PENGEMBANGAN SISTEM OTOMATISASI PENGELOLAAN DATA EXCEL BERBASIS MICROSOFT POWER APPS



Disusun Oleh:

N a m a : Satriya Adhi Pradana
NIM : 21523090

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN SISTEM OTOMATISASI PENGELOLAAN DATA EXCEL BERBASIS MICROSOFT POWER APPS

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

N a m a : Satriya Adhi Pradana
NIM : 21523090

Yogyakarta, 23 Juli 2025

Pembimbing,

(Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs.)

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN SISTEM OTOMATISASI PENGELOLAAN DATA EXCEL BERBASIS MICROSOFT POWER APPS

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 23 Juli 2025

Tim Penguji

Nur Wijayaning Rahayu, S.Kom., M.Cs. _____

Anggota 1

Novi Setiani, S.T., M.T. _____

Anggota 2

Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Satriya Adhi Pradana
NIM	:	21523090

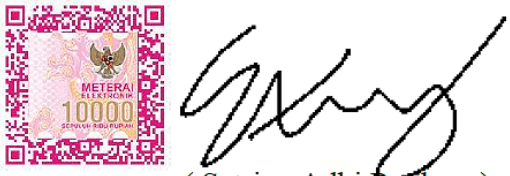
Tugas akhir dengan judul:

PENGEMBANGAN SISTEM OTOMATISASI PENGELOLAAN DATA EXCEL BERBASIS MICROSOFT POWER APPS

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya penulis sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 23 Juli 2025



(Satriya Adhi Pradana)

HALAMAN PERSEMBAHAN

Alhamdulillah, segala puji syukur bagi Allah SWT Yang Maha Pengasih dan Maha Penyayang. Berkat petunjuk, rahmat serta hidayat-Nya penulis berhasil menyelesaikan laporan Tugas Akhir ini. Tidak lupa sholawat serta salam senantiasa dilimpahkan kepada Nabi Muhamad SAW, beserta keluarganya, sahabat, dan pengikutnya hingga hari akhir. Laporan Tugas Akhir ini saya persembahkan kepada:

1. Kedua orangtua yang saya cintai, Bapak Mujamil dan Ibu Siti Nurlaela yang selalu memberikan cinta dan kasih sayang, serta doa dan dukungan, sehingga saya dapat menyelesaikan studi di Universitas Islam Indonesia. Tugas akhir ini saya persembahkan untuk mereka.
2. Ibu Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs., selaku dosen pembimbing yang telah membimbing dan memberikan saran serta masukan selama penyusunan laporan tugas akhir ini.
3. Teman-teman yang selalu menyemangati, memberi dukungan, serta mengingatkan saya untuk menyelesaikan penyusunan laporan tugas akhir ini. Terima kasih saya ucapkan semoga Allah SWT selalu memberikan karuniaNya berupa kesehatan, rezeki, dan perlindungan. Aamin ya Rabbal Alamin.

HALAMAN MOTTO

“Success isn't always about greatness. It's about consistency. Consistent hard work leads to success. Greatness will come.”

- Dwayne Johnson

"Keberhasilan bukanlah milik orang yang pintar. Keberhasilan adalah kepunyaan mereka yang senantiasa berusaha."

- B.J. Habibie

“Menuntut ilmu itu wajib bagi setiap muslim"

- HR. Ibnu Majah

"Barangsiapa menempuh jalan untuk mencari ilmu, maka Allah akan mudahkan baginya jalan menuju surga"

- HR. Muslim

KATA PENGANTAR

Alhamdulillah penulis panjatkan ke hadirat Allah SWT, yang telah melimpahkan rahmat, taufik, serta inayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul:

"PENGEMBANGAN SISTEM OTOMATISASI PENGELOLAAN DATA EXCEL BERBASIS MICROSOFT POWER APPS."

Penulis menyadari bahwa penyusunan laporan ini masih jauh dari kata sempurna. Namun, besar harapan penulis agar laporan ini dapat memberikan manfaat, terutama sebagai sumber informasi dan tambahan pengetahuan bagi para pembaca. Oleh karena itu, penulis sangat terbuka terhadap segala bentuk kritik dan saran yang membangun, demi perbaikan di masa mendatang.

Laporan ini disusun untuk memenuhi salah satu syarat akademik dalam program tugas akhir jalur magang di Fakultas Teknologi Industri, Jurusan Informatika, Universitas Islam Indonesia. Selama proses penyusunan laporan ini, penulis telah menerima banyak bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Ibu Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs., selaku dosen pembimbing tugas akhir, yang telah meluangkan waktu, pikiran, dan perhatian dalam memberikan bimbingan kepada penulis.
2. Kedua orangtua yang saya cintai, Bapak Mujamil dan Ibu Siti Nurlaela serta keluarga besar yang selalu memberikan doa dan dukungan secara moral maupun spiritual.
3. Kedua kakak, Mas Adit dan Mbak Nesya yang selalu memberikan dukungan untuk semangat kuliah.
4. Bank BTN Syariah Kantor Cabang Serang, beserta seluruh staf dan jajaran yang telah memberikan kesempatan magang serta membantu penulis selama proses kegiatan dan penyusunan laporan ini.
5. Rekan-rekan satu tim dan pembimbing di tempat magang, yang telah memberikan arahan, kerjasama, dan semangat selama pelaksanaan tugas akhir.
6. Teman-teman Tadika Mesra yang selalu supportif dan asik, dan teman – teman FS MOTO dari Cilegon, serta teman hangat yang turut memberikan motivasi dan menemani proses penyelesaian laporan ini.
7. Dan untuk diri saya sendiri, terima kasih sudah memilih untuk bertahan, terus mencoba, dan tidak menyerah sampai laporan ini selesai.

Akhir kata, penulis berharap laporan ini dapat bermanfaat bagi semua pihak yang membacanya.

Wassalamu'alaikum warahmatullahi wa barakatuh.

Yogyakarta, 23 Juli 2025



(Satriya Adhi Pradana)

SARI

Transformasi digital menjadi salah satu kebutuhan utama bagi perusahaan dalam meningkatkan efisiensi kerja dan integrasi data secara menyeluruh. Salah satu pendekatan yang banyak digunakan saat ini adalah pengembangan sistem menggunakan platform low-code, karena mampu mempercepat proses pembangunan aplikasi tanpa memerlukan keahlian pemrograman yang kompleks. Pelaksanaan magang ini dilakukan di sebuah perusahaan yang bergerak di Bank BTN Syariah kantor cabang Serang, dengan fokus pada pengembangan sistem Enterprise Resource Planning (ERP) berbasis Microsoft Power Platform, khususnya melalui penggunaan Microsoft Power Apps dan SharePoint sebagai infrastruktur penyimpanan data.

Selama masa magang, penulis bertanggung jawab dalam merancang dan mengimplementasikan aplikasi ERP yang digunakan untuk mengelola data perumahan dan developer. Sistem ini dirancang untuk menggantikan proses pencatatan manual berbasis Excel, dengan menyediakan antarmuka yang lebih interaktif dan terintegrasi. Proyek dikembangkan menggunakan pendekatan Software Development Life Cycle (SDLC) model *Waterfall*, yang dimulai dari tahap analisis kebutuhan, perancangan sistem, implementasi aplikasi, hingga pengujian fungsionalitas.

Aplikasi yang dikembangkan memiliki fitur utama berupa form input data, tampilan galeri untuk menampilkan data perumahan, serta fitur CRUD (*Create, Read, Update, Delete*) yang terhubung langsung dengan SharePoint Lists. Selain itu, Power Automate juga digunakan untuk mengotomatisasi alur kerja, seperti pengiriman notifikasi atau pencatatan otomatis ke penyimpanan cloud. Berdasarkan hasil pengujian, aplikasi mampu mempercepat proses pelaporan data dari lapangan ke pusat, serta meningkatkan akurasi dan keteraturan pengelolaan data. Kegiatan magang ini memberikan pengalaman praktis dalam mengimplementasikan solusi digital berbasis platform low-code, sekaligus memperkuat pemahaman terhadap penerapan teknologi informasi dalam dunia industri.

Kata kunci: Power Apps, ERP, SharePoint, SDLC, Low-Code, Power Automate, Waterfall.

GLOSARIUM

Waterfall	Metode pengembangan perangkat lunak yang memiliki ciri sistematis berurutan.
Bug	Suatu error yang terjadi pada perangkat lunak.
Frontend	Bagian yang berkaitan dengan tampilan yang menyuguhkan tampilan yang menarik.
Backend	Bagian yang berkaitan dengan pemrograman sistem dalam bentuk code/script.

DAFTAR ISI

HALAMAN JUDUL i
1 HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
2 HALAMAN PENGESAHAN DOSEN PENGUJI iii
3 HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
4 HALAMAN PERSEMBAHAN v
5 HALAMAN MOTTO vi
6 KATA PENGANTAR vii
7 SARI ix
8 GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiv
DAFTAR GAMBAR xiv
1 BAB I PENDAHULUAN 1
1.1 Profil Institusi Magang 1
1.1.1 Gambaran Umum 1
1.1.2 Visi dan Misi Bank BTN Syariah 1
1.1.3 Struktur Organisasi 2
1.1.4 Lokasi dan Fasilitas Kantor Bank BTN Syariah 2
1.1.5 Profil dan Sejarah Institusi 4
1.2 Latar Belakang 5
1.3 Ruang Lingkup 6
1.4 Tujuan 6
1.5 Manfaat 7
1.6 Sistematika Penulisan 7
2 BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA 9
2.1 *Software Development Life Cycle* (SDLC) model *Waterfall* 9
2.2 Microsoft Power Apps 9
2.3 Power Apps Studio 12
2.4 Microsoft Power Automate 14

2.5 Notion 15
2.6 Tinjauan Pustaka 16
2.6.1 Model SDLC dengan *Waterfall* 17
2.6.2 *Low-Code Development Platform* (LCDP) 18
2.6.3 Digitalisasi pelaporan Excel menggunakan LCDP 18
3 BAB III PELAKSANAAN MAGANG 19
3.1 Manajemen Proyek 19
3.2 Penerapan Metode *Software Development Life Cycle (SDLC) model Waterfall* 20
3.3 Analisis Kebutuhan 20
3.4 Pengembangan Desain 22
3.4.1 *Database* dan *data source* Microsoft Power Apps 22
3.4.2 *Data Source* menggunakan Dataverse 23
3.4.3 *Database record* menggunakan Microsoft SharePoint Lists 26
3.4.4 Otomatisasi *flow* menggunakan Microsoft Power Automate 27
3.4.5 Hasil Otomatisasi File Excel 27
3.4.6 *Desain Flow* aplikasi eLPA 31
3.4.7 Desain fitur form halaman detail agunan 32
3.4.8 Desain fitur form halaman tambah perumahan 33
3.4.9 Desain fitur form halaman foto agunan 34
3.4.10 Desain fitur form halaman spesifikasi bangunan 35
3.4.11 Desain fitur form halaman bangunan 36
3.4.12 Desain fitur halaman lihat data 37
3.5 Implementasi 38
3.6 *Testing*/Pengujian 41
3.7 Maintenance 45
3.8 Penutupan Proyek 45
4 BAB IV REFLEKSI PELAKSANAAN MAGANG 46
4.1 Relevansi Akademik 46

4.2 Lingkungan Magang 46
4.2.1 Aktivitas Harian 46
4.2.2 Pengembangan Softskill, Hardskill, dan Pembelajaran 47
4.3 Pembelajaran Magang 47
4.3.1 Manfaat 47
4.3.2 Kendala, Hambatan dan Tantangan 48
4.4 Pembelajaran Magang 48
4.4.1 Pemahaman Platform Low-Code Microsoft Power Platform 49
4.4.2 Pemahaman Penggunaan Rumus dan Fungsi dalam Excel 49
4.4.3 Pemahaman Manajemen Data dan Relasi 49
4.4.4 Penerapan Otomatisasi Proses Bisnis 49
5 BAB V PENUTUP 51
5.1 Kesimpulan 51
5.2 Saran 52
6 DAFTAR PUSTAKA 53
LAMPIRAN 56

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka Implementasi ERP berbasis LCDP 16
Tabel 3.1 Aktivitas Magang 19
Tabel 3.2 Hasil pengujian black box 41

DAFTAR GAMBAR

Gambar 1.1 Struktur Organisasi Bank BTN Syariah 2
Gambar 1.2 *Tampak depan Bank BTN Syariah KC Serang* 3
Gambar 1.3 *Tangkapan layar GPS lokasi kantor* 3
Gambar 2.1 Fase yang terdapat pada Software Development Life Cycle (SDLC) model Waterfall 9
Gambar 2.2 *Halaman Power Apps untuk create sebuah app* 10
Gambar 2.3 *Tampilan canvas editor Power Apps Studio* 10
Gambar 2.4 *Template canvas pada Power Apps* 11
Gambar 2.5 *Konektor yang terdapat dalam Power Apps* 12
Gambar 2.6 *Tab insert yang berisikan kontrol pada Power Apps Studio* 13
Gambar 2.7 *Formula bar Power Apps Studio* 13
Gambar 2.8 *Antarmuka Properties pada sebuah control Power Apps* 14
Gambar 2.9 *Tampilan halaman awal Power Automate* 15
Gambar 2.10 *Tampilan utama Notion* 15
Gambar 3.1 Project plan di Notion 20
Gambar 3.2 Data source statis SpesifikasiBangunan menggunakan Dataverse 24
Gambar 3.3 Data source statis kelurahan satu Provinsi Banten menggunakan Dataverse 26
Gambar 3.4 Kolom database SharePoint Lists 27
Gambar 3.5 Tampilan flow Power Automate 27
Gambar 3.6 Tampilan sheet “LPA” 28
Gambar 3.7 Data kelurahan satu Provinsi Banten dalam sheet “DATABASE” 29
Gambar 3.8 Tabel Excel LPA_Record untuk data record dari SharePoint Lists dalam sheet “RECORD” 30
Gambar 3.9 Formula Excel untuk mengambil value 30
Gambar 3.10 Formula Excel untuk mengambil data cell menggunakan fungsi VLOOKUP 30
Gambar 3.11 User flow penggunaan aplikasi 31

Gambar 3.12 Tampilan halaman pertama eLPA 32
Gambar 3.13 Tampilan halaman tambah perumahan 33
Gambar 3.14 Tampilan halaman edit data perumahan 34
Gambar 3.15 Tampilan foto eLPA untuk dokumentasi agunan 34
Gambar 3.16 Tampilan dialog untuk melakukan impor gambar 35
Gambar 3.17 Tampilan halaman spesifikasi agunan 36
Gambar 3.18 Tampilan halaman bangunan 36
Gambar 3.19 Tampilan review form 37
Gambar 3.20 Tampilan lihat data 38
Gambar 3.21 Tampilan halaman bangunan 39
Gambar 3.22 Tampilan dialog publish 39
Gambar 3.23 Tampilan notifikasi update aplikasi 40
Gambar 3.24 Tampilan eLPA yang sudah terimplementasi di ponsel 40
Gambar 3.25 Tampilan preview canvas 42
Gambar 3.26 Tampilan status eksekusi keseluruhan flow 43
Gambar 3.27 *Tampilan status eksekusi flow* 43
Gambar 3.28 Tampilan detail eksekusi salah satu flow dengan raw output JSON 44
Gambar 3.29 Tampilan flow run history Power Automate 44

BAB I
PENDAHULUAN

1.1 Profil Institusi Magang

1.1.1 Gambaran Umum

Bank BTN Syariah adalah unit usaha syariah dari PT Bank Tabungan Negara (Persero) Tbk, yang didirikan untuk mendukung layanan perbankan syariah di Indonesia. Bank ini berfokus pada sektor pembiayaan perumahan berbasis syariah dan memberikan solusi keuangan sesuai prinsip-prinsip Islam. Didirikan pada tahun 2005, Bank BTN Syariah terus berkembang untuk menjadi pilihan utama dalam pembiayaan perumahan dan layanan keuangan syariah.

Dengan kantor pusat yang berlokasi di Jakarta, Bank BTN Syariah memiliki jaringan kantor cabang dan unit pelayanan yang tersebar di seluruh Indonesia untuk melayani masyarakat dengan berbagai produk keuangan yang sesuai dengan kebutuhan nasabah dengan mengedepankan prinsip syariah.

1.1.2 Visi dan Misi Bank BTN Syariah

- Visi:

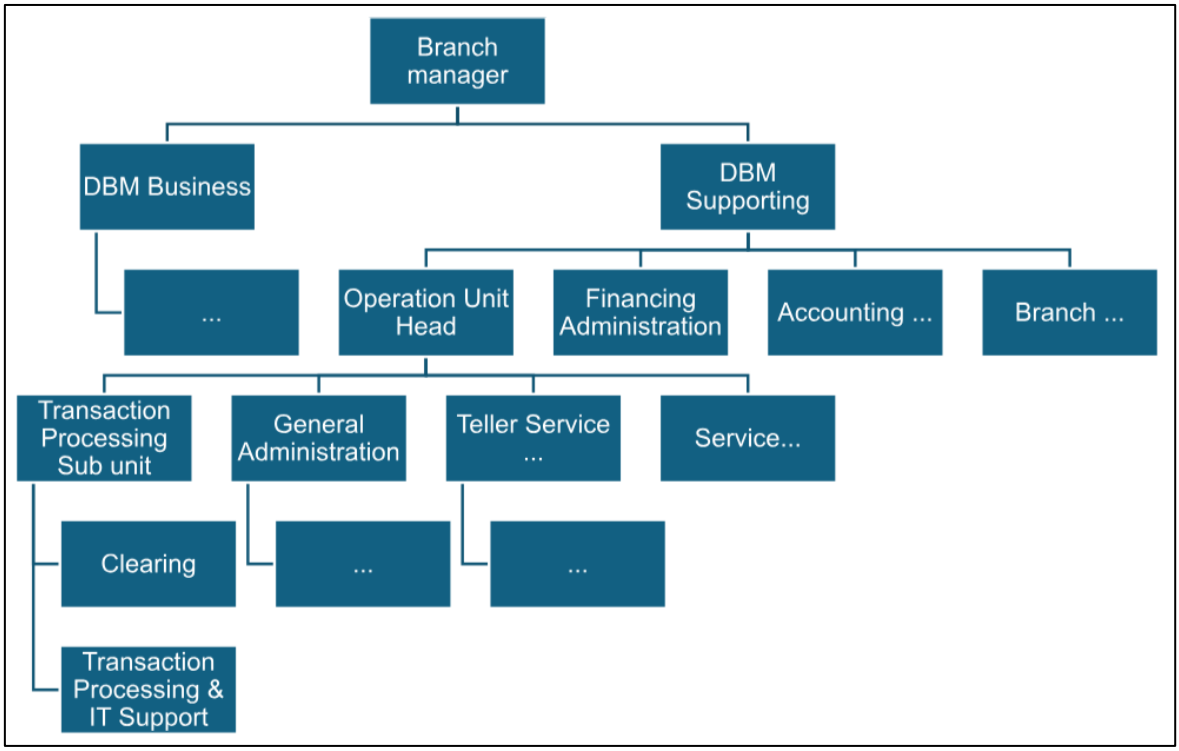
  Terdepan dan terpercaya dalam memfasilitasi sektor perumahan dan jasa layanan keuangan keluarga.

- Misi:

  1. Berperan aktif dalam mendukung sektor perumahan baik dari sisi penawaran maupun permintaan, yang terintegrasi dalam sektor perumahan di Indonesia.
  2. Memberikan layanan unggul dalam pembiayaan kepada sektor perumahan dan kebutuhan keuangan keluarga.
  3. Meningkatkan keunggulan kompetitif melalui inovasi pengembangan produk, jasa, dan jaringan strategis berbasis digital.
  4. Menyiapkan dan mengembangkan sumber daya manusia yang berkualitas, profesional, dan memiliki integritas tinggi.
  5. Meningkatkan nilai bagi pemegang saham dengan fokus pada pertumbuhan profitabilitas sesuai dengan prinsip kehati-hatian dan tata kelola perusahaan yang baik.
  6. Memperhatikan kepentingan masyarakat sosial dan lingkungan secara berkelanjutan.

1.1.3 Struktur Organisasi

Bank BTN Syariah memiliki struktur organisasi yang terdiri dari beberapa divisi utama, antara lain: Divisi Pembiayaan, Divisi Dana dan Jasa, serta Divisi Kepatuhan Syariah.



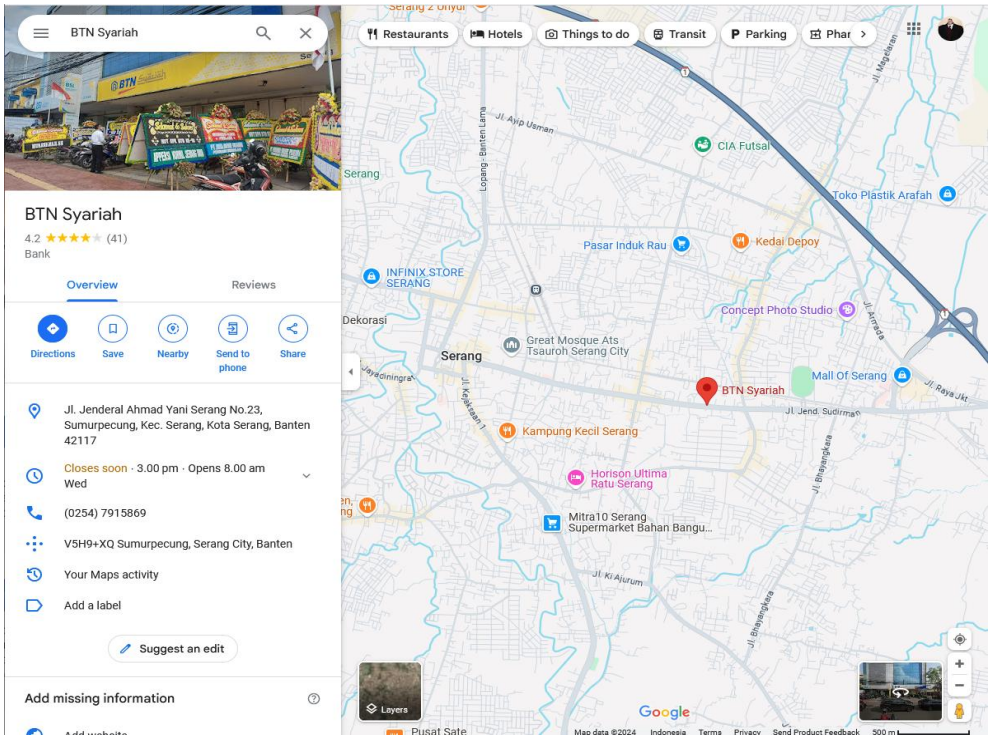
Gambar 1.1 *Struktur Organisasi Bank BTN Syariah*

1.1.4 Lokasi dan Fasilitas Kantor Bank BTN Syariah

Bank BTN Syariah Kantor Cabang Serang berlokasi di Jl. Ahmad Yani No. 175e Cipete Kel. Sumur Pecung, Kec. Serang, Kota Serang, Banten. Kantor cabang ini memiliki fasilitas berikut: ruang rapat, ruang kerja terbuka, dan area layanan pelanggan yang nyaman. Di bawah ini adalah foto tampak depan kantor dan tangkapan layar GPS dari lokasi Bank BTN Syariah Kantor Cabang Serang:



Gambar 1.2 *Tampak depan Bank BTN Syariah KC Serang*



Gambar 1.3 *Tangkapan layar GPS lokasi kantor*

1.1.5 Profil dan Sejarah Institusi

BTN Unit Usaha Syariah, yang dikenal sebagai "BTN Syariah", mengusung motto "Maju dan Sejahtera Bersama". Dalam pelaksanaan kegiatannya, unit usaha ini didampingi oleh Dewan Pengawas Syariah (DPS) yang berperan sebagai:

1. Pengawas independen yang memastikan kepatuhan terhadap prinsip syariah
2. Penasehat strategis bagi Direksi, Pimpinan Divisi Syariah, dan Pimpinan Kantor Cabang Syariah
3. Pemberi saran terkait implementasi prinsip-prinsip syariah dalam operasional bank.

Sejarah

- November 2004: Struktur organisasi kantor cabang syariah PT Bank Tabungan Negara dibentuk.
- 15 Desember 2004: Bank BTN menerima persetujuan dari Bank Indonesia (BI) untuk prinsip Kantor Cabang Syariah (KCS).
- 14 Februari 2005: KCS Jakarta dibuka.

Jaringan

Bank BTN Syariah memiliki jaringan luas di Indonesia, terdiri dari:

- 22 Kantor Cabang Syariah
- 21 Kantor Cabang Pembantu Syariah
- 7 Kantor Kas Syariah
- 240 Kantor Layanan Syariah

Fungsi

Sebagai bank BUMN, BTN Syariah menjalankan fungsi intermediasi:

1. Penghimpunan dana melalui produk giro, tabungan dan deposito.
2. Penyaluran dana ke sektor riil melalui pembiayaan KPR, multiguna, investasi dan modal kerja.
3. Mengutamakan prinsip keadilan, kesetaraan dan keseimbangan antara kepentingan nasabah dan bank.

1.2 Latar Belakang

Perkembangan teknologi informasi dan komunikasi telah mendorong transformasi besar-besaran dalam berbagai aspek bisnis, dengan contoh yang terlihat yaitu proses kerja dari sistem manual menuju sistem digital yang lebih efisien dan terintegrasi. Dalam pelaksanaan transformasi digital, yaitu proses integrasi teknologi ke dalam seluruh aspek operasional bisnis, menuntut perubahan menyeluruh baik dalam aspek teknis, dan juga dalam budaya kerja, pola pikir karyawan, serta kemampuan organisasi dalam merespons perubahan (Khaidir K.N. dkk., 2024). Salah satu solusi digital yang banyak diadopsi dalam proses transformasi ini adalah sistem Enterprise Resource Planning (ERP), yang menyatukan berbagai fungsi bisnis ke dalam satu platform terpusat dan terintegrasi guna meningkatkan efisiensi dan visibilitas proses operasional.

Di sektor perbankan, khususnya pembiayaan perumahan, pelaporan masih banyak dilakukan secara manual menggunakan Excel. Metode ini cenderung memakan waktu, rawan kesalahan, dan menyulitkan distribusi data secara real-time (Razak dkk., 2024). Salah satu contohnya adalah proses penilaian agunan, yang melibatkan petugas lapangan dan penilai internal. Proses pelaporan yang bersifat repetitif menjadi tidak efisien, terutama saat volume permohonan tinggi. Kemudian upaya digitalisasi pun menghadapi tantangan, terutama dalam hal kesiapan SDM. Tidak semua karyawan memiliki kecakapan digital yang cukup, sehingga muncul kesenjangan antara sistem dan penggunanya (Putri & Munawaroh, 2025). Sementara itu, pengguna menuntut sistem yang cepat, responsif, dan mudah diakses. Di sinilah otomatisasi berperan penting untuk mengurangi kerja manual, meningkatkan akurasi, dan mempercepat proses pelaporan (Kakolu & Faheem, 2023).

Salah satu pendekatan modern yang mendukung implementasi otomatisasi adalah penggunaan *Low-Code Development Platform* (LCDP). LCDP memungkinkan pengembangan aplikasi secara cepat dengan sedikit atau tanpa penulisan kode secara manual. Platform ini berbasis cloud melalui model Platform-as-a-Service (PaaS), menyediakan antarmuka visual seperti drag-and-drop, editor WYSIWYG (What You See Is What You Get), serta modul-modul siap pakai (Berardinelli dkk., 2020; Moon, 2020; Phalake & Joshi, 2021), sehingga memungkinkan pengembangan dan melakukan *deploy* aplikasi secara lebih mudah dan mudah dalam melakukan implementasi secara cepat.

Dalam kegiatan magang ini, penulis terlibat langsung dalam pengembangan aplikasi ERP internal berbasis Microsoft Dynamics 365, dengan produk yang dipakai yaitu Power Apps, Power Automate, dan SharePoint Lists. Microsoft Power Apps digunakan untuk membangun antarmuka pelaporan berbasis formulir, Microsoft Power Automate untuk merancang alur kerja

otomatis (workflow), dan SharePoint Lists sebagai tempat penyimpanan data. Data yang telah dilaporkan oleh pengguna kemudian diekspor dalam format Excel dan secara otomatis tersimpan di OneDrive (Boppana, 2025; Microsoft, 2025a; Microsoft, 2025b), dengan tujuan memudahkan penilaian internal dari data yang dikirim oleh karyawan yang bekerja di lapangan.

Proyek pengembangan ini menjadi bagian dari digitalisasi pelaporan properti dan developer, yang sebelumnya dilakukan secara manual. Penulis bertanggung jawab dalam proses perancangan sistem, pengembangan aplikasi, dan pengujian fitur yang dikembangkan, selama masa magang berlangsung.

1.3 Ruang Lingkup

Selama enam bulan magang di Bank BTN Syariah Kantor Cabang Serang, penulis ditempatkan dalam divisi IT Support & Transaction Processing. Tugas yang diberikan yaitu mengembangkan sistem baru dengan *environment* pekerjaan yang sudah ada, yaitu menggunakan Microsoft 365 for Enterprise. Permintaan supervisor yaitu mengembangkan sistem ERP untuk pelaporan data agunan dengan integrasi produk Microsoft Dynamics 365. Ruang lingkup pekerjaan meliputi analisis kebutuhan sistem, perancangan *database* pada Dynamics 365, pengembangan interface menggunakan Power Apps, serta implementasi *workflow automation* melalui Power Automate. Sistem ini juga dilengkapi dengan fitur *approval workflow* untuk memastikan akurasi data sebelum laporan final dihasilkan.

Target pengguna sistem adalah petugas LPA (Laporan Penilaian Agunan) yang bertugas di lapangan menilai suatu agunan untuk melakukan dokumentasi dan mendata keseluruhan progres agunan tersebut, dan Financing Administration yang membutuhkan laporan data agunan secara berkala dari lapangan. Sistem dikembangkan dengan mempertimbangkan aspek keamanan data perbankan serta kepatuhan terhadap regulasi yang berlaku di industri perbankan syariah.

1.4 Tujuan

Tujuan dari penulisan tugas akhir ini yaitu mengembangan proyek aplikasi dinamakan eLPA dengan menggunakan low-code platform untuk mengotomatisasi proses pelaporan data agunan yang sebelumnya dilakukan secara manual menggunakan Excel. Sistem baru ini diharapkan dapat meningkatkan efisiensi kerja, mengurangi human error, dan mempercepat pelaporan yang dibutuhkan oleh bagian kredit lainnya.

1.5 Manfaat

Manfaat yang dapat diperoleh dari pengembangan aplikasi ini menggunakan platform low-code adalah sebagai berikut:

a. Bagi Akademik

Sebagai wadah untuk mengaplikasikan ilmu dan keterampilan yang telah dipelajari selama masa perkuliahan, terutama dalam bidang pengembangan sistem dan logika pemrograman menggunakan pendekatan *low-code*.

b. Bagi Penulis

Laporan Tugas Akhir ini dapat menjadi referensi atau wawasan tambahan dalam memahami pengembangan aplikasi berbasis *low-code*, khususnya menggunakan Microsoft Power Platform.

c. Bagi Bank BTN Syariah

Pengembangan aplikasi ini diharapkan dapat menjadi pendukung kebutuhan internal perusahaan, khususnya dalam proses pencatatan data lapangan, otomasi alur kerja, dan pengelolaan informasi secara digital yang lebih terstruktur dan efisien.

1.6 Sistematika Penulisan

Sistematika Penulisan dibuat untuk memberikan gambaran umum mengenai laporan akhir yang telah dilakukan. Sistematika Penulisannya sebagai berikut:

a. BAB 1 PENDAHULUAN

Bab ini membahas latar belakang penulisan, ruang lingkup kegiatan magang, tujuan yang ingin dicapai, manfaat yang diharapkan, serta penjelasan mengenai sistematika penulisan laporan secara keseluruhan.

b. BAB 2 LANDASAN TEORI

Bab ini memuat teori-teori dasar yang menjadi acuan dalam penyusunan Tugas Akhir. Materi dalam bab ini mencakup metode pengembangan sistem serta penjelasan mengenai platform apa saja yang digunakan dalam proyek pengembangan aplikasi yang telah dilakukan selama magang.

c. BAB 3 PELAKSANAAN MAGANG

Bab ini menjelaskan tahapan-tahapan kegiatan yang dilakukan selama proses pengembangan proyek aplikasi yang dinamakan eLPA. Penulis berperan mendesain antarmuka dan menggunakan platform Microsoft Dynamics 365 dalam pelaksanaannya.

d. BAB 4 REFLEKSI PELAKSANAAN MAGANG

   Bab ini berisi pengalaman dan pembelajaran yang diperoleh penulis selama mengikuti kegiatan magang di Bank BTN Syariah Kantor Cabang Serang, baik dari segi teknis maupun non-teknis.

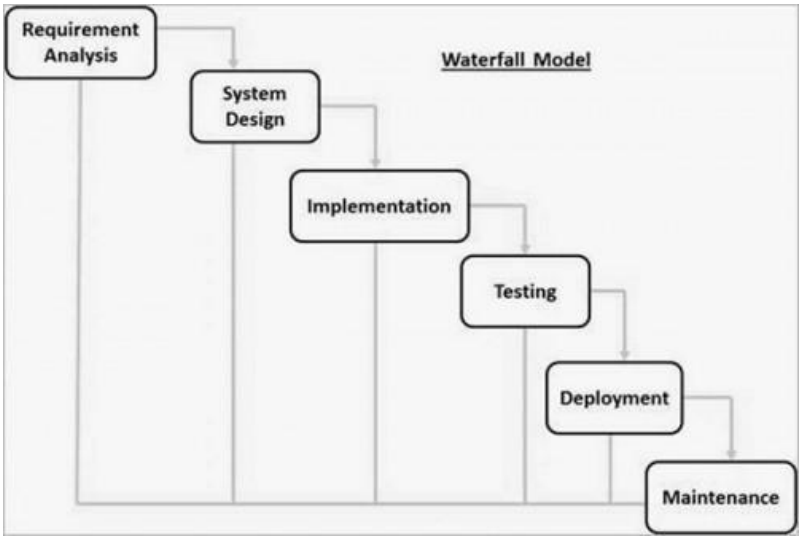
e. BAB 5 PENUTUP

   Bab terakhir ini menyampaikan kesimpulan dari hasil kegiatan magang dan penulisan laporan, serta saran yang diberikan penulis untuk pengembangan lebih lanjut di masa mendatang.

BAB II
LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1 *Software Development Life Cycle* (SDLC) model *Waterfall*

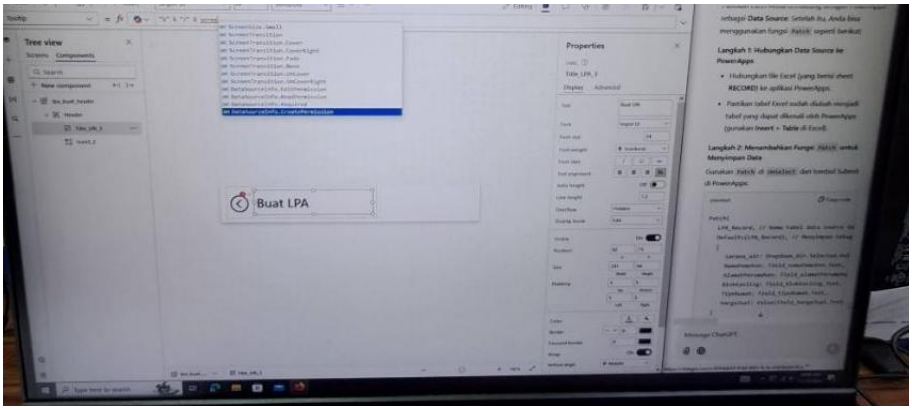
Software Development Life Cycle (SDLC) merupakan suatu kerangka kerja sistematis yang digunakan dalam proses pengembangan perangkat lunak. SDLC membagi tahapan pengembangan menjadi beberapa fase utama, mulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan (Murdiani & Sobirin, 2022). Gambar 2.1 berikut merupakan fase yang terdapat dalam SDLC model *Waterfall*.



Gambar 2.1 *Fase yang terdapat pada Software Development Life Cycle (SDLC) model Waterfall*

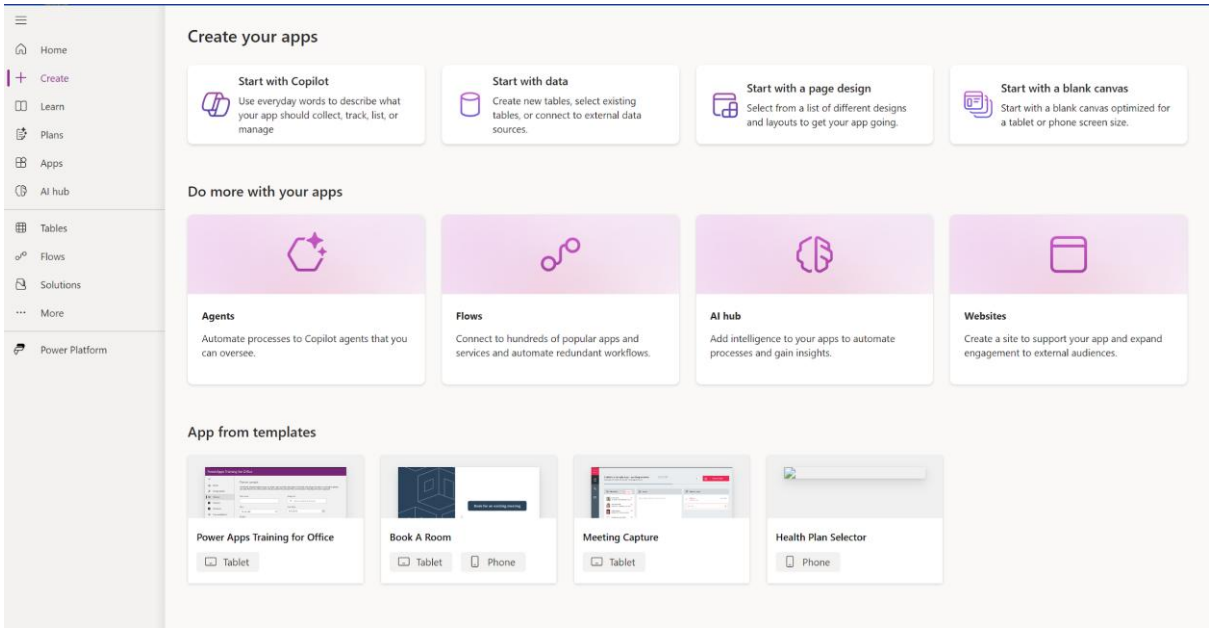
2.2 Microsoft Power Apps

Microsoft Power Apps merupakan produk yang dapat digunakan untuk penggunaan seperti ini, karena memungkinkan pengembangan aplikasi secara visual menggunakan *canvas editor*, yaitu antarmuka pengembangan berbasis *drag and drop* komponen tanpa perlu banyak *coding*, dan juga dapat melakukan publish aplikasi secara langsung (Microsoft, 2025a). Pada Gambar 2.2 berikut merupakan tampilan halaman awal Power Apps Studio yang merupakan *canvas editor* Power Apps berbasis web, untuk pengembangan keseluruhan aplikasi berbasis canvas.



Gambar 2.2 *Halaman Power Apps untuk create sebuah app*

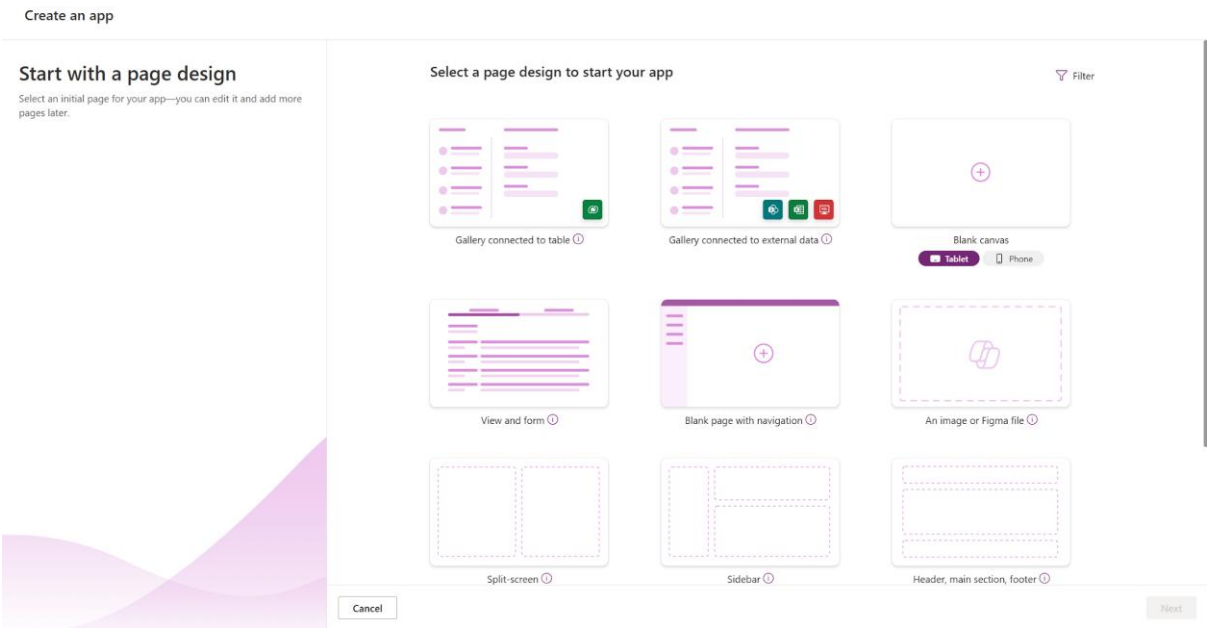
Power Apps juga mendukung Copilot AI untuk melakukan desain aplikasi, dengan hanya mendeskripsikan aplikasi apa yang ingin dibuat, nantinya akan dibuat sebuah hampir keseluruhan aplikasi secara tampilan, dan juga desain yang diinginkan sesuai pengembang. Hal ini memungkinkan pengembangan proyek dapat diselesaikan dengan waktu yang lebih ringkas. Berikut merupakan tampilan awal dari Power Apps, untuk *create* sebuah aplikasi pertama dalam bentuk canvas, pada Gambar 2.3.



Gambar 2.3 *Tampilan canvas editor Power Apps Studio*

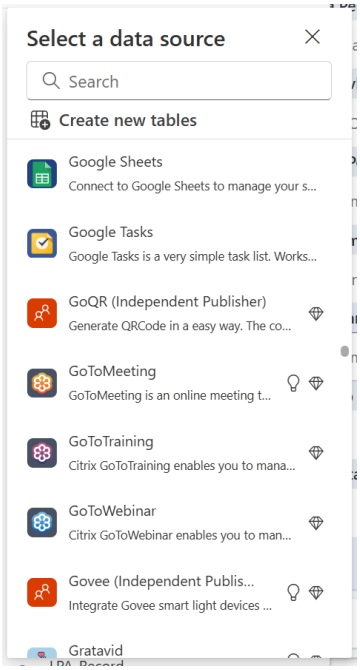
Dalam Power Apps, terdapat banyak pilihan *template* untuk desain tampilan atau *desain page*, langsung dengan data yang dapat terkoneksi langsung ke Dataverse, dan bisa terhubung ke konektor data eksternal di luar layanan Microsoft. Data eksternal ini memungkinkan koneksi ke banyak *database* dan *data source* yang langsung dapat terhubung untuk melanjutkan pengembangan siklus produk nantinya. Berikut merupakan tampilan dari halaman *Create an*

app berisikan template *canvas* yang langsung dapat dipakai untuk pengembangan (Gambar 2.4).



Gambar 2.4 *Template canvas pada Power Apps*

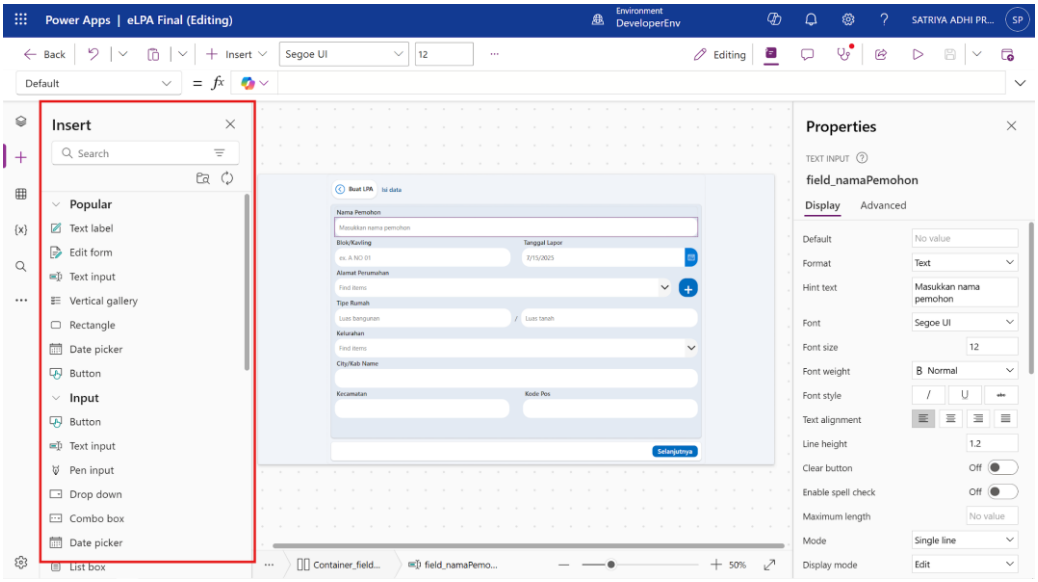
Power Apps juga memiliki kemampuan integrasi data dari sumber data maupun *database* manapun, termasuk di luar layanan Microsoft yang disebut juga dengan konektor, seperti: SQL Server, PostgreSQL, MySQL, DB2, Oracle, Azure SQL, Salesforce, Dropbox, Google Drive, Slack, Twilio, GitHub, SAP, Zendesk, dan lain sebagainya (Microsoft, 2025a). Adapun beberapa contoh yang dapat terlihat seperti pada Gambar 2.5.



Gambar 2.5 *Konektor yang terdapat dalam Power Apps*

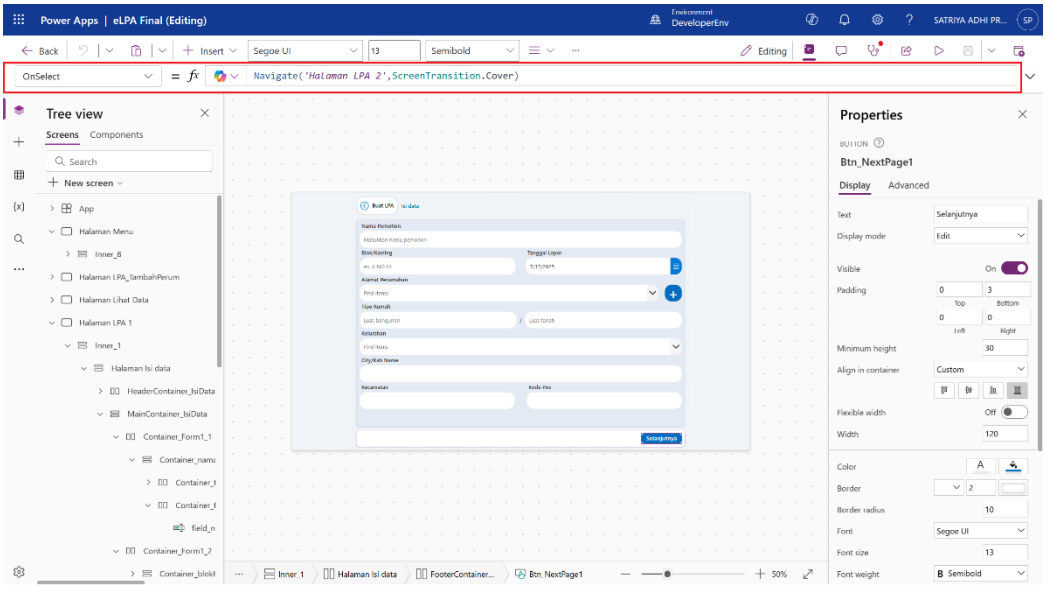
2.3 Power Apps Studio

Power Apps Studio merupakan sebutan untuk tampilan pengembangan *canvas* aplikasi dalam Power Apps atau juga disebut *canvas editor* (Microsoft, 2025b). Untuk menempatkan komponen yang disebut juga dengan *control,* dapat langsung dilakukan *drag and drop* dari bagian insert ke dalam *canvas editor,* berupa isian atau *field*, tombol, dropdown, combo box (sama seperti dropdown, perbedaannya yaitu dapat melakukan pencarian dalam kolom tersebut), dan galeri. Gambar 2.6 merupakan keseluruhan kontrol yang dapat dipakai untuk mengembangkan canvas aplikasi pada Power Apps Studio.



Gambar 2.6 *Tab insert yang berisikan kontrol pada Power Apps Studio*

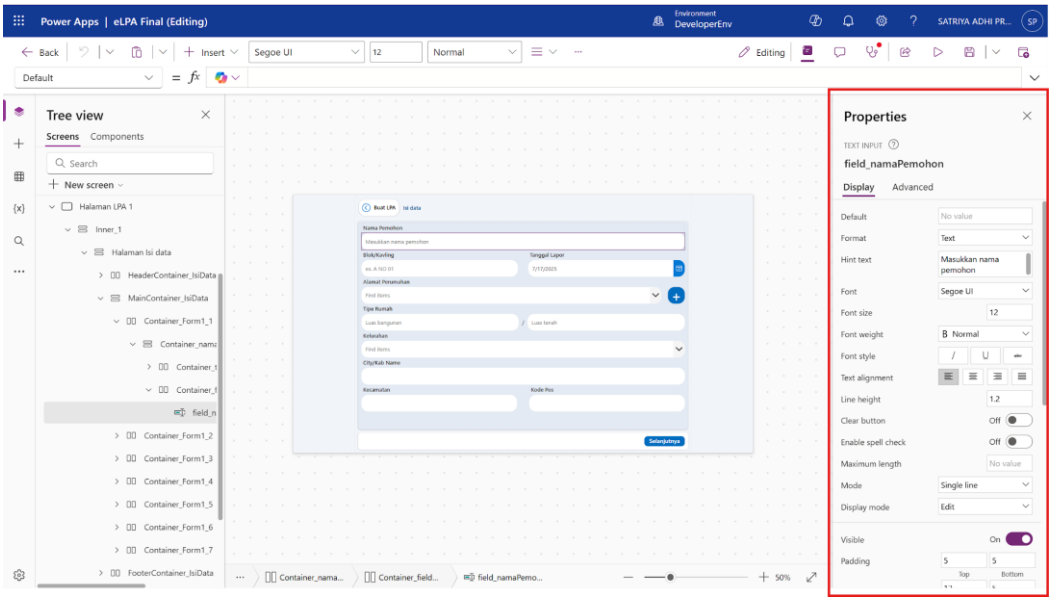
Untuk mengatur fungsi *control* seperti tombol, dropdown, atau form, digunakan *formula bar* yang menggunakan bahasa Power Fx sebagai logika pemrogramannya. Berikut merupakan gambar dari formula bar dalam Power Apps pada Gambar 2.7.



Gambar 2.7 *Formula bar Power Apps Studio*

Selain *formula bar,* Power Apps juga menyediakan panel Properties di sisi kanan tampilan. Bagian ini dapat mengedit properti dari sebuah *control* secara lebih mudah dalam bentuk visual seperti ukuran *font*, warna teks, visibilitas elemen, dan lainnya, yang merupakan tampilan singkat dari nilai-nilai properti yang sebenarnya bisa juga dikustomisasi lebih lanjut melalui formula bar. Dengan kata lain, *Properties* berfungsi sebagai bentuk *visual editor* untuk

mengatur properti dasar dari sebuah kontrol, sedangkan formula bar memberikan fleksibilitas penuh untuk mengatur fungsi dan logika kontrol tersebut secara lebih lanjut. Tampilan Properties dapat dilihat pada Gambar 2.8.

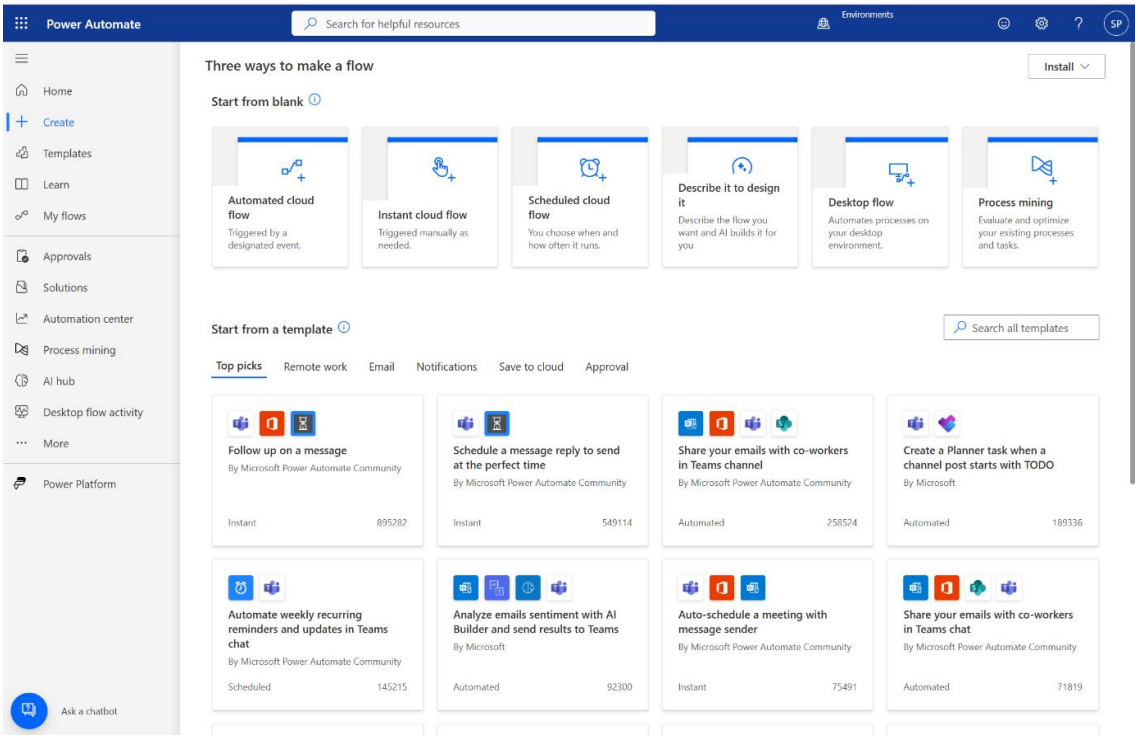


Gambar 2.8 *Antarmuka Properties pada sebuah control Power Apps*

2.4 Microsoft Power Automate

Microsoft Power Automate merupakan sistem otomatisasi untuk mengotomatisasi alur kerja dan mengelola data secara otomatis. Dengan Power Automate, pengguna aplikasi dapat melakukan berbagai tugas otomatis, seperti mengambil data secara otomatis dan integrasi antar aplikasi, yang dapat meningkatkan efisiensi operasional (Microsoft, 2025b). Dalam Power Automate, terdapat banyak pilihan *template* untuk desain *flow* yang bisa langsung dipakai.

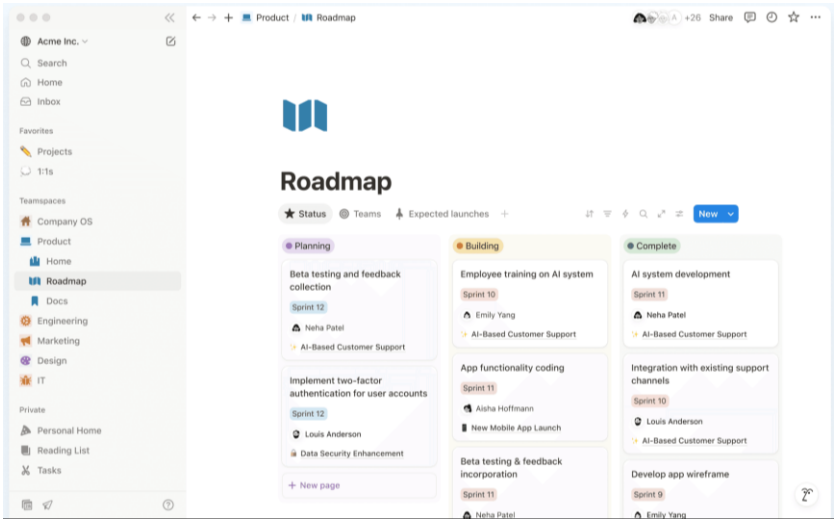
Berikut merupakan halaman awal dari Power Automate pada bagian *create,* untuk membuat flow yang sudah siap dipasang ke berbagai keperluan bisnis dalam Gambar 2.9.



Gambar 2.9 *Tampilan halaman awal Power Automate*

2.5 Notion

Platform manajemen pengetahuan dan produktivitas digital yang mendukung kolaborasi berbasis *cloud*. Notion mengintegrasikan berbagai komponen seperti catatan, basis data, kalender, manajemen tugas, serta struktur wiki dalam satu ruang kerja yang fleksibel, dengan pendekatan *no-code customization* yang memungkinkan pengguna membangun sistem kerja tanpa keterampilan pemrograman (Amt-Lab, 2021; Theo, 2022; Zhang et al., 2021). Berikut merupakan halaman awal dari Notion untuk



Gambar 2.10 *Tampilan utama Notion*

2.6 Tinjauan Pustaka

Microsoft Power Platform sebagai *low-code* platform agar memudahkan proses pelaporan dan pengelolaan data agunan di Bank BTN Syariah Kantor Cabang Serang dapat berjalan dengan lancar dan optimal. Selain itu, peneliti juga menggali informasi dari buku dan jurnal penelitian untuk memperoleh informasi teori yang telah ada sebelumnya terkait dengan judul untuk memperoleh tinjauan pustaka yang terkait dengan pengembangan sistem pelaporan data agunan yang menggunakan integrasi Microsoft Power Platform dengan Microsoft Dynamics 365.

Tabel 2.1 Tinjauan Pustaka Implementasi ERP berbasis LCDP

Topik	**Penelitian Sebelumnya / Referensi**	**Temuan Utama**	**Relevansi dengan Penelitian**
SDLC dengan Waterfall	Microsoft (2024); Nugraha & Prabowo (2023)	Waterfall cocok untuk sistem prosedural dengan dokumentasi rapi. Masih relevan untuk pengembangan *low-code* dengan tim lintas fungsi.	Model SDLC ini digunakan dalam pengembangan sistem pelaporan berbasis LCDP yang prosesnya terstruktur dan terdokumentasi.
Low-Code Development Platform (LCDP)	Forrester (2023); Gartner (2024); Kakolu & Faheem (2023)	LCDP memungkinkan pengembangan aplikasi tanpa banyak coding. Efisiensi meningkat hingga 45%, mempercepat digitalisasi dan otomatisasi proses bisnis.	LCDP menjadi solusi digitalisasi pelaporan agunan yang sebelumnya berbasis Excel manual, mengurangi ketergantungan pada *developer*.

Otomatisasi Proses Pelaporan	Razak dkk. (2024); Putri & Munawaroh (2025)	Proses manual rawan kesalahan, keterlambatan, dan tidak efisien. SDM menjadi tantangan dalam adopsi sistem digital.	Sistem berbasis Power Automate membantu otomatisasi laporan agunan dari lapangan, mempercepat input data dan distribusi informasi secara real-time.
Penerapan LCDP di Sektor Keuangan	Kakolu & Faheem (2023); studi Gartner (2024)	LCDP berhasil digunakan di industri keuangan untuk menyederhanakan sistem ERP mini. Aksesibilitas dan fleksibilitas tinggi.	Pengembangan sistem berbasis Microsoft Power Platform sesuai dengan kebutuhan pelaporan agunan di sektor pembiayaan perumahan.

Berikut rangkuman dari beberapa penelitian diatas yang dijadikan sebagai tinjauan pustaka ini, yaitu:

2.6.1 Model SDLC dengan *Waterfall*

SDLC dengan *Waterfall* merupakan pendekatan klasik dalam pengembangan perangkat lunak yang berurutan dan sistematis. Prosesnya dimulai dari tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan. Setiap tahapan harus diselesaikan terlebih dahulu sebelum berlanjut ke tahap berikutnya (Microsoft, 2024). Model ini memiliki keunggulan dalam hal struktur dokumentasi yang rapi, kontrol versi yang kuat, serta kemudahan dalam audit proyek, cocok untuk proyek-proyek dengan kebutuhan yang stabil dan prosedural, seperti sistem pelaporan yang diterapkan di sektor keuangan atau pembiayaan.

Dalam pengembangan sistem berbasis *low-code*, pendekatan *Waterfall* dapat diterapkan untuk memberikan struktur pengembangan yang terarah, terutama saat sistem dibangun oleh tim yang tidak semuanya memiliki latar belakang teknis mendalam. Penelitian oleh Nugraha & Prabowo (2023) juga menunjukkan bahwa SDLC *Waterfall* masih relevan dalam proyek pengembangan *low-code* pada sektor industri manufaktur, selama tahapan validasi dan dokumentasi dilakukan dengan baik.

2.6.2 *Low-Code Development Platform* (LCDP)

Low-Code Development Platform (LCDP) adalah pendekatan pengembangan perangkat lunak yang memungkinkan pengguna membangun aplikasi melalui antarmuka visual dengan sedikit penulisan kode (minimal coding). Platform ini mendukung pengembangan aplikasi dengan cepat, efisien, dan dapat dilakukan oleh pengguna non-programmer sekalipun (Forrester, 2023). Beberapa platform populer seperti Microsoft Power Platform, Outsystems, Mendix, dan AppSheet menyediakan fitur *drag-and-drop* UI, integrasi data otomatis, serta workflow automation. Menurut riset Gartner (2024), lebih dari 65% pengembangan aplikasi perusahaan pada tahun 2025 diperkirakan akan menggunakan platform *low-code* atau *no-code*.

LCDP terbukti mempercepat digitalisasi di sektor keuangan, terutama untuk solusi pelaporan dan otomasi proses berbasis Excel. Studi oleh Kakolu & Faheem (2023) menyebutkan bahwa LCDP dapat meningkatkan efisiensi operasional hingga 45% dengan mengurangi waktu pengembangan dan beban pemeliharaan sistem tradisional. Dalam kasus implementasi pada perbankan, LCDP memberikan solusi yang relevan karena mengurangi ketergantungan terhadap developer, membuka peluang partisipasi tim operasional dalam pengembangan aplikasi, serta mampu beradaptasi dengan perubahan kebutuhan bisnis secara cepat.

2.6.3 Digitalisasi pelaporan Excel menggunakan LCDP

Digitalisasi proses pelaporan, khususnya yang sebelumnya berbasis Excel manual, menjadi kebutuhan mendesak dalam industri pembiayaan dan perbankan. Proses manual rawan terhadap human error, keterlambatan distribusi informasi, serta kesulitan dalam konsolidasi data (Razak dkk., 2024). Dalam konteks pelaporan agunan, penggunaan LCDP untuk merancang sistem pelaporan digital seperti aplikasi eLPA berbasis SharePoint, Power Apps, dan Power Automate, telah menunjukkan peningkatan efisiensi yang signifikan.

Selain meningkatkan kecepatan dan akurasi, sistem digital juga memberikan kemampuan pelacakan histori, otomatisasi pengiriman laporan, serta peningkatan transparansi antar divisi. Namun, Putri & Munawaroh (2025) mencatat bahwa transformasi digital juga memerlukan kesiapan SDM, terutama dalam pelatihan dan adopsi teknologi baru.

BAB III
PELAKSANAAN MAGANG

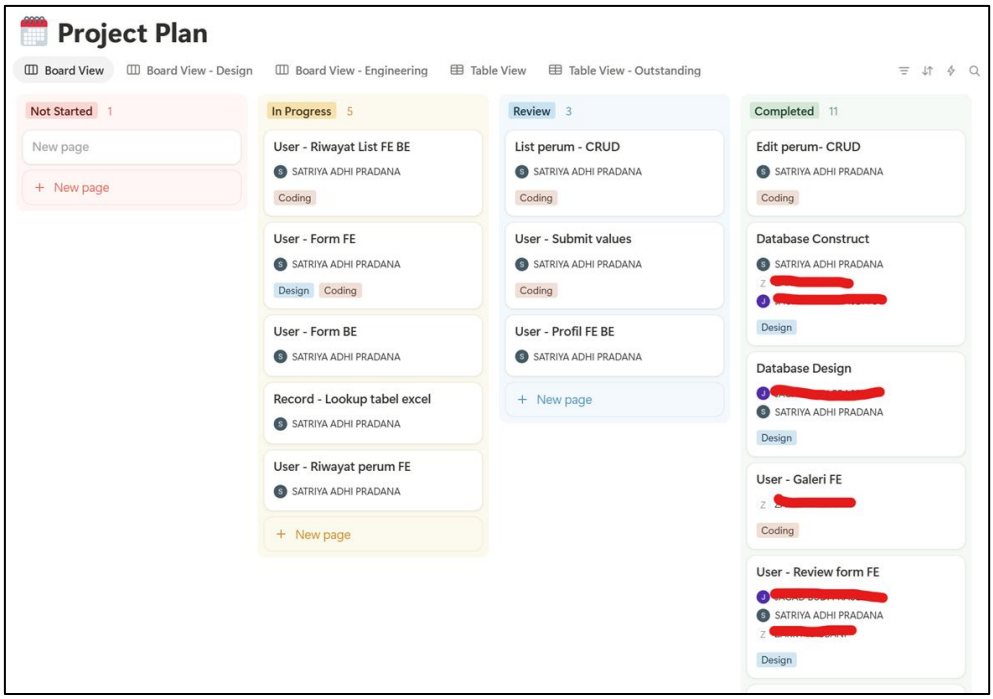
3.1 Manajemen Proyek

Dalam melaksanakan magang selama enam bulan di Bank BTN Syariah KC Serang, berbagai macam pelaksanaan yang telah dilakukan oleh penulis dalam mengerjakan proyek. Berikut merupakan ringkasan aktivitas magang yang dilakukan dari Oktober 2024 hingga Maret 2025 pada tabel berikut:

Tabel 3.1 Aktivitas Magang

No	Aktivitas	Waktu Pelaksanaan	Durasi
1	Perkenalan dan Persiapan	Oktober 2024	1 minggu
2	Analisis kebutuhan	Oktober 2024	1 minggu
3	Pengembangan aplikasi *eLPA*	Oktober 2024–Januari 2025	2 bulan
4	Perancangan dan implementasi	Februari 2025	2 minggu
5	Pengujian/Testing	Februari 2025	1 minggu
6	Penutupan proyek	Februari–Maret 2025	1 minggu

Berikut merupakan dokumentasi pelaksanaan magang beserta pembagian tugas menggunakan Notion sebagai alat untuk *planning* selama keseluruhan *timeline* proyek pada Gambar 3.1.



Gambar 3.1 *Project plan di Notion*

3.2 Penerapan Metode *Software Development Life Cycle (SDLC) model Waterfall*

Pengembangan aplikasi ERP internal untuk pelaporan perumahan KPR dilakukan dengan menggunakan pengembangan SDLC model *Waterfall*. Model ini dipilih karena kebutuhan sistem sudah jelas sejak awal, sehingga setiap tahap dapat diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya. Tahapan yang dilalui meliputi analisis kebutuhan, perancangan sistem, implementasi aplikasi, pengujian, hingga penutupan. Berikut merupakan uraian dari pekerjaan yang dilakukan dalam pelaksanaan magang:

3.3 Analisis Kebutuhan

Tahap ini dimulai dari melakukan analisa keperluan dari target pengguna, termasuk koordinasi antara pengembang, petugas lapangan, dan penilai internal (*Financing Administration).* Analisa ini termasuk melakukan survei kebutuhan isian form yang akan digunakan oleh pengguna utama bagian non-teknisnya yaitu petugas lapangan, dalam kegiatannya yaitu menentukan fitur apa saja yang akan dipakai dalam keseluruhan aplikasi, berdasarkan dari proses laporan manual sebelumnya yang masih menggunakan *WhatsApp* sebagai media mengirim dokumentasi agunan dan proses bangunan yang sedang berjalan. Selain itu, penilai internal sebagai *supervisor* dilakukan survei untuk mengetahui bagian fitur mana saja yang harus disesuaikan dengan pekerjaan dari lapangan ke penilai internal.

Metode	Wawancara dan survei langsung
Waktu Pelaksanaan	8 Oktober – 9 Oktober 2024
Responden/ Narasumber	Petugas lapangan (pengguna utama aplikasi) dan Financing Administration (memeriksa laporan dari petugas lapangan)
Tujuan	Mengembangkan sistem pelaporan yang terotomatisasi menggunakan aplikasi, yang bisa terintegrasi dengan produk Microsoft 365 milik Bank BTN.
Pertanyaan Utama	1. Selama ini bagaimana sistem kerja pelaporan agunan yang telah berjalan? 2. Kendala apa yang sering dialami? 3. Data apa saja yang perlu disimpan? 4. Kalau misal ada bagian laporan yang tidak sesuai isiannya, seperti keliru dalam memasukkan data, nomor register yang tidak sesuai, dan terdapat kesalahan seperti foto yang kurang, apa tindakan lanjutnya? 5. *File* laporan ditaruh dimana? Apakah biasanya menunggu data datang terlebih dahulu dari lapangan?
Jawaban Wawancara	1. Pelaporan agunan masih dilakukan secara manual menggunakan Microsoft Excel, di mana petugas lapangan menyerahkan data fisik dan dokumentasi agunan kepada bagian *Financing Administration*. 2. Kendala utama adalah potensi kesalahan *input* karena pekerjaan bersifat repetitif, lamanya waktu rekapitulasi ketika volume laporan tinggi, serta kesulitan distribusi data secara cepat. Selain itu, koordinasi antara petugas lapangan dan penilai internal sering terhambat jika ada data yang kurang lengkap. 3. Data yang disimpan meliputi identitas nasabah, alamat agunan, data developer, nilai agunan, jenis agunan, dan dokumentasi spesifikasi agunan, seperti: tampak depan bangunan, jalan perumahan, listrik, air, pondasi, dan plat KPR. 4. Tindakan lanjut biasanya dilakukan dengan melakukan konfirmasi ulang kepada petugas lapangan untuk melengkapi atau memperbaiki data yang keliru, namun biasanya bisa membutuhkan waktu yang lebih jika kesalahan data sebelumnya sudah terdapat pada entri *register*. 5. *File* ditaruh dalam komputer kantor dalam *local network*, yang bisa saling akses antar komputer lainnya. Untuk data agunan biasanya dari lapangan melalui grup WhatsApp, yang langsung dikirim saat survei di suatu perumahan itu selesai.
Hasil	Terdapat kebutuhan untuk aplikasi pelaporan untuk mempermudah pekerjaan petugas lapangan dan yang mampu sinkronisasi dengan OneDrive, dan memiliki opsi ekspor laporan ke OneDrive.

Pada pertemuan perdana, aktivitas dimulai dengan perkenalan lingkungan pekerjaan, tools yang dipakai untuk kebutuhan magang, dan metode pengembangan yang dipakai selama magang. Selama magang, penulis diberikan pekerjaan mendesain antarmuka keseluruhan proyek aplikasi Microsoft Power Apps. Sedangkan untuk bagian otomatisasi menggunakan Power Automate, bagian tersebut dipegang oleh tim IT Support yang sudah menghubungkan backend ke dalam *environment* internal.

3.4 Pengembangan Desain

Pada tahap ini, penulis mulai mengembangkan aplikasi bersama dengan tim, berdasarkan dari hasil keseluruhan analisa kebutuhan yang sudah dilakukan pada tahap sebelumnya. Pada pengembangannya dilakukan secara bersama dengan tim IT Support yang berjumlah dua orang. Proyek yang dikembangkan dinamakan "eLPA" ini bertujuan untuk menggantikan proses manual berbasis Excel menjadi sistem pelaporan agunan yang lebih terotomatisasi, terintegrasi, dan mudah diakses baik oleh petugas lapangan maupun pihak internal.

Penulis bekerja sama dengan *supervisor* baik dalam mendesain antarmuka aplikasi, pembuatan *workflow,* membuat tabel record dalam Excel, hingga membuat struktur data keseluruhan aplikasi. Keseluruhan pengembangan ini dilakukan dengan memanfaatkan Microsoft Dynamics 365, yang menggunakan Power Apps untuk keseluruhan antarmuka aplikasi, SharePoint untuk penyimpanan data, Power Automate untuk mengotomatisasi alur kerja, dan OneDrive untuk menyimpan hasil akhir file Excel.

Aplikasi yang dikembangkan ini nantinya digunakan oleh petugas lapangan yang bertugas mendata agunan, sehingga hanya membutuhkan satu role saja untuk aplikasi berbentuk form ini. Di sisi lain, pengembang membutuhkan proses pengembangan dan *deploy* aplikasi yang cepat, stabil, dan juga minim error selama pengembangan dan implementasi nantinya.

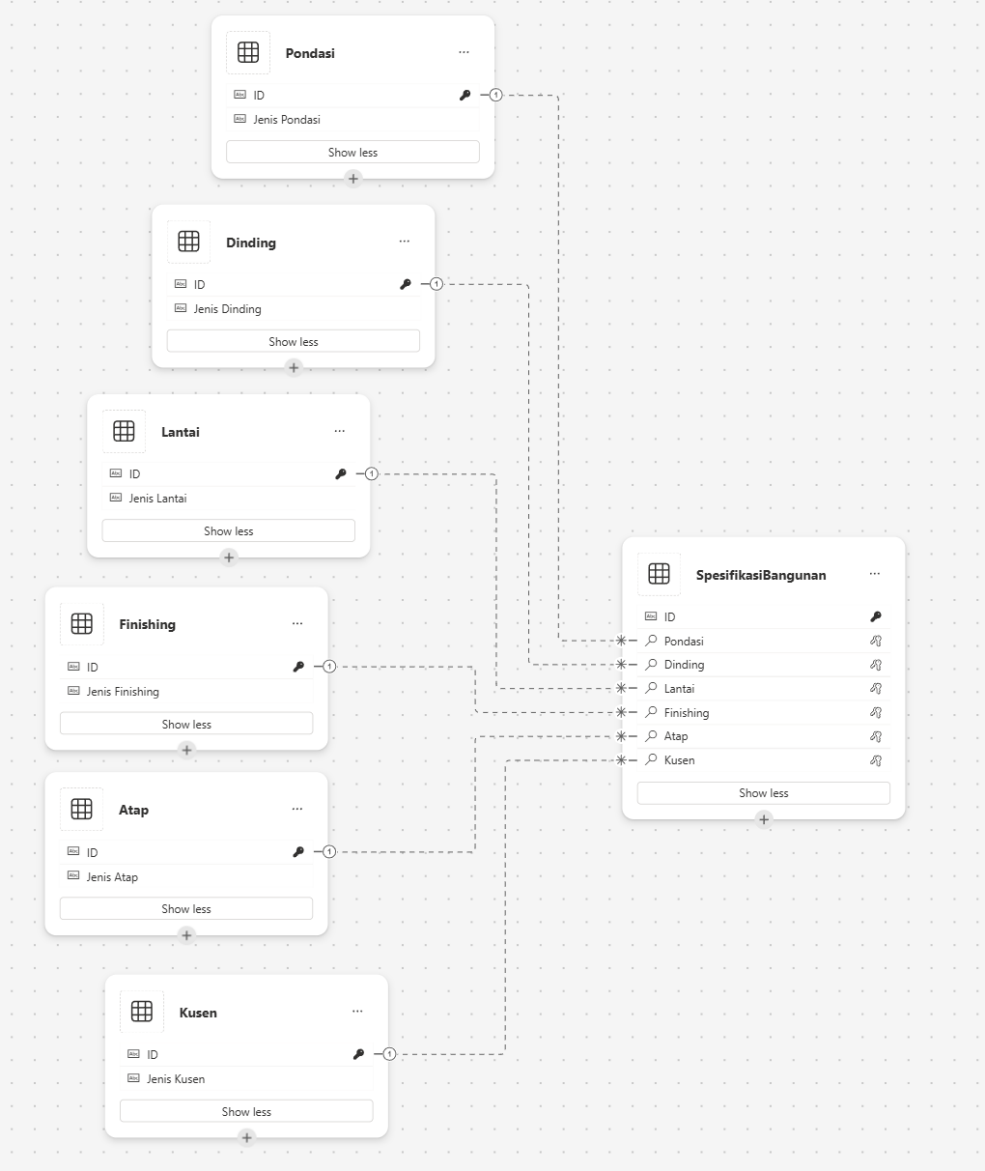
3.4.1 *Database* dan *data source* Microsoft Power Apps

eLPA menggunakan konektor Excel dan Dataverse sebagai *data source* statis untuk *value* opsi yang terdapat dalam *dropdown* dan *combo box*, dan SharePoint Lists sebagai database untuk menyimpan keseluruhan data record aplikasi, yang nantinya juga akan digunakan oleh Power Automate untuk sebagai salah satu *flow trigger.* Pada akhirnya, data record dari SharePoint Lists akan ditaruh dalam file Excel yang sudah bisa digunakan untuk pelaporan ke

pihak internal, dimana file tersebut terletak dalam OneDrive yang merupakan hasil otomatisasi Power Automate.

3.4.2 *Data Source* menggunakan Dataverse

Dataverse menggunakan ERD *(Entity Relationship Diagram)* untuk menggabungkan entitas antar tabel, sehingga memungkinkan integrasi data secara relasional. Dengan adanya hubungan ini, setiap tabel dapat saling terhubung melalui kolom referensi (*lookup*), yang menjadikan *data source* lebih terstruktur dan statis, serta memudahkan dalam pengambilan dan pemrosesan data secara konsisten dalam aplikasi Power Apps. Setiap tabel memiliki kolom ID sebagai *primary key* yang berfungsi sebagai identitas unik dari setiap entitas. *Primary key* ini digunakan untuk memastikan bahwa setiap data dapat dibedakan satu sama lain secara spesifik. Sementara itu, tabel utama bernama *SpesifikasiBangunan* memiliki beberapa kolom yang berperan sebagai *foreign key*, seperti Pondasi, Dinding, Lantai, *Finishing*, Atap, dan Kusen, yang masing-masing terhubung ke tabel referensi yang relevan. Relasi antar tabel ini dibentuk melalui mekanisme *lookup* yang merepresentasikan hubungan *many-to-one*, di mana banyak entri dalam *SpesifikasiBangunan* dapat merujuk ke satu jenis nilai pada tabel komponen. Berikut merupakan data yang telah dibuat dalam Dataverse, pada Gambar 3.9.



Gambar 3.2 *Data source statis SpesifikasiBangunan menggunakan Dataverse*

Berikut merupakan isi data dari setiap tabel pada *data source* Dataverse untuk SpesifikasiBangunan yang digunakan untuk *dropdown* dalam form aplikasi, pada Tabel 3.2.

Tabel 3.2 Isian data source tabel Dataverse

Pondasi

ID *	Jenis Pondasi
1	Beton Bertulang
2	Batu Kali
3	Kayu

a) Isi data tabel pondasi

Dinding

ID *	Jenis Dinding
1	Bata Merah Plester
2	Batako
3	Bata Ringan
4	Kayu

b) Isi data tabel dinding

Lantai

ID *	Jenis Lantai
1	Keramik
2	Marmer
3	Teraso
4	Ubin PC
5	Ubin Rusak

c) Isi data tabel lantai

Finishing

ID *	Jenis Finishing
1	Cat Tembok Halus
2	Cat Tembok Sedang
3	Cat Tembok Kasar
4	Kapur
5	Tanpa Finishing

d) Isi data tabel finishing

Atap

ID *	Jenis Atap
1	Genteng Keramik
2	Genteng Beton
3	Sirap
4	Genteng Tanah
5	Sakura Roof
6	Asbes

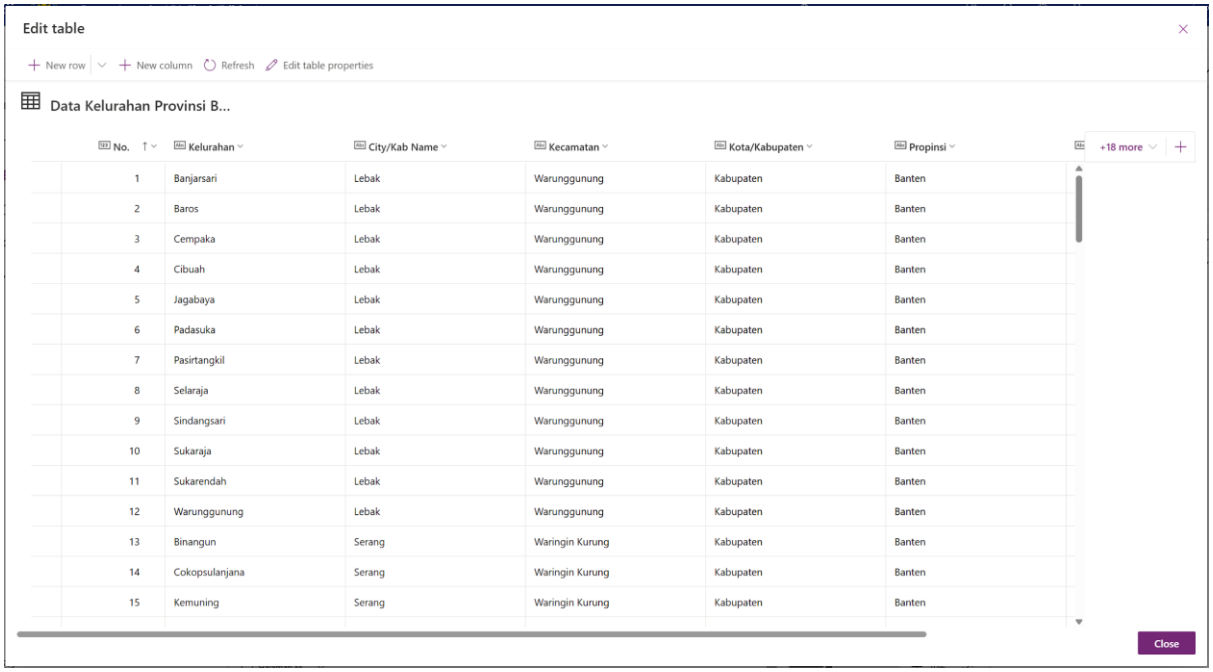
e) Isi data tabel atap

Kusen

ID *	Jenis Kusen
1	Plitur Halus
2	Plitur Kasar
3	Cat Halus
4	Cat Kasar
5	Cat Sedang

f) Isi data tabel kusen

Data source untuk satu kelurahan Provinsi Banten juga menggunakan Dataverse, dengan entitas relasi tabel yang berbeda dengan SpesifikasiBangunan tadi. Berikut merupakan isian dari Dataverse kelurahan satu Provinsi Banten pada Gambar 3.3.



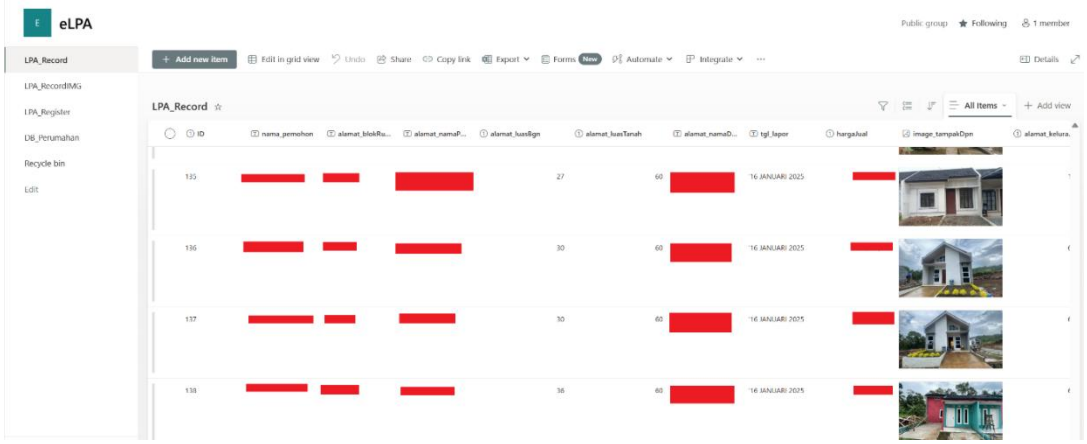
No.	Kelurahan	City/Kab Name	Kecamatan	Kota/Kabupaten	Propinsi
1	Banjarsari	Lebak	Warunggunung	Kabupaten	Banten
2	Baros	Lebak	Warunggunung	Kabupaten	Banten
3	Cempaka	Lebak	Warunggunung	Kabupaten	Banten
4	Cibuah	Lebak	Warunggunung	Kabupaten	Banten
5	Jagabaya	Lebak	Warunggunung	Kabupaten	Banten
6	Padasuka	Lebak	Warunggunung	Kabupaten	Banten
7	Pasirtangkil	Lebak	Warunggunung	Kabupaten	Banten
8	Selaraja	Lebak	Warunggunung	Kabupaten	Banten
9	Sindangsari	Lebak	Warunggunung	Kabupaten	Banten
10	Sukaraja	Lebak	Warunggunung	Kabupaten	Banten
11	Sukarendah	Lebak	Warunggunung	Kabupaten	Banten
12	Warunggunung	Lebak	Warunggunung	Kabupaten	Banten
13	Binangun	Serang	Waringin Kurung	Kabupaten	Banten
14	Cokopsulanjana	Serang	Waringin Kurung	Kabupaten	Banten
15	Kemuning	Serang	Waringin Kurung	Kabupaten	Banten

Gambar 3.3 *Data source statis kelurahan satu Provinsi Banten menggunakan Dataverse*

3.4.3 *Database record* menggunakan Microsoft SharePoint Lists

SharePoint Lists yang digunakan sebagai database untuk menyimpan seluruh record data dari aplikasi Power Apps disusun dalam bentuk kolom-kolom list, serupa dengan struktur kolom pada tabel Excel. List tersebut berada dalam situs SharePoint bernama “eLPA”, sedangkan nama list yang digunakan untuk menyimpan data utama adalah “LPA_Record”.

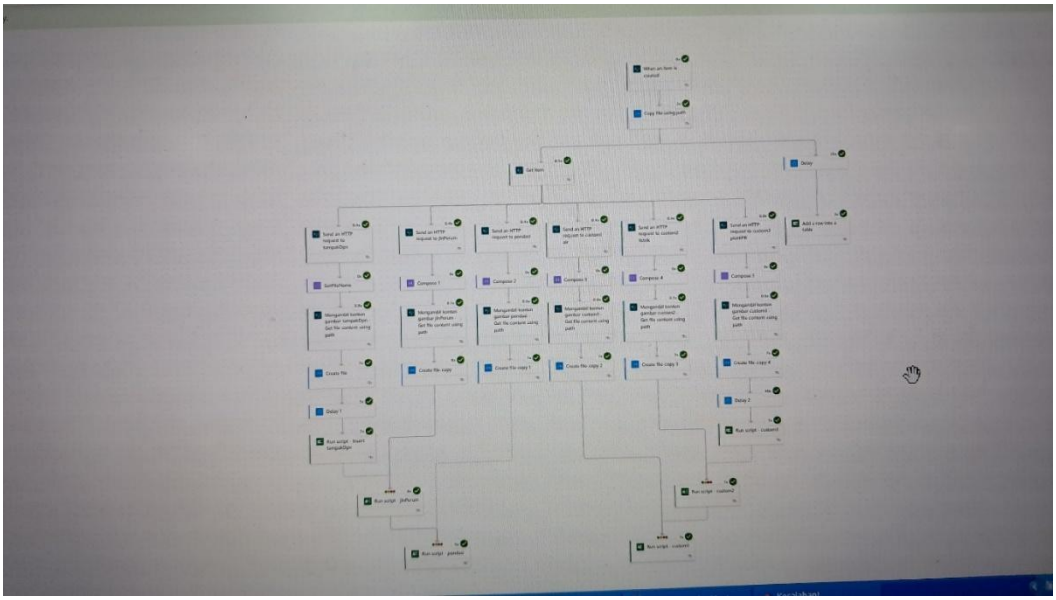
Setiap kali data record baru dimasukkan, SharePoint Lists secara otomatis menghasilkan nilai ID yang bersifat *auto-increment* pada sebuah entri list tersebut. ID yang dihasilkan ini dijadikan sebuah nomor untuk mengetahui penomoran entri yang dipakai dalam aplikasi dan database nantinya, memudahkan pengurutan data dan memungkinkan penelusuran terhadap entri tertentu, seperti mengetahui urutan rumah keberapa yang telah dicatat, serta dapat dimanfaatkan sebagai dasar penomoran untuk digunakan dalam pihak internal nantinya. Berikut merupakan Lists yang dibuat untuk *database record* pada Gambar 3.4.



Gambar 3.4 *Kolom database SharePoint Lists*

3.4.4 Otomatisasi *flow* menggunakan Microsoft Power Automate

Dalam Power Automate, IT Support merancang *flow* dari trigger awal milik SharePoint "*When an item is created*" hingga *flow* berakhir ketika data sudah masuk ke dalam Excel. Pada Gambar 3.5 merupakan *flow* keseluruhan untuk aplikasi Power Apps dari Power Automate.



Gambar 3.5 *Tampilan flow Power Automate*

3.4.5 Hasil Otomatisasi File Excel

Hasil akhir atau *output* dari otomatisasi *data record* tadi menggunakan Power Automate dihasilkan dalam file Excel yang digunakan untuk penilaian tim internal. File Excel ini merupakan file yang sudah terdapat sebelumnya untuk penggunaan pelaporan (*template* laporan), yang terdapat dua buah *sheet* yang masing-masing dinamakan "LPA" dan

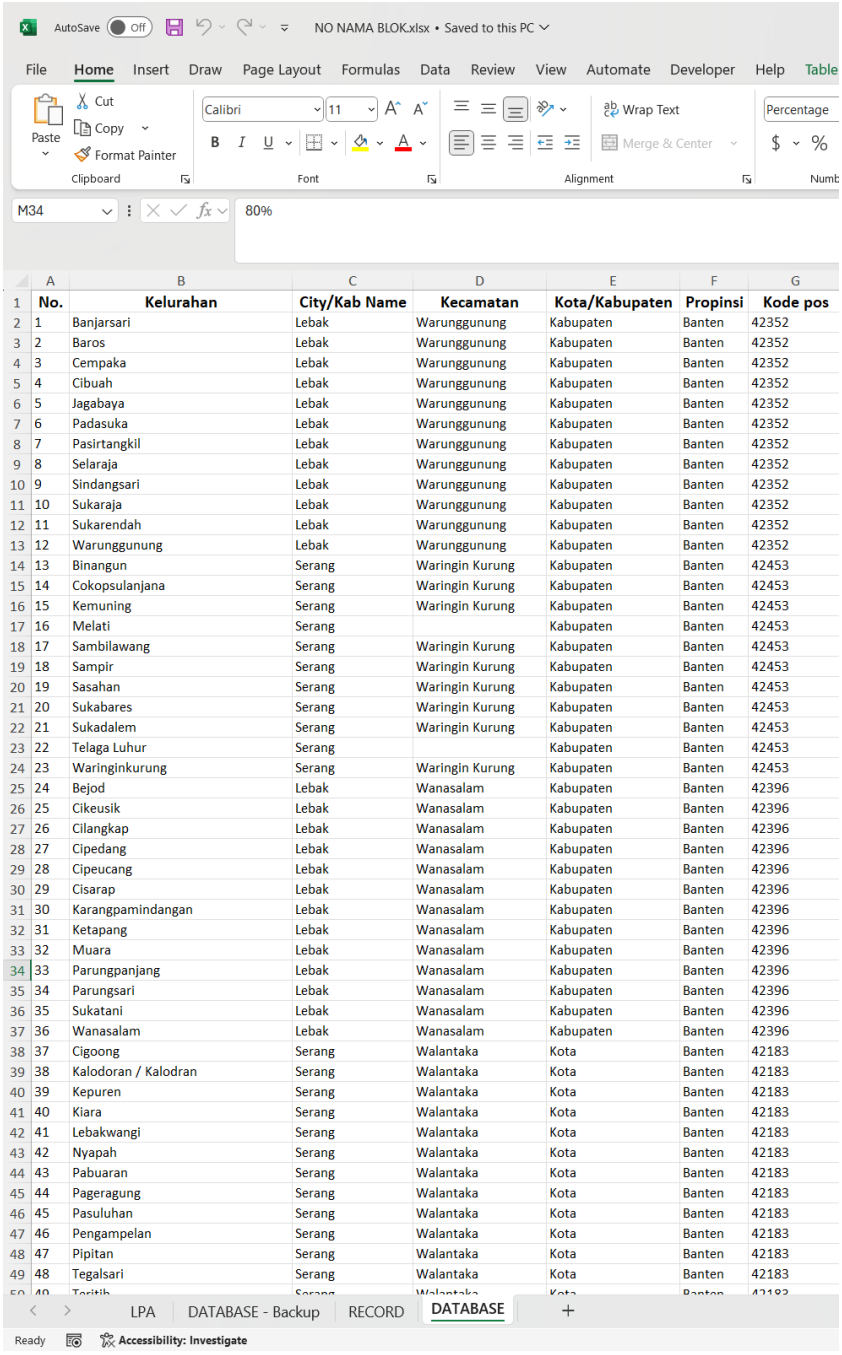
"DATABASE". Dari sheet "LPA" terdapat sebuah cell yang menunjukkan data dari agunan tersebut, seperti: kelurahan, data dalam Data Bangunan, Spesifikasi Bangunan, Sarana Bangunan, dan beberapa dari Data Tanah, serta catatan untuk menunjukkan beberapa status sarana agunan tersebut, seperti: listrik, air, jalan perumahan, dan plat KPR. Berikut pada Gambar 3.6 merupakan tampilan dari *sheet* "LPA".



Gambar 3.6 *Tampilan sheet "LPA"*

Selanjutnya pada *sheet* "DATABASE" berisikan data source statis yang terdapat dalam satu file Excel itu, sebagai sekumpulan value untuk dilakukan *lookup* ke dalam cell yang

terdapat dalam sheet “LPA”. Data source mencakup hal berikut: hasil ekspor file .csv data kelurahan satu Provinsi Banten dan data agunan yang dipakai untuk LPA (data dalam Data Bangunan, Spesifikasi Bangunan, Sarana Bangunan, dan beberapa dari Data Tanah). Berikut merupakan tampilan dari sheet “DATABASE” berisi data kelurahan satu Provinsi Banten pada Gambar 3.7.

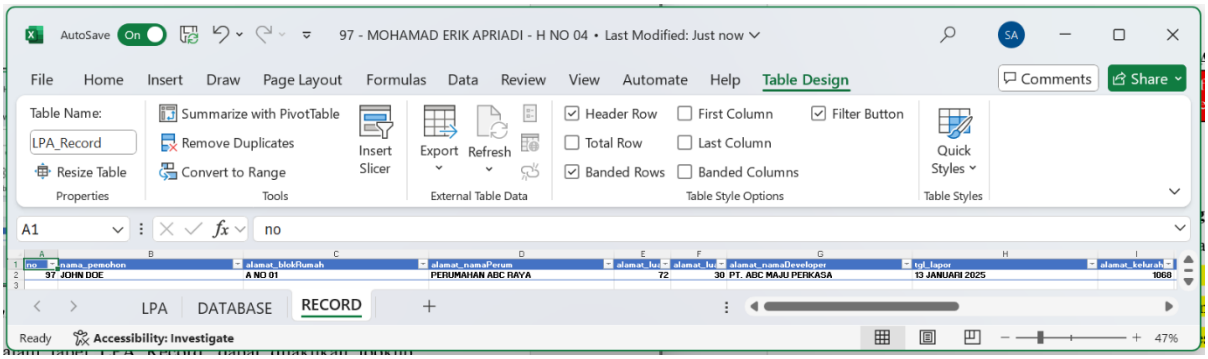


No.	Kelurahan	City/Kab Name	Kecamatan	Kota/Kabupaten	Propinsi	Kode pos
1	Banjarsari	Lebak	Warunggunung	Kabupaten	Banten	42352
2	Baros	Lebak	Warunggunung	Kabupaten	Banten	42352
3	Cempaka	Lebak	Warunggunung	Kabupaten	Banten	42352
4	Cibuah	Lebak	Warunggunung	Kabupaten	Banten	42352
5	Jagabaya	Lebak	Warunggunung	Kabupaten	Banten	42352
6	Padasuka	Lebak	Warunggunung	Kabupaten	Banten	42352
7	Pasirtangkil	Lebak	Warunggunung	Kabupaten	Banten	42352
8	Selaraja	Lebak	Warunggunung	Kabupaten	Banten	42352
9	Sindangsari	Lebak	Warunggunung	Kabupaten	Banten	42352
10	Sukaraja	Lebak	Warunggunung	Kabupaten	Banten	42352
11	Sukarendah	Lebak	Warunggunung	Kabupaten	Banten	42352
12	Warunggunung	Lebak	Warunggunung	Kabupaten	Banten	42352
13	Binangun	Serang	Waringin Kurung	Kabupaten	Banten	42453
14	Cokopsulanjana	Serang	Waringin Kurung	Kabupaten	Banten	42453
15	Kemuning	Serang	Waringin Kurung	Kabupaten	Banten	42453
16	Melati	Serang		Kabupaten	Banten	42453
17	Sambilawang	Serang	Waringin Kurung	Kabupaten	Banten	42453
18	Sampir	Serang	Waringin Kurung	Kabupaten	Banten	42453
19	Sasahan	Serang	Waringin Kurung	Kabupaten	Banten	42453
20	Sukabares	Serang	Waringin Kurung	Kabupaten	Banten	42453
21	Sukadalem	Serang	Waringin Kurung	Kabupaten	Banten	42453
22	Telaga Luhur	Serang		Kabupaten	Banten	42453
23	Waringinkurung	Serang	Waringin Kurung	Kabupaten	Banten	42453
24	Bejod	Lebak	Wanasalam	Kabupaten	Banten	42396
25	Cikeusik	Lebak	Wanasalam	Kabupaten	Banten	42396
26	Cilangkap	Lebak	Wanasalam	Kabupaten	Banten	42396
27	Cipedang	Lebak	Wanasalam	Kabupaten	Banten	42396
28	Cipeucang	Lebak	Wanasalam	Kabupaten	Banten	42396
29	Cisarap	Lebak	Wanasalam	Kabupaten	Banten	42396
30	Karangpamindangan	Lebak	Wanasalam	Kabupaten	Banten	42396
31	Ketapang	Lebak	Wanasalam	Kabupaten	Banten	42396
32	Muara	Lebak	Wanasalam	Kabupaten	Banten	42396
33	Parungpanjang	Lebak	Wanasalam	Kabupaten	Banten	42396
34	Parungsari	Lebak	Wanasalam	Kabupaten	Banten	42396
35	Sukatani	Lebak	Wanasalam	Kabupaten	Banten	42396
36	Wanasalam	Lebak	Wanasalam	Kabupaten	Banten	42396
37	Cigoong	Serang	Walantaka	Kota	Banten	42183
38	Kalodoran / Kalodran	Serang	Walantaka	Kota	Banten	42183
39	Kepuren	Serang	Walantaka	Kota	Banten	42183
40	Kiara	Serang	Walantaka	Kota	Banten	42183
41	Lebakwangi	Serang	Walantaka	Kota	Banten	42183
42	Nyapah	Serang	Walantaka	Kota	Banten	42183
43	Pabuaran	Serang	Walantaka	Kota	Banten	42183
44	Pageragung	Serang	Walantaka	Kota	Banten	42183
45	Pasuluhan	Serang	Walantaka	Kota	Banten	42183
46	Pengampelan	Serang	Walantaka	Kota	Banten	42183
47	Pipitan	Serang	Walantaka	Kota	Banten	42183
48	Tegalsari	Serang	Walantaka	Kota	Banten	42183

Gambar 3.7 *Data kelurahan satu Provinsi Banten dalam sheet “DATABASE”*

Di sini penulis membuat sheet baru “RECORD” untuk menaruh data *record* dari hasil otomatisasi tadi yang akan ditaruh dalam tabel dinamakan “LPA_Record”. Berikut merupakan

beberapa kolom tabel dari tabel “LPA_Record” dalam tampilan sheet “RECORD” pada Gambar 3.8.



Gambar 3.8 *Tabel Excel LPA_Record untuk data record dari SharePoint Lists dalam sheet “RECORD”*

Data record yang sudah masuk dalam tabel LPA_Record, dapat dilakukan *lookup* dari sebuah cell setiap column dari sheet “RECORD” ke cell yang terdapat dalam sheet “LPA” untuk digunakan *formula structured references*, seperti yang terlihat pada formula Excel berikut pada Gambar 3.9.

```
=LPA_Record(<kolom sesuai data tabel>)
```

Gambar 3.9 *Formula Excel untuk mengambil value*

Formula dengan fungsi `VLOOKUP` digunakan untuk mengambil data dari isian sebuah *cell* pada sheet “RECORD”. Nilai tersebut kemudian dicocokkan dengan *data source* yang berasal dari tabel pada sheet “DATABASE”. Setelah kecocokan ditemukan, fungsi ini akan mengambil data dari kolom yang ditentukan, lalu mengembalikannya (*return*) ke *cell* tujuan yang berada pada sheet “LPA”, sesuai dengan formula yang digunakan.

Sebagai contoh, pada *cell* Kelurahan di sheet “LPA”, tepatnya pada cell F7, dilakukan proses *lookup* terhadap nilai dari sheet “RECORD” pada cell “alamat_kelurahan” yang berisi *value* 457. Nilai tersebut kemudian dicocokkan dengan data referensi pada *sheet* DATABASE dengan rentang data A2:G1552, yang berisi data statis kelurahan untuk wilayah Provinsi Banten. Berdasarkan pencocokan tersebut, didapatkan bahwa nilai 457 merujuk pada Kelurahan Walantaka. Fungsi `VLOOKUP` untuk mengambil data *cell* sebagai berikut pada *formula* Gambar 3.10:

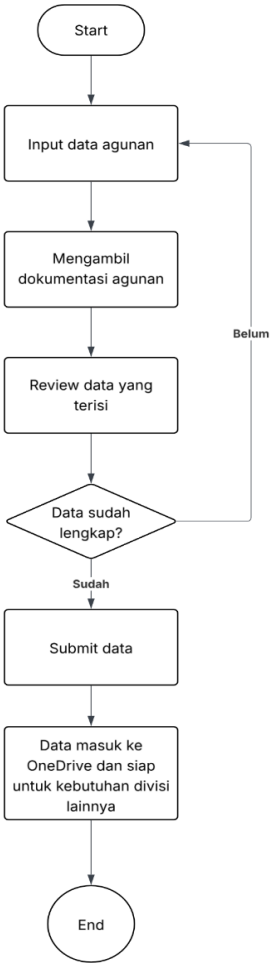
```
=VLOOKUP(LPA_Record(kolom sesuai data tabel); DATABASE! A2:G1552; <deretan kolom
ke berapa yang diambil>; FALSE)
```

Gambar 3.10 *Formula Excel untuk mengambil data cell menggunakan fungsi VLOOKUP*

3.4.6 *Desain Flow* aplikasi eLPA

Setelah semua pengaturan awal selesai, mulai dari aplikasi, sumber data, sampai integrasi dengan Excel, penulis mulai masuk ke proses mendesain tampilan aplikasi. Proses ini dikerjakan selama kurang lebih dua bulan, dengan tujuan utama membuat antarmuka yang simpel, mudah dipakai, dan sesuai dengan analisis kebutuhan di awal, terutama kebutuhan petugas lapangan. Desain dibuat menggunakan Power Apps Canvas yang memungkinkan pembuatan antarmuka secara visual, cukup dengan *drag and drop* komponen seperti *text input*, *dropdown*, dan tombol *submit* tanpa perlu menulis kode secara manual. Tampilan dirancang agar petugas bisa langsung paham alurnya saat membuka aplikasi, isi data, lalu submit. Selain itu, ditambahkan juga pengecekan otomatis supaya data yang masuk sesuai format dan lengkap sebelum disimpan ke database SharePoint Lists.

Berdasarkan dari hasil analisis kebutuhan di awal, penulis mendapatkan *user flow* sebagai berikut pada Gambar 3.11:



Gambar 3.11 *User flow penggunaan aplikasi*

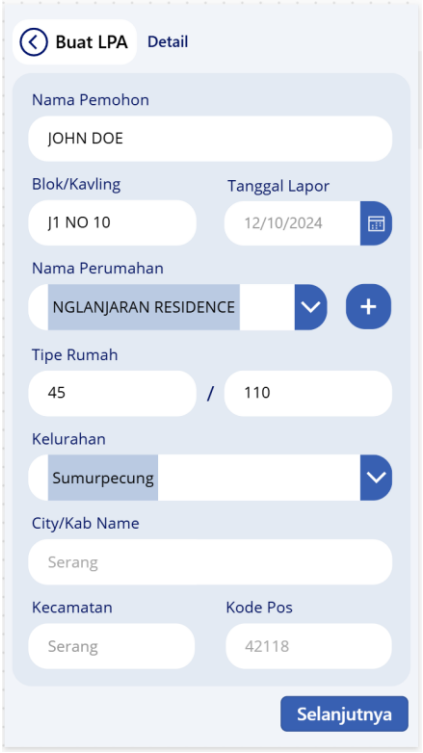
Fitur utama dari aplikasi ini yaitu dapat membantu pekerjaan petugas lapangan untuk dapat melaporkan agunan lebih cepat, bukan secara manual seperti sebelumnya, sesuai dengan hasil analisis dan tujuan awal pengembangan proyek ini. Fitur ini terdiri dari pengisian form data seperti aplikasi form lainnya, fitur foto langsung dari aplikasi, fituer *manage* data perumahan beserta developernya dengan kemampuan CRUD (*Create, Read, Update,* dan *Delete),* fitur *manage* data agunan, dan fitur review data form yang telah terisi.

3.4.7 Desain fitur form halaman detail agunan

Pada halaman pertama dari eLPA, *field* pada form berisikan data alamat agunan pemohon (nama pemohon, blok/kavling, alamat perumahan, tipe rumah, dan wilayah administratif), dan tanggal lapor. Kemudian beberapa *button* untuk melakukan navigasi antar halaman, dan tombol untuk berpindah ke halaman tambah data perumahan, di sebelah kanan field nama perumahan.

Isian Nama Pemohon dan Blok/Kavling akan otomatis menjadi huruf kapital ketika diisi huruf kecil, memudahkan pengguna untuk mengisi data pemohon agar tetap sesuai, dengan tujuan meminimalisir *human error*.

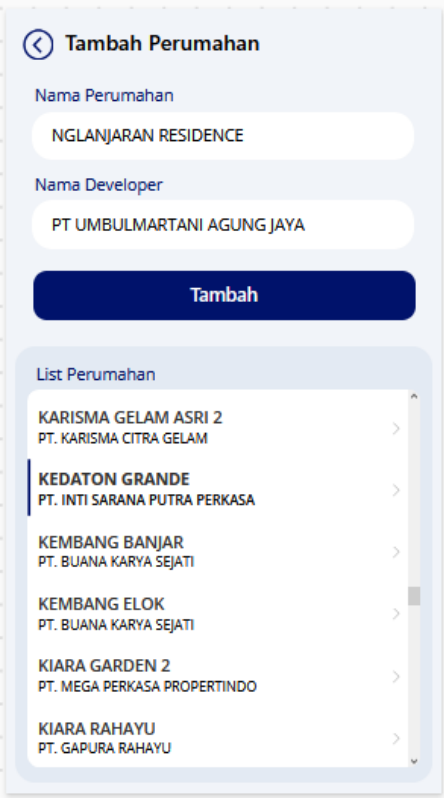
Berikut merupakan tampilan halaman pertama eLPA dalam Gambar 3.12.



Gambar 3.12 *Tampilan halaman pertama eLPA*

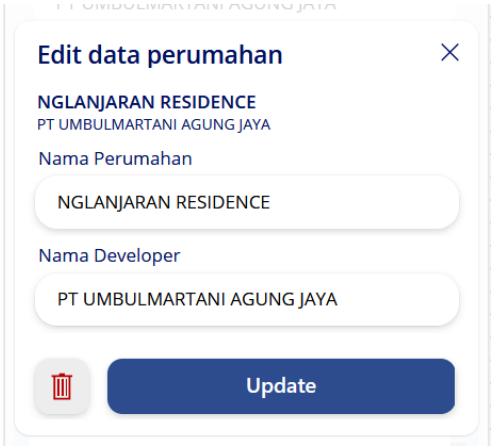
3.4.8 Desain fitur form halaman tambah perumahan

Di samping dropdown nama perumahan pada Gambar 3.18, terdapat tombol (+), yang nantinya bisa menambahkan sebuah list rumah dalam page "Tambah Perumahan" dengan sistem CRUD (*Create, Read, Update, Delete*). Pada halaman ini, terdapat field Nama Perumahan dan Nama Developer yang setelah diisi dan ditambah, akan muncul dalam sebuah *control* yaitu *gallery.* Pengguna dapat melakukan edit dalam *list* perumahan dengan menekan salah satu *list* rumah dalam gallery. Berikut merupakan tampilan halaman tambah perumahan pada Gambar 3.13.



Gambar 3.13 *Tampilan halaman tambah perumahan*

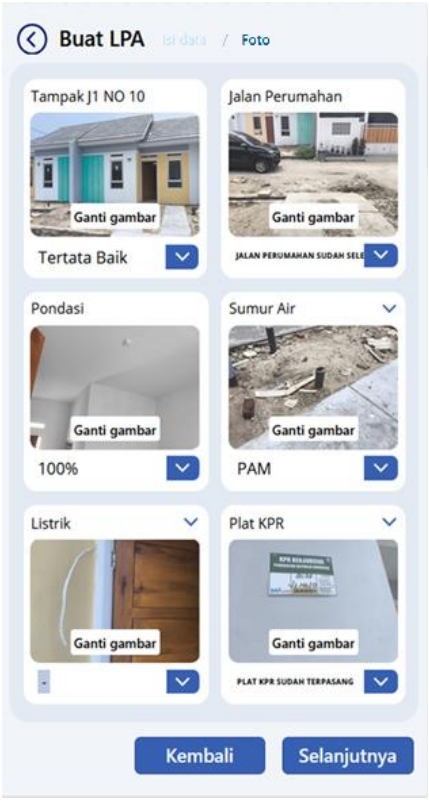
Ketika salah satu perumahan dipilih, pengguna dapat melakukan edit data perumahan tersebut, yang kemudian dapat dilakukan update atau hapus data itu. Beriut merupakan tampilan Edit data perumahan pada Gambar 3.14.



Gambar 3.14 *Tampilan halaman edit data perumahan*

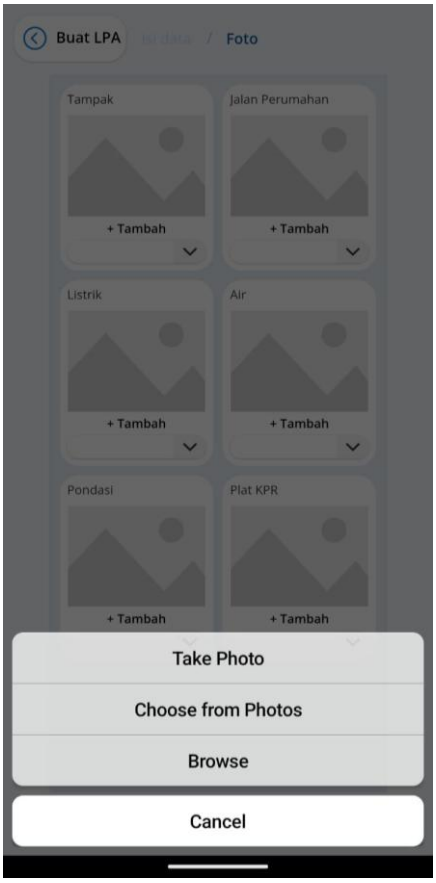
3.4.9 Desain fitur form halaman foto agunan

Pada halaman kedua eLPA merupakan tampilan untuk melakukan dokumentasi agunan, yang merupakan salah satu bukti sah laporan untuk dijadikan pelaporan kepada pihak internal. Dalam halaman ini, terdapat *control* camera dan *dropdown* untuk memberikan keterangan fasilitas atau progres terhadap gambar agunan tersebut. Berikut merupakan tampilan halaman kedua dari eLPA pada Gambar 3.15.



Gambar 3.15 *Tampilan foto eLPA untuk dokumentasi agunan*

Ketika dilakukan *input* gambar, terdapat opsi untuk melakukan *take photo* secara langsung di aplikasi dan impor gambar dari galeri. Berikut merupakan tampilan dialog saat ingin melakukan impor gambar ke dalam aplikasi, pada Gambar 3.16.



Gambar 3.16 *Tampilan dialog untuk melakukan impor gambar*

3.4.10 Desain fitur form halaman spesifikasi bangunan

Halaman ketiga merupakan bagian untuk mengisi spesifikasi fisik dari agunan. Isian *form* berbentuk dalam bentuk *dropdown* untuk setiap elemen bangunan, seperti pondasi, dinding, hingga atap. Setiap *dropdown* telah memiliki nilai default yang diisi berdasarkan material atau spesifikasi bangunan yang paling umum digunakan, agar memudahkan pengguna saat mengisi. Nilai ini bisa langsung digunakan jika sesuai, namun tetap fleksibel untuk diganti sesuai kondisi nyata di lapangan. Berikut merupakan tampilan dari halaman spesifikasi pada Gambar 3.17.



Gambar 3.17 *Tampilan halaman spesifikasi agunan*

3.4.11 Desain fitur form halaman bangunan

Halaman keempat merupakan bagian untuk mengisi data legal bangunan, berupa jenis sertifikat, status, progres arsitektur, tahun bangun, dan harga jual agunan. Gambar 3.18 merupakan tampilan dari halaman bangunan.



Gambar 3.18 *Tampilan halaman bangunan*

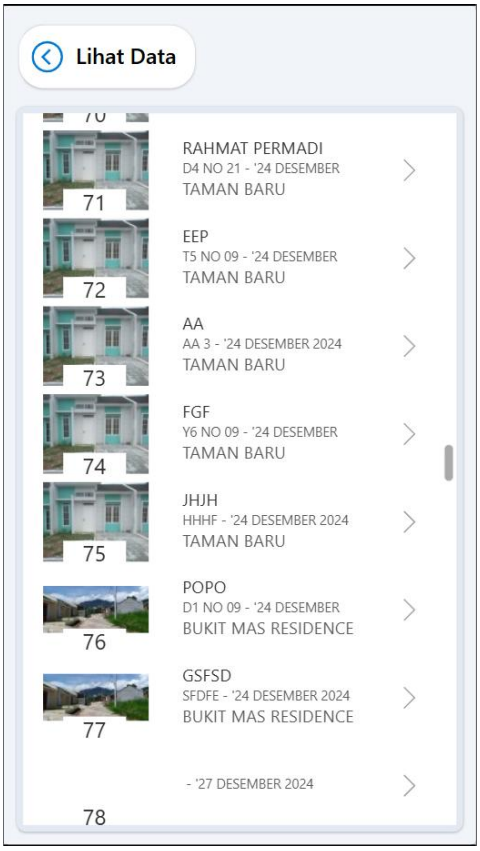
Halaman terakhir yaitu untuk melihat review dari data yang telah diisi pada halaman sebelumnya, untuk memastikan data yang diisi sudah benar sebelum dikirim ke pihak internal. Pada halaman ini terdapat gambar dari tampak depan agunan, kemudian detail di bawahnya untuk melihat isian data yang diisi pada *field* sebelumnya. Berikut merupakan tampilan dari halaman terakhir eLPA untuk melakukan review data yang terisi pada Gambar 3.19.



Gambar 3.19 *Tampilan review form*

3.4.12 Desain fitur halaman lihat data

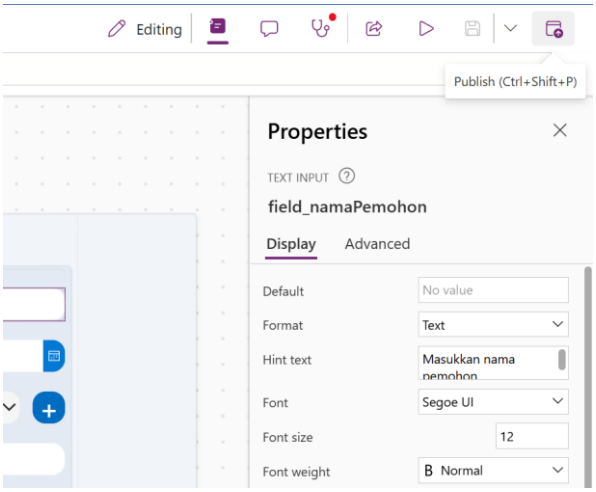
Halaman keempat merupakan bagian untuk mengisi data legal bangunan, berupa jenis sertifikat, status, progres arsitektur, tahun bangun, dan harga jual agunan. Berikut merupakan tampilan dari halaman bangunan pada Gambar 3.20.



Gambar 3.20 *Tampilan lihat data*

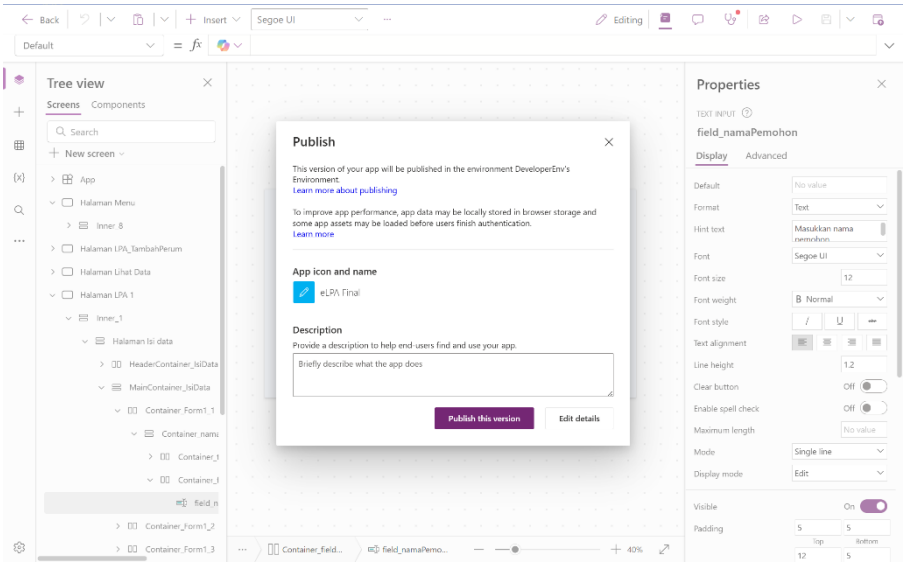
3.5 Implementasi

Dalam aplikasi yang dikembangkan dalam basis *Low-Code Development Platform* (LCDP), khususnya pada Power Apps, setelah proses desain selesai, proyek dapat langsung digunakan tanpa melalui atau deployment, karena semua perubahan tersimpan secara real time di dalam platform. Proses implementasi mencakup langkah publish atau yang umumnya dikenal sebagai proses deploy aplikasi, yang dilakukan langsung melalui Power Apps Studio menggunakan tombol “Publish” yang terletak di kanan atas editor. Berikut merupakan tampilan tombol untuk melakukan publish, seperti ditunjukkan pada Gambar 3.21.



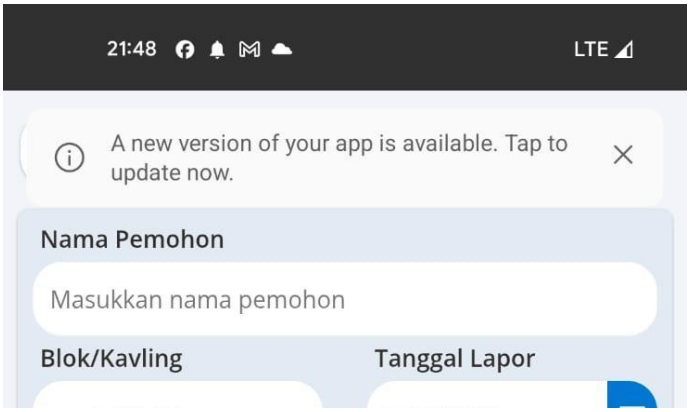
Gambar 3.21 *Tampilan halaman bangunan*

Dialog publish akan muncul untuk dapat diubah detailnya, seperti *icon* dan nama aplikasi agar pada mobile dapat muncul sesuai detail yang telah ditentukan pada proses *publish*. Jika sudah, proses *publish* dapat dilakukan dengan menekan tombol "*Publish this version*". Berikut merupakan tampilan dialog untuk melakukan publish proyek pada Gambar 3.22.



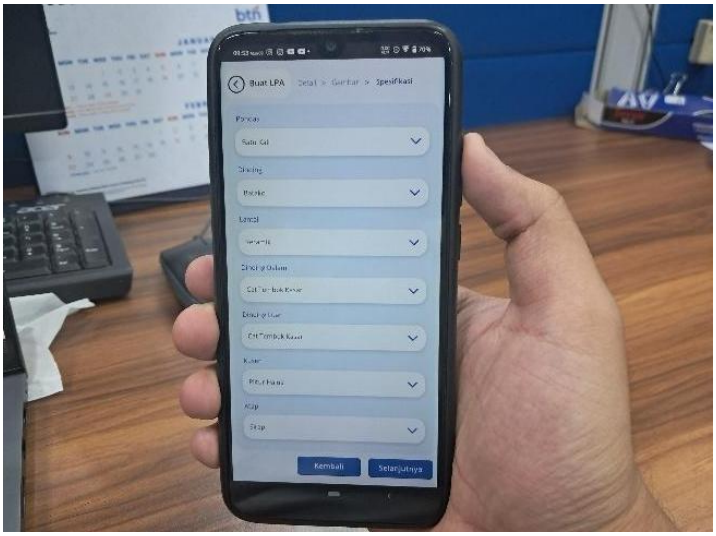
Gambar 3.22 *Tampilan dialog publish*

Pada platform *mobile* akan langsung terupdate, yang menandakan kalau terdapat versi terbaru dari tim pengembang. Pengguna dapat langsung menekan notifikasi tersebut untuk dapat langsung menggunakan versi terbaru dari aplikasi itu. Berikut merupakan tampilan notifikasi pada Power Apps mobile jika terdapat update versi terbaru pada Gambar 3.23.



Gambar 3.23 *Tampilan notifikasi update aplikasi*

Pada gambar berikut merupakan aplikasi yang sudah diimplementasikan pada platform *mobile* dari sisi pengguna. Aplikasi ini telah melewati proses desain dan *publish* di Power Apps, lalu dapat langsung diakses oleh pengguna. Tampilan ini menunjukkan bagaimana antarmuka aplikasi terlihat saat dijalankan di ponsel. Proses ini menjadi salah satu keunggulan dari Power Apps, karena memungkinkan *developer* maupun *citizen developer* untuk melihat hasil implementasi secara langsung tanpa perlu proses instalasi yang rumit. Setelah dilakukan *publish*, pengguna bisa langsung berinteraksi dengan aplikasi sesuai dengan fungsi yang telah dirancang sebelumnya. Berikut merupakan tampilan aplikasi yang sudah berjalan pada ponsel pada Gambar 3.24.



Gambar 3.24 *Tampilan eLPA yang sudah terimplementasi di ponsel*

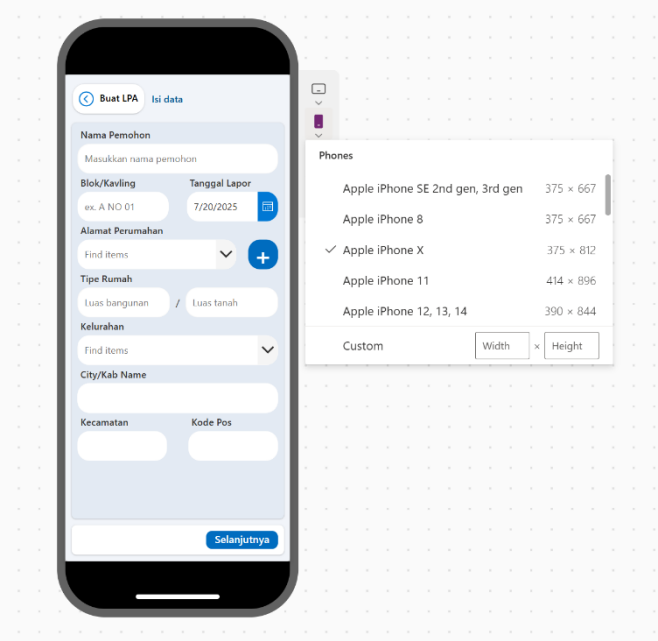
3.6 *Testing*/Pengujian

Setelah keseluruhan proyek sudah terimplementasi dalam environment, langkah selanjutnya yaitu melakukan *testing* terhadap keseluruhan fungsi aplikasi menggunakan metode *black-box testing*. *Test case* untuk pengujian mengisi *form* hingga berupa *file* Excel dapat dilihat pada Tabel 3.2.

Tabel 3.2 Hasil pengujian black box

No.	Use Case	Skenario	Hasil yang Diharapkan	Hasil Pengujian
1	Mengisi data form	Pengguna mengisi data dalam field keseluruhan form.	Field berisikan data input dari pengguna tanpa error	Berhasil
2	Melakukan edit detail salah satu perumahan dari fitur edit list perumahan	Pengguna mengubah detail salah satu perumahan yang sudah ada dalam list.	List perumahan yang terpilih dapat diupdate sesuai isian	Berhasil
3	Melakukan tambah foto pada aplikasi	Pengguna melakukan unggah foto dengan cara foto langsung dari aplikasi atau dengan impor dari galeri.	Hasil gambar dapat ditampilkan dalam form.	Berhasil
4	Menambah data agunan ke entri di SharePoint	Pengguna melakukan submit data setelah seluruh form di Power Apps terisi.	Sistem menyimpan data dari form di Power Apps ke SharePoint Lists.	Berhasil
5	Melihat hasil ekspor file Excel dalam OneDrive	Pengguna melihat laporan Excel yang terdapat dalam direktori OneDrive.	Isian data dalam Excel sama dengan input dalam aplikasi	Berhasil

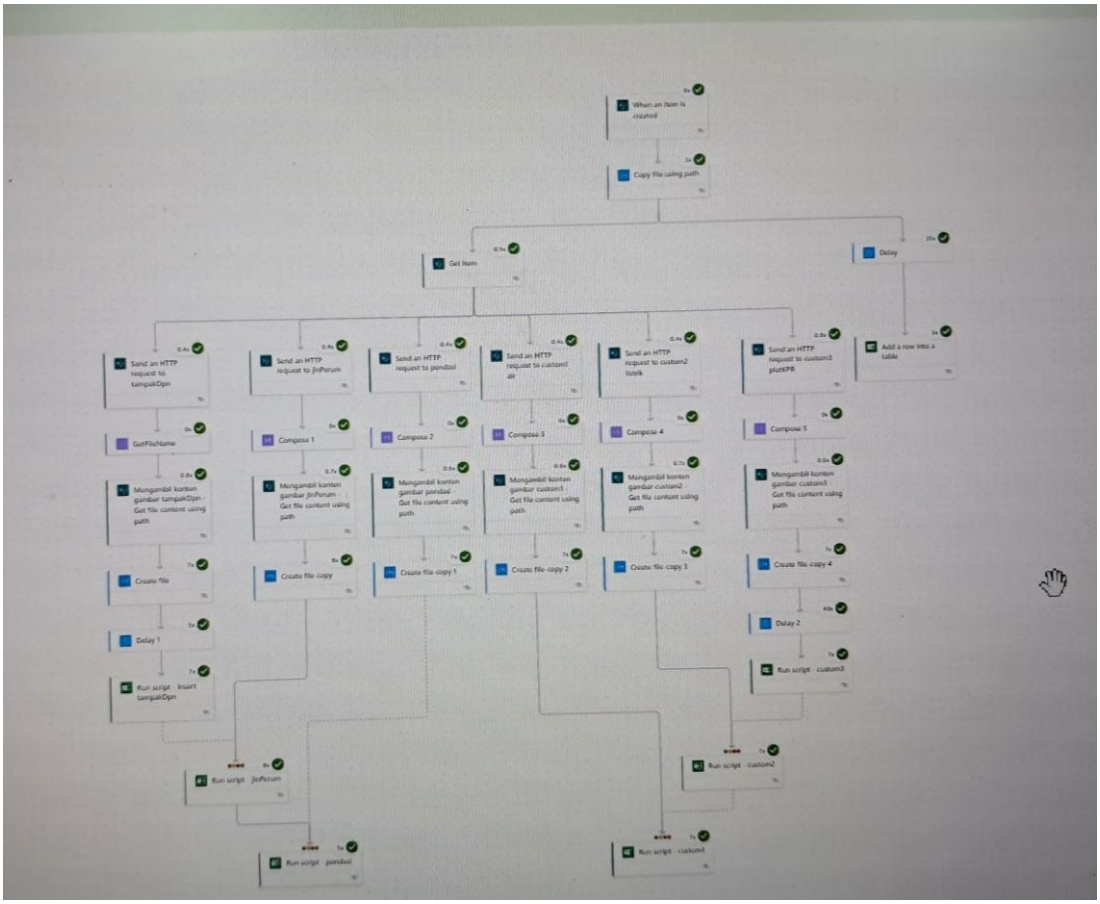
Dalam Power Apps, aplikasi yang dikembangkan dengan model *canvas* bisa langsung dicoba menggunakan fitur preview yang tersedia di dalam Power Apps Studio, memungkinkan tim pengembang untuk melihat bagaimana aplikasi akan berjalan dari sisi pengguna, tanpa harus melakukan *publish* atau *deploy* terlebih dahulu. Preview ini dapat memastikan tampilan aplikasi akan seperti bagaimana pada pengguna, meminimalisir kesalahan yang bisa muncul di versi akhir aplikasi. Berikut merupakan hasil testing aplikasi dalam Power Apps Studio pada Gambar 3.25.



Gambar 3.25 *Tampilan preview canvas*

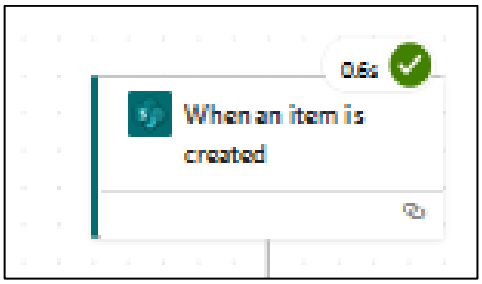
Kemudian testing ini juga berupa pengujian flow terhadap flow yang berjalan pada Power Automate. Flow yang dijalankan dari awal hingga berakhir dapat langsung diuji dalam Power Automate. Pengujian flow ini dilakukan secara menyeluruh untuk memastikan setiap tahapan dalam flow berjalan dengan baik, menggunakan fitur *flow checker* yang terintegrasi dengan Power Automate untuk menjalankan simulasi lengkap dari trigger awal hingga trigger terakhir dalam flow. Tidak hanya itu, *flow checker* juga dapat menganalisis struktur *flow* secara keseluruhan dan mengidentifikasi isu potensial *flow*, seperti *missing connections*, *invalid expressions*, atau konfigurasi yang tidak optimal.

Dalam proses testing ini, pengembang dapat memantau secara real-time bagaimana data berpindah dari satu step ke step berikutnya. Hal ini sangat membantu dalam mengidentifikasi potensi error yang mungkin terjadi di tengah proses. Setiap step dalam flow akan menampilkan status eksekusi flow, durasi pemrosesan, dan hasil output yang dihasilkan berupa JSON. Pada Gambar 3.26 merupakan status eksekusi flow yang sudah berjalan secara keseluruhan.



Gambar 3.26 *Tampilan status eksekusi keseluruhan flow*

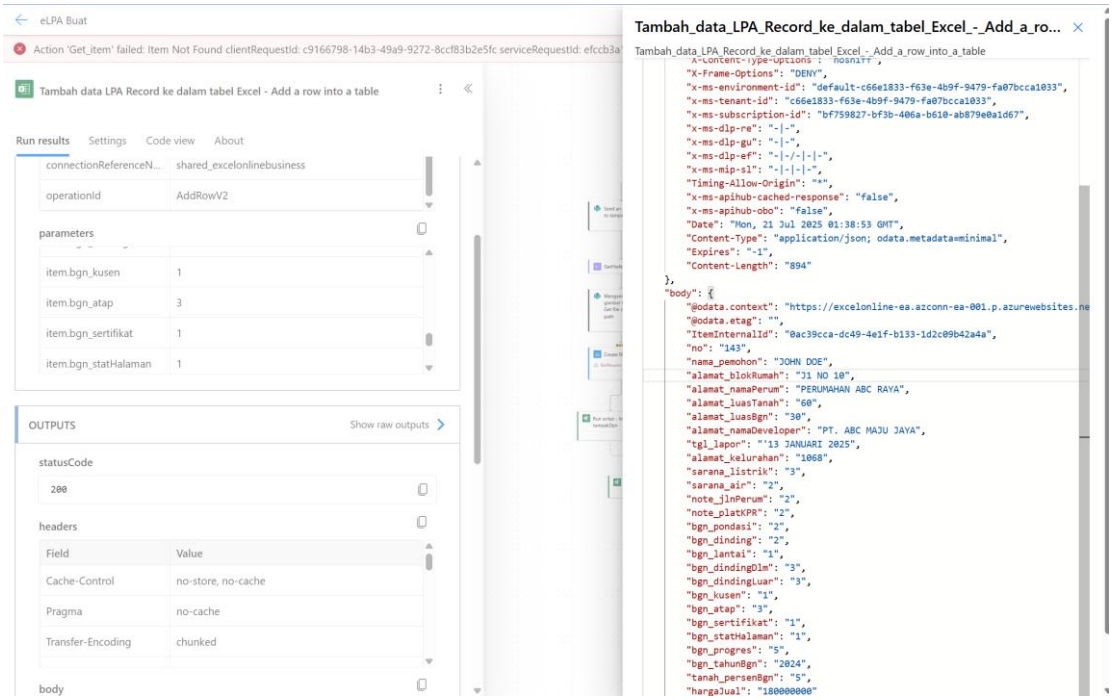
Saat salah satu flow berjalan, kemudian akan terlihat bahwa terdapat status akhir dari hasil berjalannya eksekusi, berupa durasi flow itu dieksekusi dan status akhirnya, apakah berhasil atau gagal. Pada Gambar 3.27 merupakan status eksekusi salah satu flow yang telah berjalan dengan berhasil.



Gambar 3.27 *Tampilan status eksekusi flow*

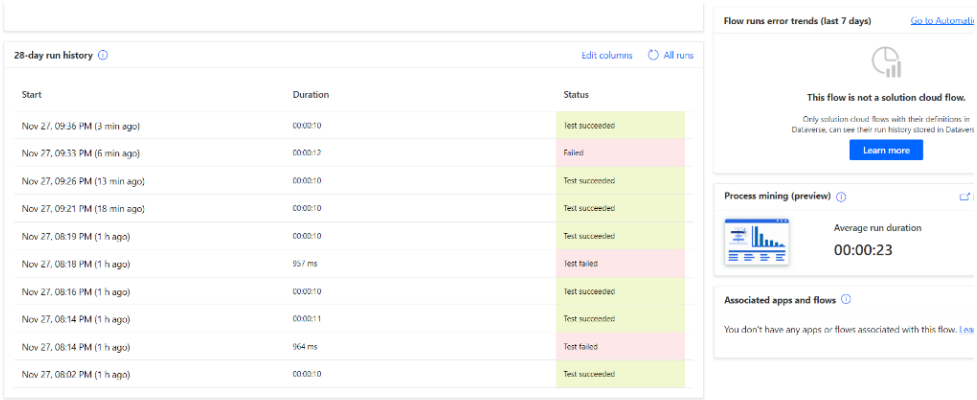
Ketika ingin melihat status keseluruhan *output* dari salah satu flow, terdapat hasil *run* yang berupa sekumpulan hasil output dalam bentuk GUI pada tampilan sebelah kiri. Jika ditekan "*Show raw outputs",* akan memunculkan deretan *output* dalam bentuk JSON pada tampilan sebelah kanan. Terlihat pada gambar bahwa flow "*Add a row into a table"* berhasil

dieksekusi dengan menghasilkan statusCode 200 “OK”, dan item parameter apa saja yang telah dikirim pada flow itu, serta *raw output* JSON untuk pengecekan lebih lanjut pada kanan layar. Pada Gambar 3.28 merupakan detail eksekusi salah satu flow serta raw outputnya.



Gambar 3.28 *Tampilan detail eksekusi salah satu flow dengan raw output JSON*

Keseluruhan satu siklus flow akan tampak hasil akhir *run* pada *flow run history*, berupa kapan flow itu dieksekusi karena trigger dari SharePoint, durasi eksekusi keseluruhannya, dan status berhasil dari keseluruhan flow di dalam siklus itu. Dalam Gambar 3.29 merupakan tampilan dari *flow run history* yang terdapat pada Power Automate.



Gambar 3.29 *Tampilan flow run history Power Automate*

3.7 Maintenance

Setelah aplikasi berjalan dan digunakan secara aktif, aktivitas utama yang dilakukan pengembang meliputi pemantauan performa aplikasi, perbaikan jika ditemukan bug atau error dari laporan pengguna, serta pembaruan data yang mungkin mengalami perubahan, seperti penambahan pilihan dropdown atau revisi isi field tertentu. Selain itu, pembaruan versi aplikasi juga bisa dilakukan langsung melalui Power Apps Studio, baik untuk memperbaiki alur kerja, menambahkan fitur, atau menyempurnakan tampilan antarmuka. Setelah perubahan selesai, aplikasi dapat dipublikasikan ulang *(republish)* agar versi terbaru bisa langsung digunakan oleh semua pengguna tanpa perlu instal ulang. Fitur di Power Apps juga memungkinkan tim pengembang untuk melihat riwayat versi yang pernah dipublikasikan, sehingga mudah untuk melakukan *rollback* jika diperlukan.

3.8 Penutupan Proyek

Penutupan aktivitas pengembangan proyek aplikasi ERP eLPA untuk pelaporan agunan berbasis Power Platform ditandai dengan selesainya proses implementasi proyek ke environment Microsoft 365 dan pengujian yang sudah minim error. Pada tahap desain antarmuka (UI/UX), masih terdapat beberapa komponen halaman seperti edit data *record* agunan dan fitur notifikasi otomatis yang belum sempat diselesaikan karena keterbatasan waktu selama masa magang berlangsung. Mengingat tugas penulis telah selesai, maka pengembangan lebih lanjut, termasuk penambahan fitur dan proses implementasi sistem secara menyeluruh, akan diteruskan oleh tim internal perusahaan yang bersangkutan.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Selama pelaksanaan magang, penulis terlibat langsung dalam pengembangan sistem Enterprise Resource Planning (ERP) berbasis platform Microsoft Dynamics 365, dengan Microsoft Power Apps sebagai aplikasi utama. Relevansi akademik yang didapatkan dari pengalaman magang ini yaitu terdapat pada aktivitas pengembangan sistem dengan materi yang telah penulis pelajari di Program Studi Informatika, termasuk membentuk relasi antar tabel berbentuk ERD.

Selain itu, keterlibatan penulis dalam proses digitalisasi dan otomasi alur bisnis perusahaan melalui platform *low-code* juga sejalan dengan tren transformasi digital yang sering dibahas dalam perkuliahan, terutama dalam konteks efisiensi operasional dan integrasi data.

4.2 Lingkungan Magang

Di tempat magang, setiap pegawai punya tugas dan tanggung jawab yang jelas. Suasana kerja cukup disiplin, baik soal kehadiran, jam kerja, maupun mengikuti aturan yang berlaku. Tapi di sisi lain, komunikasi di kantor juga terbuka. Walaupun penulis hanya mahasiswa magang, tetap diajak ikut diskusi dengan pembimbing lapangan maupun staf lain. Dari pengalaman ini, penulis belajar beradaptasi, menghargai cara kerja orang lain, dan memahami pentingnya profesionalitas di lingkungan kerja.

4.2.1 Aktivitas Harian

Selama magang, penulis menjalani jadwal kerja dari hari Senin hingga Jumat pukul 08.00–15.30, dengan waktu istirahat pada pukul 11.55–13.00. Aktivitas harian dimulai dengan briefing singkat bersama pembimbing lapangan untuk membahas pekerjaan yang harus diselesaikan, termasuk perkembangan proyek aplikasi yang sedang dikerjakan. Kegiatan utama penulis selama jam kerja berfokus pada pengembangan aplikasi, yang mencakup mempelajari sistem internal yang sudah ada, mengidentifikasi kebutuhan pengguna, serta merancang alur sistem sesuai prosedur perbankan syariah. Penulis juga membantu tugas petugas Lapangan Penilaian Agunan (LPA) dan bagian Financing Administration dalam membuat laporan agunan ketika ada data baru yang masuk, jika dari mereka memerlukan bantuan secara sesekali.

Selain mengerjakan pekerjaan teknis, penulis sering berdiskusi dengan pegawai dari unit kerja lain untuk memahami bagaimana aplikasi yang dikembangkan dapat menunjang kegiatan operasional cabang. Di sela kegiatan, penulis juga melakukan dokumentasi progres, menguji sistem yang sedang dikembangkan, serta melaporkan hasilnya kepada *supervisor*.

4.2.2 Pengembangan Softskill, Hardskill, dan Pembelajaran

Magang ini tidak hanya memberikan pengalaman teknis, tetapi juga mendorong pertumbuhan keterampilan secara menyeluruh. Dari sisi *hardskill*, penulis memperoleh pemahaman lebih mendalam tentang pengembangan aplikasi berba, integrasi dengan *database* (dalam hal ini SharePoint), serta penerapan aturan compliance yang harus selaras dengan kebijakan kantor pusat. Proses *coding*, *debugging*, dan pengujian aplikasi memberikan latihan nyata yang memperkuat keterampilan teknis penulis.

Sementara dari sisi *softskill*, penulis belajar banyak tentang manajemen waktu, komunikasi efektif, dan kerja sama tim. Pada awalnya penulis merasa canggung untuk menyampaikan pendapat atau bertanya ketika ada hal yang kurang dipahami. Namun seiring berjalannya waktu, kepercayaan diri penulis tumbuh, serta keterampilan berkomunikasi menjadi lebih baik karena terbiasa berdiskusi dengan pegawai yang latar belakangnya berbeda. Selain itu, penulis juga belajar menghadapi tantangan, misalnya ketika menemui kendala teknis atau ketika harus menyesuaikan rancangan aplikasi agar sesuai dengan prosedur perbankan. Pembelajaran utama yang diperoleh adalah bagaimana menghubungkan ilmu yang dipelajari di bangku kuliah dengan kebutuhan nyata di dunia kerja.

4.3 Pembelajaran Magang

4.3.1 Manfaat

Selama menjalani magang, penulis memperoleh berbagai manfaat terutama dapat lebih memahami hubungan antar tabel, terutama yang terkait dengan jenis data di Power Apps yang beragam jenisnya. Penulis juga belajar tentang efisiensi data, seperti konsep one-to-many dan many-to-one, yang sangat penting dalam mengelola data aplikasi dan performa aplikasi.

Penulis juga mempelajari penggunaan JSON, baik untuk mengambil dan mengirim data melalui API, maupun untuk mengolah data yang digunakan di aplikasi. Selain itu, penulis mempelajari penggunaan Power Automate untuk mengotomatisasi alur kerja dan mengelola data secara otomatis. Dengan Power Automate, pengguna aplikasi dapat melakukan berbagai

tugas otomatis, seperti mengambil data secara otomatis dan integrasi antar aplikasi, yang dapat meningkatkan efisiensi operasional.

4.3.2 Kendala, Hambatan dan Tantangan

Selama pelaksanaan magang berlangsung, penulis menemukan bahwa ruang lingkup pekerjaan di kantor cabang memerlukan sebuah sistem dalam pengolahan data serta penyimpanan dokumen. Dari sisi kelebihan, cabang memiliki fleksibilitas untuk mencoba ide atau aplikasi yang dapat membantu pekerjaan sehari-hari, selama tetap mengikuti aturan yang ditetapkan oleh kantor pusat. Namun di sisi lain, hal seperti ini terdapat juga beberapa kelemahan. Kewenangan cabang yang terbatas menyebabkan banyak inisiatif harus melalui persetujuan kantor pusat atau DPS (Dewan Pengawas Syariah). Hal ini berdampak pada *prototype* aplikasi yang telah dikembangkan tidak dapat langsung digunakan, menyebabkan proses uji coba dan perbaikan aplikasi berjalan lebih lambat. Meskipun demikian, pengalaman magang ini memberikan penulis banyak pelajaran penting bagaimana menyeimbangkan kebutuhan inovasi di cabang dengan aturan kepatuhan yang berlaku di tingkat pusat.

Salah satu kendala utama yang penulis temui adalah terbatasnya pengalaman dan kemampuan dalam menggunakan framework yang digunakan pada situs magang, yaitu Microsoft Power Platform dan Dynamics 365 dengan basis platform Power Apps yang menggunakan metode pengembangan SDLC model *Waterfall*. Penulis juga merasa sedikit terhambat dalam menyelesaikan tugas yang diberikan oleh supervisor karena penulis belum sepenuhnya menguasai framework tersebut. Dalam hal ini penulis dapat mengatasi masalah tersebut dengan mencari solusi berupa tutorial di internet, dan sebagian banyak dengan bantuan beberapa *tools AI,* seperti ChatGPT, Claude AI, dan Perplexity, dengan tujuan penulis dapat menyelesaikan sebagian banyak masalah dalam *code* dan sebagian banyak untuk perbaikan antarmuka Power Apps.

4.4 Pembelajaran Magang

Selama pelaksanaan magang, penulis memperoleh berbagai pembelajaran dari sisi teknis maupun non-teknis yang benar-benar berarti untuk ke depannya, terutama ke masa depan penulis apabila bekerja di dunia nyata. Berikut pembelajaran magang yang didapatkan penulis selama magang:

4.4.1 Pemahaman Platform Low-Code Microsoft Power Platform

Penulis mempelajari cara membangun aplikasi dengan basis LCDP menggunakan Microsoft Power Apps, yang biasana termasuk mengatur *layout*, menambahkan kontrol seperti tombol (*button*), input teks, dropdown, dan galeri data. Penulis juga memahami bagaimana menghubungkan komponen tersebut dengan *data source* seperti SharePoint Lists dan Microsoft Dataverse.

4.4.2 Pemahaman Penggunaan Rumus dan Fungsi dalam Excel

Penulis memahami rumus dan fungsi yang dipakai pada Excel, beberapa contohnya seperti: VLOOKUP, cell reference, dan fungsi *Logical* (AND, OR, dan NOT) serta *Conditional* (fungsi IF) untuk penggunaan referensi data dalam suatu cell ke cell lain, atau untuk penggunaan data yang memerlukan fungsi *logical* atau *conditional*.

4.4.3 Pemahaman Manajemen Data dan Relasi

Dalam pengelolaan data, penulis belajar bagaimana menyusun struktur data yang efisien di SharePoint, melakukan pemetaan relasi antar tabel (*lists*), serta integritas data saat dilakukan proses *Create, Read, Update,* dan *Delete* (CRUD), yang melibatkan pembuatan list untuk data perumahan, developer, dan keterkaitan antar entitas tersebut.

4.4.4 Penerapan Otomatisasi Proses Bisnis

Penulis juga mempelajari penggunaan Power Automate untuk membuat alur kerja otomatis, seperti pengiriman notifikasi email secara otomatis saat data baru ditambahkan, pembaruan status data, atau penyimpanan file ke OneDrive secara otomatis berdasarkan kondisi tertentu. Pembelajaran ini memperkenalkan penulis pada konsep *event-driven automation* dan *conditional logic*.

a. Desain Antarmuka dan Pengalaman Pengguna (UI/UX)

   Selain sisi fungsional, penulis juga mempelajari pentingnya merancang antarmuka yang intuitif dan mudah digunakan, terutama karena aplikasi ini akan digunakan oleh tim lapangan non-teknis.

b. Komunikasi antar tim

   Di luar aspek teknis, penulis mempelajari bagaimana menyampaikan hasil pekerjaan secara lebih profesional, baik dalam bentuk dokumentasi maupun presentasi. Penulis juga berlatih untuk bisa menerima masukan dari supervisor dan pengguna, serta refleksi diri berdasarkan umpan balik tersebut.

BAB V
PENUTUP

5.1 Kesimpulan

Penulisan tugas akhir ini bertujuan untuk memberikan gambaran secara komprehensif mengenai proses pengembangan aplikasi sistem ERP berbasis platform *low-code* yang dilakukan oleh penulis selama menjalani kegiatan magang di Bank BTN Syariah Kantor Cabang Serang. Selain sebagai sarana implementasi pengetahuan yang diperoleh selama masa studi, penulisan ini juga dimaksudkan untuk memperkenalkan lebih lanjut penggunaan platform low-code Microsoft Power Platform, yang terdiri dari Power Apps, Power Automate, serta integrasi data menggunakan SharePoint Lists dan OneDrive. Diharapkan pengetahuan yang disampaikan dalam laporan ini dapat menjadi pijakan awal dalam merancang dan membangun sistem berbasis *low-code*, khususnya untuk kebutuhan digitalisasi dan otomasi proses bisnis seperti pelaporan agunan dari lapangan ke pusat perusahaan. Berdasarkan hasil kegiatan magang dan penyusunan laporan tugas akhir, maka dapat diambil kesimpulan sebagai berikut:

a. Aplikasi pelaporan agunan yang dikembangkan dengan Power Apps dapat mempermudah petugas lapangan dalam melakukan input data secara langsung melalui perangkat mobile, sehingga mempercepat proses pelaporan dan mengurangi risiko kesalahan manual dalam pencatatan.
b. Integrasi otomatis dengan Power Automate dan OneDrive memungkinkan sistem untuk secara otomatis menyimpan hasil pelaporan ke dalam folder perusahaan yang terstruktur, sehingga proses dokumentasi menjadi lebih tertata dan mudah diakses oleh pihak terkait.
c. SharePoint List berperan sebagai pusat penyimpanan data (database) yang fleksibel dan mudah dikelola, sekaligus menjadi media integrasi antara Power Apps dan Power Automate dalam alur kerja pelaporan.
d. Metode pengembangan yang digunakan berbasis SDLC model *Waterfall*, yang terdiri dari tahapan perencanaan, pengembangan, hingga pengujian dan penyempurnaan secara bertahap. Namun, tantangan tetap ada, seperti keterbatasan waktu dalam daily meeting dan kesulitan mengumpulkan umpan balik dari end-user secara langsung selama masa pengujian.

e. Platform low-code Microsoft Power Platform terbukti dapat mempercepat proses pengembangan aplikasi tanpa memerlukan kemampuan pemrograman lanjutan. Fitur seperti *template, connector, dan automated flow* membantu dalam membangun sistem yang kompleks secara efisien, serta dapat dikembangkan lebih lanjut.

5.2 Saran

Berdasarkan hasil implementasi dan evaluasi sistem pelaporan agunan menggunakan Microsoft Power Platform selama kegiatan magang, terdapat beberapa aspek yang memerlukan pengembangan lebih lanjut untuk mencapai optimalisasi sistem yang maksimal. Berikut adalah saran dari penulis untuk Bank BTN Syariah Kantor Cabang Serang:

a. Menambahkan fitur pencarian dalam halaman lihat data agunan agar mudah dicari jika laporan sudah dalam jumlah yang banyak.
b. Menambahkan fitur ekspor data ke PDF langsung dari aplikasi jika memungkinkan data untuk dikirim secara langsung dari lapangan atau dicetak.
c. Menambahkan fitur sinkronisasi OneDrive untuk dapat membaca agunan yang terlapor dari direktori, bukan dari list record pada SharePoint, karena data bisa saja diubah apabila terdapat kemajuan progres dari agunan tersebut, seperti: bangunan yang masih pondasi, saat kunjungan selanjutnya sudah terpasang atap.

DAFTAR PUSTAKA

Amt-Lab. (2021). *Notion: Task management and knowledge management combined*. Arts Management and Technology Laboratory. https://amt-lab.org/reviews/2021/12/notion-task-management-and-knowledge-management-combined

Banjarnahor, A. R., Hariningsih, E., Mathory, E. A. S., Yusditara, W., Fuadi, Muliana, Handiman, U. T., Elistia, Sisca, Rimbano, D., Jatiningrum, C., Simarmata, J., Mardia, Purba, B., Fandeli, H., Muadzah, & SN, A. (2022). *Teknologi Digital Kewirausahaan dan UMKM*. Yayasan Kita Menulis. https://www.researchgate.net/publication/362706911

Berardinelli, L., Colantoni, A., Ruscio, D. D., Khorram, F., Indamutsa, A., Ibrahimi, I., Pierantonio, A., & Sahay, A. (2020). *Low-Code Engineering Repository Architecture Specification* (Deliverable No. D4.1). University of L'Aquila, Lowcomote Project. https://www.lowcomote.eu/

Boppana, V. R. (2025). Automation Using Power Platform. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5134991

Cebekhulu, B., & Ozor, P. (2022). THE INFLUENCE OF QUALITY MANAGEMENT AND ERP SYSTEMS ON ORGANISATIONAL CULTURE AND PERFORMANCE. *Proceedings on Engineering Sciences*, *4*(1), 41–50. https://doi.org/10.24874/PES04.01.007

Kakolu, S., & Faheem, M. A. (2023). Revolutionizing industries through digitization: The strategic importance of mobile application automation. *International Journal of Science and Research Archive*, *8*(1), 973–987. https://doi.org/10.30574/ijsra.2023.8.1.0276

Khaidir K.N., L., Nuragustin, N., Aldina, S. A. F., Noviyana, N., & Nurbaiti, N. (2024). Transformasi Bisnis di Era Digital: Peran Teknologi Internet dalam Mengembangkan E-Bisnis. *Musytari : Jurnal Manajemen, Akuntansi, dan Ekonomi*, *Vol. 11 No. 6 (2024): Musytari : Neraca Manajemen, Akuntansi, dan Ekonomi*, 130–140. https://doi.org/10.8734/musytari.v11i6.8141

Martinez, E., & Pfister, L. (2023). Benefits and limitations of using low-code development to support digitalization in the construction industry. *Automation in Construction*, *152*, 104909. https://doi.org/10.1016/j.autcon.2023.104909

Microsoft. (2025a). *Overview of Power Apps*. What is Power Apps? https://learn.microsoft.com/en-us/power-apps/powerapps-overview

Microsoft. (2025b). *Types of flows in Power Automate*. What is Power Automate? https://learn.microsoft.com/en-us/power-automate/flow-types

Moon, H. (2020). *Implementation of a Mobile Business Application Built in Microsoft Power Platform* (Bachelor's Thesis, Haaga-Helia University of Applied Sciences). https://www.theseus.fi/handle/10024/339880

Phalake, V., & Joshi, S. (2021). Optimized Low Code Platform for Application Development. *International Journal of Contemporary Architecture*, *Vol. 8, No. 2*. https://www.researchgate.net/publication/362519472_Optimized_Low_Code_Platform_for_Application_Development

Putri, I. A., & Munawaroh. (2025). Pemanfaatan Teknologi dalam Membangun Sumber Daya Manusia di Era Bisnis Digital: Mengetahui Tantangan dan Solusinya. *Profit: Jurnal Manajemen, Bisnis dan Akuntansi*, *4*(1), 147–156. https://doi.org/10.58192/profit.v4i1.2967

Razak, S. F. A., Ernn, Y. P., Yussoff, F. I., Ali Bukar, U., & Yogarayan, S. (2024). Enhancing Business Efficiency through Low-Code/No-Code Technology Adoption: Insights from an Extended UTAUT Model. *Journal of Human, Earth, and Future*, *5*(1), 85–99. https://doi.org/10.28991/HEF-2024-05-01-07

Thaker, B. (2023). Low-Code Platforms and Business Process Transformation. *Journal of the Maharaja Sayajirao University of Baroda*, *57*(1 (II)), 63–69.

Theo, J. (2022). *Knowledge management made easy with Notion: Here's how*. Medium. https://medium.com/%40theo-james/knowledge-management-made-easy-with-notion-heres-how-c94820f5c58e

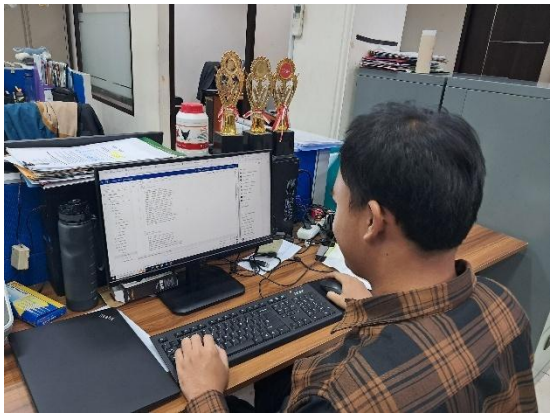
Utama, R. P., & Wardana, R. S. (2022). Utilization of Low Code Framework Tool for Knowledge Management in Well Service. *Proceedings of the IATMI 2022 Conference*. https://journal.iatmi.or.id/index.php/ojs/article/view/302/309

Zhang, J., Wu, Q., & Ma, S. (2021). *Personal knowledge management in the digital era* [Preprint]. arXiv. https://arxiv.org/abs/2112.14073

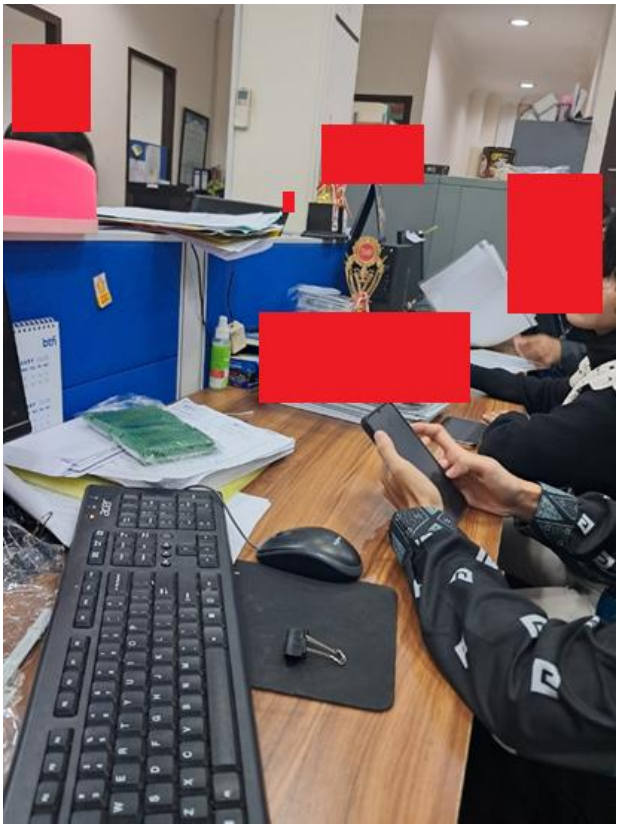
LAMPIRAN



Penulis dengan dua Tim IT Support (kiri) dan dua Financing Administration selaku supervisor



Penulis yang sedang mendesain eLPA



Testing aplikasi setelah implementasi proyek, pada petugas lapangan