

PERANCANGAN SISTEM *MONITORING & CONTROLLING* MENGGUNAKAN *SMARTPHONE* PADA STASIUN KERJA *FILLING, SEPARATING, DAN PROCESSING SIMULATOR BOTTLING PLANT* DENGAN METODE *V-MODEL*

MONITORING & CONTROLLING SYSTEM DESIGN USING SMARTPHONE AT WORK STATIONS FILLING, SEPARATING, AND PROCESSING SIMULATOR BOTTLING PLANT WITH V-MODEL

Muhamad Hilmy Abdullah¹, Haris Rachmat², Denny Sukma Eka Atmaja³

^{1,2,3} Universitas Telkom, Bandung

hilmyabdullah@student.telkomuniversity.ac.id¹, harisrachmat@telkomuniversity.ac.id²,
dennysukma@telkomuniversity.ac.id³

Abstrak

Pandemi Covid-19 berdampak terhadap berbagai bidang yang mengharuskan untuk melakukan social distancing, pada bidang industri manufaktur pandemi tersebut memiliki pengaruh terhadap pembatasan karyawan atau staf perusahaan yang bekerja, mengacu kepada instruksi pemerintah maka sebagian besar karyawan bekerja secara *Work From Home* (WFH), sehingga bagi karyawan atau staf tidak selalu dapat melakukan pekerjaannya secara langsung atau *Work From Office* (WFO). Untuk mengatasi hal tersebut, diperlukan suatu sistem yang dapat membantu untuk melakukan proses *monitoring* dan *controlling* pada suatu stasiun kerja, sehingga pada penelitian tugas akhir ini dilakukan pendekatan terhadap suatu alat yang merupakan alat simulator *bottling plant* yang dimana alat tersebut dapat merepresentasikan suatu proses produksi pada perusahaan minuman kemasan botol. Simulator *bottling plant* adalah alat yang dapat mensimulasikan proses pengisian cairan kepada botol hingga proses pengemasan botol, terdapat beberapa stasiun yang mewakili masing-masing proses seperti proses pengisian pada *filling station*, proses pemisahan botol pada *separating station*, proses pemasangan tutup botol pada *processing station*, proses pendistribusian box pada *distribution box station*, hingga proses pengemasan pada *pick & place station*. Untuk menyelesaikan masalah tersebut, dilakukan perancangan sistem *monitoring* dan *controlling* yang menggunakan *smartphone*, dimana terdapat aplikasi android yang terhubung dengan *Programmable Logic Controller* (PLC) agar dapat digunakan untuk *monitoring* dan *controlling* secara *real-time*. Sistem tersebut dirancang menggunakan metode *v-model*, dimana metode *v-model* melakukan pengembangan pada sistem yang setiap langkahnya dilakukan pengujian untuk pengembangan yang sesuai, dimana tahapan selanjutnya dapat dimulai ketika tahapan sebelumnya telah selesai. Berdasarkan hasil simulasi, didapatkan sistem berfungsi dengan baik dimana fungsi pembacaan (*read*) pada PLC dan pengubahan (*write*) ke PLC dapat dilakukan secara *real-time* sesuai dengan yang diharapkan.

Kata kunci: *Real-time Monitoring & Controlling, Programmable Logic Controller (PLC), V-Model*

Abstract

The Covid-19 pandemic has an impact on various fields that require social distancing, in the manufacturing industry the pandemic has an effect on limiting employees or company staff who work, referring to government instructions, most employees Work From Home (WFH), so that for employees or staff are not always able to do their work directly or Work From Office (WFO). To overcome this, we need a system that can help to carry out the process of monitoring and controlling at a work station, so that in this final project research an approach is carried out on a tool which is a bottling plant simulator tool where the tool can represent a production process at the

company. bottled drinks. The bottling plant simulator is a tool that can simulate the process of filling liquid into bottles to the bottle packaging process, there are several stations representing each process such as the filling process at the filling station, the bottle separation process at the separating station, the process of installing bottle caps at the processing station, the process distribution box at the distribution box station, to the packaging process at the pick & place station. To solve this problem, a monitoring and controlling system is designed using a smartphone, where there is an android application that is connected to the Programmable Logic Controller (PLC) so that it can be used for monitoring and controlling in real-time. The system is designed using the v-model method, where the v-model method develops the system in which each step is tested for appropriate development, where the next stage can be started when the previous stage has been completed. Based on the simulation results, it is found that the system is functioning properly where the read function on the PLC and the conversion (write) to the PLC can be carried out in real-time as expected.

Keywords: *Real-time Monitoring & Controlling, Programmable Logic Controller (PLC), V-Model*

1. PENDAHULUAN

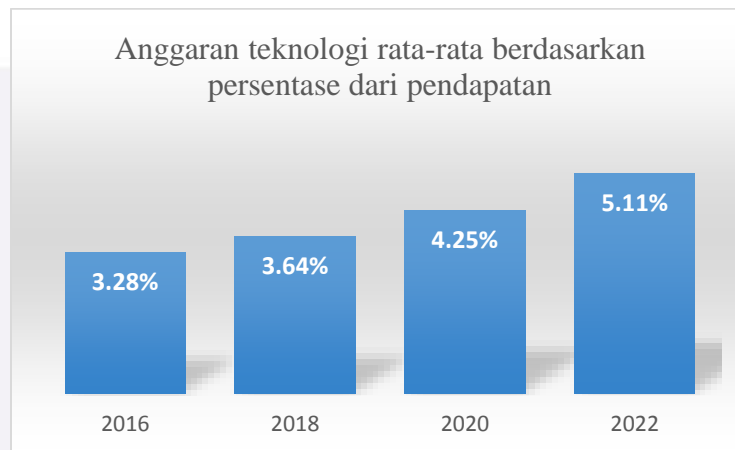
Memasuki tahun 2020, dunia berada dalam masa pandemi Covid-19 yang menyebar dan membahayakan kesehatan setiap orang, pandemi tersebut berdampak terhadap semua bidang, salah satunya bidang industri manufaktur, di Indonesia tersendiri pandemi ini berdampak cukup signifikan terhadap dunia industri yang mengharuskan karyawannya bekerja secara *Work From Home* (WFH). Berdasarkan peraturan yang diterbitkan oleh Pemerintah Indonesia tersendiri yang diturunkan melalui Instruksi Menteri Dalam Negeri Nomor 15 Tahun 2021, untuk sektor industri esensial diberlakukan 50% maksimal staf yang bekerja secara *Work From Office* (WFO) [1], *operator* yang menangani mesin produksi perlu untuk bekerja langsung pada area produksi dengan protokol kesehatan yang ketat, maka dari itu untuk beberapa karyawan *middle level* seperti *supervisor* ataupun *manager* hingga *top level management* yang dimana berperan sebagai pengambil keputusan tidak dapat selalu bekerja secara WFO, sehingga tidak selalu dapat melakukan pekerjaannya secara langsung. Untuk membantu karyawan yang sedang dalam masa WFH sedangkan diharuskan menganalisis jalannya proses produksi, diperlukan suatu sistem yang dapat membantu untuk melakukan proses *monitoring* dan *controlling* pada suatu stasiun kerja. Sehingga pada penelitian tugas akhir ini dilakukan pendekatan terhadap suatu alat yang merupakan alat simulator *bottling plant* yang dimana alat tersebut dapat merepresentasikan suatu proses produksi pada perusahaan minuman kemasan.

Simulator *bottling plant* merupakan simulator alat yang mensimulasikan proses pengisian cairan kepada botol hingga proses pengemasannya, pada alat tersebut terdapat beberapa stasiun kerja yang mewakili masing-masing proses, berawal dari stasiun kerja *filling* yang terdapat proses pengisian cairan kepada botol, lalu stasiun kerja *separating* yang melakukan proses pemisahan botol, stasiun kerja *processing* yang melakukan proses pemasangan tutup botol, stasiun kerja *distribution box* yang melakukan pendistribusian *box* untuk pengemasan botol, hingga stasiun kerja *pick & place* yang melakukan proses pengemasan botol kedalam *box*. Penelitian ini merupakan penelitian yang berintegrasi, dimana pada penelitian yang dilakukan akan berfokus pada stasiun kerja *filling*, *separating*, dan *processing*.

Saat ini dunia industri berada dalam revolusi industri keempat (*Industry 4.0*) yang didorong oleh perkembangan Teknologi Informasi dan Komunikasi atau *Information and Communication Technologies* (ICT). Dasar dari teknologi tersebut adalah otomatisasi cerdas atau *cyber-physical systems* dengan kontrol yang terdesentralisasi dan konektivitas canggih (fungsi IoT) [2]. Konsep industri 4.0 dibuat agar perusahaan dapat menjalankan proses manufaktur yang fleksibel dan terintegrasi untuk menganalisis data secara *real-time* dan melakukan improvisasi dalam strategi ataupun sistem di perusahaannya [3]. *Real-time system* merupakan bagian dari *cyber-physical system*, *real-time system* terbagi dalam tiga cluster yakni *operator cluster*, *computational cluster* atau *real-time system*, dan *controlled cluster* atau *controlled object*. Berdasarkan cluster tersebut, maka *environment* dari *real-time computer system* terdiri dari *operator* dan *controlled object* dimana antara *operator* dengan *real-time system* adalah *man-machine interface* yang menghubungkan antara keduanya, sedangkan yang menjadi penghubung antara *real-time system* dengan *controlled*

object merupakan *instrumentation interface*. *Man-machine interface* yang dimaksud disini terdiri dari perangkat *input* dan perangkat *output* yang terhubung dengan *operator*. Pada *instrumentation interface* terdiri dari sensor dan aktuator yang mengubah sinyal fisik kedalam bentuk digital ataupun sebaliknya [4].

Selama pandemi Covid-19 ini perusahaan industri perlu meningkatkan efisiensi dalam penggunaan teknologi otomasi, sehingga perusahaan tersebut dapat secara selektif dalam mengotomatiskan langkah kerja individu untuk memastikan *social distancing* dalam kegiatan produksi. Teknologi ini dapat meningkatkan produktivitas dan daya saing perusahaan, disamping itu dalam waktu dekat tampaknya manusia dan robot sebagai salah satu teknologi otomasi akan berdampingan dalam segala bidang terutama sektor industri [5]. Selain itu, sistem pengontrolan *remote* untuk mesin dievaluasi sebagai model aplikasi teknologi pintar dan otomatisasi yang sesuai saat keadaan Covid-19 saat ini. Penggunaan sistem pengontrolan *remote* atau pengendali mesin jarak jauh ini digunakan untuk mendukung transisi cepat ke tenaga kerja yang terdistribusi, pengembangan infrastruktur teknologi saat ini mengarah kepada penggunaan alat yang secara virtual dan aman bagi penggunaannya untuk berkolaborasi pada perubahan kondisi ekstrim seperti pandemi Covid-19 [6]. Dengan kondisi seperti itu, perusahaan diharuskan untuk mencari peluang untuk mengotomatiskan proses dan mengurangi keterlibatan manusia, meskipun telah banyak perusahaan yang meningkatkan efisiensi dan investasi yang signifikan dalam otomatisasi, kondisi pandemi mempercepat peningkatan investasi tersebut [7]. Seperti yang dapat terlihat pada Gambar 1. *average technology budget as a percentage of revenue* dibawah ini yang menunjukkan peningkatan investasi perusahaan pada teknologi otomasi.



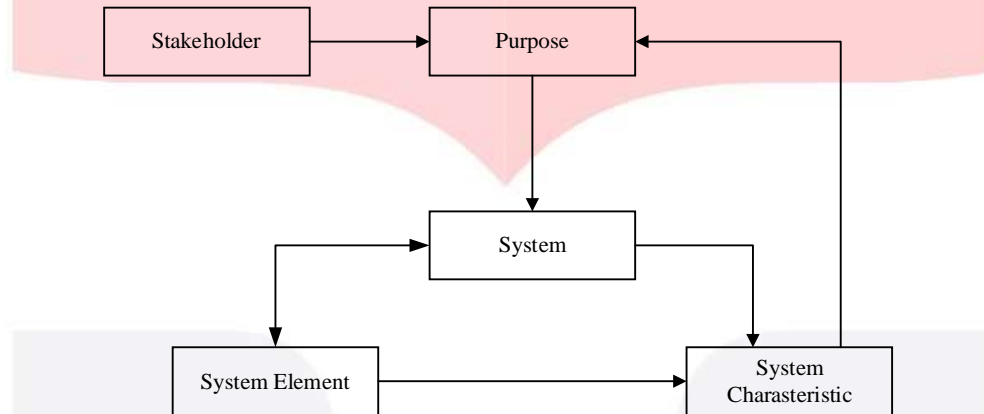
Gambar 1 *Average Technology Budget as A Percentage of Revenue*

Untuk dapat melakukan perancangan sistem *monitoring* dan *controlling* menggunakan *smartphone* pada stasiun kerja *filling*, *separating*, dan *processing* pada penelitian ini digunakan metode *v-model*. Metode *v-model* merupakan pendekatan untuk pengembangan sistem yang kompleks dimana pada setiap fase pengembangannya perlu dilakukan tes validasi sebelum lanjut ke tahapan berikutnya [8]. Pada metode *v-model*, tahap awal yang dilakukan adalah melakukan analisis kebutuhan terhadap data yang telah dikumpulkan pada stasiun kerja *filling*, *separating*, dan *processing*, data-data tersebut dapat berupa waktu proses ataupun skenario proses pada stasiun kerja tersebut. Setelah berhasil melakukan analisis terhadap kebutuhannya, selanjutnya dapat ditentukan area yang akan dikembangkan. Berdasarkan penentuan area yang dikembangkan, dapat dibuat perancangan sistem secara garis besar untuk mencapai tujuan tersebut. Sistem yang telah dirancang menjadi acuan untuk membuat program aplikasi pada *smartphone* android dan program *ladder* agar dapat dikembangkan sesuai dengan kebutuhan. Apabila aplikasi android telah selesai dibangun, selanjutnya perlu dilakukan pengujian terhadap program yang telah dibuat, serta dilihat apakah program pada aplikasi android tersebut telah terintegrasi dengan PLC atau tidak. Jika berhasil terintegrasi dengan baik, maka tahapan selanjutnya adalah melakukan uji terhadap sistem tersebut dan melakukan analisis terhadapnya untuk meninjau apakah telah sesuai dengan kebutuhan yang diperlukan pengguna.

2. DASAR TEORI

2.1 Perancangan Sistem (*System Architecture*)

Sistem merupakan kombinasi dari elemen-elemen interaktif yang disusun untuk menampilkan atribut, perilaku dan kemampuan untuk mencapai satu atau lebih tujuan yang telah ditetapkan. Model dari sistem dapat dilihat pada Gambar 2 model of system. Model dari sistem tersebut merupakan definisi sistem secara general, suatu sistem yang dibuat memiliki tujuan yang berasal dari kebutuhan (requirement) dari stakeholder dan sistem tersendiri terdiri atas integrasi dari elemen sistem, interaksi pada elemen sistem akan mencapai karakteristik dari sistem seperti property, perilaku, dan kapabilitas dari sistem. Dengan terbentuknya karakteristik sistem tersebut akan mencapai tujuan yang telah ditetapkan [9].



Gambar 2 Model of System

2.2 Kebijakan Pemerintah Terkait Covid-19

Dengan adanya pandemi Covid-19 yang melanda seluruh dunia termasuk Indonesia, untuk menekan penyebaran virus tersebut Pemerintah Indonesia membuat peraturan yang tercantum pada Surat Edaran Menteri Komunikasi Dan Informatika Republik Indonesia Nomor 4 Tahun 2020. Menurut surat edaran tersebut, dengan adanya wabah atau pandemic Covid-19 perlu dilakukan pembatasan dalam skala besar yang dimana pemanfaatan teknologi informasi dan komunikasi menjadi sangat penting untuk menunjang pekerjaan yang dilakukan dari rumah atau *Work From Home* (WFH) [10]. Terkait dengan pelaksanaannya pada sektor industri di Indonesia, diberlakukan peraturan yang mengacu kepada Instruksi Menteri Dalam Negeri Nomor 15 Tahun 2021 mengenai penetapan pemberlakuan pembatasan kegiatan masyarakat darurat Covid-19 pada Kabupaten dan Kota di wilayah Jawa dan Bali, untuk sektor industri esensial diberlakukan 50% maksimal staf *Work From Office* (WFO) dengan protokol kesehatan yang ketat [1].

2.3 Otomasi

Otomasi adalah penggunaan perintah pemrograman logis dan peralatan mekanik untuk menggantikan pengambilan keputusan ataupun aktivitas repons manual dari manusia [11]. Secara umum, otomasi dapat didefinisikan sebagai proses yang mengikuti urutan operasi yang telah ditentukan dengan menggunakan sedikit ataupun tanpa tenaga manusia, menggunakan peralatan mekanik, dan perangkat khusus yang melakukan proses kontrol pada manufaktur dengan memanfaatkan penggunaan perangkat, sensor, aktuator, dan peralatan yang mampu mengamati proses manufaktur yang berjalan serta membuat keputusan terhadap perubahan yang perlu dilakukan dalam operasi dan mengendalikannya [12]. Suatu sistem otomasi terdiri dari tiga elemen dasar, yaitu power, program instruction, dan control system [13].

2.4 Human Machine Interface (HMI)

Setiap *Human Machine Interface* (HMI) dalam *Cyber-Physical System* (CPS) dikembangkan untuk mendukung interaksi antara CPS dengan *user*, maka setiap fungsionalitas yang dibangun dalam HMI harus dirancang dengan benar dan memerlukan konsep yang menjadikan penggunaan fungsi HMI dengan benar oleh *user* [14]. Berdasarkan [15], terdapat dua fungsi utama dari interface tersebut, yaitu *direct monitoring* dan *direct controlling*.

1. Direct Monitoring

Direct monitoring untuk melakukan observasi atau pemantauan terhadap variabel proses yang ditampilkan, *user* dapat melakukan pemantauan secara langsung dan mengetahui apa saja yang terjadi pada proses/mesin melalui *visual display* dari parameter yang telah ditentukan.

2. Direct Controlling

Direct controlling untuk melakukan manipulasi atau pengendalian proses dari parameter yang ditentukan.

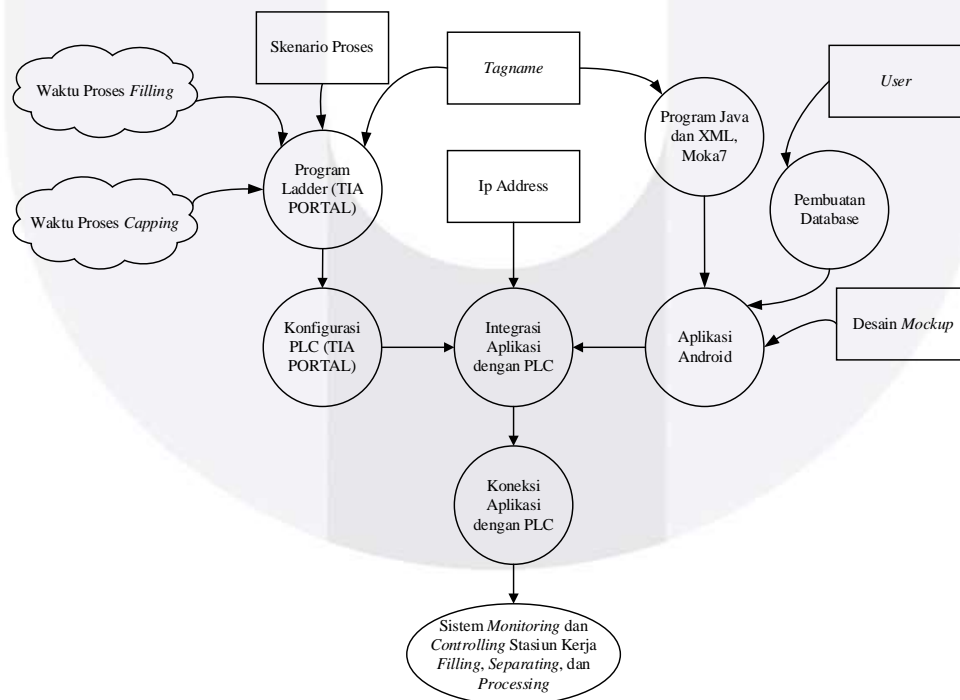
2.5 Programmable Logic Controller (PLC)

Programmable Logic Controller (PLC) merupakan bentuk khusus dari pengendali/kontrol berbasis mikroprosesor yang menggunakan memori yang dapat diprogram untuk memberikan instruksi dan mengimplementasikan beberapa fungsi seperti *logic*, *sequencing*, *timing*, *counting*, dan *arithmetic* dalam hal mengendalikan mesin dan proses tertentu serta PLC dirancang agar memudahkan *engineer* melakukan pekerjaannya dengan pengetahuan tentang bahasa komputer (*computing language*) yang terbatas [16].

2.6 Android

Android adalah perangkat lunak *open-source* yang mencakup sistem operasi, *middleware*, dan aplikasi utama untuk perangkat seluler. Android tersedia bebas dan dapat dilakukan kostumisasi, tidak ada konfigurasi perangkat keras atau perangkat lunak yang tetap untuk sistem android [17].

3. Model Konseptual



Gambar 3. Model Konseptual

Pada penelitian yang dilakukan terhadap stasiun kerja *filling*, *separating*, dan *processing* pada simulator *bottling plant*, terdapat beberapa *input* yang digunakan dalam melakukan perancangan sistem *monitoring* dan *controlling* menggunakan *smartphone* yaitu skenario proses, waktu proses *filling*, waktu proses *capping*, *user*, *ip adress* dan tagname PLC. Skenario proses, waktu proses *filling* maupun waktu proses *capping* digunakan sebagai inputan untuk membuat program *ladder* pada *software* TIA PORTAL. Tagname PLC ditentukan sebelum membuat program *ladder* dan aplikasi android agar dapat digunakan sebagai acuan dalam membuat program *ladder* dan aplikasi android untuk penyesuaian alamat masing-masing *input/output*. *IP address* diperlukan untuk melakukan konfigurasi pada PLC dan juga merupakan inputan yang digunakan sebagai salah satu inputan pada pembuatan aplikasi android. Pada aplikasi android dimulai dengan perlunya *user* untuk melakukan *login* hingga menginput alamat *ip* untuk dapat masuk ke halaman utama aplikasi. Setelah program *ladder* pada TIA PORTAL dan aplikasi android yang dirancang pada Android Studio selesai, tahapan selanjutnya adalah melakukan pengkoneksian antara *interface* yang telah dibuat (aplikasi) dengan PLC sebagai *controller* pada sistem. Setelah berhasil terkoneksi, maka proses *monitoring* dan *controlling* pada *filling*, *separating*, dan *processing station* dapat dilakukan secara *online* melalui aplikasi berbasis *smartphone* android. Perancangan sistem tersebut dilakukan agar proses *monitoring* dan *controlling* dapat dilakukan secara *real-time* sehingga dapat mempermudah *user* dalam melakukan pekerjaannya secara *mobile* dengan hanya memerlukan *smartphone*. *Interface* yang dirancang memanfaatkan penggunaan aplikasi *smartphone* android yang dirancang menggunakan Android Studio dengan melakukan penyesuaian sesuai dengan variabel proses yang dipilih dan serta dapat melakukan fungsi pembacaan (*read*) dan perubahan (*write*) data ke PLC.

4. Pembahasan

4.1 Pengumpulan Data

Penentuan parameter yang akan diambil dari stasiun kerja *filling*, *separating*, dan *processing* ditentukan berdasarkan skenario proses yang ditinjau secara langsung dari *simulator bottling plant* dan membaca *ladder diagram* yang telah terprogram pada PLC. Berikut merupakan skenario proses yang terdapat pada stasiun kerja *filling*:

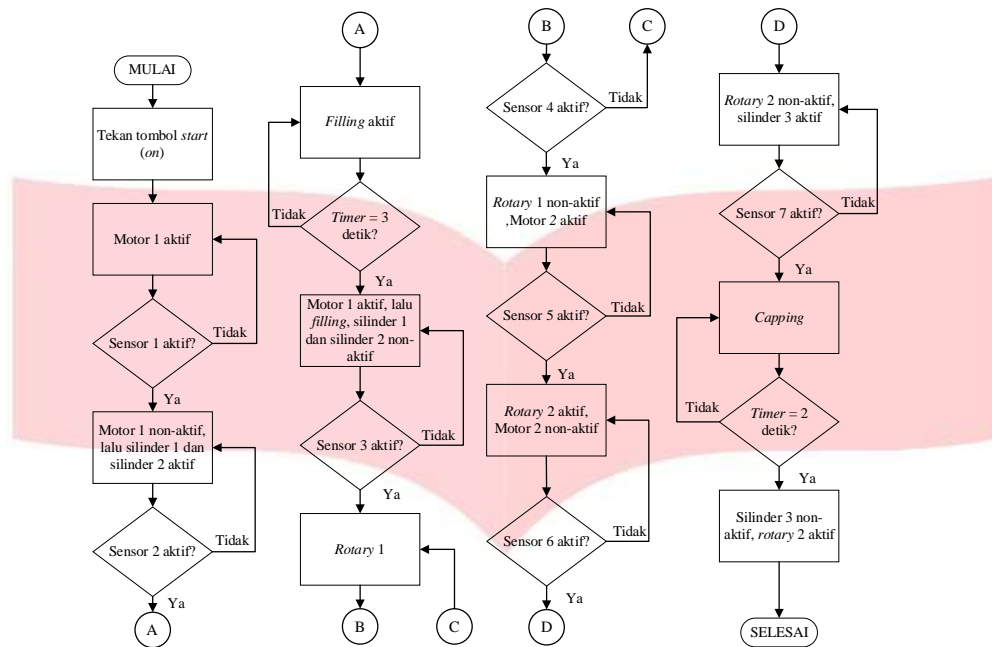
1. Tombol *start (on)* ditekan untuk menyalakan motor 1 (*conveyor 1*).
2. Ketika botol mengenai sensor 1 maka motor 1 (*conveyor 1*) non-aktif, silinder 1 aktif untuk mencengkram botol dan silinder 2 aktif turun hingga titik maksimum.
3. Ketika silinder 2 mencapai titik maksimum, maka sensor 2 aktif dan memicu *filling* (pengisian botol) untuk aktif selama 3 detik.
4. Setelah 3 detik maka silinder 1 dan silinder 2 non-aktif dan motor 1 (*conveyor 1*) kembali aktif.

Berikut merupakan skenario proses yang terdapat pada stasiun kerja *separating*:

1. Botol yang tiba di stasiun kerja *separating* akan mengenai sensor 3, sensor 3 aktif maka *rotary 1* aktif.
2. Sensor 4 aktif akan memicu *rotary 1* non-aktif dan motor 2 (*conveyor 2*) aktif untuk menyalurkan botol ke stasiun kerja *processing*.

Sementara itu, skenario proses yang terdapat pada stasiun kerja *processing* adalah sebagai berikut:

1. Botol yang tiba di *processing station* akan dideteksi oleh sensor 4 yang akan aktif dan men-trigger *rotary 2* aktif yang akan berputar menuju tempat pemberian tutup botol dan menonaktifkan motor 2 (*conveyor 2*).
2. Ketika sampai pada proses *capping*, sensor 5 aktif dan akan men-trigger *rotary 2* non-aktif dan mengaktifkan silinder 3 turun hingga titik maksimal dan sensor 6 aktif untuk memulai proses *capping* selama 2 detik.
3. Setelah proses *capping* selesai maka silinder 3 akan kembali ke posisi awal (non-aktif) dan mengaktifkan kembali *rotary 2*.



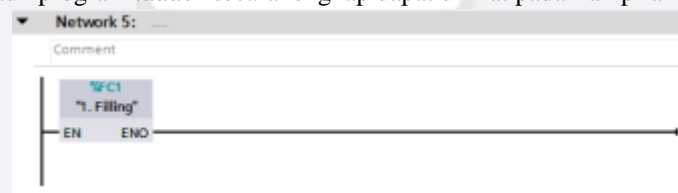
Gambar 4 Skenario Proses

4.2 Pembuatan Program Ladder

Program *ladder* untuk PLC dikerjakan pada *software* TIA PORTAL dengan referensi *tagname* yang telah dirancang sebelumnya, program *ladder* pada *software* tersebut dimulai dengan penentuan jenis PLC yang digunakan (CPU 1212 AC/DC/RLY), lalu program dibuat menggunakan fungsi *program blocks* yang dimana terbagi menjadi *main block*, *function*, dan *data block*.

1. Main Block (OB)

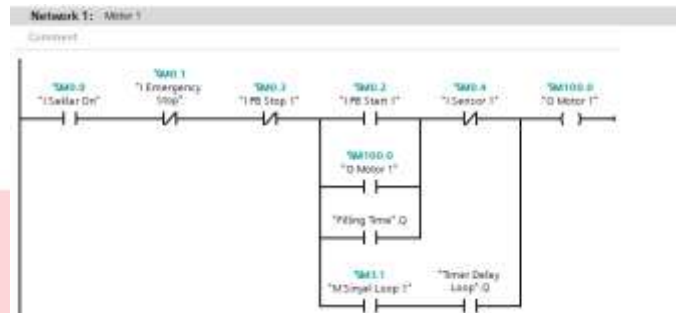
Pada bagian *main block* berisi mengenai *function* apa saja yang akan dijalankan termasuk program untuk setiap stasiun kerja yang dibuat pada *function block*. *Main block* berfungsi sebagai program inti yang akan dijalankan pertama kali. Pada bagian *main block* terdapat beberapa *function* (FC) dari setiap stasiun kerja. Gambar 5 *Main Block* (OB) merupakan salah satu contoh *function block* stasiun kerja *filling* yang terdapat pada *main block*. Untuk program *ladder* secara lengkap dapat dilihat pada Lampiran A.



Gambar 5 Main Block (OB)

2. Function (FC)

Pada bagian *function* dijalankan logika yang terdapat pada skenario proses, didalamnya terdapat sekumpulan nilai *input* yang menghasilkan *output* tertentu. Digunakannya alamat internal memori karena terdapat batasan pada jumlah *input/output* pada PLC yang digunakan, yang kemudian jika ingin diterapkan langsung pada simulator *bottling plant* dapat menggunakan alamat *input* dan *output* yang sesuai dengan PLC. Untuk setiap stasiun kerja mulai dari *filling* (FC1), *separating* (FC2), dan *processing* (FC3) dibuat dalam *function* yang terpisah. Berikut pada Gambar 6 menunjukkan salah satu contoh *function* pada *filling station*, dimana terdapat sekumpulan *input* yang menghasilkan *output* berupa motor 1 (*conveyor*).



Gambar 6 *Function Block (FC)*

3. *Data Block (DB)*

Data block disini digunakan untuk fungsi *start/stop* pada *main block* dan *timer* pada setiap *FC*. Dimana *DB* dapat dipanggil ke masing-masing fungsi baik pada *OB* maupun *FC*. Gambar 7 *Data Block (DB)* menunjukkan *data block* dimana terdapat *start* dan *stop*. Untuk *data block* selengkapnya dapat dilihat pada Lampiran E.

DB Action				
	Name	Data type	Offset	Start value
1	Static			
2	Start	Bool	0.0	false
3	Stop	Bool	0.1	false

Gambar 7 *Data Block (DB)*

4.3 Pembuatan Aplikasi Android

Pembuatan aplikasi android dikerjakan pada software Android Studio, pada software tersebut terdapat dua hal yang perlu dilakukan pengerjaan, yaitu bagian layout yang memiliki format xml dan program Java. Sebelum mengerjakan layout dan pemrograman java, dilakukan inisiasi import moka7 yang merupakan port Java dari klien Snap7. Kegunaannya adalah untuk membaca alamat PLC yang telah dibuat program ladder pada tahap sebelumnya.

1. Pembuatan Program *Layout*



Pembuatan *layout* dikerjakan untuk setiap *activity* atau halaman pada aplikasi android, dimana terdiri dari 9 *activity* diantaranya yaitu *login*, *register*, *forget password*, *set connection*, *filling*, *separating*, *processing*, *pick & place*, dan *distribution box*. Pada halaman *filling*, terdapat beberapa gambar eksternal seperti *background station* dan lampu indikator, penambahan gambar eksternal dilakukan dengan cara membuat gambar baru ataupun *input* gambar ke bagian *drawable* pada *App* → *java* → *res* → *drawable*. Seperti pada pembuatan lampu indikator yang dibuat dengan menggunakan fungsi *vector*:

```
<vector android:height="24dp" android:tint="#F42217"
    android:viewportHeight="24" android:viewportWidth="24"
    android:width="24dp"
    xmlns:android="http://schemas.android.com/apk/res/android">
</vector>
```

Untuk setiap gambar diposisikan dan diatur ukurannya seperti perancangan *layout* yang telah dibuat, pengaturan ukuran setiap gambar dapat menggunakan fungsi '*android:layout_width=""*' dan '*android:layout_height=""*', sedangkan untuk memposisikannya dapat memanfaatkan fungsi *constraint* pada xml.

Tabel 1 indikator lampu aplikasi android menjabarkan tentang kondisi indikator dengan warna lampunya pada masing-masing *input/output* yang dilakukan *monitoring*

Tabel 1 Indikator Lampu Aplikasi Android

Kondisi	Warna Lampu Indikator	Nama Program
Indikator <i>Off</i>		R.drawable.ic_baseline_brightness_1_24
Indikator <i>On</i>		R.drawable.on

2. Pembuatan Program Java

Program yang dibangun pada aplikasi ini menggunakan Java dengan *port* Moka7. Dengan telah dibuatnya *layout* untuk setiap halaman, maka selanjutnya dapat dibuat program Java untuk melakukan logika yang akan dibangun pada aplikasi ini. Untuk membuat program pada setiap *activity* dapat dilakukan pada *App* → *java* → *com.application.nama_aplikasi*.

a. Login Activity

Pada halaman *login*, *user* akan melakukan proses autentikasi, dimana data *user* tersebut telah tersimpan pada *database*. Berikut merupakan langkah-langkah dalam pembuatan program pada halaman *login*.

1) Inisiasi Variabel dan Deklarasi Fungsi

Inisiasi variabel dan pendeklarasian fungsi dibuat untuk menentukan variabel yang dapat di *input* oleh *user* serta fungsi apa yang akan dilakukan oleh aplikasi. Seperti pada halaman *login* terdapat variabel tombol *login*, ataupun *input email/password*. Selain itu, karena proses *login* disini memerlukan autentikasi ke *firebase* sebagai *database*, perlu dilakukan deklarasi fungsi autentikasi *firebase*.

2) Merancang Fungsi IF

Perancangan fungsi IF dilakukan untuk menentukan logika yang akan dijalankan, seperti saat *user* melakukan *input email* ataupun *password* akan menjadi *output* tertentu sesuai dengan logika yang telah ditentukan.

3) Merancang Fungsi Button

Pada halaman *login* terdapat beberapa *button* yang perlu dirancang fungsi setiap *button* nya. Beberapa *button* seperti *forget password*, *register*, dan *login* dilakukan dengan menggunakan fungsi *setOnClickListener*

4) Pengkoneksian ke Firebase

Pada fungsi *login* dilakukan autentikasi yang terhubung dengan *database* pada *firebase*. Berikut merupakan *script* untuk autentikasi *user* dengan *database*. Pengkoneksian ini digunakan untuk melakukan proses cek berdasarkan data yang telah dilakukan *input* oleh *user*.

b. Register Activity

Halaman *register* diperuntukkan agar *user* baru yang belum memiliki akses akun bisa melakukan registrasi data. Berikut ini merupakan langkah-langkah pembuatan program untuk halaman *register*.

1) Inisiasi Variabel dan Deklarasi Fungsi

Menginisiasikan variabel beserta fungsinya yang dibutuhkan pada halaman registrasi diantaranya seperti *email*, nama, dan *password user* baru. Pada halaman *register*, perlu ditambahkannya fungsi penyimpanan data *user* ke *firebase* dengan menggunakan fungsi “private *Firestore fStore*” dan autentikasi data *user* dengan fungsi “private *FirebaseAuth firebaseAuth*”.

2) Merancang Fungsi IF

Fungsi IF yang dirancang pada halaman *register* berfungsi agar *user* melakukan penginputan data saat registrasi sesuai dengan standar yang ditetapkan. Berikut ini merupakan *script* dari fungsi IF yang telah dibuat.

3) Merancang Fungsi *Button*

Tombol atau *button* yang terdapat pada halaman *register* adalah *button* “registrasi” yang dimana saat *user* ingin menyelesaikan tahapan registrasi. Penggunaan fungsi *button* dapat menggunakan “setOnClickListener” untuk *button* “registrasi”.

4) Pengkoneksian ke *Firebase*

Pengkoneksian ke *firebase* sebagai *database* yang digunakan berfungsi sebagai tempat penyimpanan data *user* yang telah berhasil melakukan proses registrasi agar dapat melakukan proses *login* pada aplikasi.

c. *Set Connection Activity*

Pada halaman *set connection* terdapat fungsi *input ip address* agar aplikasi dapat terhubung dengan PLC. Jika *user* benar dalam menginputkan *ip address* sesuai dengan PLC yang dituju, maka akan masuk ke halaman proses *filling*, sebaliknya jika *ip address* salah maka akan muncul peringatan kepada *user*, seperti yang dapat dilihat pada *script* dibawah ini.

```
String connection="";

    if (res == 0) {
        connection = "sukses";
    }else{
        connection = " gagal";
    }

    if ( connection.equals("sukses")){

        Intent intent = new Intent(SetConnection.this, Filling.class);
        intent.putExtra("ipaddress",IpAddr);
        startActivity(intent);
    }
    else {
        TextViewtextView=(TextView)findViewById(R.id.textView70);
        textView.setText("Maaf IP yang anda masukkan salah");
    }
}
```

d. *Filling Activity*

Pada halaman *filling* terdapat indikator masing-masing *input/output* pada stasiun kerja *filling* beserta tombol mulai dan berhenti, dimana setiap indikator *input/output* dihubungkan secara *real-time* dengan PLC, oleh karena itu *port* Moka7 digunakan sebagai program tambahan untuk pembacaan/penulisan program dari/ke PLC. Berikut merupakan langkah-langkah pembuatan program pada halaman *filling*.

1) Inisiasi Variabel dan Deklarasi Fungsi

Beberapa tombol dan variabel fungsi perlu dilakukan inisiasi dan pendeklarasian agar program dapat terpanggil pada halaman ini, terdapat beberapa tombol untuk berpindah ke halaman proses yang lain maupun indikator *input/output* yang telah ditentukan.

2) Merancang Fungsi *Button*

Terdapat beberapa fungsi *button* pada halaman proses *filling*, yaitu *button* untuk berpindah ke halaman proses yang lain, serta *button* untuk melakukan *start*, *stop*, *emergency*, maupun *logout*.

3) Pengkoneksian *Library* Moka7 ke PLC

Untuk melakukan komunikasi antara aplikasi android dengan PLC diperlukan *library* Moka7 seperti yang telah disampaikan sebelumnya. Penyesuaian perlu dilakukan dengan melihat data apa yang akan dibaca (*read*) ataupun diubah (*write*) ke PLC pada *program ladder* yang telah diunggah.

4) Penambahan Fungsi *Running*

Fungsi *running* ditambahkan untuk menjalankan program yang telah dibangun untuk dapat menjalankan komunikasi antara aplikasi android dengan PLC.

5) Membuat Fungsi Device PLC yang Tersedia

Untuk terhubung dengan PLC, perlu dilakukannya pengecekan terhadap ketersediaan *device* PLC, dimana PLC memiliki *slot* dan *rack* tertentu yang perlu disesuaikan pada penghubungan aplikasi ke PLC untuk berkomunikasi.

6) Merancang Fungsi IF

Fungsi IF disini dirancang untuk melakukan pengkoreksian terhadap pengkoneksian dengan PLC yang dituju.

5. Pengkoneksian Aplikasi Android dengan PLC

Pengkoneksian aplikasi android dengan PLC dilakukan dengan menyesuaikan IP Address pada keduanya, komunikasi antara keduanya menggunakan protokol komunikasi Modbus TCP/IP dimana PLC yang terhubung dengan router yang memiliki IP Address tersendiri dengan jenis koneksi privat. Pada TIA PORTAL itu tersendiri, penyesuaian IP Address dapat dilihat pada Gambar 8 berikut ini.



Gambar 8 Set IP Address

Sedangkan pada aplikasi android yang telah berhasil terhubung ke database dapat melakukan input Ip Address yang sama dengan yang terhubung dengan PLC, yaitu “192.168.1.24” pada halaman Set Connection. Maka keduanya dapat terhubung untuk melakukan komunikasi

6. KESIMPULAN

Berdasarkan perancangan sistem yang telah dibuat dengan menggunakan metode *v-model*, dimulai dari tahap pengumpulan data, identifikasi kebutuhan, spesifikasi fungsional, perancangan arsitektur sistem, perancangan *database*, pembuatan program *ladder* dan aplikasi android, hingga pengujian fungsionalitas sistem maka didapatkan kesimpulan bahwa perancangan sistem *monitoring* dan *controlling* menggunakan *smartphone* pada stasiun kerja *filling*, *separating*, dan *processing* telah berhasil dilakukan. Dari hasil pengujian fungsionalitas sistem pada masing-masing parameter *input/output* yang digunakan untuk komunikasi antara aplikasi android dengan PLC berfungsi dengan baik, dimana fungsi pembacaan (*read*) pada PLC dan pengubahan (*write*) ke PLC dapat dilakukan secara *real-time* sesuai dengan yang diharapkan. Dari segi tampilan, desain antarmuka pada aplikasi android yang memiliki *layout* stasiun kerja yang dilengkapi dengan indikator setiap *input/output* memudahkan *user* untuk melakukan *monitoring* dan *controlling* pada masing-masing stasiun kerja, selain itu penggunaan aplikasi android dapat meningkatkan mobilitas *user* dalam penggunaannya. Dengan protokol komunikasi Modbus TCP/IP yang digunakan terdapat *delay* pada masing-masing *input/output* setiap stasiun kerja yang disebabkan oleh kecepatan sinyal serta kemampuan pembacaan banyaknya jumlah data pada aplikasi android.

Referensi

- [1] Menteri Dalam Negeri, “Instruksi Menteri Dalam Negeri Nomor 15 Tahun 2021.” MENTERI DALAM NEGERI, 2021.
- [2] A. Rojko, “Industry 4.0 Concept: Background and Overview,” vol. 11, no. 5, pp. 77–90, 2017.
- [3] R. Kumar, R. K. Singh, and Y. K. Dwivedi, “Application of industry 4.0 technologies in SMEs for ethical and sustainable operations: Analysis of challenges,” *J. Clean. Prod.*, vol. 275, p. 124063, 2020, doi: 10.1016/j.jclepro.2020.124063.
- [4] H. Kopetz, *Real-time systems: Design principles for distributed embedded applications*, vol. 34, no. 10. 2011.
- [5] N. Isac, C. Dobrin, and W. Badshah, “The Impact of Sustainable Transition of Automation on Employees in the Automotive Sector and the Influence of Corona Pandemic,” *Rev. Int. Comp. Manag.*, vol. 21, no. 4, pp. 429–436, 2020, doi: 10.24818/RMCI.2020.4.429.
- [6] T. Chen and C. Lin, “Smart and automation technologies for ensuring the long-term operation of a factory amid the Covid-19 pandemic: an evolving fuzzy assessment approach,” pp. 3545–3558, 2020.
- [7] K. Kark, J. Gill, and T. Smith, “Maximizing the impact of technology investments in the new normal,” no. February, 2021.
- [8] P. R. Childs, *Mechanical Design Engineering Handbook*, 2nd ed. 2019.
- [9] T. Weilkiens, G. Lamm, Jesko, S. Roth, and M. Walker, *Model-Based System Architecture*. 2016.
- [10] Menteri Komunikasi dan Informatika, “Surat Edaran Menteri Komunikasi dan Informatika Republik Indonesia Nomor 4 Tahun 2020.” MENTERI KOMUNIKASI DAN INFORMATIKA, 2020.
- [11] F. Lamb, *Industrial Automation Hands-On*. 2013.
- [12] A. K. Gupta, S. K. Arora, and J. R. Westcott, *Industrial Automation and Robotics*. 2017.
- [13] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, 4th ed. Pearson Higher Education, Inc., 2015.
- [14] H. Kopetz, *Simplicity is Complex: Foundations of Cyber-Physical System Design*. 2019.
- [15] K. L. S. Sharma, *Overview of Industrial Process Automation: Second Edition*, 2nd ed. Elsevier Inc, 2017.
- [16] W. Bolton, *Programmable Logic Controllers*, 6th ed., vol. 195. Elsevier Ltd, 2016.
- [17] J. F. DiMarzio, *Beginning Android® Programming with Android Studio*. John Wiley & Sons, Inc., 2017.