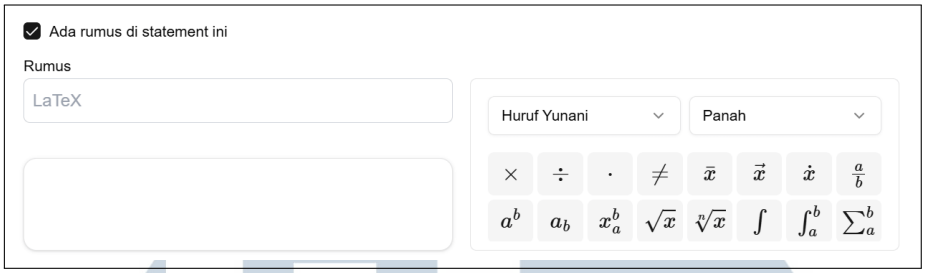
Resistansi AC / *AC Resistance*

Arus Uji Nominal *Nominal Test Current*		Resistansi Terukur *Measured Resistance*	Ketidakpastian *Uncertainty*
3A	20Hz	1,0028mV/A	0,0028mV/A
3A	50Hz	1,0018mV/A	0,0028mV/A
3A	1kHz	1,0034mV/A	0,0028mV/A
15A	20Hz	1,0056mV/A	0,0023mV/A
15A	50Hz	1,0056mV/A	0,0020mV/A
15A	1kHz	1,0063mV/A	0,0020mV/A

Gambar 3.26. Hasil tabel yang disalin ke *file* Word

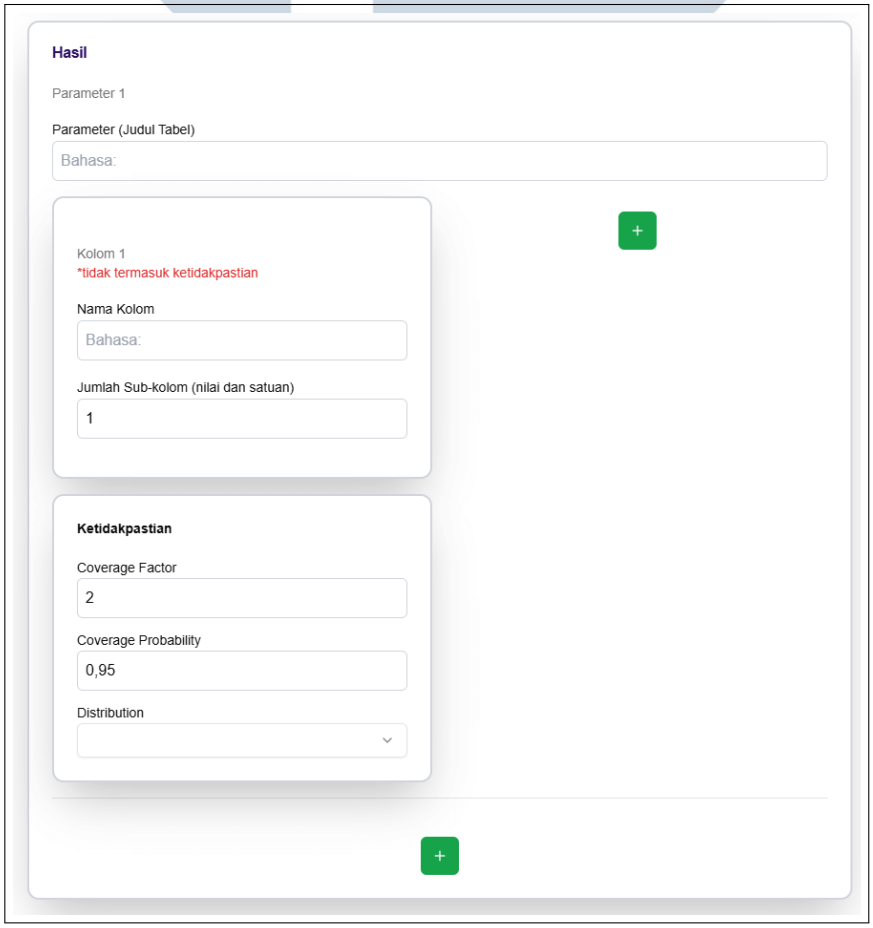
3.3.6 Editor Matematika LaTeX dan Card ”Hasil”

Editor LaTeX dibutuhkan dalam antarmuka pengguna karena sertifikat kalibrasi memuat rumus atau notasi matematika yang disimpan dalam format LaTeX di dalam *file* XML. Editor ini ditambahkan pada komponen *card* ”Metode” dan ”Statement”, sebagaimana ditunjukkan pada Gambar 3.27. Editor LaTeX ini dikembangkan menggunakan MathJax, yang berfungsi untuk me-*render* atau menampilkan notasi LaTeX ke dalam bentuk visual yang mudah dibaca.



Gambar 3.27. Editor matematika LaTeX

Seluruh daftar notasi matematika yang tersedia ditambahkan secara manual ke dalam kode program dan dikelompokkan berdasarkan kategori, seperti huruf Yunani, panah, dan operasi matematika. Pengguna dapat memilih notasi dari bagian kanan antarmuka. Notasi yang dipilih akan langsung muncul dalam tampilan pratinjau di kiri bawah, sementara representasi LaTeX-nya akan muncul di kolom input di kiri atas. Pengguna juga dapat mengedit notasi tersebut langsung dari kolom input, misalnya untuk mengganti variabel atau angka sesuai kebutuhan.



Gambar 3.28. *Card* "Hasil"

Selain penambahan editor LaTeX, *card* ”Hasil” juga mengalami sejumlah perubahan, seperti yang ditunjukkan pada Gambar 3.28. Adapun perubahan-perubahan tersebut dijelaskan sebagai berikut.

1. Jumlah subkolom

   Di dalam tabel hasil kalibrasi, ada kolom-kolom tertentu yang memiliki dua satuan, seperti kolom ”Arus Uji Nominal” pada Gambar 3.29. Dengan demikian, kolom ”Arus Uji Nominal” dapat dikatakan memiliki dua subkolom. Pengguna dapat menginput jumlah subkolom yang dimiliki suatu kolom untuk memudahkan proses pengolahan tabel Excel menjadi format XML. Nilai *default* (bawaan) dari input jumlah subkolom ini diatur menjadi 1 karena sebagian besar kolom dalam tabel hasil kalibrasi hanya memiliki satu subkolom.

2. Ketidakpastian

   Dalam setiap tabel hasil kalibrasi, terdapat kolom yang menyatakan nilai ketidakpastian, seperti yang dapat dilihat pada Gambar 3.29. Di XML, ketidakpastian memiliki beberapa parameter, yaitu *coverage factor* (faktor cakupan), *coverage probability* (tingkat kepercayaan), dan distribusi. *Coverage factor* dan *coverage probability* diatur agar nilai *default*-nya masing-masing adalah 2 dan 0,95.

Resistansi AC / *AC Resistance*

Arus Uji Nominal *Nominal Test Current*		**Resistansi Terukur** *Measured Resistance*		**Ketidakpastian** *Uncertainty*	
3	A	20 Hz	1,0028	mV/A	0,0028 mV/A
3	A	50 Hz	1,0018	mV/A	0,0028 mV/A
3	A	1 kHz	1,0034	mV/A	0,0028 mV/A
15	A	20 Hz	1,0056	mV/A	0,0023 mV/A
15	A	50 Hz	1,0056	mV/A	0,0020 mV/A
15	A	1 kHz	1,0063	mV/A	0,0020 mV/A
20	A	20 Hz	1,0038	mV/A	0,0018 mV/A
20	A	50 Hz	1,0037	mV/A	0,0017 mV/A
20	A	1 kHz	1,0045	mV/A	0,0017 mV/A

Gambar 3.29. Contoh tabel hasil kalibrasi

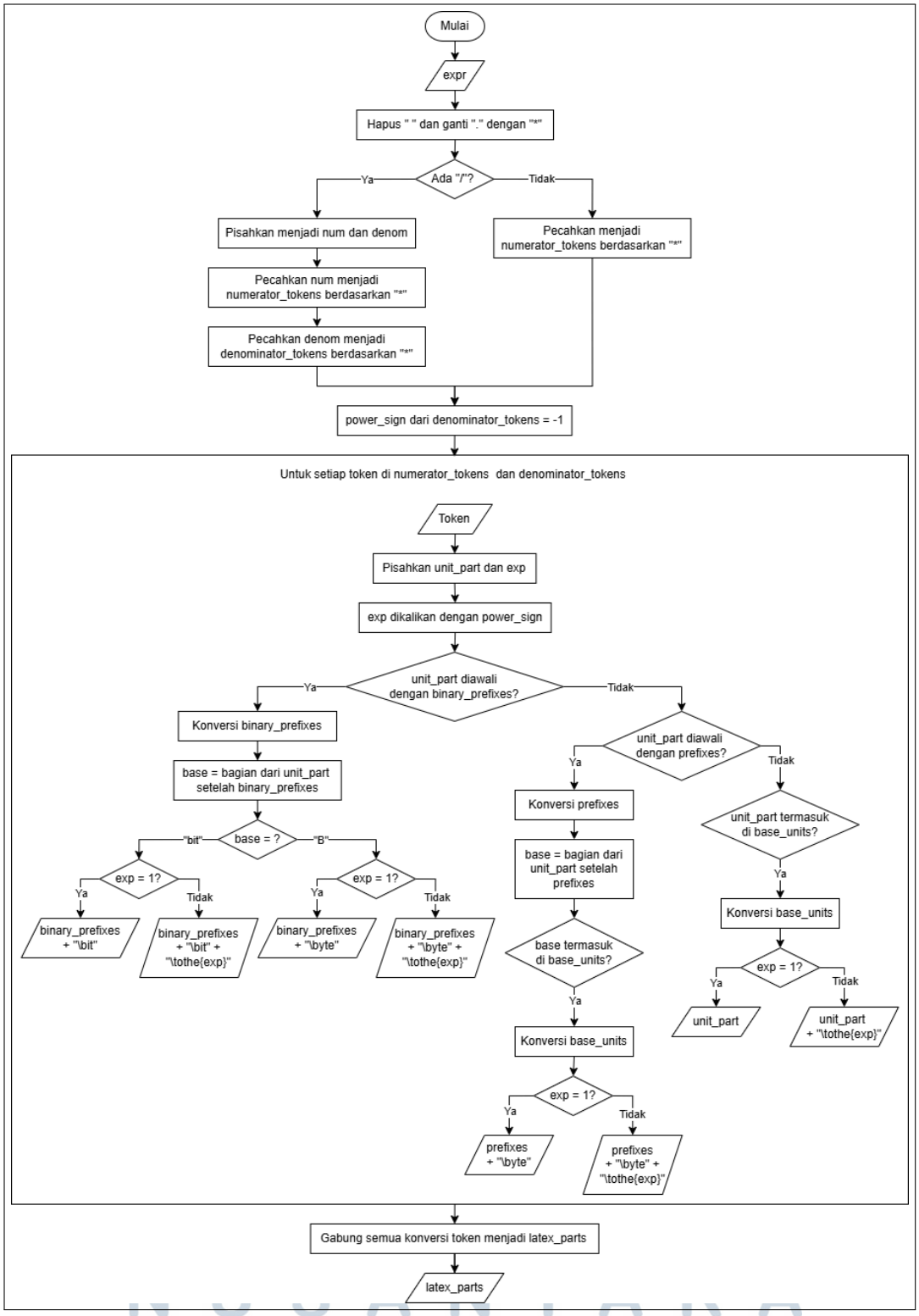
3.3.7 Konversi Tabel Excel menjadi XML

Seperti terlihat pada Gambar 3.29, satuan dalam tabel hasil kalibrasi pada *file* Excel ditampilkan dalam bentuk simbol. Namun, dalam *file* XML, satuan tersebut harus dituliskan dalam format *Digital System of Units* (D-SI). Oleh karena itu, diperlukan proses konversi dari simbol satuan di Excel ke format D-SI. Proses ini diawali dengan penginisialisasi *dictionary* yang memuat daftar pasangan antara simbol satuan dan format D-SI, seperti yang dicontohkan pada Kode 3.7.

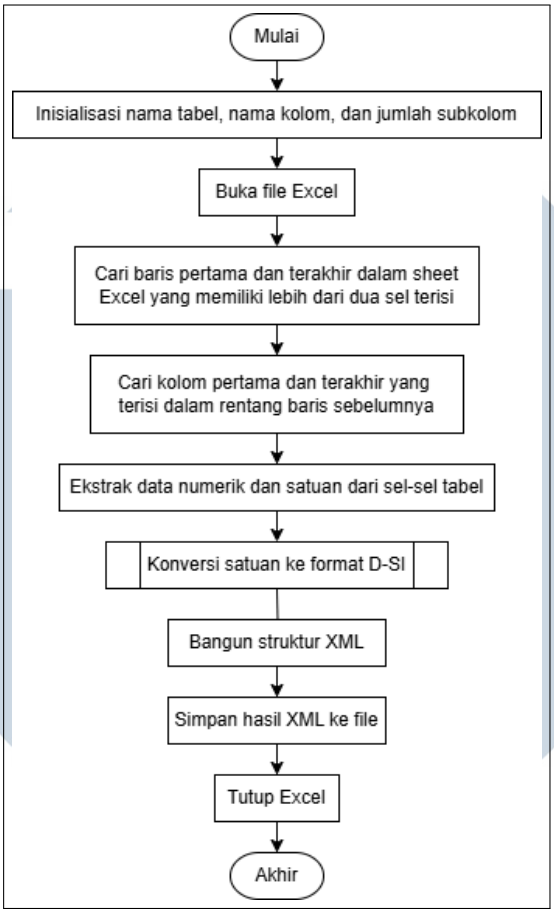
```
prefixes = {
    "k": "\\kilo",
    "c": "\\centi",
    "m": "\\milli"
}

binary_prefixes = {
    "Ki": "\\kibi",
    "Mi": "\\mebi",
    "Gi": "\\gibi"
}

base_units = {
    "m": "\\metre",
    "s": "\\second",
    "V": "\\volt",
}
```

Kode 3.7: Cuplikan kode *dictionary* D-SI

Selanjutnya, dibuat fungsi untuk mengonversi simbol satuan di Excel menjadi format D-SI, yaitu fungsi `d_si` yang ditunjukkan pada Gambar 3.30. Fungsi ini menerima input berupa *string* satuan, seperti `"kV.s / m2"`, lalu menghilangkan spasi dan mengganti titik (.) dengan tanda perkalian (*) untuk memudahkan *parsing*. Jika terdapat tanda pembagian (/), satuan akan dipisah menjadi bagian pembilang dan penyebut, yang kemudian dipecah lagi berdasarkan tanda perkalian (*). Setiap *token* satuan akan dianalisis apakah diawali dengan `binary_prefixes`, `prefixes` (seperti `k` dalam `kV`), atau merupakan `base_units` (seperti `s`, `V`, `m`). *Token* tersebut dikonversi ke dalam format LaTeX D-SI, dan jika berasal dari penyebut, maka pangkatnya dikalikan -1. Sebagai contoh, satuan `"kV.s / m2"` akan dikonversi menjadi `\kilo\volt\second\metre\tothe-2`.



Gambar 3.30. *Flowchart* konversi D-SI



Gambar 3.31. *Flowchart* konversi tabel Excel ke XML

Alur pembacaan data Excel dan konversi ke XML dapat dilihat pada Gambar 3.31. Proses konversi data kalibrasi dari Excel ke format XML dimulai dengan inisialisasi informasi tabel, termasuk nama tabel, nama kolom, dan jumlah subkolom untuk setiap jenis pengukuran. Inisialisasi tersebut dibuat dalam bentuk *nested dictionary* dan contohnya ditunjukkan pada Kode 3.8.

```
input_tables = {
    "Resistansi DC": {
        "Arus Uji Nominal": 1,
        "Resistansi Terukur": 1,
    },
    "Resistansi AC": {
        "Arus Uji Nominal": 2,
        "Resistansi Terukur": 1,
    },
}
```

Kode 3.8: Contoh kode inisialisasi informasi tabel

Selanjutnya, program membuka *file* Excel dan membaca seluruh isi *sheet* yang ditentukan. Untuk mengidentifikasi lokasi data tabel, dilakukan pencarian baris pertama dan terakhir yang memiliki lebih dari dua sel terisi. Setelah itu, dalam rentang baris tersebut, dicari pula kolom pertama dan terakhir yang mengandung data. Setelah lokasi tabel diketahui, program mengekstrak data numerik dan satuan dari setiap sel dalam tabel tersebut. Setiap satuan kemudian dikonversi ke dalam format D-SI menggunakan fungsi `d_si`.

Data hasil ekstraksi ini digunakan untuk membangun struktur XML, yang mengikuti skema DCC, menggunakan *library* `yattag`. Struktur ini menyimpan setiap nilai dalam *tag* yang sesuai, lengkap dengan satuan, dan juga menyediakan bagian khusus untuk ketidakpastian pengukuran. Setelah struktur XML selesai dibentuk, hasilnya disimpan dalam sebuah *file* XML dengan nama yang sesuai dengan nama *file* Excel. Terakhir, *file* Excel ditutup untuk mengakhiri proses. Proses ini memastikan bahwa data kalibrasi dari Excel dapat dikonversi ke format XML yang sesuai dengan standar internasional. Hasil format D-SI di *file* XML ditunjukkan pada Gambar 3.32.

```
▼<si:hybrid>
  ▼<si:realListXMLList>
     <si:valueXMLList>3.0 3.0 3.0 15.0 15.0 15.0 20.0 20.0 20.0</si:valueXMLList>
     <si:unitXMLList>\ampere \ampere \ampere \ampere \ampere \ampere \ampere \ampere \ampere</si:unitXMLList>
   </si:realListXMLList>
  ▼<si:realListXMLList>
     <si:valueXMLList>20.0 50.0 1.0 20.0 50.0 1.0 20.0 50.0 1.0</si:valueXMLList>
     <si:unitXMLList>\hertz \hertz \kilo\hertz \hertz \hertz \kilo\hertz \hertz \hertz \kilo\hertz</si:unitXMLList>
   </si:realListXMLList>
 </si:hybrid>
```

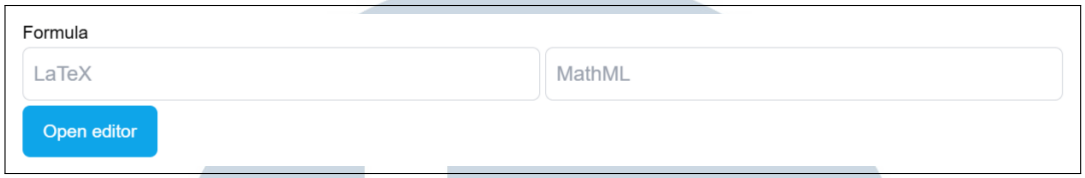
Gambar 3.32. Hasil D-SI di XML

3.3.8 Integrasi iMathEQ

Pada tahap awal pengembangan, sistem dirancang untuk mendukung input notasi matematika dalam format LaTeX saja, sesuai dengan kebutuhan awal pengguna. Namun, seiring berjalannya waktu dan setelah dilakukan diskusi lebih lanjut dengan pengguna, ditemukan bahwa sistem juga perlu mendukung format MathML agar sesuai dengan skema yang digunakan oleh PTB. Oleh karena itu, dilakukan penyesuaian pada fitur input ekspresi matematika agar mampu menangani dan mengonversi kedua format, yakni LaTeX dan MathML.

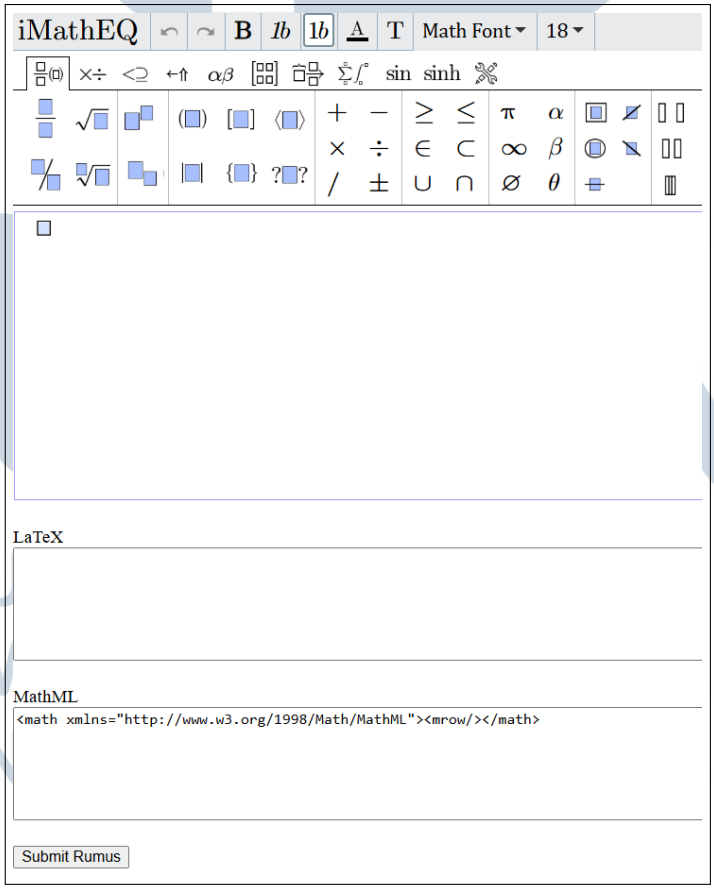
iMathEQ merupakan editor rumus matematika daring yang dapat diintegrasikan ke dalam *website* maupun aplikasi [9]. Layanan ini bersifat berbayar, namun dalam proses pengembangan ini digunakan lisensi *free trial* yang disediakan

secara terbatas oleh pihak pengembang. Editor ini dipilih karena mendukung kedua format, yaitu LaTeX dan MathML.



Gambar 3.33. UI ”Formula”

iMathEQ diintegrasikan pada bagian ”Formula” (rumus) di *card* ”Metode” dan ”Statement”. Antarmuka bagian ”Formula” ditunjukkan pada Gambar 3.33. Ketika tombol ”Open editor” ditekan, *popup window* iMathEQ di *browser* akan muncul seperti pada Gambar 3.34. Setelah pengguna selesai mengedit notasi matematika, pengguna dapat menekan tombol ”Submit Rumus” lalu masing-masing format LaTeX dan MathML akan otomatis ditampilkan pada kolom ”LaTeX” dan ”MathML”.

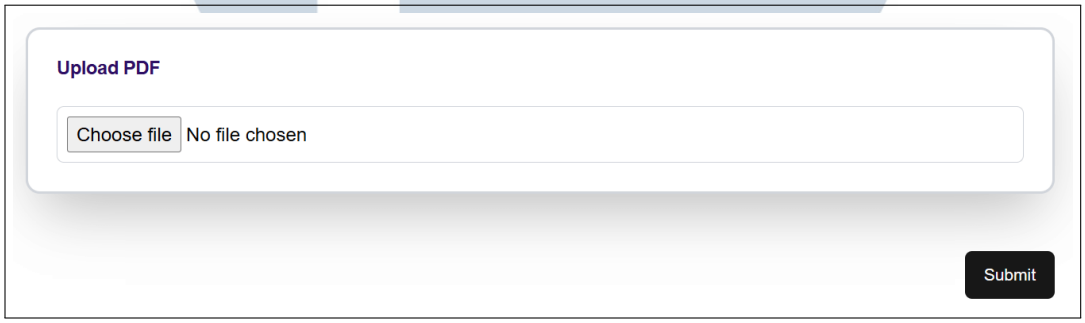


Gambar 3.34. iMathEQ

Antarmuka dari iMathEQ menyerupai editor rumus pada Microsoft Word sehingga memudahkan pengguna. Seperti ditunjukkan pada Gambar 3.34, notasi matematika dapat ditambahkan melalui tombol-tombol notasi yang tersedia di atas, lalu dimodifikasi dengan karakter seperti angka atau huruf di *text area*. Ketika pengguna melakukan pengeditan pada *text area*, format LaTeX dan MathML akan otomatis disinkronkan.

3.3.9 Front-end dari Importer

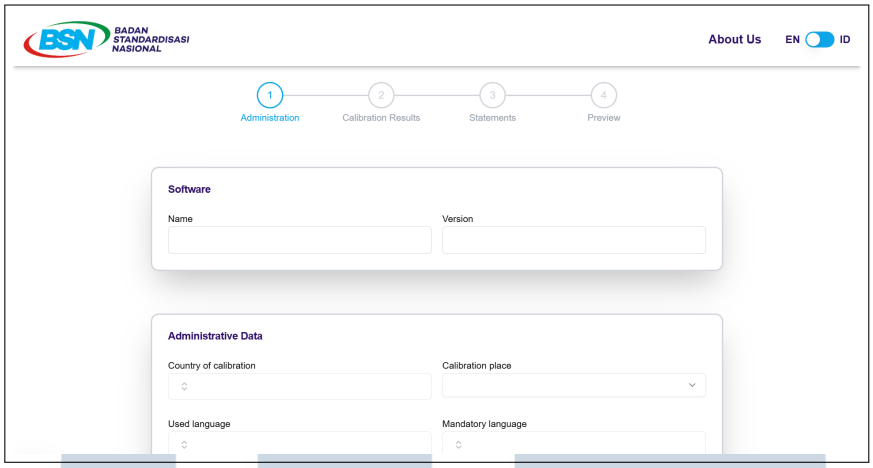
Di halaman *importer*, yang dapat diakses melalui *path* `/load-dcc`, terdapat kolom unggah untuk *file* PDF yang mengandung *file* XML. Setelah pengguna menekan tombol ”Submit”, sistem akan mengonversi *file* PDF tersebut menjadi *file* Excel. Tampilan antarmuka dari *importer* ditunjukkan pada Gambar 3.35.



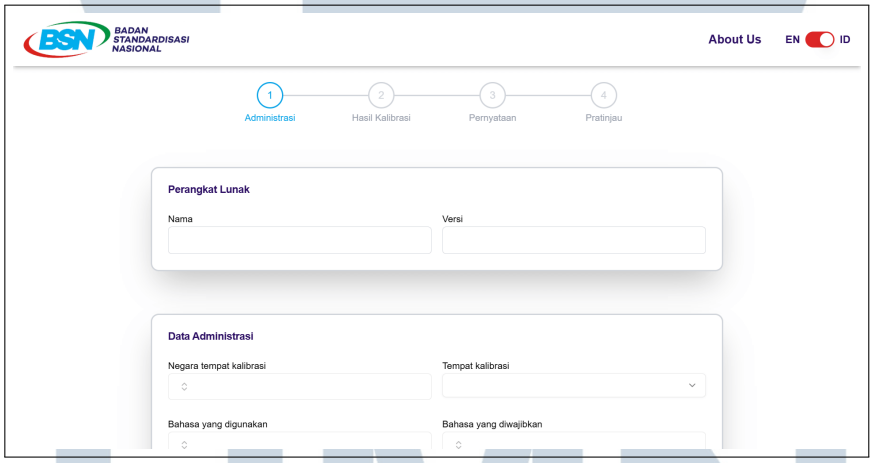
Gambar 3.35. Halaman *importer*

3.3.10 Language Switcher

Dalam upaya meningkatkan aksesibilitas bagi pengguna dari Indonesia maupun mancanegara, dikembangkan fitur *language switcher* yang memungkinkan pengguna mengganti bahasa tampilan situs antara bahasa Indonesia dan bahasa Inggris. Fitur ini ditempatkan di bagian paling kanan pada *navbar*. Bahasa *default* yang ditampilkan saat situs pertama kali dimuat adalah bahasa Inggris. Contoh tampilan antarmuka dalam bahasa Inggris ditunjukkan pada Gambar 3.36, sementara contoh tampilan antarmuka dalam bahasa Indonesia ditunjukkan pada Gambar 3.37.



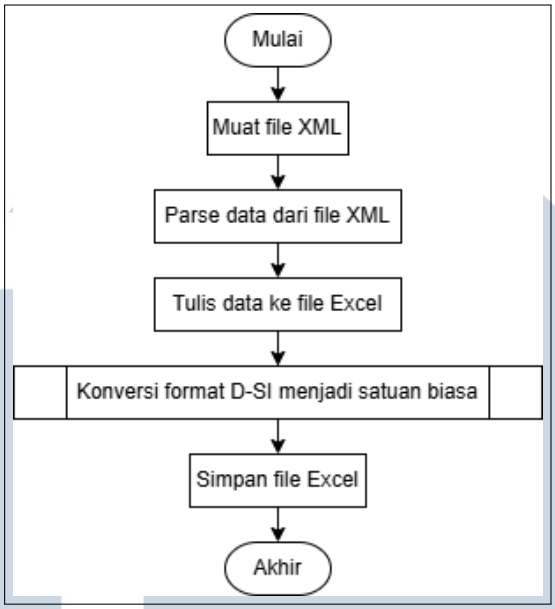
Gambar 3.36. Tampilan antarmuka dalam bahasa Inggris



Gambar 3.37. Tampilan antarmuka dalam bahasa Indonesia

3.3.11 Konversi XML menjadi *file* Excel untuk *Importer*

Alur proses simulasi *importer* ditunjukkan pada Gambar 3.38. Proses dimulai dengan memuat *file* XML menggunakan *library* `dcc` dan `ElementTree`. *File* XML yang digunakan dalam pengembangan ini merupakan contoh dari PTB. *File* tersebut kemudian di-*parse* untuk memperoleh data yang diperlukan. Data yang telah diambil dituliskan ke dalam *file* Excel menggunakan *library* `openpyxl`, dengan format dan *style* tertentu agar mudah dibaca oleh pengguna. Data berupa satuan D-SI dikonversi menjadi satuan biasa menggunakan fungsi `d_si`. Setelah seluruh data selesai dituliskan, *file* Excel disimpan. Cuplikan contoh *file* Excel yang dihasilkan ditunjukkan pada Gambar 3.39.



Gambar 3.38. *Flowchart* proses *importer*

Kondisi Lingkungan

Parameter 1

Parameter lingkungan	:	Immersion depth in water bath	
Deskripsi	:	-	
Immersion depth	:	0.1	m

Parameter 2

Parameter lingkungan	:	Ambient condition temperature	
Deskripsi	:	These values were not measured, but were given based on typical weather conditions at a time of year.	
temperature min	:	293	K
temperature max	:	299	K

Parameter 3

Parameter lingkungan	:	Ambient condition relative humidity	
Deskripsi	:	These values were not measured, but were given based on typical weather conditions at a time of year.	
humidity min	:	20	%
humidity max	:	70	%

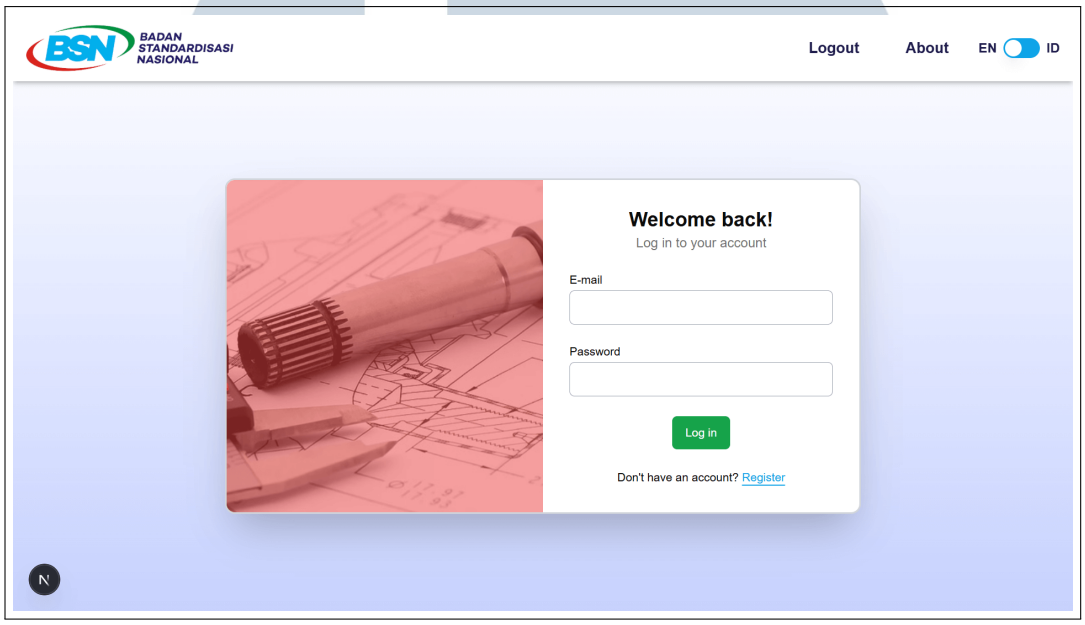
Hasil Kalibrasi

Measuring results									
Reference value				Indicated measured value probe				Measurement error	
306,248	K	33,098	°C	306,32	K	33,17	°C	0,072	K
373,121	K	99,971	°C	373,21	K	100,06	°C	0,089	K
448,253	K	175,103	°C	448,36	K	175,21	°C	0,107	K
523,319	K	250,169	°C	523,31	K	250,16	°C	-0,009	K
593,154	K	320,004	°C	593,07	K	319,92	°C	-0,084	K

Gambar 3.39. Cuplikan *file* Excel dari *importer*

3.3.12 Halaman Login

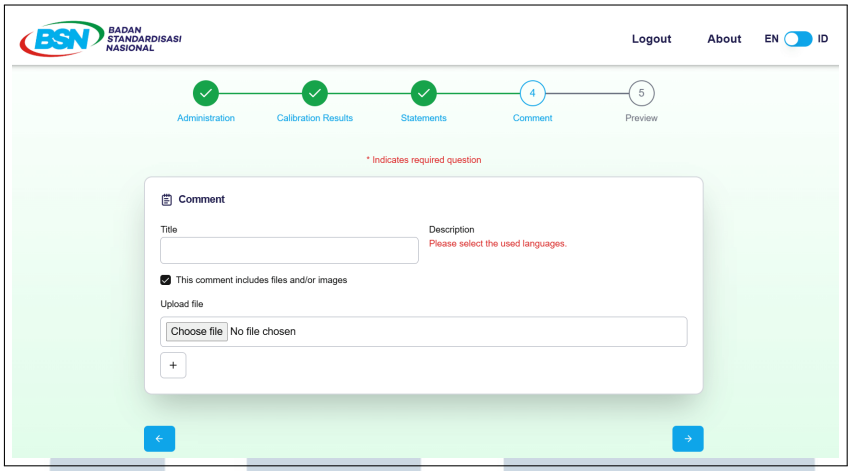
Tampilan antarmuka halaman *login* ditunjukkan pada Gambar 3.40. Halaman ini ditempatkan pada *root path* (/) sehingga menjadi halaman pertama yang dimuat saat *website* diakses. Jika pengguna belum memiliki akun, tersedia *link* yang mengarahkan ke halaman *register*.



Gambar 3.40. Tampilan halaman *login*

3.3.13 Tahap Comment di Formulir Generator

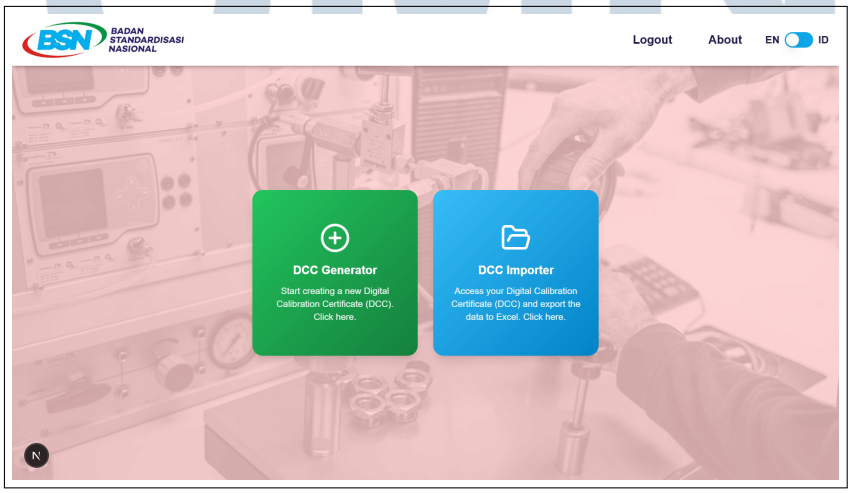
Karena pada XML dibutuhkan *comment* (komentar), maka ditambahkan tahap *comment*, yang ditunjukkan pada Gambar 3.41. Di dalam *card comment*, terdapat *checkbox* untuk menampilkan kolom untuk mengunggah *file* yang mendukung *comment*.



Gambar 3.41. Halaman formulir *generator* tahap *comment*

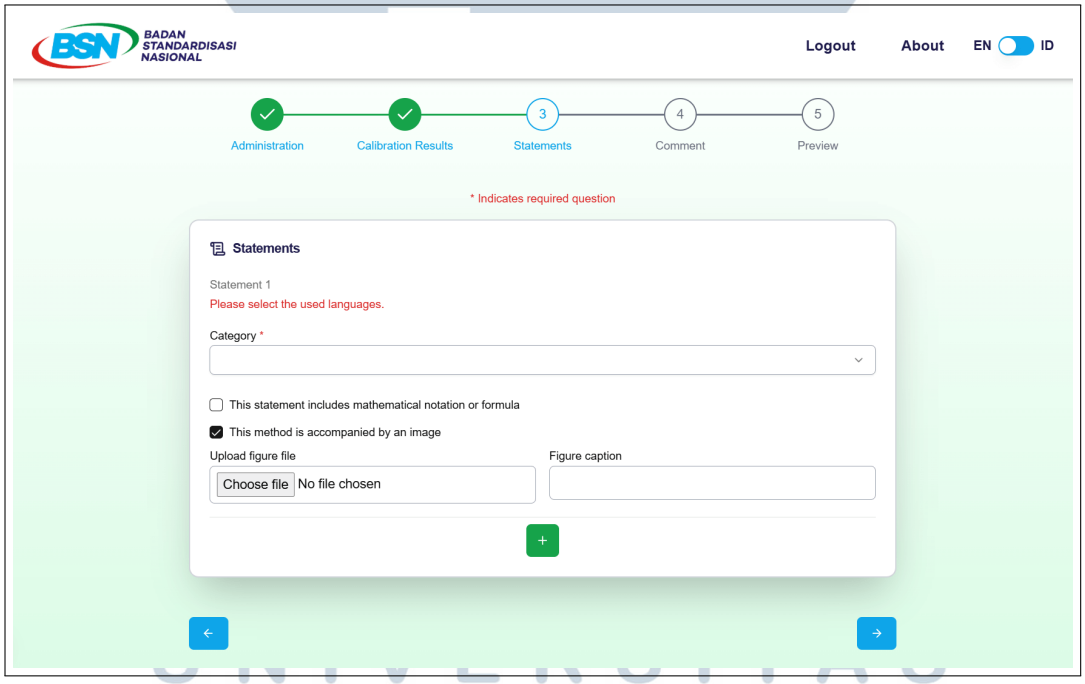
3.3.14 Revisi pada Front-end

Pada tahap ini, dilakukan beberapa perubahan pada tampilan antarmuka dan fitur penginputan data. *Navbar* diperbarui agar memiliki tombol ”Logout” untuk keluar dari akun, tombol ”About” untuk pindah ke halaman *about*, dan *language switcher*. Tampilan menu utama diperbarui dengan memperbesar tombol *generator* dan *importer* agar lebih mudah terlihat dan diakses oleh pengguna. Masing-masing tombol juga dilengkapi dengan tulisan petunjuk singkat untuk memudahkan pengguna baru memahami fungsinya. Selain itu, ditambahkan gambar *background* (latar belakang) bertema kalibrasi agar halaman tidak terlihat polos dan lebih mencerminkan konteks aplikasi. Tampilan halaman menu utama ditunjukkan pada Gambar 3.42.



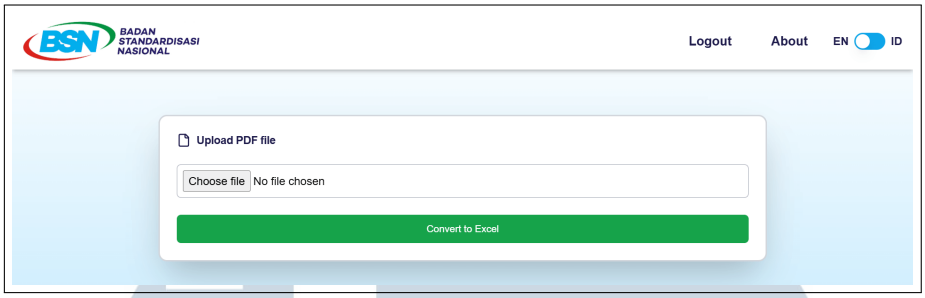
Gambar 3.42. Tampilan halaman menu utama

Pada halaman *generator* dan *importer*, setiap nama *card* dilengkapi dengan ikon simbolis di sebelah kiri judul, yang memberikan konteks visual terhadap isi formulir tersebut. Pada halaman formulir generator, seperti yang ditunjukkan pada Gambar 3.43, ditambahkan latar belakang (background) berupa gradasi warna putih dan hijau, yang diambil berdasarkan warna dari tombol *generator* di halaman menu utama. Kolom-kolom yang wajib diisi ditandai dengan simbol bintang merah (*) pada labelnya. Untuk memperjelas makna simbol tersebut, ditambahkan pula petunjuk teks berwarna merah di bagian atas formulir. Selain itu, dilakukan penyesuaian pada tata letak kolom untuk meningkatkan keterbacaan dan kemudahan dalam proses pengisian. Warna tombol juga mengalami perubahan. Tombol untuk menambahkan *card* dan tombol *submit* menjadi warna hijau, sedangkan tombol untuk berpindah tahap menjadi warna biru. Pemilihan warna ini disesuaikan dengan warna pada logo BSN untuk menjaga konsistensi identitas visual.



Gambar 3.43. Halaman formulir *generator* tahap *statement*

Seperti yang ditunjukkan pada Gambar 3.43, pada kolom-kolom input yang dapat diisi dengan beberapa bahasa, sekarang diwajibkan untuk memilih bahasa yang digunakan dulu agar kolom inputnya terlihat. Selain itu, di *card* metode dan pernyataan, terdapat kolom unggah gambar yang hanya muncul saat *checkbox* dicentang.

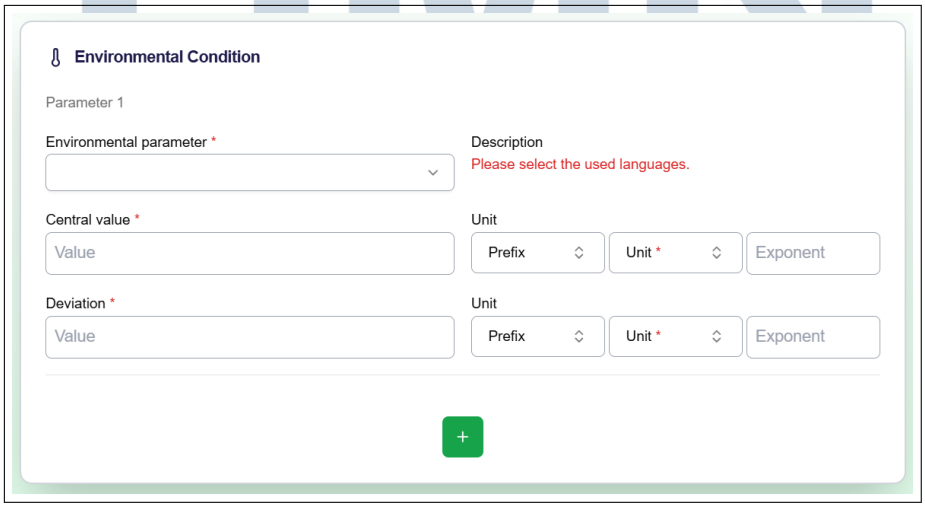


Gambar 3.44. Halaman *importer*

Selain halaman *generator*, *background* dari halaman *importer* juga diperbarui untuk menyesuaikan warna tombol *importer* di halaman menu utama. Halaman ini ditunjukkan pada Gambar 3.44. Tombol *submit* dipindahkan menjadi tombol ”Convert to Excel”.

3.3.15 Perubahan Card Kondisi Lingkungan

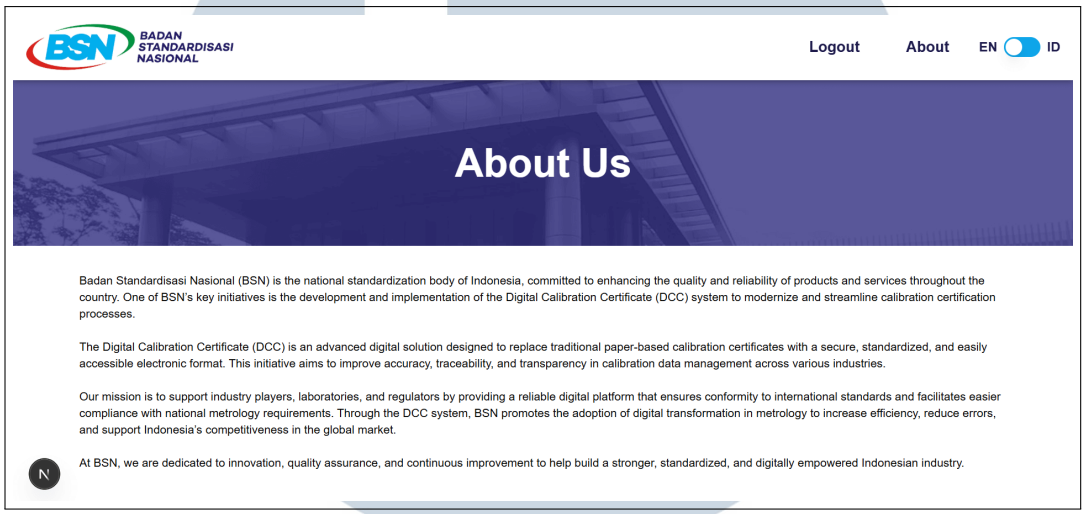
Perubahan pada *card* kondisi lingkungan, yang ditunjukkan pada Gambar 3.45, dilakukan untuk mengakomodasi parameter selain suhu dan kelembapan. Di kolom ”Environmental parameter”, pengguna dapat membuka menu *select* dan memilih parameter suhu, kelembapan, atau opsi ”other” untuk mengetik parameter lainnya secara manual di kolom input. Selain itu, kolom satuan disusun menjadi tiga elemen, yaitu *prefix*, *unit* (satuan dasar), dan *exponent* (pangkat). Kolom *prefix* dan *unit* menggunakan *combobox* agar pengguna dapat mencari dan memilih nilai yang sesuai.



Gambar 3.45. *Card* kondisi lingkungan

3.3.16 Halaman About

Halaman ini ditunjukkan pada Gambar 3.46, yang berada pada *path* `/about`. Halaman *about* memuat profil dari *website* ini serta informasi mengenai *website* ini. Saat ini, konten pada halaman tersebut masih berupa contoh dan belum menggunakan konten resmi dari perusahaan.



Gambar 3.46. Halaman *about*

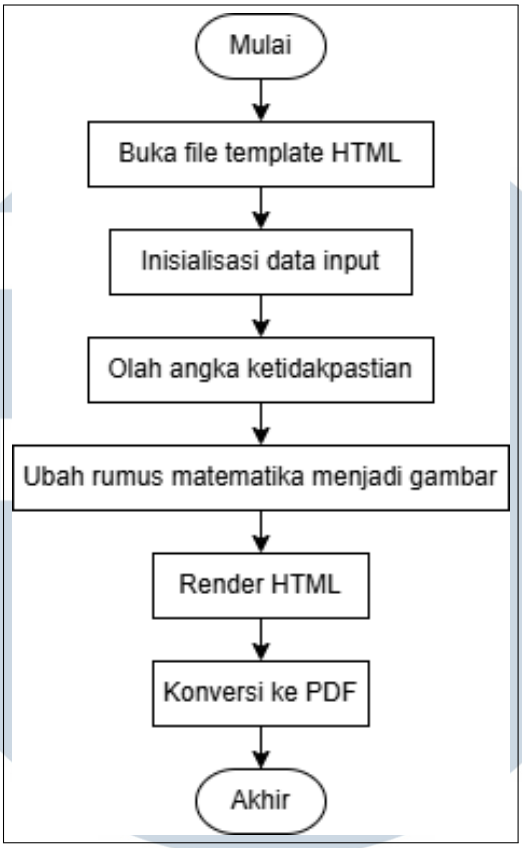
3.3.17 Konversi Input menjadi PDF dengan HTML

Proses konversi pada *generator* yang semula dilakukan melalui pembuatan *file* Word, kini diubah menjadi konversi ke format HTML. Perubahan ini meningkatkan efisiensi karena memungkinkan pembuatan *file* PDF yang bersifat dinamis sehingga dapat menyesuaikan jumlah label dengan banyaknya data yang ditampilkan. Proses ini menggunakan Jinja2 dan WeasyPrint. Jinja2 berfungsi untuk memasukkan data ke dalam *template* HTML, sedangkan WeasyPrint digunakan untuk mengubah *file* HTML tersebut menjadi *file* PDF. Cuplikan *template* HTML yang digunakan ditunjukkan pada Kode 3.9. Kode tersebut menampilkan beberapa objek secara dinamis sesuai dengan jumlah data yang diberikan. Karena pengguna dapat menginput data dalam berbagai bahasa, maka bahasa yang ditampilkan pada PDF diprioritaskan secara berurutan, dengan bahasa Indonesia sebagai pilihan utama. Jika bahasa Indonesia tidak tersedia, maka digunakan bahasa Inggris. Apabila kedua bahasa tersebut tidak tersedia, maka digunakan bahasa yang berada di urutan pertama dalam data.

```
{% for obj in objects %}
    {% if objects|length > 1 %}
        <h6 style="font-size: 13px; margin: 0; margin-bottom: 3px;
    ">Objek {{ loop.index }}</h6>
    {% endif %}

    <div style="margin-bottom: 5px; page-break-inside: avoid;">
        <div id="Jenis alat atau objek" class="row">
            <div class="left">
                <div class="label">Jenis alat atau objek</div>
                <div class="sub-label">Type of instrument or
    object</div>
            </div>
            <div class="colon">:</div>
            <div class="value">
                {% if obj.jenis.id is defined %}
                    {{ obj.jenis.id }}
                {% elif obj.jenis.en is defined %}
                    {{ obj.jenis.en }}
                {% else %}
                    {{ obj.jenis | dictsort | rejectattr('0', '
    equalto', 'id') | map(attribute='1') | list | first }}
                {% endif %}
            </div>
        </div>
```

Kode 3.9: Cuplikan *template* HTML

Alur simulasi proses konversi ditunjukkan pada Gambar 3.47. Proses dimulai dengan membuka *file template* HTML. Kemudian, data input simulasi diinisialisasi dalam format *dictionary*. Contoh data input ditunjukkan pada Kode 3.10.

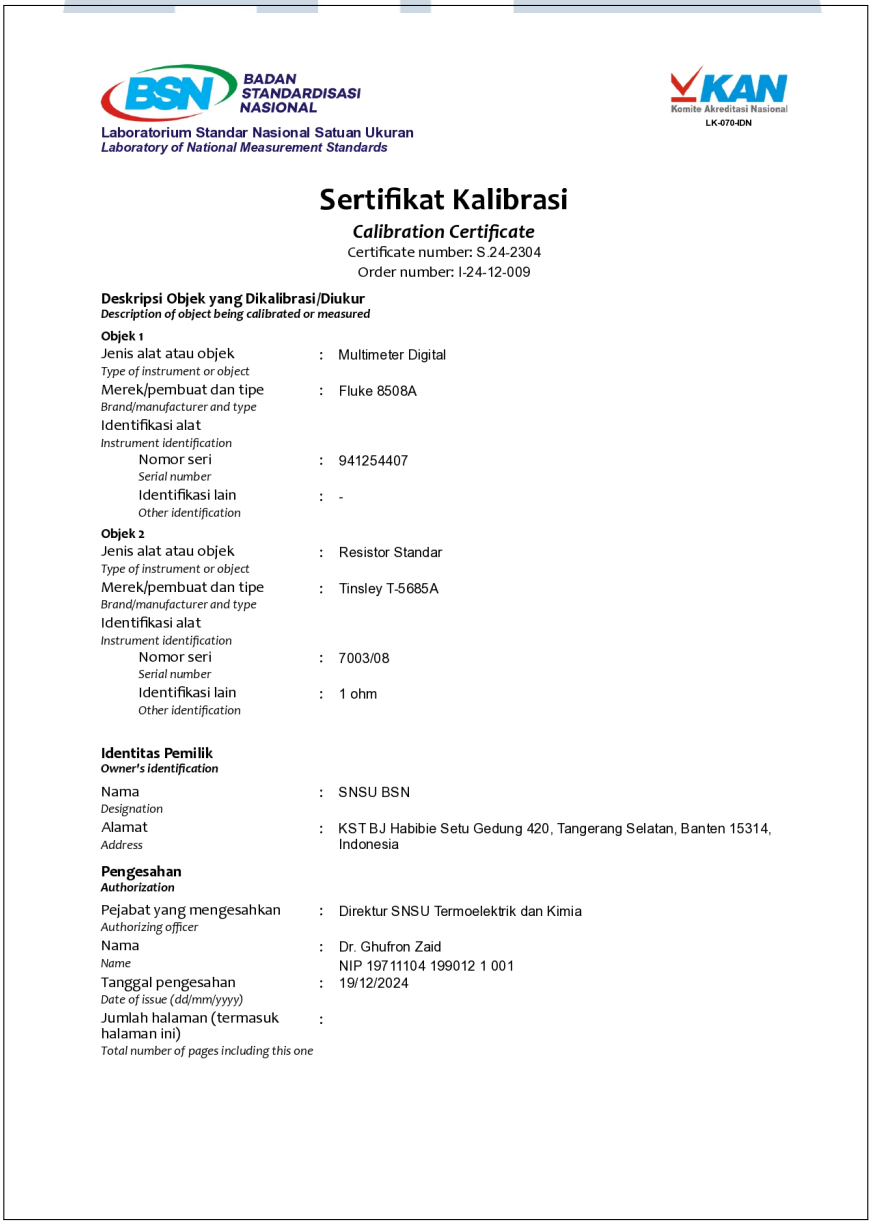


Gambar 3.47. *Flowchart* konversi input menjadi *file* PDF sertifikat dengan *template* HTML

```
data = {
    'objects': [
        {
            'jenis': {
                'id': 'Multimeter Digital',
                'en': 'Digital Multimeter',
            },
        },
    ],
    'uncertainty': {
      'probability': '0.95',
    },
    'statements': [
        {
            'formula': '1\\frac{2}{3}',
        },
    ],
}
```

Kode 3.10: Cuplikan kode data input untuk *template* HTML

Angka ketidakpastian yang diinput di formulir berupa angka desimal yang berkoma. Namun, angka ketidakpastian yang ada di *file* PDF berupa persentase. Oleh karena itu, angka desimal tersebut harus diolah menjadi persentase. Selain itu, karena rumus matematika yang berformat LaTeX atau MathML tidak bisa di-*render* langsung oleh Weasyprint, maka rumus tersebut harus dikonversi ke gambar dahulu, menggunakan *library* `matplotlib`. Setelah data input diolah, *file template* HTML di-*render* dengan data input dan dikonversi menjadi *file* PDF. *File* PDF yang dihasilkan ditunjukkan pada Gambar 3.48.



BADAN STANDARDISASI NASIONAL

Laboratorium Standar Nasional Satuan Ukuran
Laboratory of National Measurement Standards

KAN Komite Akreditasi Nasional
LK-070-IDN

Sertifikat Kalibrasi

Calibration Certificate

Certificate number: S.24-2304
Order number: I-24-12-009

Deskripsi Objek yang Dikalibrasi/Diukur
Description of object being calibrated or measured

Objek 1

Jenis alat atau objek *Type of instrument or object*	:	Multimeter Digital
Merek/pembuat dan tipe *Brand/manufacturer and type*	:	Fluke 8508A
Identifikasi alat *Instrument identification*		
Nomor seri *Serial number*	:	941254407
Identifikasi lain *Other identification*	:	-

Objek 2

Jenis alat atau objek *Type of instrument or object*	:	Resistor Standar
Merek/pembuat dan tipe *Brand/manufacturer and type*	:	Tinsley T-5685A
Identifikasi alat *Instrument identification*		
Nomor seri *Serial number*	:	7003/08
Identifikasi lain *Other identification*	:	1 ohm

Identitas Pemilik
Owner's identification

Nama *Designation*	:	SNSU BSN
Alamat *Address*	:	KST BJ Habibie Setu Gedung 420, Tangerang Selatan, Banten 15314, Indonesia

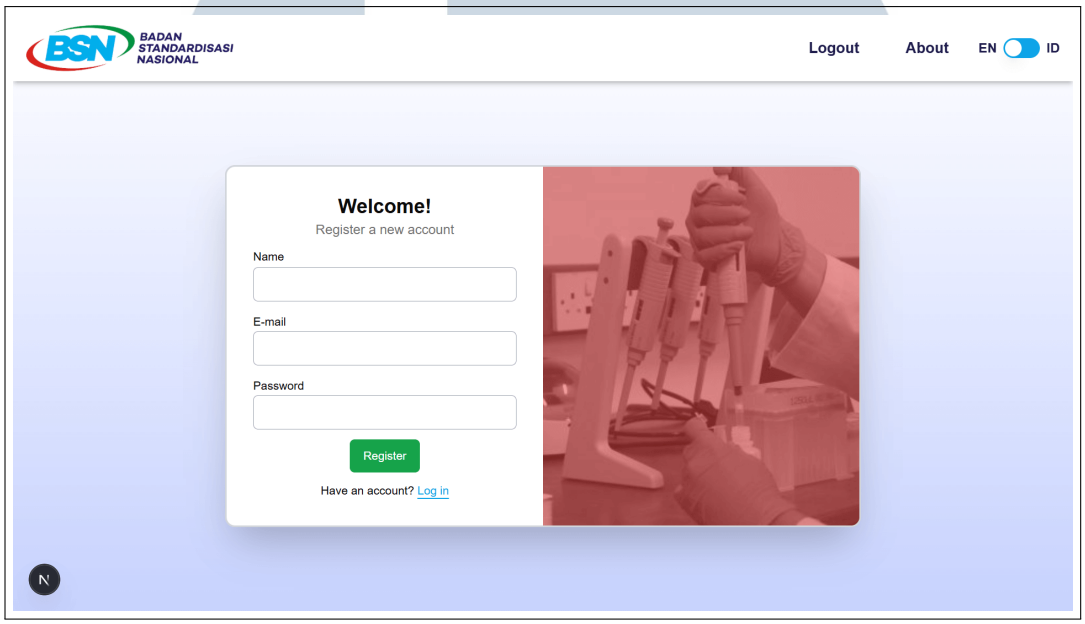
Pengesahan
Authorization

Pejabat yang mengesahkan *Authorizing officer*	:	Direktur SNSU Termoelektrik dan Kimia
Nama *Name*	:	Dr. Ghufron Zaid NIP 19711104 199012 1 001
Tanggal pengesahan *Date of issue (dd/mm/yyyy)*	:	19/12/2024
Jumlah halaman (termasuk halaman ini) *Total number of pages including this one*	:	

Gambar 3.48. Cuplikan *file* PDF dari *template* HTML

3.3.18 Halaman Register

Halaman *register* digunakan oleh pengguna untuk membuat akun baru. Halaman ini berada pada *path* `/register` dan ditunjukkan pada Gambar 3.49. Apabila pengguna sudah memiliki akun, tersedia *link* menuju halaman *login* yang dapat ditekan untuk beralih ke halaman tersebut.



Gambar 3.49. Halaman *register*

3.4 Kendala dan Solusi yang Ditemukan

Dalam pelaksanaan magang, beberapa kendala yang dihadapi meliputi:

- Tidak adanya mentor IT di perusahaan. Hal ini membuat proses pengembangan menjadi lebih menantang karena tidak ada pihak dengan latar belakang IT yang dapat membantu secara langsung saat mengalami kesulitan teknis.
- Penyesuaian data dari *front-end* ke *back-end* agar sesuai dengan struktur dan format yang diharapkan. Ketidaksesuaian seperti struktur JSON yang berbeda atau tipe data yang tidak sesuai sempat menghambat proses integrasi.

Adapun solusi yang dilakukan untuk mengatasi kendala-kendala tersebut antara lain:

- Dilakukan riset mandiri dengan mencari referensi pembelajaran. Diskusi dengan rekan magang juga dimanfaatkan untuk saling memberi masukan dan solusi. Selain itu, konsultasi dengan dosen pembimbing dari kampus dilakukan guna memperoleh arahan terkait pendekatan teknis yang digunakan.
- Menganalisis langsung kode *back-end*, memahami logika yang digunakan, melakukan *debugging*, serta menguji pengiriman data secara langsung. Diskusi dengan rekan magang yang menangani *back-end* juga turut membantu dalam menemukan solusi yang lebih efisien.