

LAPORAN PRAKTIK KERJA LAPANGAN

**Pengembangan Front-End Aplikasi Desktop Berbasis QML untuk
Konfigurasi Touch IC pada Layar Smartphone di Viewtrix
Technology Limited**



Stella Maureen Ignacia Santoso

312110014

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI DAN DESAIN

UNIVERSITAS MA CHUNG

MALANG

2024

**LEMBAR PENGESAHAN
PRAKTIK KERJA LAPANGAN**

**PENGEMBANGAN FRONT-END APLIKASI DESKTOP BERBASIS QML
UNTUK KONFIGURASI TOUCH IC PADA LAYAR SMARTPHONE DI
VIEWTRIX TECHNOLOGY LIMITED**

Oleh:

**STELLA MAUREEN IGNACIA SANTOSO
NIM. 312110014**

dari:

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI dan DESAIN
UNIVERSITAS MA CHUNG**

Dosen Pembimbing,



Mochamad Subianto, S.Kom., M.Cs.
NIP. 20100002

Dekan Fakultas Teknologi dan Desain,



Prof. Dr. Eng. Romy Budhi Widodo
NIP. 20070035

KATA PENGANTAR

Segala puji syukur penulis haturkan kepada Tuhan Yang Maha Esa, yang telah melimpahkan segala berkat, karunia dan kasih setia – Nya kepada penulis, serta memberikan kesempatan kepada penulis untuk dapat menyelesaikan Laporan PKL (Praktik Kerja Lapangan) ini dengan lancar. Laporan ini disusun sebagai bentuk tanggung jawab untuk memenuhi syarat kelulusan bagi mahasiswa Fakultas Teknologi dan Desain Universitas Ma Chung.

Penulis juga menyadari bahwa Laporan PKL ini masih jauh dari kata sempurna, baik dalam segi penulisan, segi bahasa, dan segi penyusunan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun sehingga penulis memiliki acuan untuk dapat menjadi lebih baik lagi ke depannya.

Penulis juga mengucapkan terima kasih kepada beberapa pihak yang telah memberikan masukan, dukungan, bantuan dan doa dalam menyelesaikan Laporan PKL, yaitu kepada :

1. Bapak Prof. Dr. Ir. Stefanus Yufra Menahen Taneo, MS., selaku Rektor Universitas Ma Chung.
2. Bapak Dr.Eng. Romy Budhi, ST., MT. selaku Dekan dari Fakultas Teknologi dan Desain Universitas Ma Chung.
3. Bapak Hendry Setiawan, ST, M.Kom selaku Kepala Program Studi Teknik Informatika Fakultas Teknologi dan Desain Universitas Ma Chung, sekaligus ketua penguji.
4. Bapak Mochamad Subianto, S.Kom., M.Cs. selaku dosen pembimbing dan penguji yang telah bersedia membimbing dan mengarahkan penulis sehingga penulis dapat menyelesaikan laporan PKL dengan lancar.
5. Seluruh Dosen Teknik Informatika Universitas Ma Chung, yang telah berkontribusi membantu mahasiswa untuk

mendapat bekal pengetahuan dan wawasan yang baik.

6. Seluruh Staff Universitas Ma Chung, yang memberikan bantuan, ilmu dan inspirasi bagi penulis.
7. Bapak Ahmad Rofiq Isnaini Subandi, selaku Supervisor yang telah menerima dan menyambut penulis untuk bekerja bersama dalam satu kesatuan Viewtrix. Penulis ucapkan terima kasih atas pengalaman kerja yang baik dan berkesan.
8. Segenap Staff dan team Viewtrix, yang telah memberikan dukungan, wawasan, keterbukaan dan sikap menerima kepada penulis, sehingga penulis merasa mendapatkan lingkungan kerja yang baik.
9. Orang tua, yang telah memberikan doa, dan dukungan baik spiritual maupun material kepada penulis sehingga penulis bisa menyelesaikan Laporan PKL.
10. Keluarga, yang telah memberikan doa dan dukungannya kepada penulis sehingga penulis bisa menyelesaikan Laporan PKL tepat waktu.
11. Teman – teman, yang telah memberikan doa dan dukungannya kepada penulis sehingga penulis bisa menyelesaikan Laporan PKL tepat waktu.
12. Diri Sendiri, yang telah berjuang dan mau bertekad untuk menyelesaikan Laporan PKL tepat waktu, sehingga laporan ini dapat terjadwal dan tersusun tepat waktu.

Semoga, Laporan PKL ini bisa menambah wawasan para pembaca, dan dapat bermanfaat untuk meningkatkan ilmu pengetahuan dan wawasan pembaca.

Malang, 03 Agustus 2024

A handwritten signature in black ink, consisting of stylized, overlapping letters that appear to be 'SMI'.

Stella Maureen Ignacia Santoso

NIM : 312110014

DAFTAR ISI

LEMBAR PENGESAHAN	2
KATA PENGANTAR.....	3
DAFTAR ISI	6
DAFTAR GAMBAR	8
BAB I PENDAHULUAN	10
1.1 Latar Belakang	10
1.2 Identifikasi Masalah.....	11
1.3 Batasan Masalah.....	12
1.4 Rumusan Masalah	13
1.5 Tujuan.....	14
1.6 Manfaat	15
BAB II GAMBARAN UMUM PERUSAHAAN.....	16
2.1 Deskripsi Perusahaan	16
2.2 Visi	16
2.3 Misi	17
2.4 Struktur Organisasi.....	17
2.5 Produk dan Layanan.....	18
BAB III TINJAUAN PUSTAKA.....	22
3.1 Aplikasi	22
3.2 Aplikasi Desktop	22
3.3 Qt.....	22
3.4 QML.....	23
3.5 Python	24
3.6 Pyside	24
3.7 YAML.....	25
3.8 Touch IC	26
3.9 LCD.....	27
BAB IV HASIL DAN PEMBAHASAN.....	28
4.1 Prosedur Pengerjaan.....	28
4.2 Analisa Kebutuhan Data.....	28
4.3 Desain.....	29

4.4 Pembuatan	41
4.5 Hasil Akhir	46
4.6 Pengujian.....	54
BAB V PENUTUP.....	59
DAFTAR PUSTAKA	60

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Viewtrix.....	17
Gambar 4.1 Desain Tampilan Awal Ketika Aplikasi Berjalan	29
Gambar 4.2 Desain Tampilan Ketika Tombol Notifikasi Aktif.....	31
Gambar 4.3 Desain Tampilan Ketika Tombol Pengaturan Aktif.....	31
Gambar 4.4 Desain Tampilan Ketika Menambahkan Item di Project Set	32
Gambar 4.5 Desain Tampilan Ketika Menyunting Item di Project Set.....	33
Gambar 4.6 Desain Tampilan Scan Arrangement	34
Gambar 4.7 Desain Tampilan Workspace Scan Arrangement dengan Scan List ...	35
Gambar 4.8 Desain Tampilan Scan Arrangement Ketika Menambahkan Item pada Scan List.....	36
Gambar 4.9 Desain Tampilan Scan Arrangement Ketika Menambahkan Item pada Tabel Scan	37
Gambar 4.10 Desain Tampilan Scan Arrangement Ketika Menambahkan Item pada Tabel Scan Arrangement.....	37
Gambar 4.11 Desain Tampilan Scan Arrangement Ketika Menyunting Item pada Tabel Scan	38
Gambar 4.12 Desain Tampilan Scan Arrangement Ketika Menyunting Item pada Tabel Scan Arrangement	Error! Bookmark not defined. 9
Gambar 4.13 Desain Tampilan Hardware Utility - Window Coefficient Generator	39
Gambar 4.14 Desain Tampilan Hardware Utility dengan Grafik Window Coefficient Generator	40
Gambar 4.15 Desain Tampilan Hardware Utility - Demodulator Coefficient Generator.....	41
Gambar 4.16 Coding File Utama – SenseTool.qml	42
Gambar 4.17 Coding Project Set Item – PrjSetObject.qml.....	43
Gambar 4.18 Coding Project Set Window – PrjSetWindow.qml.....	44
Gambar 4.19 Coding Project Set untuk menyimpan riwayat aktivitas – PrjSetPage.qml	45
Gambar 4.20 Coding Pemanggilan Project Set Window – PrjSetPage.qml	45
Gambar 4.21 Coding Fungsi Hardware Utility – HwUtilPage.qml	46
Gambar 4.22 Tampilan Aplikasi saat Dijalankan.....	47
Gambar 4.23 Tampilan Aplikasi saat Menambahkan Item di Project Set.....	47
Gambar 4.24 Tampilan Aplikasi Sesudah Menambahkan Item di Project Set.....	48

Gambar 4.25 Tampilan Aplikasi Saat Menyunting Item di Project Set	48
Gambar 4.26 Tampilan Aplikasi Scan Arrangement	49
Gambar 4.27 Tampilan Aplikasi saat Menambahkan Item di Scan Arrangement Bagian Tabel Kiri	49
Gambar 4.28 Tampilan Aplikasi saat Menambahkan Item di Scan Arrangement Bagian Tabel Kanan	50
Gambar 4.29 Tampilan Aplikasi Sesudah Menambahkan Item di Scan Arrangement.....	50
Gambar 4.30 Tampilan Aplikasi saat Meyunting Item di Scan Arrangement Bagian Tabel Kiri	51
Gambar 4.31 Tampilan Aplikasi saat Meyunting Item di Scan Arrangement Bagian Tabel Kanan	51
Gambar 4.32 Tampilan Aplikasi Hardware Utility Bagian Window Coefficient Generator.....	52
Gambar 4.33 Tampilan Aplikasi Setelah Memberi Inputan di Window Coefficient Generator.....	52
Gambar 4.34 Tampilan Aplikasi Hardware Utility Bagian Demodulator Coefficient Generator	53
Gambar 4.35 Tampilan Aplikasi Setelah Memberi Inputan di Window Coefficient Generator.....	53
Gambar 4.36 Coding tst_HeaderList.qml	55
Gambar 4.37 Coding tst_SidebarList.qml.....	56
Gambar 4.38 Hasil Pengujian	57

Bab I

Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai industri, termasuk industri elektronik dan perangkat keras. Salah satu inovasi penting dalam bidang ini adalah pengembangan layar LCD (Liquid Crystal Display) yang banyak digunakan pada perangkat mobile seperti smartphone. Dalam upaya untuk meningkatkan efisiensi dan efektivitas produksi, perusahaan sering kali memanfaatkan teknologi pemrograman untuk mengotomatiskan proses yang sebelumnya dilakukan secara manual. Pengalaman PKL di Hongkong Viewtrix Technology Limited memberikan penulis wawasan mendalam tentang penerapan teknologi pemrograman dalam industri ini.

Hongkong Viewtrix adalah perusahaan yang bergerak dalam bidang teknologi display, khususnya pada layar LCD yang banyak digunakan pada perangkat elektronik seperti smartphone. Perusahaan ini berfokus pada pengembangan dan penyediaan solusi display yang berkualitas tinggi untuk memenuhi kebutuhan pasar yang semakin berkembang. Viewtrix memainkan peran penting dalam pengalaman pengguna perangkat elektronik modern.

Perusahaan ini tidak hanya memproduksi layar LCD tetapi juga mengembangkan solusi teknologi yang mendukung fungsionalitas dan performa layar. Salah satu komponen kunci dari produk LCD mereka adalah chip / touch IC yang terdapat di dalamnya. Touch IC ini berfungsi untuk mengatur berbagai aspek dari layar LCD, termasuk deteksi sensor, pengaturan warna, dan komunikasi antar komponen.

Dalam proses produksinya, Viewtrix menghadapi tantangan terkait dengan konfigurasi touch IC yang kompleks. Saat ini, konfigurasi touch IC dilakukan secara manual menggunakan perangkat lunak spreadsheet seperti Microsoft Excel.

Proses manual ini berpotensi menimbulkan berbagai masalah, seperti ketidakakuratan data dan efisiensi yang rendah. Masalah-masalah ini memerlukan perhatian khusus dan solusi yang efektif untuk memastikan kualitas produk akhir tetap tinggi. Untuk memahami lebih lanjut mengenai tantangan yang dihadapi dan bagaimana solusinya dapat diimplementasikan, pada sub bab berikutnya akan dibahas secara mendalam mengenai identifikasi masalah dalam proses konfigurasi touch IC.

1.2 Identifikasi Masalah

Dalam proses konfigurasi touch IC yang digunakan dalam produk Viewtrix, terdapat beberapa masalah signifikan yang mempengaruhi efisiensi dan akurasi. Salah satu masalah utama adalah ketergantungan pada perangkat lunak spreadsheet seperti Microsoft Excel untuk melakukan konfigurasi. Meskipun Excel adalah alat yang fleksibel, penggunaannya dalam konteks ini menimbulkan beberapa kendala:

1. Kemungkinan Kesalahan Manual: Proses konfigurasi yang dilakukan secara manual melalui Excel rentan terhadap kesalahan manusia, seperti kesalahan dalam memasukkan data atau ketidaksesuaian format. Kesalahan ini dapat mengakibatkan konfigurasi touch IC yang tidak akurat, yang pada gilirannya dapat mempengaruhi kinerja dan kualitas layar LCD.
2. Proses Manual yang Memakan Waktu: Proses konfigurasi yang dilakukan secara manual memerlukan waktu yang cukup lama dan dapat menjadi tidak efisien, terutama ketika jumlah konfigurasi yang perlu dilakukan dalam skala besar. Hal ini dapat menghambat produktivitas dan memperlambat waktu produksi.
3. Integrasi Data yang Tidak Optimal: Data konfigurasi yang disimpan dalam format Excel sulit untuk diintegrasikan secara langsung dengan sistem lain yang mungkin digunakan dalam proses produksi. Kurangnya otomatisasi dan integrasi dapat menambah kerumitan dan risiko kesalahan.

Untuk mengatasi masalah-masalah ini secara efektif, penting untuk menetapkan batasan masalah yang jelas agar solusi yang dikembangkan dapat difokuskan pada area-area yang paling kritis. Batasan masalah akan menentukan ruang lingkup dan keterbatasan dari solusi yang akan diusulkan, memastikan bahwa pendekatan yang diambil sesuai dengan kebutuhan dan sumber daya yang tersedia. Pada sub bab berikutnya, akan dibahas batasan masalah yang relevan dengan proyek ini, yang akan membantu dalam merancang solusi yang lebih tepat dan terarah.

1.3 Batasan Masalah

Dalam rangka memastikan proyek PKL berjalan efektif dan sesuai dengan waktu yang tersedia, penting untuk menetapkan batasan-batasan yang jelas pada tugas dan pekerjaan yang diberikan. Batasan ini bertujuan untuk menjaga fokus dan efisiensi kerja, serta mencegah proyek menjadi terlalu luas atau tidak selesai dalam jangka waktu yang ditentukan. Berikut adalah batasan-batasan yang diterapkan selama periode PKL:

1. Ruang Lingkup Aplikasi: Tugas utama adalah mengembangkan aplikasi desktop untuk konfigurasi touch IC. Aplikasi ini harus dapat mengimpor dan mengekspor file Excel sebagai fitur utama. Selain itu, aplikasi juga harus mencakup fitur tambahan yang penting untuk operasional, seperti pengaturan konfigurasi, sistem notifikasi, dukungan multi-bahasa. Namun, pengembangan fitur-fitur yang bersifat tambahan atau kompleks di luar kebutuhan dasar, seperti integrasi dengan sistem eksternal atau fungsionalitas lanjutan yang tidak langsung berhubungan dengan konfigurasi touch IC, tidak termasuk dalam cakupan proyek ini.
2. Waktu dan Sumber Daya: Proyek ini harus diselesaikan dalam periode PKL yang telah ditetapkan, yaitu enam bulan. Oleh karena itu, pengembangan aplikasi harus dilakukan dengan memanfaatkan waktu dan sumber daya

yang ada secara efisien, menghindari penambahan fitur atau fungsi yang dapat memperpanjang waktu pengerjaan.

3. Keterbatasan Teknologi: Mengingat penggunaan bahasa pemrograman Python dan Qt QML yang telah dipelajari selama PKL, aplikasi dikembangkan dengan memanfaatkan pengetahuan dan keterampilan yang ada. Batasan ini berarti bahwa teknologi dan alat yang digunakan harus sesuai dengan keahlian yang sudah ada, tanpa perlu mempelajari teknologi baru yang dapat memperlambat proses pengembangan.

Dengan menetapkan batasan-batasan ini, proyek PKL dapat lebih terfokus dan dapat diselesaikan dalam waktu yang ditentukan, sesuai dengan tujuan dan kebutuhan yang telah ditetapkan oleh perusahaan. Sub bab berikutnya akan membahas rumusan masalah yang dihadapi dalam proyek ini, untuk memberikan pemahaman lebih dalam tentang tantangan-tantangan spesifik yang harus diatasi.

1.4 Rumusan Masalah

Dalam pengembangan aplikasi desktop untuk konfigurasi touch IC di Viewtrix, terdapat beberapa masalah utama perlu dipecahkan. Beberapa masalah ini digabungkan menjadi satu rumusan masalah, yaitu bagaimana cara mengembangkan aplikasi yang efektif dan user-friendly untuk konfigurasi touch IC, dengan fokus pada implementasi fitur impor dan ekspor file secara akurat dan efisien, serta beberapa fitur tambahan, sambil memastikan performa aplikasi yang stabil dan bebas dari bug?

Rumusan masalah ini akan membimbing proses pengembangan aplikasi dan membantu mengatasi tantangan-tantangan yang ada. Sub bab berikutnya akan membahas tujuan dari pengembangan aplikasi ini, yaitu untuk memenuhi kebutuhan spesifik Viewtrix dan mencapai hasil yang diharapkan.

1.5 Tujuan

Kegiatan PKL ini memiliki berbagai tujuan yang penting baik bagi mahasiswa maupun bagi perusahaan.

Tujuan PKL bagi Mahasiswa:

- Mengaplikasikan pengetahuan dari kuliah dalam pengembangan aplikasi desktop yang efektif dan user-friendly untuk konfigurasi touch IC.
- Meningkatkan keterampilan dalam penggunaan bahasa pemrograman Python dan Qt QML serta memahami penerapannya dalam pengembangan aplikasi desktop.
- Memahami tantangan yang muncul dalam pengembangan aplikasi yang stabil, bebas dari bug, dan memiliki performa optimal dalam konteks industri.
- Mengasah kemampuan untuk memecahkan masalah teknis dalam pengembangan aplikasi, seperti memastikan akurasi dan efisiensi fitur.
- Mempersiapkan diri menghadapi tantangan di dunia kerja, termasuk penanganan bug dan peningkatan performa aplikasi.

Tujuan PKL bagi Perusahaan:

- Mengganti proses konfigurasi touch IC yang manual dengan solusi perangkat lunak yang lebih efisien dan akurat.
- Meningkatkan akurasi dan produktivitas dalam proses konfigurasi touch IC.
- Mempercepat pengembangan aplikasi yang diperlukan dan memberikan kontribusi tambahan pada proyek yang sedang berjalan.
- Menyediakan perspektif baru dalam penyelesaian masalah teknis serta ide inovatif untuk meningkatkan performa dan stabilitas sistem.
- Membangun hubungan dengan calon profesional di bidang teknologi, khususnya yang mampu mengembangkan aplikasi yang stabil dan berkinerja optimal.

1.6 Manfaat

Pelaksanaan dan pelaporan kegiatan PKL ini membawa berbagai manfaat, baik bagi perusahaan maupun bagi penulis sendiri. Bagi Viewtrix, aplikasi yang dikembangkan akan meningkatkan efisiensi dan akurasi dalam proses konfigurasi touch IC, menggantikan metode manual yang ada dengan sistem yang lebih terautomasi dan user-friendly.

Bagi penulis, pengalaman ini memberikan kesempatan untuk menerapkan pengetahuan yang diperoleh selama studi, khususnya dalam bahasa pemrograman Python dan Qt QML, ke dalam proyek nyata yang berdampak langsung pada operasional perusahaan. Selain itu, penulis memperoleh keterampilan praktis dalam pengembangan aplikasi desktop dan penyelesaian masalah teknis, yang akan berguna dalam karir profesional di masa depan.

Bab II

Gambaran Umum Perusahaan

2.1 Deskripsi Perusahaan

Hongkong Viewtrix Technology Limited adalah perusahaan teknologi dengan penelitian dan pengembangan teknologi layar sebagai intinya, yang mengkhususkan diri dalam penelitian dan pengembangan, desain, dan penjualan chip driver layar OLED.

Perusahaan mengadopsi model operasi Fabless, dengan produk utama termasuk chip driver layar AMOLED dan *chip backplane driver* layar berbasis silikon *Micro* OLED. Di antaranya, terminal aplikasi hilir *chip driver* layar AMOLED milik perusahaan adalah ponsel pintar (smartphone), sedangkan terminal aplikasi hilir utama *chip backplane driver* layar berbasis silikon *Micro* OLED adalah VR/AR dan perangkat *headwear* cerdas lainnya.

Viewtrix berkantor pusat di Shenzhen, dengan pusat penelitian dan pengembangan, penjualan, dan operasi berlokasi di Shanghai, Beijing, Kunshan, dan Hong Kong. Viewtrix Technology sudah berdiri selama 12 tahun, dengan karyawan yang telah menembus angka hingga 200 lebih, dan telah mengantongi 69 paten resmi.

2.2 Visi

Kami berorientasi pada pelanggan dan berkomitmen untuk selalu menempatkan kebutuhan serta kepuasan mereka sebagai prioritas utama. Melalui pemahaman yang mendalam terhadap keinginan pelanggan, kami berupaya menghadirkan produk dan layanan terbaik. Kami berdedikasi untuk terus berinovasi dalam teknologi, dengan tujuan membuat tampilan layar menjadi lebih menarik dan canggih. Dengan inovasi ini, kami ingin memberikan pengalaman visual yang luar biasa kepada setiap pengguna.

2.3 Misi

Kami percaya bahwa karyawan adalah aset terbesar kami. Oleh karena itu, kami menyediakan berbagai peluang untuk pengembangan karir dan pertumbuhan profesional, menciptakan lingkungan kerja yang mendukung dan memotivasi. Selain itu, kami berkomitmen untuk memberikan manfaat yang berkelanjutan bagi pemegang saham kami. Dengan kinerja yang solid dan strategi bisnis yang efektif, kami berusaha menciptakan nilai tambah bagi mereka. Kami juga berusaha menciptakan nilai positif bagi masyarakat melalui produk kami, yang berkontribusi pada peningkatan kualitas hidup dan pembangunan sosial yang berkelanjutan.

2.4 Struktur Organisasi



Gambar 2.1. Struktur Organisasi Viewtrix

Dalam struktur organisasi di perusahaan Hongkong Viewtrix Technology Limited, di bawah General Manager, terdapat dua bidang utama yaitu Bidang Firmware dan Bidang Hardware. Masing-masing bidang dipimpin oleh seorang

Leader. Penulis ditempatkan di bawah Bidang Firmware sebagai anggota tim dan bekerja langsung di bawah Leader Bidang Firmware.

2.5 Produk dan Layanan

Viewtrix memiliki beberapa produk, di antaranya:

1. Chip penggerak tampilan AMOLED

a) VTDR6110C

VTDR6110C adalah FHD DDIC yang mengintegrasikan berbagai algoritma ke dalam chip untuk meningkatkan kualitas gambar, seperti kompensasi kecerahan sub-piksel (Demura), manajemen warna otonom, kompensasi keseragaman warna (IRC), teknologi rendering sub-piksel (SPR), penyesuaian kontrol kecerahan (BC), penyesuaian suhu warna (TC), dll. Pada saat yang sama, fungsi chip decoding eksternal diintegrasikan ke dalam DDIC untuk memungkinkan modul yang sama mendukung berbagai aplikasi dengan resolusi yang berbeda.

b) VTDR6130

Ini adalah FHD+AMOLED DDIC yang mengintegrasikan fungsi GRAM dan Demura, dan dirancang untuk mengintegrasikan IP eksklusif (Demura/GIP) dari berbagai produsen panel serta fungsi baru yang dibutuhkan oleh pelanggan sistem ponsel, seperti pemrosesan sidik jari di bawah layar (LHBM), tampilan kecerahan panjang (AOD), kompensasi kecerahan sekeliling (EBC), kamera di bawah layar (CUP), penggerak yang ditingkatkan (OD), dll.

c) VTDR7120

VTDR7120 adalah WQHD+AMOLED DDIC yang mengintegrasikan fungsi GRAM dan Demura, menggabungkan berbagai algoritma pada chip untuk meningkatkan kualitas gambar, dan mengusulkan "Arsitektur Penggerak Terdistribusi (DD)" untuk memperpanjang

waktu pengisian sirkuit piksel dan meningkatkan kualitas tampilan panel secara keseluruhan.

d) VTDR5120

Ini adalah DDIC AMOLED kustom pertama dari perusahaan, yang meminimalkan ketinggian sumbu Y IC untuk mengurangi ukuran papan panel dan lebih meningkatkan proporsi layar ponsel.

e) VTDR6115

Ini adalah chip penggerak pertama yang mendukung panel tampilan AMOLED dual data line (DDL) di industri, menyediakan solusi FHD+Ramless AMOLED DDIC sambil menawarkan solusi dengan frame rate tinggi (120Hz), konsumsi daya rendah, dan biaya rendah untuk meningkatkan output IC penggerak.

f) VTDR6130A

VTDR6130A menggabungkan fungsi terkait LTPO berdasarkan VTDR6130, yang memungkinkan peralihan visual tanpa kehilangan dari rentang dinamis frame rate yang luas (1Hz hingga 120Hz).

g) VTDR6126

VTDR6126 adalah generasi pertama 1.5k ramless AMOLED DDIC dari Viewtrix Technology. Chip ini telah meningkatkan algoritma kualitas gambar, seperti algoritma kompensasi crosstalk, serta kompensasi kecerahan sub-piksel (Demura), kompensasi keseragaman warna (IRC), teknologi pewarnaan sub-piksel (SPR), penyesuaian kontrol kecerahan (BC), penyesuaian suhu warna (TC), dll., yang meningkatkan kualitas panel OLED.

h) VTDR6130B

VTDR6130B adalah DDIC WFHD yang mendukung panel OLED LTPO tingkat lanjut dan mengintegrasikan berbagai algoritma terkait LTPO untuk meningkatkan kualitas gambar. Algoritma kualitas gambar lainnya, seperti kompensasi kecerahan sub-piksel (Demura), kompensasi keseragaman warna (IRC), teknologi pewarnaan sub-piksel (SPR), dan penyesuaian kontrol kecerahan (BC), juga diperbarui pada

chip ini untuk memberikan kualitas yang lebih baik dan konsumsi daya yang lebih rendah.

2. **Chip tampilan mikro berbasis silikon OLED**

a) VTOS6202

VIEWTRIX Technology VTOS6202 adalah backplane OLED mikro berbasis silikon dengan ukuran 0,39 inci dan resolusi 1920x1080. Melalui pengaturan piksel khusus yang disesuaikan dan pemrosesan algoritma, mencapai layar OLED berbasis silikon dengan kepadatan piksel tertinggi di dunia yaitu 5644ppi. Desain chip tunggal yang sangat terintegrasi, menggabungkan sirkuit penggerak dan array piksel.

b) VTOS6205

VIEWTRIX Technology VTOS6205 adalah backplane OLED mikro berbasis silikon dengan ukuran 0,49 inci dan resolusi 1920x1080. Desain chip tunggal yang sangat terintegrasi, menggabungkan sirkuit penggerak dan array piksel. Mendukung antarmuka berkecepatan tinggi dan refresh rate 90Hz. Fitur-fitur seperti kompensasi suhu dan kompensasi tegangan sudah terintegrasi. Digunakan dalam kacamata augmented reality (AR) yang ringan dan produk video mobile.

c) VTOS6203

VIEWTRIX Technology VTOS6230 adalah backplane OLED mikro berbasis silikon dengan ukuran 0,71 inci dan resolusi 1920x1080. Desain chip tunggal yang sangat terintegrasi, menggabungkan sirkuit penggerak dan array piksel. Mendukung antarmuka berkecepatan tinggi dan refresh rate 120Hz. Fitur-fitur seperti kompensasi suhu dan kompensasi tegangan sudah terintegrasi. Digunakan dalam kacamata augmented reality (AR) yang ringan dan produk video mobile.

d) VTOS8200

VIEWTRIX Technology VTOS8200 adalah backplane OLED mikro berbasis silikon dengan ukuran 1,03 inci dan resolusi 2560x2560. Desain chip tunggal yang sangat terintegrasi, menggabungkan sirkuit penggerak dan array piksel. Mendukung antarmuka berkecepatan tinggi

dan refresh rate 120Hz. Fitur-fitur seperti kompensasi suhu dan kompensasi tegangan sudah terintegrasi. Digunakan dalam kacamata virtual reality (VR) yang ringan dan produk video mobile.

3. Aplikasi Industri

a. Smartphone

- Chip penggerak tampilan yang dikembangkan dan dipatenkan secara independen oleh VIEWTRIX Technology telah disuplai ke produsen layar domestik terkemuka dan produsen ponsel bermerek.
- Efektif meningkatkan hasil panel
- Tampilan berkualitas tinggi, konsumsi daya rendah
- Mengurangi biaya modul

b. Metaverse Display Screen

VIEWTRIX Technology menyediakan teknologi tampilan yang diterapkan pada sistem VR/AR, dan berkomitmen untuk menyediakan solusi tampilan dengan resolusi tinggi, refresh rate tinggi, kontras tinggi, kepadatan piksel (PPI) tinggi, area tampilan yang moderat, kecerahan tinggi, konsumsi daya rendah, dan integrasi tinggi.

Bab III

Tinjauan Pustaka

3.1 Aplikasi

Menurut Wikipedia, aplikasi adalah subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Beberapa aplikasi yang digabung bersama menjadi suatu paket terkadang disebut sebagai suatu "paket". Contohnya adalah Microsoft Office dan LibreOffice, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. (Wikipedia, 2024).

3.2 Aplikasi Desktop

Aplikasi desktop adalah program komputer yang dirancang untuk dijalankan pada komputer desktop atau laptop. Aplikasi desktop biasanya diinstal pada sistem operasi seperti Microsoft Windows, Mac OS, atau Linux dan dijalankan langsung dari komputer, tanpa memerlukan koneksi internet. Aplikasi desktop dapat melakukan berbagai tugas, seperti pengolahan data, pengeditan foto dan video, pembuatan dokumen, dan sebagainya. (Deriota, 2024).

3.3 Qt

Qt adalah kerangka kerja pengembangan aplikasi lintas platform untuk desktop, embedded, dan mobile. Platform yang didukung meliputi Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS, dan lainnya. Qt bukanlah bahasa pemrograman tersendiri. Qt merupakan kerangka kerja yang ditulis dalam C++. (Qt Wiki, 2024).

Haavard Nord dan Eirik Chambe-Eng (pengembang awal Qt, sekarang menjabat sebagai CEO dan President Trolltech) memulai pengembangan "Qt" pada

tahun 1991, tiga tahun sebelum membentuk perusahaan yang bernama Quasar Technologies yang kemudian berganti nama dua kali, Troll Tech dan akhirnya Trolltech. Toolkit tersebut dinamakan Qt karena aksara Q terlihat cantik karena font Emacs yang digunakan Haavard, sedangkan huruf diilhami dari Xt (X toolkit). (Wikipedia, 2024).

Kerangka kerja Qt awalnya dikembangkan oleh Trolltech, sebuah perusahaan perangkat lunak Norwegia, pada pertengahan 1990-an. Pada tahun 2008, Nokia mengakuisisi Trolltech dan terus mendukung serta mengembangkan Qt. Selanjutnya, Qt diadopsi oleh berbagai industri dan menjadi sumber terbuka (saat ini kerangka kerja Qt dijalankan oleh The Qt Company). (Scythe Studio, 2024).

Selain C++, Qt mendukung bahasa pemrograman lain: Python hingga PySide. Ada juga pengikatan bahasa ke Rust, Java, dan bahasa lainnya. (Scythe Studio, 2024).

3.4 QML

Menurut Qt Programming Course, QML adalah bahasa yang berbasis pada JavaScript dan digunakan untuk membuat antarmuka pengguna suatu aplikasi. Bahasa ini memungkinkan pengguna untuk menggunakan komponen tradisional (tombol, daftar, dll.) dan elemen grafis yang ditambahkan logika. (Qt Programming Course, 2024).

Kode JavaScript sebaris menangani aspek imperatif. Ini terkait dengan Qt Quick, kit pembuatan UI yang awalnya dikembangkan oleh Nokia dalam kerangka kerja Qt. Modul QML yang dikirimkan dengan Qt mencakup blok penyusun grafis primitif (misalnya, Rectangle, Image), komponen pemodelan (misalnya, FolderListModel, XmlListModel), komponen perilaku (misalnya, TapHandler, DragHandler, State, Transition, Animation), dan kontrol yang lebih kompleks (misalnya, Button, Slider, Drawer, Menu). Elemen-elemen ini dapat dikombinasikan untuk membangun komponen yang kompleksitasnya bervariasi

mulai dari tombol dan penggeser sederhana hingga program lengkap yang mendukung internet. (Wikipedia, 2024).

Logika ini dapat ditulis dalam beberapa bahasa, tergantung pada kebutuhan kita. Jika aplikasi tersebut adalah permainan dan membutuhkan daya komputasi, bahasa yang dipilih adalah C/C++. Di sisi lain, jika kinerja aplikasi tidak terlalu penting dapat menggunakan JavaScript. Saat ini, aplikasi kursus akan menggunakan JavaScript dan nantinya akan ditulis dalam C/C++. (Qt Programming Course, 2024).

3.5 Python

Python merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja. (Dicoding, 2024).

Selain itu, bahasa pemrograman ini juga cocok digunakan oleh pemula atau developer yang baru terjun di dunia programming. Sebab, Python adalah bahasa yang mudah dipahami dan dipelajari. (Nadya Nafisah, 2024).

3.6 Pyside

PySide adalah library Python yang memungkinkan pengguna untuk membangun aplikasi GUI dengan memanfaatkan Qt Framework (Framework Desktop GUI yang ditulis dengan bahasa C++). Jadi, dengan PySide, pengguna bisa membuat aplikasi berbasis Qt dengan menggunakan bahasa Python, bukan bahasa C++. (Jago Ngoding, 2024).

Seperti Qt, PySide adalah perangkat lunak bebas. PySide mendukung Linux / X11, macOS, dan Microsoft Windows. Proyek ini juga dapat dikompilasi silang ke sistem tertanam seperti Raspberry Pi, dan perangkat Android. (Wikipedia, 2024).

Pada tahun 2009, Nokia, pemilik Qt toolkit saat itu, menginginkan agar pengikatan Python tersedia di bawah lisensi LGPL. Nokia gagal mencapai kesepakatan dengan Riverbank Computing, pengembang pengikatan Python PyQt. Pada bulan Agustus, Nokia merilis PySide. PySide menyediakan fungsionalitas serupa, tetapi di bawah LGPL. 'Side' dalam bahasa Finlandia berarti pengikatan. (Wikipedia, 2024).

Ada tiga versi utama PySide:

- PySide mendukung Qt 4
- PySide2 mendukung Qt 5
- PySide6 mendukung Qt 6

3.7 YAML

YAML, awalnya dikenal sebagai Yet Another Markup Language, diciptakan pada tahun 2001 tetapi sekarang merupakan singkatan dari YAML Ain't Markup Language. (Dionysia Lemonaki, 2022). YAML adalah format data serialisasi yang dirancang untuk mudah dibaca oleh manusia dan mesin. YAML digunakan untuk menyimpan konfigurasi dan data dalam berbagai aplikasi. Keunggulannya meliputi kesederhanaan, kemampuan untuk merepresentasikan struktur data kompleks seperti daftar dan peta, serta dukungan untuk berbagai bahasa pemrograman. YAML sering digunakan dalam konteks DevOps, CI/CD pipelines, dan aplikasi berbasis kontainer seperti Kubernetes untuk konfigurasi deployment.

Dalam YAML, penekanan diberikan pada indentasi dan pemisahan baris untuk menunjukkan level dan struktur dalam data. Sistem indentasi ini cukup mirip dengan yang digunakan Python. YAML tidak menggunakan simbol seperti kurung

kurawal, tanda kurung siku, atau tag pembuka atau penutup - hanya indentasi. (Dionysia Lemonaki, 2022).

Ditambahkan juga oleh Dionysia Lemonaki (2022), YAML adalah superset resmi JSON meskipun terlihat sangat berbeda dari JSON. YAML dapat melakukan semua hal yang dapat dilakukan JSON dan lebih banyak lagi. File YAML yang valid dapat berisi JSON, dan JSON dapat diubah menjadi YAML. YAML memiliki sintaksis yang paling mudah dibaca manusia, intuitif, dan ringkas untuk mendefinisikan konfigurasi dibandingkan dengan XML dan JSON.

3.8 Touch IC

Touch IC adalah sirkuit yang mengubah sinyal input (jari atau objek yang bersentuhan dengan permukaan layar) menjadi sinyal terukur yang dihitung dan diproses oleh algoritma perangkat lunak pengontrol untuk menentukan posisi sentuhan pada layar. (Anders, 2024).

Touch IC dibuat dengan pin periferil khusus untuk memungkinkan interfacing dengan sensor sentuh dan prosesor host. Umumnya, rangkaian ini dipasang pada FPC (flexible Print Circuit) yang menghubungkan sensor ke prosesor host. (Anders, 2024).

Terlepas dari teknologi sentuh, baik sensor sentuh resistif maupun kapasitif, saat jari atau stylus bersentuhan dengan permukaan layar sentuh, IC melacak dan melaporkan lokasi tempat sentuhan tersebut dirasakan. Misalnya, dengan menggunakan layar sentuh kapasitif biasa, pengontrol IC mengukur perubahan kapasitif di persimpangan antara garis X dan Y dan membandingkannya dengan konfigurasi ambang batas sensitivitas yang diberikan oleh firmware untuk menentukan sentuhan yang sebenarnya. (Anders, 2024).

3.9 LCD

LCD adalah jenis layar yang menggunakan kristal cair untuk menampilkan gambar. Layar ini umum digunakan dalam berbagai perangkat elektronik seperti televisi, monitor komputer, smartphone, jam tangan digital, dan kalkulator. LCD bekerja dengan memanipulasi cahaya yang melewati lapisan kristal cair yang disusun di antara dua lapisan kaca atau plastik.

LCD bekerja berdasarkan prinsip bahwa kristal cair dapat mengubah orientasinya ketika diberi tegangan listrik, mempengaruhi bagaimana cahaya melewati mereka. Struktur dasar LCD terdiri dari beberapa lapisan, termasuk lapisan polarisator, elektroda transparan, lapisan kristal cair, dan sumber cahaya belakang (backlight).

Bab IV

Hasil Dan Pembahasan

4.1 Prosedur Pengerjaan

Proses pengerjaan proyek dimulai dengan tahap pengenalan dan adaptasi terhadap lingkungan kerja serta alat-alat yang digunakan di perusahaan Hongkong Viewtrix Technology Limited. Pada minggu pertama, penulis bersama rekannya diberikan tugas awal untuk membuat aplikasi kalkulator menggunakan Qt QML dan Python sebagai latihan untuk menguasai teknologi baru. Setelah berhasil menyelesaikan tugas tersebut, penulis melanjutkan ke proyek utama yaitu pengembangan aplikasi desktop untuk mengkonfigurasi touch IC. Prosedur pengerjaan meliputi analisis kebutuhan, perancangan arsitektur aplikasi, pengkodean, pengujian, hingga implementasi fitur impor dan ekspor file. Setiap tahap diawasi oleh supervisor yang memberikan arahan dan umpan balik untuk memastikan kesesuaian dengan standar perusahaan. Seluruh proses dilakukan secara kolaboratif dengan tim firmware untuk memastikan aplikasi yang dikembangkan dapat menggantikan proses manual dengan lebih efisien dan akurat.

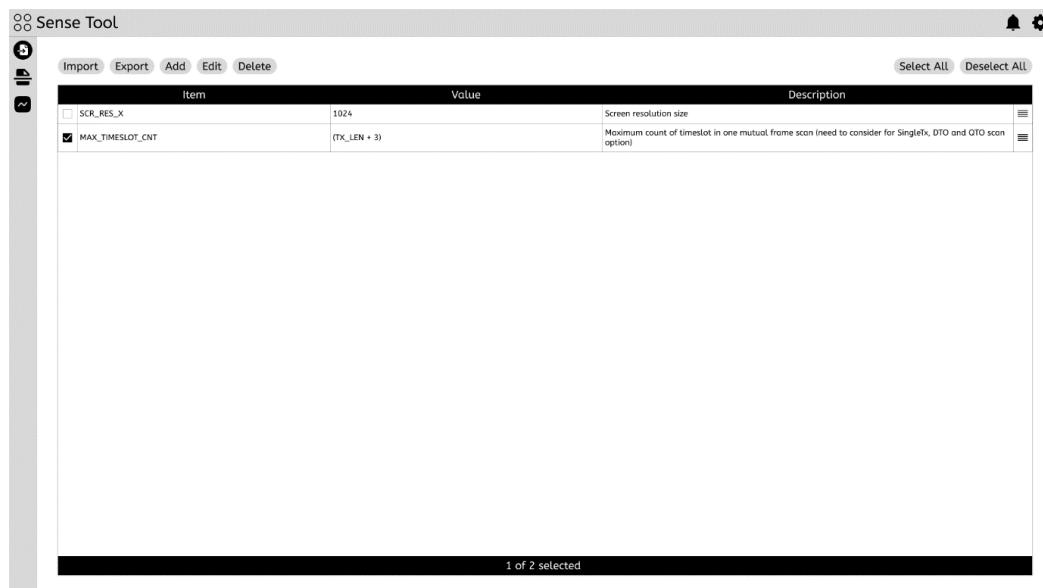
4.2 Analisa Kebutuhan Data

Dalam pengembangan aplikasi desktop untuk konfigurasi touch IC di perusahaan Hongkong Viewtrix Technology Limited, terdapat berbagai data yang diperlukan untuk memastikan aplikasi berfungsi dengan baik dan memenuhi kebutuhan pengguna. Data yang dibutuhkan meliputi spesifikasi teknis touch IC, format dan struktur file Excel yang digunakan dalam proses konfigurasi manual, serta parameter konfigurasi yang harus diatur dalam aplikasi. Selain itu, informasi tentang alur kerja konfigurasi saat ini dan umpan balik sangat penting untuk memahami kebutuhan dan meningkatkan efisiensi proses. Data-data ini dikumpulkan melalui dokumentasi teknis, wawancara dengan staf terkait, dan observasi langsung terhadap proses konfigurasi yang sedang berlangsung. Semua

data tersebut dianalisis dan diintegrasikan ke dalam desain dan pengembangan aplikasi untuk memastikan solusi yang dikembangkan sesuai dengan kebutuhan operasional perusahaan.

4.3 Desain

Desain antarmuka pengguna (UI) dan pengalaman pengguna (UX) untuk aplikasi desktop konfigurasi touch IC di Viewtrix Technology Limited dibuat menggunakan Figma. Desain ini berfokus pada kemudahan penggunaan dan efisiensi, memastikan setiap elemen antarmuka intuitif dan mudah diakses. Tampilan antarmuka dirancang dengan gaya minimalis namun informatif, dengan navigasi yang jelas dan konsisten. Setiap fitur, termasuk impor dan ekspor file Excel, dirancang untuk meminimalisir langkah pengguna dan mengurangi potensi kesalahan. Prototipe interaktif yang dikembangkan di Figma juga memfasilitasi umpan balik dari pengguna selama tahap pengujian, memungkinkan iterasi dan penyempurnaan desain sebelum implementasi akhir.



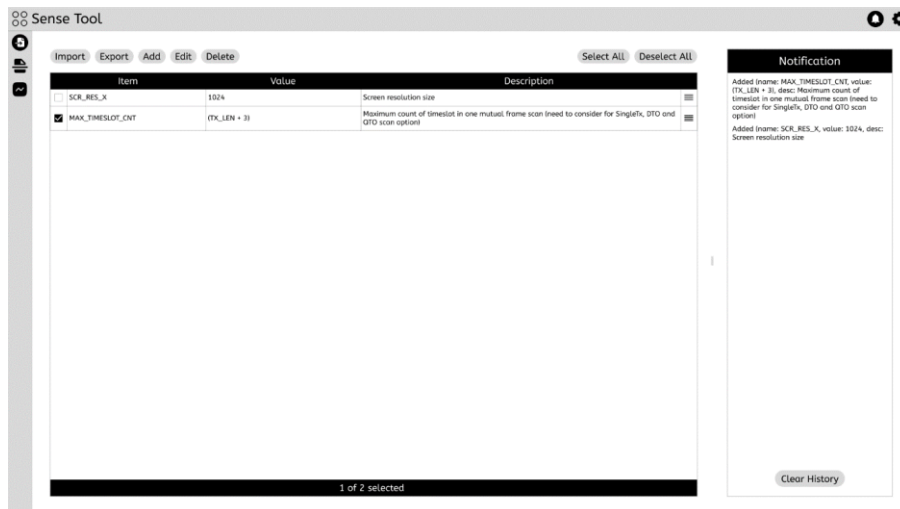
Gambar 4.1 Desain Tampilan Awal Ketika Aplikasi Berjalan

Di bagian paling atas, terdapat header yang menampilkan elemen-elemen penting seperti tombol menu di sisi kiri, lalu nama aplikasi, serta tombol pengaturan dan notifikasi di sisi kanan.

Di sisi kiri tampilan, terdapat sidebar yang memungkinkan pengguna untuk berpindah antar workspace dengan mudah. Terdapat tiga tombol navigasi yang masing-masing mewakili tiga macam workspace. Tombol pertama mengarahkan pengguna ke workspace "Project Set", tombol kedua menuju "Scan Arrangement", dan tombol ketiga membuka "Hardware Utility". Gambar desain mock-up menunjukkan bahwa saat ini workspace yang sedang ditampilkan adalah Project Set.

Project Set adalah workspace utama yang dirancang khusus untuk konfigurasi touch IC di dalam aplikasi ini. Dalam workspace ini, pengguna dapat mengimpor data dari file YAML serta mengekspor data kembali ke format YAML. Data yang diimpor dan diekspor mencakup informasi penting mengenai touch IC, yaitu nama item, nilai (value), dan deskripsi. Data-data touch IC ini merupakan inti dari proses konfigurasi, memungkinkan pengguna untuk secara efisien mengelola dan memodifikasi pengaturan touch IC sesuai kebutuhan.

Bagian tengah layar yang paling luas adalah workspace Project Set, di mana terdapat dua bagian utama dalam area ini, yaitu toolbar dan tabel. Toolbar terletak di bagian atas, berisi tombol-tombol fungsional seperti import, export, add, edit, delete, select all, dan deselect all yang mendukung berbagai operasi yang dapat dilakukan dalam aplikasi. Di bawah toolbar terdapat tabel yang memiliki header dan footer. Tabel ini berisi daftar item yang telah ditambahkan dan dikelola dalam aplikasi, memungkinkan pengguna untuk dengan mudah melihat dan mengelola data yang ada. Header tabel menampilkan tiga kolom: nama item, nilai (value), dan deskripsi. Sementara itu, footer tabel memberikan informasi mengenai jumlah total item serta jumlah item yang diseleksi.



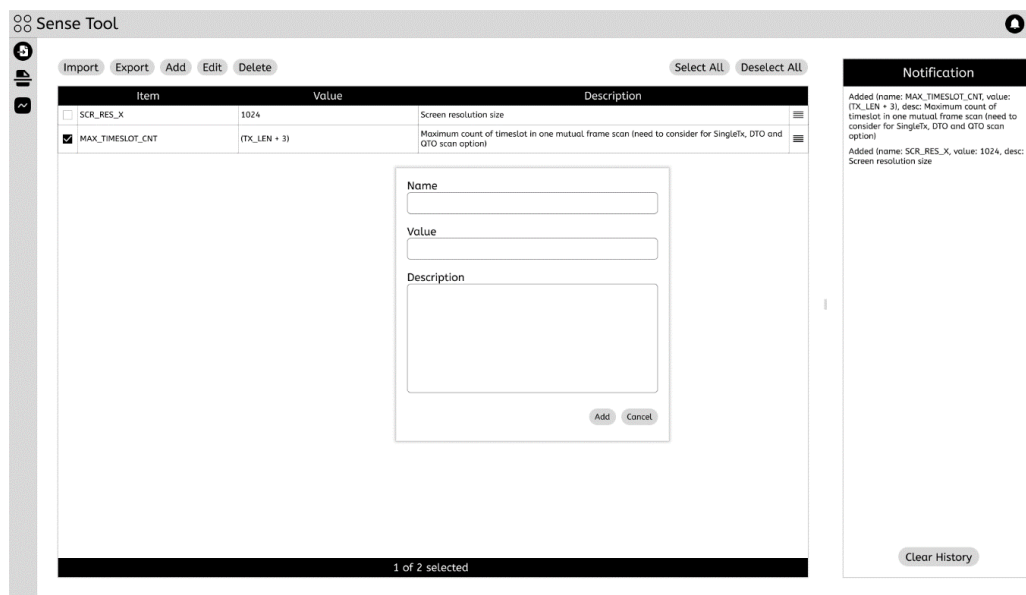
Gambar 4.2 Desain Tampilan Ketika Tombol Notifikasi Aktif

Saat tombol lonceng / notifikasi ditekan, sebuah pop-up akan muncul di sebelah kanan layar. Pop-up ini menampilkan layar notifikasi yang memberikan informasi tentang aktivitas apa saja yang dilakukan pengguna dalam aplikasi, seperti penambahan, penghapusan item. Selain itu, terdapat tombol "Clear History" yang memungkinkan pengguna untuk menghapus riwayat kegiatan yang ditampilkan.



Gambar 4.3 Desain Tampilan Ketika Tombol Pengaturan Aktif

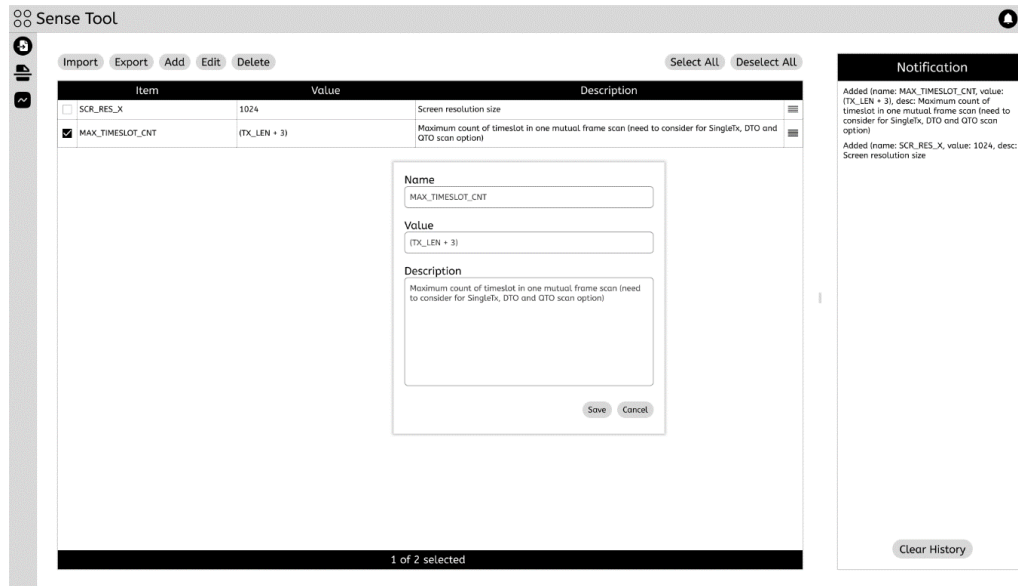
Saat tombol pengaturan ditekan, sebuah pop-up akan muncul di sebelah kanan layar. Di dalam layar pengaturan, pengguna dapat mengatur tema aplikasi sesuai preferensi mereka, memilih bahasa yang diinginkan, dan memeriksa pembaruan aplikasi dengan menggunakan tombol “Check for updates”. Di sini penulis menggunakan switch untuk pengaturan tema dan bahasa dikarenakan hanya memiliki dua pilihan. Switch ini memberikan pengguna cara yang cepat dan intuitif untuk mengaktifkan atau menonaktifkan fitur tertentu, seperti beralih antara mode gelap dan terang. Meskipun saat ini switch hanya digunakan untuk opsi dengan dua pilihan, desain ini masih dapat dikembangkan lebih lanjut di masa depan untuk mencakup lebih banyak fitur dan pengaturan yang lebih kompleks.



Gambar 4.4 Desain Tampilan Ketika Menambahkan Item di Project Set

Pengguna juga dapat melakukan penambahan item satu per satu, dengan menekan tombol "Add". Sebuah pop-up kecil akan muncul di tengah layar. Pop-up ini menyerupai form yang dirancang untuk memungkinkan pengguna menambahkan item baru ke dalam proyek. Form tersebut meminta pengguna untuk mengisi beberapa informasi penting, termasuk nama item, nilai (value), dan

deskripsi item. Pengguna dapat menekan tombol “Add” untuk menambahkan item ke tabel, atau menekan tombol “Cancel” untuk membatalkan.

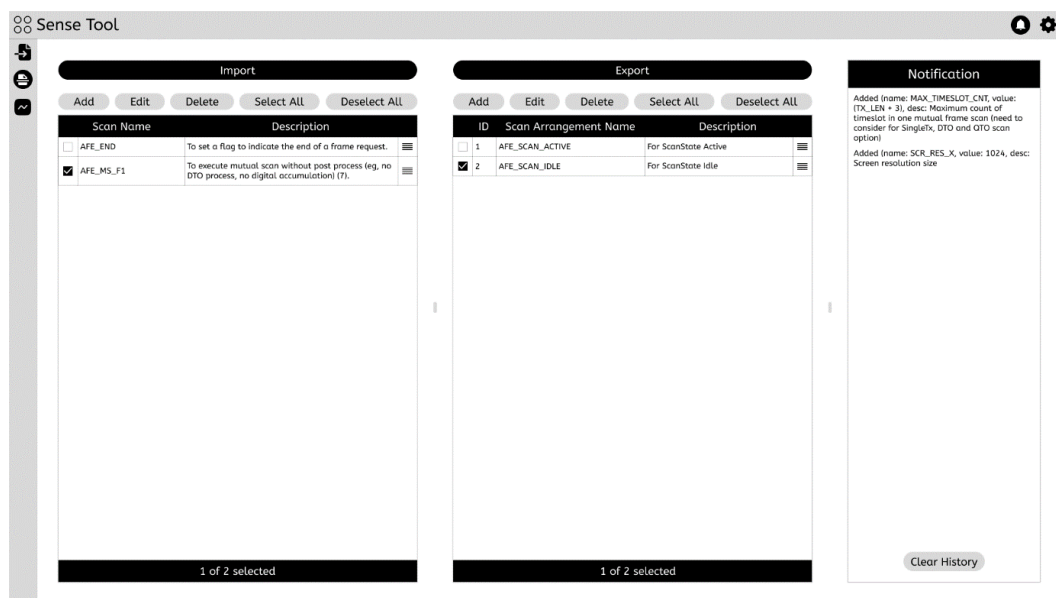


Gambar 4.5 Desain Tampilan Ketika Menyunting Item di Project Set

Gambar desain menunjukkan tampilan aplikasi saat pengguna menekan tombol "Edit" di Project Set. Ketika tombol tersebut ditekan, sebuah pop-up yang mirip dengan pop-up saat menambahkan item baru akan muncul. Namun, pada mode edit ini, pop-up tersebut sudah terisi dengan informasi yang relevan dari item tercentang. Form di dalam pop-up menampilkan nama item, nilai (value), dan deskripsi yang sudah ada, memungkinkan pengguna untuk melakukan perubahan pada detail yang diperlukan. Pengguna dapat menekan tombol “Save” untuk mengonfirmasi perubahan, atau menekan tombol “Cancel” untuk membatalkan.

Terlihat bahwa setiap item dalam tabel memiliki kotak centang (checkbox) di sebelah kiri dan ikon drag di sebelah kanan. Tombol "Edit" hanya akan aktif ketika ada satu item yang dicentang, seperti yang ditunjukkan dalam gambar. Hal ini memastikan bahwa pengguna hanya dapat mengedit satu item pada satu waktu, mencegah kebingungan dan potensi kesalahan dalam pengelolaan data.

Untuk menggunakan tombol "Edit," pengguna harus terlebih dahulu mencentang kotak di sebelah kiri item yang ingin mereka edit. Setelah kotak centang dicentang, item tersebut akan ditandai dengan tanda centang yang jelas, menunjukkan bahwa item tersebut telah dipilih untuk diedit. Pada saat yang sama, tombol "Edit" di toolbar menjadi aktif, memungkinkan pengguna untuk mengkliknya dan membuka pop-up edit.



Gambar 4.6 Desain Tampilan Scan Arrangement

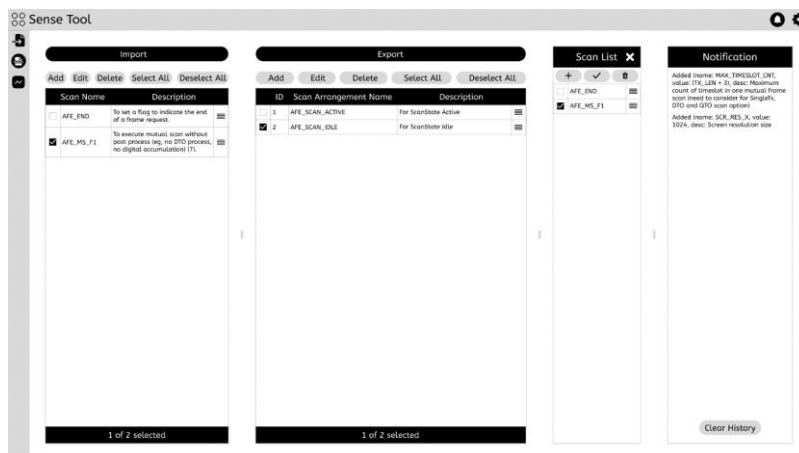
Fungsi dari fitur Scan Arrangement dalam aplikasi ini adalah untuk mengatur dan mengelola urutan serta detail dari berbagai pemindaian (scan) yang dilakukan pada touch IC. Dalam konteks pengaturan touch IC, pemindaian bisa berarti pengecekan atau konfigurasi berbagai parameter dan fungsi dari touch IC tersebut.

Dalam kesimpulannya, Scan Arrangement berfungsi sebagai alat penting untuk mengatur dan mengelola pemindaian yang diperlukan dalam proses konfigurasi. Fitur ini memberikan pengguna kontrol penuh atas urutan dan detail setiap pemindaian, memastikan proses yang efisien, terorganisir, dan sesuai dengan spesifikasi teknis. Dengan kemampuan untuk menyesuaikan dan mengelola pemindaian dengan mudah, pengguna dapat meningkatkan akurasi dan efisiensi

konfigurasi touch IC, yang pada akhirnya berdampak positif pada kualitas produk akhir.

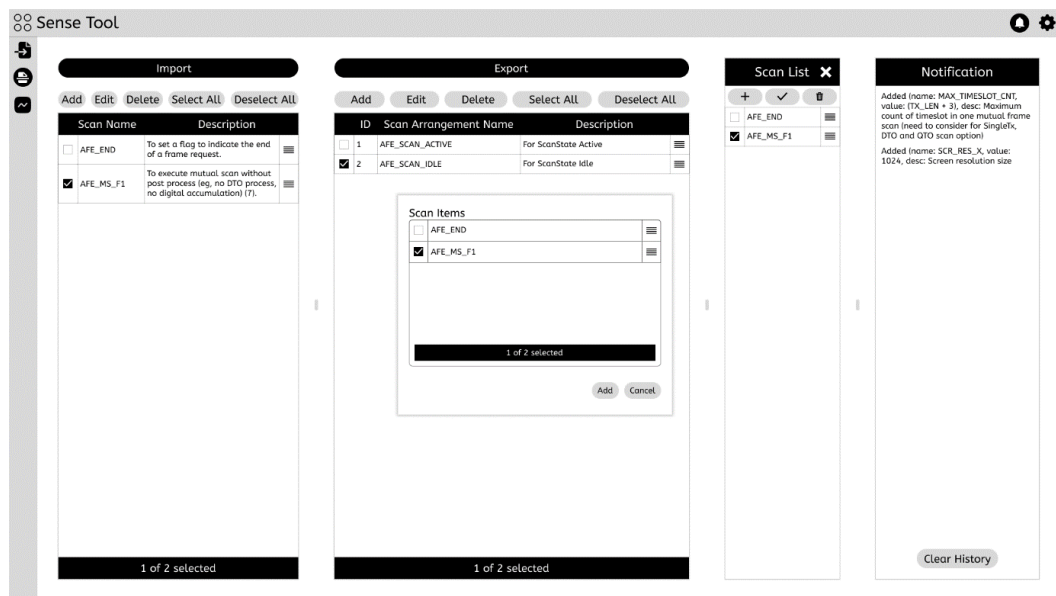
Gambar 4.6 menunjukkan tampilan workspace kedua, yaitu Scan Arrangement, yang memiliki kesamaan dengan workspace Project Set dalam hal pembagian area menjadi dua bagian utama: toolbar dan tabel. Namun, perbedaan utama pada Scan Arrangement adalah bahwa tampilannya dibagi lagi menjadi dua bagian yang terpisah dengan menggunakan split view. Split view membagi area kerja menjadi dua bagian, memungkinkan pengguna dapat menggeser pembatas Split view ke kiri atau kanan, yang akan secara dinamis mengubah ukuran kedua bagian workspace.

Di tabel kanan, pengguna dapat melihat daftar "Scan List" yang dimiliki per item. Dengan menekan salah satu item di tabel ini, akan muncul pop-up baru yang menampilkan Scan List. Scan List ini hanya dapat diisi dengan item yang ada di tabel sebelah kiri. Untuk menambahkan item ke dalam Scan List, pengguna dapat menekan tombol "Add" yang ada di pop-up tersebut. Kemudian, akan muncul pop-up baru yang memungkinkan pengguna memilih item dari tabel sebelah kiri untuk dimasukkan ke dalam Scan List. Selain itu, tombol "Add" dan "Edit" berfungsi serupa dengan yang ada di workspace Project Set, yaitu menampilkan pop-up form untuk mengelola data dengan cara yang konsisten di seluruh workspace.



Gambar 4.7 Desain Tampilan Workspace Scan Arrangement dengan Scan List

Di dalam pop-up ini, terdapat ikon tambah ("+") yang memungkinkan pengguna untuk menambahkan item baru ke dalam daftar scan. Selain itu, terdapat juga ikon sampah yang berfungsi untuk menghapus item dari Scan List. Untuk memudahkan proses seleksi, pop-up Scan List juga dilengkapi dengan ikon seleksi yang memungkinkan pengguna untuk menyeleksi semua item dalam daftar atau membatalkan seleksi yang ada.



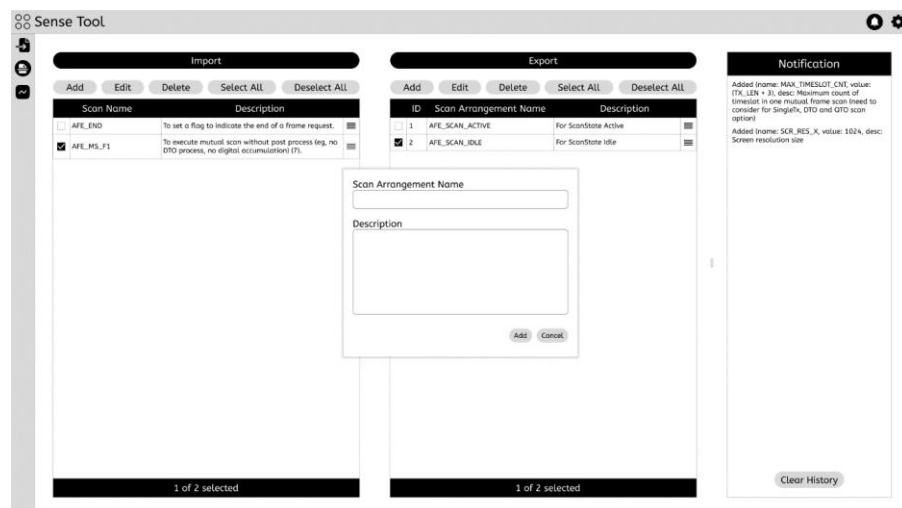
Gambar 4.8 Desain Tampilan Scan Arrangement Ketika Menambahkan Item pada Scan List

Kotak centang / checkbox memungkinkan pengguna untuk memilih satu atau beberapa item dalam daftar scan, memberikan fleksibilitas dalam mengelola dan mengedit beberapa entri secara bersamaan. Ikon drag, yang dapat digunakan dengan cara menekan dan menyeret item, memungkinkan pengguna untuk menggeser dan memindahkan item dalam daftar, menata ulang urutan item sesuai kebutuhan. Kombinasi fitur-fitur dalam scan list dirancang untuk meningkatkan efisiensi dan kenyamanan pengguna dalam mengelola dan mengkonfigurasi daftar pemindaian.



Gambar 4.9 Desain Tampilan Scan Arrangement Ketika Menambahkan Item pada Tabel Scan

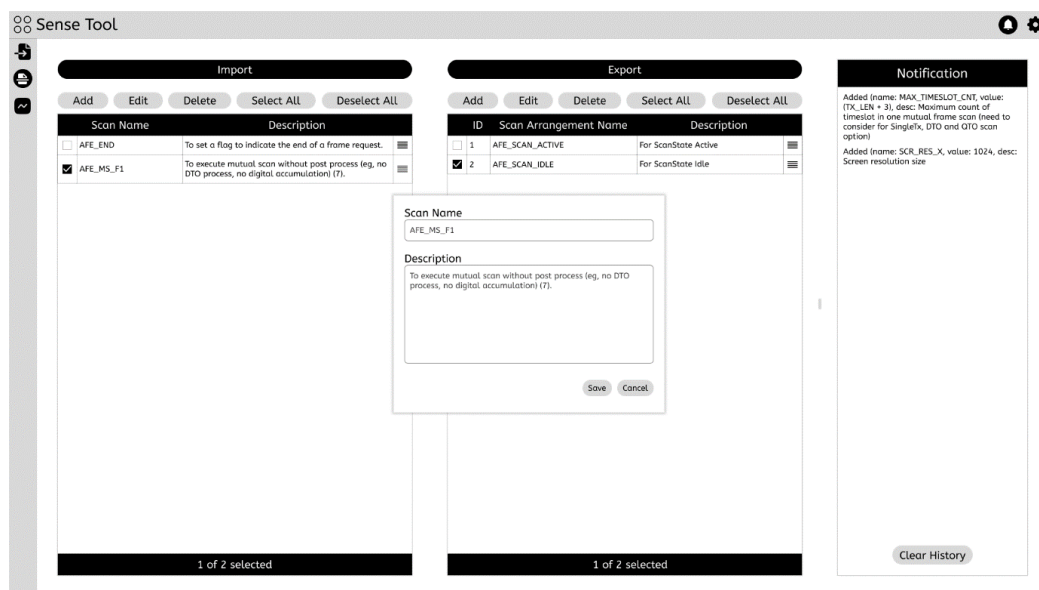
Gambar menunjukkan tampilan Scan Arrangement ketika menambahkan item pada tabel scan. Sebuah pop-up akan muncul di tengah layar, meminta pengguna memasukkan nama item scan dan deskripsi.



Gambar 4.10 Desain Tampilan Scan Arrangement Ketika Menambahkan Item pada Tabel Scan Arrangement

Gambar menunjukkan tampilan Scan Arrangement ketika menambahkan item pada tabel scan arrangement. Sebuah pop-up akan muncul di tengah layar, meminta pengguna memasukkan nama item scan arrangement dan deskripsi.

Hal yang membedakan tabel scan dan scan arrangement adalah fungsinya. Tabel scan di sebelah kiri berfungsi sebagai daftar item utama, sementara tabel scan arrangement di sebelah kanan berfungsi sebagai daftar status touch IC. Sama seperti Project Set, setiap item di tabel ini memiliki kotak centang (checkbox) dan ikon drag di sebelah kiri, memungkinkan pengguna untuk memilih dan menyusun ulang item sesuai kebutuhan. Item yang dipilih dari tabel ini dapat ditambahkan ke dalam daftar pemindaian yang lebih spesifik di tabel kanan melalui pop-up.



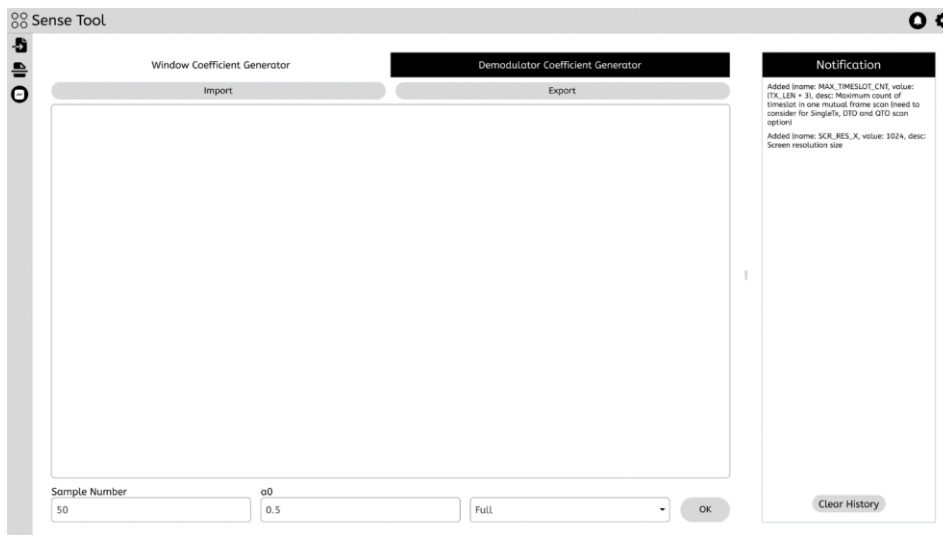
Gambar 4.11 Desain Tampilan Scan Arrangement Ketika Menyunting Item pada Tabel Scan

Ketika pengguna menekan tombol "Edit" pada item yang tercentang di tabel scan, sebuah pop-up akan muncul dengan form yang telah terisi sebelumnya dengan data item tersebut. Form ini menampilkan dua inputan, yaitu nama item scan dan deskripsi.



Gambar 4.12 Desain Tampilan Scan Arrangement Ketika Menyunting Item pada Tabel Scan Arrangement

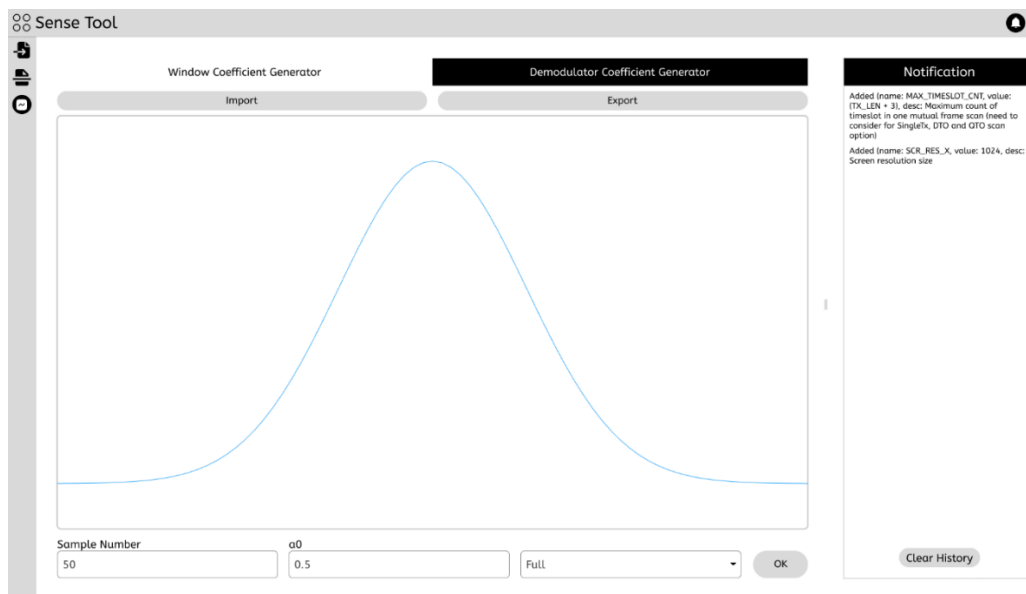
Ketika pengguna menekan tombol "Edit" pada item yang tercentang di tabel scan arrangement, sebuah pop-up akan muncul dengan form yang telah terisi sebelumnya dengan data item tersebut. Form ini menampilkan dua inputan, yaitu nama item scan arrangement dan deskripsi.



Gambar 4.13 Desain Tampilan Hardware Utility - Window Coefficient Generator

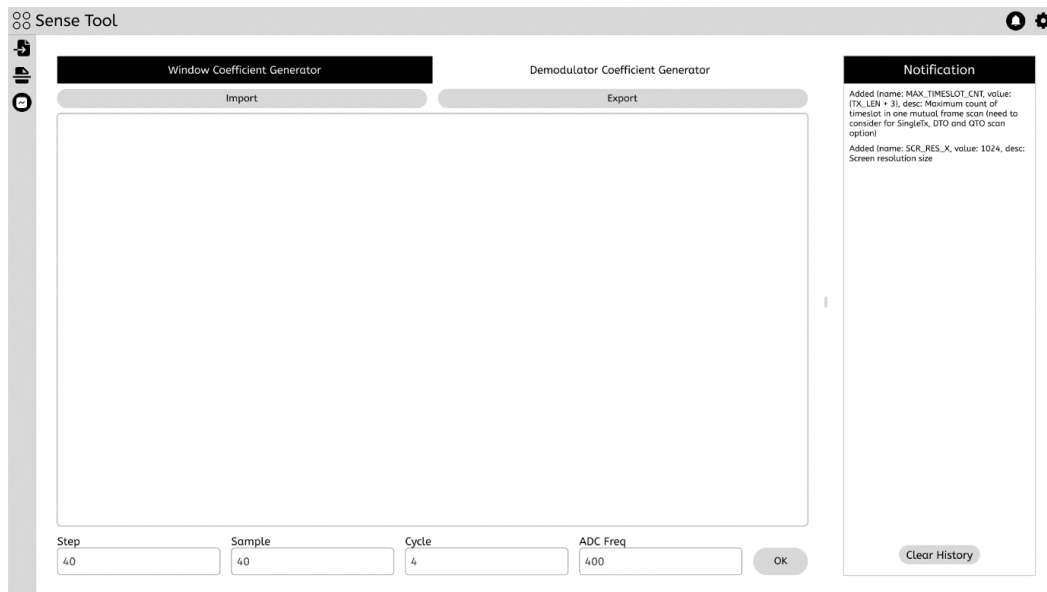
Hardware Utility adalah alat atau modul dalam aplikasi yang dirancang untuk mempermudah interaksi dan pengelolaan perangkat keras (hardware) yang digunakan dalam sistem.

Gambar 4.13 menunjukkan tampilan workspace Hardware Utility yang terbagi menjadi dua tab utama: "Window Coefficient Generator" dan "Demodulator Coefficient Generator". Pada tab Window Coefficient Generator, terdapat dua inputan, yaitu "Sample Number" dan "a0". Pengguna diminta memasukkan nilai-nilai ini untuk menghasilkan grafik diperlukan. Selain itu, terdapat opsi untuk memilih tampilan grafik, yaitu penuh (full) atau setengah (half).



Gambar 4.14 Desain Tampilan Hardware Utility dengan Grafik Window Coefficient Generator

Setelah tombol "OK" ditekan, grafik akan muncul di area tampilan yang menunjukkan hasil perhitungan koefisien jendela berdasarkan inputan yang dimasukkan. Grafik ini membantu pengguna dalam memahami dan menganalisis efek dari pengaturan koefisien terhadap kinerja touch IC, memungkinkan penyesuaian lebih lanjut jika diperlukan.



Gambar 4.15 Desain Tampilan Hardware Utility - Demodulator Coefficient Generator

Gambar desain mock-up menunjukkan tampilan "Demodulator Coefficient Generator". Di tab ini, terdapat empat inputan utama yang perlu diisi oleh pengguna, yaitu "Step," "Sample," "Cycle," dan "ADC Freq". Setelah semua inputan terisi, pengguna dapat menekan tombol "OK" untuk menampilkan grafik. Grafik ini akan muncul di area tampilan dan menunjukkan hasil perhitungan koefisien demodulator berdasarkan nilai yang dimasukkan.

4.4 Pembuatan

Pada bab ini, akan dijelaskan secara detail proses implementasi dan pengembangan aplikasi yang dilakukan selama program PKL di perusahaan Hongkong Viewtrix Technology Limited. Pengembangan aplikasi ini bertujuan untuk mempermudah konfigurasi touch IC yang digunakan dalam perangkat smartphone, menggantikan metode manual yang sebelumnya menggunakan Excel. Dengan menggunakan Qt QML, aplikasi ini dirancang dengan antarmuka yang intuitif dan fungsional. Bab ini akan menguraikan struktur kode, penjelasan

mendetail mengenai setiap bagian dari aplikasi, termasuk workspace utama, toolbars, dan berbagai fungsi lainnya, akan diberikan untuk memberikan gambaran yang komprehensif tentang proses pengembangan yang telah dilakukan.

```
Code - Qt Design Studio
11  Rectangle {
12      id: root
13      width: 1280
14      height: 720
15      Material.theme: popUp.isDarkTheme ? Material.Dark : Material.Light
16  ListModel {
17      id: historyList
18  }
19  ColumnLayout {
20      anchors.fill: parent
21      spacing: 0
22      //header rect
23  Header {
24      id: header
25  }
26      //sidebar + workspace
27  RowLayout {
28      Layout.fillHeight: true
29      spacing: 0
30      //sidebar
31  SideBar {
32      id: sidebar
33  }
34      //workspace and pop-up
35  SplitView {
36      Layout.fillHeight: true
37      Layout.fillWidth: true
38      //content
39  StackLayout {
40      SplitView.fillHeight: true
41      SplitView.fillWidth: true
42      currentIndex: sidebar.index
43      // currentIndex: 2
44      PrjSetPage {}
45      ScanArrPage {}
46      HwUtilPage {}
47  }
48      //pop-up
49  PopUpPage {
50      id: popUp
51      visible: header.index !== -1
52      currentIndex: header.index
53  }
54  }
55  }
56  }
```

Gambar 4.16 Coding File Utama – SenseTool.qml

Gambar di atas adalah file kode/coding utama dari aplikasi, yang merupakan entry point untuk mendefinisikan berbagai komponen UI utama seperti header, sidebar, workspace, pop-up, dan lain-lain. Untuk menjaga agar kode tetap modular dan mudah dikelola, setiap komponen dipisahkan menjadi file QML terpisah. Misalnya, komponen header dibuat dalam file Header.qml, sidebar dalam SideBar.qml, dan sebagainya. Dengan pendekatan ini, jika terjadi kesalahan atau jika ada kebutuhan untuk melakukan penyesuaian, penulis hanya perlu mengakses file yang relevan tanpa harus mencari melalui keseluruhan kode di satu file besar. Hal ini membuat kode lebih mudah dibaca, dipahami, dan dipelihara. Kode ini juga menampilkan penggunaan Material. Theme untuk mendukung tampilan gelap dan terang, serta ListModel untuk menyimpan riwayat yang nantinya dapat digunakan dalam aplikasi.

The image shows a code editor window titled "Code - Qt Design Studio". The code is as follows:

```
1 import QtQuick 2.15
2 import QtQuick.Controls 2.15
3
4 QtObject {
5     id: object
6     property string name
7     property string value
8     property string desc
9
10    function set(object) {
11        name = object?.name ?? ""
12        value = object?.value ?? ""
13        desc = object?.desc ?? ""
14    }
15
16    function get() {
17        let obj = {
18            "name": name,
19            "value": value,
20            "desc": desc
21        }
22        return obj
23    }
24
25    function reset() {
26        set(null)
27    }
28 }
29
```

The editor interface includes a Windows taskbar at the bottom with icons for File Explorer, Google Chrome, VS Code, and other applications. The system tray shows the date and time as 11/08/2024, 20:06.

Gambar 4.17 Coding Project Set Item – PrjSetObject.qml

Kode ini berada dalam file PrjSetObject.qml dan berfungsi untuk mengelola objek touch IC yang memiliki tiga properti: nama item (name), nilai (value), dan deskripsi (description). Dalam komponen QtObject, properti-properti tersebut didefinisikan sebagai string yang dapat diakses dan dimodifikasi sesuai kebutuhan.

Fungsi ‘set(object)’ digunakan untuk menetapkan nilai awal/bawaan saat pengguna ingin menambahkan item baru. Ketika pengguna mengisi parameter kosong tersebut, maka akan menjadi tugas dari fungsi ‘get()’ untuk mengembalikan objek yang sudah terisi tersebut. Selain itu, ada fungsi ‘reset()’ yang berfungsi untuk mengatur ulang properti ke nilai default dengan memanggil fungsi ‘set(null)’. Kode ini digunakan ketika pengguna ingin menambahkan item baru agar tidak terjadi penumpukan data.

```
14 signal create(var object)
15 signal modify(int index, var object)
16 property PrjSetObject object: PrjSetObject{}
17 property int index: -1
18 property bool isEdit: false
19 property HoverHandler cursor: HoverHandler{
20     cursorShape: Qt.IBeamCursor
21 }
22 function manage(i: int, model: var){
23     if(i !== -1){
24         isEdit = true
25         object.set(model)
26     }else {
27         isEdit = false
28         root.object.reset()
29     }
30     index = i
31     root.show()
32 }
```

Gambar 4.18 Coding Project Set Window – PrjSetWindow.qml

Kode ini adalah bagian dari komponen QML yang bertanggung jawab untuk mengelola objek PrjSetObject dalam konteks pembuatan atau pengeditan item dalam aplikasi. Kode ini mengandung sinyal dan properti yang diperlukan untuk melakukan operasi terkait objek.

Sinyal ‘create(var object)’ digunakan untuk mengirim informasi ketika sebuah ‘object’ baru dibuat. Parameter ‘object’ berisi data objek yang akan dikirim ketika sinyal ini dipancarkan.

Sinyal ‘modify(int index, var object)’ digunakan ketika objek yang sudah ada dimodifikasi. Parameter ‘index’ menunjukkan posisi item dalam model yang akan dimodifikasi, dan ‘object’ berisi data objek yang baru setelah dimodifikasi.

Fungsi ‘manage(i: int, model: var)’ digunakan untuk mengatur mode pembuatan atau pengeditan. Jika parameter i bukan -1, maka berarti pengguna sedang mengedit item yang ada, dan ‘object’ akan diatur menggunakan data dari

‘model’. Jika i adalah -1 , berarti pengguna sedang membuat objek baru, dan ‘object’ akan di-reset ke nilai default.

```
7 //project set workspace
8 ShadowRect{
9   id: layout
10  Layout.fillHeight: true
11  Layout.fillWidth: true
12  function history(status, object){
13    var temp = {
14      "status": status,
15      "name": object["name"],
16      "value": object["value"],
17      "desc": object["desc"]
18    }
19    historyList.append(temp)
20  }
```

Gambar 4.19 Coding Project Set untuk menyimpan riwayat aktivitas – PrjSetPage.qml

Fungsi ‘history’ bertugas untuk menyimpan riwayat aktivitas pengguna ke dalam daftar (historyList) yang berfungsi sebagai log dari setiap perubahan yang dilakukan pada item dalam aplikasi.

```
118 PrjSetWindow{
119   id: prjSetWindow
120   onCreate: (object) => {
121     itemList.append(object);
122     layout.history("Added", object)
123   }
124   onModify: (index, object) => {
125     itemList.set(index, object);
126     layout.history("Modified", object)
127   }
128 }
```

Gambar 4.20 Coding Pemanggilan Project Set Window – PrjSetPage.qml

Kode ini adalah bagian dari pemanggilan kelas PrjSetWindow di dalam workspace Project Set yang mengatur pengelolaan objek dalam aplikasi.

Sinyal ‘onCreate’:

- Ketika sinyal ‘create’ dipancarkan dari PrjSetWindow (misalnya, ketika pengguna membuat item baru), fungsi yang terhubung dengan ‘onCreate’ akan dieksekusi.
- Dalam fungsi ini, objek yang baru dibuat (object) akan ditambahkan ke itemList melalui itemList.append(object).

```

48  FileDialog {
49      id: importWinCoef
50      selectedNameFilter.index: 1
51      fileMode: FileDialog.OpenFile
52      nameFilters: ["YAML files (*.yaml *.yml)"]
53      onAccepted: {
54          let importRes = window.manager.logic.importWinCoef(selectedFile)
55          win.clear()
56          win.drawWin(importRes[0])
57          winFields.itemAt(0).text = importRes[1]
58          winFields.itemAt(1).text = importRes[2]
59          winLength.currentIndex = winLength.find(importRes[3])
60          window.manager.addHistory("Imported", selectedFile)
61      }
62  }
63  ToolbarBtn {
64      Layout.fillWidth: true
65      text: "Export"
66      onClicked: {
67          exportWinCoef.open()
68      }
69  }
70  FileDialog {
71      id: exportWinCoef
72      selectedNameFilter.index: 1
73      fileMode: FileDialog.SaveFile
74      nameFilters: ["YAML files (*.yaml *.yml)"]
75      onAccepted: {
76          win.clear()
77          let y = win.createWin(winFields.itemAt(0).text, winFields.itemAt(1).text, winFields.itemAt(2).text)
78          window.manager.logic.exportWinCoef(selectedFile, y, winFields.itemAt(0).text, winFields.itemAt(1).text,
79          winFields.itemAt(2).text)
80          window.manager.addHistory("Exported", selectedFile)
81      }
82  }

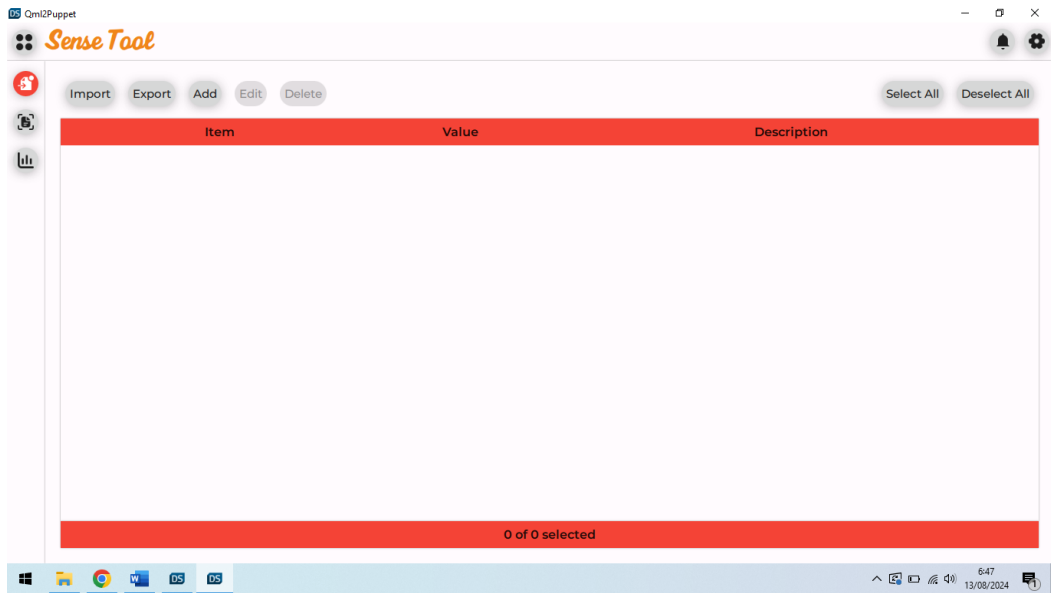
```

Gambar 4.21 Coding Fungsi Hardware Utility – HwUtilPage.qml

Fungsi utama dari kode ini adalah untuk memfasilitasi pengguna dalam memilih dan memuat file YAML, yang berisi koefisien-koefisien penting yang akan digunakan. Koefisien yang menghasilkan grafik mengacu pada nilai-nilai numerik yang memengaruhi bentuk dan karakteristik grafik dari suatu persamaan matematika atau fungsi. Dalam konteks fungsi aljabar atau trigonometri, koefisien adalah faktor yang mengalikan variabel, yang secara langsung menentukan skala, orientasi, dan posisi grafik di bidang koordinat.

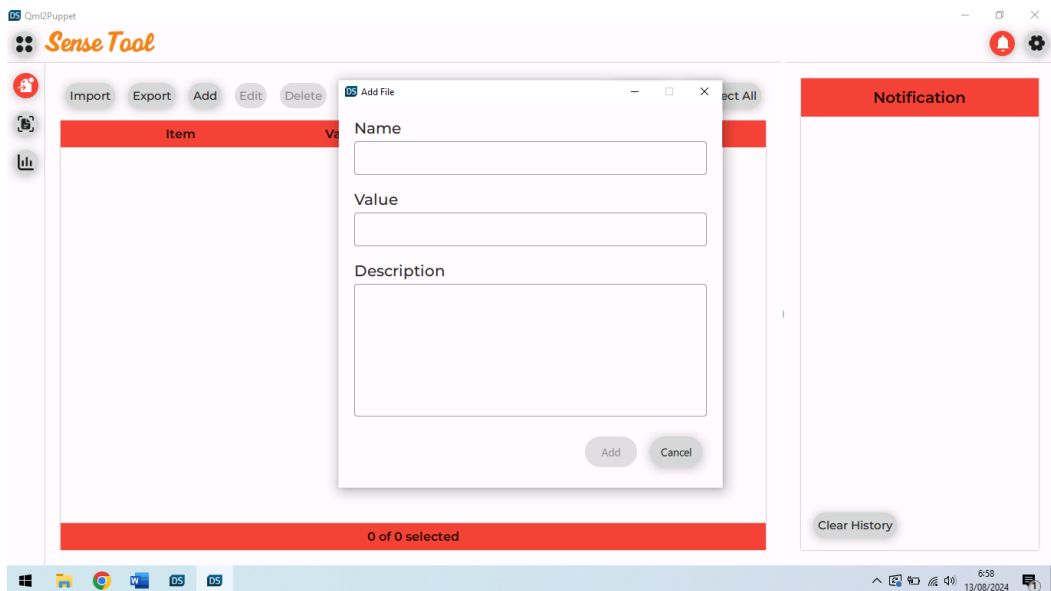
4.5 Hasil Akhir

Pada sub bab ini, akan dibahas hasil akhir dari pengembangan aplikasi yang telah dikerjakan. Secara keseluruhan, aplikasi ini berhasil memberikan solusi yang lebih cepat dan akurat dalam proses konfigurasi touch IC, menggantikan metode manual yang sebelumnya digunakan oleh staff Viewtrix Technology Limited. Implementasi fitur-fitur utama dalam aplikasi ini tidak hanya meningkatkan kecepatan dan akurasi proses konfigurasi, tetapi juga meminimalisir risiko kesalahan manusia yang mungkin terjadi pada metode manual. Proses pengembangan menggunakan Python dan QML menghasilkan antarmuka pengguna yang intuitif dan interaktif.



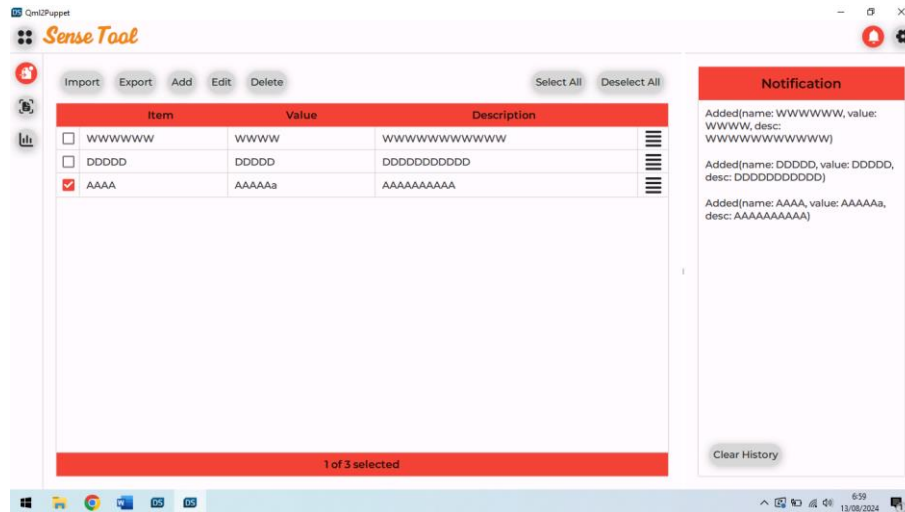
Gambar 4.22 Tampilan Aplikasi saat Dijalankan

Gambar menunjukkan tampilan awal aplikasi saat pertama kali dijalankan. Pada tampilan ini, workspace yang aktif adalah Project Set, yang merupakan workspace utama dalam aplikasi ini.



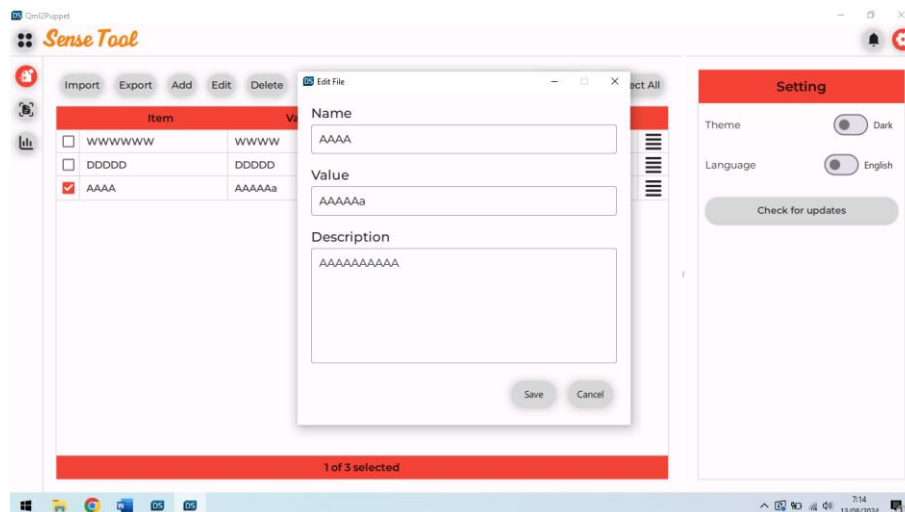
Gambar 4.23 Tampilan Aplikasi saat Menambahkan Item di Project Set

Gambar tersebut menampilkan aplikasi saat pengguna berada dalam proses menambahkan item baru di Project Set.



Gambar 4.24 Tampilan Aplikasi Sesudah Menambahkan Item di Project Set

Gambar tersebut menunjukkan tampilan aplikasi setelah pengguna berhasil menambahkan item baru di Project Set.



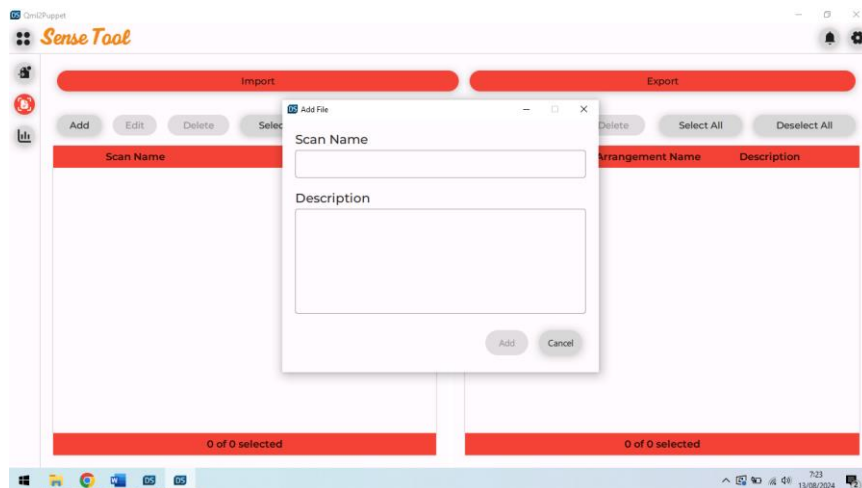
Gambar 4.25 Tampilan Aplikasi Saat Menyunting Item di Project Set

Gambar tersebut menunjukkan saat pengguna berada dalam proses menyunting item di Project Set.



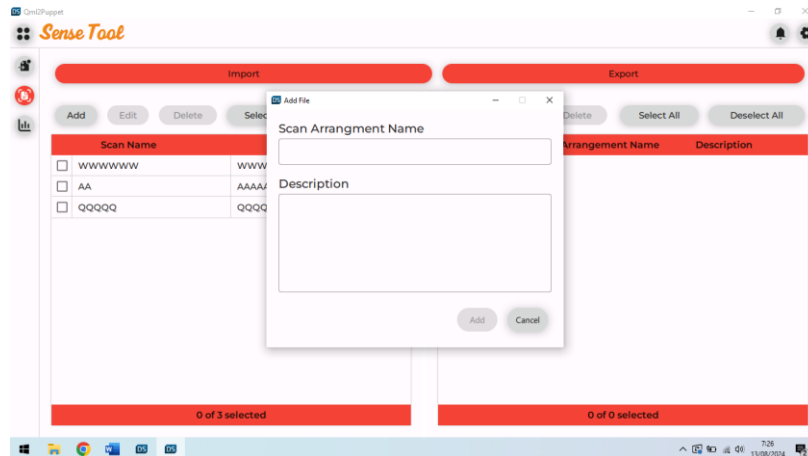
Gambar 4.26 Tampilan Aplikasi Scan Arrangement

Gambar tersebut menunjukkan saat pengguna berada dalam Scan Arrangement.



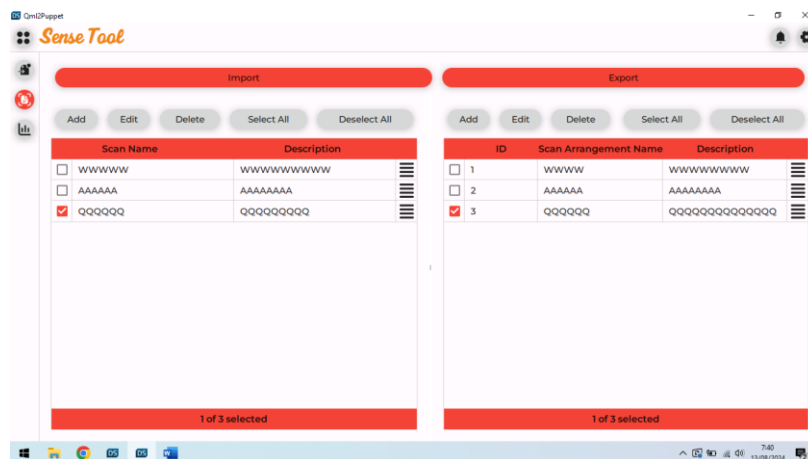
Gambar 4.27 Tampilan Aplikasi saat Menambahkan Item di Scan Arrangement Bagian Tabel Kiri

Gambar tersebut menampilkan aplikasi saat pengguna berada dalam proses menambahkan item baru di Scan Arrangement bagian tabel kiri.

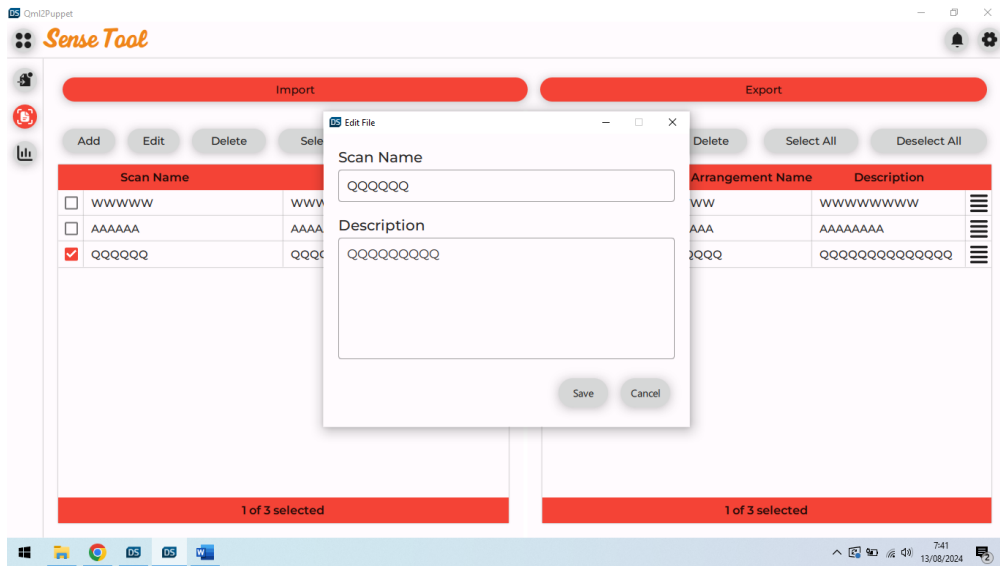


Gambar 4.28 Tampilan Aplikasi saat Menambahkan Item di Scan Arrangement Bagian Tabel Kanan

Gambar tersebut menampilkan aplikasi saat pengguna berada dalam proses menambahkan item baru di Scan Arrangement bagian tabel kanan.

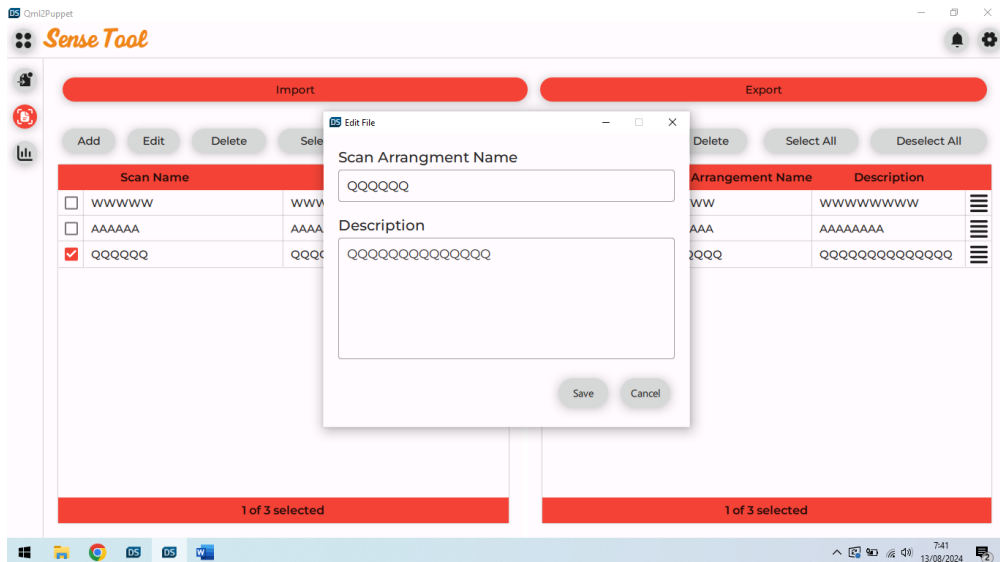


Gambar 4.29 Tampilan Aplikasi Sesudah Menambahkan Item di Scan Arrangement



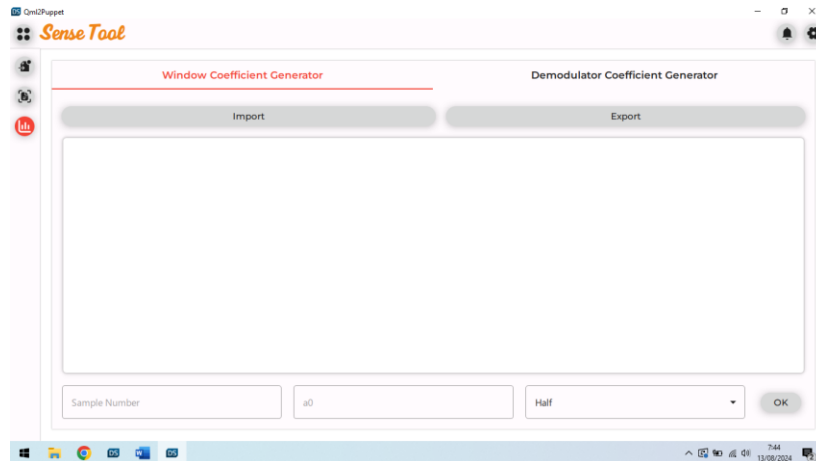
Gambar 4.30 Tampilan Aplikasi saat Meyunting Item di Scan Arrangement Bagian Tabel Kiri

Gambar tersebut menunjukkan saat pengguna berada dalam proses menyunting item di Scan Arrangement bagian tabel kiri.



Gambar 4.31 Tampilan Aplikasi saat Meyunting Item di Scan Arrangement Bagian Tabel Kanan

Gambar tersebut menunjukkan saat pengguna berada dalam proses menyunting item di Scan Arrangement bagian tabel kanan.



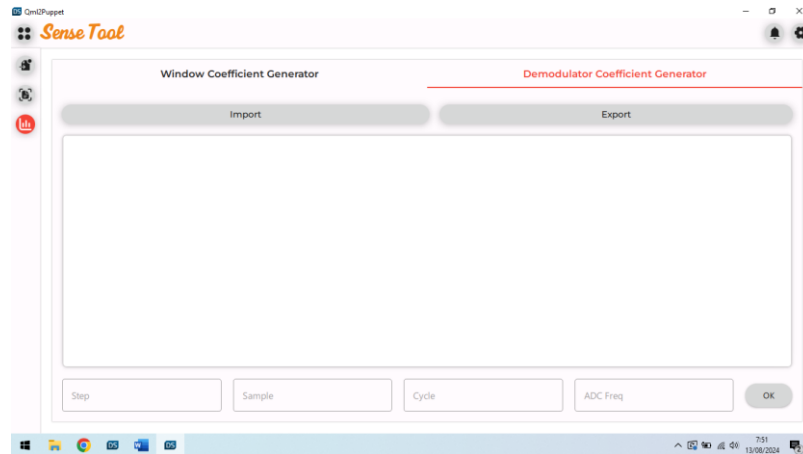
Gambar 4.32 Tampilan Aplikasi Hardware Utility Bagian Window Coefficient Generator

Gambar tersebut menunjukkan saat pengguna berada dalam Hardware Utility ketika pertama kali membuka workspace.



Gambar 4.33 Tampilan Aplikasi Setelah Memberi Inputan di Window Coefficient Generator

Gambar menampilkan antarmuka aplikasi setelah pengguna memberikan input pada tab Window Coefficient Generator.



Gambar 4.34 Tampilan Aplikasi Hardware Utility Bagian Demodulator Coefficient Generator

Gambar menunjukkan saat pengguna berada dalam Hardware Utility ketika menekan tab kedua, yaitu Demodulator Coefficient Generator.



Gambar 4.35 Tampilan Aplikasi Setelah Memberi Inputan di Window Coefficient Generator

Gambar menampilkan antarmuka aplikasi setelah pengguna memberikan input pada tab Demodulator Coefficient Generator.

4.6 Pengujian

Setelah pengembangan aplikasi selesai, pengujian unit perlu dilakukan terlebih dahulu untuk memastikan bahwa tidak ada kesalahan yang terlewatkan oleh pengembang. Pengujian unit ini bertujuan untuk memeriksa bagian-bagian kecil dari aplikasi secara individual, sehingga setiap komponen bekerja dengan benar sesuai yang diharapkan. Dalam konteks pengembangan aplikasi ini, langkah pertama yang dilakukan adalah membuat folder khusus untuk menyimpan file-file uji, yang diberi nama test. Di dalam folder ini, file dengan format `tst_filename.qml` dibuat untuk setiap bagian dari aplikasi yang ingin diuji. Sebagai contoh, terdapat dua file uji yang dibuat, yaitu `tst_HeaderList.qml` dan `tst_Sidebar.qml`.

Pentingnya pengujian unit tidak bisa diabaikan dalam siklus pengembangan perangkat lunak. Pengujian ini membantu dalam mendeteksi bug atau kesalahan pada tahap awal sebelum aplikasi masuk ke tahap pengujian yang lebih besar atau bahkan produksi. Dengan melakukan pengujian unit, pengembang dapat memastikan bahwa setiap modul atau komponen dalam aplikasi berfungsi sebagaimana mestinya secara terpisah sebelum digabungkan dengan modul lainnya. Selain itu, pengujian unit yang baik dapat meningkatkan kepercayaan diri tim pengembang terhadap stabilitas dan kualitas aplikasi yang dibuat.

Namun, perlu diakui bahwa pengujian unit pada aplikasi ini dilakukan secara seadanya, terutama di bagian front-end. Pengujian hanya difokuskan pada elemen-elemen sederhana seperti header dan sidebar, yang meskipun penting, sebenarnya tidak mencakup seluruh kompleksitas aplikasi. Hal ini disebabkan oleh kenyataan bahwa sebagian besar logika utama dan fitur inti dari aplikasi ini berada di sisi back-end. Karena logika dan fungsionalitas yang paling kritis terletak di back-end, pengujian yang dilakukan di front-end terasa kurang relevan dan tidak mampu sepenuhnya mencerminkan performa dan stabilitas aplikasi secara keseluruhan.

Selain itu, kondisi di mana aplikasi belum sepenuhnya selesai juga mempengaruhi tingkat prioritas dalam melakukan pengujian unit. Dengan waktu yang terbatas dan tenggat waktu yang ketat, fokus tim lebih diarahkan pada penyelesaian pengembangan fitur utama daripada melakukan pengujian unit yang mendalam. Dalam situasi ini, pengujian unit pada front-end lebih bersifat pelengkap daripada sebagai alat utama untuk memastikan kualitas aplikasi. Karena itu, pengujian unit lebih difokuskan pada elemen-elemen yang paling terlihat dan interaktif dari user interface, seperti header dan sidebar, untuk memberikan gambaran dasar mengenai fungsionalitas aplikasi, sementara aspek-aspek lain dari pengujian tertunda hingga aplikasi mencapai tahap yang lebih matang.

```
1  import QtQuick
2  import QtQuick.Controls
3  import QtQuick.Layouts
4  import QtTest
5  import "../layouts"
6
7  Item {
8      width: 800
9      height: 600
10
11     HeaderList {
12         id: headerList
13         anchors.fill: parent
14         anchors.centerIn: parent
15     }
16
17     TestCase {
18         name: "MyTestCase"
19         when: windowShown
20
21         function test_checkHeader() {
22             const firstButton = headerList.itemAtIndex(0)
23             const secondButton = headerList.itemAtIndex(1)
24             const indexBeforeClick = headerList.currentIndex
25             mouseClicked(secondButton)
26             const indexAfterClick = headerList.currentIndex
27             verify(indexBeforeClick < indexAfterClick,
28                 "Failed to change popup page by clicking the header button")
29             verify(headerList.currentIndex === indexAfterClick)
30         }
31     }
32 }
33
```

Gambar 4.36 Coding tst_HeaderList.qml

Pada file `tst_HeaderList.qml`, pengujian dilakukan untuk memverifikasi bahwa elemen header pada aplikasi berfungsi dengan baik. Kode ini mendefinisikan sebuah `TestCase` dengan nama `"MyTestCase"` dan menjalankan fungsi `'test_checkHeader'` ketika jendela aplikasi ditampilkan (`windowShown`). Dalam fungsi ini, elemen pertama (`firstButton`) dan kedua (`secondButton`) dari daftar header diambil menggunakan metode `'itemAtIndex'`. Sebelum tombol kedua di-

klik, indeks saat ini (`currentIndex`) dari `'headerList'` dicatat sebagai `'indexBeforeClick'`. Setelah tombol kedua di-klik menggunakan `'mouseClick'`, indeks baru dicatat sebagai `'indexAfterClick'`. Pengujian ini memastikan bahwa setelah tombol kedua di-klik, indeks baru harus lebih besar dari indeks sebelumnya, menunjukkan bahwa halaman pop-up berhasil diubah dengan mengklik tombol header. Fungsi `verify` digunakan untuk memverifikasi perubahan ini, dan jika ada kesalahan, pesan "Failed to change popup page by clicking the header button" akan ditampilkan.

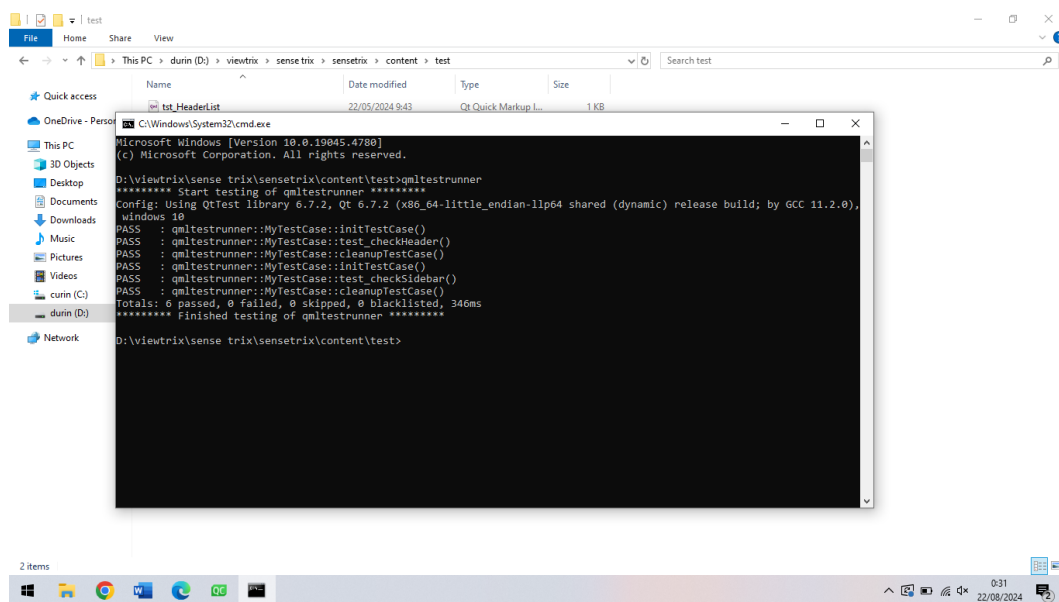
```
1  import QtQuick
2  import QtQuick.Controls
3  import QtQuick.Layouts
4  import QtTest
5  import "../layouts"
6
7  Item {
8      width: 800
9      height: 600
10
11     SideBarList {
12         id: sidebarList
13         anchors.fill: parent
14         anchors.centerIn: parent
15     }
16
17     TestCase {
18         name: "MyTestCase"
19         when: windowShown
20
21         function test_checkSidebar() {
22             const firstButton = sidebarList.itemAtIndex(0)
23             const secondButton = sidebarList.itemAtIndex(1)
24             const thirdButton = sidebarList.itemAtIndex(2)
25             const indexBeforeClick = sidebarList.currentIndex
26             mouseClick(secondButton)
27             const indexAfterClick = sidebarList.currentIndex
28             verify(indexBeforeClick < indexAfterClick,
29                 "Failed to change workspace by clicking on the sidebar button")
30         }
31     }
32 }
33
```

Gambar 4.37 Coding `tst_SidebarList.qml`

Sedangkan pada file `tst_Sidebar.qml`, pengujian dilakukan untuk memeriksa apakah sidebar berfungsi dengan benar. Sama seperti sebelumnya, sebuah `TestCase` didefinisikan dengan nama `"MyTestCase"`. Ketika jendela aplikasi muncul, fungsi `'test_checkSidebar'` dijalankan untuk memeriksa perubahan pada sidebar saat tombol diklik. Tombol pertama (`firstButton`), kedua (`secondButton`), dan ketiga (`thirdButton`) diambil dari daftar sidebar menggunakan `'itemAtIndex'`. Indeks sidebar saat ini sebelum tombol kedua diklik dicatat sebagai `'indexBeforeClick'`. Setelah tombol kedua diklik, indeks baru dicatat sebagai `'indexAfterClick'`, dan

pengujian memverifikasi bahwa indeks baru ini lebih besar dari indeks sebelumnya, yang menunjukkan bahwa workspace berhasil diubah dengan mengklik tombol sidebar. Jika ada kesalahan dalam pengubahan workspace, pesan "Failed to change workspace by clicking on the sidebar button" akan ditampilkan.

Setelah pengaturan file uji selesai, langkah berikutnya adalah masuk ke folder test dan menjalankan command prompt (cmd). Dengan mengetik perintah 'qmltestrunner', aplikasi default yang disebut qmltestrunner akan dijalankan. Aplikasi ini secara otomatis membaca file yang memiliki nama tst_something.qml dan menampilkan hasil apakah tes tersebut berhasil atau tidak. Pada pengujian ini, semua tes berhasil (pass), yang menunjukkan bahwa elemen-elemen aplikasi berfungsi dengan baik sesuai yang diharapkan.



Gambar 4.38 Hasil Pengujian

Hasil pengujian unit yang ditampilkan dalam gambar menunjukkan bahwa seluruh proses pengujian berjalan dengan sukses. Pengujian ini dimulai dengan menampilkan informasi konfigurasi, termasuk versi QTest dan arsitektur sistem yang digunakan, memastikan bahwa lingkungan yang tepat telah disiapkan untuk menjalankan pengujian QML. Setiap test case yang dijalankan—termasuk

inisialisasi (`initTestCase()`), pengujian fungsionalitas header (`test_checkHeader()`), pengujian sidebar (`test_checkSidebar()`), dan pembersihan setelah pengujian (`cleanupTestCase()`)—semuanya berhasil tanpa ada kesalahan, yang ditunjukkan dengan status PASS. Ini berarti elemen-elemen seperti header dan sidebar pada aplikasi berfungsi sesuai harapan, sesuai dengan skenario uji yang dirancang. Selain itu, tidak ada test case yang dilewati atau gagal, dan waktu total yang diperlukan untuk menyelesaikan pengujian ini relatif singkat, yaitu sekitar 346ms. Keseluruhan hasil ini mengindikasikan bahwa aplikasi telah mencapai tingkat fungsionalitas yang memadai pada bagian-bagian yang diuji, meskipun pengujian ini dilakukan secara terbatas pada elemen-elemen tertentu di front-end.

Bab V

Kesimpulan dan Saran

Kesimpulan dari pengembangan aplikasi Sense Tool berbasis QML menunjukkan hasil yang memuaskan dengan sebagian besar fitur utama, seperti Project Set dan Hardware Utility, berhasil diimplementasikan. Aplikasi ini efektif menggantikan proses manual konfigurasi chip LCD, memberikan fleksibilitas dalam pengaturan proyek, serta meningkatkan efisiensi kerja.

Namun, beberapa bug masih ditemukan dan perlu diperbaiki untuk memastikan performa yang optimal. Fitur Scan Arrangement, meski belum selesai karena kesulitan teknis dan keterbatasan waktu, memiliki potensi besar untuk meningkatkan fungsionalitas aplikasi di masa depan. Pengembangan lebih lanjut juga diperlukan dalam hal pengujian, terutama untuk skenario kompleks.

Kombinasi Python dan QML memungkinkan aplikasi ini menjadi lebih interaktif dan user-friendly, meski adaptasi terhadap QML memerlukan waktu karena perbedaan sintaks. Umpan balik pengguna sangat penting dalam penyempurnaan fitur dan pengembangan aplikasi lebih lanjut. Aplikasi Sense Tool siap digunakan dengan beberapa perbaikan dan memiliki potensi untuk terus berkembang sesuai dengan kebutuhan industri teknologi layar.

DAFTAR PUSTAKA

- Anders. (2024, Agustus 12). Touch IC - Integrated Circuit. Diakses pada 12 Agustus 2024, dari <https://www.andersdx.com/touch-ic-integrated-circuit/>
- Deriota. (2024, Agustus 12). Berkenalan dengan Aplikasi Desktop. Diakses pada 12 Agustus 2024, dari <https://deriota.com/news/read/1155/berkenalan-dengan-aplikasi-desktop.html>
- Dicoding Blog. (2024, Agustus 12). Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya. Diakses pada 12 Agustus 2024, dari <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>
- Dionysia Lemonaki. (2024, Agustus 12). What is YAML? The YML File Format. Diakses pada 12 Agustus 2024, dari <https://www.freecodecamp.org/news/what-is-yaml-the-yml-file-format/>
- GitBook. (2024, Agustus 12). Introduction to Qt/QML. Diakses pada 12 Agustus 2024, dari <https://mimecar.gitbook.io/qt-course/en/chapter-04-qml/chapter-04-s01>
- Wikipedia. (2024, Agustus 12). Aplikasi. Diakses pada 12 Agustus 2024, dari <https://id.wikipedia.org/wiki/Aplikasi>
- Wikipedia. (2024, Agustus 12). Qt. Diakses pada 12 Agustus 2024, dari <https://id.wikipedia.org/wiki/Qt>