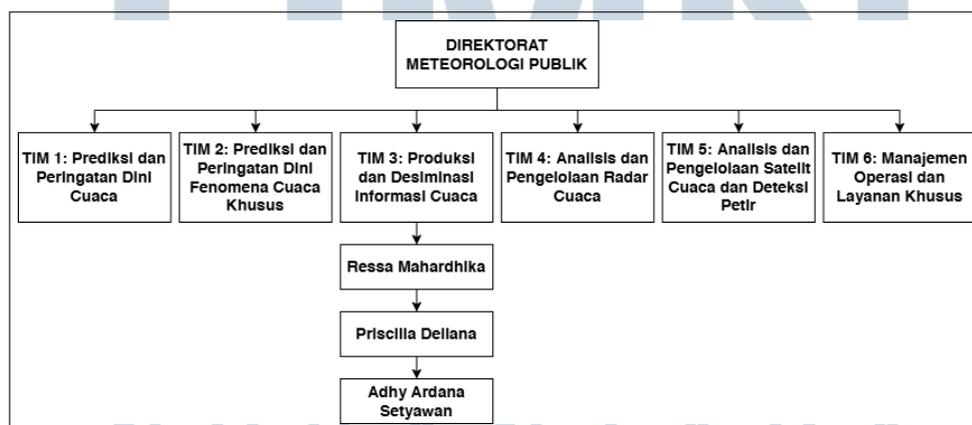


BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kegiatan kerja magang dilaksanakan selama empat bulan sebagai *staff IT* di divisi Direktorat Meteorologi Publik. Kegiatan magang ini didampingi oleh Ibu Priscilla Deliana selaku anggota tim tiga yang berfokus untuk produksi dan diseminasi informasi cuaca. Struktur dari Direktorat Meteorologi Publik dapat dilihat pada gambar 3.1. *staff IT* bertanggung jawab terhadap pembuatan aplikasi pencatatan laporan harian untuk keperluan internal. Selama menjalani magang, koordinasi kerja dengan karyawan lain dilakukan melalui *WhatsApp*. Supervisi juga mengadakan *meeting* secara langsung (tatap muka) untuk memantau perkembangan dari pekerjaan yang telah diberikan. *Meeting* dilakukan satu sampai dua kali seminggu menyesuaikan dengan jadwal supervisi. Selain pembuatan aplikasi, *staff IT* juga bertanggung jawab terhadap pemeliharaan perangkat lunak dan perangkat keras yang digunakan untuk kegiatan produksi harian. Tanggung jawab ini mencakup memastikan perangkat lunak tetap berfungsi dengan baik dan juga harus memastikan perangkat keras, seperti komputer, jaringan, dan peralatan pendukung lainnya, selalu dalam kondisi optimal dengan melakukan pemantauan performa dan perbaikan jika diperlukan. Peran ini sangat penting untuk menjamin kelancaran dan efisiensi proses produksi harian.



Gambar 3.1. Struktur organisasi Direktorat Meteorologi Publik

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang di Badan Meteorologi, Klimatologi, dan Geofisika, tugas yang diberikan adalah pembuatan aplikasi berbasis *desktop* untuk melakukan pencatatan laporan harian. Pencatatan laporan harian diperlukan sebagai bukti bahwa pegawai telah melakukan tugasnya di hari itu. Laporan harian berisi data diri pegawai dan bukti gambar bahwa pegawai telah melakukan pekerjaannya. Aplikasi dibuat menggunakan bahasa pemrograman Python karena Python merupakan bahasa pemrograman yang *open source*, fleksibel, serta ramah bagi pemula [5]. Selain itu, beberapa aplikasi yang telah dibuat di tim tiga menggunakan bahasa pemrograman Python sehingga dapat memudahkan pegawai untuk memantau dan memodifikasi kode yang dibuat. *Library* yang digunakan adalah Tkinter dan CustomTkinter untuk membuat tampilan pengguna. Alasan menggunakan *library* ini adalah karena beberapa aplikasi sebelumnya menggunakan *library* ini sehingga *library* ini familiar bagi pegawai di tim tiga. Selain itu, aplikasi ini juga menggunakan *library* FPDF untuk mengonversi input pengguna ke dalam bentuk PDF.

Dalam pembuatan aplikasi ini terdapat fitur untuk memilih *shift*. *shift* berfungsi sebagai role yang akan dipilih oleh pengguna. Setiap *shift* yang dipilih akan menampilkan menu yang berbeda karena setiap *shift* memiliki tugas yang berbeda. Terdapat tiga *shift* pada aplikasi ini yaitu PS1 (Pagi Siang 1), PS3 (Pagi Siang 3), dan PSM (Pagi Siang Malam). Jika pengguna tidak memilih salah satu di antara tiga *shift* yang telah disediakan, maka menu untuk menginput gambar tidak akan muncul. Selain pembuatan aplikasi, *staff IT* juga bertanggung jawab terhadap pemeliharaan perangkat keras serta perangkat lunak yang dipakai dalam memproduksi produk cuaca.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang di BMKG untuk pembuatan aplikasi pencatatan laporan harian yang berbasis *desktop* diuraikan pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Pengenalan lingkungan kerja dan briefing untuk pembuatan aplikasi pencatatan laporan harian
2	Melakukan proses pra produksi yang terdiri dari riset untuk pembuatan aplikasi, pembuatan <i>flowchart</i> , dan pembuatan <i>user requirement</i>
3	Mempelajari <i>library</i> yang digunakan untuk pembuatan aplikasi yaitu CustomTkinter dan FPDF
4	Membuat tampilan untuk pengguna menginput data diri dan unggah gambar
5	Membuat menu dan fitur untuk konversi input pengguna menjadi PDF
6	Membuat fitur untuk mengunggah beberapa gambar dan menghapus gambar
7	Melakukan tes terhadap aplikasi dan memperbaiki bug yang ada
8	Menambahkan fitur pemilihan <i>shift</i> untuk menampilkan tugas sesuai dengan <i>shift</i> yang dipilih
9	Melakukan optimasi terhadap kode supaya program tidak terlalu berat
10	Melakukan tes terhadap aplikasi dan memperbaiki <i>bug</i> yang ada
11	Melakukan konversi dari final script ke <i>executable</i> program menggunakan auto-py-to-exe
12	Melakukan instalasi aplikasi ke komputer dan evaluasi hasil proyek

3.3.1 User Requirement dan Perangkat Penunjang

Dalam pengembangan aplikasi pencatatan laporan harian, terdapat beberapa *requirement* yang perlu diperhatikan.

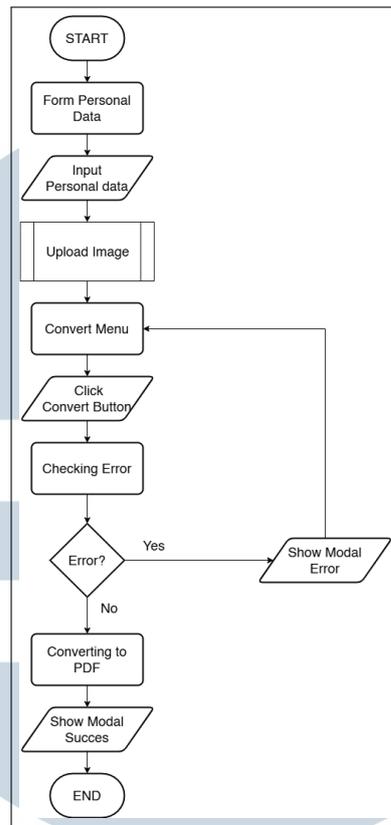
1. Halaman *form* untuk mengisi data diri dan juga tanggal pembuatan laporan.
2. Fitur untuk mengunggah gambar dan menghapus gambar serta kolom untuk memberikan catatan.
3. Fitur untuk mengkonversi input yang diberikan oleh pengguna menjadi *file* dalam format PDF.

Dalam pengembangan aplikasi pencatatan laporan harian, terdapat beberapa perangkat penunjang.

1. Python: bahasa pemrograman Python adalah salah satu pilihan utama dalam pengembangan aplikasi *desktop* karena kesederhanaannya dan ekosistemnya yang kaya. Kemampuan Python untuk mendukung sistem operasi Linux, macOS, serta Windows menjadikannya ideal untuk aplikasi lintas platform [7].
2. CustomTkinter: adalah *library* berbasis Python yang dibangun di atas Tkinter untuk mempermudah pembuatan aplikasi *desktop* dengan antarmuka modern. Keunggulan lain dari CustomTkinter adalah kemampuannya untuk mendukung fitur seperti tombol, slider, progress bar, dan kotak teks dengan gaya yang lebih modern dan responsif [8].
3. FPDF: adalah *library* Python yang digunakan untuk membuat dokumen PDF secara programatis. *library* ini menawarkan cara yang sederhana dan fleksibel untuk menghasilkan dokumen PDF, tanpa memerlukan instalasi perangkat lunak tambahan [9].
4. auto-py-to-exe: adalah alat berbasis Python yang mempermudah proses konversi *file* Python (.py) menjadi *file* eksekusi (.exe) untuk platform Windows. Dengan antarmuka grafis (GUI) yang intuitif, alat ini memungkinkan pengembang untuk mengonfigurasi berbagai opsi konversi tanpa harus mengingat sintaks kompleks dari *library* PyInstaller, yang menjadi inti dari proses konversi [10].

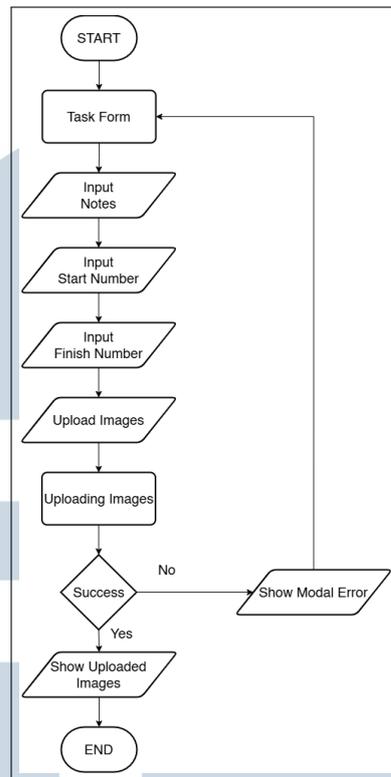
3.3.2 Perancangan Sistem

Pembuatan aplikasi pencatatan laporan harian terdiri dari pra produksi, produksi, dan pasca produksi. Tahapan yang dilakukan pada pra produksi berfokus untuk melakukan riset serta pembuatan *flowchart*. Tujuan dari pembuatan *flowchart* adalah untuk mengetahui alur dari aplikasi yang akan dibuat. *Flowchart* dari aplikasi pencatatan laporan harian dapat dilihat pada Gambar 3.2.



Gambar 3.2. *Flowchart* aplikasi pencatatan laporan harian

Ketika pengguna menjalankan aplikasi, halaman yang akan tampil pertama adalah halaman *form* untuk mengisi data diri. Halaman *form* mengharuskan pengguna untuk mengisi kolom yang telah disediakan. Halaman berikutnya adalah *form* tugas untuk mengunggah gambar. Halaman *form* tugas, pengguna diminta untuk mengunggah gambar dan terdapat kolom. Setelah gambar berhasil diunggah, pengguna bisa mengonversi hasil gambar yang *input* menjadi bentuk PDF dengan menekan tombol yang ada pada menu konversi PDF. Setelah tombol ditekan, aplikasi akan mengecek apakah ada *error* pada program ketika mengonversi. Aplikasi akan menampilkan modal *error* jika ada kesalahan dalam program, sedangkan aplikasi akan membuat *file* dengan format PDF jika berhasil. Pada aplikasi pencatatan laporan, fitur mengunggah gambar merupakan fitur yang penting dalam aplikasi ini. *Flowchart* dari fitur mengunggah gambar dapat dilihat pada Gambar 3.3.

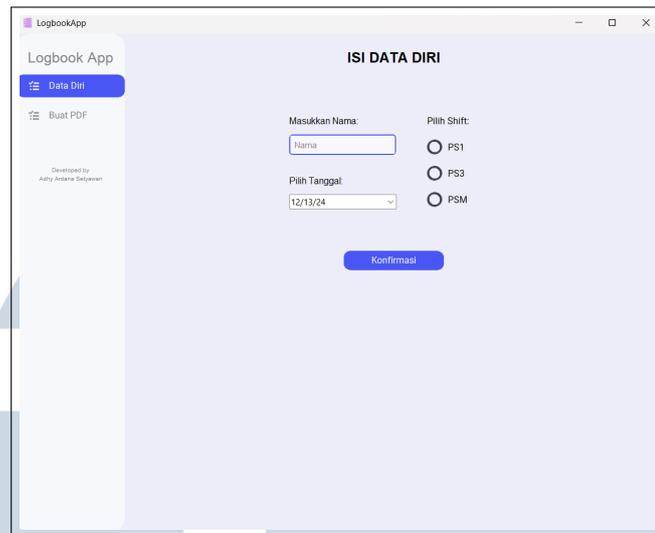


Gambar 3.3. *Flowchart Upload Image*

Fitur mengunggah gambar terdapat pada halaman *form* tugas. Halaman *form* tugas memiliki kolom catatan yang bisa diisi oleh pengguna. Kolom catatan berfungsi untuk memberikan keterangan terkait tugas yang telah dilakukan. Setelah kolom catatan terdapat kolom untuk memasukkan angka mulai dan angka selesai. Kolom angka mulai berfungsi untuk menentukan gambar ke berapa yang ingin diunggah, dan kolom angka selesai untuk menentukan sampai gambar ke berapa yang diunggah. Pengguna diharuskan menekan tombol upload setelah memasukkan angka mulai dan angka selesai. Aplikasi akan menampilkan modal *error* jika angka yang dimasukkan tidak sesuai dengan ketentuan. Setelah pengguna berhasil mengunggah gambar, gambar yang telah diunggah akan tampil pada halaman tersebut.

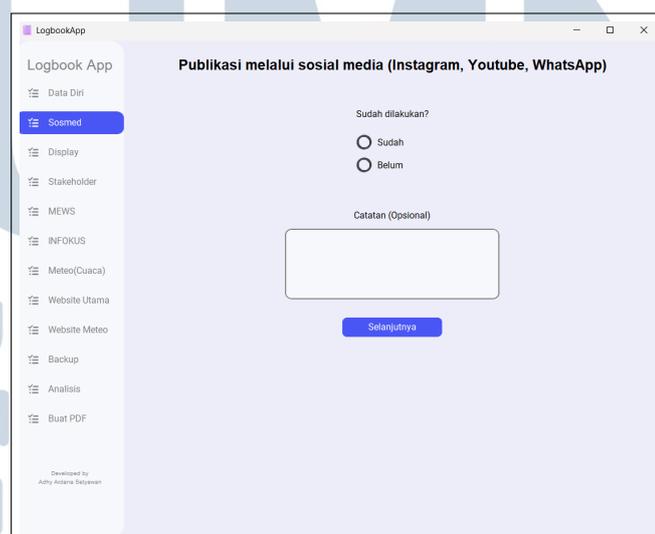
3.3.3 Implementasi

Aplikasi pencatatan laporan harian terdiri dari tiga halaman utama yaitu halaman *form* data diri, halaman *form* tugas, dan halaman konversi ke PDF. Tampilan halaman *form* data diri bisa dilihat pada Gambar 3.4.



Gambar 3.4. Halaman *form* data diri

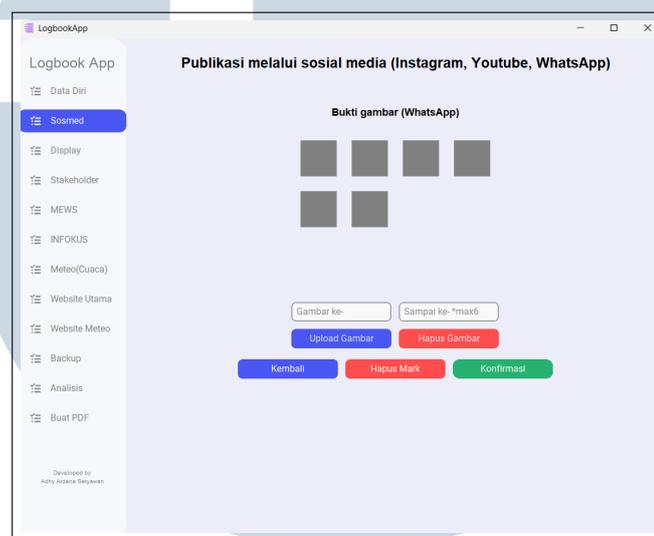
Halaman *form* data diri mengharuskan pengguna untuk mengisi nama dan menentukan tanggal dibuatnya laporan harian serta memilih *shift* yang tersedia. Setiap *shift* memiliki tugas yang berbeda, sehingga halaman tugas yang akan muncul juga akan berbeda-beda sesuai dengan *shift* yang dipilih. Setelah mengisi semua data yang dibutuhkan, pengguna harus menekan tombol konfirmasi untuk bisa lanjut ke halaman berikutnya. Halaman *form* tugas adalah halaman yang bisa diakses ketika pengguna telah melakukan konfirmasi data diri. Tampilan halaman *form* tugas bisa dilihat pada Gambar 3.5.



Gambar 3.5. Halaman *form* tugas

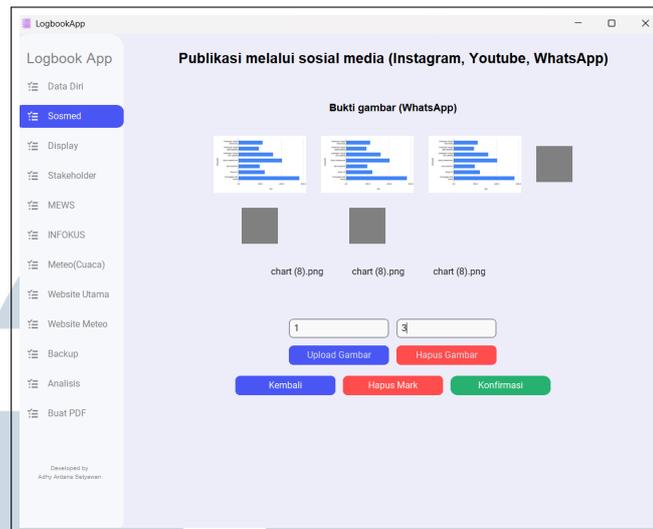
Pada *sidebar* di sebelah kiri akan muncul tombol menu sesuai dengan *shift* yang telah dipilih. Halaman *form* tugas akan muncul di bagian sebelah

kanan ketika tombol menu ditekan. Halaman form tugas akan menampilkan *radio button* untuk mengonfirmasi tugas yang telah dilakukan dan juga kolom catatan untuk memberikan keterangan singkat dari tugas yang dilakukan. Setelah pengguna mengisi data yang dibutuhkan, pengguna diminta untuk menekan tombol selanjutnya untuk masuk ke menu unggah gambar. Tampilan menu unggah gambar bisa dilihat pada Gambar 3.6



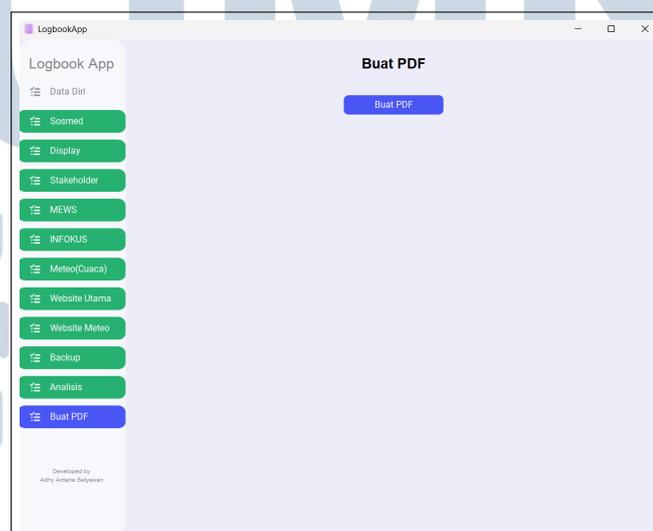
Gambar 3.6. Halaman *form* unggah gambar

Kotak berwarna abu-abu merupakan *display* untuk gambar yang akan diunggah. Sebelum pengguna mengunggah gambar, pengguna diharuskan untuk mengisi kolom angka yang berada di bawah *display*. Kolom tersebut berfungsi untuk menentukan jumlah gambar yang akan diunggah. Kolom pertama adalah kolom untuk menentukan mulai dari gambar ke berapa yang ingin diunggah, sedangkan kolom kedua adalah kolom untuk menentukan sampai gambar ke berapa yang ingin diunggah. Contoh penggunaannya adalah jika pengguna ingin mengunggah tiga gambar, maka pengguna harus memasukkan angka satu pada kolom pertama dan angka tiga pada kolom kedua. Gambar yang akan diunggah oleh pengguna akan disimpan ke *variable array* pada program. Pengguna diminta untuk menekan tombol *upload* gambar setelah mengisi kolom angka. Aplikasi akan menampilkan modal *error* ketika *variable* yang dimasukkan bukan angka atau angka yang dimasukkan tidak sesuai. Contoh tampilan setelah gambar diunggah bisa dilihat pada Gambar 3.7



Gambar 3.7. Halaman setelah gambar diunggah

Halaman *form* tugas memiliki fitur untuk menghapus gambar yang telah diunggah. Cara penggunaannya sama seperti mengunggah gambar, yaitu pengguna harus mengisi kolom angka untuk menentukan gambar ke berapa yang ingin dihapus. Pengguna harus menekan tombol hapus gambar setelah mengisi kolom angka untuk menghapus gambar yang telah diunggah. Setelah pengguna mengisi semua data yang dibutuhkan, pengguna dapat menekan tombol konfirmasi atau tombol menu pada *sidebar* untuk lanjut ke halaman berikutnya. Halaman yang terakhir adalah halaman untuk mengonversi data yang *diinput* oleh pengguna menjadi *file* dengan format PDF. Tampilan halaman ini dapat dilihat pada gambar 3.8



Gambar 3.8. Halaman konversi

Halaman hanya berisikan tombol untuk menjalankan proses konversi. Aplikasi akan menampilkan modal *error* jika proses konversi gagal, dan akan menampilkan modal berhasil jika konversi berhasil dilakukan. *File* hasil konversi akan disimpan pada *folder* bernama PDF di tempat yang sama dengan aplikasi.

3.3.4 Kendala dan Solusi

Terdapat beberapa kendala yang dialami selama pelaksanaan magang yang dilakukan di BMKG, yaitu: **Kendala:**

1. Perbedaan ukuran gambar yang diunggah oleh pengguna berbeda-beda. Gambar yang diunggah oleh pengguna berasal dari hasil *screenshot* atau hasil tangkapan kamera ponsel. Kendala ini membuat kepada hasil susunan PDF yang dibuat menjadi berantakan karena ukuran gambar tidak sesuai dengan yang ditentukan.
2. Teks yang berubah warna ketika komputer memakai tema terang atau gelap. Aplikasi dikembangkan di komputer yang menggunakan tema terang pada *windows*, sehingga beberapa tulisan susah terlihat ketika aplikasi dijalankan di komputer dengan tema *windows* yang gelap karena warna teks berubah sesuai dengan tema pada komputer.

Solusi:

1. Membuat kode untuk mengubah ukuran gambar yang diunggah oleh pengguna. Gambar yang diunggah oleh pengguna akan dirubah ukurannya oleh aplikasi sesuai dengan ukuran yang ditetapkan. Aplikasi akan merubah ukuran tinggi gambar yang diunggah, sehingga semua gambar memiliki tinggi yang sama.
2. Membuat *variable* pada kode yang berisikan kode warna untuk digunakan pada setiap teks. Warna teks tidak akan berubah sesuai tema komputer karena telah diberikan *variable* warna yang tetap. Penggunaan *variable* warna juga dapat diterapkan untuk tombol dan kolom yang ada pada aplikasi.