

LAPORAN MAGANG

PEMBANGUNAN

SISTEM ANTIFRAUD

BERBASIS INTERNET OF THINGS



Dipersiapkan oleh:

Yohanes Dwiki Witman Gusti Made / 140707748

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA

2018

HALAMAN PENGESAHAN

Laporan Magang



Dosen Pembimbing

Martinus Maslim, S.T., M.T.

Pembimbing Lapangan



Ir. Arvin M.Sc.

Surat Keterangan Selesai Magang
No: 048/INS-SK/IX/2018

Dengan ini, kami yang bertandatangan di bawah ini :

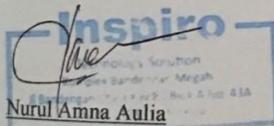
Nama : Nurul Amna Aulia
Jabatan : HR Generalist

Menerangkan bahwa :

Nama : Yohanes Dwiki Witman Gusti Made
NIM : 140707748
Universitas : ATMA JAYA YOGYAKARTA

Telah menyelesaikan kegiatan Magang di PT. INSPIRO dari tanggal 22 Juni 2018 sampai dengan 21 September 2018. Selama bekerja di Perusahaan ini, mahasiswa yang bersangkutan telah bekerja dengan baik. Demikian surat keterangan PKL ini kami buat untuk dapat digunakan sebagaimana mestinya.

Jakarta, 21 September 2018



Nurul Amna Aulia
HR Generalist

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga penulis bisa menyelesaikan pelaksanaan magang di perusahaan PT. Inspiro dengan lancar dan baik. Penulis juga bersyukur dapat menuntaskan seluruh pekerjaan yang telah diberikan dengan tepat waktu dan pada akhirnya boleh terangkum ke dalam laporan magang ini.

Magang adalah salah satu mata kuliah wajib yang harus ditempuh oleh setiap mahasiswa Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta. Penyusunan laporan magang ini dilakukan untuk memenuhi hal tersebut, juga sebagai salah satu syarat kelulusan akademik Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta.

Penulis banyak mendapatkan pengalaman, bantuan, serta bimbingan baik dari pihak perusahaan PT. Inspiro maupun dari pihak Universitas Atma Jaya Yogyakarta. Pada kesempatan ini, penulis mengucapkan rasa syukur dan terima kasih kepada:

1. Tuhan Yang Maha Esa atas berkat dan rahmat yang diberikan oleh-Nya.
2. Orang tua tercinta dan sanak saudara yang selalu mendoakan, mengunjungi, dan mendukung penulis selama masa pelaksanaan magang.
3. Bapak Martinus Maslim, S.T., M.T. selaku Kepala Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta dan selaku Dosen Pembimbing Magang yang telah memberikan arahan serta dukungan bagi penulis sehingga pelaksanaan magang bisa berjalan dengan baik.
4. Bapak Ating Kencana, Diplom. Informatiker. selaku Direktur perusahaan PT. Inspiro yang telah menerima penulis dan memberikan arahan, nasehat, pemikiran, motivasi serta bantuan moril maupun materiil selama masa pelaksanaan magang.

5. Bapak Ir. Arvin M.Sc. selaku Wakil Direktur perusahaan PT. Inspiro dan selaku Pembimbing Lapangan yang senantiasa memberikan bantuan dan arahan secara langsung kepada penulis, baik secara teknis maupun non-teknis selama masa pelaksanaan magang.
6. Bapak Suharmin S.Kom. selaku IT Manager perusahaan PT. Inspiro yang telah memberi bimbingan selama masa pelaksanaan magang.
7. Bapak Robert Andi Haloho yang memimpin proyek *Internet of Things* (IoT) selama dua bulan di perusahaan PT. Inspiro dan Bapak Taufik Hidayanto, Yoanes Palmariato, Reandro Kristanto, dan adik Dylan Edbert sebagai rekan kerja dan magang di Tim Developer IoT.
8. Bapak Harianto, Felix, Bagus, Syaiful, dan Ibu Dewi sebagai .NET Software Developer serta Wan Adli sebagai System Analyst.
9. Bapak Vicky Setiadi, Andi, dan Leo sebagai Xamarin Mobile Developer.
10. Bapak Jackson dan Dede Dharma yang telah membimbing penulis untuk mendalami Accounting dan ERP. Bapak Naufal sebagai Quality Assurance yang telah membimbing penulis untuk mendalami proses pengujian program.
11. Ibu Suvisakha Sudiarto sebagai Account Manager yang telah banyak membantu penulis dalam mengkomunikasikan hasil produk kepada klien.
12. Ibu Nia Kurnia Dian sebagai Human Resources dan General Affairs yang telah banyak membantu penulis dalam hal peminjaman dan pengadaan barang.
13. Ibu Nurul, July, dan Putri sebagai Human Resources Development yang telah banyak membantu penulis pada saat proses penerimaan dan urusan administratif.
14. Davin, Farrel, Levina, dan Edellyn sebagai rekan magang dari Universitas Bina Nusantara yang telah banyak membantu Tim Developer IoT.
15. Semua pihak yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa laporan ini jauh dari sempurna yang disebabkan oleh pengetahuan penulis yang kurang luas. Oleh karena itu, penulis mengharapkan saran dan kritik yang bersifat membangun untuk Penulis dan perbaikan laporan ini.

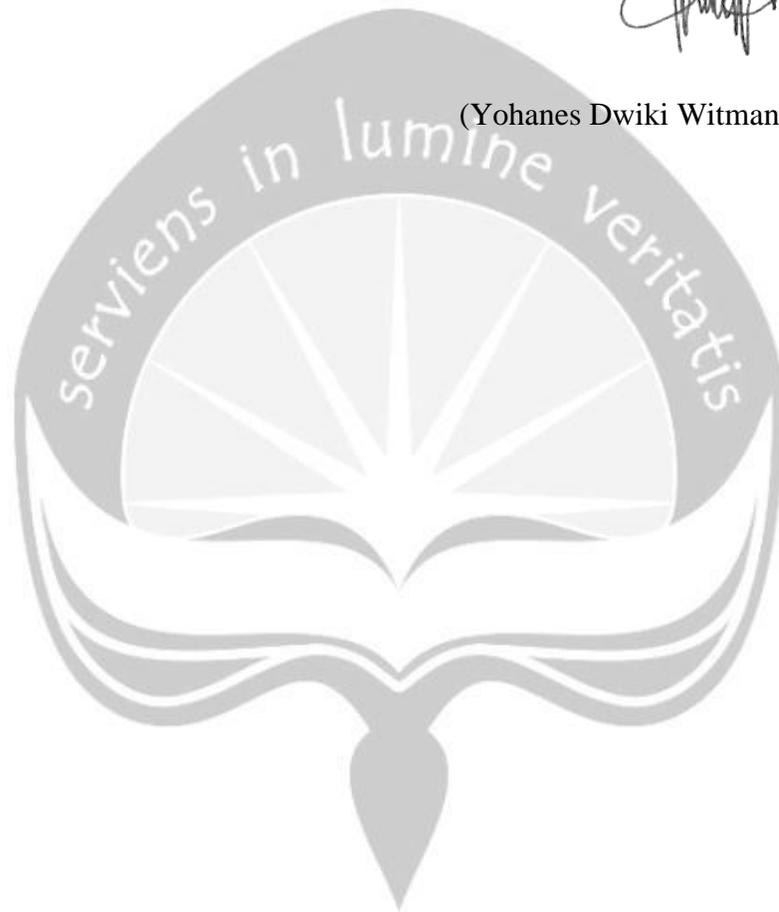
Akhir kata, semoga laporan ini berguna bagi penulis dan pihak-pihak lainnya bisa menemukan manfaat yang nyata dari laporan ini.

Yogyakarta, 22 September 2018

Penulis,



(Yohanes Dwiki Witman Gusti Made)



DAFTAR ISI

HALAMAN PENGESAHAN	i
KATA PENGANTAR	iii
DAFTAR ISI	vi
DAFTAR GAMBAR DAN TABEL	vii
BAB I	1
1.1 SEKILAS PERUSAHAAN	1
1.2 SEJARAH PERUSAHAAN	1
1.3 VISI, MISI, DAN TUJUAN	2
1.4 STRUKTUR ORGANISASI	3
1.5 DESKRIPSI TUGAS STRUKTUR ORGANISASI	4
1.6 DEPARTEMEN IT DALAM PERUSAHAAN	6
BAB II	8
2.1 PENJELASAN LOGBOOK	8
2.2 HASIL PEKERJAAN SECARA UMUM	60
2.2.1 DESKRIPSI PEKERJAAN DAN KEBUTUHAN KLIEN	60
2.2.2 ARSITEKTUR SISTEM ANTIFRAUD	61
2.2.3 METODE PENGKERJAAN PROYEK	63
2.2.4 WIRING DIAGRAM MASTER / SLAVE	67
BAB III	68
3.1 MANFAAT MAGANG	68
3.2 PENERAPAN ILMU DALAM MAGANG	69
BAB IV	71
4.1 KESIMPULAN	71
4.2 SARAN	71
LAMPIRAN	72

DAFTAR GAMBAR DAN TABEL

<u>Gambar 1.1.1 – Logo Perusahaan</u>	1
<u>Gambar 1.4.1 – Struktur Organisasi PT. Inspiro</u>	3
<u>Gambar 2.1.1 – Kalkulator Web</u>	9
<u>Gambar 2.1.2 – Menu Awal Aplikasi Kiosk</u>	13
<u>Gambar 2.1.3 – Menu Pembayaran Aplikasi Kiosk</u>	13
<u>Gambar 2.1.4 – Program CS ERP</u>	17
<u>Gambar 2.1.5 – Kode Main Program Sistem Antifraud</u>	55
<u>Gambar 2.2.1 – Arsitektur Sistem Antifraud Versi 1</u>	61
<u>Gambar 2.2.2 – Arsitektur Sistem Antifraud Versi 2</u>	62
<u>Gambar 2.2.3 – Flowchart Sistem Antifraud</u>	66
<u>Gambar 2.2.4 – Wiring Diagram Sistem Antifraud</u>	67
<u>Gambar Lampiran</u>	72
<u>Tabel 2.1.1 – Penjelasan Logbook</u>	59
<u>Tabel 2.1.2 – Tabel Komparasi Bill Acceptor</u>	11
<u>Tabel 2.1.3 – Interaksi Modul Purchase Order</u>	20
<u>Tabel 2.1.4 – Interaksi Modul Material Request</u>	22
<u>Tabel 2.1.5 – Interaksi Modul Sales Order</u>	23
<u>Tabel 2.1.6 – Interaksi Modul Supplier Delivery Note</u>	25
<u>Tabel 2.1.7 – Interaksi Modul Purchase Invoice</u>	27
<u>Tabel 2.1.8 – Interaksi Modul Delivery Note</u>	28
<u>Tabel 2.1.9 – Interaksi Modul Invoice (Sales)</u>	30
<u>Tabel 2.1.10 – Interaksi Modul Accounting</u>	31
<u>Tabel 2.1.11 – Interaksi Modul Storages</u>	32
<u>Tabel 2.1.12 – Interaksi Modul Master Data Item</u>	33

<u>Tabel 2.1.13 – Pengujian Modul Purchase Order</u>	35
<u>Tabel 2.1.14 – Pengujian Modul Supplier Delivery Note</u>	36
<u>Tabel 2.1.15 – Pengujian Modul Purchase Invoice</u>	36
<u>Tabel 2.1.16 – Pengujian Modul Material Request</u>	37
<u>Tabel 2.1.17 – Pengujian Modul Sales Order</u>	38
<u>Tabel 2.1.18 – Pengujian Modul Delivery Note</u>	39
<u>Tabel 2.1.19 – Pengujian Modul Invoice (Sales)</u>	40
<u>Tabel 2.1.20 – Pengujian Modul Master Data Item</u>	41
<u>Tabel 2.1.21 – Pengujian Modul Storages</u>	42
<u>Tabel 2.1.22 – Pengujian Modul Accounting</u>	42
<u>Tabel 2.1.23 – COA Untuk Pembuatan Buku Besar</u>	44
<u>Tabel 2.1.24 – Diskusi Program Antifraud</u>	51



BAB I

PENDAHULUAN

1.1 Sekilas Perusahaan

PT. Inspiro adalah salah satu perusahaan teknologi yang menyediakan layanan teknologi informasi berdasarkan teknologi tren utama. Kantor PT. Inspiro berada di Kompleks Ruko Bandengan Megah Blok A No.43-A, Jalan Bandengan Utara Kav.81, Jakarta Utara. PT. Inspiro adalah konsultan profesional yang memberikan solusi IT dalam bentuk pengembangan perangkat lunak yang berkualitas dengan berkonsentrasi pada dua teknologi utama, yaitu *Microsoft .NET* dan solusi berbasis *Java*.



Gambar 1.1.1 – Logo Perusahaan

1.2 Sejarah Perusahaan

Persaingan yang semakin ketat antara perusahaan berbasis teknologi di Indonesia mendorong setiap perusahaan untuk memberikan pelayanan yang sebaik-baiknya. PT. Inspiro memulai usaha yang berbasis di Indonesia pada bulan Maret 2009 untuk memberikan layanan terbaik kepada klien dibidang teknologi informasi berdasarkan teknologi tren utama. Sistem utama mereka berbasis *.NET* dan *Java* dengan berfokus pada tiga layanan, yaitu *Software Development*, *Consulting*, dan *Managed / Administering Service*.

PT. Inspiro diperkenalkan ke publik oleh Bapak Ating Kencana. Beliau menjabat sebagai Direktur PT. Inspiro, lulus dari Hamburg University of Applied Sciences, Jerman, dengan lebih dari 20 tahun pengalaman internasional dan nasional

tentang pengembangan perangkat lunak yang berkualitas bagi klien. Sebagian besar klien PT. Inspiro adalah perusahaan-perusahaan besar di Indonesia yang bergerak di bidang jasa asuransi, seperti AIA, Prudential, Zurich, Manulife, dan FWD. Selain itu, PT. Inspiro juga menyediakan *outsourcing* karyawan mereka untuk ditempatkan bekerja di kantor klien seperti perusahaan Astra Internasional dan AHM.

Bapak Ating Kencana membangun PT. Inspiro bersama kedua rekan beliau, yaitu Bapak Arvin dan Suharmin. Bapak Arvin selaku Wakil Direktur PT. Inspiro, lulusan University of Colorado, Amerika Serikat, dengan lebih dari 15 tahun pengalaman kerja. Bapak Suharmin, selaku Manajer IT, lulusan Universitas Bina Nusantara, Jakarta, dengan lebih dari 10 tahun pengalaman kerja dalam pengembangan perangkat lunak (*MCP* dan *Blackpearl K2*). Beliau berdua bertanggung jawab untuk mengelola proyek dan membantu tim untuk mencapai hasil yang sukses sesuai dengan kebutuhan klien.

1.3 Visi, Misi, dan Tujuan Perusahaan

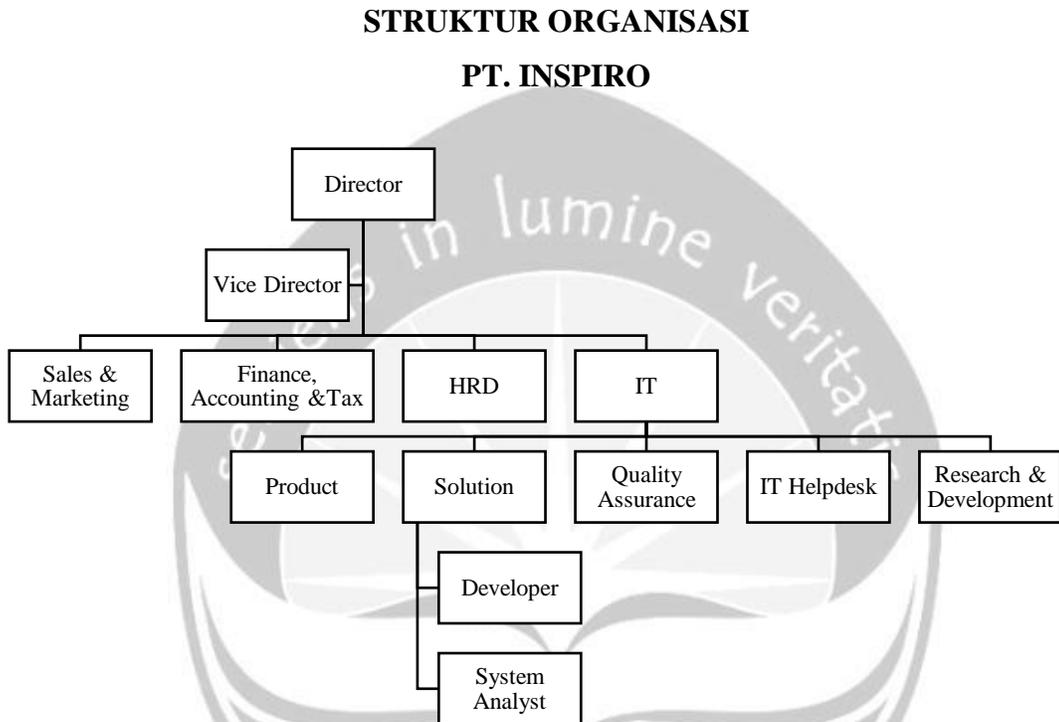
Visi:

Menjadi perusahaan solusi IT terkemuka dan perusahaan konsultan di Indonesia.

Misi:

- Memberikan layanan dan komitmen yang berkualitas baik kepada pelanggan.
- Membangun dan mempertahankan hubungan baik dengan pelanggan.
- Membangun tim yang solid antara karyawan, manajemen, dan pemilik saham perusahaan.
- Fokus pada proses kolaborasi dengan teknologi tingkat pertama.
- Membangun dan mempertahankan budaya dan nilai perusahaan.

1.4 Struktur Organisasi



Gambar 1.4.1 – Struktur Organisasi PT. Inspiro

1.5 Deskripsi Tugas Struktur Organisasi

1. Director dan Vice Director

Direktur dan Wakil Direktur memiliki tanggung jawab dalam memimpin dan mengarahkan perusahaan. Dalam melaksanakan tugasnya, Direktur dan Wakil Direktur menyelenggarakan fungsi:

- Memimpin seluruh karyawan dan eksekutif PT. Inspiro.
- Menawarkan visi dan misi tingkat strategis PT. Inspiro.
- Memimpin rapat umum, menyesuaikan alokasi waktu per item masalah, menentukan urutan agenda, diskusi ke arah konsensus, dan menyimpulkan tindakan dan kebijakan.
- Bertindak sebagai perwakilan PT. Inspiro dalam hubungannya dengan organisasi luar.
- Mengambil keputusan atau pada situasi tertentu yang dianggap perlu diputuskan dalam *meeting* klien.
- Menjalankan tanggung jawab dari direktur perusahaan sesuai dengan standar etika dan hukum PT. Inspiro.

2. Divisi IT

Divisi IT mempunyai tugas merumuskan dan melaksanakan kebijakan dibidang informatika, yang terbagi dan terinci secara sistematis ke dalam tugas masing-masing seksi. Dalam melaksanakan tugas, divisi IT melakukan fungsi:

- Membangun teknologi informasi sistem komputer dan aplikasinya (sub-divisi *product* dan *solution*)
- Memberikan solusi teknologi informasi (sub-divisi *solution*)
- Pengawasan dan pengetesan kualitas produk Teknologi Informasi PT. Inspiro (sub-divisi *quality assurance*)
- Membantu dalam melayani atau menanggapi pertanyaan teknis pengguna (sub-divisi *IT help desk*)

- Melakukan aktivitas riset dan pengembangan di perusahaan (sub-divisi *R&D*).

3. Divisi Finance, Accounting & Tax

Divisi Keuangan bertugas untuk merencanakan, mengembangkan, memberikan informasi, dan mengontrol fungsi keuangan dan akuntansi di PT. Inspiro. Dalam melaksanakan tugas, divisi keuangan menyelenggarakan fungsi:

- Mengelola fungsi akuntansi dalam memproses data keuangan untuk menghasilkan laporan keuangan.
- Mengkoordinasikan dan mengontrol perencanaan, pelaporan dan pembayaran kewajiban pajak perusahaan serta rencana keuangan untuk gaji dan jaminan kesehatan & ketenagakerjaan karyawan.
- Mengontrol arus kas perusahaan terutama pengelolaan piutang hutang dan memastikan ketersediaan dana.
- Merencanakan anggaran perusahaan, dan mengontrol penggunaan anggaran tersebut secara efektif dan efisien.
- Melakukan analisa keuangan untuk memberikan masukan kepada *Director* dalam mengambil keputusan bisnis.

4. Divisi Sales & Marketing

Divisi Penjualan dan Pemasaran bertanggungjawab pada operasi penjualan dan pemasaran produk secara keseluruhan di PT. Inspiro. Dalam melaksanakan tugas, divisi penjualan dan pemasaran menyelenggarakan fungsi:

- Perencanaan strategi pemasaran dengan memperhatikan tren pasar dan sumber daya di PT. Inspiro.
- Mengkomunikasikan hasil produk atau *prototype* kepada klien.
- Merencanakan *marketing research* yaitu dengan mengikuti perkembangan pasar, terutama terhadap aplikasi yang sejenis dari perusahaan pesaing.

- Melakukan perencanaan tindakan antisipatif dalam menghadapi penurunan peminatan produk PT. Inspiro.
- Merencanakan pengembangan jaringan pemasaran dengan mitra melalui produk PT. Inspiro.

5. Divisi Human Resources Development

Divisi sumber daya manusia menangani berbagai masalah pada ruang lingkup pegawai, manajer dan tenaga kerja lainnya untuk menunjang aktivitas PT. Inspiro. Dalam melaksanakan tugas, divisi HRD melakukan fungsi:

- Memasang iklan lamaran, menyortirnya dan melaksanakan tes psikologi dan wawancara kerja terhadap pelamar kerja.
- Melakukan pengembangan dan evaluasi pegawai serta pemberian kompensasi atau proteksi pegawai.
- Menangani hubungan yang berkaitan dengan perusahaan yang berhubungan dengan PT. Inspiro.
- Mengurus proses pengambilan cuti dan tukar *shift* sumber daya manusia.
- Melakukan kegiatan administratif untuk pegawai atau perusahaan yang berhubungan dengan PT. Inspiro.

1.6 Departemen IT dalam Perusahaan

Divisi IT merupakan divisi dimana kegiatan IT berjalan dan menggerakkan roda bisnis PT. Inspiro.

Bidang *Developer* mempunyai tugas-tugas seperti berikut.

- Melaksanakan perancangan, pengembangan, dan implementasi fungsi baru produk serta keamanan dari produk tersebut.
- Melaksanakan norma, kriteria, dan pedoman pengembangan fungsi aplikasi dan produk sesuai dengan ketentuan klien dan PT. Inspiro.

- Merancang dan mengembangkan sistem bagi pihak atau instansi lain yang bekerjasama dengan PT. Inspiro.
- Menyampaikan informasi dan laporan yang baik serta tepat waktu untuk kebutuhan internal maupun kebutuhan eksternal unit kerja yang bersifat rutin dan insidental.
- Mengkomunikasikan hasil produk atau *prototype* kepada bagian sales untuk diteruskan ke klien PT. Inspiro.
- Memperbaiki temuan *bug* untuk menjamin produk berkualitas bagi klien serta menangani insiden keamanan produk Inspiro dan instansi yang terkait sesuai ketentuan yang berlaku.
- Melaksanakan pemeliharaan barang-barang perusahaan terutama laptop, tablet, router, alat-alat IoT, dan lainnya.

Bidang *System / Business Analyst* mempunyai tugas-tugas sebagai berikut.

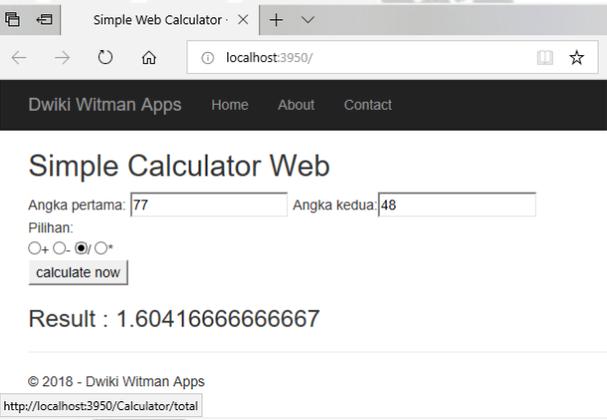
- Menentukan lingkup sistem untuk produk-produk klien dan PT. Inspiro.
- Mengumpulkan fakta berupa data yang kemudian diolah dan dianalisis sesuai dengan tujuan yang ingin dicapai.
- Berkonsultasi dengan manajemen dan klien untuk memastikan kesepakatan pada prinsip-prinsip sistem.
- Mengembangkan dokumen dan merevisi prosedur desain sistem, prosedur pengujian, dan standar kualitas.
- Mengkomunikasikan temuan-temuan analisis dalam bentuk laporan analisis sistem.

BAB II

PELAKSANAAN MAGANG

2.1 Penjelasan Log Book

No	Tanggal	Kegiatan
1.	22 Juni 2018	Hari pertama masuk kantor PT. Inspiro, penulis bertemu dengan Pak Ating, Pak Arvin, Pak Suharmin, Ibu Nurul dan HRD, Tim Developer, dan seluruh karyawan PT. Inspiro untuk berkenalan dan mengakrabkan diri. Penulis diberi proyek pertama untuk mencoba membuat kalkulator sederhana menggunakan framework .NET.
2.	23 Juni 2018	Penulis melanjutkan kalkulator sederhana dengan menginputkan dan menampilkan hasilnya melalui <i>Web Browser</i> . Operand-nya ada 2 (Angka1, Angka2) dan operator-nya ada 4 (tambah, kurang, kali, bagi).
3.	24 Juni 2018	Penulis memakai .NET MVC dengan melihat tutorial di website dokumentasi Microsoft. Skema MVC digunakan untuk memisahkan antara <i>model</i> , <i>view</i> , dan <i>control</i> dari kode program kalkulator itu. <ul style="list-style-type: none">▪ Di bagian Model, penulis membuat <i>get</i> dan <i>set</i> dari Angka1, Angka2, Operator, dan Result.▪ Di bagian View, penulis membuat form sederhana untuk menginputkan angka, memilih operator, tombol, dan pengiriman data ke controller melalui metode POST.

		<ul style="list-style-type: none"> ▪ Di bagian Controller, penulis membuat fungsi untuk menerima data dari metode <i>HttpPost</i> dan melakukan perhitungan. Controller menyimpan hasil perhitungan dalam bentuk <i>Temporary Data</i> dan menampilkannya tanpa melakukan <i>refresh page</i>. ▪ Penulis juga membuat <i>Route</i> supaya aplikasi tersebut bisa diakses melalui pemanggilan URL.
4.	25 Juni 2018	<p>Penulis mempresentasikan kalkulator yang telah penulis buat. Pembelajaran kalkulator sederhana ini dilakukan untuk mengingat kembali sintax bahasa C#, framework .NET, dan markup HTML serta belajar skema MVC.</p>  <p style="text-align: center;">Gambar 2.1.1 – Kalkulator Web</p> <p>Mata kuliah yang pernah penulis ambil adalah <i>Machine Learning</i> dan <i>Internet of Things</i> sehingga penulis sudah meminta Pak Ating untuk memberi proyek yang sesuai bidang penulis (selain <i>software engineering</i>).</p>
5.	26 Juni 2018	<p>Perusahaan sempat ditawari pembangunan chatbot <i>customer service</i>, namun penawaran itu sudah lama. Penawaran yang baru dan ditawarkan ke penulis adalah</p>

		hardware <i>bill acceptor</i> . Modul itu sering dipakai pada mesin kiosk seperti <i>Vending Machine</i> . Fungsi utamanya adalah menerima uang yang masuk dan memindai nominalnya. Maka, penulis ditugaskan untuk melakukan riset mengenai harga dan kualitas <i>bill acceptor</i> dari berbagai vendor yang ada.																																												
6.	27 Juni 2018	<p>Penulis melanjutkan riset barang yang berhubungan dengan <i>bill acceptor</i>. Penulis mendokumentasikan hasil pekerjaan ke dalam bentuk Excel, sehingga mudah untuk melihat hasil komparasi <i>bill acceptor</i>. Berikut hasil komparasi <i>bill acceptor</i> yang penulis cari dari tiga vendor.</p> <p style="text-align: center;">Hasil Komparasi Bill Acceptor</p> <table border="1"> <thead> <tr> <th>Keterangan</th> <th>Bill Acceptor 1</th> <th>Bill Acceptor 2</th> <th>Bill Acceptor 3</th> </tr> </thead> <tbody> <tr> <td>Tipe dan Vendor</td> <td>BV100 / Innovative Technology Ltd</td> <td>ICT L70 / ICT Taiwan Ltd</td> <td>APEX 7000 / Apex Usa Ltd</td> </tr> <tr> <td>Situs Website</td> <td>www.innovative-technology.com</td> <td>www.ictgroup.com.tw</td> <td>www.pyramidacceptors.com</td> </tr> <tr> <td>Harga (2018)</td> <td>\$234</td> <td>\$250</td> <td>\$300</td> </tr> <tr> <td>Power Supply</td> <td>DC 12V</td> <td>DC 12V</td> <td>DC 12V</td> </tr> <tr> <td>Kapasitas Bill Box</td> <td>300/550/800/1050</td> <td>Stackerless (0) / L70-P50 (500 bill)</td> <td>Stackerless (0) / 200 / 500 / 700 / 1000</td> </tr> <tr> <td>Lebar uang</td> <td>66/72 mm</td> <td>67/71 mm</td> <td>up to 72 mm</td> </tr> <tr> <td>Rata-rata Penerimaan</td> <td>99.80%</td> <td>96.00%</td> <td>95.00%</td> </tr> <tr> <td>Dukungan Rupiah</td> <td>Ya</td> <td>Ya</td> <td>Ya</td> </tr> <tr> <td>Interface Logic / Protokol</td> <td>SSP / ccTalk / MDB / Pulse / Parallel / Binary / Serial</td> <td>Pulse / Rs232 / CCNet / MDB</td> <td>Rs232 / IGT / VAC / MDB</td> </tr> <tr> <td>Tahun Rilis</td> <td>2011</td> <td>2010</td> <td>2007</td> </tr> </tbody> </table>	Keterangan	Bill Acceptor 1	Bill Acceptor 2	Bill Acceptor 3	Tipe dan Vendor	BV100 / Innovative Technology Ltd	ICT L70 / ICT Taiwan Ltd	APEX 7000 / Apex Usa Ltd	Situs Website	www.innovative-technology.com	www.ictgroup.com.tw	www.pyramidacceptors.com	Harga (2018)	\$234	\$250	\$300	Power Supply	DC 12V	DC 12V	DC 12V	Kapasitas Bill Box	300/550/800/1050	Stackerless (0) / L70-P50 (500 bill)	Stackerless (0) / 200 / 500 / 700 / 1000	Lebar uang	66/72 mm	67/71 mm	up to 72 mm	Rata-rata Penerimaan	99.80%	96.00%	95.00%	Dukungan Rupiah	Ya	Ya	Ya	Interface Logic / Protokol	SSP / ccTalk / MDB / Pulse / Parallel / Binary / Serial	Pulse / Rs232 / CCNet / MDB	Rs232 / IGT / VAC / MDB	Tahun Rilis	2011	2010	2007
Keterangan	Bill Acceptor 1	Bill Acceptor 2	Bill Acceptor 3																																											
Tipe dan Vendor	BV100 / Innovative Technology Ltd	ICT L70 / ICT Taiwan Ltd	APEX 7000 / Apex Usa Ltd																																											
Situs Website	www.innovative-technology.com	www.ictgroup.com.tw	www.pyramidacceptors.com																																											
Harga (2018)	\$234	\$250	\$300																																											
Power Supply	DC 12V	DC 12V	DC 12V																																											
Kapasitas Bill Box	300/550/800/1050	Stackerless (0) / L70-P50 (500 bill)	Stackerless (0) / 200 / 500 / 700 / 1000																																											
Lebar uang	66/72 mm	67/71 mm	up to 72 mm																																											
Rata-rata Penerimaan	99.80%	96.00%	95.00%																																											
Dukungan Rupiah	Ya	Ya	Ya																																											
Interface Logic / Protokol	SSP / ccTalk / MDB / Pulse / Parallel / Binary / Serial	Pulse / Rs232 / CCNet / MDB	Rs232 / IGT / VAC / MDB																																											
Tahun Rilis	2011	2010	2007																																											

		Deskripsi Produk	<ul style="list-style-type: none"> ■ Integrated standard MDB ■ Proven field reliability ■ Versatile design ■ Easy maintenance, free firmware updates ■ High acceptance rate for multi-currency ■ Integrated USB ■ Direct USB communication & reprogramming 	<ul style="list-style-type: none"> ■ Fixed width/multi-width bill acceptable ■ Multinational currencies acceptable ■ New generation design of verification system ■ Anti-string technology ■ Lightweight and durable plastic construction ■ Speedy bill transaction ■ Selective interfaces ■ Multicolor illumination bezel design ■ Easy maintenance ■ Fast program update 	<ul style="list-style-type: none"> ■ Lighted arrows on bezel and lighted bill entry area. ■ High security against fraudulent bills. ■ Flash downloadable software ■ On-board pushbutton/LED and advanced diagnostics via PC. ■ Many interfaces available ■ Dual-stage optical anti-stringing. ■ Simple configuration and setup ■ Removable bill path for easy cleaning. ■ Auto-calibration
--	--	------------------	---	--	---

Tabel 2.1.2 – Tabel Komparasi Bill Acceptor

Sekitar siang hari penulis diajak untuk bertemu klien di daerah Cengkareng - Kapuk (Pantai Indah Kapuk). Penulis belajar langsung dari Pak Ating untuk *deal* proyek dengan klien dan teknisi IT klien.

Penulis merangkum hasil pertemuan ke dalam *note*. Mesin kiosk mereka berasal dari Singapura. Mereka memiliki banyak cabang dan segera mengadakan mesin kiosk kembali supaya pembayarannya tidak menggunakan kasir. Namun, mereka tidak memiliki banyak waktu untuk *import* dan urusan ke luar negeri rumit bila ada kerusakan pada mesin, contohnya *bill acceptor* tidak memindai uang

		dengan benar, printer nge-jam, dan sebagainya. Mesin kiosk cukup berat sehingga <i>troubleshoot</i> biasanya dilakukan dengan <i>remote</i> dari luar negeri. Oleh karena itu, klien memutuskan untuk membeli buatan dalam negeri saja, sehingga <i>maintenance</i> menjadi lebih mudah.
7.	28 Juni 2018	Penulis tidak bekerja karena sakit. Pak Ating memberi dispensasi pengganti hari Sabtu, 7 Juni 2018 dan berharap bisa menyelesaikan pekerjaan yang belum selesai. Rencananya akan datang orang yang ahli di bidang IoT dan penulis bisa ikut ambil bagian dalam membantu proyek kiosk dengan <i>bill acceptor</i> tersebut.
8.	29 Juni 2018	Penulis masih sanggup masuk walaupun masih kurang enak badan. Penulis tidak banyak bekerja, hanya melanjutkan riset <i>bill acceptor</i> sebelumnya. Setelah bertemu Pak Ating, penulis melakukan riset ke arah penggunaan atau pemrograman <i>bill acceptor</i> . Penulis berhenti melakukan komparasi barang sebab klien menyediakan sampel barang yang bisa dipinjam tim.
9.	30 Juni 2018	Penulis melanjutkan riset pemrograman untuk persiapan proyek kiosk. Hasil yang penulis dapat kebanyakan menggunakan bahasa C untuk memberi perintah ke mesin. Perintah yang dikenali bahasa mesin biasanya berbentuk bit atau byte dan bahasa C dapat memanipulasi data tersebut.
10.	1 Juli 2018	Penulis melanjutkan riset pemrograman di kost. Penulis menyadari bahwa jalur komunikasi ke mesin pada umumnya menggunakan komunikasi serial dengan RS232 yang datanya berbentuk byte.

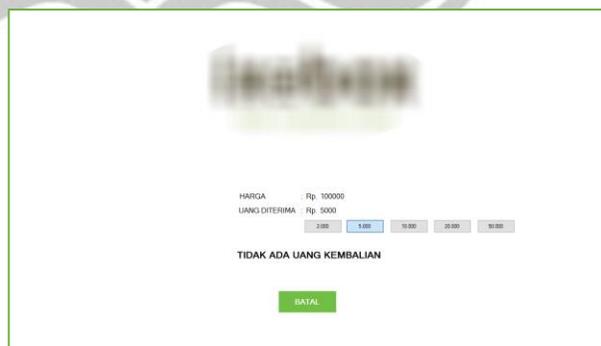
Penulis bertemu Pak Robert Andi Haloho untuk memulai proyek kiosk berbasis IoT. Penulis juga bertemu Adik Dylan Edbert. Kami memulai pekerjaan dengan pertama kali mengenali komunikasi serial (Rs232) dari *bill acceptor* ke aplikasi (desktop).



Gambar 2.1.2 – Menu Awal Aplikasi Kiosk

11. 2 Juli 2018

Penulis menggunakan bahasa C dan membuat GUI program dengan Visual Basic 6. Setelah itu, program diubah ke dalam bahasa C# dengan Visual Studio 2017.

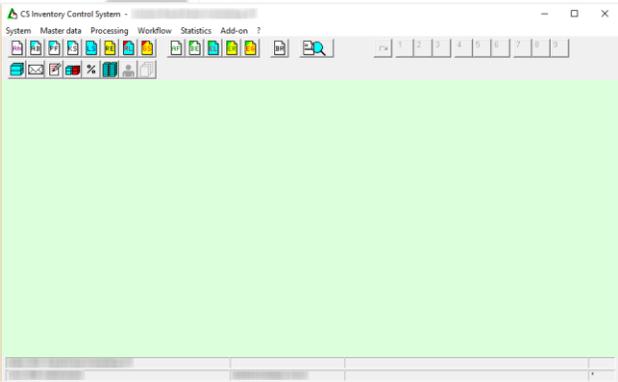


Gambar 2.1.3 – Menu Pembayaran Aplikasi Kiosk

		<p>Program yang kami buat terdiri atas dua fungsional.</p> <ul style="list-style-type: none"> ▪ Fungsi pertama adalah mengenali perintah <i>hexabyte</i> dari / ke <i>bill acceptor</i> yang pada awalnya sudah bisa membaca mata uang rupiah. Semua perintahnya telah tersedia di buku manual ICT Taiwan. Perintah tersebut terdiri dari: <ul style="list-style-type: none"> - Start dan reset <i>bill acceptor</i>. - Mengenali tiap nominal uang rupiah yang mana punya <i>hexabyte</i> masing-masing. - Kondisi <i>enable</i> untuk menerima uang dimana led <i>bill acceptor</i> berubah menjadi warna biru-hijau. Sebaliknya, warna merah artinya <i>bill acceptor</i> berstatus <i>disable</i> sehingga <i>bill acceptor</i> tidak bisa menerima uang. - Kondisi <i>accept</i>, artinya uang yang dimasukkan itu asli, layak diterima, dan nanti masuk ke dalam <i>stacker</i>. Sebaliknya, kondisi <i>reject</i> atau penolakan bisa disebabkan karena uang palsu atau kondisi fisik uang yang sudah rusak atau tidak bisa dibaca <i>bill acceptor</i>. - Kondisi penanganan kesalahan atau <i>exception handling</i> dari <i>bill acceptor</i>. Contohnya kasus uang tertahan (<i>stuck</i>) di dalam <i>bill acceptor</i>. Setelah 5 detik secara otomatis uang itu dikeluarkan dari <i>bill acceptor</i>. Umumnya karena komputer gagal memberi perintah <i>accept</i> ke <i>bill acceptor</i>. Selain itu, ada juga pengecekan respon yang bisa ditangkap dari
--	--	---

		<p>komputer untuk mengetahui jenis error yang dihasilkan <i>bill acceptor</i> dan sebagainya.</p> <ul style="list-style-type: none"> ▪ Fungsi kedua adalah penggunaan fungsi pertama untuk menerima uang dan melakukan pembayaran (transaksi) sederhana pada mesin kiosk tersebut. Pecahan uang yang dimasukkan ke dalam <i>bill acceptor</i> harus memenuhi jumlah tagihan pada layar sehingga memunculkan pop-up print struk.
12.	3 Juli 2018	<p>Penulis melanjutkan proyek kiosk dan membuat program yang betul jadi, karena besok klien menyempatkan waktu untuk melihat <i>progress</i> kami. Tim kami menyempurnakan fungsional dan GUI program, lalu latihan presentasi dengan petinggi Inspiro.</p> <p>Step 1</p> <p>a. Menyalakan Kiosk (Bill Acceptor).</p> <ul style="list-style-type: none"> ▪ Perintah [AC 20], [8FH] diterima komputer dari <i>bill acceptor</i> melalui <i>serial interface</i> Rs232. ▪ Perintah [02H] diterima komputer dari <i>bill acceptor</i> untuk menetapkan bahwa status modulnya OK. ▪ Kalau [26H] artinya ada error sehingga <i>bill acceptor</i> perlu di <i>restart</i>. <p>Step 2</p> <p>a. Bill Acceptor membaca nominal uang</p> <ul style="list-style-type: none"> ▪ Perintah [81H] dari komputer ke <i>bill acceptor</i> untuk memvalidasi uang.

		<ul style="list-style-type: none"> ▪ Jika uangnya asli akan memberikan [40H – 45H] dimana: <ul style="list-style-type: none"> - [40H] : Rp. 1.000,00 - [41H] : Rp. 2.000,00 - [42H] : Rp. 5.000,00 - [43H] : Rp. 10.000,00 - [44H] : Rp. 20.000,00 - [45H] : Rp. 50.000,00 <p>b. Bill Acceptor menerima atau menolak uang</p> <ul style="list-style-type: none"> ▪ Komputer memberi perintah [02H] agar <i>bill acceptor</i> menerima uang ke <i>stacker</i> dan <i>bill acceptor</i> mengirimkan [10H] yang berarti uang berhasil disimpan. ▪ Komputer memberi perintah [0FH] agar <i>bill acceptor</i> untuk menolak / <i>reject</i> uang dan <i>bill acceptor</i> mengirimkan [11H] yang berarti uang itu ditolak atau gagal disimpan ke <i>stacker</i>. <p>Step 3</p> <p>a. Program mengakumulasi uang yang masuk</p> <ul style="list-style-type: none"> ▪ Setelah <i>bill acceptor</i> menerima dan menyimpannya ke <i>stacker</i>, maka nominal uang itu ditambahkan ke variable global <i>total_uang</i> yang nilai awalnya nol dan akumulatif sesuai uang yang masuk. ▪ Di layar akan tampil jumlah uang yang ditagih dan yang sudah masuk. <p>Step 4</p> <p>a. Program menampilkan atau mencetak struk</p>
--	--	--

		<ul style="list-style-type: none"> ▪ Apabila variabel <code>total_uang</code> sama dengan harga yang ditagih, maka program menampilkan struk atau print struk ke <i>printer</i>. ▪ Dengan variable <code>total_uang > tagihan</code>, maka tampil notifikasi <i>Yes/No</i> bahwa uang yang diberikan melebihi apa yang ditagih. <ul style="list-style-type: none"> ▪ Jika dipilih <i>Yes</i>, uang yang diberikan akan di <i>accept</i> dan tiketnya akan diprint. ▪ Jika dipilih <i>No</i>, uang yang diberikan akan dikembalikan sehingga kekurangannya harus ditutup dengan memberi uang pas. ▪ Interval maksimum <i>bill acceptor</i> menahan uang ketika melakukan validasi adalah selama 5 detik. Maka durasi notifikasinya adalah kurang dari 5 detik saja. Jika lebih dari 5 detik maka otomatis uang tersebut akan ditolak / <i>reject</i>.
13.	4 Juli 2018	<p>Tim mempresentasikan program kiosk ke klien. Setelah itu rencananya penulis belajar program CS ERP.</p>  <p style="text-align: center;">Gambar 2.1.4 – Program CS ERP</p>

14.	5 Juli 2018	<p>Penulis diminta Pak Ating untuk membantu Pak Dede Dharma dalam melakukan pengujian produk CS ERP. Posisi penulis sebagai penguji (<i>tester</i>). Penulis bukan hanya mencari kesalahan program tapi juga perlu mengetahui proses bisnis klien dan kenapa <i>test case</i> bisnis klien tidak sesuai dengan program CS ERP.</p>						
15.	6 Juli 2018	<p>Penulis mencari kesalahan program sekaligus mempelajari proses bisnis klien.</p> <p>Penulis juga ikut melihat produk IoT dari peserta magang Bina Nusantara (Binus). Mereka mempresentasikan program IoT <i>smart parking</i> dengan sensor ultrasonik. Menurut Bapak Andi, segala kemungkinan bisa terjadi, contohnya ada orang lain membuka pintu mobil dan orang tersebut terkena deteksi sensor yang seharusnya tidak benar. Maka penempatan sensor, jumlah, dan pemakaian algoritma tertentu perlu diperhatikan.</p>						
16.	7 Juli 2018	<p>Penulis membaca user manual yang diberikan Pak Dede Dharma terkait cara penggunaan program. Di tahap ini penulis mengikuti tutorial yang diajarkan untuk membuat dokumen Purchase Order, Material Request, dan Sales Order.</p> <p>Interaksi User dan Sistem dalam pembuatan Purchase Order adalah sebagai berikut.</p> <table border="1" data-bbox="638 1619 1370 1799"> <thead> <tr> <th data-bbox="638 1619 1016 1661">User</th> <th data-bbox="1016 1619 1370 1661">Sistem</th> </tr> </thead> <tbody> <tr> <td data-bbox="638 1661 1016 1734">User klik ikon Purchase Order</td> <td data-bbox="1016 1661 1370 1734"></td> </tr> <tr> <td data-bbox="638 1734 1016 1799"></td> <td data-bbox="1016 1734 1370 1799">Sistem menampilkan Purchase Order form</td> </tr> </tbody> </table>	User	Sistem	User klik ikon Purchase Order			Sistem menampilkan Purchase Order form
User	Sistem							
User klik ikon Purchase Order								
	Sistem menampilkan Purchase Order form							

		Klik tombol tanda tanya di sebelah field Address No. atau klik field address dan tekan tombol f3 .	
			Sistem menampilkan pop-up list Address.
		Pilih supplier/address lalu klik OK	
			Sistem memasukkan Address yang dipilih ke Address Text Box
		Klik tombol Transact. Items	
			Sistem menampilkan window Transaction Item
		Klik tombol Append	
			Sistem menampilkan Append Form
		Klik A , lalu ketik item di TextBox <i>Item No.</i> dan tekan f3 .	
			Sistem menampilkan list items terkait item yang diketik
		Pilih Item lalu klik tombol OK	
			Sistem kembali ke Append Form
		Ketik Quantity dan Price dari item lalu klik tombol OK	
			Sistem menampilkan item yang dipilih ke dalam list di Transacted Items
		Klik tombol OK	
			Sistem kembali ke Purchase Order form dan menampilkan total price berdasarkan item yang dipesan
		Klik tombol Print Preview untuk menyimpan Purchase Order	
			Sistem menampilkan print out dari Purchase Order
		Click the OK button	

			Sistem kembali ke main page
		Klik tombol Transaction Overview	
			Sistem menampilkan menu transaction overview
		Pastikan memilih opsi " Transaction Type " dibawah " Purchase Orders " lalu klik tombol Search	
			Sistem menampilkan list Purchase Orders
		Di header list, buka header untuk menampilkan heading yang tersembunyi. Buka terus sampai nama header " Approved " muncul	
		Scroll halaman sampai ke akhir halaman dan lihat Purchase Order yang tadi dibuat	
			Status " Approved " seharusnya tidak tercentang supaya bisa diproses oleh pihak yang lebih berwenang

Tabel 2.1.3 – Interaksi Modul Purchase Order

Interaksi User dan Sistem dalam pembuatan Material Request adalah sebagai berikut.

User	Sistem
Klik menu Master Data	
	Sistem menampilkan drop-down list
Pilih Material Request dan klik New	
	Sistem menampilkan window Material Request

		Di box sebelah <i>Item No.</i> , ketik item yang diinginkan lalu tekan f3	
			Sistem menampilkan list items terkait item yang diinputkan
		Klik 2x item yang dipilih	
			Sistem kembali ke halaman <i>Material Request</i> .
		Di box sebelah <i>Address No.</i> , ketik address dan tekan f3	
			Sistem menampilkan list address terkait address yang diinputkan
		Klik 2x address supplier yang dipilih	
			Sistem menampilkan window utama <i>Material Request</i> beserta items dan available suppliers
		Di sebelah TextBox, input nominal Quantity-nya	
			Sistem melakukan update Quantity
		Di TextBox <i>Item price</i> , input nominal price (jika perlu)	
			Sistem melakukan update Price
		Klik tombol Save lalu klik OK	
			Sistem keluar dari window <i>Material Requests</i>
		Klik menu Master Data	
			Sistem menampilkan drop-down list
		Pilih <i>Material Request</i> dan klik Overview	
			Sistem menampilkan window <i>Material Requests Overview</i>
		Klik tombol Search	
			Sistem menampilkan list dari available <i>Material Requests</i> secara menurun

Di header, geser heading setelah price untuk menampilkan status approval	
	Sistem menampilkan status approval dari Material Purchase Form
Cek status approval dari Material Request	
	Sistem menampilkan Material Request yang seharusnya belum di approved
Click OK	
	Sistem kembali ke menu awal

Tabel 2.1.4 – Interaksi Modul Material Request

Interaksi User dan Sistem dalam pembuatan Sales Order adalah sebagai berikut.

User	Sistem
User klik menu Order Confirmation 	
	Sistem menampilkan Order Confirmation Form
Pilih Address (kustomer) dengan klik tombol Question Mark 	
	Sistem menampilkan list Address (Name) beserta data kustomer
Pilih Address/Customer	
Klik tombol OK	
	Sistem menyimpan data customer yang dipilih ke Order Confirmation Form
Klik Transact. Items 	

			Sistem menampilkan Popup Transaction Items
		Klik tombol Append <input type="button" value="Append"/>	
			Sistem menampilkan Append Line Form
		Klik A <input type="text" value="A"/>	
		Klik tombol No di Item No. <input type="text" value="No"/>	
			Sistem menampilkan Item popup
		Pilih Item dan klik tombol OK	
			Sistem menampilkan Item ke field Item
		Isi Quantity dan Price field <input type="text" value="Quantity"/> <input type="text" value="Price"/> <input type="text" value="1"/> <input type="text" value="5000"/>	
		Klik OK	
			Sistem menyimpan data ke Transaction Items
		Klik OK pada menu Transaction Items	
			Sistem menampilkan Order Confirmation Form
		Klik Print Preview untuk menyimpan Transaction data <input type="button" value="Print preview"/>	
			Sistem menampilkan print out dari Order Confirmation dan menyimpan data yang sudah dimasukkan
		Klik ceklis hijau <input checked="" type="checkbox"/>	
			Sistem menampilkan Order Confirmation Form
		Klik tombol OK	
			Sistem menampilkan menu awal
Tabel 2.1.5 – Interaksi Modul Sales Order			
17.	8 Juli 2018	Penulis membaca user manual CS ERP dan melanjutkan tutorial pembuatan Supplier Delivery Note dan Purchase	

Invoice dari Purchase Order (PO), lalu membuat Delivery Note dan Invoice dari Sales Order (SO).

Interaksi User dan Sistem dalam pembuatan Supplier Delivery Note adalah sebagai berikut.

User	Sistem
Klik ikon Transaction Overview 	
	Sistem menampilkan Transaction Overview Form
Pada Transaction Type, pilih Purchase Order	
	Sistem menampilkan Purchase Order ke dalam field
Menginput nama item dan nomor Q	
Klik tombol Search untuk mengecek data Purchase Order	
	Sistem menampilkan data Purchase Order ke dalam list Transaction Overview
Selanjutnya, klik ikon Supplier Delivery Note 	
	Sistem menampilkan Supplier Delivery Note Form (Item Inflow)
Pada form type, pilih Purchase Order dan klik ikon Question Mark 	
	Sistem menampilkan popup Transaction Selection
Klik tombol OK untuk memunculkan data Purchase Order	
	Sistem menutup Transaction Selection Form dan menampilkan data

			Purchase Order ke dalam list Purchase Order
		Pada field To Storage, klik ikon Question Mark 	
			Sistem menampilkan popup Searching Storage Number
		Klik pada data Storage lalu klik tombol OK	
			Sistem menutup popup Searching Storage Number dan menampilkan data Storage ke dalam field To Storage
		Cek jumlah pada field Quantity. Jika jumlah tidak sesuai dengan Purchase Order, isi field Quantity tersebut sesuai dengan Purchase Order.	
		Klik tombol Book Item Inflow	
			Sistem menampilkan popup Item Inflow/outflow
		Klik tombol OK untuk mengahiri proses Supplier Delivery Note	
			Sistem menutup popup Item Inflow/outflow dan kembali ke menu awal

Tabel 2.1.6 – Interaksi Modul Supplier Delivery Note

Interaksi User dan Sistem dalam pembuatan Purchase Invoice adalah sebagai berikut.

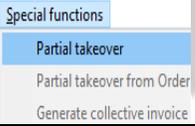
User	Sistem
Klik ikon Purchase Invoice 	
	Sistem menampilkan Invoice Form
Ketik nama supplier pada field Address no. atau klik Question Mark 	

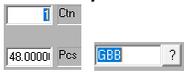
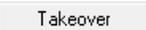
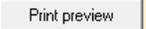
			Sistem menampilkan Match Addresses Form (Name)
		Pilih supplier lalu klik tombol OK	
			Sistem menutup form Match Addresses (Name) dan memasukkan data supplier ke dalam field Address no.
		Pilih menu Special Function	
			Sistem menampilkan opsi dropdown
		Pilih opsi Partial Takeover	
			Sistem menampilkan Match Transactions Form (Address Debtor)
		Pilih data transaksi pada kolom Purchase Delivery Note, lalu tekan tombol OK	
			Sistem menampilkan Partial Takeover Form
		Pilih data yang akan dibuatkan Purchase invoice, lalu klik tombol Takeover	
			Sistem menampilkan data ke dalam kolom Destination Transaction
		Klik tombol OK	
			Sistem menampilkan Match Transaction Form (Creditor)
		Klik tombol Cancel	
			Sistem menutup Match Transaction Form dan kembali ke Invoice Form
		Klik tombol Print Preview	
			Sistem menyimpan data Purchase invoice dan memunculkan Print Preview

Klik ikon ceklis ✓ untuk mengakhiri proses Purchase invoice	
	Sistem menutup Print Preview dan kembali ke menu awal

Tabel 2.1.7 – Interaksi Modul Purchase Invoice

Interaksi User dan Sistem dalam pembuatan Delivery Note adalah sebagai berikut.

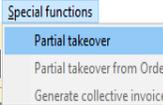
User	Sistem
Klik tombol Delivery Note 	
	Sistem menampilkan Delivery Note form
Ketik nama customer di field Address no. atau klik tombol Question Mark 	
	Sistem menampilkan list Match Addresses (Name) yang berisi data customer
Pilih customer dan klik tombol OK	
	Sistem menampilkan detail data customer ke Delivery Note Form
Pilih Special Functions lalu pilih Partial takeover 	
	Sistem menampilkan Match Transactions (Address Debitor)
Pilih nomor Sales Order yang akan dibuatkan Delivery Order-nya	
Klik tombol OK	

			Sistem menampilkan Partial Takeover Form
		Pilih baris data yang akan dibuatkan Delivery Note. Isi Quantity & storage/gudang. 	
		Klik tombol Takeover. 	
			Sistem memindahkan data ke kolom Destination Transaction
		Klik tombol OK	
			Sistem menampilkan Match Transaction (Address Debitor)
		Klik tombol Cancel	
			Sistem menutup Match Transaction Form dan kembali ke Delivery Note Form
		Klik tombol Print Preview 	
			Sistem menyimpan data Invoice ke Print Preview Form
		Klik ikon ceklis  untuk mengakhiri proses Delivery Note	
			Sistem menutup Print Preview Form dan kembali ke menu awal

Tabel 2.1.8 – Interaksi Modul Delivery Note

Interaksi User dan Sistem dalam pembuatan Invoices (Sales) adalah sebagai berikut.

User	Sistem
Klik ikon Invoice 	

			Sistem menampilkan Invoice Form
		Ketik nama kustomer di field Address no. atau klik tombol Question Mark 	
			Sistem menampilkan list Match Addresses (Name) yang berisi data kustomer
		Pilih customer dan klik tombol OK	
			Sistem menutup popup Match Addresses (Name) dan menampilkan data kustomer ke field Address no.
		Pilih menu Special Function dan pilih opsi Partial Takeover 	
			Sistem menampilkan Match Transactions (Address Debitor)
		Pilih baris data di field Delivery Note, lalu klik tombol OK	
			Sistem menampilkan Partial Takeover Form
		Pilih baris data yang akan dibuatkan Invoice. Isi Quantity & storage/gudang.  Klik tombol Takeover. 	
			Sistem memindahkan data ke kolom Destination Transaction
		Klik tombol OK	

			Sistem menampilkan Match Transaction (Address Debitor)
		Klik tombol Cancel	
			Sistem menutup Match Transaction Form dan kembali ke Invoice Form
		Klik tombol Print Preview 	
			Sistem menyimpan data Invoice data ke Print Preview Form
		Klik tombol ceklis hijau <input checked="" type="checkbox"/> untuk mengakhiri proses pembuatan Invoice	
			Sistem menutup Print Preview Form dan kembali ke menu awal

Tabel 2.1.9 – Interaksi Modul Invoice (Sales)

18.	9 Juli 2018	<p>Ada rapat terbatas antara Tim kami dengan Pak Suharmin dan Pak Ating. Rapat ini membahas perbedaan bisnis proses klien dengan alur program, sehingga pengujian ini bisa menjadikan acuan pemakaian program CS ERP menjadi lebih baik dengan menghindari kesalahan dari program tersebut. Setelah itu, penulis melanjutkan tutorial untuk modul Accounting, Stock Inventory, dan Master Data Item.</p> <p>Interaksi User dan Sistem dalam modul Accounting (Hutang/Piutang) adalah sebagai berikut.</p> <table border="1"> <thead> <tr> <th>User</th> <th>Sistem</th> </tr> </thead> <tbody> <tr> <td>Klik Add – On, pilih dan klik financial Accounting</td> <td></td> </tr> <tr> <td></td> <td>Sistem menampilkan halaman utama financial accounting</td> </tr> </tbody> </table>	User	Sistem	Klik Add – On, pilih dan klik financial Accounting			Sistem menampilkan halaman utama financial accounting
User	Sistem							
Klik Add – On, pilih dan klik financial Accounting								
	Sistem menampilkan halaman utama financial accounting							

	Klik Booking 	
		Sistem menampilkan booking form
	Pilih dan klik 2x data Purchase Invoices (ER) atau Sales Invoices (RE) yang ada pada kolom Accounting of ERP	
		Sistem menampilkan transaksi di column Stack dengan tipe KF jika memilih data Purchase Invoices atau tipe DF jika memilih Sales Invoices
	Pilih transaksi dan klik Booking	
		Sistem menampilkan pop up pesan konfirmasi untuk melakukan pencatatan hutang (KF) atau piutang (DF)
	Klik Yes	
		Sistem menampilkan pop up hasil dari pencatatan hutang (KF) atau piutang (DF)
	Klik OK	

Tabel 2.1.10 – Interaksi Modul Accounting

Interaksi User dan Sistem dalam modul Storages (Add Item) adalah sebagai berikut.

User	Sistem
Klik menu Master Data	
	Sistem menampilkan opsi dropdown
Arahkan mouse ke Storages	
	Sistem menampilkan opsi dari Storages
Pilih opsi item inflow	

			Sistem menampilkan window Item Inflow
		Input item no. pada field item no atau tekan <i>f3</i> untuk memunculkan window Match Items (Name)	
			
			Sistem menampilkan window Match Items (Name)
		Pilih data lalu klik tombol OK	
			Sistem menutup window Match Items (Name) dan menampilkan data pada Item no., On Stock dan To Storage
		Input angka pada field quantity 	
		Klik Book Item Inflow	
			
		Klik OK	
			Sistem menampilkan window Item inflow/outflow
		Klik OK	
			Sistem menampilkan menu awal

Tabel 2.1.11 – Interaksi Modul Storages

Interaksi User dan Sistem dalam modul Master Data (Create Item) adalah sebagai berikut.

User	Sistem
Pilih master data → items → combined	
	Sistem menampilkan form master data item
Input no item pada field Item no.	

		<table border="1"> <tr> <td>Pilih Field Group dengan klik </td> <td></td> </tr> <tr> <td></td> <td>Sistem menampilkan list Grup</td> </tr> <tr> <td>Pilih Group dan klik OK</td> <td></td> </tr> <tr> <td></td> <td>Sistem menampilkan nama group yang dipilih</td> </tr> <tr> <td>Input Nama item pada field text</td> <td></td> </tr> <tr> <td>Pilih tab Properties</td> <td></td> </tr> <tr> <td></td> <td>Sistem menampilkan Field yang ada di Tab Properties</td> </tr> <tr> <td>Pilih Unit pada field Unit dengan klik </td> <td></td> </tr> <tr> <td></td> <td>Sistem menampilkan list unit</td> </tr> <tr> <td>Pilih unit dan klik OK</td> <td></td> </tr> <tr> <td>Pilih storage pada field storage dengan klik </td> <td></td> </tr> <tr> <td></td> <td>Sistem menampilkan list storage</td> </tr> <tr> <td>Pilih storage dan klik OK</td> <td></td> </tr> <tr> <td>Ceklist Stock Item. Klik Save dan OK</td> <td></td> </tr> </table>	Pilih Field Group dengan klik 			Sistem menampilkan list Grup	Pilih Group dan klik OK			Sistem menampilkan nama group yang dipilih	Input Nama item pada field text		Pilih tab Properties			Sistem menampilkan Field yang ada di Tab Properties	Pilih Unit pada field Unit dengan klik 			Sistem menampilkan list unit	Pilih unit dan klik OK		Pilih storage pada field storage dengan klik 			Sistem menampilkan list storage	Pilih storage dan klik OK		Ceklist Stock Item. Klik Save dan OK	
Pilih Field Group dengan klik 																														
	Sistem menampilkan list Grup																													
Pilih Group dan klik OK																														
	Sistem menampilkan nama group yang dipilih																													
Input Nama item pada field text																														
Pilih tab Properties																														
	Sistem menampilkan Field yang ada di Tab Properties																													
Pilih Unit pada field Unit dengan klik 																														
	Sistem menampilkan list unit																													
Pilih unit dan klik OK																														
Pilih storage pada field storage dengan klik 																														
	Sistem menampilkan list storage																													
Pilih storage dan klik OK																														
Ceklist Stock Item. Klik Save dan OK																														
Tabel 2.1.12 – Interaksi Modul Master Data Item																														
19.	10 Juli 2018	Penulis membuat rencana dengan Pak Dede Dharma untuk melakukan pengujian CS ERP. Tujuan pengujian itu adalah mengetahui kekurangan-kekurangan yang muncul jika proses bisnis klien diterapkan pada program CS ERP. Go-Life dijadwalkan pada tanggal 31 Juli 2018 dan batas pengujian maksimal pada tanggal 15 Juli 2018.																												
20.	11 Juli 2018	Setelah belajar semua modul, penulis mulai melakukan pengujian modul. Pengujian pertama diantaranya modul																												

Purchase Order, Supplier Delivery Note, Purchase Invoice, dan Material Request.

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Purchase Order adalah sebagai berikut.

Test Case	Test Result
User memasukkan seluruh informasi yang dibutuhkan dalam pembuatan Purchase Order	Sistem membuat Purchase Order dengan benar, namun ada beberapa letak logo yang tumpang tindih dan menutupi tabel. Hal itu bisa diperbaiki dengan mengubah koordinat logo (X, Y) pada file <i>.lst</i>
User tidak memasukkan informasi yang dibutuhkan dengan tepat dalam pembuatan Purchase Order (No. Address, No. Transacted Items)	Sistem membuat Purchase Order dalam setiap keadaan. Tidak ada persyaratan untuk pembuatannya.
User mencoba untuk menyetujui Purchase Order tanpa memiliki privilege/authorization.	Purchase Order tidak di approved oleh User yang tidak memiliki authorization
Pembuatan Purchase Order dengan diskon	Purchase Order tidak memperlihatkan detail diskon seluruh items, tapi hanya memperlihatkan diskon global
Pembuatan Purchase Order raw material dari Material Request	Purchase Order berhasil terbentuk dengan status belum di approved
Pembuatan Purchase Order <i>finished good</i> dari Material Request	Purchase Order <i>finished goods</i> tidak terbentuk dari Material Request
Item yang akan dibuatkan Purchase Order telah dinonaktifkan (disabled)	Purchase Order tidak terbentuk karena item tidak dapat ditemukan User

User dapat melihat Purchase Order	Purchase Order dapat dilihat oleh user manapun di Transaction Overview
User dapat mengedit Purchase Order	Purchase Order yang telah dibuat User tidak dapat di ubah oleh User lain yang tidak memiliki privilege untuk mengubah Purchase Order
User mengubah Approval Purchase Order	Purchase Order yang telah di approved masih bisa di ubah dan statusnya tetap ter-approved
Pembuatan Purchase Order dengan banyak items	Ketika halaman Purchase Order sampai ke halaman ketiga, list akan tumpang tindih di halaman kedua. (sistem tidak menampilkan dengan baik)
Pembuatan Purchase Order dengan User ID yang salah	User ID yang tidak memiliki authorization tidak bisa mengakses modul Purchase Order (contohnya user dengan privileges untuk Sales)

Tabel 2.1.13 – Pengujian Modul Purchase Order

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Supplier Delivery Note adalah sebagai berikut.

Test Case	Test Result
User membuat Supplier Delivery Note dengan memasukkan seluruh informasi yang diharapkan	Informasi dimasukkan dengan benar
User membuat Supplier Delivery Note dengan memasukkan informasi yang beberapa kosong	Supplier Delivery Note tidak dapat dibentuk tanpa items yang dipilih

Quantity yang dimasukkan User lebih banyak dari yang dipesan di Purchase Order	Supplier Delivery Note masih berhasil dibentuk dengan Quantity yang diterima lebih banyak daripada yang dipesan
Quantity yang dimasukkan User lebih sedikit dari yang dipesan di Purchase Order	Supplier Delivery Note masih berhasil dibentuk dengan Quantity yang diterima lebih sedikit daripada yang dipesan

Tabel 2.1.14 – Pengujian Modul Supplier Delivery Note

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Purchase Invoice adalah sebagai berikut.

Test Case	Test Result
Pembuatan Purchase Invoice dengan memasukkan seluruh informasi yang dibutuhkan	Purchase Invoice berhasil dibentuk
Pembuatan Purchase Invoice dengan beberapa informasi yang kosong	Purchase Invoice tidak dapat dibentuk jika tanpa no address/ no items
Pembuatan Purchase Invoice dengan Quantity yang lebih banyak daripada yang ada di Supplier Delivery Note	Purchase Invoice tidak dapat dibentuk karena items tidak cukup
Pembuatan Purchase Invoice dengan Quantity yang lebih sedikit daripada yang ada di Supplier Delivery Note	Purchase Invoice masih tetap terbentuk dengan jumlah items yang lebih sedikit

Tabel 2.1.15 – Pengujian Modul Purchase Invoice

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Material Request adalah sebagai berikut.

Test Case	Test Result
-----------	-------------

		<table border="1"> <tr> <td>User memasukkan seluruh informasi yang dibutuhkan untuk pembuatan Material Request</td> <td>Sistem berhasil membentuk Material Request</td> </tr> <tr> <td>User memasukkan beberapa information untuk pembuatan Material Request</td> <td>Sistem masih bisa membentuk Material Request</td> </tr> <tr> <td>User memasukkan Quantity yang lebih banyak daripada yang tersedia di stock (storage)</td> <td>Sistem masih bisa membentuk Material Request</td> </tr> <tr> <td>Pengujian user yang memiliki authority untuk melakukan approve Material Request</td> <td>Seluruh User ID dapat melakukan approve Material Request</td> </tr> <tr> <td>Membutuhkan approval untuk edit Material Request</td> <td>Setelah Material Request telah di edit, maka tidak dibutuhkan approval lagi</td> </tr> </table>	User memasukkan seluruh informasi yang dibutuhkan untuk pembuatan Material Request	Sistem berhasil membentuk Material Request	User memasukkan beberapa information untuk pembuatan Material Request	Sistem masih bisa membentuk Material Request	User memasukkan Quantity yang lebih banyak daripada yang tersedia di stock (storage)	Sistem masih bisa membentuk Material Request	Pengujian user yang memiliki authority untuk melakukan approve Material Request	Seluruh User ID dapat melakukan approve Material Request	Membutuhkan approval untuk edit Material Request	Setelah Material Request telah di edit, maka tidak dibutuhkan approval lagi
User memasukkan seluruh informasi yang dibutuhkan untuk pembuatan Material Request	Sistem berhasil membentuk Material Request											
User memasukkan beberapa information untuk pembuatan Material Request	Sistem masih bisa membentuk Material Request											
User memasukkan Quantity yang lebih banyak daripada yang tersedia di stock (storage)	Sistem masih bisa membentuk Material Request											
Pengujian user yang memiliki authority untuk melakukan approve Material Request	Seluruh User ID dapat melakukan approve Material Request											
Membutuhkan approval untuk edit Material Request	Setelah Material Request telah di edit, maka tidak dibutuhkan approval lagi											
		Tabel 2.1.16 – Pengujian Modul Material Request										
21.	12 Juli 2018	<p>Penulis membantu bagian pengujian modul Sales Order, Delivery Note, dan Invoice. Ada rapat terbatas karena klien datang meminta <i>progress</i>, kelemahan apa saja yang sudah kami temukan jika klien memakai program CS ERP. Kami sepakat kesalahan karena bug program di-<i>ignore</i> saja dan fokus mencari ketidaksesuaian program karena proses bisnis klien.</p> <p>Hasil pengujian <i>Test Case</i> yang telah penulis buat untuk modul Sales Order adalah sebagai berikut.</p> <table border="1"> <thead> <tr> <th>Test Case</th> <th>Test Result</th> </tr> </thead> <tbody> <tr> <td>Pembuatan Sales Order dengan memasukkan seluruh informasi yang dibutuhkan</td> <td>Sistem berhasil membentuk Sales Order</td> </tr> <tr> <td>Pembuatan Sales Order dengan tidak memasukkan Quantity atau Price</td> <td>Sistem masih bisa membentuk Sales Order</td> </tr> </tbody> </table>	Test Case	Test Result	Pembuatan Sales Order dengan memasukkan seluruh informasi yang dibutuhkan	Sistem berhasil membentuk Sales Order	Pembuatan Sales Order dengan tidak memasukkan Quantity atau Price	Sistem masih bisa membentuk Sales Order				
Test Case	Test Result											
Pembuatan Sales Order dengan memasukkan seluruh informasi yang dibutuhkan	Sistem berhasil membentuk Sales Order											
Pembuatan Sales Order dengan tidak memasukkan Quantity atau Price	Sistem masih bisa membentuk Sales Order											

			terlepas dari kurangnya informasi yang ada
		User melakukan approve Sales Order dengan authorization yang tepat	Sistem berhasil approve Sales Order
		User melakukan reject Sales Order dengan authorization yang tepat	Sistem berhasil melakukan reject Sales Order
		Pembuatan Sales Order ketika kurang stock pada items	Sistem masih bisa membentuk Sales Order
		Pembuatan Sales Order dengan Credit Limit	Ketika suatu address (perusahaan) mencapai Credit Limit, User tidak dapat membentuk Sales Order dengan address tersebut
		Pembuatan Sales Order dengan Item yang dinonaktifkan (disable)	User tidak dapat membentuk Sales Order
		Pembuatan Sales Order dimana Item memiliki Minimum Price	Bila input price melebihi minimum price Item, maka muncul peringatan " <i>Warning : Price lower than minimum price !</i> ". Apabila dilanjutkan maka Sales Order masih tetap terbentuk.
		Edit Sales Order Setelah Delivery Note tersebut dibentuk	User tidak bisa melakukan update Sales Order ketika Quantity Item lebih besar daripada stock
		Pembuatan Sales Order dengan diskon	Sistem tidak menampilkan diskon per Item namun hanya menampilkan diskon total price
		Pembuatan Sales Order dengan banyak Items	Sistem berhasil membentuk Sales Order dan menampilkan pratinjau dengan baik
Tabel 2.1.17 – Pengujian Modul Sales Order			

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Delivery Note adalah sebagai berikut.

Test Case	Test Result
Pembuatan Delivery Note dengan memasukkan seluruh informasi yang dibutuhkan	Sistem berhasil membentuk Delivery Note
User melakukan approve Delivery Note dengan authorization yang tepat	Sistem berhasil approve Delivery Note
User melakukan reject Sales Order dengan authorization yang tepat	Sistem berhasil reject Delivery Note
User memasukkan Quantity yang lebih sedikit daripada Quantity Sales Order	Sistem berhasil membentuk Delivery Note and mencatat sisa Quantity pada Sales Order
User memasukkan Quantity yang lebih banyak daripada Quantity Sales Order	Sistem tidak berhasil membentuk Delivery Note
Pembuatan Delivery Note dengan Credit Limit untuk Kustomer	Sistem tidak berhasil membentuk Delivery Note karena kustomer melebihi kredit yang telah ditentukan
Storage berbeda ketika pembuatan Delivery Note	Sistem menampilkan peringatan tetapi tetap berhasil membentuk Delivery Note dengan gudang penyimpanan Item yang salah
User melakukan edit Delivery Note setelah Invoice terbentuk	Sistem tidak dapat melakukan edit/update Delivery Note

Tabel 2.1.18 – Pengujian Modul Delivery Note

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Sales Invoice adalah sebagai berikut.

Test Case	Test Result
Creating Invoice (All fields are not filled)	Successful creates Invoice
Creating Invoice (Quantity or Storage fields are not filled)	User cannot create Invoice. Without items or address, it would not create Invoice
Quantity received that is claimed is lower than what was ordered	Successfully creates Invoice and records the remaining Quantity in the Delivery Note
Quantity received that is claimed is higher than what was ordered	User couldn't create Invoice
Invoice fail to be created because Credit Limit Customer	User couldn't create Invoice
Creating an Invoice but Item Storage location is wrong	The system displays a warning but is still successful in creating an Invoice with a wrong Item storage

Tabel 2.1.19 – Pengujian Modul Invoice (Sales)

22.	13 Juli 2018	<p>Penulis melakukan pengujian modul Master Data Item, Storage/Inventory, dan Accounting.</p> <p>Hasil pengujian <i>Test Case</i> yang telah penulis buat untuk modul Master Data Item adalah sebagai berikut.</p>	
		Test Case	Test Result
		Pembuatan Item di Master Data	Sistem berhasil membentuk Item baru
		Menonaktifkan Item di Master Data untuk Sales Order dan Purchase Order	Sistem berhasil menonaktifkan Item yang dipilih pada Master Data sehingga tidak terlihat saat memilih Item tersebut
		Mengaktifkan Item di Master Data untuk Sales Order dan Purchase Order	Sistem berhasil mengaktifkan Item yang dipilih pada Master Data

sehingga tidak terlihat saat memilih Item tersebut

Tabel 2.1.20 – Pengujian Modul Master Data Item

Hasil pengujian *Test Case* yang telah penulis buat untuk modul Storages adalah sebagai berikut.

Test Case	Test Result
User menambahkan stock Item tertentu dan otomatis membuat Supplier Delivery Note	Sistem berhasil menambahkan stock Item dan otomatis membuat Supplier Delivery Note
Sistem mencegah pembuatan Supplier Delivery Note karena data Supplier tidak terisi	User dapat menambahkan stock ke gudang penyimpanannya tanpa data supplier dan tidak mencegah User membentuk Supplier Delivery Note nya
User melakukan Partial Takeover dan mencatatnya dalam satu Delivery Note	Sistem berhasil membentuk Invoice-nya dan mencatat sisa Quantity Delivery Note-nya
(Item Transfer) Transfer Items ke gudang penyimpanan (storage) lainnya dengan mengisi seluruh informasi yang dibutuhkan	Sistem berhasil melakukan transfer stok Items ke storage lainnya dan membentuk Delivery Note-nya
(Item Transfer) Transfer Items ke gudang penyimpanan (storage) lainnya dengan tidak mengisi informasi From/To Storage atau Quantity	Sistem tidak berhasil melakukan transfer item tanpa Supplier atau Quantity , dan mencegah User membentuk Delivery Note
(Item Inflow) Menambahkan Stock dengan memilih Supplier	Stock Item berhasil ditambahkan dengan memilih Supplier dan membentuk Supplier Delivery Note

		(Item Transfer) Transfer kembali Item ke tempat asalnya karena salah pengiriman, cacat produk, dsb	Stocks pada Item berhasil diperbaharui dan membentuk Delivery Note						
		<p>Tabel 2.1.21 – Pengujian Modul Storages</p> <p>Hasil pengujian <i>Test Case</i> yang telah penulis buat untuk modul Accounting adalah sebagai berikut.</p> <table border="1"> <thead> <tr> <th>Test Case</th> <th>Test Result</th> </tr> </thead> <tbody> <tr> <td>Pembuatan Jurnal Payable</td> <td>Berhasil membentuk Jurnal Payable</td> </tr> <tr> <td>Pembuatan Jurnal Receivable</td> <td>Berhasil membentuk Jurnal Receivable</td> </tr> </tbody> </table> <p>Tabel 2.1.22 – Pengujian Modul Accounting</p>		Test Case	Test Result	Pembuatan Jurnal Payable	Berhasil membentuk Jurnal Payable	Pembuatan Jurnal Receivable	Berhasil membentuk Jurnal Receivable
Test Case	Test Result								
Pembuatan Jurnal Payable	Berhasil membentuk Jurnal Payable								
Pembuatan Jurnal Receivable	Berhasil membentuk Jurnal Receivable								
23.	14 Juli 2018	<p>Pak Jackson membantu penulis tentang CS ERP untuk modul <i>accounting</i>, seperti pembuatan COA (<i>Charts Of Accounts</i>). Penulis membuat buku besar sederhana sebagai berikut.</p> <p style="text-align: center;">Studi Kasus</p> <p>Catatan:</p> <ul style="list-style-type: none"> ▪ Dr : <i>for Debit</i>. ▪ Cr : <i>for Credit</i>. ▪ A/P: <i>Account Payable</i>. ▪ A/R: <i>Account Receivable</i>. ▪ Accrue: penyandingan pendapatan dan biaya pada periode terjadinya transaksi. <p>Penerimaan barang dari PT. ABC. Saat tanggal 30 Januari terjadi jurnal, yaitu :</p>							

- Dr Inventory 1.000.000
- Cr A/P Accrue 1.000.000

Penerimaan barang dari PT. XYZ. Saat tanggal 30 Januari terjadi jurnal, yaitu :

- Dr Inventory 2.000.000
- Cr A/P Accrue 2.000.000

Tanggal 05 Februari pada saat terima Invoice sesungguhnya terjadi jurnal yaitu :

- Dr A/P Accrue 1.000.000
- Cr A/P PT.ABC 900.000
- Cr Other Income 100.000

Tanggal 06 Februari pada saat terima Invoice sesungguhnya terjadi jurnal yaitu :

- Dr A/P Accrue 2.000.000
- Cr A/P PT.XYZ 2.200.000
- Dr Additional Cost 200.000

Maka COA untuk pembuatan buku besar (*general ledger*) dari jurnal diatas adalah sebagai berikut.

Account : Inventory (1xxx)

Tanggal	Debet	Kredit	Saldo
30-Jan	1,000,000.00		1,000,000.00
30-Jan	2,000,000.00		3,000,000.00

Account: A/P PT ABC(2101)

Tanggal	Debet	Kredit	Saldo
30-Jan		1,000,000.00	1,000,000.00
30-Jan		2,000,000.00	3,000,000.00
5-Feb	1,000,000.00		2,000,000.00
6-Feb	2,000,000.00		-

		<p>Account: A/P PT ABC(2101)</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Debet</th> <th>Kredit</th> <th>Saldo</th> </tr> </thead> <tbody> <tr> <td>5-Feb</td> <td></td> <td>900,000.00</td> <td>900,000.00</td> </tr> </tbody> </table> <p>Account: A/P PT XYZ (2102)</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Debet</th> <th>Kredit</th> <th>Saldo</th> </tr> </thead> <tbody> <tr> <td>6-Feb</td> <td></td> <td>2,200,000.00</td> <td>2,200,000.00</td> </tr> </tbody> </table> <p>Account: Other Income (5001)</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Debet</th> <th>Kredit</th> <th>Saldo</th> </tr> </thead> <tbody> <tr> <td>5-Feb</td> <td></td> <td>100,000.00</td> <td>100,000.00</td> </tr> </tbody> </table> <p>Account: Additional Cost (6001)</p> <table border="1"> <thead> <tr> <th>Tanggal</th> <th>Debet</th> <th>Kredit</th> <th>Saldo</th> </tr> </thead> <tbody> <tr> <td>6-Feb</td> <td>200,000.00</td> <td></td> <td>200,000.00</td> </tr> </tbody> </table> <p>Tabel 2.1.23 – COA Untuk Pembuatan Buku Besar</p>	Tanggal	Debet	Kredit	Saldo	5-Feb		900,000.00	900,000.00	Tanggal	Debet	Kredit	Saldo	6-Feb		2,200,000.00	2,200,000.00	Tanggal	Debet	Kredit	Saldo	5-Feb		100,000.00	100,000.00	Tanggal	Debet	Kredit	Saldo	6-Feb	200,000.00		200,000.00
Tanggal	Debet	Kredit	Saldo																															
5-Feb		900,000.00	900,000.00																															
Tanggal	Debet	Kredit	Saldo																															
6-Feb		2,200,000.00	2,200,000.00																															
Tanggal	Debet	Kredit	Saldo																															
5-Feb		100,000.00	100,000.00																															
Tanggal	Debet	Kredit	Saldo																															
6-Feb	200,000.00		200,000.00																															
24.	15 Juli 2018	Penulis membantu merapikan dokumentasi yang terdiri atas kumpulan <i>test case</i> dari modul-modul yang ada pada program CS ERP. Dokumentasi oleh Dylan dan penulis kemudian dijadikan satu dan penulis kirim hasilnya ke Pak Dede Darma dan Pak Suharmin.																																
25.	16 Juli 2018	Penulis membantu Pak Andi terkait masalah proyek IoT lainnya. Lingkup proyeknya cukup besar, yaitu mengontrol ruangan dan komunikasi antar gedung dengan modul LoRa. LoRa (Long Range) adalah suatu format modulasi FM untuk komunikasi data <i>wireless</i> yang dibuat oleh Semtech. LoRa banyak digunakan di industri dan area rural untuk jaringan IoT. Fitur utamanya adalah jarak konektivitasnya yang jauh (100 km) dan berdaya rendah.																																
26.	17 Juli 2018	Penulis hadir dalam <i>meeting</i> dan membantu Pak Andi untuk mengkomunikasikan Arduino dan program .NET																																

		<p>dalam rangka membantu skripsi peserta magang Binus. Edellyn mempresentasikan API tentang <i>smart parking</i>, namun dia memiliki kendala komunikasi dari Arduino ke program API. Penulis membantu mereka terkait kode program Arduino, yaitu mengirimkan data yang diperoleh sensor ultrasonik ke komputer dengan protokol <i>http</i> dan menampilkannya secara <i>realtime</i> melalui web browser. Lalu mereka dibantu program API-nya oleh senior .NET.</p>
27.	18 Juli 2018	<p>Penulis membantu Pak Andi tentang proyek kiosk kembali. Riset <i>bill acceptor</i> yang didemokan kemarin kami serahkan kepada developer lain untuk dibuatkan <i>Cashless Payment</i>. Tim Developer IoT mendapat pekerjaan baru untuk membuat sistem <i>antifraud</i>. Sistem itu digunakan untuk mencegah kecurangan penggunaan listrik oleh <i>Hair Stylist</i> (HS). Maka kami membuat rencana kerja dahulu sebelum memulai proyek tersebut.</p>
28.	19 Juli 2018	<p>Penulis dan Tim Developer IoT memulai pengerjaan sistem <i>antifraud</i>. Pertama, kami melakukan tes barang-barang IoT yang telah diadakan oleh perusahaan yaitu Arduino Uno dan LED.</p> <p>Arduino Uno kami gunakan sebagai <i>microcontroller</i> yang membaca sensor / modul serta mengerjakan perintah yang diberikan dari program desktop.</p> <p>LED kami gunakan sebagai simulasi perintah dari Arduino ke sensor/modul.</p>
29.	20 Juli 2018	<p>Penulis belanja alat IoT yang masih kurang dengan Pak Andi. Kami membeli komponen lain seperti keypad 4x4,</p>

		<p>buzzer, sensor ultrasonik, dan LED. Setelah belanja, kami menguji kondisi barang dengan koding di Arduino IDE.</p> <p>Penulis membuat program untuk mengetahui jarak objek dengan sensor melalui sensor ultrasonik. Cara menghitung nilai jarak dengan sensor ultrasonik secara manual adalah sebagai berikut.</p> $S = 340 \times \frac{t}{2}$ <ul style="list-style-type: none"> ▪ S (m) adalah jarak dari sensor ke objek. ▪ 340 (m/s) adalah kecepatan suara. ▪ t (sekon) adalah selisih antara waktu pemancaran gelombang oleh <i>transmitter</i> dan waktu ketika gelombang pantul diterima <i>receiver</i>. ▪ Konstanta 2 adalah waktu yang dibutuhkan secara bolak-balik, karena hanya perlu mengetahui jarak dari sensor ke objek dan bukan jarak bolak-balik. <p>Contoh :</p> <p>Hitung jarak dimana diketahui selisih waktu pemancaran gelombang oleh <i>transmitter</i> dan waktu ketika gelombang pantul diterima <i>receiver</i> adalah 1.000.000 mikro sekon.</p> $S=340*(1.000.000/2)$ $S=340 * (1/2)$ $S=170 \text{ m}$
30.	21 Juli 2018	<p>Penulis menyelesaikan simulasi awal sistem <i>antifraud</i> dengan barang IoT yang sudah ada.</p> <p>Lampu 220v AC masih kami simulasikan dengan LED dan modul buzzer dipakai untuk suara notifikasi dari sistem.</p>

		<p>Sensor ultrasonik digunakan untuk menangkap jarak objek dari sensor. Datanya masih <i>dummy</i> untuk simulasi pemindaian tiket dengan <i>scanner</i> di kiosk. Keypad 4x4 digunakan untuk menginputkan data oleh user, seperti input nomor <i>id stylist</i>. Hal itu akan memenuhi kebutuhan dua prasyarat sebelum listrik dinyalakan.</p>
31.	22 Juli 2018	<p>Penulis mencoba barang IoT milik penulis (NodeMCU) sekaligus menyelesaikan program yang belum selesai. Penulis mengerjakannya di Arduino IDE dengan bahasa C.</p>
32.	23 Juli 2018	<p>Penulis mengikuti rencana kerja yang telah dibuat yaitu melakukan demo sistem <i>antifraud</i> ke Pak Ating. Alur program yang kami demokan adalah sebagai berikut.</p> <ol style="list-style-type: none"> a. Arduino dan seluruh modul/sensor sudah menyala. Kondisi keypad 4x4 di awal akan di <i>disable</i> sehingga tidak bisa ditekan oleh user. b. Arduino membaca data pada sensor ultrasonik secara terus menerus dan menghitung nilai jarak objek. Jika jaraknya kurang dari 10 cm maka kondisi pertama menyalakan listrik terpenuhi. c. Keypad 4x4 akan aktif ketika kondisi pertama terpenuhi. Jika kondisi pertama tidak terpenuhi maka keypad 4x4 terus di <i>disable</i> oleh program. d. User menekan tombol sesuai nomor stylist (1 digit saja). Setelah itu, kondisi kedua terpenuhi. e. Program akan mengecek kondisi pertama dan kedua terpenuhi. Jika terpenuhi, maka akan menyalakan listrik dari Arduino ke relay. Hal itu masih <i>dummy</i>

		<p>dengan menyalakan modul buzzer yang menandakan bahwa listrik sedang menyala.</p> <p>Setelah itu, Pak Andi pergi ke tempat klien dengan agenda membahas proyek <i>monitoring storage</i> dengan LoRa.</p>
33.	24 Juli 2018	Izin bimbingan KRS dan skripsi.
34.	25 Juli 2018	Izin bimbingan skripsi.
35.	26 Juli 2018	<p>Penulis menggabungkan <i>prototype</i> sistem <i>antifraud</i> dan keypad yang bisa input banyak karakter. Program akan menunggu (<i>wait</i>) keypad sampai ditekan tombol “A” untuk mengakhiri inputan dari user. Setelah itu, penulis mengintegrasikan hasil koding di Arduino (<i>client</i>) dan memakai <i>Windows Form Application</i> untuk mengolah komunikasi Serial tersebut di komputer (<i>server</i>).</p>
36.	27 Juli 2018	<p>Penulis melakukan komunikasi antara Arduino dan komputer. Sesuai rencana kerja, penulis mengerjakan program menerima data dari keypad ke Arduino lalu menuju komputer. Rencananya, data yang diterima adalah nomor <i>id stylist</i> dan status relay box di tiap kursi.</p> <p>Di komputer sudah ada database <i>id stylist</i> sehingga pengecekan dilakukan dengan membandingkan data yang diterima dari Arduino dengan isi database. Komunikasi Serial digunakan oleh penulis supaya konektivitasnya stabil melalui kabel serial usb.</p>
37.	28 Juli 2018	<p>Penulis melanjutkan pekerjaan sesuai dengan rencana kerja, yaitu mengirimkan data dari komputer ke Arduino melalui aplikasi desktop dengan framework .NET.</p> <p>Rencananya, data yang dikirimkan adalah karakter perintah tertentu yang dikenali Arduino.</p>

38.	29 Juli 2018	Penulis mencoba fungsi barang IoT penulis (NodeMCU).
39.	30 Juli 2018	Izin bimbingan skripsi.
40.	31 Juli 2018	Izin bimbingan skripsi.
41.	1 Agustus 2018	Penulis mengirim <i>progress</i> program <i>antifraud</i> untuk menerima data dari Arduino ke aplikasi desktop. Ada komponen SerialPort di Visual Studio 2017 dan event DataReceived untuk menerima data yang masuk ke dalam string. Pada string itu sudah ditambahkan <i>header</i> untuk mengenali jenis data yang dikirimkan. Contohnya untuk string “K10101” diawali karakter “K” yang dikenali sebagai data untuk status relay box. Maka bisa dilakukan <i>parsing</i> untuk lima karakter setelah karakter “K” sebagai status lima kursi (0=tersedia, 1=terpakai).
42.	2 Agustus 2018	Penulis membuat <i>interrupt handler</i> pada program. Interrupt handler digunakan untuk mengatur interupsi perintah luar dari <i>hardware</i> ketika proses tertentu sedang berjalan di Arduino. Pin Arduino Uno yang bisa diberikan interupsi adalah pin nomor 2 dan 3. Interupsi itu ketika keypad ditekan maka Arduino akan menjalankan <i>task</i> tertentu. Contohnya membaca inputan keypad ketika program sedang membaca status relay box setiap kursi.
43.	3 Agustus 2018	Penulis melanjutkan <i>progress</i> program <i>antifraud</i> dengan <i>interrupt handler</i> . Di Arduino IDE terdapat fungsi bawaan untuk memanggil prosedur/fungsi tertentu ketika ada <i>trigger</i> di pin interupt. Contohnya pin 2 atau 3 dihubungkan ke keypad dan ketika keypad ditekan maka <i>state</i> dari pin itu berubah dari <i>low</i> ke <i>high</i> dan sebaliknya. Perubahan ini

		diketahui oleh Arduino dan segera melakukan interupsi dengan memanggil prosedur/fungsi yang sudah ditentukan.				
44.	4 Agustus 2018	Penulis mendapatkan tugas dari Pak Arvin untuk membuat alur dari sistem <i>antifraud</i> (<i>flowchart</i>). Tujuannya untuk menyamakan alur sistem dengan koding setiap case.				
45.	5 Agustus 2018	Penulis melanjutkan tugas untuk membuat alur program dari sistem <i>antifraud</i> (<i>flowchart</i>).				
46.	6 Agustus 2018	Penulis mendapatkan perbaikan dari Pak Arvin untuk <i>flowchart</i> program <i>antifraud</i> dengan case supaya mudah mencocokkan alur pada <i>case</i> dengan koding Arduino. Ada tambahan LCD Display untuk melihat respon dari Arduino secara visual.				
47.	7 Agustus 2018	Penulis memperbaiki alur dan penamaan case (A-E). <ul style="list-style-type: none"> ▪ Case A: <i>display</i> ke layar LCD “<i>Tiket tidak valid</i>”. ▪ Case B: <i>display</i> ke layar LCD “<i>Silahkan input Id Stylist</i>”. ▪ Case C: <i>display</i> ke layar LCD “<i>Id Stylist tidak valid</i>”. ▪ Case D: Cek status kursi → aktifkan keypad → input nomor kursi → cek inputan nomor → kirim ke desktop → save ke Database. ▪ Case E: Aktifkan relay di <i>box</i> kursi yg dipilih. 				
48.	8 Agustus 2018	<table border="1"> <thead> <tr> <th>Pertanyaan/task</th> <th>Jawaban</th> </tr> </thead> <tbody> <tr> <td>Apakah baca keypad itu berulang terus atau menunggu (loop/wait)?</td> <td>Berulang ketika memakai <i>while</i> karena fungsi <code>myKeypad.getKey()</code></td> </tr> </tbody> </table>	Pertanyaan/task	Jawaban	Apakah baca keypad itu berulang terus atau menunggu (loop/wait)?	Berulang ketika memakai <i>while</i> karena fungsi <code>myKeypad.getKey()</code>
Pertanyaan/task	Jawaban					
Apakah baca keypad itu berulang terus atau menunggu (loop/wait)?	Berulang ketika memakai <i>while</i> karena fungsi <code>myKeypad.getKey()</code>					

			tidak menahan <i>loop</i> .
		Apakah flow sudah dirapikan?	Sudah dirapikan.
		Apakah sudah dibuatkan cek kursi yang <i>available</i> dengan batas maksimum 5 box kursi?	Sudah dibuatkan.
		Bagaimana statusnya bila gagal menyimpan ke database?	Belum, sedang dipikirkan.
		Bagaimana fungsi <i>interrupt</i> -nya?	Sudah bisa <i>interrupt</i> secara <i>hardware</i> .
		Bagaimana alternatif input Id Stylist dengan <i>scanner</i> ?	Rencananya dengan keypad.

Tabel 2.1.24 – Diskusi Program Antifraud

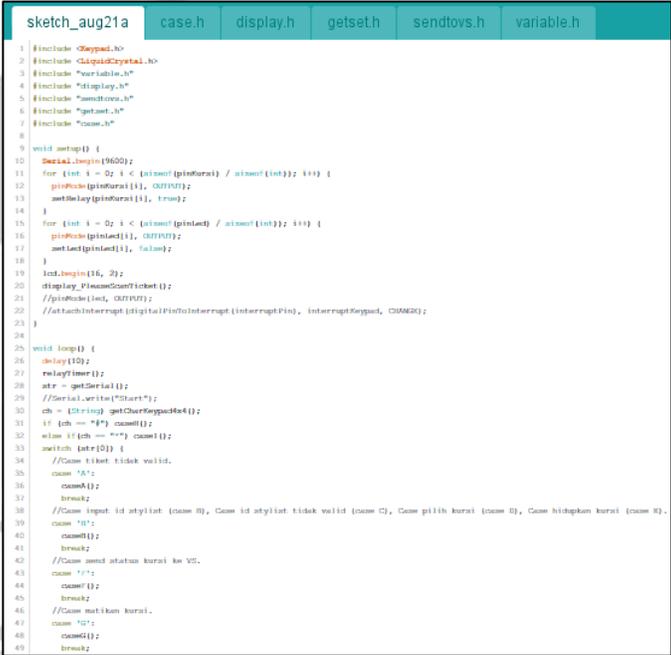
Perbaiki *flowchart* sesuai case.

- Case A: *display* ke layar LCD “*Tiket tidak valid*”.
- Case B: *display* ke layar LCD “*Silahkan input Id Stylist*”.
- Case C: *display* ke layar LCD “*Id Stylist tidak valid*”.
- Case D: Cek status kursi → aktifkan keypad → input nomor kursi → cek inputan nomor → kirim ke desktop → save ke Database.
- Case E: Aktifkan relay di *box* kursi yg dipilih.
- Case F: Kirim status kursi ke aplikasi (*realtime*).

		<ul style="list-style-type: none"> Case G: Mematikan relay <i>box</i> kursi yang dipilih untuk keadaan <i>emergency</i>. Biasanya, relay <i>box</i> kursi mati secara otomatis dengan <i>milis timer</i>.
49.	9 Agustus 2018	Penulis menambahkan case E untuk menghidupkan relay <i>box</i> kursi. Penulis menambahkan <i>header</i> lainnya saat kirim atau terima data ke Serial supaya memudahkan program dalam mengenali jenis data apa yang diberikan dari Serial.
50.	10 Agustus 2018	Penulis menambahkan respon yang tepat untuk ditampilkan ke LCD Display. Di aplikasi desktop Visual Studio, penulis berhasil melakukan <i>parsing</i> data dari <i>header</i> tambahan dan melakukan <i>update</i> ke database.
51.	11 Agustus 2018	Penulis melakukan pengecekan barang dan melakukan peminjaman barang Arduino untuk mengujinya di kost.
52.	12 Agustus 2018	Penulis melakukan pengujian program di kost.
53.	13 Agustus 2018	Penulis menggabung dan memperbaiki case yang sudah ada. Penulis mengganti alat dari Arduino Uno ke Mega karena jumlah pin Mega lebih banyak dari Uno. Hanya beberapa kode program yang perlu disesuaikan. Penulis menunggu <i>relay-timer (hardware)</i> untuk <i>countdown</i> otomatis selama 25 menit yang merupakan durasi waktu seorang <i>Hair Stylist</i> .
54.	14 Agustus 2018	Penulis menambahkan <i>alternative flow</i> seperti keypad yang digunakan untuk input Id Stylist. Ada penambahan kode program di aplikasi desktop untuk mengecek Id Stylist itu valid atau tidak. Alat <i>relay-timer</i> sudah dicoba, hasilnya yaitu alat pertama rusak dan alat kedua belum diketahui cara memberi perintahnya dari Arduino.

55.	15 Agustus 2018	Penulis melakukan <i>debugging</i> error. Beberapa kode program di setiap case telah digabung dan sesuai flow. Namun komunikasi antara aplikasi desktop dengan Arduino kadang berjalan kadang tidak. Maka perlu perubahan arsitektur kode program baik di Arduino dan aplikasi desktop.
56.	16 Agustus 2018	Penulis mencoba solusi <i>multi-serial port</i> yang ternyata tidak bisa dilakukan karena membutuhkan port yang lebih dari satu, sedangkan kami hanya perlu satu port saja dan bekerja secara paralel (<i>multi-threading</i>). Maka fungsi untuk case cek <i>box</i> kursi secara <i>realtime</i> itu dimatikan dahulu, karena bisa terjadi bentrok komunikasi di SerialPort yang sama. Maka penulis perlu mencari solusi dari kasus tersebut.
57.	17 Agustus 2018	Libur Hari Raya Kemerdekaan Indonesia.
58.	18 Agustus 2018	Penulis menguji coba fungsi case B, C, dan D di kost.
59.	19 Agustus 2018	Penulis menguji coba fungsi case B, C, dan D di kost.
60.	20 Agustus 2018	Penulis mendemokan hasil pekerjaan ke Pak Arvin. Pak Arvin telah melihat hasilnya dan Tim kami disuruh untuk membuat <i>pseudocode</i> -nya. Semua kode program di tiap case sudah berjalan, namun tampaknya belum sesuai dengan <i>flowchart</i> . Maka perlu perbaikan dan harus sejalan dengan <i>flowchart</i> yg merupakan dasar acuan tim untuk mengembangkan sistem <i>antifraud</i> . Semuanya sistem harus jelas sebab sistemnya akan terintegrasi dengan program lain yaitu <i>Cashless Payment</i> .

61.	21 Agustus 2018	<p>Penulis menemukan solusi <i>relay-timer</i> menggunakan <i>milis</i> (secara software) untuk mematikan <i>box</i> kursi ketika sudah 25 menit dan secara bersamaan menghidupkan <i>box</i> kursi ketika ada pelanggan baru.</p> <p>Seharusnya pembuatan <i>pseudocode</i> dilakukan sebelum <i>coding</i>, sehingga tidak banyak revisi di program Arduino dan aplikasi desktop. Pak Arvin memberitahu bahwa proses <i>flowchart</i> harus sesuai dengan kode program.</p>
62.	22 Agustus 2018	Libur Hari Raya Idul Adha.
63.	23 Agustus 2018	<p>Penulis mengikuti rapat internal sebelum Tim kami menunjukkan produk <i>antifraud</i> ke klien. Penulis telah membuat <i>Minutes of Meeting</i> atau Notulensi Rapat. Contohnya, pengemasan produk supaya terlihat rapi dan kabelnya yang tidak teratur bisa disembunyikan. Keamanan juga diutamakan karena relay tersambung dengan listrik 220v AC dan jangan sampai tersengat listrik.</p>
64.	24 Agustus 2018	Penulis memperbaiki case untuk input Id Stylist. Ada 2 cara, yaitu dari aplikasi (<i>main</i>) atau keypad (<i>alternative</i>)
65.	25 Agustus 2018	Penulis memperbaiki kode program untuk case yang belum selesai di kost.
66.	26 Agustus 2018	Penulis memperbaiki kode program untuk case yang belum selesai di kost.
67.	27 Agustus 2018	<p>Penulis melakukan pengujian aplikasi <i>antifraud</i> dengan tiga nomor <i>QR-Code</i> yang tersimpan di database, hidupkan <i>box</i> kursi, dan menambahkan waktu 5 menit (<i>extra time</i>).</p> <p>Ada penambahan logika pada case baru, yaitu jika customer sudah selesai melakukan <i>haircutting</i>, namun masih ada</p>

		waktu tersisa, maka Stylist bisa mematikan <i>box</i> kursi tersebut melalui keypad.
68.	28 Agustus 2018	<p>Penulis ikut rapat dengan klien proyek sistem <i>antifraud</i> pada jam 10.00 WIB bersama Bapak Alex. Pertama, kami presentasikan garis besar alur yang dibuat dan menunjukkan hasil produk berupa simulasi/demo.</p>  <pre> 1 #include <Keypad.h> 2 #include <LiquidCrystal.h> 3 #include "variable.h" 4 #include "display.h" 5 #include "sendtovs.h" 6 #include "getset.h" 7 #include "case.h" 8 9 void setup() { 10 Serial.begin(9600); 11 for (int i = 0; i < (sizeof(pinturani) / sizeof(int)); i++) { 12 pinMode(pinturani[i], OUTPUT); 13 digitalWrite(pinturani[i], true); 14 } 15 for (int i = 0; i < (sizeof(pinled) / sizeof(int)); i++) { 16 pinMode(pinled[i], OUTPUT); 17 digitalWrite(pinled[i], false); 18 } 19 lcd.begin(16, 2); 20 display_PleaseScanTicket(); 21 //pinMode(led, OUTPUT); 22 //attachInterrupt(digitalPinToInterrupt(interruptPin), interruptKeypad, CHANGE); 23 } 24 25 void loop() { 26 delay(10); 27 relayTimer(); 28 str = getSerial(); 29 //Serial.println(str); 30 ch = (str[0] - '0') % 10; 31 if (ch == '0') case0(); 32 else if (ch == '*') case1(); 33 switch (getP()) { 34 //Case tiket tidak valid. 35 case 'A': 36 caseA(); 37 break; 38 //Case input id stylist (case B), Case id stylist tidak valid (case C), Case pilih kursi (case D), Case hidpihan kursi (case E). 39 case 'B': 40 caseB(); 41 break; 42 //Case send status kursi ke VS. 43 case 'C': 44 caseC(); 45 break; 46 //Case matikan kursi. 47 case 'D': 48 caseD(); 49 break; </pre>
		Gambar 2.1.5 – Kode Main Program Sistem Antifraud
69.	29 Agustus 2018	Penulis menguji ketahanan Arduino dengan mencoloknya ke listrik 220v AC selama satu hari. Penulis lalu membuat dokumentasi berupa <i>wiring diagram</i> dan data yang dibutuhkan Arduino.
70.	30 Agustus 2018	<p>Penulis mendapatkan solusi yg diberikan oleh Pak Winata untuk Case Failover perlu dua IP publik (statis).</p> <ul style="list-style-type: none"> ▪ Satu IP untuk monitoring Arduino <i>master</i> dan satu IP untuk monitoring Arduino <i>slave</i>.

		<ul style="list-style-type: none"> ▪ Perlu input token dari keypad. ▪ Di pusat akan memberikan nomor token untuk diinputkan ke keypad agar Arduino <i>slave</i> bisa berjalan. ▪ Perlu fitur dari sensor PIR untuk mendeteksi ada orang atau tidak yang dipasangkan di tiap <i>box</i> kursi. ▪ Arduino <i>master</i> menggunakan jaringan biasa. ▪ Arduino <i>slave</i> menggunakan jaringan dari modem. ▪ Setiap <i>box</i> kursi berisi <i>emergency button</i> untuk menghidupkan atau mematikan relay.
71.	31 Agustus 2018	<p>Penulis ikut rapat di tempat klien proyek sistem <i>antifraud</i> dengan hasil rapatnya sebagai berikut.</p> <ul style="list-style-type: none"> ▪ Adanya <i>feedback</i> dari Arduino untuk mengetahui apakah Arduino itu berfungsi. Jika tidak berfungsi, perintah akan beralih ke Arduino cadangan. ▪ Memakai WiFi dan Router untuk komunikasi Arduino jika <i>Hair Stylist</i> jadi memakai <i>smartdevice</i> android. ▪ Memakai dua IP untuk identifikasi Arduino utama dan cadangan. ▪ Keypad tetap dipakai untuk keadaan <i>emergency</i> seperti mematikan/menghidupkan <i>box</i> kursi. ▪ Keputusan memakai <i>smartdevice</i> atau scanner masih menunggu keputusan dari manajemen klien. ▪ Data yang dikirimkan ke Arduino adalah nomor kursi dan durasi waktu nyala <i>box</i> kursi. Format data akan disamakan, sebisa mungkin sudah

		dipersiapkan atau di laporkan ke pihak yang mengerjakannya.
72.	1 September 2018	Penulis memperbaiki program di kost.
73.	2 September 2018	Penulis memperbaiki program di kost.
74.	3 September 2018	<p>Progress antifraud dengan modul WiFi + Arduino Mega.</p> <ul style="list-style-type: none"> ▪ Berhasil menghubungkan modul WiFi ESP01 (<i>client</i>) dengan <i>Access Point</i>. ▪ Berhasil membuat Web Server di Arduino Mega. ▪ Webserver digunakan untuk mematikan / menghidupkan relay yang ada di <i>box</i> kursi melalui web. Rencananya hasil yang dibalikkan berbentuk JSON, namun sekarang masih berbentuk HTML. ▪ Tim sedang mencoba Arduino Wemos D1 yang sudah <i>built-in</i> modul WiFi.
75.	4 September 2018	Penulis melanjutkan progress antifraud dengan modul WiFi + Arduino Mega. Penulis berhasil mengkoneksikan Web Server dan Keypad untuk kontrol <i>box</i> kursi saat <i>emergency</i> .
76.	5 September 2018	Penulis melanjutkan progress antifraud dengan modul WiFi + Arduino Mega. Penulis bersama Reandro berhasil membuat <i>URL</i> statis, yaitu alamat untuk menghidupkan atau mematikan relay <i>box</i> kursi melalui AT Command.
77.	6 September 2018	<p>Penulis melanjutkan riset untuk AT Command. Ada beberapa command seperti mengecek status modul WiFi, <i>connect</i> ke <i>Access Point</i>, <i>disconnect</i> dari <i>Access Point</i>, merubah mode Client/AP/Both, dan reset setting ke semula.</p> <p>Penulis dan Reandro juga berhasil mematikan atau menghidupkan relay pada tiap <i>box</i> kursi dengan mengakses URL-nya. Jadi, semua orang bisa mengakses URL tersebut</p>

		dari browser dengan syarat masih dalam satu jaringan koneksi sehingga bisa memberikan perintah lewat WiFi.
78.	7 September 2018	<p>Penulis melanjutkan riset untuk AT Command dengan cara memasukkan command tersebut secara <i>hardcode</i>. Beberapa command berhasil dilakukan pada serial console di IDE Arduino. Namun masalah terjadi saat penulis mematikan relay <i>box</i> kursi tertentu yang terkadang perintah itu berhasil atau gagal dilakukan, seperti ada <i>delay</i> disitu.</p> <p>Penulis segera menanyakan masalah ini kepada Pak Arvin sehingga Tim kami mendemokan hasil produk dan menampilkan masalah kemarin. Setelah dirapatkan, Tim kami masih belum menemukan solusinya sehingga kami ditugaskan untuk memakai <i>device</i> yang sudah <i>built-in</i> WiFi seperti Wemos D1 dan NodeMCU.</p>
79.	8 September 2018	Penulis melanjutkan riset dengan NodeMCU.
80.	9 September 2018	Penulis melanjutkan riset dengan NodeMCU.
81.	10 September 2018	Penulis mengganti <i>device</i> Mega 2560 + modul WiFi eksternal dengan NodeMCU. Penulis mengkondisikan kode program agar berjalan dengan library dari NodeMCU.
82.	11 September 2018	Libur Hari Raya Tahun Baru Islam.
83.	12 September 2018	Penulis mendapatkan penjelasan alur pengiriman data secara <i>realtime</i> dari NodeMCU ke <i>cloud server</i> dengan pemrograman socket. Maka penulis membuat <i>server</i> berupa aplikasi console dengan bahasa C# dan <i>client</i> nya adalah NodeMCU. Server yang digunakan masih jaringan lokal dan bisa diakses oleh <i>host</i> yang masih dalam satu jaringan router.

84.	13 September 2018	Penulis berhasil mengkoneksikan NodeMCU dengan Server lokal dengan socket namun data yang dikirimkan secara terus menerus tidak berhasil didapatkan oleh Server. Penulis mencoba mengubah data tersebut ke dalam bentuk <i>Byte</i> dan masih menjadi masalah yang harus dicari solusi.
85.	14 September 2018	Penulis mengkoneksikan NodeMCU dengan Server lokal melalui <i>Socket</i> dan berhasil mengirimkan data secara terus menerus (<i>realtime</i>). Penulis juga mengujinya dengan <i>ip host</i> yang berbeda ke Server tersebut.
86.	15 September 2018	Penulis melanjutkan riset dengan NodeMCU.
87.	16 September 2018	Penulis melanjutkan riset dengan NodeMCU.
88.	17 September 2018	Penulis membuat dokumentasi arsitektur <i>antifraud</i> Arduino (v1) dan NodeMCU (v2) ke dalam bentuk gambar.
89.	18 September 2018	Penulis membuat dokumentasi, yaitu skema dan <i>wiring diagram</i> sistem <i>antifraud</i> ke dalam bentuk gambar.
90.	19 September 2018	Penulis memperbaiki <i>flowchart</i> sistem <i>antifraud</i> .
91.	20 September 2018	Penulis menggabungkan dokumentasi, seperti arsitektur, <i>wiring diagram</i> , <i>flowchart</i> , dan program desktop/web dalam bentuk dokumen. Penulis merevisi laporan supaya laporan sesuai dengan ketentuan klien dan PT. Inspiro.
92.	21 September 2018	Penulis merampungkan revisi program dan laporan.

Tabel 2.1.1 – Penjelasan Logbook

2.2 Hasil Pekerjaan Secara Umum

2.2.1 Deskripsi Pekerjaan dan Kebutuhan Klien

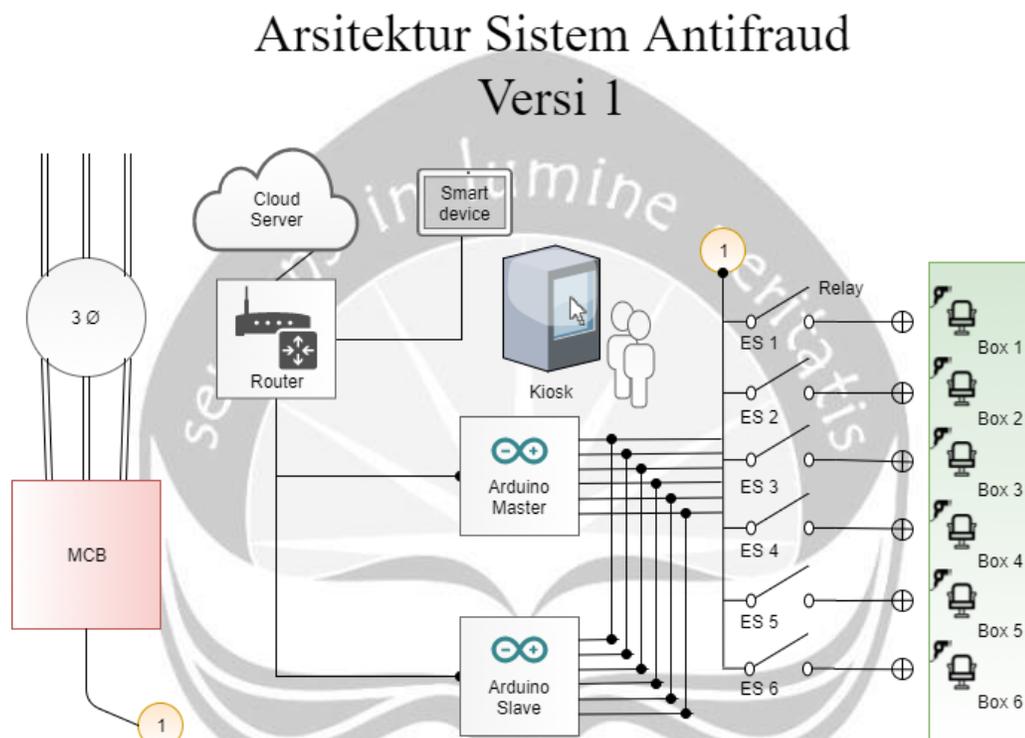
Penulis masuk ke Tim Developer IoT selama tiga bulan pelaksanaan magang di PT. Inspiro. Tim Developer IoT bertugas untuk membuat dan mengembangkan sistem *antifraud* dengan *Internet of Things* (IoT). Klien kami adalah perusahaan yang bergerak di bidang jasa *haircutting*. Teknologi yang sudah diterapkan klien adalah *kiosk machine* sehingga pembayaran lebih efisien dan mengurangi waktu antri. Saat itu, ada kebutuhan klien dalam membuat sistem *antifraud*, yaitu sistem untuk mencegah kecurangan yang dapat dilakukan oleh *Hair Stylist* (HS). Contohnya seperti menghindari HS melakukan *haircutting* tanpa izin, atau aktivitas lain yang tidak melalui proses pembayaran di mesin *kiosk*.

Bisnis klien tidak menggunakan kasir, karena sudah digantikan dengan mesin *kiosk* untuk proses pembayaran. Modul penerima uang (*bill acceptor*) berada di dalam mesin *kiosk* dan menerima uang pembayaran berupa mata uang Rupiah. Proses bisnis klien dimulai saat kustomer melakukan pembayaran dengan cara menekan tombol *cash* di mesin *kiosk*. Setelah itu, kustomer memasukkan uang ke slot penerima uang sampai memenuhi tarif yang ditagihkan. Mesin *kiosk* akan membuat struk dan kustomer bisa mengambil kertas nomor antrian itu dari mesin *kiosk*. Selanjutnya, kustomer menunggu nomor antriannya dipanggil oleh HS dan diarahkan menuju *box* kursi yang telah disediakan. Kustomer akan memperoleh potongan rambut yang sesuai, pembersihan rambut yang ramah lingkungan, dan sisir baru yang digunakan HS pada saat memotong rambut kustomer.

Klien mengharapkan aplikasi dan *hardware* sistem *antifraud* ini bisa berjalan secara terus-menerus dan selalu *online* supaya seluruh aktivitas bisnis bisa dikontrol dari pusat. Oleh karena sistemnya kontinyu dan dipasang di banyak cabang, maka sistem *antifraud* ini memakai prinsip *failover*. Prinsipnya apabila *device* utama (*Master*) rusak, maka secara otomatis sistem memfungsikan *device* cadangan (*Slave*).

HS tidak lagi dibebankan masalah teknis ketika terjadinya gangguan sistem, sehingga aktivitas bisnis tetap berjalan sembari teknisi pusat melakukan perbaikan.

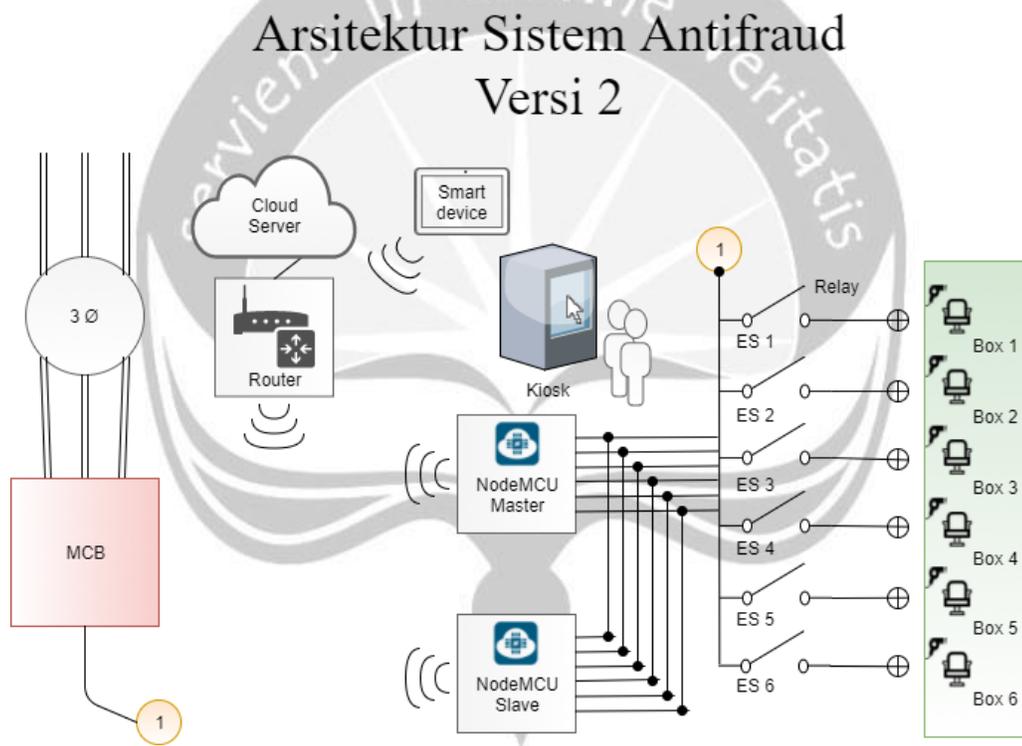
2.2.2 Arsitektur Sistem Antifraud



Gambar 2.2.1 – Arsitektur Sistem Antifraud Versi 1

Sistem *antifraud* yang dikerjakan oleh penulis memiliki dua arsitektur. Arsitektur pertama memakai komunikasi *Serial Ethernet*, sedangkan arsitektur kedua memakai *WiFi*. *Arduino Master/Slave* berfungsi untuk mengontrol relay di setiap *box* kursi. Durasi waktu untuk menyalanya *box* kursi sudah diprogram di *Arduino* dengan *threading* dari fungsi *milis*. Konsep ini memungkinkan *Arduino* mengontrol relay secara bersamaan atau paralel tanpa menunggu aksi sebelumnya terpenuhi.

Sistem *antifraud* bekerja dengan mengatur relay melalui sinyal bit (*Hexabyte*) yang dikirimkan dari aplikasi (*desktop / web*) ke perangkat Arduino atau NodeMCU. Sinyal itu akan mematikan atau mengalirkan listrik 220v AC ke *box* kursi. Aplikasi juga mengirimkan data seperti nomor HS, nomor kursi, dan durasi waktu menyalnya *box* kursi yang dipilih user. Status listrik di setiap *box* kursi akan dikirimkan secara *realtime* ke *cloud server* sehingga status kursi bisa dipantau dari pusat. Namun, penulis hanya ditugaskan membuat simulasinya di jaringan lokal saja untuk mengirim perintah dari aplikasi (*desktop / web*) ke perangkat Arduino /NodeMCU dan sebaliknya.



Gambar 2.2.2 – Arsitektur Sistem Antifraud Versi 2

Master/Slave pada arsitektur Gambar 2.2.2 dibentuk dari NodeMCU dan modul keypad 4x4 saja, karena *interface* program bisa dilihat melalui *smartdevice*. User bisa mengakses Arduino (*Web Server*) melalui protokol *http* (web) dari web browser

smartdevice (Android atau perangkat lain), asalkan *host* user masih terhubung ke dalam jaringan yang sama dengan sistem dan user mengetahui *ip address* Arduino.

Baik perangkat *Master* atau *Slave* mempunyai *ip static* yang sudah didefinisikan dari awal. Keduanya hidup secara terus menerus untuk mendukung konsep *failover*. Perangkat yang menerima perintah secara terus menerus adalah *Master*. Namun, ketika *Master* mengalami kerusakan atau gangguan, maka *Slave* akan menggantikan peran *Master*. Setelah *smartdevice* mengirimkan perintah ke *Master* tapi tidak menerima *handshake*, maka *smartdevice* akan mengirimkan perintah ke *Slave*. Perangkat *Master* lalu bisa diperbaiki tanpa mengganggu jalannya bisnis klien.

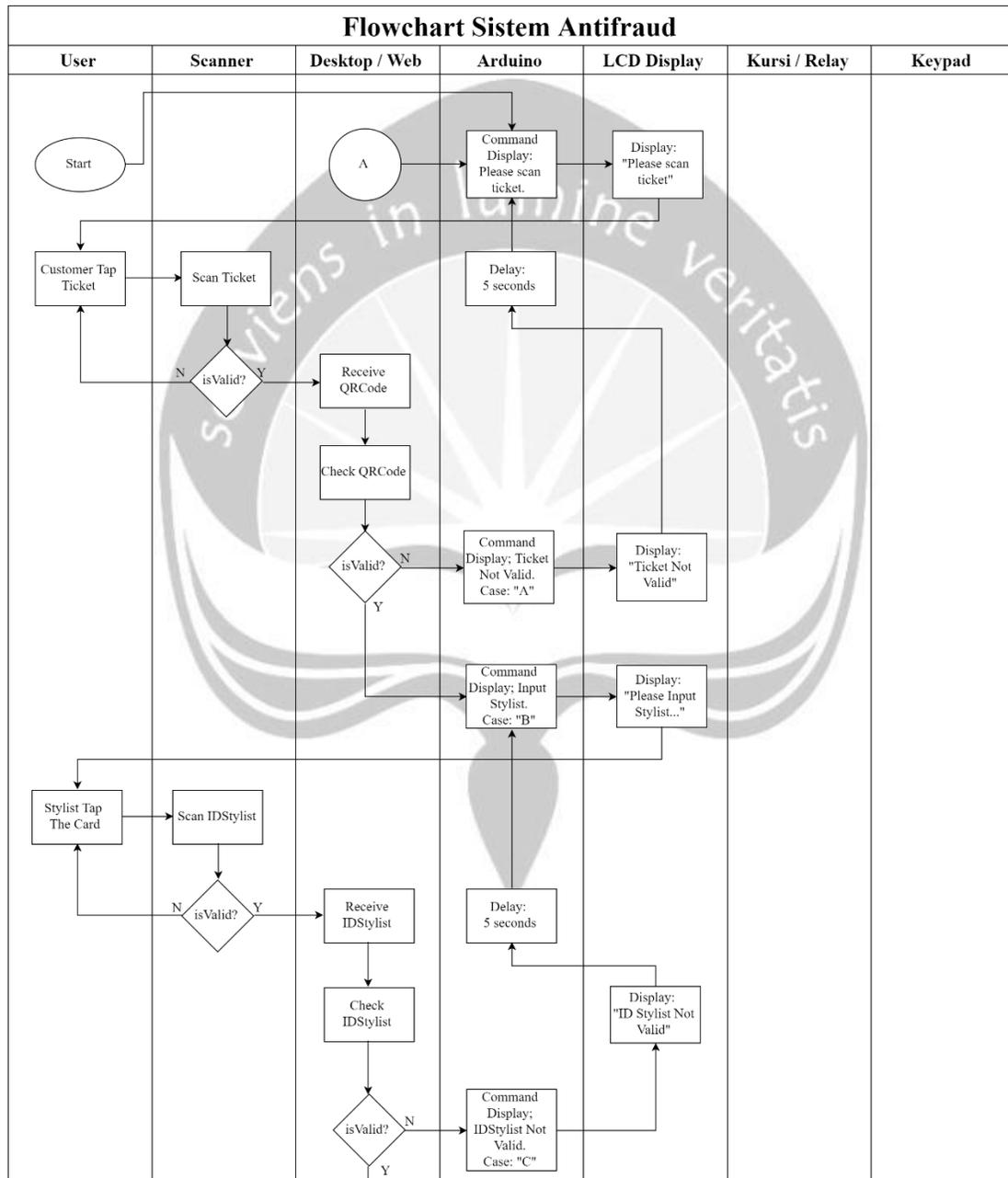
2.2.3 Metode Pengerjaan Proyek

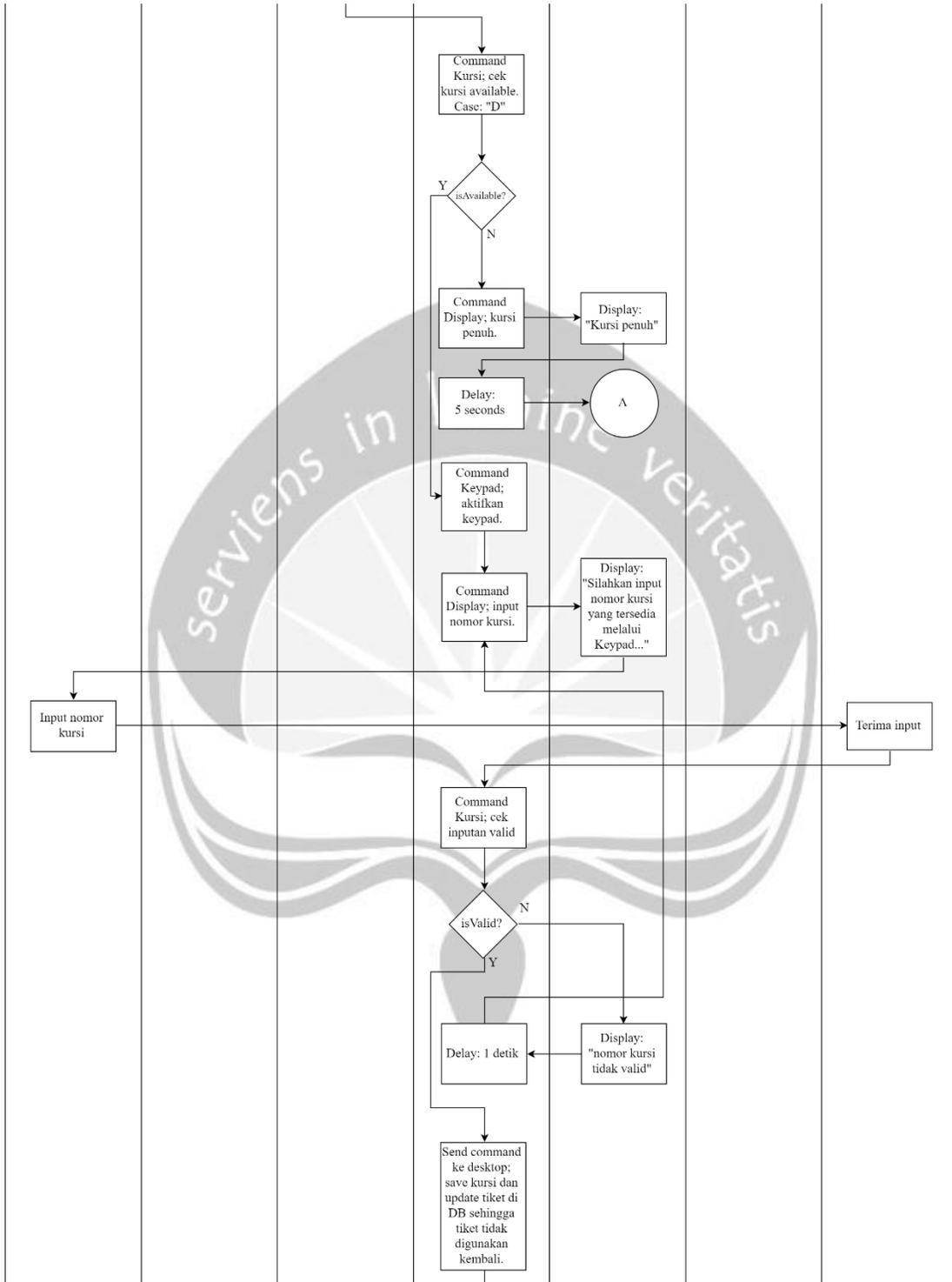
Selama pelaksanaan magang, penulis melakukan tiga aktivitas utama, yaitu riset dan pengembangan (*R&D*), perancangan, dan dokumentasi. Riset dan pengembangan adalah kegiatan mencari solusi dari suatu masalah secara iteratif, seperti *coding* dan *wiring hardware*. Aktivitas perancangan lalu dilakukan setelah menemukan solusi yang paling tepat. Dari situ terbentuk sebuah arsitektur, skema, atau rancangan kerja baik sebelum atau sesudah proyek karena kebutuhan bisnis klien yang dinamis dan berubah-ubah. Aktivitas dokumentasi dilakukan supaya penulis bisa menyimpan perubahan sistem itu ke dalam bentuk dokumen dan gambar.

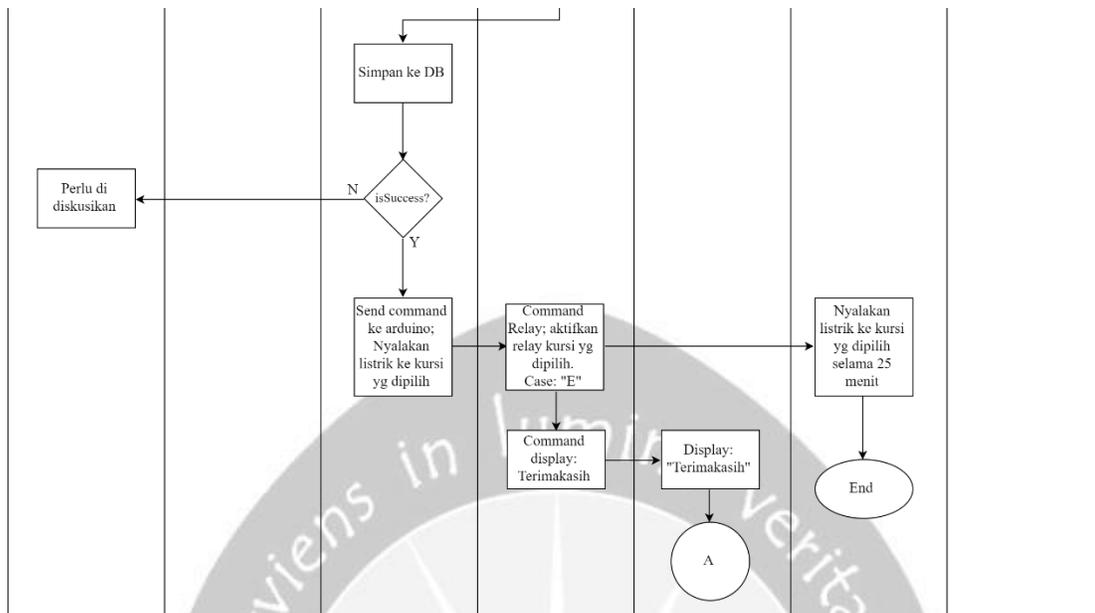
Aktivitas riset dan pengembangan dilakukan untuk menguji solusi yang ditemukan dari internet. Di tahap ini, penulis secara berulang mencoba kode program yang tepat sehingga Arduino / NodeMCU bisa melakukan perintah tiap case sesuai dengan *flowchart* pada Gambar 2.2.3. Penulis melakukan pembuatan dan pengembangan program untuk Arduino / NodeMCU dengan bahasa pemrograman C dari IDE Arduino. Program itu lalu dikoneksikan ke aplikasi desktop / web memakai Visual Studio 2017 dengan bahasa pemrograman C# dan *markup* HTML.

Pada aktivitas perancangan, penulis bertugas untuk merancang arsitektur, *flowchart*, dan *wiring diagram* yang dibutuhkan untuk pengembangan proyek sesuai

dengan keinginan klien dan perusahaan PT. Inspiro. Penulis banyak dibantu oleh Bapak Arvin untuk merancang arsitektur dan *case flowchart*, serta Bapak Andi Haloho membantu penulis dalam pembuatan skematik / *wiring diagram* yang menggambarkan koneksi antara pin Arduino dengan sensor / modul.







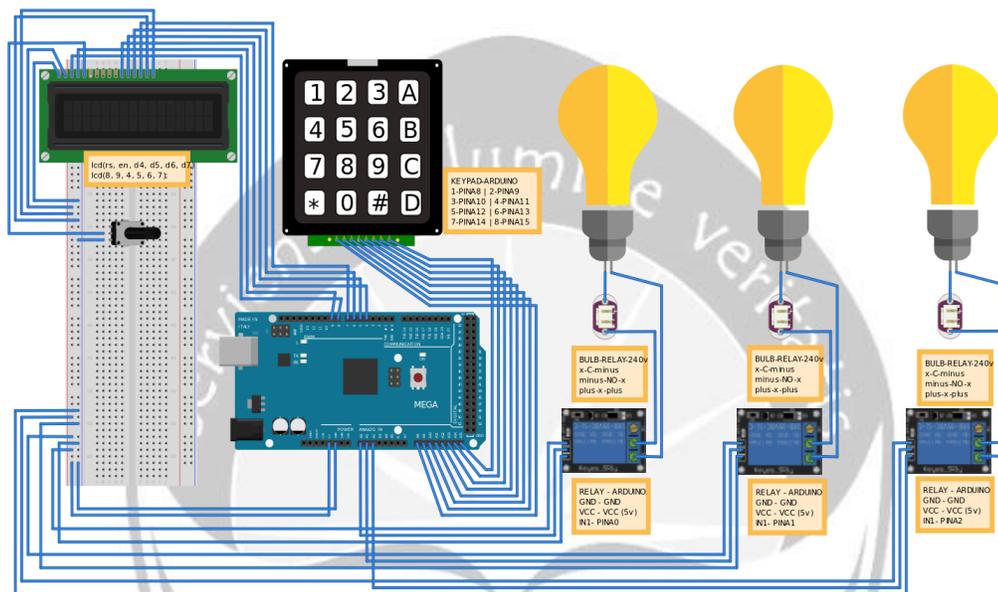
Gambar 2.2.3 – Flowchart Sistem Antifraud.

Gambar 2.2.3 menjelaskan proses bisnis sistem secara keseluruhan dan dipetakan ke dalam beberapa *case*. Berikut inti penjelasan di setiap *case* tersebut.

1. Case A memberikan *display* ke layar LCD yaitu “*Tiket tidak valid*”.
2. Case B memberikan *display* ke layar LCD yaitu “*Silahkan input ID Stylist*”.
3. Case C memberikan *display* ke layar LCD yaitu “*ID Stylist tidak valid*”.
4. Case D melakukan *command* yang sekuensial untuk mengecek status kursi, mengaktifkan input keypad, user menginputkan nomor kursi, sistem mengecek lagi nomor inputan user, dan jika lolos validasi maka *id stylist* dan nomor kursi akan dikirimkan ke program desktop untuk disimpan ke dalam database.
5. Case E melakukan *command* untuk mengalirkan listrik ke *box* kursi yang dipilih dengan cara memberikan perintah dari Arduino ke relay.
6. Case F ada di program namun tidak termasuk *case* Gambar 2.2.3. Fungsinya adalah memberi *command* yang bisa mengirimkan status kursi ke aplikasi.

7. Case G ada di program namun tidak termasuk *case* Gambar 2.2.3. Fungsinya adalah memberi *command* yang memutuskan arus listrik dengan memberi perintah ke relay yang dipilih (selain timer dengan fungsi *milis*).

2.2.4 Wiring Diagram Master atau Slave



Gambar 2.2.4 – Wiring Diagram Sistem Antifraud.

Diagram pada Gambar 2.2.4 adalah perangkat *Master* atau *Slave* yang dibentuk dari Arduino Mega 2560, LCD Display 16x2, Potensio, Breadboard, Keypad 4x4, kabel *jumper male to male*, kabel *jumper female to male*, dan usb 2.0 tipe A/B. Koneksi pin antara Arduino dan modul tertera di label kuning Gambar 2.2.4. Pin tersebut perlu dihubungkan dengan kabel *jumper* supaya modulnya bekerja dengan baik.

Perangkat *Master* atau *Slave* memiliki IP statis yang berbeda namun fungsinya sama, yaitu mengontrol aliran listrik. Penulis melakukan simulasinya dengan memakai tiga buah lampu 220v AC. Lampu tersebut menandakan status *box* kursi itu sedang menyala atau mati. Program lalu membaca keadaan relay (*state: high/low*) sehingga user bisa memantau keadaan listrik di setiap *box* kursi dari komputer.

BAB III

HASIL PEMBELAJARAN

3.1. Manfaat Magang

Manfaat yang penulis dapatkan dari pelaksanaan magang selama tiga bulan di PT. Inspiro adalah sebagai berikut.

- a) Penulis memperoleh pengetahuan dan gambaran dunia kerja di bidang teknologi.
- b) Penulis memperoleh kesempatan untuk ikut serta menangani *real project* perusahaan.
- c) Penulis memperoleh pengetahuan untuk membuat dan mempresentasikan produk ke klien.
- d) Penulis bisa mengasah kemampuan *hardskill* dan *softskill* yang hanya bisa diperoleh dalam dunia kerja.
- e) Penulis bisa mengaplikasikan kemampuan yang penulis dapatkan semasa perkuliahan.
- f) Penulis bisa mengetahui perbedaan antara teori yang diperoleh di universitas dengan praktik langsung di lapangan, khususnya dibidang teknologi.
- g) Penulis memperoleh pengetahuan baru tentang pengujian *software* dengan melihat proses bisnis klien.
- h) Penulis memperoleh pengetahuan baru tentang *hardware* untuk membangun sistem berbasis *Internet of Things*.
- i) Penulis bisa mengukur seberapa jauh kemampuan diri penulis di dunia kerja yang sesungguhnya.

3.2 Penerapan Ilmu dalam Magang

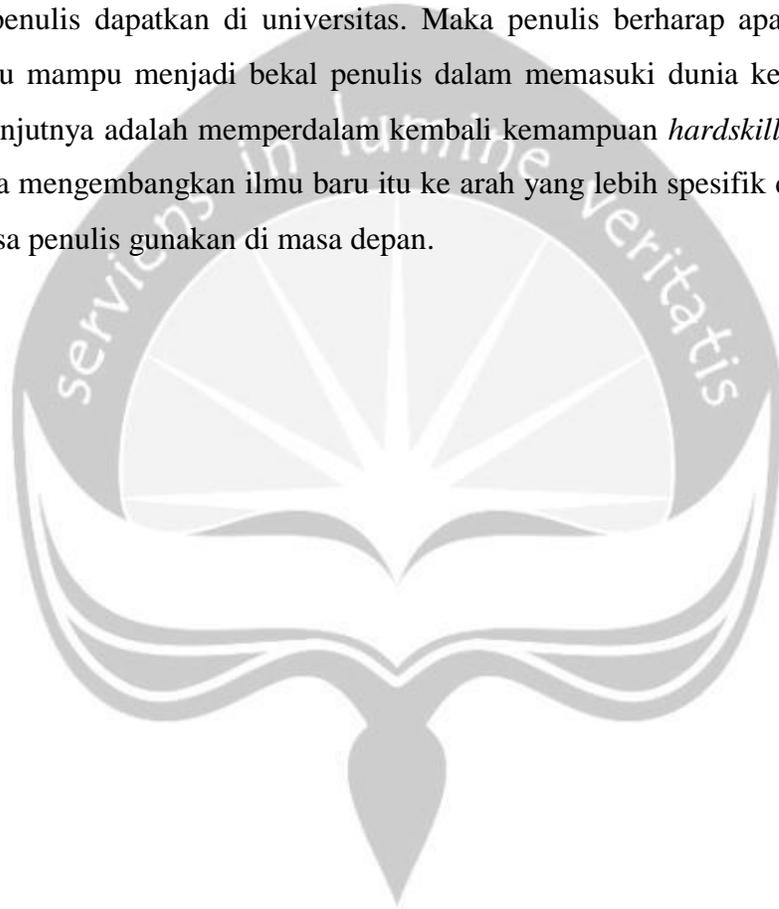
Persyaratan magang adalah sudah menempuh minimal 108 SKS serta pernah menempuh mata kuliah P3LP (Projek Pembangunan Perangkat Lunak Profesional). Persyaratan itu dibutuhkan oleh penulis supaya penulis memperoleh pengetahuan teknologi informasi yang dibutuhkan sewaktu pelaksanaan magang, baik itu teori maupun praktek, serta mengetahui proses pembuatan aplikasi yang berkualitas dan profesional. Penerapan ilmu penulis sewaktu pelaksanaan magang adalah cara berpikir untuk menemukan solusi dari setiap masalah, kemampuan dokumentasi, serta kemampuan analisis yang sudah penulis pelajari dari mata kuliah P3LP.

Selama magang di PT. Inspiro, penulis berada di Tim Developer IoT. Proyek yang dikerjakan oleh penulis adalah pembuatan sistem *antifraud* dengan *Internet of Things* (IoT). Penulis merasa terbantu dari materi mata kuliah Pengembangan Perangkat Lunak Berbasis IoT, Jaringan Komputer, Projek Pembangunan Perangkat Lunak Profesional, Basis Data, Pemrograman Visual, dan Pemrograman Web. Materi mata kuliah tersebut memuat teori-teori yang bisa dipakai penulis selama pengerjaan sistem *antifraud* di PT. Inspiro. Ilmu tersebut mendukung penulis dalam hal pemakaian kontroler (Arduino / NodeMCU) dan sensor, pembuatan *form application* dan *database*, pembuatan *webpage* sederhana, koneksi dan transfer data pada jaringan lokal (*Ethernet/WiFi*), serta berbagai hal lainnya sangat membantu penulis sewaktu pelaksanaan magang.

Mata kuliah Pengantar Sistem Informasi, Pengujian Perangkat Lunak, dan Rekayasa Perangkat Lunak juga membantu penulis untuk mempelajari kualitas sistem yang layak untuk bisnis perusahaan. Pengujian program *Enterprise Resource Planning* (ERP) seperti CS ERP memerlukan analisis yang kuat terhadap program dan bisnis klien, sehingga penulis bisa menghasilkan *scenario* dan *test case* dalam bentuk dokumen pengujian. Dokumen pengujian tersebut memiliki skenario dan *test case* di setiap bidang ERP seperti *Purchase Order*, *Material Request*, *Supplier Delivery Note*, *Purchase Invoice*, *Sales Order*, *Delivery Note*, *Invoice (Sales)*, *Accounting*, *Stock Inventory*, dan *Master Data Item*. Dokumen *user manual* penguji (*tester*) juga perlu

diperhatikan supaya penguji paham fungsional program secara keseluruhan dan memenuhi ekspektasi penguji terhadap program itu.

Pelaksanaan magang ini membantu penulis mengaplikasikan apa yang telah dipelajari sewaktu masa perkuliahan. Penulis mulai bisa menempatkan sejauh mana kualitas diri penulis, terutama untuk kemampuan *hardskill* dan *softskill* penulis. Kemampuan itu terus diasah selama masa pelaksanaan magang, dimana tidak semuanya penulis dapatkan di universitas. Maka penulis berharap apa yang sudah diperoleh itu mampu menjadi bekal penulis dalam memasuki dunia kerja. Langkah penulis selanjutnya adalah memperdalam kembali kemampuan *hardskill* dan *softskill* penulis serta mengembangkan ilmu baru itu ke arah yang lebih spesifik dan potensial sehingga bisa penulis gunakan di masa depan.



BAB IV

KESIMPULAN

4.1 Kesimpulan

Setelah menjalani magang selama tiga bulan di PT. Inspiro, penulis mendapatkan banyak pengalaman baru mengenai *real project*, lingkungan kerja, dan pengetahuan secara teori maupun praktik. Pelaksanaan magang juga mendorong penulis untuk mencari pengetahuan yang banyak dan paling baru di bidang teknologi. Ilmu yang diberikan saat perkuliahan hanya mengajarkan penulis tentang logika dan cara berpikir yang baik. Selibhnya, penulis melihat perlu adanya pengembangan dan penyesuaian diri sehingga bisa mengikuti era teknologi yang berkembang pesat dan dinamis sesuai dengan kebutuhan para pelaku industri teknologi.

4.2 Saran

Saran yang dapat diberikan dari pelaksanaan magang di PT.Inspiro adalah sebagai berikut.

1. Perlu adanya *collective knowledge management* yang baik supaya individu yang sudah berpengalaman bisa memberikan ilmunya kepada peserta magang dan karyawan baru.
2. Perlu mengatasi kecepatan *internet* terutama pada jam sibuk sehingga melancarkan kinerja peserta magang dan karyawan.
3. Perlu memberikan pekerjaan yang sesuai dengan bidang keahlian karyawan.

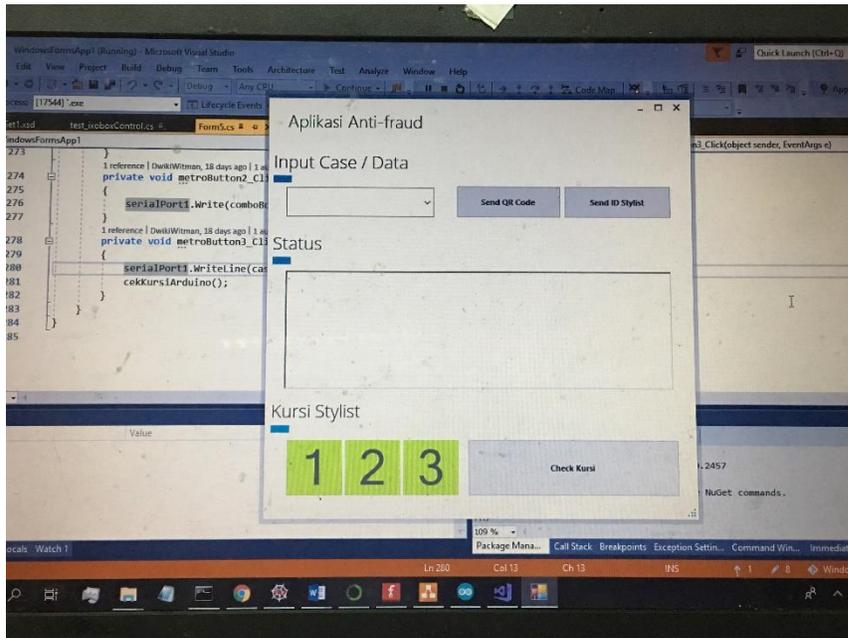
LAMPIRAN



Gambar Lampiran – Tampak Depan Kantor PT. Inspiro.



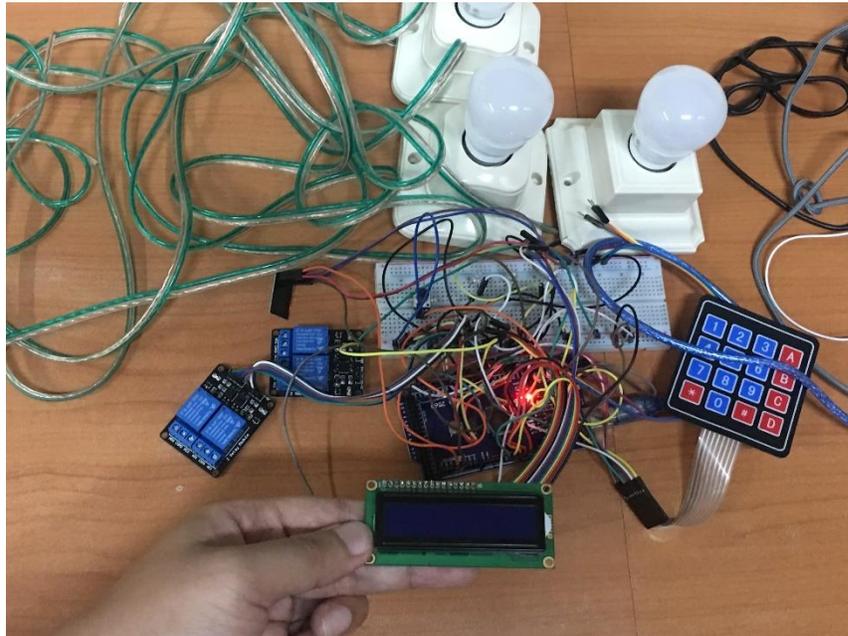
Gambar Lampiran – Ruang Developer PT. Inspiro.



Gambar Lampiran – Aplikasi Desktop Sistem Antifraud.



Gambar Lampiran – Aplikasi Desktop + Program Arduino.



Gambar Lampiran – Hardware Sistem Antifraud.



Gambar Lampiran – Hardware IoT Di Dalam Box.



Gambar Lampiran – Karyawan PT. Inspiro.

