



**RANCANG BANGUN SISTEM PELACAK BARANG UJI
LABORATORIUM BERBASIS RFID (*RADIO FREQUENCY
IDENTIFICATION*) MENGGUNAKAN DATABASE SQL
SERVER 2017 PADA BALAI RISET DAN STANDARDISASI
INDUSTRI SURABAYA**

KERJA PRAKTIK

Program Studi

S1 Teknik Komputer

**INSTITUT BISNIS
DAN INFORMATIKA**

stikom
SURABAYA

Oleh :

JEREMIAH PRASETYO

16410200012

FAKULTAS TEKNOLOGI DAN INFORMATIKA

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

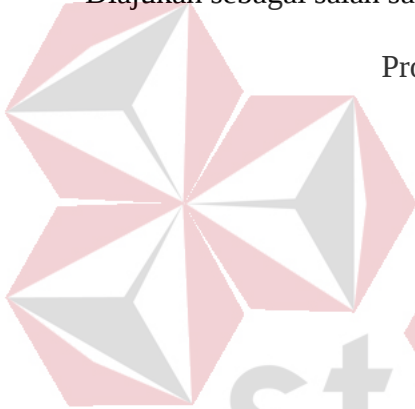
2019

LAPORAN KERJA PRAKTIK

RANCANG BANGUN SISTEM PELACAK BARANG UJI LABORATORIUM BERBASIS RFID (*RADIO FREQUENCY IDENTIFICATION*) MENGGUNAKAN DATABASE SQL SERVER 2017 PADA BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA

Diajukan sebagai salah satu syarat untuk menempuh ujian Tahap Akhir

Program Strata Satu (S1)



INSTITUT BISNIS
DAN INFORMATIKA

Disusun Oleh:

Nama : Jeremiah Prasetyo

NIM : 16.41020.0012

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

2019

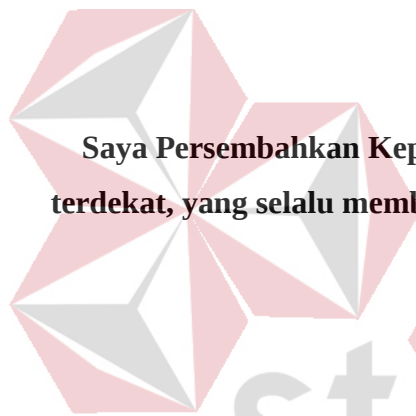


“I Can Do All Things Through Him Who Gives Me Strength”

PHILIPPIANS 4 : 13

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA



**Saya Persembahkan Kepada Papa dan Mama, Keluarga dan Orang
terdekat, yang selalu memberi semangat, motivasi, arahan dan masukan**

^_^

stikom
SURABAYA

LEMBAR PENGESAHAN

**RANCANG BANGUN SISTEM PELACAK BARANG UJI
LABORATORIUM BERBASIS RFID (*RADIO FREQUENCY
IDENTIFICATION*) MENGGUNAKAN DATABASE SQL SERVER 2017
PADA BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA**

Laporan Kerja Praktik oleh

Jeremiah Prasetyo

NIM: 16.41020.0012

Telah diperiksa, diuji, dan disetujui

Surabaya, Juli 2019

Disetujui:

Dosen Pembimbing,



Weny Indah Kusumawati, S.Kom., M.MT.

NIDN. 0721047201



Ratimah, S.E., M.M.

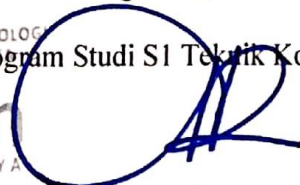
NIP. 196403151991032001

Mengetahui,



FAKULTAS TEKNOLOGI
DAK
Ketua Program Studi S1 Teknik Komputer

stikom
SURABAYA



Pauladie Susanto, S.Kom., M.T.

NIDN. 0729047501

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya:

Nama : Jeremiah Prasetyo
NIM : 16.41020.0012
Program Studi : SI Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Kerja Praktik
Judul Karya : **RANCANG BANGUN SISTEM PELACAK BARANG
UJI LABORATORIUM BERBASIS RFID (*RADIO
FREQUENCY IDENTIFICATION*) MENGGUNAKAN
DATABASE SQL SERVER 2017 PADA BALAI RISET
DAN STANDARDISASI INDUSTRI SURABAYA**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Juli 2019
Yang Menyatakan,


Jeremiah Prasetyo
NIM 16.41020.0012

ABSTRAK

Dengan berkembangnya zaman dan dunia teknologi, sistem pelacak barang merupakan metode yang digunakan untuk membantu perusahaan dalam manajemen administrasi dan proses bisnis, salah satu metode digunakan ialah dengan *Radio Frequency Identification* atau yang biasa disebut RFID, merupakan sebuah sistem yang menggunakan tag atau label yang ditanamkan pada barang untuk diidentifikasi.

Kurang lebih terdapat dua ratus sampel bahan uji masuk setiap harinya pada Baristand Industri Surabaya, sehingga berdampak pada manajemen administrasi dan proses bisnisnya, guna menunjang hal tersebut, maka dibangun sistem pelacak barang uji laboratorium berbasis RFID menggunakan database SQL Server 2017 yang terintegrasi server lokal XAMPP dan mikrokontroler yang bekerja secara kabel, yaitu Arduino serta nirkabel menggunakan NodeMCU untuk transmisi data ke database dengan protokol HTTP dan jembatan Program PHP. Sistem ini bekerja dengan menggunakan tiga jenis alat, yaitu Scanner loket yang digunakan untuk memasukan data identitas barang uji beserta pelanggan pada database dan tag RFID, Scanner Laboratorium yang berfungsi merekam jejak posisi barang uji, serta Scanner Reset untuk menghapus data pada database dan tag RFID agar dapat digunakan kembali.

Dengan dibuatnya sistem ini, maka terciptalah sistem yang terintegrasi dengan database untuk menunjang manajemen administrasi dan proses bisnis yang telah terkomputerisasi dan dapat meminimalisir kesalahan manusia.

Kata Kunci: Arduino, HTTP, NodeMCU, PHP, RFID, SQL Server, XAMPP

KATA PENGANTAR

Puji Syukur penulis panjatkan kepada Tuhan Yesus Kristus atas segala hikmat dan anugerah yang telah diberikan, maka Laporan Kerja Praktik ini dapat selesai dengan baik, sehingga terpenuhinya salah satu syarat dalam menempuh Tugas Akhir pada Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya.

Pada kesempatan ini, penulis mengucapkan terima kasih kepada pihak-pihak yang telah memberikan bimbingan, serta kritik dan saran mendukung. Ucapan terima kasih ini penulis tujukan kepada:

1. Kedua orang tua penulis yang telah memberikan bimbingan dan dukungan moral dan materi, sehingga penulis dapat menyelesaikan Kerja Praktik beserta laporan ini dengan baik. Begitu pula bagi seluruh keluarga besar penulis yang telah memberikan kritik dan saran membangun.
2. Kelompok penulis semasa Kerja Praktik yang telah memberikan ide-ide serta motivasi dalam menyelesaikan tugas yang diberikan. Kami menyebut kelompok ini ESKA, terdiri atas Dewa, Fahmi, dan Rofiq.
3. Balai Riset dan Standardisasi Industri Surabaya, khususnya Penyelia kami Ibu Fatimah, S.E., M.M., yang telah memberikan kepercayaan dalam menyelesaikan kebutuhan dan permasalahan pada bagian Pengembangan dan Jasa Teknik.
4. Dosen Pembimbing penulis, Ibu Weny Indah Kusumawati, S.Kom., M.MT., yang telah memberikan masukan, saran dan kritik membangun, sehingga Kerja Praktik dan Laporan ini dapat selesai dengan baik.

5. Ketua Program Studi S1 Teknik Komputer, Bapak Pauladie Susanto, S.Kom., M.T., yang telah memberikan izin kepada penulis untuk melaksanakan Kerja Praktik.
6. Orang Terdekat penulis, UKPD, dan Keluarga Besar S1 Teknik Komputer yang telah memberikan semangat dan dorongan.

Surabaya, Juli 2019

Penulis



DAFTAR ISI

	Halaman
ABSTRAK	v
KATA PENGANTAR	vii
DAFTAR ISI	x
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Kontribusi	4
BAB II GAMBARAN UMUM PERUSAHAAN	5
2.1 Sejarah dan Perkembangan	5
2.2 Visi, Misi, dan Strategi	7
2.2.1 Visi	7
2.2.2 Misi	8

2.2.3	Strategi.....	8
2.3	Tugas Pokok Fungsi dan Susunan Organisasi	8
2.3.1	Tugas Pokok	9
2.3.2	Fungsi	9
2.3.3	Susunan Organisasi	10
2.4	Sumber Daya.....	11
2.5	Moto dan Logo.....	13
2.6	Lokasi Perusahaan.....	14
BAB III LANDASAN TEORI.....		15
3.1	RFID (<i>Radio Frequency Identification</i>).....	15
3.2	Ultralight Mifare NFC Tag 203	17
3.3	Modul RFID MFRC522.....	17
3.4	Arduino Nano.....	18
3.5	NodeMCU V3	20
3.6	I2C LCD Karakter 16 x 2.....	22
3.7	Microsoft SQL Server 2017	22
3.7.1	Microsoft SQL Server Management Studio 2017	23
3.7.2	<i>Statement SQL</i>	23
3.8	Microsoft Visual Studio 2017	24
3.9	XAMPP	25
3.9.1	Bagian dari XAMPP	26

3.9.2	Komponen XAMPP	27
3.10	PHP	28
3.10.1	Variabel PHP	29
3.10.2	Operator Aritmatika PHP	29
3.10.3	Operator Logika PHP	30
3.10.4	Operator Pembandingan PHP	30
BAB IV DESKRIPSI PEKERJAAN		32
4.1	Analisis Sistem.....	32
4.2	Diagram Alur Proses Sistem Pelacak Barang Uji Laboratorium ..	33
4.3	Spesifikasi Alat	34
4.3.1	Scanner Locket.....	34
4.3.2	Scanner Laboratorium	34
4.3.3	Scanner Reset	35
4.4	Skematik Alat.....	36
4.4.1	Skematik Scanner Locket.....	36
4.4.2	Skematik Scanner Laboratorium	37
4.4.3	Skematik Scanner Reset	37
4.5	Desain Alat.....	38
4.5.1	Desain 3D Scanner Locket.....	38
4.5.2	Desain 3D Scanner Laboratorium	38
4.5.3	Desain 3D Scanner Reset	39

4.6	Pembuatan Server.....	39
4.7	Pembuatan Jaringan Lokal	43
4.8	Pembuatan Sistem Basis Data	46
4.9	Pembuatan Program	50
4.9.1	Pembuatan Program Desktop Locket.....	51
4.9.2	Pembuatan Program Mikrokontroler Scanner Locket.....	61
4.9.3	Program Mikrokontroler Scanner Laboratorium.....	66
4.9.4	Pembuatan Program Mikrokontroler Scanner Reset	72
4.9.5	Program PHP Jembatan Mikrokontroler dan Database.....	73
4.10	Uji Coba Alat dan Program.....	78
4.10.1	Uji Coba Scanner Locket dan Program Desktop Locket	78
4.10.2	Uji Coba Scanner Laboratorium.....	82
4.10.3	Uji Coba Scanner Reset.....	85
4.11	Hasil Alat	86
BAB V PENUTUP.....		88
5.1	Kesimpulan	88
5.2	Saran.....	89
DAFTAR PUSTAKA		90

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Sumber Daya Baristand Industri Surabaya	11
Gambar 2.2 Moto Baristand Industri Surabaya.....	13
Gambar 2.3 Logo Baristand Industri Surabaya.....	13
Gambar 2.4 Lokasi Baristand Industri dari Stikom Surabaya.....	14
Gambar 3.1 Jenis-Jenis RFID Tag	15
Gambar 3.2 Komunikasi NFC Tag dengan Perangkat Pembaca dan Penulis	17
Gambar 3.3 Pin Modul RFID RC-522	18
Gambar 3.4 Papan Arduino Nano	18
Gambar 3.5 Pin Arduino Nano	20
Gambar 3.6 Pin NodeMCU V3.....	21
Gambar 3.7 Pin I2C LCD Karakter 16 x 2.....	22
Gambar 3.8 Pin LCD Karakter 16 x 2.....	22
Gambar 3.9 Antarmuka Microsoft SQL Server Management Studio 2017.....	23
Gambar 3.10 Antarmuka Microsoft Visual Studio 2017	24
Gambar 3.11 Antarmuka XAMPP Control Panel v3.2.2	26
Gambar 4.1 Diagram Alur Proses Sistem Pelacak Barang Uji Laboratorium	33
Gambar 4.2 Skematik Alat Scanner Locket.....	36
Gambar 4.3 Skematik Scanner Laboratorium	37
Gambar 4.4 Skematik Scanner Reset	37
Gambar 4.5 Desain 3D Scanner Locket.....	38
Gambar 4.6 Desain 3D Scanner Laboratorium	38

Gambar 4.7 Desain 3D Scanner Reset	39
Gambar 4.8 Tampilan Awal Instalasi XAMPP	40
Gambar 4.9 Komponen XAMPP Yang Akan Diinstalasi.....	40
Gambar 4.10 Pemilihan Folder Untuk Instalasi XAMPP	41
Gambar 4.11 Tampilan XAMPP Control Panel.....	41
Gambar 4.12 Tampilan Awal Localhost (Server Lokal).....	42
Gambar 4.13 Tampilan Direktori Server XAMPP	42
Gambar 4.14 Tampilan Menu Windows Settings	43
Gambar 4.15 Tampilan Menu Network & Internet.....	44
Gambar 4.16 Tampilan Menu Mobile Hotspot	44
Gambar 4.17 Tampilan Command Prompt Perintah ipconfig	45
Gambar 4.18 Tampilan Server Yang Telah Terintegrasi Jaringan Lokal	46
Gambar 4.19 Tampilan Microsoft SQL Server Management Studio 2017	47
Gambar 4.20 Terbuka Microsoft SQL Server Management Studio 2017	47
Gambar 4.21 Tampilan New Database	48
Gambar 4.22 Tampilan Menu New Database	48
Gambar 4.23 Mengklik Kanan New Table	49
Gambar 4.24 Tabel Master_Loket	49
Gambar 4.25 Isi Tabel Antrian, Proses, Selesai dan Reset	50
Gambar 4.26 Isi Tabel tb_user	50
Gambar 4.27 Isi Database RFID_Scanner	50
Gambar 4.28 Mengklik File > New Project Microsoft Visual Studio 2017	51
Gambar 4.29 Tampilan Menu New Project	52
Gambar 4.30 Tampilan Menu Form Kosong Microsoft Visual Studio 2017	52

Gambar 4.31 Tampilan Menu Login Aplikasi	53
Gambar 4.32 Mendesain Form Utama	55
Gambar 4.33 Tampilan Form Login	79
Gambar 4.34 Tampilan Form Utama	80
Gambar 4.35 Alat Scanner Locket.....	80
Gambar 4.36 Tampilan Program Desktop Locket	81
Gambar 4.37 Tampilan Program Desktop Data Berhasil Dimasukkan	81
Gambar 4.38 Database Tabel Master_Locket.....	82
Gambar 4.39 Gambar Rangkaian Scanner Laboratorium	82
Gambar 4.40 Hasil Kode Baca.....	83
Gambar 4.41 Database Tabel Antrian_LFisika.....	83
Gambar 4.42 Kode Write pada Tag RFID	84
Gambar 4.43 Hasil Database Tabel Proses_LFisika	84
Gambar 4.44 Hasil Database Selesai_LFisika	84
Gambar 4.45 Mengecek Kode Barang T0003 Tidak Ada Pada Master_Locket	85
Gambar 4.46 Mengecek Tag RFID dengan Scanner Reset	85
Gambar 4.47 Mengecek RFID Setelah Diatur Ulang	86
Gambar 4.48 Alat Scanner Locket Dengan Stiker Tag RFID	86
Gambar 4.49 Alat Scanner Laboratorium	87
Gambar 4.50 Alat Scanner Reset	87

DAFTAR TABEL

	Halaman
Tabel 2.1 SDM Baristand Surabaya Berdasarkan Tingkat Pendidikan	11
Tabel 2.2 SDM Baristand Industri Surabaya Berdasarkan Golongan	12
Tabel 2.3 SDM Baristand Industri Surabaya Berdasarkan Fungsional	12
Tabel 3.1 Spesifikasi Arduino Nano	19
Tabel 3.2 Spesifikasi NodeMCU V3	21
Tabel 3.3 Operator Aritmatika PHP	30
Tabel 3.4 Operator Logika PHP	30
Tabel 3.5 Operator Pembanding PHP	31
Tabel 4.1 Spesifikasi Scanner Locket	34
Tabel 4.2 Spesifikasi Scanner Laboratorium	35
Tabel 4.3 Spesifikasi Scanner Reset	35
Tabel 4.4 Spesifikasi Database RFID_Scanner	46

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Balasan Perusahaan	91
Lampiran 2 Form KP 5 Halaman 1	92
Lampiran 3 Form KP 5 Halaman 2	93
Lampiran 4 Form KP 6 Halaman 1	94
Lampiran 5 Form KP 6 Halaman 2	95
Lampiran 6 Form KP 7 Halaman 1	96
Lampiran 7 Form KP 7 Halaman 2	97
Lampiran 8 Kartu Bimbingan	98
BIODATA PENULIS	99



BAB I

PENDAHULUAN

Pada bab satu, menjelaskan mengenai Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, yang ada di Baristand Industri Surabaya, selanjutnya dijabarkan tujuan dan manfaat serta kontribusi.

1.1 Latar Belakang Masalah

Balai Riset dan Standardisasi Industri Surabaya (Baristand Industri Surabaya) berperan penting dalam menentukan kualitas dan mutu dari sebuah produk yang dibuat dalam skala industri kecil, menengah maupun besar, khususnya dalam pengujian bahan dan barang dalam ruang lingkup tertentu yang diuji pada setiap laboratoriumnya masing-masing, yaitu: Elektronika, Fisika, Kimia, Lingkungan, dan Telematika.

Untuk mengetahui mutu dan kualitas, serta mendapatkan sertifikasi SNI (Standar Nasional Indonesia), maka dalam satu hari, kurang lebih terdapat dua ratus sampel bahan uji sesuai ruang lingkupnya masing-masing yang diajukan kepada Baristand Industri Surabaya oleh pelanggan. Hal ini berdampak pula pada manajemen administrasi dan proses bisnis Baristand Industri Surabaya, sehingga diperlukannya sebuah sistem pelacak terhadap sampel tersebut.

Sejauh ini dalam manajemen administrasi dan proses bisnis di Baristand Industri Surabaya, khususnya dalam pengujian bahan dan barang, telah menerapkan sistem basis data (Database) pada komputer, sehingga manajemen administrasi dan proses bisnis pada sampel uji dapat dengan mudah dikelola dan dipantau. Namun

pada sistem tidak ada rekam jejak untuk mengetahui kondisi terkini sampel uji dalam proses pengujian tertentu, maka dari itu tahapan pengujian tidak dapat terpantau dengan baik dan rentan terhadap kemungkinan kelalaian manusia. Adapun dalam pemberian identitas dan keterangan pada sampel uji masih manual dan tidak memiliki kode unik tertentu yang membedakan antara setiap sampel uji.

Atas dasar tersebut, dirancanglah sebuah sistem pelacak barang uji laboratorium menggunakan RFID (*Radio Frequency Identification*) untuk memberikan identitas berupa kode unik pada setiap sampel uji dan sebuah mikrokontroler untuk mentransmisikan data dan kode RFID ke Database.

1.2 Perumusan Masalah

Bagaimana cara merancang dan membangun Sistem Pelacak Barang Uji Laboratorium dengan menggunakan RFID (*Radio Frequency Identification*) dan Mikrokontroler dengan Database Microsoft SQL Server 2017 pada Balai Riset dan Standardisasi Industri Surabaya.

1.3 Batasan Masalah

Berdasarkan permasalahan yang ada, adapun batasan masalah sesuai dengan kebutuhan di Baristand Industri Surabaya, yaitu:

1. Chip RFID, menggunakan Mifare Ultralight NFC tag, berupa stiker.
2. Pembaca dan Penulis RFID menggunakan modul MFRC522.
3. Mikrokontroler pertama menggunakan Arduino Nano untuk transmisi data melalui kabel.

4. Mikrokontroler kedua menggunakan NodeMCU V3 untuk transmisi data melalui jaringan nirkabel.
5. Database menggunakan Microsoft SQL Server 2017.
6. Server lokal menggunakan program XAMPP.
7. Transmisi data ke sistem basis data melalui kabel menggunakan Serial Port.
8. Transmisi data ke sistem basis data melalui jaringan nirkabel menggunakan protokol TCP/IP dan Program PHP yang ditanamkan pada Server lokal.
9. Pembuatan program Desktop menggunakan Visual Studio dengan bahasa pemrograman C#.
10. Pembuatan program Mikrokontroler menggunakan Arduino IDE.

1.4 Tujuan

Adapun berdasarkan rumusan masalah, tujuannya adalah membuat Sistem Pelacak Barang Uji Laboratorium dengan menggunakan RFID (*Radio Frequency Identification*) dan Mikrokontroler dengan Database Microsoft SQL Server 2017, untuk menunjang manajemen administrasi dan proses bisnis pada barang uji, agar dapat dengan mudah dikelola, dipantau dan dilacak kondisi terkini dari barang uji, sehingga memberikan informasi barang uji yang akurat dan tepat bagi Pegawai Baristand Industri Surabaya.

1.5 Manfaat

1. Bagi Mahasiswa
 - a. Menambah relasi dengan pegawai di Baristand Industri Surabaya.

- b. Membekali diri dengan wawasan dan pengetahuan dalam dunia kerja, baik itu *hardskill* dan *softskill*.
- c. Dapat menerapkan ilmu pengetahuan yang didapatkan selama perkuliahan.

2. Bagi Baristand Industri Surabaya

- a. Dapat mempererat hubungan antara Baristand Industri Surabaya dengan Institut Bisnis dan Informatika Stikom Surabaya.
- b. Beberapa permasalahan di Baristand Industri Surabaya dapat terselesaikan, karena mendapatkan bantuan tenaga kerja dari mahasiswa.
- c. Dengan adanya mahasiswa Kerja Praktik, maka beban kerja pegawai dapat berkurang.

1.6 Kontribusi

Memberikan kontribusi ke Balai Riset dan Standardisasi Industri Surabaya dengan merancang dan membangun Sistem Pelacak Barang Uji Laboratorium Berbasis RFID (*Radio Frequency Identification*) menggunakan Database SQL Server 2017, sehingga dapat menunjang manajemen administrasi dan proses bisnis pada barang uji, agar dapat dengan mudah dikelola, dipantau, dan dilacak kondisi terkini dari barang uji.

BAB II

GAMBARAN UMUM PERUSAHAAN

Pada bab dua, menjelaskan mengenai gambaran umum dari Balai Riset dan Standardisasi Industri Surabaya, yang terdiri atas sejarah dan perkembangan, visi dan misi, serta strategi, tugas pokok fungsi dan susunan organisasi, sumber daya, moto dan logo, serta lokasi perusahaan.

2.1 Sejarah dan Perkembangan

Sejak awal berdirinya Balai Riset dan Standardisasi Industri Surabaya (Baristand Industri Surabaya) telah mengalami beberapa kali perubahan nama dan perpindahan lokasi dari satu kota ke kota lain.

Perusahaan ini didirikan pada 4 Maret 1947 di Klaten Jawa Tengah dengan nama Balai Penyelidikan Kimia, berada di bawah Kementerian Kemakmuran. Dari Klaten pindah ke Solo pada 25 April 1950 dan pindah untuk ke dua kalinya ke Yogyakarta pada 25 April 1951. Dari Yogyakarta pindah ke Jalan Garuda No. 2 Surabaya dan pada Mei 1961, pindah untuk ke empat kalinya ke Jl. Perak Timur 358 Surabaya. Untuk terakhir kalinya bersamaan dengan peringatan hari Pahlawan 10 November 1975, menempati gedung milik sendiri seluas 4.200 m² di atas tanah 10.200 m² yang berlokasi di Jl. Jagir Wonokromo 360 Surabaya.

Selain perpindahan lokasi, juga mengalami perubahan nama dari semula Balai Penyelidikan Kimia, berubah menjadi Balai Penelitian Kimia dibawah PNPR Nupika Yasa (1966-1980). Sesuai dengan tuntutan perkembangan industrialisasi maka berdasar Keputusan Menteri Perindustrian No. 357/MK/SK/8/1980, tanggal

26 Agustus 1980, nama, Struktur Organisasi, Tugas Pokok dan Fungsinya ditingkatkan menjadi Balai Penelitian dan Pengembangan Industri Surabaya (BISb), yang berada dibawah Badan Penelitian dan Pengembangan Industri Departemen Perindustrian.

Untuk menunjang peningkatan daya saing industri dalam perdagangan bebas, Struktur Organisasi, Tugas Pokok dan Fungsi BISb ditingkatkan dan namanya diubah menjadi Balai Riset dan Standardisasi Industri dan Perdagangan Surabaya (Baristand Indag Surabaya) berdasar Surat Keputusan Menteri Perindustrian dan Perdagangan No. 784/MPP/SK/11/2002 tanggal 29 November 2002.

Sehubungan dengan pemisahan Departemen Perindustrian dan Departemen Perdagangan serta dalam rangka menyesuaikan misi organisasi Balai Riset dan Standardisasi Industri dan Perdagangan sesuai dengan kebutuhan nyata masyarakat industri maka berdasar Surat Keputusan Menteri Perindustrian No. 49/M-IND/PER/6/2006 maka struktur organisasi Balai Riset dan Standardisasi Industri dan Perdagangan Surabaya diubah menjadi Balai Riset dan Standardisasi Industri Surabaya (Baristand Industri Surabaya).

Sejak awal berdirinya sampai dengan tahun 2005, kegiatan jasa pelayanan teknis lebih terkonsentrasi pada bidang kimia dan logam, namun sejak tahun 2005 fokus kegiatan diarahkan ke bidang peralatan listrik dan elektronika (termasuk audio video), namun sejak tahun 2007 untuk mendukung pengembangan industri nasional yang berbasis produk elektronika telematika, maka kegiatan riset dan jasa layanan teknis pada Baristand Industri Surabaya lebih difokuskan pada bidang elektronika telematika.

Baristand Industri Surabaya sebagai unit pelaksana teknis yang menangani litbang industri elektronika telematika, berperan dalam melaksanakan kebijakan pengembangan industri nasional untuk menopang pengembangan industri elektronika telematika di Indonesia. Dengan melaksanakan tugas tersebut maka diharapkan akan berkembang industri elektronika telematika yang kuat dan mandiri sehingga dapat memperluas lapangan kerja dan mendorong percepatan pembangunan industri nasional.

Di samping tugas pembangunan yaitu mendorong tumbuhnya industri elektronika telematika nasional, Baristand Industri Surabaya secara internal mempunyai tugas untuk meningkatkan kemampuan diri melalui peningkatan kompetensi serta memberikan jasa layanan teknis kepada industri kecil, menengah dan besar yang juga merupakan suatu kegiatan bisnis. Pada dasarnya upaya peningkatan kompetensi Balai merupakan sumber yang dapat meningkatkan peran Baristand Industri Surabaya dalam menunjang program pembangunan industri elektronika telematika maupun meningkatkan jasa pelayanan teknis yang diberikan kepada industri dan masyarakat.

2.2 Visi, Misi, dan Strategi

2.2.1 Visi

Visi Balai Riset dan Standardisasi Industri Surabaya merupakan potret masa depan yang dicita-citakan, yaitu:

1. Sebagai Lembaga Riset Dan Standardisasi Terkemuka Yang Menjadi Mitra Industri Elektronika Dan Telematika Nasional.

2. Dalam Berperan Sebagai Basis Produksi Yang Melayani Kebutuhan Nasional Maupun Dunia Pada Tahun 2025.

2.2.2 Misi

Baristand Industri Surabaya dalam usaha keras mencapai Visi yang telah ditetapkan diatas, mengemban Misi sebagai berikut:

1. Menghasilkan riset dan rancang bangun perekayasaan industri elektronika dan telematika.
2. Menghasilkan pelayanan kesesuaian (pengujian, kalibrasi dan sertifikasi) produk industri elektronika dan telematika.
3. Mengembangkan kompetensi sumber daya manusia pada industri elektronika dan telematika.

2.2.3 Strategi

Berdasar tugas dan fungsinya, Baristand Industri Surabaya menetapkan Strategi Bisnis sebagai berikut:

“Memperbesar pangsa pasar secara agresif yang didukung dengan pertumbuhan segmen pasar, spesialisasi produk dan investasi yang selektif serta memperkuat kemampuan internal”.

2.3 Tugas Pokok Fungsi dan Susunan Organisasi

Menurut "Peraturan Menteri Perindustrian tentang Organisasi dan Tata Kerja Balai Besar, Balai Riset dan Standardisasi Industri Baristand Industri Surabaya" Nomor 49/M-IND/PER/6/2006 adalah unit pelaksana teknis di lingkungan

Departemen Perindustrian yang berada dibawah dan bertanggung jawab kepada Kepala Badan Penelitian dan Pengembangan Industri.

2.3.1 Tugas Pokok

Baristand Industri Surabaya mempunyai tugas melaksanakan riset dan standardisasi serta sertifikasi di bidang industri.

2.3.2 Fungsi

Dalam melaksanakan tugas, Baristand Industri menyelenggarakan fungsi:

1. Pelaksanaan penelitian dan pengembangan teknologi industri di bidang bahan baku, bahan penolong, proses, peralatan / mesin dan hasil produk serta penanggulangan pencemaran industri.
2. Penyusunan program dan pengembangan kompetensi di bidang jasa riset / litbang.
3. Perumusan dan penerapan standar, pengujian dan sertifikasi dalam bidang bahan baku, bahan penolong, proses, peralatan / mesin, dan hasil produk.
4. Pemasaran, kerjasama, promosi, pelayanan informasi, penyebarluasan dan pendayagunaan hasil riset / penelitian dan pengembangan.
5. Pelaksanaan urusan kepegawaian, keuangan, tata persuratan, perlengkapan, kearsipan, rumah tangga, koodinasi penyusunan bahan rencana dan program, penyiapan bahan evaluasi dan pelaporan industri.

2.3.3 Susunan Organisasi

Baristand Industri Surabaya terdiri dari:

1. Subbagian Tata Usaha: Mempunyai tugas melakukan urusan kepegawaian, keuangan, inventarisasi barang milik negara, tata persuratan, perlengkapan, kearsipan, rumah tangga, koordinasi penyusunan bahan rencana dan program, penyiapan bahan evaluasi dan pelaporan Baristand Industri, serta pengelolaan perpustakaan.
2. Seksi Teknologi Industri: Mempunyai tugas melakukan penyiapan bahan penelitian dan pengembangan teknologi industri bahan baku, bahan penolong, proses, peralatan / mesin, dan hasil produk, serta penanggulangan pencemaran industri.
3. Seksi Program dan Pengembangan Kompetensi: mempunyai tugas melakukan penyiapan bahan penyusunan program dan pengembangan kompetensi di bidang jasa riset / litbang.
4. Seksi Standardisasi dan Sertifikasi: mempunyai tugas melakukan penyiapan bahan perumusan dan penerapan standar, pengujian dan sertifikasi dalam bidang bahan baku, bahan penolong, proses, peralatan / mesin, dan hasil produk.
5. Seksi Pengembangan Jasa Teknis: mempunyai tugas melakukan penyiapan bahan pemasaran, kerjasam, promosi, pelayanan informasi, penyebarluasan dan pendayagunaan hasil penelitian dan pengembangan.
6. Kelompok Jabatan Fungsional: mempunyai tugas melakukan kegiatan sesuai dengan jabatan fungsional masing-masing berdasarkan peraturan perundang-undangan yang berlaku.

2.4 Sumber Daya



Gambar 2.1 Sumber Daya Baristand Industri Surabaya

Baristand Industri Surabaya dalam melaksanakan tugas pokok fungsinya memiliki kekuatan sumber daya manusia dengan komposisi menurut latar belakang pendidikan, kepangkatan / golongan dan status fungsi jabatan memiliki perkembangan seperti ditunjukkan Tabel 2.1, Tabel 2.2 dan Tabel 2.3. Sumber daya manusia Baristand Industri Surabaya juga dapat dikelompokkan berdasarkan keahlian / profesi, termasuk yang telah disertifikasi lembaga personil sebagaimana ditunjukkan Tabel 2.3.

Tabel 2.1 SDM Baristand Surabaya Berdasarkan Tingkat Pendidikan

NO	LATAR BELAKANG PENDIDIKAN	2005	2006	2007	2008	2009	2010	2011	2013	2014	2015	2016	2017	2018
1	S3	-	-	-	-	-	-	-	-	-	-	-	-	1
2	S2	1	3	-	8	16	20	19	22	22	21	13	14	15
3	S1	41	46	44	45	37	45	44	41	42	40	38	36	32
4	D3	10	9	8	7	4	7	7	4	5	6	6	6	7
5	SMA	46	38	42	33	33	26	25	16	16	16	14	12	10
6	SMP	7	5	6	6	6	4	2	1	1	1	1	1	-
7	SD	2	2	-	-	1	1	1	1	1	1	1	-	-
TOTAL		107	103	100	99	97	103	98	85	87	85	76	69	66

Tabel 2.2 SDM Baristand Industri Surabaya Berdasarkan Golongan

NO	GOLONGAN	2005	2006	2007	2008	2009	2010	2011	2013	2014	2015	2016	2017	2018
1	Golongan IV	4	6	7	11	11	13	13	14	14	11	7	6	6
2	Golongan III	84	80	72	67	68	72	70	58	59	59	54	50	47
3	Golongan II	19	17	16	16	16	17	14	13	14	15	15	13	13
4	Golongan I	-	-	-	-	1	1	1	-	-	-	-	-	-
TOTAL		107	103	100	99	97	103	98	85	87	85	76	69	66

Tabel 2.3 SDM Baristand Industri Surabaya Berdasarkan Fungsional

NO	URAIAN	TAHUN												
		2005	2006	2007	2008	2009	2010	2011	2013	2014	2015	2016	2017	2018
1	Fungsional :													
	Peneliti	9	9	9	9	8	7	7	7	9	4	6	8	12
	Perekayasa	-	-	-	-	-	-	-	-	-	-	-	-	2
	Penyuluh													
	Perindustrian dan Perdagangan	5	5	3	3	2	2	2	2	2	2	1	-	-
	Penguji Mutu Barang	6	6	6	6	2	2	2	2	5	4	4	7	7
	Pengendali Dampak Lingkungan	10	10	9	9	9	8	3	2	2	1	1	1	1
	Analisis Kepegawaian	2	2	2	1	1	1	-	-	-	-	-	-	-
	Auditor Manajemen Industri	-	-	-	-	-	-	-	-	-	-	-	-	3
	Pranata Hubungan Masyarakat	3	3	2	2	2	2	-	-	-	-	-	-	-
	Arsiparis	2	1	1	1	1	1	1	1	1	1	1	1	-
	Dokter	-	-	1	1	1	1	1	1	1	1	1	1	1
	Pustakawan	-	-	-	-	-	-	-	-	-	-	-	-	1
2	Non Fungsional	70	67	67	67	71	79	82	66	65	66	61	58	45
TOTAL		107	103	100	99	97	103	98	85	87	85	81	76	69

Pada tahun 2018, Baristand Industri Surabaya juga masih melakukan sub kontrak tenaga ahli tertentu yang keahliannya belum dimiliki pegawai Baristand Industri Surabaya dan / atau jumlahnya belum mencukupi kebutuhan (umumnya untuk auditor dan tenaga ahli). Disamping itu, Baristand Industri Surabaya juga masih melakukan *outsourcing* untuk teknisi laboratorium kimia lingkungan, teknisi

laboratorium fisika, administrasi pelayanan, administrasi kepegawaian, administrasi keuangan, tenaga kebersihan dan tenaga keamanan.

2.5 Moto dan Logo



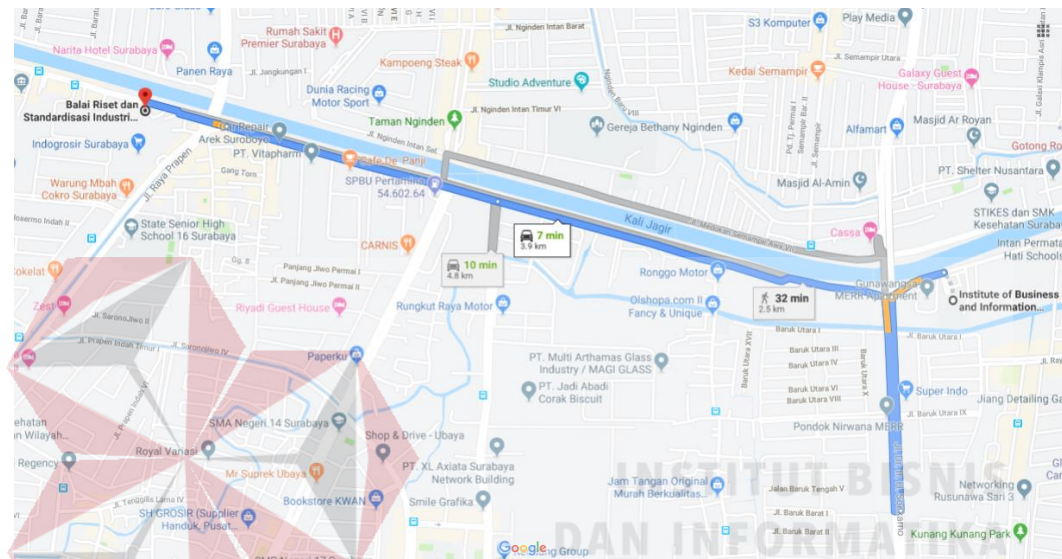
Gambar 2.2 Moto Baristand Industri Surabaya



Gambar 2.3 Logo Baristand Industri Surabaya

2.6 Lokasi Perusahaan

Lokasi Balai Riset dan Standardisasi Industri Surabaya, terletak di Jl. Jagir Wonokromo No. 360 Surabaya, Jawa Timur. Adapun peta lokasi Baristand Industri Surabaya dari Institut Bisnis dan Informatika Stikom Surabaya sebagai berikut, dengan jarak 2,5 km dan estimasi 10 menit dengan keadaan lancar.



Gambar 2.4 Lokasi Baristand Industri dari Stikom Surabaya

BAB III

LANDASAN TEORI

Pada bab tiga, menjelaskan mengenai landasan teori dari Kerja Praktik di Balai Riset dan Standardisasi Industri Surabaya, yang memuat perangkat keras maupun perangkat lunak yang digunakan untuk memenuhi pekerjaan selama Kerja Praktik berlangsung, meliputi Modul RFID MFRC522, NFC Tag 203, Arduino Nano, NodeMCU V3, I2C LCD Karakter 16 x 2, Microsoft SQL Server 2017, Microsoft Visual Studio C#, XAMPP dan PHP.

3.1 RFID (*Radio Frequency Identification*)



Gambar 3.1 Jenis-Jenis RFID Tag

Radio Frequency Identification (RFID) merupakan sistem nirkabel yang menggunakan gelombang radio untuk membaca data yang terletak dalam tag. RFID menggunakan medan elektromagnetik untuk berkomunikasi dengan tag. Pembaca

RFID mampu membaca tag pada jarak 1 cm hingga 100 cm. Pada jenis tertentu, pembacaan dapat mencapai ratusan meter (Kadir, 2018).

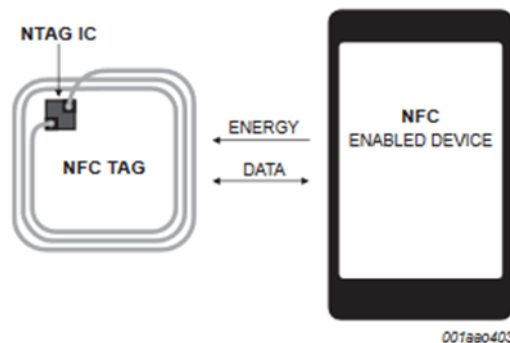
Tag dapat bersifat aktif maupun pasif. Tag pasif tidak memiliki sumber listrik. Tag seperti ini mendapatkan energi dari medan elektromagnetik yang dipancarkan oleh pembaca RFID dan berada di dekat tag sehingga pembaca RFID mampu membaca data yang tersimpan di tag. Tag aktif memiliki baterai, sehingga jarak antara pembaca RFID dan tag relatif lebih jauh. Sejatinya tag aktif relatif lebih mahal dibandingkan dengan tag pasif. Adapun pengelompokan frekuensi agar tag, sebagai berikut:

1. Tag Pasif

- a. 125-134 KHz (*Low Frequency*): Jarak baca sekitar 1-10 cm. Umumnya, tag ini digunakan untuk pelacakan hewan.
- b. 13.56 MHz (*High Frequency & Near-Field Communication/NFC*): Jarak baca dari 1 cm hingga 100 cm. Umumnya, tag ini untuk pengaman paspor atau identitas buku di perpustakaan.
- c. 865-960 MHz (*Ultra High Frequency / UHF*): Biasanya digunakan untuk pembacaan data jarak jauh. Beberapa jenis tag ini mampu dibaca pada jarak 30 meter. Tag ini misalnya digunakan pada pembayaran tol.

2. Tag Aktif, menggunakan dua jenis frekuensi, yaitu 433 HHZ dan 915 MHz. jenis tertentu pada tag ini mampu memancarkan data hingga 100 meter.

3.2 Ultralight Mifare NFC Tag 203



Gambar 3.2 Komunikasi NFC Tag dengan Perangkat Pembaca dan Penulis

Ultralight Mifare NFC Tag 203, merupakan tag berjenis pasif yang telah memenuhi ISO/IEC 14443A, bekerja pada frekuensi 13.56 MHz dengan jarak operasi 100 mm dan kecepatan laju data sebesar 106 Kbps. Tag ini dapat bekerja dengan maksimal jika IC pada tag terhubung dengan sehingga medan elektromagnetik mengaktifkan IC dan data pada tag dapat ditransmisikan pada Pembaca RFID (NXP Semiconductors, 2011).

Spesifikasi EEPROM dari tag ini ialah 168 bytes memori total, yang dibagi dalam 42 *pages* (4 bytes per *pages*), 144 bytes area memori untuk baca dan tulis data, yang dibagi dalam 36 *pages* (4 bytes per *pages*). Tag ini mampu menyimpan data selama 5 tahun, dengan daya tahan 10000 kali penulisan data.

NFC Tag 203 ini, mampu menghasilkan kode unik yang biasa disebut UID (*Unique Identifier*) sepanjang 7 bytes.

3.3 Modul RFID MFRC522

RFID-RC522 atau biasa disebut MFRC522 adalah jenis modul pembaca dan penulis RFID yang beroperasi pada frekuensi 13.56 MHz (Kadir, 2018). Untuk

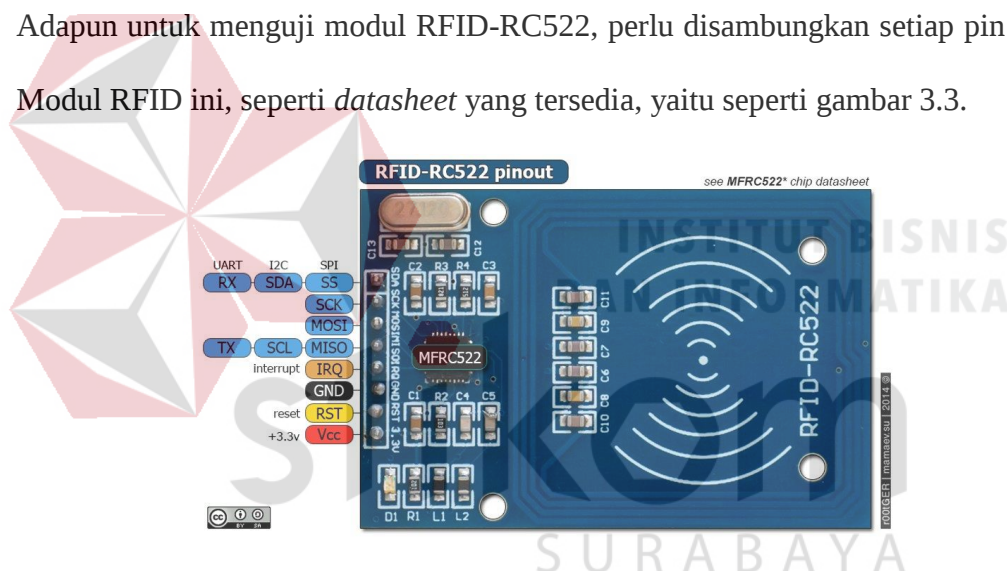
mempermudah penggunaan modul RFID ini, diperlukan *library* bernama MFRC522, yang dapat diunduh pada alamat:

<https://github.com/miguelbalboa/rfid>

Cara menginstal *library* ini adalah sebagai berikut:

1. Mengklik menu *Sketch* pada program Arduino IDE.
2. Lalu pilih menu *Include Library*.
3. Selanjutnya mengklik Add .ZIP Library.
4. Lalu carilah file bernama “rfid-master.zip” dan selanjutnya klik tombol Open.

Adapun untuk menguji modul RFID-RC522, perlu disambungkan setiap pin pada Modul RFID ini, seperti *datasheet* yang tersedia, yaitu seperti gambar 3.3.



Gambar 3.3 Pin Modul RFID RC-522

3.4 Arduino Nano



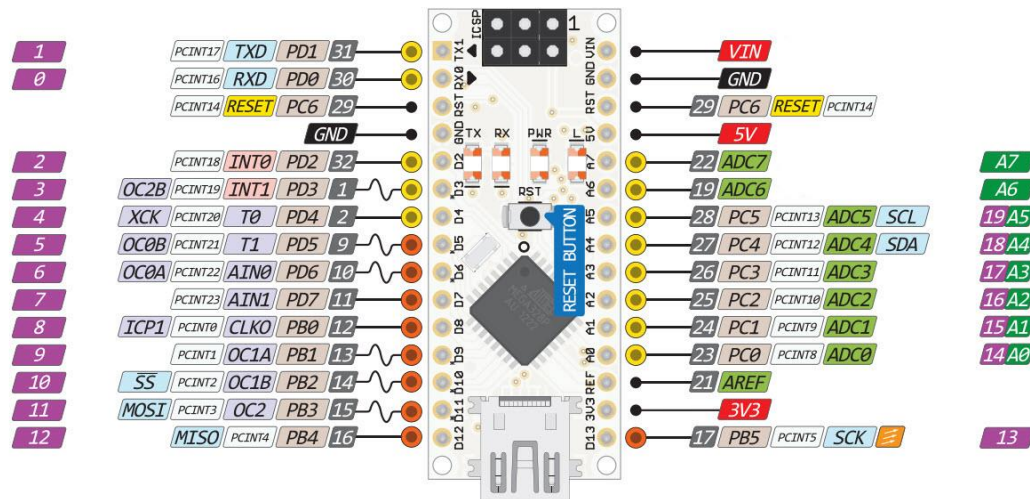
Gambar 3.4 Papan Arduino Nano

Papan Arduino adalah jenis papan elektronis yang saat ini populer untuk mempelajari ataupun mewujudkan berbagai proyek elektronika dan melibatkan pemrograman. Papan Arduino sebenarnya sangat bervariasi, dari ukuran kecil bernama Lilypad, yang dapat ditempelkan di baju hingga Arduino Yun yang mendukung koneksi ke internet secara langsung.

Dari sekian jenis papan Arduino yang tersedia, Arduino Nano (Gambar 3.4) merupakan papan Arduino yang sangat kompleks dan efisien, karena memiliki bentuk yang kecil, namun memiliki komponen pendukung yang cukup, diantaranya 14 Pin Digital *Input / Output*, 6 diantaranya merupakan pin PWM. Adapun berikut merupakan spesifikasi dari Arduino Nano (Tabel 3.1).

Tabel 3.1 Spesifikasi Arduino Nano

Spesifikasi Arduino Nano	
Mikrokontroler	ATmega328
Arsitektur	AVR
Operasi Volt	5V
Memori Flash	32 KB dimana 2 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB
Kecepatan <i>Clock</i>	16 MHz
Pin Analog	8
EEPROM	1 KB
Arus DC per Pin I/O	40 mA
Voltase <i>Input</i>	7 - 12 V
Pin Digital	22 (6 Diantaranya PWM)
<i>Output</i> PWM	6
Konsumsi Daya	19 mA
Ukuran PCB	18 x 45 mm
Berat	7 g



Gambar 3.5 Pin Arduino Nano

3.5 NodeMCU V3

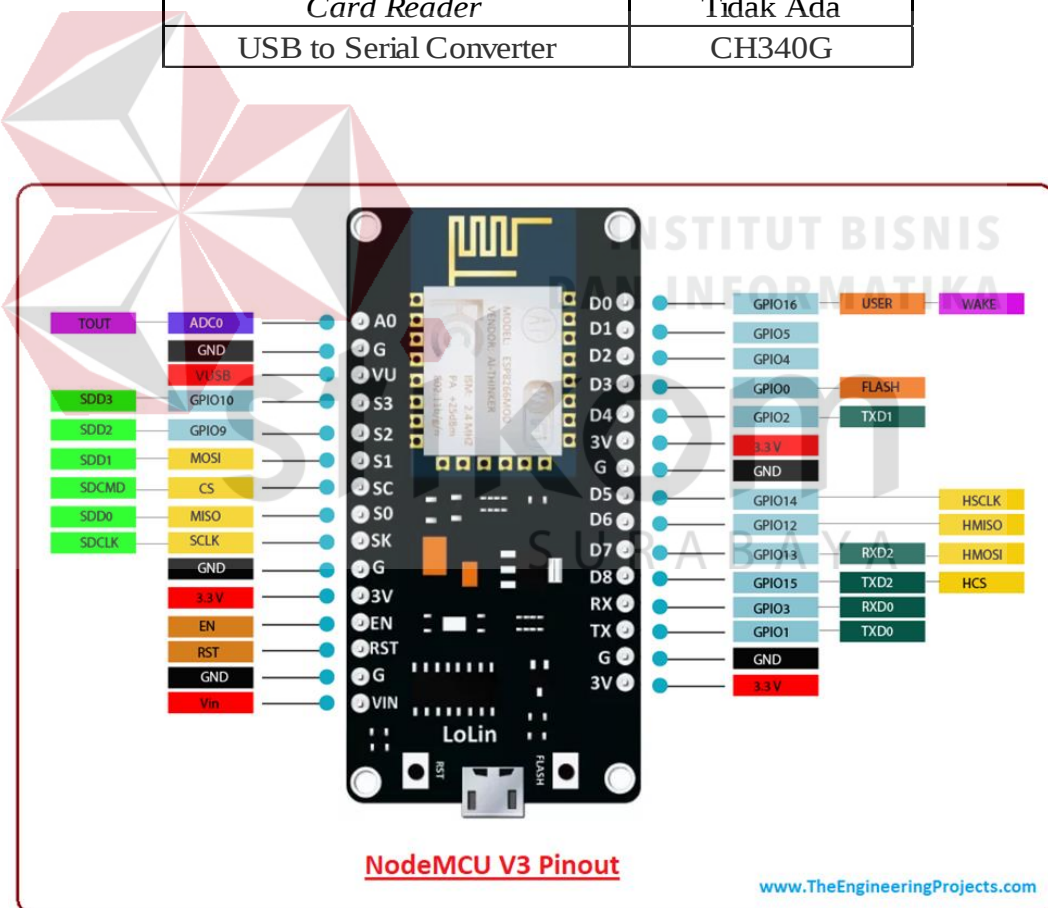
NodeMCU adalah sebuah platform IoT (*internet of things*) yang bersifat *open source*, yaitu sebuah *board* elektronik yang di dalamnya sudah memiliki *firmware* dan *hardware* yang memiliki fitur WiFi (Komputronika, 2019).

NodeMCU memiliki *firmware* yang berjalan pada chip ESP8266 – sebuah *chip* buatan perusahaan Espressif Systems, dan *hardware* berupa modul ESP-12. Istilah “NodeMCU” itu sendiri mengacu pada *firmware*, tidak pada kit elektroniknya. *Firmware* bawaan NodeMCU menggunakan bahasa pemrograman Lua, yaitu bahasa yang merupakan proyek eLua dan dibuat berdasarkan SDK Espressif untuk ESP8266. Selain itu *firmware* NodeMCU juga menggunakan banyak proyek *open source*, seperti lua-cjson dan SPIFFS.

NodeMCU dapat diprogram menggunakan bahasa Arduino (C/C++), Lua dan Micropython. Untuk memprogram NodeMCU menggunakan Arduino, dapat menggunakan Arduino IDE, sedangkan untuk menggunakan Lua atau Micropython, dapat menggunakan software ESPlorer.

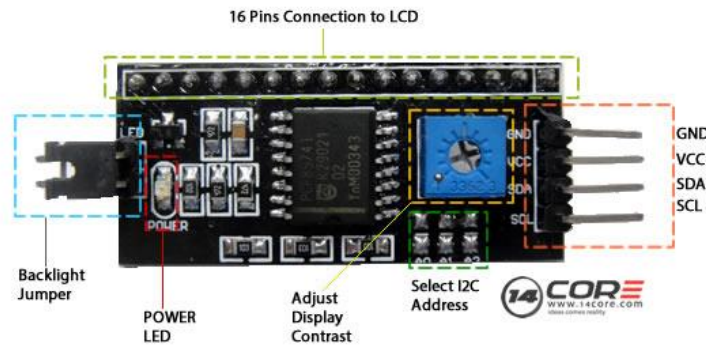
Tabel 3.2 Spesifikasi NodeMCU V3

Spesifikasi	NodeMCU V3
Mikrokontroler	ESP8266
Ukuran <i>Board</i>	57 mm x 30 mm
Tegangan <i>Input</i>	3.3v ~ 5.0 V
GPIO	13 Pin
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Memori <i>Flash</i>	4 MB
<i>Clock Speed</i>	40/26/20 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz - 22.5 GHz
<i>Port USB</i>	<i>Micro USB</i>
<i>Card Reader</i>	Tidak Ada
USB to Serial Converter	CH340G



Gambar 3.6 Pin NodeMCU V3

3.6 I2C LCD Karakter 16 x 2



Gambar 3.7 Pin I2C LCD Karakter 16 x 2

Pada umumnya LCD Karakter dikendalikan dengan menggunakan metode paralel, metode ini banyak menggunakan pin dari mikrokontroler, enam sampai tujuh pin yang digunakan, maka dari itu untuk meminimalisir penggunaan pin digunakan modul I2C untuk mengendalikan LCD karakter 16 x 2, hanya menggunakan dua pin, yaitu SDA dan SCL.



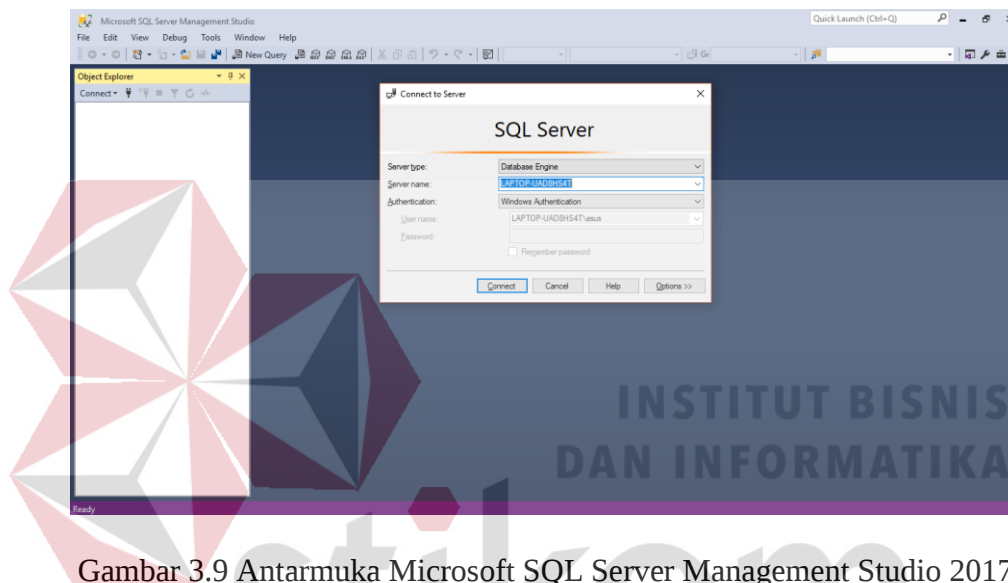
Gambar 3.8 Pin LCD Karakter 16 x 2

3.7 Microsoft SQL Server 2017

Microsoft SQL Server merupakan produk dari Microsoft yang berfungsi sebagai sistem manajemen data relasional. Bahasa *Query* utamanya adalah

Transact-SQL yang merupakan implemementasi dari SQL standar ANSI/ISO yang digunakan oleh Microsoft dan Sybase. Microsoft SQL Server juga mendukung ODBC (*Open Database Connectivity*), dan mempunyai *driver* JDBC untuk bahasa pemrograman Java (Wikipedia, 2016).

3.7.1 Microsoft SQL Server Management Studio 2017



Gambar 3.9 Antarmuka Microsoft SQL Server Management Studio 2017

Microsoft SQL Server Management Studio 2017 merupakan program pendukung dari Microsoft SQL Server 2017 untuk membuat dan mengatur database dengan menggunakan antarmuka GUI (*Graphical User Interface*), antarmuka program tersebut, seperti gambar 3.9.

3.7.2 Statement SQL

Statement SQL digunakan untuk mengoperasikan sistem basis data, yang termuat dalam berbagai *Statement*, diantaranya yang paling sering digunakan adalah sebagai berikut:

1. *Statement* SQL INSERT

Statement ini digunakan untuk memasukan data ke dalam sistem basis data SQL. Adapun *Syntax* dari *Statement* tersebut, sebagai berikut.

```
INSERT INTO nama_tabel (kolom1, kolom2, kolom3,...)
VALUES (kolom1, kolom2, kolom3,...);
```

2. *Statement* SQL SELECT

Statement ini digunakan untuk memilih data pada sistem basis data SQL. Adapun *Syntax* dari *Statement* tersebut, sebagai berikut.

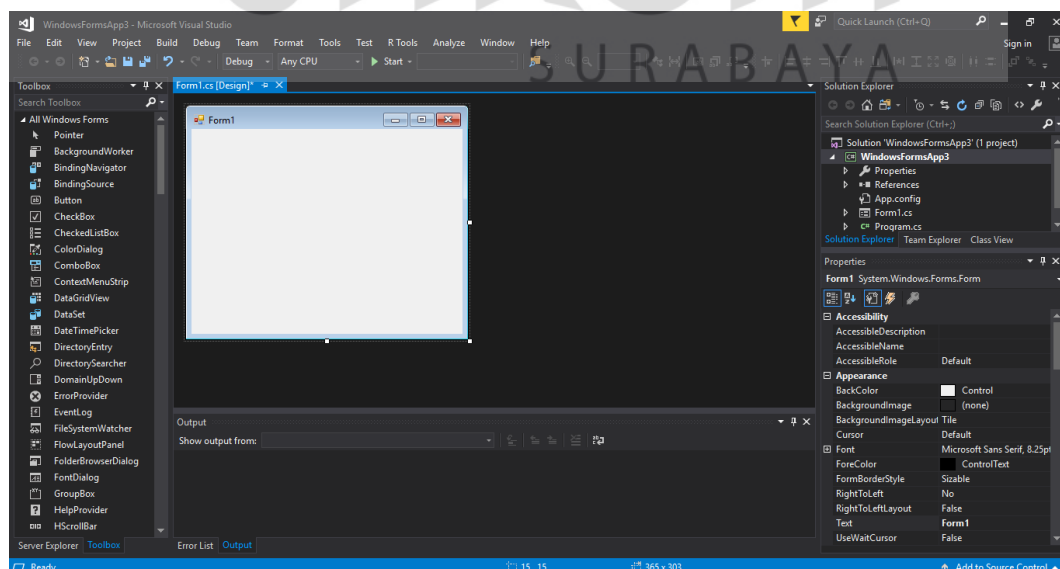
```
SELECT FROM nama_tabel WHERE kondisi;
```

3. *Statement* SQL DELETE

Statement ini digunakan untuk menghapus data pada sistem basis data SQL. Adapun *Syntax* dari *Statement* tersebut, sebagai berikut.

```
DELETE FROM nama_tabel WHERE kondisi;
```

3.8 Microsoft Visual Studio 2017



Gambar 3.10 Antarmuka Microsoft Visual Studio 2017

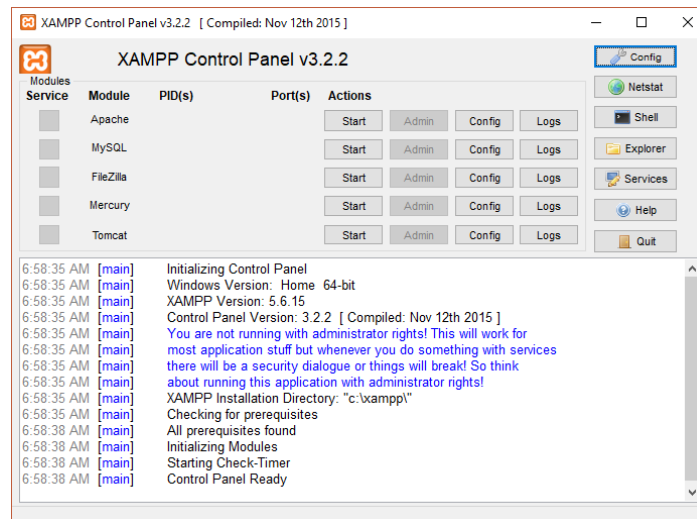
C# adalah bahasa pemrograman komputer, sama seperti C, C++, Java, maupun lainnya. Perbedaan, C# menggunakan *library* kelas yang terdapat pada .NET Framework. Hal ini tentu berbeda dengan C, C++, maupun Java, yang masing-masing, memiliki *library* kelas sendiri-sendiri. Kelebihan *library* kelas yang terdapat di dalam .NET Framework adalah dapat digunakan oleh bahasa-bahasa lain yang mendukung .NET, seperti Visual Basic (VB) dan Visual C++ (VC) (Raharjo, 2015).

Karena C# menggunakan *library* kelas yang terdapat pada .NET Framework, maka kita perlu mempelajari dan melakukan eksplorasi terhadap kelas-kelas tersebut. *Library* kelas dalam .NET Framework merupakan *library* yang sangat besar dan kompleks, dimana informasi mengenai *library* tersebut, dapat diperoleh dari Microsoft Developer Network (MSDN).

3.9 XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl (Dosen Pendidikan, 2019).

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU (*General Public License*) dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis.



Gambar 3.11 Antarmuka XAMPP Control Panel v3.2.2

3.9.1 Bagian dari XAMPP

Adapun bagian dari XAMPP, sebagai berikut :

1. htdocs

Merupakan media tempat diletakkannya file yang akan dijalankan, seperti file PHP, HTML, dan skrip lainnya.

2. phpMyAdmin

Merupakan media untuk mengelola sistem basis data MySQL atau MariaDB, untuk membukanya, perlu digunakan sebuah browser untuk mengakses URL <http://localhost/phpMyAdmin>.

3. Control Panel

Merupakan media untuk mengelola layanan XAMPP, seperti untuk mengatur port, menghidupkan maupun mematikan setiap layanan yang ada pada XAMPP.

3.9.2 Komponen XAMPP

Adapun Komponen XAMPP pada sistem operasi Windows, adalah sebagai berikut:

1. Apache

Merupakan Server aplikasi Web. Apache tugas utama adalah untuk menghasilkan halaman web yang benar kepada pengguna terhadap kode PHP yang sudah dituliskan oleh pembuat halaman web.

2. MySQL

Merupakan Server aplikasi sistem basis data. Pertumbuhannya disebut SQL singkatan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang difungsikan untuk mengolah sistem basis data. MySQL dapat digunakan untuk membuat dan mengelola sistem basis data dan isinya. Bisa juga memanfaatkan MySQL guna untuk menambahkan, mengubah, dan menghapus data dalam sistem basis data.

3. PHP

Merupakan bahasa pemrograman web. Bahasa pemrograman PHP adalah bahasa pemrograman untuk membuat web yang *server-side scripting*. PHP digunakan untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. namun PHP juga mendukung Pengelolaan sistem database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.

4. phpMyAdmin

Merupakan media, untuk mengolah sistem basis data MySQL dan MariaDB.

5. FileZilla FTP Server

Merupakan media, untuk mengolah maupun mentransmisikan data, antara server dan klien.

6. Tomcat

Merupakan web server buatan Apache Foundation yang digunakan untuk menjalankan aplikasi Java berbasis web.

7. Strawberry Perl

Merupakan bahasa pemrograman untuk semua tujuan, pertama kali dikembangkan oleh Larry Wall, mesin Unix. Perl dirilis pertama kali tanggal 18 Desember 1987 yang ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl juga tersedia untuk berbagai sistem operasi Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketPC.

8. XAMPP Control Panel

Merupakan media untuk mengatur layanan yang ada pada XAMPP, seperti untuk mematikan dan menghidupkan layanan.

3.10 PHP

Asal mula PHP (*Personal Home Page*) dimulai pada tahun 1995, yaitu saat seorang pengembang perangkat lunak independen yang bernama Rasmus Ledorf mengembangkan kode Perl/CGI dengan tujuan untuk memonitor siapa saja yang pernah mengunjungi situsnya. Kodenya dibuat dengan dua tugas, yaitu *logging visitor information* dan menampilkan jumlah *visitor* pada halaman web (Wicaksono

& Hidayat, 2017). Program PHP diawali dengan simbol “<?php” dan diakhiri dengan simbol “?>” seperti berikut ini:

```
<?php
// Program PHP
?>
```

3.10.1 Variabel PHP

Merupakan suatu wadah / tempat untuk menyimpan sementara. Data di dalam sebuah variabel dapat dipanggil dan diubah. Dalam menuliskan variabel PHP terdapat beberapa hal yang diperhatikan, sebagai berikut:

1. Variabel diawali dengan simbol dollar “\$”.
2. Nama variabel diawali dengan huruf atau underscore “_”.
3. Dalam menuliskan nama variabel tidak boleh mengandung spasi.
4. Tidak boleh memuat karakter spesial (!, @, #, % dan lain-lain).
5. Bersifat case sensitive.

Contoh penulisan variabel PHP yang benar, sebagai berikut:

```
$_variabelku, $awal_variabelku, $variabel12, $variabel_99
```

3.10.2 Operator Aritmatika PHP

Merupakan operator yang digunakan untuk mengelola operasi aritmatika, seperti penjumlahan, perkalian, pembagian dan pengurangan, sebagai berikut, sesuai tabel 3.3.

Tabel 3.3 Operator Aritmatika PHP

Operator	Keterangan	Contoh Penggunaan
%	Modulo (Sisa Bagi)	\$A % \$B
+	Penjumlahan	\$A + \$B
-	Pengurangan	\$A - \$B
*	Perkalian	\$A * \$B
/	Pembagian	\$A / \$B

3.10.3 Operator Logika PHP

Merupakan operator yang digunakan untuk membandingkan dua buah variabel dan menghasilkan nilai **TRUE** atau **FALSE**. Adapun Operator tersebut sesuai tabel 3.4.

Tabel 3.4 Operator Logika PHP

Operator	Nama	Keterangan
&&	AND	TRUE, jika kedua atau lebih variabel yang dibandingkan bernilai benar
	OR	TRUE, jika salah satu variabel yang dibandingkan bernilai benar
XOR	XOR	TRUE, jika kedua atau lebih variabel yang dibandingkan nilainya berbeda
!	NOT	Operator pembalik nilai (nilai yang benar menjadi salah, atau sebaliknya)

3.10.4 Operator Pembanding PHP

Merupakan operator yang digunakan dalam seleksi kondisi dan perulangan. Operator ini membandingkan dua nilai atau variabel. Adapun operator tersebut sesuai tabel 3.5.

Tabel 3.5 Operator Pembandingan PHP

Operator	Keterangan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
==	Sama dengan
!=	Tidak sama dengan



BAB IV

DESKRIPSI PEKERJAAN

Pada bab empat, menjelaskan mengenai deskripsi pekerjaan yang meliputi Analisis Sistem, Spesifikasi Alat, Skematik Alat, Desain Alat, Pembuatan Server Pembuatan Program, Uji Coba Alat dan Program, serta Hasil Alat.

4.1 Analisis Sistem

Berdasarkan hasil wawancara dan pengkajian yang dilakukan pada saat Kerja Praktik di Balai Riset dan Standardisasi Industri Surabaya, terdapat beberapa masalah pada proses pengujian barang uji, yaitu:

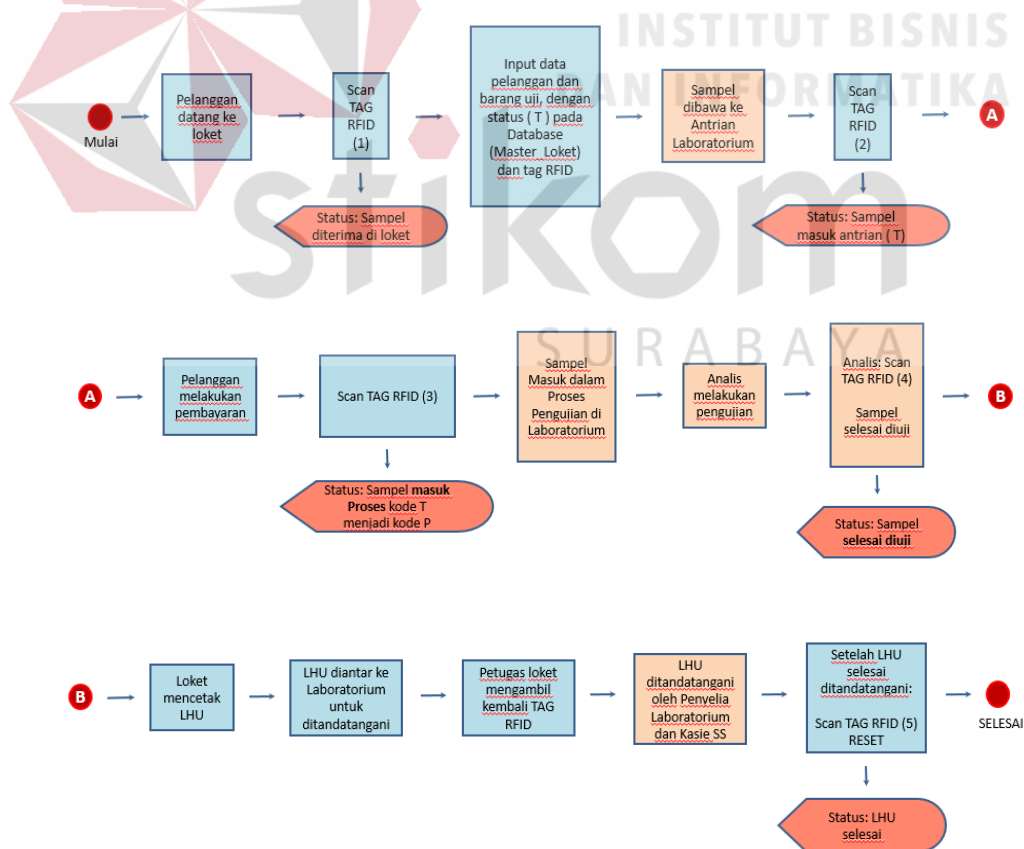
1. Tidak ada rekam jejak barang uji, sehingga pegawai tidak dapat mengetahui posisi proses barang uji dan barang uji rentan hilang.
2. Sistem tidak sepenuhnya terintegrasi secara langsung dengan komputer dan database, khususnya pada tahapan pemberian identitas barang uji, seperti nomor urut barang uji dan status pembayarannya hanya menggunakan kertas dan bolpoin, lalu secara manual dimasukan ke dalam sistem basis data. Hal tersebut sangat rentan dengan kesalahan manusia sehingga dapat mempengaruhi tingkat pelayanan.
3. Pegawai sulit membedakan status barang yang telah dibayar dan belum.

Atas dasar tersebut dibangunlah sebuah Sistem Pelacak Barang Uji Laboratorium Berbasis RFID (*Radio Frequency Identification*) Menggunakan Database SQL Server 2017, guna menjawab permasalahan diatas. Adapun Sistem

Pelacak Barang Uji Laboratorium yang terintegrasi dengan SQL Server 2017 dibangun agar sistem mampu merekam jejak barang uji. Dengan memasukan data pada RFID sebagai identitas setiap barang uji, pegawai dapat mengetahui nomor urut barang dan status pembayaran setiap barang uji yang masuk. Maka dari itu terciptalah sebuah sistem yang dapat terintegrasi dengan komputer dan sistem basis data, sehingga meminimalisir terjadinya kesalahan manusia.

4.2 Diagram Alur Proses Sistem Pelacak Barang Uji Laboratorium

Adapun berdasarkan Analisis Sistem dan Permasalahan yang ada, terbentuklah Diagram Alur ini, untuk menggambarkan Alur Proses Sistem Pelacak Barang Uji Laboratorium sebagai jawaban dari permasalahan yang ada.



Gambar 4.1 Diagram Alur Proses Sistem Pelacak Barang Uji Laboratorium

4.3 Spesifikasi Alat

Berdasarkan perancangan sistem diatas, maka untuk menjawab permasalahan tersebut, dirancanglah alat-alat untuk menunjang setiap tahapan dari proses pelacakan barang uji, yaitu: Scanner Locket, Scanner Laboratorium dan Scanner Reset.

4.3.1 Scanner Locket

Scanner locket dirancang untuk digunakan pegawai yang bertugas dibagian locket tempat pelayanan pelanggan untuk memasukan sampel barang uji. Alat ini digunakan untuk memasukan identitas beserta data pendukung dari sampel barang uji dan pelanggan pada sistem basis data dan untuk memasukan kode urut sampel uji serta status pembayaran dari kode tersebut pada tag RFID. Adapun spesifikasi alat ini seperti pada tabel 4.1.

Tabel 4.1 Spesifikasi Scanner Locket

Spesifikasi	Keterangan
Mikrokontroler	Arduino Nano
Modul RFID	MFRC-522
Tag RFID	Ultralight NFC Tag 203
Antarmuka Pengguna	LCD Karakter 16 x 2 + i2c
Media Transmisi Data	Kabel USB Micro B
Catu Daya	5V dari Kabel USB Komputer

4.3.2 Scanner Laboratorium

Scanner Laboratorium dirancang untuk digunakan pegawai yang bertugas di laboratorium yang merupakan tempat antrian barang uji dan proses pengujian. Alat ini digunakan untuk mengubah status pembayaran dari keterangan belum dibayar

(T) dan telah dibayar (P), sekaligus untuk menginputkan informasi ke sistem basis data secara nirkabel, agar pegawai dapat mengetahui lokasi terakhir dari barang uji. Adapun spesifikasi alat ini seperti tabel 4.2.

Tabel 4.2 Spesifikasi Scanner Laboratorium

Spesifikasi	Keterangan
Mikrokontroler	NodeMCU V3
Modul RFID	MFRC-522
Tag RFID	Ultralight NFC Tag 203
Antarmuka Pengguna	LCD Karakter 16 x 2 + i2c
	Push Button
	Slider Switch
Media Transmisi Data	Jaringan Nirkabel
Catu Daya	Baterai 9V & Regulator 5V

4.3.3 Scanner Reset

Scanner Reset dirancang untuk digunakan pegawai yang bertugas di loket. Alat ini digunakan untuk mengatur ulang (*reset*) tag RFID agar dapat digunakan kembali oleh sampel barang uji lainnya, sekaligus menghapus rekam jejak barang uji pada sistem basis data, karena barang tersebut telah selesai melewati tahapan pengujian dan administrasi. Adapun spesifikasi alat tersebut seperti tabel 4.3.

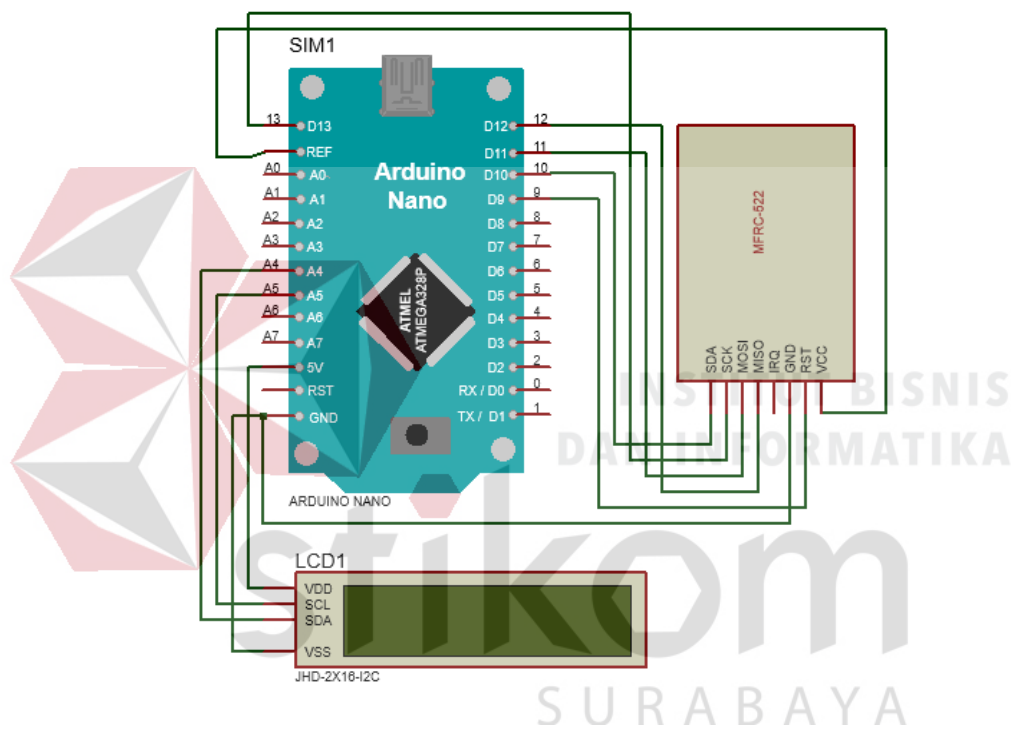
Tabel 4.3 Spesifikasi Scanner Reset

Spesifikasi	Keterangan
Mikrokontroler	NodeMCU V3
Modul RFID	MFRC-522
Tag RFID	Ultralight NFC Tag 203
Antarmuka Pengguna	LCD Karakter 16 x 2 + i2c
	Push Button
Media Transmisi Data	Jaringan Nirkabel
Catu Daya	5V dari Kabel USB

4.4 Skematik Alat

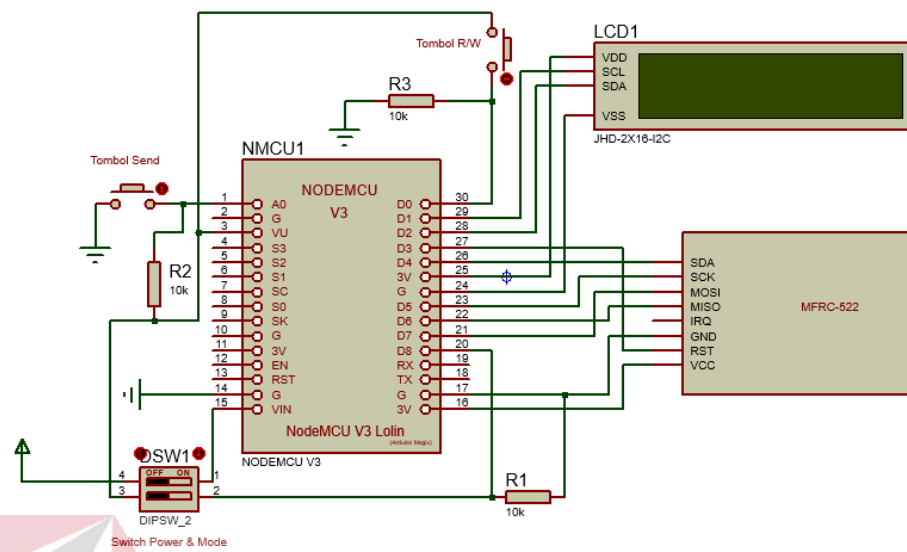
Adapun skematik alat yang dibuat berdasarkan spesifikasi alat diatas, terbagi atas tiga skematik, yaitu: Skematik Scanner Locket, Skematik Scanner Laboratorium dan Skematik Scanner Reset.

4.4.1 Skematik Scanner Locket



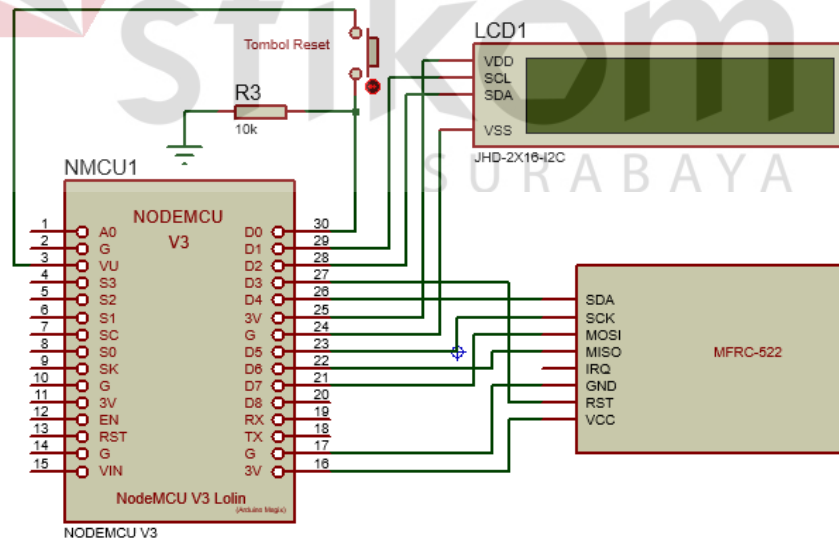
Gambar 4.2 Skematik Alat Scanner Locket

4.4.2 Skematik Scanner Laboratorium



Gambar 4.3 Skematik Scanner Laboratorium

4.4.3 Skematik Scanner Reset

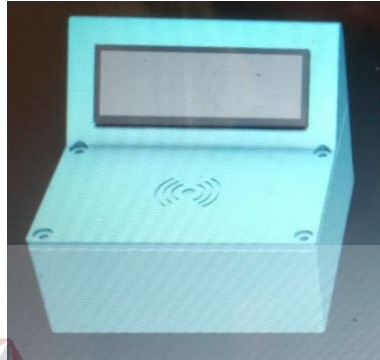


Gambar 4.4 Skematik Scanner Reset

4.5 Desain Alat

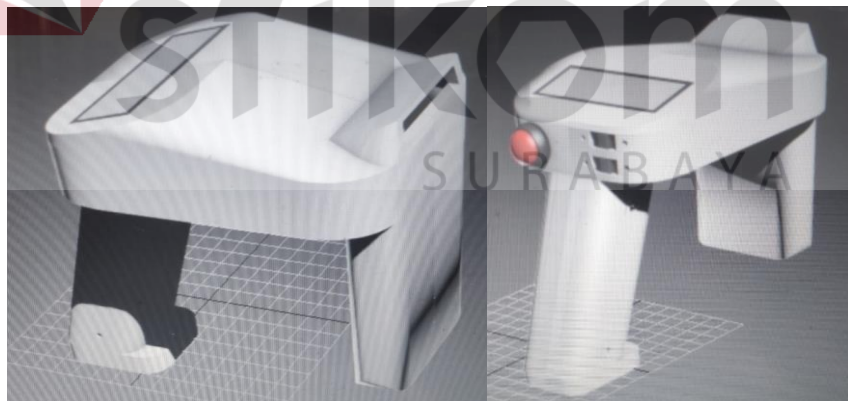
Adapun Desain 3D Alat Scanner, yang terdiri atas Desain 3D Scanner Locket, Scanner Laboratorium dan Scanner Reset.

4.5.1 Desain 3D Scanner Locket



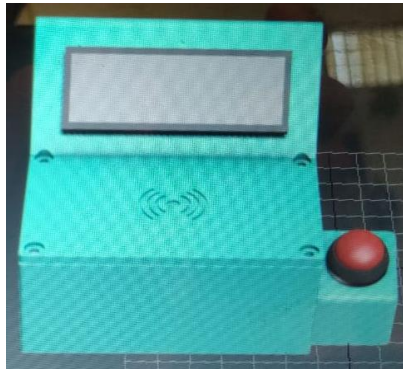
Gambar 4.5 Desain 3D Scanner Locket

4.5.2 Desain 3D Scanner Laboratorium



Gambar 4.6 Desain 3D Scanner Laboratorium

4.5.3 Desain 3D Scanner Reset

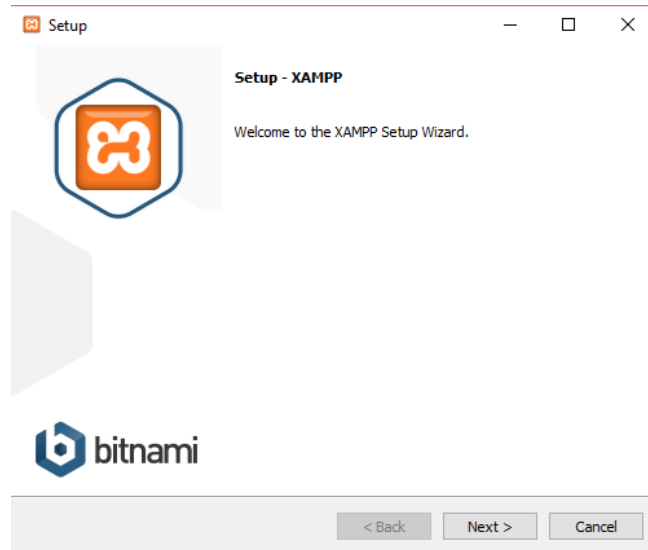


Gambar 4.7 Desain 3D Scanner Reset

4.6 Pembuatan Server

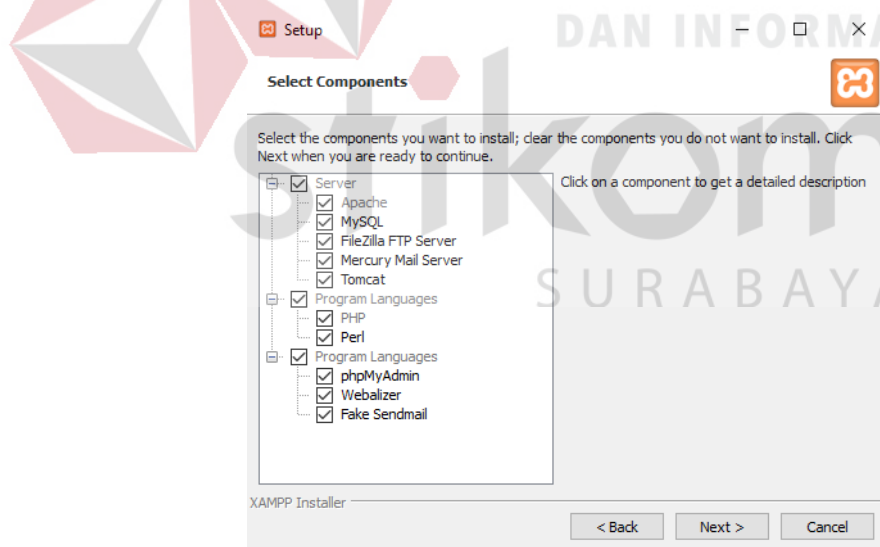
Adapun server dibuat sebagai media transmisi data dari mikrokontroler NodeMCU V3 ke database Microsoft SQL Server 2017. Karena NodeMCU bekerja melalui media jaringan nirkabel, sehingga untuk mentransmisikan data ke database, maka perlu jembatan Protokol HTTP melalui perantara kode PHP yang disematkan pada server, sehingga komputer server dapat berkomunikasi langsung dengan NodeMCU V3. Adapun server yang dimaksud ialah komputer server yang sifatnya lokal menggunakan bantuan perangkat lunak XAMPP, tata cara instalasi aplikasi tersebut, adalah sebagai berikut:

1. Perangkat lunak XAMPP, dapat diunduh pada alamat.
www.apachefriends.org/download.html
2. Setelah XAMPP diunduh, langkah berikutnya membuka aplikasi XAMPP.
3. Lalu mengklik Next untuk melanjutkan instalasi.



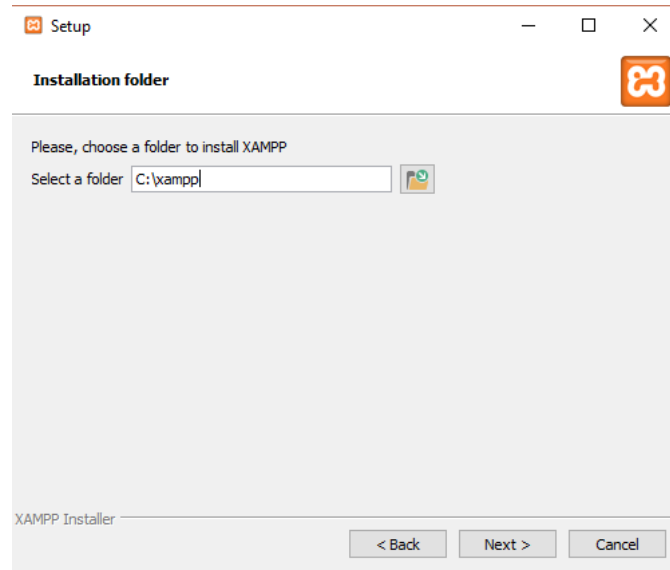
Gambar 4.8 Tampilan Awal Instalasi XAMPP

4. Langkah berikutnya memilih komponen yang akan diinstalasi, setelah itu mengklik tombol Next.



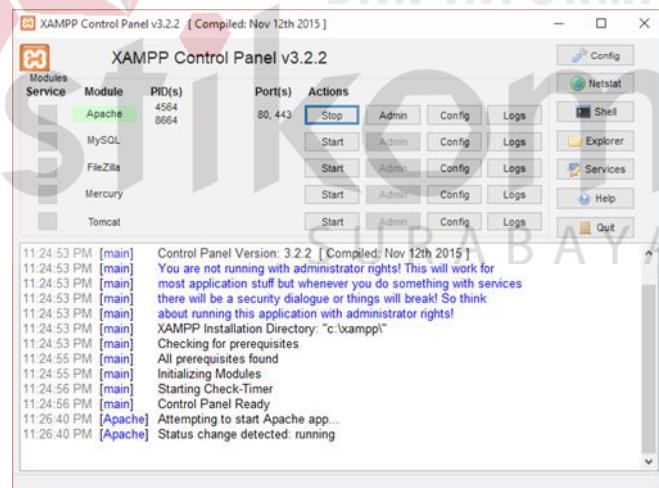
Gambar 4.9 Komponen XAMPP Yang Akan Diinstalasi

5. Langkah berikutnya adalah memilih folder instalasi (direkomendasikan pada C:\xampp), kemudian mengklik tombol Next untuk melakukan instalasi.



Gambar 4.10 Pemilihan Folder Untuk Instalasi XAMPP

6. Setelah proses instalasi selesai, selanjutnya mengklik finisih.
7. Untuk memulai Server, mengklik tombol Start pada Apache.



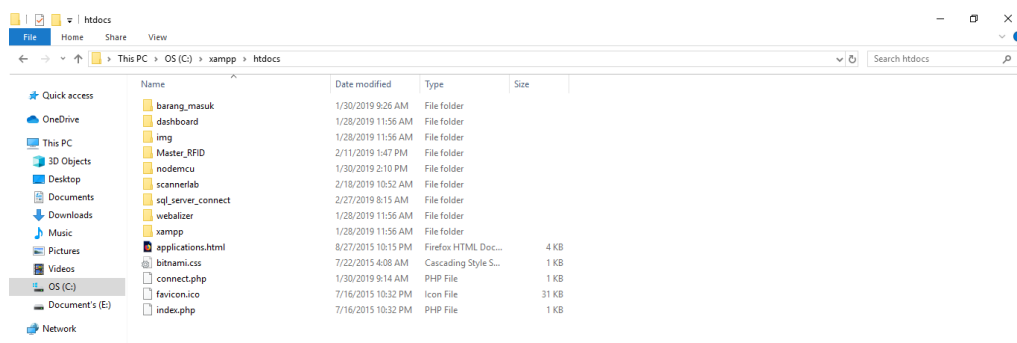
Gambar 4.11 Tampilan XAMPP Control Panel

8. Langkah selanjutnya untuk memastikan server telah berjalan dengan baik maka, alamat localhost/dashboard dibuka pada browser dan akan tampil antarmuka seperti gambar 4.12.



Gambar 4.12 Tampilan Awal Localhost (Server Lokal)

9. Adapun untuk menyimpan kode PHP pada server, maka harus diletakkan pada direktori `c:\xampp\htdocs` dan pada folder “htdocs” ditambahkan folder “sql_server_connect” sebagai tempat kode PHP yang terintegrasi dengan mikrokontroler dan database.

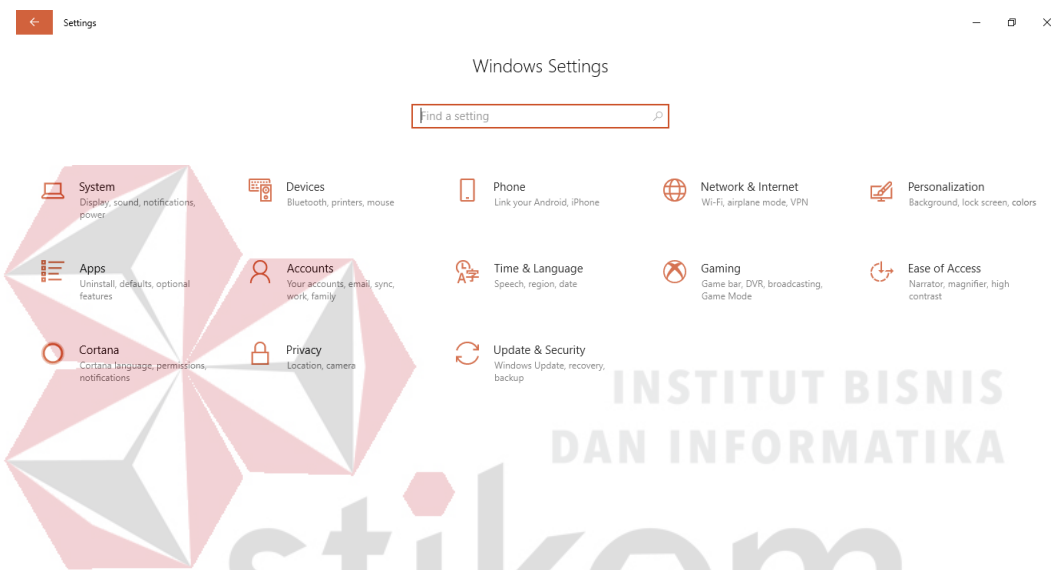


Gambar 4.13 Tampilan Direktori Server XAMPP

4.7 Pembuatan Jaringan Lokal

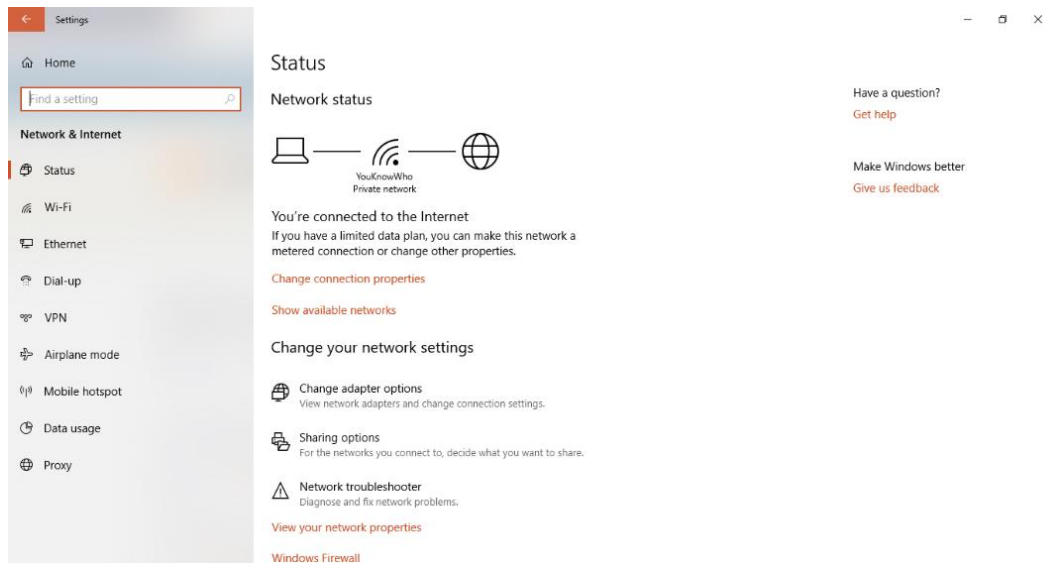
Adapun pembuatan jaringan lokal ini bertujuan untuk menghubungkan mikrokontroler NodeMCU V3 dengan server lokal yang telah dibangun. Operasi Sistem yang digunakan pada saat pembuatan jaringan lokal ini menggunakan Windows 10. Pembuatan jaringan lokal itu sebagai berikut.

1. Membuka Menu Windows Settings



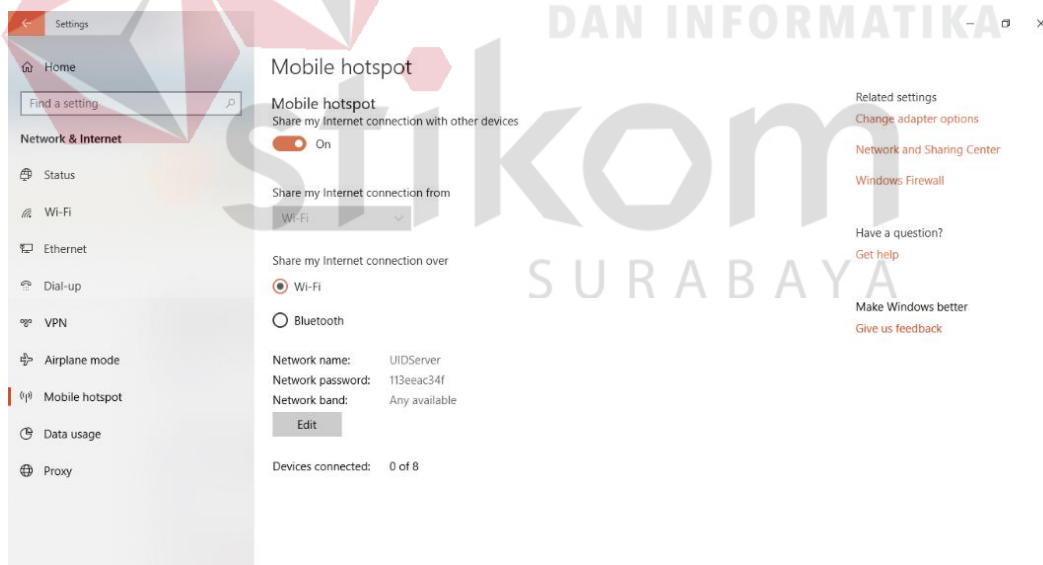
Gambar 4.14 Tampilan Menu Windows Settings

2. Lalu mengklik menu Network & Internet, maka akan tampil antarmuka seperti gambar dibawah ini.



Gambar 4.15 Tampilan Menu Network & Internet

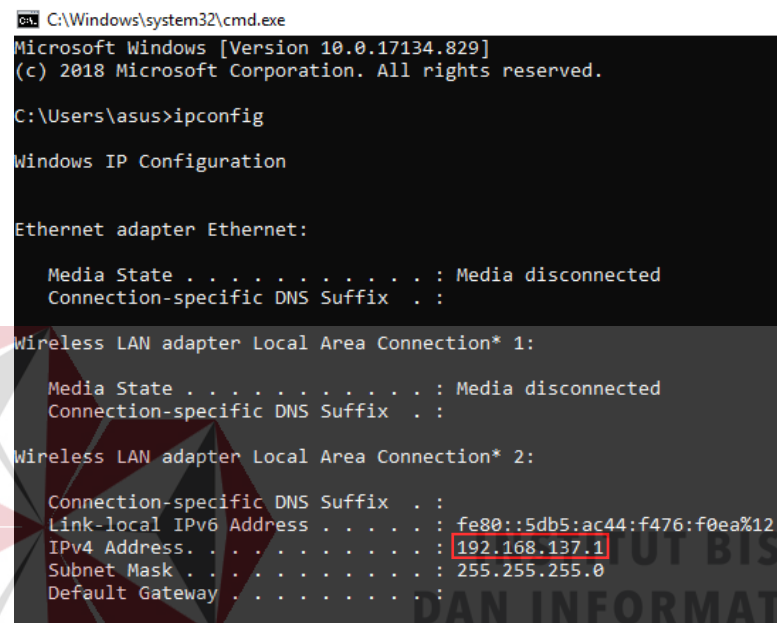
3. Selanjutnya mengklik menu Mobile hotspot, lalu mengklik tombol off untuk mengubah pengaturan menjadi on.



Gambar 4.16 Tampilan Menu Mobile Hotspot

4. Mengklik tombol edit, untuk mengatur Network name dan Network password.

5. Selanjutnya, diperlukan alamat IP Address untuk penghubung antara server yang telah dibuat dengan mikrokontroler NodeMCU V3. Membuka Command Prompt, lalu mengetik “ipconfig”. Adapun IP Address yang didapatkan adalah 192.168.137.1.



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.829]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\asus>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::5db5:ac44:f476:f0ea%12
    IPv4 Address. . . . . : 192.168.137.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
  
```

Gambar 4.17 Tampilan Command Prompt Perintah ipconfig

6. Adapun Jaringan lokal telah berhasil dibuat. Untuk memastikan jaringan tersebut dapat terkoneksi dengan server adalah dengan membuka pada browser alamat 192.168.137.1/dashboard. Jika berhasil, maka jaringan lokal dan server telah siap dipakai.



Gambar 4.18 Tampilan Server Yang Telah Terintegrasi Jaringan Lokal

4.8 Pembuatan Sistem Basis Data

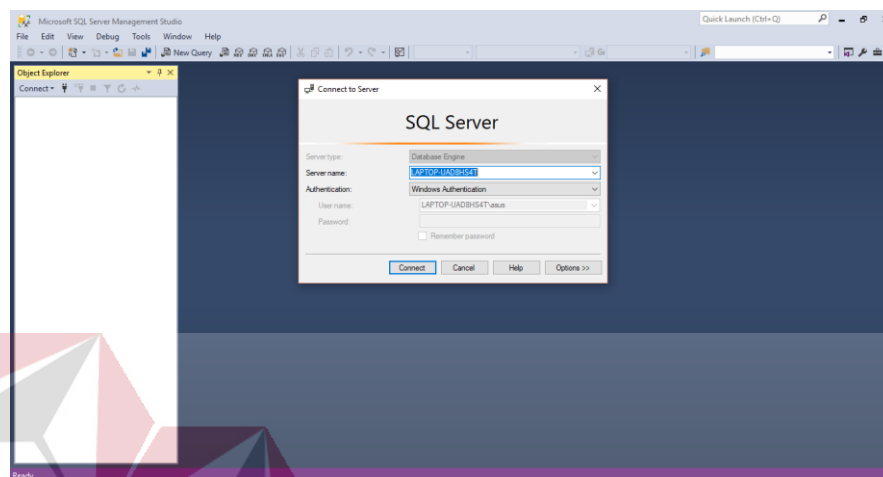
Sistem basis data digunakan sebagai tempat penyimpanan data identitas sampel barang uji dan data pelanggan, serta rekam jejak dari barang uji. Sistem basis data atau yang biasa disebut database ini menggunakan Microsoft SQL Server 2017. Adapun spesifikasi database yang dibuat sesuai dengan tabel dibawah ini.

Tabel 4.4 Spesifikasi Database RFID_Scanner

Nama Database : RFID_Scanner		Keterangan
Tabel	Master_Loket	Database Data Identitas Barang Uji dan Pelanggan
	Antrian_LFisika	Database Rekam Jejak Barang Uji dalam Antrian
	Proses_LFisika	Database Rekam Jejak Barang Uji dalam Proses
	Selesai_LFisika	Database Rekam Jejak Barang Uji telah Selesai
	Reset_LFisika	Database Rekam Jejak Barang Uji yang telah direset
	tb_user	Database Hak Akses Aplikasi Desktop Locket

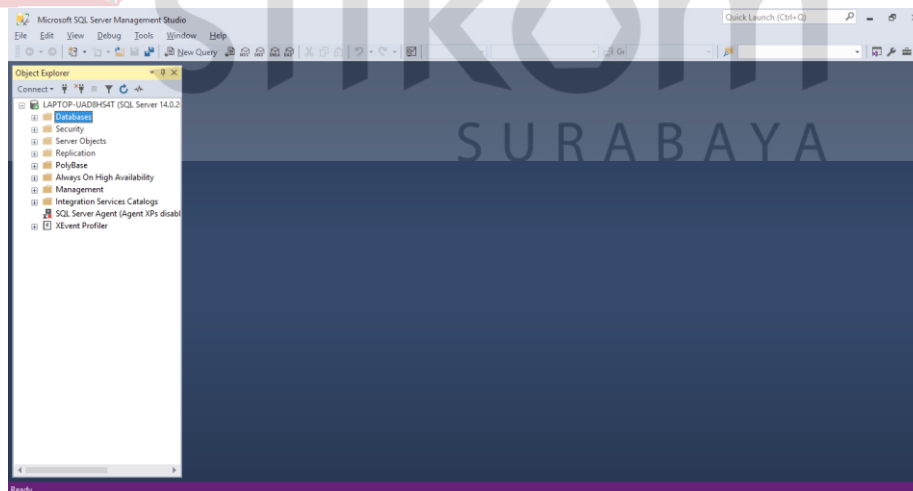
Adapun untuk membuat sistem basis data yang sesuai spesifikasi diatas adalah sebagai berikut:

1. Membuka program Microsoft SQL Server Management Studio 2017.



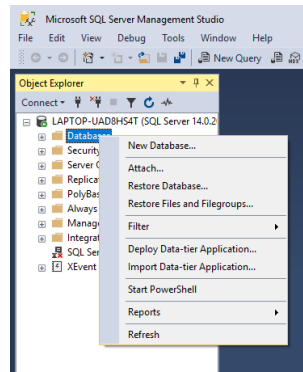
Gambar 4.19 Tampilan Microsoft SQL Server Management Studio 2017

2. Selanjutnya mengklik Connect, maka tampilan seperti gambar dibawah ini.



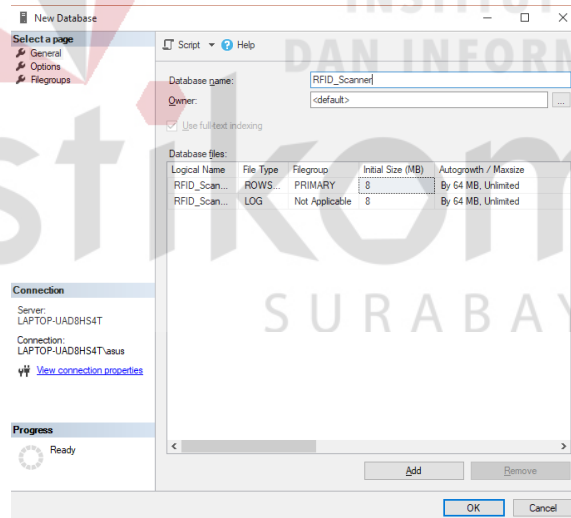
Gambar 4.20 Terbuka Microsoft SQL Server Management Studio 2017

- Setelah itu, mengklik kanan pada bagian database, lalu mengklik “New Database”.



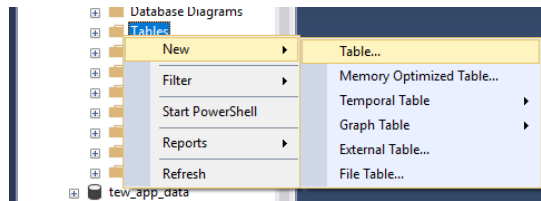
Gambar 4.21 Tampilan New Database

- Selanjutnya isi nama database sesuai dengan tabel spesifikasi, lalu mengklik “ok”.



Gambar 4.22 Tampilan Menu New Database

- Setelah itu mengklik kanan pada tables, lalu mengklik New Table.



Gambar 4.23 Mengklik Kanan New Table

6. Selanjutnya mengisi tabel seperti gambar dibawah, dan menyimpan dengan nama Master_Loket, untuk membuat tabel program desktop loket. Sebagai catatan bahwa secara khusus kolom “uid” diatur sebagai *Primary Key*.

Column Name	Data Type	Allow Nulls
uid	varchar(50)	☐
tanggal_masuk	varchar(50)	☒
nama_barang	varchar(50)	☒
kode_barang	varchar(50)	☒
nama_perusahaan	varchar(50)	☒
nama_pengirim	varchar(50)	☒
jenis_sample	varchar(50)	☒
merk	varchar(50)	☒
volume	varchar(50)	☒
bentuk	varchar(50)	☒
parameter_uji	varchar(50)	☒
biaya_uji	varchar(50)	☒
metode_uji	varchar(50)	☒
lokasi	varchar(50)	☒

Gambar 4.24 Tabel Master_Loket

7. Selanjutnya mengisi ulang tahapan ke lima hingga empat kali dan mengisi kolom seperti gambar dibawah ini, untuk membuat tabel dengan nama Antrian_LFisika, Proses_LFisika, Selesai_LFisika dan Reset_LFisika. Pemberian nama ini disesuaikan dengan rencana penerapan sistem ini pada Laboratorium Fisika. Sebagai catatan bahwa secara khusus kolom “uid” diatur sebagai *Primary Key*.

	Column Name	Data Type	Allow Nulls
►	uid	varchar(50)	☐
	kode_barang	varchar(50)	☒
	lokasi	varchar(50)	☒

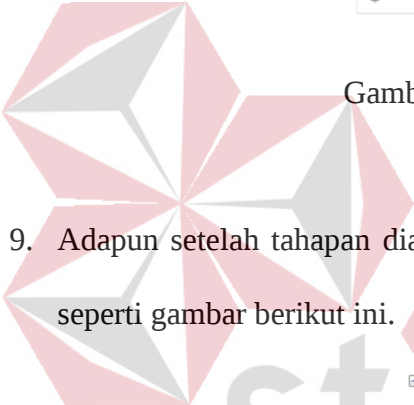
Gambar 4.25 Isi Tabel Antrian, Proses, Selesai dan Reset

8. Selanjutnya membuat tabel tb_user, yang berfungsi untuk hak ases pengguna program desktop loket. Adapun isi tabel tersebut ialah sebagai berikut.

	username	password
►	admin	admin123
*	NULL	NULL

Gambar 4.26 Isi Tabel tb_user

9. Adapun setelah tahapan diatas dijalankan, memastikan tabel yang dihasilkan seperti gambar berikut ini.



RFID_Scanner
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.Antrian_LFisika
dbo.Master_Loket
dbo.Proses_LFisika
dbo.Reset_LFisika
dbo.Selesai_LFisika
dbo.tb_user

Gambar 4.27 Isi Database RFID_Scanner

4.9 Pembuatan Program

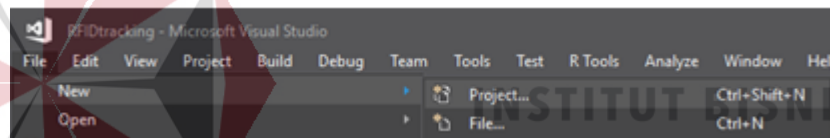
Adapun pembuatan program terdiri atas pembuatan program Desktop Locket yang dibuat dengan Microsoft Visual Studio 2017 dan Program Mikrokontroler yang terdiri dari Scanner Locket, Scanner Laboratorium, Scanner Reset, serta

program PHP untuk jembatan transmisi data dari program mikrokontroler dengan Database.

4.9.1 Pembuatan Program Desktop Locket

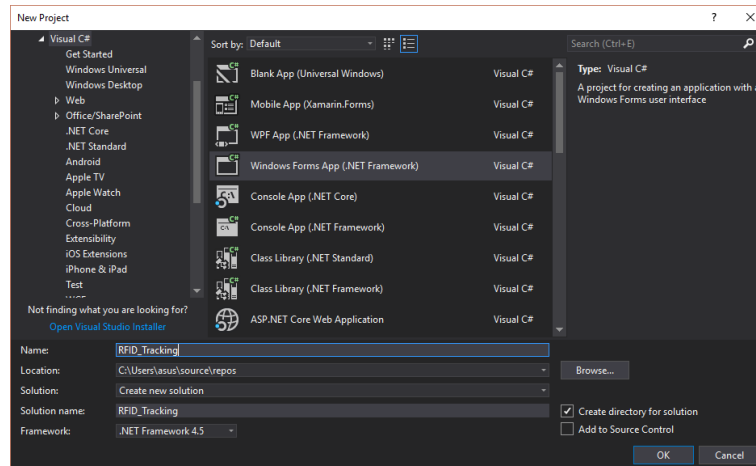
Adapun program Desktop Locket dirancang untuk digunakan oleh pegawai yang bertugas di loket untuk mengisi identitas sampel barang uji dan pelanggan. Dalam pembuatan program ini, digunakan program Microsoft Visual Studio 2017 dengan .NET Framework 4.5, agar program dapat mendukung Sistem Operasi terendah Windows 10. Adapun berikut adalah tahapan pembuatan tersebut.

1. Mengklik pada toolbar “File”, selanjutnya mengklik New dan memilih Project.



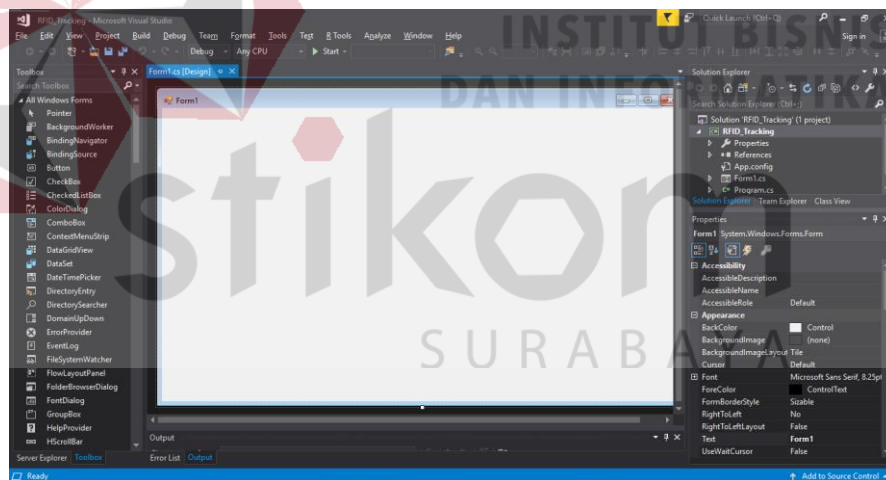
Gambar 4.28 Mengklik File > New Project Microsoft Visual Studio 2017

2. Selanjutnya memilih pada bagian Visual C#, karena bahasa pemrograman yang digunakan dalam pembuatan program ini menggunakan bahasa pemrograman C#. Lalu mengklik “Windows Forms App (.NET Framework), dan memberi nama RFID_Tracking, lalu memilih Framework 4.5., setelah itu mengklik tombol “ok”.



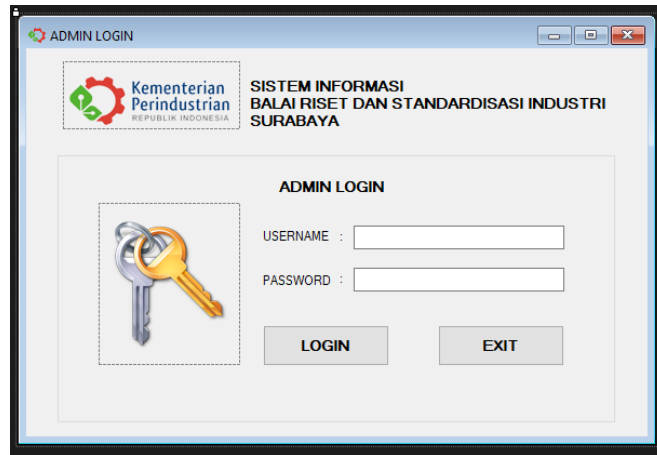
Gambar 4.29 Tampilan Menu New Project

Selanjutnya akan tampil form kosong. Pada form ini komponen-komponen pendukung akan ditaruh.



Gambar 4.30 Tampilan Menu Form Kosong Microsoft Visual Studio 2017

3. Lalu mendesain form Login yang digunakan untuk memberikan hak akses agar aplikasi ini lebih aman, seperti gambar dibawah ini.



Gambar 4.31 Tampilan Menu Login Aplikasi

4. Lalu mengklik tombol login pada form dan memasukkan kode berikut ini.

Pada bagian ini, kode untuk membuka koneksi dengan database SQL Server 2017, dengan sumber dari LAPTOP-UAD8HS4T dan database RFID_Scanner.

```
string connectionString;
SqlConnection cnn;
connectionString = @"Data Source=LAPTOP-UAD8HS4T;Initial
Catalog=RFID_Scanner;Integrated Security=True";
cnn = new SqlConnection(connectionString); cnn.Open();
Pada bagian ini, kode untuk mengirim query dengan statement SELECT pada
database SQL Server 2017, untuk memastikan data yang dimasukan pengguna
sesuai dengan isi tabel tb_user pada database.
```

```
SqlCommand command;
SqlDataAdapter adapter = new SqlDataAdapter();
String sql = "";
sql = "select * from tb_user where username='" + txtUser.Text + "'
and password='" + txtPass.Text + "'";
command = new SqlCommand(sql, cnn);
adapter.InsertCommand = new SqlCommand(sql, cnn);
```

Pada bagian ini, kode untuk menghitung jumlah baris pada tabel dt yang disimpan pada variabel i, yang digunakan sebagai penanda bahwa pengguna memasukkan data yang benar.

```

adapter.InsertCommand.ExecuteNonQuery();
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(command);
da.Fill(dt);
i = Convert.ToInt32(dt.Rows.Count.ToString());

```

Pada bagian ini, kode untuk memastikan apakah data yang dimasukkan pengguna benar atau salah, dilihat dari jumlah baris yang ada pada tabel, jika variabel i bernilai 0, maka ditampilkan peringatan dan bila variabel i bernilai 1, maka form login aplikasi disembunyikan dan selanjutnya form utama ditampilkan, lalu koneksi dengan database diputus.

```

if (i == 0)
{
    MessageBox.Show("Username atau password yang anda masukan salah !");
}
else
{
    this.Hide();
    utama form = new utama();
    form.Show();
}
cnn.Close();

```

5. Selanjutnya mengklik tombol exit pada form login, untuk memberikan kode menutup form login dan aplikasi tersebut, seperti gambar dibawah ini.

```

this.Close();
Application.Exit();

```

6. Setelah itu membuat form utama, untuk tempat pengisian identitas sampel barang uji dan pelanggan, Seperti gambar dibawah ini.

The image shows a Windows application window titled 'BARISTAND SURABAYA'. Inside the window, there is a header section with the logo of the 'Kementerian Perindustrian REPUBLIK INDONESIA' and the text 'SISTEM PEDATAAN BARANG MASUK BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA'. Below the header, there is a form titled 'Informasi Barang' with various input fields: 'Tanggal Masuk' (a date picker showing Monday, June 24, 2019), 'UID Chip', 'Nama Barang', 'Kode Barang', 'Nama Perusahaan', 'Nama Pengirim', 'Jenis Sample', 'Merk', 'Volume', 'Bentuk', 'Parameter Uji', 'Biaya Uji', and 'Metode Uji'. To the right of the form, there is a 'Serial Connection' section with a dropdown menu, a 'Connect' button, and buttons for 'Read ID', 'Clear', and 'Save'. At the bottom of the window, it says 'Copyright - Sistem Komputer 16'.

Gambar 4.32 Mendesain Form Utama

7. Menambahkan header `SqlConnection` pada program form utama.

```
using System.Data.SqlClient;
```

8. Mengklik dua kali pada form utama, selanjutnya menambahkan kode dibawah ini.

```
String reading;
bool isConnected = false;
String[] ports; SerialPort port;
```

Pada bagian diatas, variabel `reading` bertipe data string, digunakan untuk menyimpan data UID dari tag RFID, lalu variabel `isConnected` bertipe bool untuk memastikan serialport terhubung, selanjutnya variabel `ports` bertipe data string dengan array satu dimensi digunakan untuk menyimpan alamat port yang terhubung dengan komputer, yang akan ditampilkan pada combobox dalam

bentuk index, serta variabel port digunakan untuk interface program menggunakan serialport.

Pada bagian ini fungsi disableControls, digunakan untuk menonaktifkan komponen form seperti tombol, textbox dan groupbox.

```
private void disableControls()
{
    btnClear.Enabled = false;
    btnDB.Enabled = false;
    textBox1.Enabled = false;
    groupBox3.Enabled = false;
    button2.Enabled = false;
}
```

Pada bagian ini fungsi getAvailableComPorts, digunakan untuk mengumpulkan alamat port dari setiap perangkat yang terhubung dengan komputer menggunakan serialport.

```
void getAvailableComPorts()
{
    ports = SerialPort.GetPortNames();
}
```

Pada bagian ini fungsi utama, digunakan saat program pertama kali dijalankan, dimana fungsi disableControls dan getAvailableComPorts dipanggil, dimana sebelum Port COM terhubung, pengguna tidak dapat mengisi data, karena setiap komponen form tidak dihidupkan. Pada waktu yang bersamaan pula, nilai comboBox1 diisi oleh alamat serialport yang didapatkan dari fungsi getAvailableComPorts. comboBox1 ini dibuat untuk membantu pengguna menghubungkan port alat scanner loket.

```
public utama()
{
    InitializeComponent();
    disableControls();
    getAvailableComPorts();

    foreach (string port in ports)
    {
        comboBox1.Items.Add(port);
        Console.WriteLine(port);
    }
}
```

```

if (ports[0] != null)
{
    comboBox1.SelectedItem = ports[0];
}
}
}

```

Pada bagian ini fungsi `setText` digunakan untuk menampilkan data UID yang dikirimkan oleh mikrokontroler melalui serial pada `TextBox`.

```

delegate void SetTextCallback(string text);
private void SetText(string text)
{
    if (this.txtUid.InvokeRequired)
    {
        SetTextCallback d = new SetTextCallback(SetText);
        this.Invoke(d, new object[] { text });
    }
    else
    {
        this.txtUid.Text = text;
    }
}

```

Pada bagian ini fungsi `connectToArduino` digunakan untuk menghubungkan komunikasi program dengan alat scanner loket, yang menggunakan mikrokontroler Arduino Nano. Pada saat fungsi ini dipanggil, maka nilai variabel `isConnected` bernilai `true` lalu nilai variabel `selectedPort` bernilai port COM yang telah dipilih pengguna, selanjutnya variabel port berisi pengaturan port dengan alamat yang telah dipilih, *baudrate* sebesar 9600, lalu serialport dijalankan dan text pada `button1` menjadi “Disconnect”, serta fungsi `enableControls` dipanggil sehingga pengguna dapat mengisi data.

```

private void connectToArduino()
{
    isConnected = true;
    string selectedPort = comboBox1.GetItemText(comboBox1.SelectedItem);
    port = new SerialPort(selectedPort, 9600, Parity.None, 8, StopBits.One);
    port.Open();
    button1.Text = "Disconnect";
    enableControls();
}

```

Pada bagian ini fungsi `disconnectFromArduino` digunakan untuk mematikan komunikasi program dengan alat scanner loket. Pada saat fungsi ini dijalankan, maka variabel `isConnected` bernilai `false`, dan nilai variabel port dikosongkan, lalu text pada tombol dirubah menjadi "Connect", dan fungsi `disableControls` dipanggil kembali, agar pengguna tidak dapat mengisi data.

```
private void disconnectFromArduino()
{
    isConnected = false;
    port.Write(" ");
    port.Close();
    button1.Text = "Connect";
    disableControls();
}
```

9. Selanjutnya mengklik tombol connect, lalu mengisi kode berikut ini.

```
if (!isConnected)
{
    connectToArduino();
} else
{
    disconnectFromArduino();
}
```

Pada bagian ini digunakan untuk mengendalikan tombol agar dapat digunakan untuk menyambung dan memutuskan komunikasi program dengan alat scanner loket. Hal tersebut dilihat jika variabel `isConnected` bernilai `false` maka fungsi `connectToArduino` diaktifkan namun jika sebaliknya fungsi `disconnectFromArduino` yang dipanggil.

10. Selanjutnya mengklik tombol Read ID, dan masukan kode sebagai berikut.

```
if ((txtUId.Text == "") && (reading == ""))
{
    reading = port.ReadLine();
    SetText(reading);
} else
{
    txtUId.Text = "";
    reading = port.ReadLine();
    SetText(reading);
}
```

Tombol ini berfungsi untuk mengambil nilai UID dengan tipe data string dari mikrokontroler yang disalurkan melalui serialport dan dibaca oleh program tiap barisnya lalu disimpan pada variabel reading, dan ditampilkan dengan fungsi SetText. Kode tersebut akan dijalankan jika textbox UID dan variabel reading dalam keadaan kosong, jika tidak maka textbox UID dikosongkan.

11. Setelah itu mengklik tombol save dan memasukan kode dibawah ini.

```
if ((txtBentuk.Text ==
"")|| (txtBiaya.Text=="") || (txtJenis.Text=="") ||
(txtMerk.Text=="") || (txtMetode.Text=="") || (txtNamaBarang.Text=="") ||
(txtNamaPengirim.Text=="") || (txtNamaPerusahaan.Text=="") || (txtParameter.Text=="") || (txtUid.Text==" ") || (txtVolume.Text=="") ||
(dateTimePicker1.Text == ""))
{
    MessageBox.Show("Lengkapi semua data dengan benar.");
}
```

Tombol ini berfungsi untuk menyimpan data identitas sampel barang uji dan pelanggan pada database RFID_Scanner di tabel Master_Loket, serta menyimpan data Kode Barang pada tag RFID, melalui alat scanner loket. Berdasarkan logika kode diatas, jika tombol save ditekan dan ada salah satu textbox yang tidak diisi maka akan tampil peringatan “Lengkapi semua data dengan benar !”, Saat berlogika salah maka Kode Barang yang ada pada textbox1 ditulis (ditransmisikan) pada tag RFID, lalu dibuka komunikasi antara program dengan database Microsoft SQL Server 2017, seperti kode dibawah ini.

```
else {
    port.Write(textBox1.Text);
    string connetionString;
    SqlConnection cnn;
    connetionString = @"Data Source=LAPTOP-UAD8HS4T;Initial
Catalog=RFID_Scanner;Integrated Security=True";
    cnn = new SqlConnection(connetionString);
    cnn.Open(); SqlConnection command;
```

Saat berlogika salah maka dikirim pula perintah *Statement* INSERT INTO pada tabel Master_Loket sesuai dengan kolom-kolom diatas, yaitu kolom uid,

tanggal_masuk, nama_barang, kode_barang, nama_perusahaan, nama_pengirim, jenis_sample, merk, volume, bentuk, parameter_uji, biaya_uji, metode_uji dan lokasi dengan nilai sesuai dengan textbox yang ada pada form utama, sesuai kode dibawah ini.

```
String sql = "";
sql = "Insert into Master_Loket
(uid,tanggal_masuk,nama_barang,kode_barang,nama_perusahaan,nama_pengirim," +
"jenis_sample,merk,volume,bentuk,parameter_uji,biaya_uji,metode_uji, lokasi) " +
"values ('"+txtUid.Text+"','" +
dateTimePicker1.Text + "','" + txtNamaBarang.Text + "','" +
textBox1.Text + "','" + txtNamaPerusahaan.Text + "','" +
txtNamaPengirim.Text + "','" + txtJenis.Text + "','" + txtMerk.Text +
"','" + txtVolume.Text + "','" + txtBentuk.Text + "','" +
txtParameter.Text + "','" + txtBiaya.Text + "','" + txtMetode.Text +
"','loket')";
command = new SqlCommand(sql, cnn);
```

Saat *Statement* INSERT INTO berhasil ditransmisikan pada tabel Master_Loket maka akan keluar peringatan “Input data Berhasil !” dan seluruh isi *textbox* di form utama dihapus agar dapat diisi kembali oleh identitas barang uji dan pelanggan selanjutnya, jika gagal maka akan keluar peringatan “Gagal Input Data !”, seperti kode dibawah ini.

```
try{
if (command.ExecuteNonQuery()==1){
    MessageBox.Show("Input Data Berhasil !");
    this.txtUid.Text = "";
    this.txtVolume.Text = "";
    this.txtParameter.Text = "";
    this.txtNamaPerusahaan.Text = "";
    this.txtNamaPengirim.Text = "";
    this.txtNamaBarang.Text = "";
    this.txtMetode.Text = "";
    this.txtMerk.Text = "";
    this.txtJenis.Text = "";
    this.txtBiaya.Text = "";
    this.txtBentuk.Text = "";
    this.textBox1.Text = "";
    enableControls();
} else{
    MessageBox.Show("Gagal Input Data !");
}
}
```

Kode dibawah ini berfungsi untuk menampilkan peringatan apabila ada kesalahan dalam sistem, selanjutnya perintah terakhir ialah menutup komunikasi antara program Desktop Locket dengan Database Microsoft SQL Server 2017.

```
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
command.Dispose();
cnn.Close();
```

12. Ketika tombol silang merah ditekan, maka form login akan tampil dan form utama akan ditutup, adapun kodenya sebagai berikut.

```
login lg = new login();
lg.Show();
```

4.9.2 Pembuatan Program Mikrokontroler Scanner Locket

Adapun pembuatan program Mikrokontroler Scanner Locket adalah untuk membaca dan menulis tag RFID yang berfungsi sebagai identitas dari sampel barang uji dan alat ini akan terintegrasi langsung dengan program desktop locket. Adapun kode program ini, sebagai berikut.

1. *Library* yang digunakan program ini sebagai berikut.

```
#include <LiquidCrystal_I2C.h>
#include <EEPROM.h>
#include <SPI.h>
#include <MFRC522.h>
```

Adapun *library* tersebut secara urut dari atas merupakan *library* Modul i2c LCD, EEPROM, SPI BUS, dan Modul RFID MFRC-522.

2. Pendefinisian PIN Pada Arduino Nano.

```
#define SS_PIN 10
#define RST_PIN 9
```

3. Inisialisasi Modul i2c LCD

Inisialisasi modul tersebut berdasarkan kode i2c yang berbeda-beda, biasa 0x27 atau 0x3F dan ukuran LCD, yaitu 16 kali 2.

```
LiquidCrystal_I2C lcd(0x27 ,16, 2);
```

4. Inisialisasi PIN SS dan RESET pada MFRC522.

```
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::StatusCode status;
```

5. Inisialisasi variabel buffer dan size.

```
byte buffer[18];
byte size = sizeof(buffer);
```

6. Inisialisasi variabel pageAddr1.

0x06 merupakan halaman pertama untuk menulis data pada tag RFID.

Ultralight-C memiliki 16 halaman yang setiap halamannya. Contoh:

```
/*Pages 0x06 = |D1|D2|D3|D4|
Pages 0x07 = |D1|D2|D3|D4|
.
.
.
Pages 0x16 = |D1|D2|D3|D4| berkapasitas sebesar 4 bytes.
*/

uint8_t pageAddr = 0x06;
```

Secara khusus halaman 0 - 4 memiliki fungsi tersendiri, sehingga hanya halaman selanjutnya yang dapat digunakan.

7. Variabel pendukung program, diantaranya seperti berikut ini.

```
char data[16];
String uidTag="";
char c;
int tahan;
int g=7;
String myData;
char buff[10];
String gg;
```

8. Fungsi setup pada program.

Program dibawah ini mengatur komunikasi serial dengan *baudrate* 9600, dengan memulai LCD dan menyalakan lampu belakangnya, dan menampilkan tulisan “Baristand Surabaya”. Selanjutnya Memulai komunikasi SPI dan Insialisasi modul RFID MFRC522.

```
void setup() {
  Serial.begin(9600);
  lcd.begin();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print("Baristand");
  lcd.setCursor(4,1);
  lcd.print("Surabaya");
  SPI.begin();
  mfrc522.PCD_Init();
}
```

9. Fungsi loop baigan pertama, untuk menerima data dari program Desktop Locket berupa Kode Barang, yang disimpan pada variabel myData.

```
while (Serial.available()>0){
  delay(10);
  c = Serial.read();
  myData+=c;
}
```

10. Memastikan Module mendeteksi tag RFID dan Memilih salah satu tag RFID.

```
if ( ! mfrc522.PICC_IsNewCardPresent())
  return;

if ( ! mfrc522.PICC_ReadCardSerial())
  return;
```

11. Membaca Nilai UID dari tag RFID.

```
uidTag="";
for (byte i = 0; i < mfrc522.uid.size; i++) {
  char teks[3];
  sprintf(teks,"%02X", mfrc522.uid.uidByte[i]);
  uidTag+=teks;
}
Serial.println(uidTag);
```

12. Menampilkan Tulisan Baristand dan Nilai UID pada LCD Karakter 16 x 2.

```

if(tahan==0)
{
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print("Baristand");
  lcd.setCursor(4,1);
  lcd.print("Surabaya");
  delay(300);
}
if(uidTag!="")
{tahan=1;}
else{tahan=0;}

if(tahan==1)
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("UID:");
  lcd.setCursor(6,0);
  lcd.print(uidTag);
  delay(1000);
}

```

13. Menyimpan data Kode Barang pada EEPROM.

```

if (myData.length()>0){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("KODE :");
  lcd.setCursor(7,0);
  lcd.print(myData);
  lcd.setCursor(0,1);
  lcd.print("DATA TERSIMPAN");
  delay(2000);
  myData.toCharArray(buff, 10);
  gg=myData;
  EEPROM.put(0,buff);
  myData="";
}

```

14. Mengambil nilai EEPROM dan disimpan pada variabel hasil.

```
String hasil = EEPROM.get(0,buff);
```

15. Mentransmisikan data dari Scanner Locket ke tag RFID.

```

for(int i=0;i<16;i++)
{data[i]=hasil[i];}
memcpy(buffer,data,5);

```

16. Memastikan data yang ditransmisikan masuk pada tag RFID. Pada kode tersebut, jika status MFRC522 tidak OK, maka pengguna harus mendekatkan tag RFID, agar data dapat ditransmisikan dengan baik pada tag RFID.

```
for (int i=0; i < 4; i++) {
  Status=(MFRC522::StatusCode)
  mfrc522.MIFARE_Ultralight_Write(pageAddr+i, &buffer[i*4], 4);

  if (status != MFRC522::STATUS_OK) {
    Serial.println(F("DEKATKAN CARD : "));
    lcd.clear();
    lcd.setCursor(4,0);
    lcd.print("Baristand");
    lcd.setCursor(4,1);
    lcd.print("Surabaya");
    delay(300);
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;}
}
```

17. Selanjutnya jika, nilai variabel buff tidak ada nilai, maka pengguna harus memasukkan data, dan peringatan ditampilkan pada LCD karakter.

```
else if (buff [0]=="")
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("UID:");
  lcd.setCursor(6,0);
  lcd.print(uidTag);
  lcd.setCursor(0,1);
  lcd.print("Masukan Data");
}
```

18. Jika variabel buff berisi, maka nilai UID dan Kode Barang ditampilkan pada layar LCD Karakter.

```
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("UID:");
  lcd.setCursor(6,0);
  lcd.print(uidTag);
  lcd.setCursor(0,1);
  lcd.print("KODE :");
  for (byte i = 0; i< 5;i++)
  {
    lcd.write(buffer[i]);
    g++;
  }
  delay (1000);
}}
```

19. Melakukan jeda pada MFRC-522 setelah melakukan proses baca dan tulis data pada tag RFID.

```
mfr522.PICC_HaltA();
```

4.9.3 Program Mikrokontroler Scanner Laboratorium

Adapun pembuatan program Mikrokontroler Scanner Laboratorium adalah untuk membaca dan menulis tag RFID yang berada pada antrian, proses dan telah selesai diuji dan alat ini akan terintegrasi langsung dengan Database SQL Server 2017, dengan menggunakan protokol komunikasi HTTP yang menggunakan kode PHP pada server. Adapun kode program mikrokontroler ini, sebagai berikut.

1. Library yang digunakan program ini sebagai berikut.

```
#include <SPI.h>
#include <ESP8266WiFi.h>
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>
```

Adapun library tersebut secara urut dari atas merupakan library Modul SPI BUS, Modul ESP8266, Modul RFID MFRC-522, dan i2c LCD.

2. Pendefinisian PIN RFID-RC522 pada NodeMCU V3.

```
#define SS_PIN D4
#define RST_PIN D3
```

3. Inisialisasi Modul i2c LCD.

Inisialisasi modul tersebut berdasarkan kode i2c yang berbeda-beda, biasa 0x27 atau 0x3F dan ukuran LCD, yaitu 16 kali 2.

```
LiquidCrystal_I2C lcd(0x27 ,16, 2);
```

4. Inisialisasi PIN SS dan RESET pada MFRC522.

Kode tersebut untuk inisialisasi pin ss dan reset sesuai dengan pendefinisian diatas dan untuk mendapatkan status dari MFRC522.

```
MFRC522 mfrc522(SS_PIN, RST_PIN);
MFRC522::StatusCode status;
```

5. Inisialisasi variabel network untuk Jaringan Lokal.

Variabel ini dibuat untuk, menghubungkan NodeMCU V3 dengan jaringan lokal yang telah dibuat.

```
const char* ssid = "UIDServer";
const char* password = "113eeac34f";
const char* host = "192.168.137.1";
```

6. Inisialisasi variabel alamat server

Variabel ini dibuat untuk, menyimpan alamat file PHP yang tersimpan dalam server.

```
String path = "sql_server_connect/koneksi.php?";
String url = "GET /sql_server_connect/koneksi.php?";
```

7. Inisialisasi variabel buffer dan size.

```
byte buffer[18];
byte size = sizeof(buffer);
```

8. Inisialisasi variabel pageAddr1.

0x06 merupakan halaman pertama untuk menulis data pada tag RFID.

Ultralight-C memiliki 16 halaman yang setiap halamannya. Contoh:

```
/*Pages 0x06 = |D1|D2|D3|D4|
Pages 0x07 = |D1|D2|D3|D4|
.
.
.
.
Pages 0x16 = |D1|D2|D3|D4| berkapasitas sebesar 4 bytes.
```

```
*/
```

```
uint8_t pageAddr = 0x06;
```

Secara khusus halaman 0 - 4 memiliki fungsi tersendiri, sehingga hanya halaman selanjutnya yang dapat digunakan.

9. Inisialisasi Pin Tombol.

```
int start = 16;
int w = 15;
int Send = A0;
```

Adapun secara terurut, Tombol power pada pin 16, Read/Write pada pin 15 dan Send pada pin A0.

10. Variabel pendukung program, diantaranya seperti berikut ini.

```
int power= 0;
int baca = 0;
int c = 0;
char x;
String data_UID = "";
String kode_Barang = "";
String Lokasi = "";
String location="FISIKA ";
String uidTag="";
String kode;
char data[16];
```

11. Fungsi Setup pada Program Mikrokontroler Scanner Laboratorium.

```
void setup()
{
  lcd.begin();
  lcd.clear();
  lcd.backlight();
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
    lcd.setCursor(0,0);
    lcd.print("Mencari Koneksi!");
  }
  Serial.println("");
  Serial.println("WiFi Terhubung !");
  Serial.println("IP Address:" + WiFi.localIP());
  SPI.begin();
```

```

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("WiFi Terhubung !");
    delay(1000);
    mfrc522.PCD_Init(); // Init MFRC522
    Serial.println("RFID reading UID");
    pinMode(start, INPUT);
    pinMode(w, INPUT);
    pinMode(Send, INPUT);
}

```

Fungsi ini memuat, pengaturan LCD dimulai dan pengaktifan lampu dibelakang layar, *baudrate* 115200, pengaktifan SPI, serta menghubungkan SSID dengan Katakunci pada jaringan lokal. Selanjutnya jika status WiFi tidak terhubung dengan jaringan lokal, maka LCD akan menampilkan pesan “Mencari Koneksi !”, program tersebut diulang hingga Status WiFi terhubung dengan Jaringan Lokal. Jika telah terhubung, maka ditampilkan pesan “WiFi Terhubung” pada LCD dan pada menu serial monitor, beserta IP Address yang didapatkan. Selanjutnya inisialisasi modul MFRC-522 dan dilanjutkan dengan pengaturan mode pin pada fungsi setup. Pada bagian ini, tombol power, read/write, dan send diatur sebagai mode INPUT.

12. Fungsi loop pada scanner laboratorium.

```

power = digitalRead(start);
c = analogRead(Send);

```

Pada fungsi ini variabel power menerima nilai dari masukan digital dari pin start, lalu variabel c menerima nilai dari masukan analog dari pin send. Pada fungsi ini, terdapat beberapa kondisi, yaitu:

A.) Kondisi Tombol Power dan Baca/Tulis tidak ditekan, maka LCD hanya menampilkan Tulisan “Baristand Surabaya”.

```

if(power!=HIGH && baca !=HIGH )
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("Baristand"));
    lcd.setCursor(8,1);
    lcd.print(F("Surabaya"));
    delay (1000);
}

```

B.) Kondisi Tombol Power ditekan dan Baca/Tulis tidak ditekan, maka modul MFRC-522 membaca UID dan LCD menampilkan Kode Barang beserta lokasi.

```

if(power==HIGH)
{
    if ( ! mfrc522.PICC_IsNewCardPresent())
        return;

    if ( ! mfrc522.PICC_ReadCardSerial())
        return;

    //read UID
    uidTag="";
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
        char teks[3];
        sprintf(teks,"%02X", mfrc522.uid.uidByte[i]);
        uidTag+=teks;
    }

    baca = digitalRead(w);
    if(baca!=HIGH&& power==HIGH)
    {
        //read
        status = (MFRC522::StatusCode) mfrc522.MIFARE_Read(pageAddr,
        buffer, &size);
        if (status != MFRC522::STATUS_OK)
        {
            Serial.print(F("MIFARE_Read() failed: "));
            Serial.print(mfrc522.GetStatusCodeName(status));
            return;
        }

        Serial.print("UID:");
        Serial.println(uidTag);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(F("KodeBaca:"));
        kode=(char*)buffer;
        lcd.println(kode);
        lcd.setCursor(0,1);
        lcd.print("Lokasi:");
    }
}

```

```
        lcd.println(location);
    }

```

C.) Kondisi Tombol Power dan Baca/Tulis ditekan, maka modul MFRC-522 menulis status kode barang dari T menjadi P, dengan mengubah isi variabel data pada array ke 0, lalu LCD menampilkan kode barang yang telah dirubah dan lokasi dari barang tersebut.

```

if(baca==HIGH && power==HIGH)
{
    //write
    data[0]='P';
    for(int i=1;i<16;i++)
    {data[i]=kode[i];}
    memcpy(buffer,data,5);
    for (int i=0; i < 4; i++) {
        status=(MFRC522::StatusCode)mfr522.MIFARE_Ultralight_Write
        (pageAddr+i, &buffer[i*4], 4);

        if (status != MFRC522::STATUS_OK) {
            Serial.println(F("MIFARE_Read() failed: "));
            Serial.print(mfr522.GetStatusCodeName(status));
            lcd.setCursor(0,0);
            lcd.print(F("KodeTulis:"));
            kode=(char*)buffer;
            lcd.println(kode);
            lcd.setCursor(0,1);
            lcd.print("Belum Ditulis !");
            return;}
        }
        Serial.print("UID:");
        Serial.println(uidTag);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print(F("KodeTulis:"));
        kode=(char*)buffer;
        lcd.println(kode);
        lcd.setCursor(0,1);
        lcd.print("Lokasi:");
        lcd.println(location);
    }
}

```

D.) Kondisi Tombol Send ditekan, maka LCD akan menampilkan pesan “Kirim Data” dan client menghubungkan ke port 80, jika berhasil terhubung ke port tersebut, maka data UID, Kode Barang dan Lokasi dikirim dengan menggunakan protokol HTTP. Selanjutnya, jika berhasil koneksi diputus.

```

if( c!=1024 )
{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Kirim Data");
    Serial.print("Terhubung ke");
    Serial.println(host);

    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort))
    {
        Serial.println("Koneksitivitas Gagal !");
        lcd.setCursor(0,1);
        lcd.println("Gagal !");
        return;
    }

    data_UID = uidTag ;
    kode_Barang = kode;
    Lokasi = location;

    delay(1000);
    // Kirim Data-data ke Server
    client.print( url + "uid=" + data_UID + "&" + "kode_barang="
+ kode_Barang + "&" + "lokasi="+ Lokasi );
    client.print(" HTTP/1.1\r\nHost: host\r\nConnection:keep-
alive\r\nAccept: */*\r\n\r\n");
    Serial.println(data_UID);
    Serial.println(kode_Barang);
    Serial.println(Lokasi);

    while (client. available())
    {
        String line = client.readStringUntil('\r');
        Serial.print(line);
    }
    Serial.println();
    lcd.setCursor(9,1);
    lcd.println("Selesai :)");
    Serial.println("Koneksitivitas diputus");
    delay(10);
}
    mfrc522.PICC_HaltA();
}

```

4.9.4 Pembuatan Program Mikrokontroler Scanner Reset

Adapun pembuatan program Mikrokontroler Scanner Reset adalah untuk membaca dan menghapus data pada tag RFID yang terintegrasi langsung dengan database SQL Server 2017, dengan menggunakan protokol komunikasi HTTP yang

menggunakan kode PHP pada server. Adapun kode program mikrokontroler ini sama dengan Program Mikrokontroler Scanner Laboratorium, hanya saja berbeda pada variabel alamat server, yaitu sebagai berikut.

```
String path = "sql_server_connect/reset.php?";
String url = "GET /sql_server_connect/reset.php?";
```

Serta perbedaan pada logika pengaktifan mode baca dan tulis, yaitu jika tombol reset tidak ditekan maka, pengguna dapat mengecek kode barang pada tag RFID dan jika sebaliknya maka pengguna dapat menghapus data kode barang pada tag RFID dan pada database tabel Master_Loket, agar tag RFID dapat dipakai kembali.

4.9.5 Program PHP Jembatan Mikrokontroler dan Database

Adapun program Transmisi Data Mikrokontroler ke Database, menggunakan program PHP, yang dibagi menjadi dua program, yaitu program koneksi.php dan reset.php.

1. Program koneksi.php

Dengan menggunakan metode GET variabel bernama uid, kode_barang dan lokasi menerima data yang ditransmisikan oleh Protokol HTTP dan menyimpannya pada database.

```
$uid = $_GET['uid'];
$kode_barang = $_GET['kode_barang'];
$lokasi = $_GET['lokasi'];
```

Adapun untuk membangun koneksi dengan database, diperlukan informasi nama server dan database yang disimpan masing-masing pada variabel serverName dan connectionInfo. Setelah informasi didapatkan, maka fungsi

sqlsrv_connect dipanggil untuk menghubungkan database dengan informasi yang telah didapatkan sebelumnya dan disimpan pada variabel conn.

```
$serverName = "LAPTOP-UAD8HS4T";
$connectionInfo = array( "Database"=>"RFID_Scanner");
$conn = sqlsrv_connect( $serverName, $connectionInfo );

if( $conn === false ) {
    die( print_r( sqlsrv_errors(), true));
}
else
{
    echo "Koneksi Berhasil !". "<br>";
}
```

Setelah database dihubungi dan hasil feedback diterima oleh variabel conn, maka untuk memastikan apakah program PHP sudah terhubung dengan database, dibuatlah kondisi ini, dimana jika variabel conn berisi nilai false maka akan tampil pesan error pada halaman web, jika sebaliknya maka akan tampil pesan koneksi berhasil, yang menandakan bahwa database telah terhubung dengan program PHP.

Kode ini dibuat untuk mengolah tabel Antrian_LFisika. Pada kode tersebut variabel sqlAntrian_LFisika menyimpan query dengan statement SELECT dari tabel Antrian_LFisika dengan kondisi nilai UID, lalu variabel paramsAntrian_LFisika berisikan fungsi array, selanjutnya variabel optionsAntrian_LFisika berisikan fungsi array yang akan menyimpan hasil hitung baris pada tabel tersebut. Lalu variabel stmtAntrian_LFisika berisikan fungsi sqlsrv_query yang membutuhkan parameter status koneksi database, pendefinisian query, parameter array dan opsi. Selanjutnya untuk menghitung jumlah baris pada tabel Antrian_LFisika, maka dipanggil fungsi sqlsrv_num_rows dengan parameter variabel stmtAntrian_LFisika yang dimana

nilai jumlah baris pada tabel tersebut disimpan oleh variabel `row_count_Antrian_LFisika`.

```
$sqlAntrian_LFisika = "SELECT * FROM Antrian_LFisika WHERE uid = '$uid'";
$paramsAntrian_LFisika = array();
$optionsAntrian_LFisika = array("Scrollable" =>
SQLSRV_CURSOR_KEYSET);
$stmtAntrian_LFisika = sqlsrv_query( $conn, $sqlAntrian_LFisika ,
$paramsAntrian_LFisika, $optionsAntrian_LFisika );

$row_count_Antrian_LFisika = sqlsrv_num_rows( $stmtAntrian_LFisika
);
```

Bagian dibawah ini sama dengan kode sebelumnya, hanya saja perbedaan pada tabel, yaitu pada tabel `Proses_LFisika` dan `Selesai_LFisika`.

```
$sqlProses_LFisika = "SELECT * FROM Proses_LFisika WHERE uid = '$uid'";
$paramsProses_LFisika = array();
$optionsProses_LFisika = array("Scrollable" =>
SQLSRV_CURSOR_KEYSET);
$stmtProses_LFisika = sqlsrv_query( $conn, $sqlProses_LFisika ,
$paramsProses_LFisika, $optionsProses_LFisika );

$row_count_Proses_LFisika = sqlsrv_num_rows( $stmtProses_LFisika );

$sqlSelesai_LFisika = "SELECT * FROM Selesai_LFisika WHERE uid = '$uid'";
$paramsSelesai_LFisika = array();
$optionsSelesai_LFisika = array("Scrollable" =>
SQLSRV_CURSOR_KEYSET);
$stmtSelesai_LFisika = sqlsrv_query( $conn, $sqlSelesai_LFisika ,
$paramsSelesai_LFisika, $optionsSelesai_LFisika );

$row_count_Selesai_LFisika = sqlsrv_num_rows( $stmtSelesai_LFisika
);
```

Untuk memasukan data yang didapatkan dari mikrokontroler pada tabel `Antrian_LFisika` yang menandakan bahwa barang uji pada tahapan antrian, kondisi yang harus dipenuhi ialah variabel `row_count_Antrian_LFisika` harus bernilai kosong, jika sebaliknya maka data disalurkan pada tabel `Proses_LFisika` dan data yang masuk pada tabel `Proses_LFisika` yang memiliki UID sama pada tabel `Antrian_LFisika` dihapus dengan perintah `DELETE`, untuk

menandakan bahwa barang telah berada pada tahapan proses dan selanjutnya untuk tahapan Selesai.

```

if ($row_count_Antrian_LFisika == 0)
{
    $antri = "INSERT INTO Antrian_LFisika (uid, kode_barang,
lokasi) VALUES('$uid', '$kode_barang', '$lokasi')";
    $result = sqlsrv_query($conn,$antri);
}
else if ($row_count_Antrian_LFisika != 0)
{
    $proses = "INSERT INTO Proses_LFisika (uid, kode_barang,
lokasi) VALUES('$uid', '$kode_barang', '$lokasi')";
    $result = sqlsrv_query($conn,$proses);
    $hapus = "DELETE FROM Antrian_LFisika WHERE
Antrian_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn,$hapus);
}

if (($row_count_Antrian_LFisika == 0) && ($row_count_Proses_LFisika
!= 0))
{
    $selesai = "INSERT INTO Selesai_LFisika (uid, kode_barang,
lokasi) VALUES('$uid', '$kode_barang', '$lokasi')";
    $result = sqlsrv_query($conn, $selesai);
    $hapuspro = "DELETE FROM Proses_LFisika WHERE
Proses_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn, $hapuspro);
    $hapusstri = "DELETE FROM Antrian_LFisika WHERE
Antrian_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn,$hapusstri);
}

```

2. Program reset.php

Dengan menggunakan metode GET variabel bernama uid, kode_barang dan lokasi menerima data yang ditransmisikan oleh Protokol HTTP dan menyimpannya pada database.

```

$uid = $_GET['uid'];
$kode_barang = $_GET['kode_barang'];
$lokasi = $_GET['lokasi'];

```

Adapun untuk membangun koneksi dengan database, diperlukan informasi nama server dan database yang disimpan masing-masing pada variabel serverName dan connectionInfo. Setelah informasi didapatkan, maka fungsi

sqlsrv_connect dipanggil untuk menghubungkan database dengan informasi yang telah didapatkan sebelumnya dan disimpan pada variabel conn.

```
$serverName = "LAPTOP-UAD8HS4T";
$connectionInfo = array( "Database"=>"RFID_Scanner");
$conn = sqlsrv_connect( $serverName, $connectionInfo );

if( $conn === false ) {
    die( print_r( sqlsrv_errors(), true));
}
else
{
    echo "Koneksi Berhasil !". "<br>";
}
```

Setelah database dihubungi dan hasil feedback diterima oleh variabel conn, maka untuk memastikan apakah program PHP sudah terhubung dengan database, dibuatlah kondisi ini, dimana jika variabel conn berisi nilai false maka akan tampil pesan error pada halaman web, jika sebaliknya maka akan tampil pesan koneksi berhasil, yang menandakan bahwa database telah terhubung dengan program PHP.

Program dibawah ini sama dengan program yang ada pada koneksi.php dengan tujuan untuk menghitung jumlah baris pada tabel Antrian_LFisika dan Selesai_LFisika.

```
$sqlAntrian_LFisika = "SELECT * FROM Antrian_LFisika WHERE uid = '$uid'";
$paramsAntrian_LFisika = array();
$optionsAntrian_LFisika = array("Scrollable" => SQLSRV_CURSOR_KEYSET);
$stmtAntrian_LFisika = sqlsrv_query( $conn, $sqlAntrian_LFisika , $paramsAntrian_LFisika, $optionsAntrian_LFisika );

$row_count_Antrian_LFisika = sqlsrv_num_rows( $stmtAntrian_LFisika );

$sqlSelesai_LFisika = "SELECT * FROM Selesai_LFisika WHERE uid = '$uid'";
$paramsSelesai_LFisika = array();
$optionsSelesai_LFisika = array("Scrollable" => SQLSRV_CURSOR_KEYSET);
$stmtSelesai_LFisika = sqlsrv_query( $conn, $sqlSelesai_LFisika , $paramsSelesai_LFisika, $optionsSelesai_LFisika );
```

```
$row_count_Selesai_LFisika = sqlsrv_num_rows( $stmtSelesai_LFisika
);
```

Adapun program dibawah ini dibuat untuk reset data, maka perintah yang diberikan ialah perintah DELETE data pada tabel Selesai_LFisika dan Master_Loket dengan ketentuan UID yang sama. Hal ini menyatakan bahwa bahwa barang uji telah selesai melewati semua tahapan sehingga tag RFID dapat digunakan kembali oleh sampel barang uji lainnya.

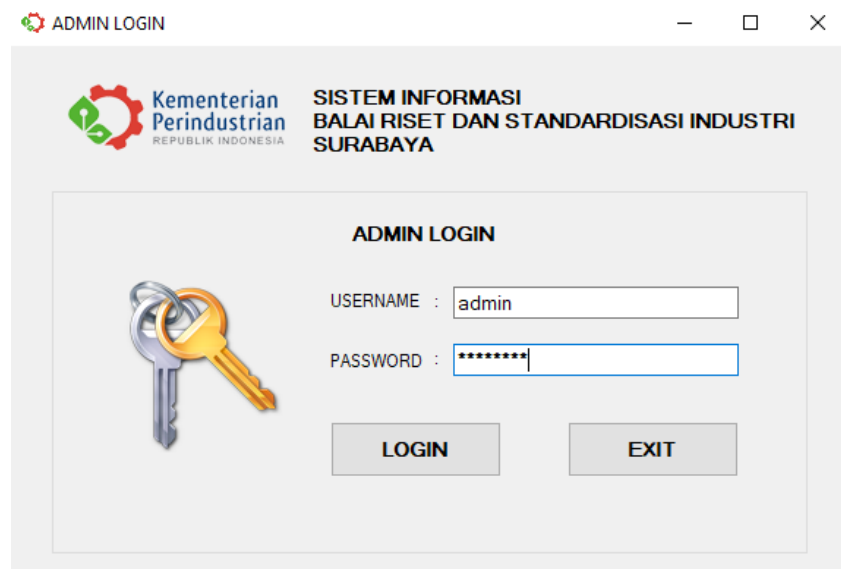
```
if(($row_count_Antrian_LFisika == 0) && ($row_count_Selesai_LFisika
!= 0))
{
    $hapusel = "DELETE FROM Selesai_LFisika WHERE
Selesai_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn, $hapusel);
    $hapuslok = "DELETE FROM Master_Loket WHERE
Master_Loket.uid='$uid'";
    $result = sqlsrv_query($conn,$hapuslok);}
```

4.10 Uji Coba Alat dan Program

Adapun uji coba alat terdiri atas alat Scanner Locket, Laboratorium dan Reset, serta Program Desktop Locket.

4.10.1 Uji Coba Scanner Locket dan Program Desktop Locket

Adapun untuk menggunakan program ini, pertama pengguna harus login dahulu dengan mengisi Username dan Password lalu mengklik Login seperti gambar dibawah ini.



ADMIN LOGIN

Kementerian Perindustrian
REPUBLIK INDONESIA

SISTEM INFORMASI
BALAI RISET DAN STANDARDISASI INDUSTRI
SURABAYA

ADMIN LOGIN

USERNAME : admin

PASSWORD : *****

LOGIN EXIT

Gambar 4.33 Tampilan Form Login

Selanjutnya, setelah login, maka tampil form utama sebagai formulir untuk mengisi identitas sampel barang uji dan pelanggan, namun sebelum mengisi formulir, pengguna perlu menghubungkan program dengan Alat Scanner Locket dengan cara memilih Port dan mengklik Connect. Pada percobaan ini Port COM13 yang digunakan.

Gambar 4.34 Tampilan Form Utama

Selanjutnya, setelah port terhubung, maka pengguna harus meletakkan Tag RFID pada alat Scanner Loket seperti gambar dibawah.



Gambar 4.35 Alat Scanner Loket

Setelah itu pengguna dapat mengisi Identitas Sampel Barang Uji pada program, seperti gambar 4.35.

Gambar 4.36 Tampilan Program Desktop Loket

Setelah mengisi identitas sampel barang uji dan pelanggan, maka mengklik tombol save dan akan tampil pesan “Input Data Berhasil !”. Pesan tersebut menyatakan bahwa data berhasil diinputkan pada database dan tag RFID.

Gambar 4.37 Tampilan Program Desktop Data Berhasil Dimasukkan

Adapun isi database pada tabel Master_Loket telah berhasil dimasukkan dan hasilnya dapat dilihat seperti gambar dibawah ini.

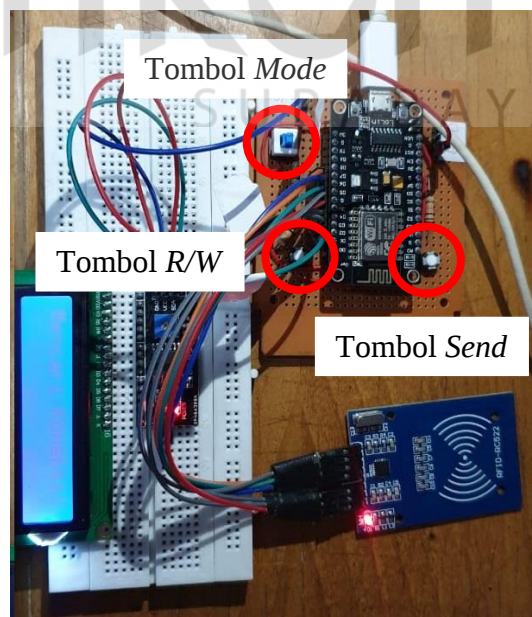
LAPTOP-UAD8HS4T....dbo.Master_Loket X								
	uid	tanggal_masuk	nama_barang	kode_barang	nama_perusah...	nama_pengirim	jenis_sample	merk
▶	04F0CE120A3C80	Monday, June 2...	Air Mineral	T0007	Sejuk Sejahterah	Ilham Rizaldy	Cairan	Sejuklah
	04F8CE120A3C80	Monday, June 2...	Sabun Mandi	T0003	Segar Alami Gr...	Titania	Sabun Cair	Hmmmm
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 4.38 Database Tabel Master_Loket

4.10.2 Uji Coba Scanner Laboratorium

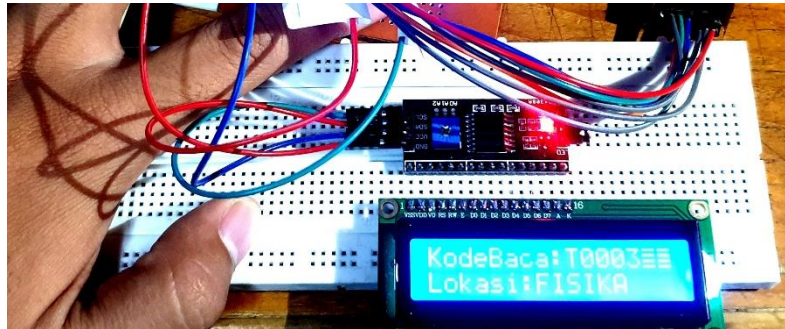
Adapun komponen scanner laboratorium terdiri atas slider switch power untuk mengaktifkan alat, lalu switch mode untuk pilihan alat sebagai penulis atau pembaca tag RFID, tombol scan yang digunakan untuk memindai tag RFID dan tombol send yang digunakan untuk mengirim data RFID ke database.

Pada percobaan ini komponen belum dikemas, bentuknya seperti gambar dibawah ini.



Gambar 4.39 Gambar Rangkaian Scanner Laboratorium

Adapun untuk melihat kode pada tag RFID, pengguna harus mengklik tombol R/W dengan tombol mode tidak ditekan (mode Read). Hasil seperti pada gambar dibawah ini.



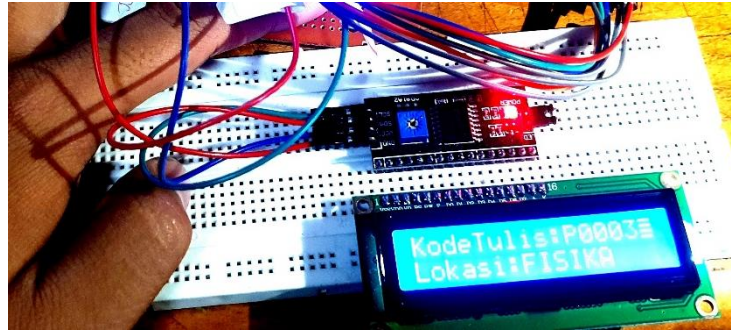
Gambar 4.40 Hasil Kode Baca

Selanjutnya mengklik tombol send untuk mengirim data ke database, kiriman pertama ini, menandakan bahwa tag RFID telah berada pada antrian. Hasil dapat dilihat pada gambar database dibawah ini.

LAPTOP-UAD8HS4T.R...o.Antrian_LFisika			LAPTOP-UAD8HS4T....dbo.Master_Loket
uid	kode_barang	lokasi	
04E7CF120A3C80	T0003	FISIKA	
04F8CE120A3C80	P0005	FISIKA	
NULL	NULL	NULL	

Gambar 4.41 Database Tabel Antrian_LFisika

Selanjutnya, mengklik tombol mode untuk mengubah mode menjadi Write (Tulis), agar kode barang berubah status menjadi P, lalu mengklik tombol R/W dan mengklik tombol send untuk mengirim data ke database, yang menandakan bahwa tag RFID telah berada pada tahapan proses.



Gambar 4.42 Kode Write pada Tag RFID

LAPTOP-UAD8HS4T.R...o.Antrian_LFisika		LAPTOP-UAD8H	
uid	kode_barang	lokasi	
04E7CF120A3C80	P0003	FISIKA	
NULL	NULL	NULL	

Gambar 4.43 Hasil Database Tabel Proses_LFisika

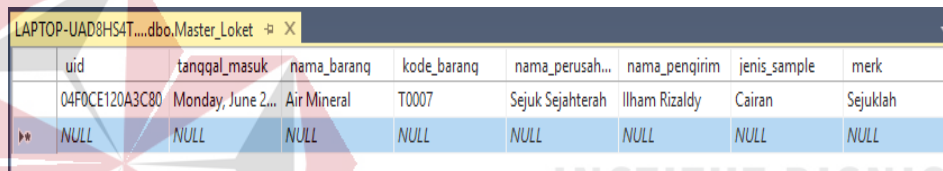
Selanjutnya, untuk menandakan bahwa barang uji telah selesai diuji, maka pengguna harus kembali cek kode barang pada tag RFID seperti cara diatas dan mengirimkan data tersebut pada database dengan tombol send. Adapun hasil database seperti gambar dibawah ini.

LAPTOP-UAD8HS4T.R...o.Selesai_LFisika			
uid	kode_barang	lokasi	
04E7CF120A3C80	P0003	FISIKA	
NULL	NULL	NULL	

Gambar 4.44 Hasil Database Selesai_LFisika

4.10.3 Uji Coba Scanner Reset

Adapun untuk melakukan reset pada tag RFID dan database pada tabel Master_Loket, pengguna harus melakukan reset dengan menggunakan alat scanner reset, dengan cara meletakkan tag RFID pada alat dan memastikan bahwa kode barang yang ingin diatur ulang benar, selanjutnya menekan tombol reset untuk mengatur ulang tag RFID agar kode barang terhapus dan mengirimkan perintah hapus data pada database tabel Master_Loket sesuai dengan UID pada tag RFID. Hal tersebut menandakan bahwa data telah diatur ulang agar tag RFID dapat digunakan kembali.

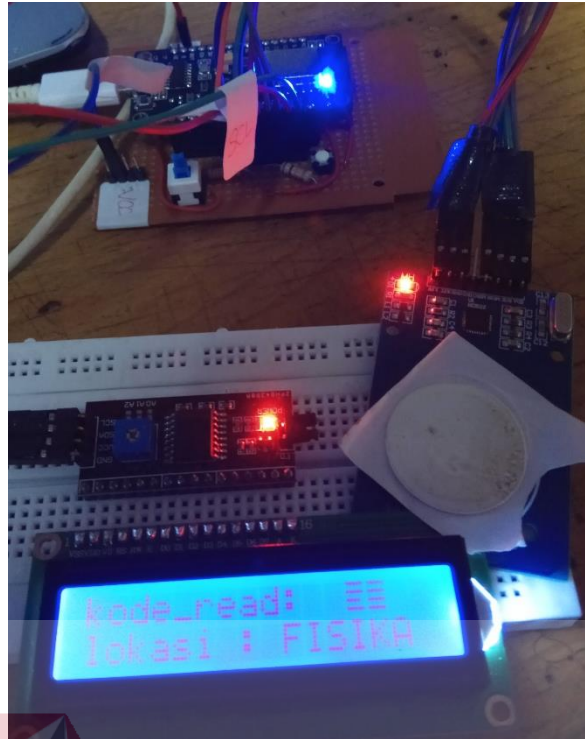


uid	tanggal_masuk	nama_barang	kode_barang	nama_perusah...	nama_pengirim	jenis_sample	merk
04F0CE120A3C80	Monday, June 2...	Air Mineral	T0007	Sejuk Sejahterah	Ilham Rizaldy	Cairan	Sejuklah
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 4.45 Mengecek Kode Barang T0003 Tidak Ada Pada Master_Loket



Gambar 4.46 Mengecek Tag RFID dengan Scanner Reset



Gambar 4.47 Mengecek RFID Setelah Diatur Ulang

4.11 Hasil Alat



Gambar 4.48 Alat Scanner Loket Dengan Stiker Tag RFID



Gambar 4.49 Alat Scanner Laboratorium



Gambar 4.50 Alat Scanner Reset

BAB V

PENUTUP

Pada bab lima ini akan dibahas mengenai kesimpulan dan saran dari Rancang Bangun Sistem Pelacak Barang Uji Laboratorium Berbasis RFID (*Radio Frequency Identification*) Menggunakan Database SQL Server 2017 pada Balai Riset dan Standardisasi Industri Surabaya.

5.1 Kesimpulan

Kesimpulan pada pembuatan Sistem Pelacak Barang Uji Laboratorium Berbasis RFID (*Radio Frequency Identification*) Menggunakan Database SQL Server 2017 adalah sistem tersebut menggunakan tiga buah alat, yaitu alat scanner loket, laboratorium dan reset, dimana secara urut terintegrasi dengan Loket dan Jaringan Lokal untuk kebutuhan transmisi data dari setiap scanner kepada database (sistem basis data). Sistem tersebut telah menjawab kebutuhan dan permasalahan yang ada, yaitu sebagai berikut:

1. Sistem rekam jejak dari setiap tahapan sampel barang uji telah dibuat dan telah terintegrasi dengan sistem basis data, sehingga dapat menunjang administrasi dan proses bisnis dari setiap sampel barang uji yang masuk.
2. Sistem telah sepenuhnya terintegrasi dengan komputer, sehingga dapat meminimalisir terjadinya kesalahan manusia.

5.2 Saran

Adapun saran dari penulis adalah sistem dapat dikembangkan lebih baik lagi, dari segi sistem basis data, UI program dekstop, protokol transmisi data dan keamanan jaringan saat transmisi data melalui jaringan nirkabel, serta rangkaian elektronika yang dibuat, serta *bug-bug* lain pada program yang mungkin tidak diketahui.



DAFTAR PUSTAKA

- Dosen Pendidikan. (2019, Februari 01). *Pengertian, Komponen Dan Fungsi XAMPP Lengkap Dengan Penjelasannya*. Retrieved Juni 22, 2019, from Dosen Pendidikan: <https://www.dosenpendidikan.com/pengertian-komponen-dan-fungsi-xampp-lengkap-dengan-penjasannya/>
- Kadir, A. (2018). *Arduino & Sensor Tuntunan Praktis Mempelajari Penggunaan Sensor untuk Aneka Proyek Elektronika Berbasis Arduino*. Yogyakarta: Penerbit Andi.
- Komputronika. (2019, Januari 20). *Apa NodeMCU Itu?* Retrieved Juni 18, 2019, from <https://www.komputronika.com/arduino/apa-nodemcu-itu/>
- NXP Semiconductors. (2011). *NTAG203 NFC Forum Type 2 Tag compliant IC with 144 bytes user memory*. Retrieved Juni 17, 2019, from <http://www.orangetags.com/wp-content/downloads/datasheet/NXP/NTAG203.pdf>
- Raharjo, B. (2015). *Mudah Belajar C# (Pemrograman C# dan Visual C#)*. Bandung: Informatika Bandung.
- Wicaksono, M. F., & Hidayat. (2017). *Mudah Belajar Mikrokontroler Arduino Disertai 23 Proyek, termasuk Proyek Ethernet dan Wireless Client Server*. Bandung: Informatika Bandung.
- Wikipedia. (2016, Oktober 20). *Microsoft SQL Server*. Retrieved Juni 20, 2019, from https://id.wikipedia.org/wiki/Microsoft_SQL_Server