



IMPLEMENTASI RFID (*RADIO FREQUENCY IDENTIFICATION*) UNTUK TRACKING BARANG PADA LOKET DAN LABORATORIUM KIMIA DI BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA



Oleh:

Fahmi Andriansyah

16410200031

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA
2019**

LAPORAN KERJA PRAKTIK

IMPLEMENTASI RFID (*RADIO FREQUENCY IDENTIFICATION*) UNTUK *TRACKING* BARANG PADA LOKET DAN LABORATORIUM KIMIA DI BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA

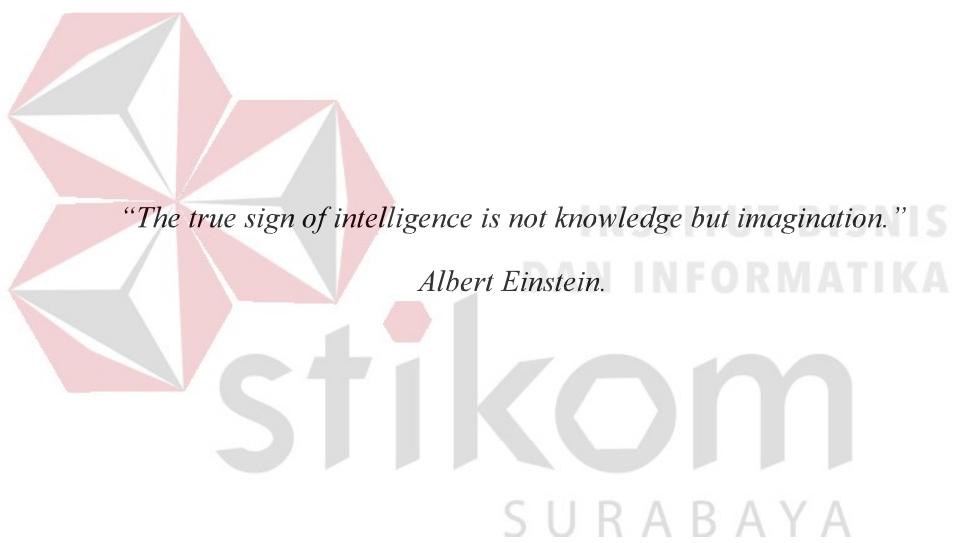
Diajukan sebagai salah satu syarat untuk menyelesaikan

mata kuliah Kerja Praktik



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

2019





LEMBAR PENGESAHAN

IMPLEMENTASI RFID (*RADIO FREQUENCY IDENTIFICATION*) UNTUK *TRACKING* BARANG PADA LOKET DAN LABORATORIUM KIMIA DI BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA

Laporan Kerja Praktik oleh

Fahmi Andriansyah

NIM : 16.41020.0031

Telah diperiksa, diuji, dan disetujui

Surabaya, Juli 2019

INSTITUT BISNIS

Disetujui: INFORMATIKA

Dosen Pembimbing,



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

NIDN 0721047201



Mengetahui,



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 : Fahmi Andriansyah

NIM : 16.41020.0031

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Laporan Kerja Praktik

Judul Karya : **IMPLEMENTASI RFID (RADIO FREQUENCY IDENTIFICATION) UNTUK TRACKING BARANG PADA LOKET DAN LABORATORIUM KIMIA DI 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 Royalty 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 kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Juli 2019



NIM : 16.41020.0031

ABSTRAK

Sistem RFID (*Radio Frequency Identifier*) untuk *Tracking* Barang Pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya merupakan sistem yang memanfaatkan kartu RFID sebagai media penyimpan identitas dari barang uji yang melakukan proses pengujian di Baristand Industri Surabaya.

Menurut Kepala Bagian Jasa Pelayanan Teknik Baristand Industri Surabaya, proses pendataan sampel masih dilakukan secara manual yaitu dengan menuliskan identitas sampel pada kertas HVS yang selanjutnya ditempelkan pada setiap sampel. Proses pemantauan sampel atau barang uji masih dilakukan dengan cara cek fisik, sehingga sering terjadi kekeliruan dalam pencatatan, pencarian, dan penelusuran data historis barang uji dan membutuhkan waktu yang lama. Teknologi RFID dipilih karena dapat menyimpan beberapa data yang diperlukan kedalam *chip* RFID *card* dan dapat dikombinasikan dengan penyimpanan *database* sehingga dapat menyimpan data dalam kapasitas besar sekaligus dapat dipantau melalui program aplikasi pada loket.

Dalam perancangan sistem ini penulis membuat aplikasi sebagai media pendataan barang masuk yang dirancang menggunakan C# Visual Studio 2017 dan disimpan pada *database* SQL Server 2017 dan juga *scanner* yang ditempatkan pada loket dan juga sebuah *scanner* yang ditempatkan di laboratorium untuk mempermudah pendataan dan pemantauan barang uji.

Kata kunci: NodeMCU, C#, RFID, SQL Server 2017

KATA PENGANTAR

Puji dan syukur atas kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan Kerja Praktik yang berjudul “Implementasi RFID (*Radio Frequency Identifier*) untuk *Tracking* Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya”. Laporan ini disusun berdasarkan hasil studi dalam pelaksanaan Kerja Praktik di Balai Riset dan Standardisasi Industri Surabaya yang dapat membantu Bagian Jasa Pelayanan Teknis Balai Riset dan Standardisasi Industri Surabaya.

Dalam pelaksanaan Kerja Praktik dan penyelesaian laporan Kerja Praktik ini, penulis mendapatkan bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Orang tua dan keluarga besar penulis yang selalu memberikan dukungan dan motivasi.
2. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Institut Bisnis dan Informatika Stikom Surabaya.
3. Ibu Weny Indah Kusumawati, S.Kom., M.MT., selaku dosen pembimbing yang telah memberikan dukungan penuh berupa motivasi, saran, dan wawasan bagi penulis selama pelaksanaan Kerja Praktik dan pembuatan laporan Kerja Praktik.
4. Ibu Fatimah, S.E., M.M., selaku penyelia dari Balai Riset dan Standardisasi Industri Surabaya yang telah memberikan berbagai informasi yang dibutuhkan penulis selama proses Kerja Praktik.

5. Dulur Sistem Komputer yang selalu siap memberikan bantuan, arahan, dan motivasi kepada penulis untuk dapat menyelesaikan laporan Kerja Praktik ini
6. Serta rekan-rekan Dewan Mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, yang selalu memberikan semangat, motivasi, dan bersedia membantu penulis dalam melaksanakan tugas di Dewan Mahasiswa selama melaksanakan Kerja Praktik.

Semoga Tuhan Yang Maha Esa memberikan rahmat-Nya kepada seluruh pihak yang membantu penulis dalam pelaksanaan Kerja Praktik dan penyelesaian laporan Kerja Praktik. Penulis menyadari di dalam laporan Kerja Praktik ini masih memiliki banyak kekurangan, meskipun demikian penulis tetap berharap laporan Kerja Praktik ini dapat bermanfaat bagi semua pihak dan dapat menjadi bahan acuan untuk penelitian selanjutnya.

INSTITUT BISNIS
DAN INFORMATIKA
stikom
SURABAYA
Surabaya, Juli 2019
Penulis

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
BAB II GAMBARAN UMUM PERUSAHAAN.....	5
2.1 Sejarah Balai Riset dan Standardisasi Industri Surabaya.....	5
2.2 Visi dan Misi	7
2.2.1 Visi.....	7
2.2.2 Misi.....	7
2.3 Lokasi Perusahaan.....	8
2.4 Struktur Organisasi	8
BAB III LANDASAN TEORI.....	10
3.1 RFID (<i>Radio Frequency Identifier</i>)	10
3.2 NodeMCU	12
3.3 Arduino Nano.....	13

3.4	Visual Studio 2017.....	14
3.5	PHP Hypertext Preprocessor.....	16
3.6	XAMPP	19
3.7	Wifi	20
3.8	<i>Hotspot</i>	20
	BAB IV HASIL DAN PEMBAHASAN	21
4.1	Prosedur Penelitian.....	21
4.2	Analisis Kebutuhan	23
4.3	Desain Sistem.....	24
4.3.1	Alur Sistem Registrasi Awal	25
4.3.2	Alur Sistem Proses Pengujian.....	26
4.3.3	Alur Sistem Registrasi Akhir.....	28
4.4	Simulasi dan Implementasi	29
4.4.1	Implementasi Desain Database Sistem	29
4.4.2	Perancangan dan Implementasi Aplikasi Pada Loket.....	35
4.4.3	Implementasi Alat <i>Scanner</i>	50
4.4.4	Koneksi <i>Database</i> Melalui PHP.....	51
4.5	Hasil dan Pembahasan Alat.....	55
4.5.1	Pengujian Aplikasi dan <i>Scanner</i> Loket	56
4.5.2	Pengujian <i>Scanner</i> Laboratorium Kimia	62
4.5.3	Pengujian <i>Reset</i> Kartu RFID	64
	BAB V PENUTUP.....	66
5.1	Kesimpulan	66
5.2	Saran.....	67
	DAFTAR PUSTAKA	68

DAFTAR TABEL

Halaman

Tabel 4.1 Kebutuhan Perangkat Lunak / *Software*..... 23

Tabel 4.2 Kebutuhan Perangkat Keras..... 23



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Lokasi Perusahaan.....	8
Gambar 2.2 Struktur Organisasi.....	9
Gambar 3.1 <i>RFID Reader RC 522</i>	12
Gambar 3.2 NodeMCU ESP8266 V3	12
Gambar 3.3 <i>Datasheet NodeMCU ESP8266 V3</i>	13
Gambar 3.4 Konfigurasi Pin Board Arduino Nano.....	14
Gambar 3.5 Tampilan Microsoft Visual Studio 2017	15
Gambar 3.6 Tampilan Awal SQL Server 2017	18
Gambar 3.7 Tampilan XAMPP <i>Control Panel</i>	19
Gambar 4.1 Tahapan Penggeraan	21
Gambar 4.2 Desain Alur Sistem.....	24
Gambar 4.3 Alur Sistem Registrasi Awal	25
Gambar 4.4 Alur Pendataan Barang Pada Laboratorium Kimia.....	26
Gambar 4.5 Alur Sistem <i>Scanner</i> Laboratorium	27
Gambar 4.6 <i>New Database</i>	30
Gambar 4.7 Konfigurasi <i>New Database</i>	30
Gambar 4.8 Membuat Tabel Baru.....	31
Gambar 4.9 Desain Tabel Master_Loket	32
Gambar 4.10 Desain Tabel tb_user.....	32
Gambar 4.11 Cara Mengisi tb_user Secara Manual.....	33
Gambar 4.12 Isi Data tb_user	33
Gambar 4.13 Desain Tabel Antrian_Lkimia.....	34

Gambar 4.14 Desain Tabel Proses_LKimia.....	34
Gambar 4.15 Desain Tabel Selesai_LKimia.....	35
Gambar 4.16 Konfigurasi <i>New Project</i> Visual Studio 2017	36
Gambar 4.17 Desain Aplikasi Loket.....	37
Gambar 4.18 Tampilan <i>Solution Explorer</i>	39
Gambar 4.19 Tampilan <i>Source Code</i> Program.Cs	39
Gambar 4.20 Cara Membuat Form Baru.....	40
Gambar 4.21 Desain Tampilan Utama Aplikasi Loket.....	41
Gambar 4.22 Desain Tampilan <i>Form Dataloket</i>	46
Gambar 4.23 <i>Source Code</i> Integrasi Ke <i>Database</i>	48
Gambar 4.24 Desain <i>Form labkimia</i>	48
Gambar 4.25 <i>Form Login</i>	56
Gambar 4.26 Kesalahan Memasukkan <i>Username</i> Atau <i>Password</i>	57
Gambar 4.27 Pengisian Data Registrasi.....	58
Gambar 4.28 Kesalahan Pengisian Data	58
Gambar 4.29 Data Yang Tersimpan Pada Tabel Master_Loket	59
Gambar 4.30 <i>Scanner</i> Loket	59
Gambar 4.31 Tampilan Menu Data Loket	60
Gambar 4.32 Tampilan Data Laboratorium Setelah <i>Scan</i> Pertama	61
Gambar 4.33 Tampilan Data Laboratorium Setelah <i>Scan</i> Kedua	61
Gambar 4.34 Tampilan Data Laboratorium Setelah <i>Scan</i> Ketiga.....	62
Gambar 4.35 Desain <i>Scanner</i> Laboratorium Kimia.....	63
Gambar 4.36 Isi <i>Database</i> Antrian_LKimia Setelah <i>Scanning</i> Pertama	63
Gambar 4.37 Isi <i>Database</i> Proses_LKimia Setelah <i>Scanning</i> Kedua.....	63

Gambar 4.38 Isi <i>Database</i> Selesai_LKimia Setelah <i>Scanning</i> Kedua.....	63
Gambar 4.39 Mencari Koneksi Pada <i>Scanner Reset</i>	64
Gambar 4.40 Mendapatkan Koneksi <i>Wireless</i>	64
Gambar 4.41 Berhasil Menghapus Data Barang Uji Pada <i>Database</i>	65
Gambar 4.42 Tidak Bisa Menampilkan Data Karena Telah Terhapus	65



DAFTAR LAMPIRAN

Halaman

Lampiran 1 Surat Balasan Instansi.....	69
Lampiran 2 Form KP-5 Acuan Kerja (Halaman 1).....	70
Lampiran 3 Form KP-5 Acuan Kerja (Halaman 2).....	71
Lampiran 4 Form KP-6 Log Harian dan Catatan Perubahan Acuan Kerja.....	72
Lampiran 5 Form KP-6 Log Harian dan Catatan Perubahan Acuan Kerja.....	73
Lampiran 6 Form KP-7 Kehadiran Kerja Praktik Halaman 1.....	74
Lampiran 7 Form KP-7 Kehadiran Kerja Praktik Halaman 2.....	75
Lampiran 8 Form Kartu Bimbingan Kerja Praktik	76
Lampiran 9 Biodata Diri	77



BAB I

PENDAHULUAN

1.1 Latar Belakang

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. Tugas pokok dari Baristand Industri Surabaya adalah melaksanakan riset dan standardisasi serta sertifikasi di bidang industri. Berdasarkan data Laporan Pelayanan Publik Baristand Industri tahun 2018, yang dikutip dari baristandsurabaya.kemenperin.go.id terdapat kenaikan jumlah barang yang melakukan pengujian di Baristand Industri Surabaya secara signifikan. Tercatat pada tahun 2017 terdapat 8911 sampel barang uji dan meningkat pada tahun 2018 menjadi 12402 sampel. Namun kenaikan jumlah sampel tersebut kurang didukung dengan adanya sistem yang memadai untuk proses pendataan sampel. Berdasarkan studi lapangan yang dilakukan oleh Tim Kerja Praktik Stikom Surabaya tahun 2019, Kepala Bagian Jasa Pelayanan Teknik Baristand Industri Surabaya menyatakan bahwa proses pendataan sampel masih dilakukan secara manual yaitu dengan menuliskan identitas sampel pada kertas HVS yang selanjutnya ditempelkan pada setiap sampel. Proses pemantauan sampel atau barang uji masih dilakukan dengan cara cek fisik pada sampel.

Pendataan sampel / barang uji secara manual memiliki banyak sekali resiko dan kekurangan, mulai dari kehilangan barang uji ataupun mekanisme proses pengujian yang lamban. Tercatat pada tahun 2018, berdasarkan Data Keluhan yang dipublikasikan oleh baristandsurabaya.kemenperin.go.id terdapat peningkatan jumlah keluhan sebesar 6% dari tahun 2017 hingga 2018 tentang lambannya proses pengujian dan terdapat beberapa sampel yang hilang.

Untuk itu pada penelitian ini akan dilakukan perancangan suatu alat *tracking* barang uji (sampel) di Baristand Industri Surabaya untuk mempermudah proses pencatatan maupun pendataan barang uji, sehingga dapat meningkatkan efisiensi waktu dalam proses pengujian maupun mengurangi resiko kehilangan barang uji. Mekanisme dari sistem ini yaitu dengan menyimpan identitas barang pada kartu RFID dan selanjutnya proses pengecekan atau pendataan dilakukan dengan *scan* kartu tersebut dengan *scanner* RFID, sehingga aktifitas barang uji dapat dilakukan secara otomatis dan dapat dipantau melalui aplikasi yang telah disiapkan.

1.2 Perumusan Masalah

Berdasarkan uraian tersebut, maka dapat dirumuskan “Bagaimana mengimplementasikan RFID (*Radio Frequency Identifier*) Untuk *Tracking* Barang Pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya?”.

1.3 Batasan Masalah

Dalam perancangan sistem ini, pembahasan masalah dibatasi pada beberapa hal berikut:

1. Perangkat *microcontroller* yang digunakan adalah NodeMCU dan Arduino Nano.
2. Komunikasi yang digunakan adalah komunikasi serial dan *wireless*.
3. Menggunakan SQL Server sebagai media penyimpanan data (*database system*).
4. Aplikasi desktop dirancang menggunakan C# Visual Studio 2017.

1.4 Tujuan Penelitian

Berdasarkan uraian latar belakang dan rumusan masalah di atas, dalam Kerja Praktik ini didapatkan tujuan pembuatan laporan adalah membangun sistem Implementasi RFID (*Radio Frequency Identifier*) untuk *Tracking* Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya.

1.5 Manfaat Penelitian

Manfaat yang diperoleh dari implementasi RFID (*Radio Frequency Identifier*) untuk *Tracking* Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya yaitu:

- Dapat mempermudah karyawan dalam mengakses informasi setiap barang uji.
- Dapat mengetahui proses yang sedang dilakukan pada setiap barang uji.

1.6 Sistematika Penulisan

Sistematika dalam penyusunan laporan Kerja Praktik ini akan dijabarkan dalam setiap bab dengan pembagian sebagai berikut:

1. BAB I : Pendahuluan

Bab ini membahas mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan yang berisi tentang penjelasan singkat pada masing-masing bab.

2. BAB II : Gambaran Umum Perusahaan

Bab ini membahas mengenai gambaran umum, visi dan misi, serta struktur organisasi Balai Riset dan Standardisasi Industri Surabaya.

3. BAB III : Landasan Teori

Bab ini membahas mengenai berbagai macam teori yang mendukung dalam pengimplementasian RFID (*Radio Frequency Identifier*) untuk *tracking* barang pada loket dan laboratorium kimia di Balai Riset dan Standardisasi Industri Surabaya.

4. BAB IV : Hasil dan Pembahasan

Bab ini membahas mengenai proses perancangan dan hasil dari implementasi RFID (*Radio Frequency Identifier*) untuk *Tracking* Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya.

5. BAB V : Penutup

Pada bab ini membahas tentang kesimpulan dari hasil analisis.

BAB II

GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah Balai Riset dan Standardisasi Industri Surabaya

Balai Riset dan Standardisasi Industri Surabaya (BARISTAND INDUSTRI SURABAYA), sejak awal berdirinya telah mengalami beberapa kali perubahan nama dan perpindahan lokasidari satu kota ke kota lain. 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.

Guna 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 dan Misi

2.2.1 Visi

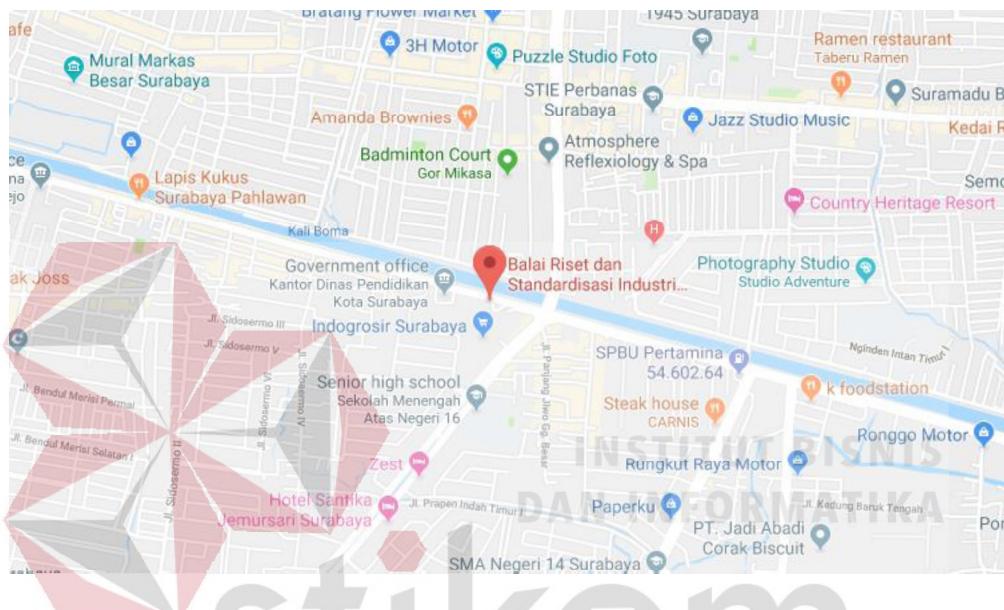
Sebagai Lembaga Riset Dan Standarisasi Terkemuka Yang Menjadi Mitra Industri Elektronika Dan Telematika Nasional Dalam Berperan Sebagai Basis Produksi Yang Melayani Kebutuhan Nasional Maupun Dunia Pada Tahun 2025.

2.2.2 Misi

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.

2.3 Lokasi Perusahaan

Lokasi Balai Riset dan Standardisasi Industri Surabaya yaitu di Jl. Jagir Wonokromo No.360, Panjang Jiwo, Tenggilis Mejoyo, Kota SBY, Jawa Timur 60244. Peta dari lokasi Balai Riset dan Standardisasi Industri Surabaya dapat dilihat pada gambar 2.1.



Gambar 2.1 Lokasi Perusahaan

2.4 Struktur Organisasi

Struktur organisasi yang ada di Balai Riset dan Standardisasi Industri Surabaya dapat dilihat pada gambar 2.2.



Gambar 2.2 Struktur Organisasi



BAB III

LANDASAN TEORI

3.1 RFID (*Radio Frequency Identifier*)

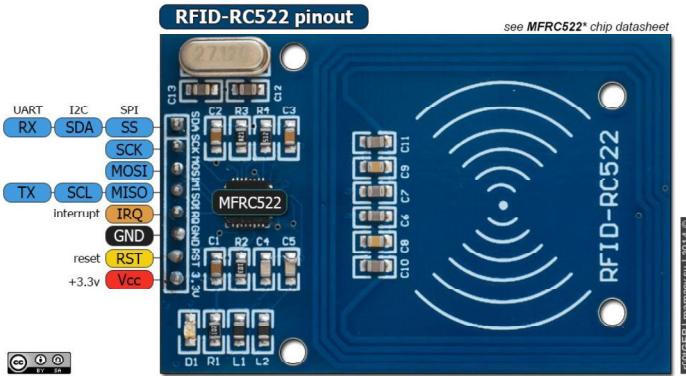
RFID (*Radio Frequency Identification*) merupakan kombinasi dari frekuensi radio berbasis teknologi dan teknologi *microchip*. Informasi yang terkandung di dalam tag *microchip* dan ditempelkan pada bahan pustaka dapat dibaca menggunakan teknologi frekuensi radio. Sebuah alat pembaca (alias sensor, pemindai, atau interrogator) mencari antena pada tag dan mengambil informasi dari *microchip* dalam perangkat RFID. (Boss, Richard). Chip RFID menjadi bagian yang sangat penting, karena chip yang digunakan telah menjadi lebih kecil dan lebih pintar sampai ke titik di mana *chip* tersebut dapat ditambahkan pada setiap jenis dokumen dan dapat dibaca dan diperbarui dari jarak. (Narayanan, 2005)

Pengertian RFID secara umum adalah sebuah teknologi terbaru untuk mengidentifikasi atau mendeteksi sebuah objek (benda/orang) dengan menggunakan gelombang radio, yang terdiri dari satu atau lebih alat pembaca/*transponder interrogator* dan RF transfer data yang dicapai dengan cara yang sesuai dimodulasi induktif atau memancarkan pembawa *elektromagnetik*. Selain itu dapat digunakan sebagai pembawa data, dengan informasi yang ditulis dan diperbarui untuk tag pada saat digunakan.

Sistem RFID membawa data transponder yang sesuai, umumnya dikenal sebagai tag, dan mengambil data, dengan mesin yang bisa membacakan arti, pada waktu dan tempat yang sesuai untuk memenuhi kebutuhan aplikasi tertentu. RFID

(*Radio Frequency Identification*) diciptakan pada tahun 1969, dipatenkan pada 1973, pertama kali digunakan dalam lingkungan industri di tahun 1980-an, dan standar yang disajikan pada tahun 2001 dan dikembangkan sebagai pengganti atau penerus dari teknologi *barcode*. Penggunaan RFID di perpustakaan yang mana menggantikan sistem *barcode* telah ada sejak tahun 1990-an. Dengan menggunakan RFID memungkinkan pengamanan, dan penemuan kembali bahan pustaka di perpustakaan dengan mudah. Secara keseluruhan rak bahan pustaka dapat dibaca dengan alat pembaca sinyal pada *portable scan reader*. Kemudian pada hasil *portable scan reader* akan dilaporkan apakah ada bahan pustaka yang hilang atau dipinjam (keluar dari rak). Sebuah label RFID yang ditempelkan pada bahan pustaka akan mengidentifikasi bahan pustaka dan akan melindunginya. Ketika pemustaka melakukan peminjaman dan membawa bahan pustaka keluar dari perpustakaan maka label RFID akan terbaca oleh sistem. (Ahson & Mohammed, 2008)

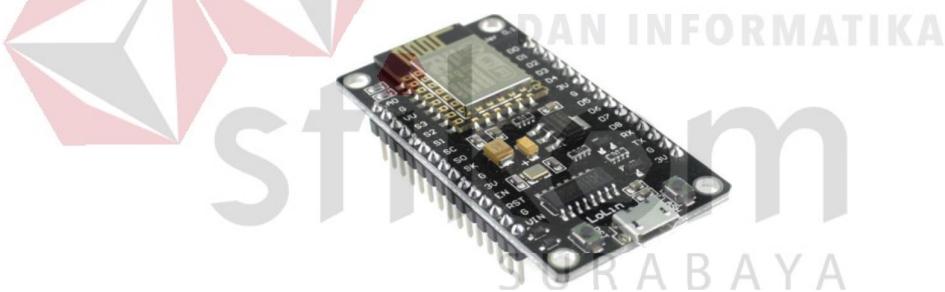
Pada implementasi sistem RFID (*Radio Frequency Identifier*) untuk tracking barang pada loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya ini penulis menggunakan modul RFID *reader RC 522* yang dapat dilihat pada gambar 3.1.



Gambar 3.1 RFID Reader RC 522

3.2 NodeMCU

NodeMCU adalah sebuah platform IoT yang bersifat opensource. Terdiri dari perangkat keras berupa System On Chip ESP8266 dari ESP8266 buatan Esperessif System, seperti tampak pada Gambar 3.2.



Gambar 3.2 NodeMCU ESP8266 V3

NodeMCU bisa dianalogikan sebagai *board arduino* yang terkoneksi dengan ESP8622. NodeMCU telah mem-*package* ESP8266 ke dalam sebuah *board* yang sudah terintegrasi dengan berbagai *feature* selayaknya *microcontroller* dan kapalitas akses terhadap wifi dan juga chip komunikasi yang berupa USB to serial, sehingga dalam pemrograman hanya dibutuhkan kabel data USB (Syahwil, 2013). Karena sumber utama dari NodeMCU adalah ESP8266 khususnya seri

ESP-12 yang termasuk ESP-12E, maka fitur-fitur yang dimiliki oleh NodeMCU akan lebih kurang serupa dengan ESP-12. Beberapa fitur yang tersedia antara lain:

1. 10 Port GPIO dari D0 – D10
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antaruka 1 Wire
5. *Analog to Digital Converter*

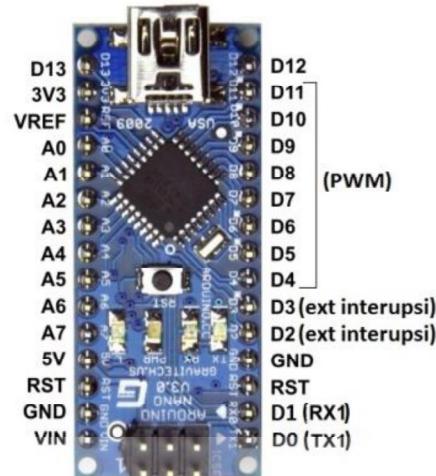


Gambar 3.3 Datasheet NodeMCU ESP8266 V3

3.3 Arduino Nano

Arduino Nano adalah salah satu varian dari produk *board* mikrokontroller keluaran Arduino. Arduino Nano adalah *board* Arduino terkecil, menggunakan mikrokontroller Atmega 328 untuk Arduino Nano 3.x dan Atmega168 untuk Arduino Nano 2.x. Varian ini mempunyai rangkaian yang sama dengan jenis Arduino Duemilanove, tetapi dengan ukuran dan desain PCB yang berbeda. Arduino Nano tidak dilengkapi dengan soket catudaya, tetapi terdapat pin untuk

catu daya luar atau dapat menggunakan catu daya dari *mini USB port*. Arduino Nano didesain dan diproduksi oleh Gravitech. (Djapri, 2015)



Gambar 3.4 Konfigurasi Pin Board Arduino Nano

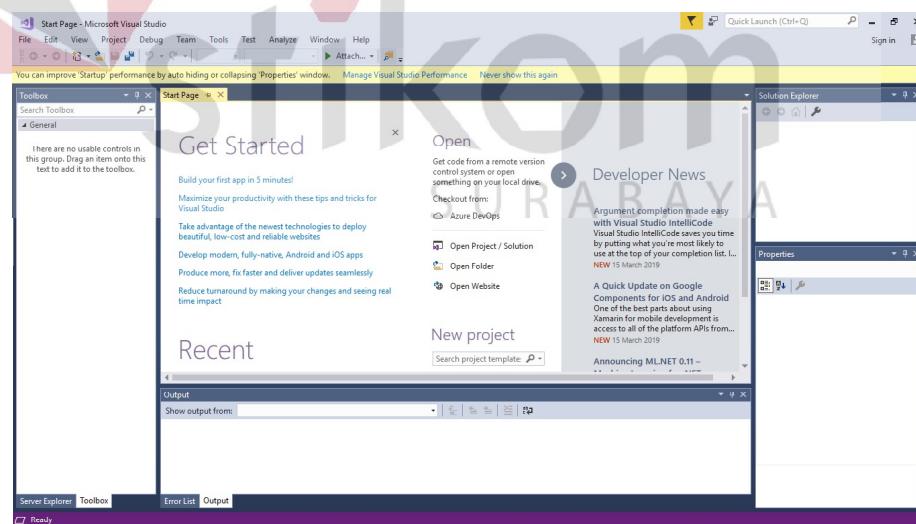
3.4 Visual Studio 2017

Microsoft Visual Studio by merupakan sebuah perangkat lunak lengkap (suite) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya, dalam bentuk aplikasi console, aplikasi Windows, ataupun aplikasi Web. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa MSDN Library). Komplier yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows)

ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).

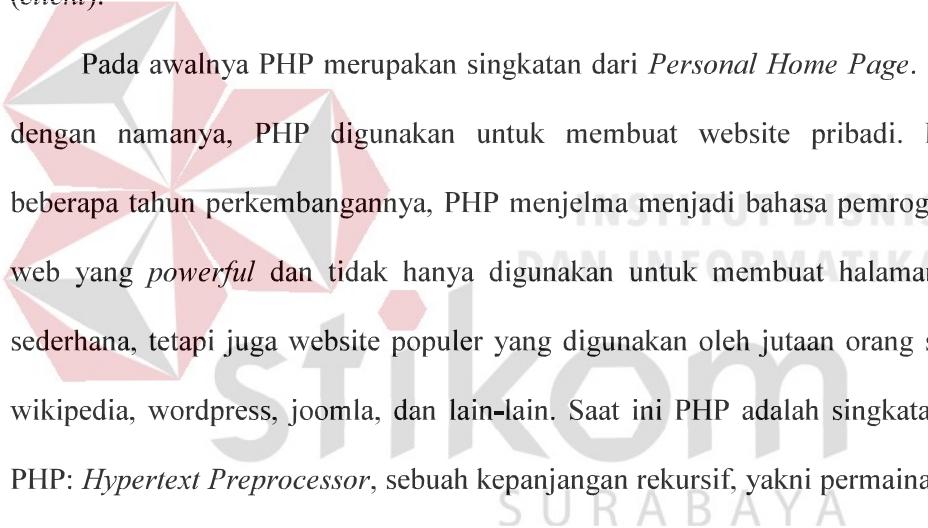
Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform Microsoft .NET Framework 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET Framework 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET Framework 1.1, dan Visual Studio 2002 ditujukan untuk .NET Framework 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998). (Wikipedia, 2019)



Gambar 3.5 Tampilan Microsoft Visual Studio 2017

3.5 PHP Hypertext Preprocessor

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>. PHP disebut bahasa pemrograman *server-side* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada web browser (*client*).



Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang *powerful* dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dan lain-lain. Saat ini PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor*.

PHP dapat digunakan dengan gratis (*free*) dan bersifat *Open Source*. PHP dirilis dalam lisensi PHP *License*, sedikit berbeda dengan lisensi GNU *General Public License* (GPL) yang biasa digunakan untuk proyek *Open Source*. Kemudahan dan kepopuleran PHP sudah menjadi standar bagi *programmer* web di seluruh dunia. Menurut wikipedia pada februari 2014, sekitar 82% dari web server di dunia menggunakan PHP. PHP juga menjadi dasar dari aplikasi CMS

(*Content Management System*) populer seperti Joomla, Drupal, dan WordPress.

(Alan, 2017)

SQL Server 2017 Microsoft SQL Server merupakan produk *Relational Database Management System* (RDBMS) yang dibuat oleh Microsoft. Orang sering menyebutnya dengan SQL Server saja. Microsoft SQL Server juga mendukung SQL sebagai bahasa untuk memproses *query* ke dalam *database*. Microsoft SQL banyak digunakan pada dunia bisnis, pendidikan atau juga pemerintahan sebagai solusi *database* atau penyimpanan data. Microsoft mengeluarkan SQL Server yang merupakan versi yang banyak digunakan.

Berikut ini adalah beberapa fitur yang dari sekian banyak fitur yang ada pada SQL Server:

A. XML Support.

Dengan fitur ini, Anda bisa menyimpan dokumen XML dalam suatu tabel, meng-*query* data ke dalam format XML melalui *Transact-SQL* dan lain sebagainya.

B. Multi-Instance Support.

C. Fitur ini memungkinkan anda untuk menjalankan beberapa *database engine* SQL Server pada mesin yang sama.

D. Data Warehousing and Business Intelligence (BI) Improvements.

SQL Server dilengkapi dengan fungsi-fungsi untuk keperluan *Business Intelligence* melalui *Analysis Services*. Selain itu, SQL Server 2000 juga ditambahi dengan *tools* untuk keperluan *data mining*.

E. Performance and Scalability Improvements.

SQL Server menerapkan *distributed partitioned views* yang memungkinkan untuk membagi *workload* ke beberapa *server* sekaligus. Peningkatan lainnya juga dicapai di sisi DBCC, *indexed view*, dan *index reorganization*.

F. *Query Analyzer Improvements*.

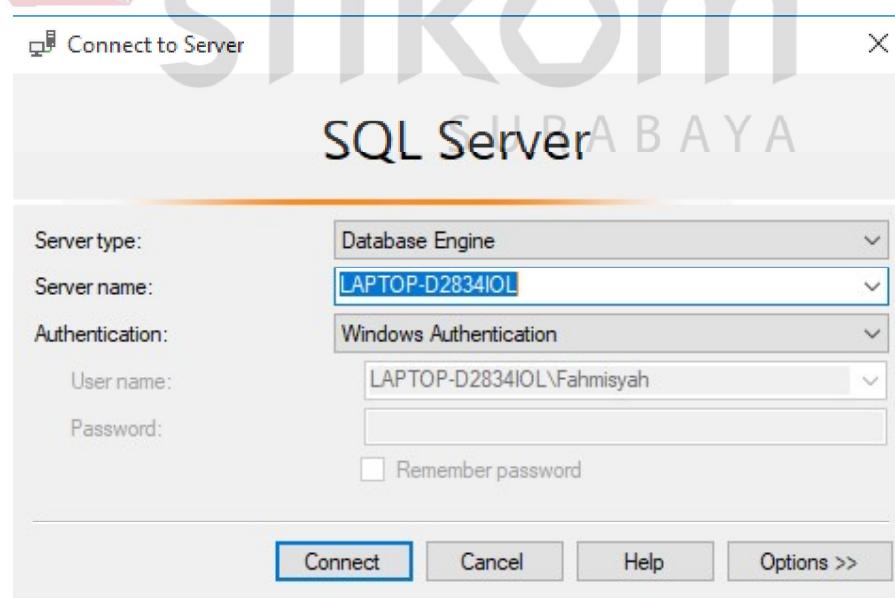
Fitur yang dihadirkan antara lain: *integrated debugger*, *object browser*, dan fasilitas *object search*.

G. DTS Enhancement.

Fasilitas ini sekarang sudah mampu untuk memperhatikan *primary key* dan *foreign key constraints*. Ini berguna pada saat *migrasi* tabel dari RDBMS lain.

H. Transact-SQL Enhancements.

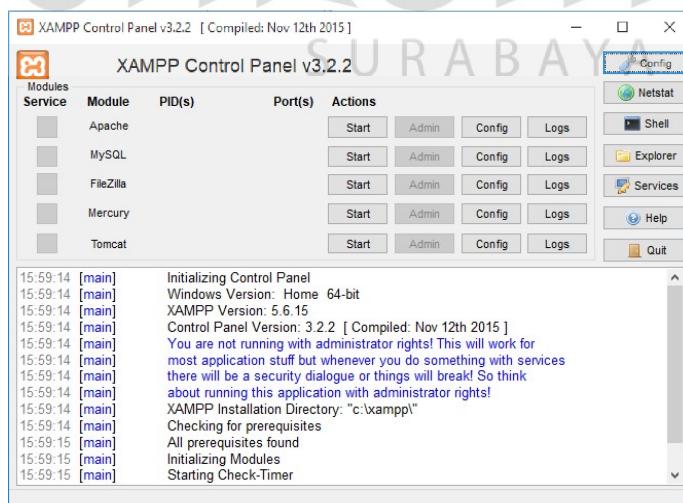
Salah satu peningkatan disini adalah T-SQL sudah mendukung UDF (*User-Definable Function*). Ini memungkinkan Anda untuk menyimpan rutin-rutin ke dalam *database engine*.



Gambar 3.6 Tampilan Awal SQL Server 2017

3.6 XAMPP

XAMPP adalah aplikasi *web server* instan yang dibutuhkan untuk membangun aplikasi berbasis web. Fungsi XAMPP adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache, *http server*, MySQL, *database*, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (X=Cross Platform), Apache, MySQL, PHP dan Perl. XAMPP adalah *software web server* apache yang di dalamnya tertanam *server* MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis. XAMPP sendiri mendukung dua sistem operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalannya menggunakan *command line* sedangkan untuk windows dalam proses penginstalannya menggunakan *interface graphic* sehingga lebih mudah dalam penggunaan XAMPP di Windows dibanding dengan Linux. Didalam XAMPP ada 3 komponen utama yang ditanam di dalamnya yaitu *web server* Apache, PHP, dan MySQL. (Rahastri, 2015)



Gambar 3.7 Tampilan XAMPP *Control Panel*

3.7 Wifi

Wireless Fidelity (Wifi) merupakan sebuah teknologi yang terdiri dari beberapa komputer yang terhubung dalam sebuah jaringan tanpa menggunakan lantaran kabel penghubung tapi menggunakan *Local Area Network* (WLAN).

Wireless LAN (WLAN) merupakan teknologi LAN yang menggunakan media penghantar frekuensi dan transmisi radio pada lokasi atau area tertentu sebagai pengganti kabel. *Hotspot* (wifi) adalah satu standar *Wireless Networking* tanpa kabel, hanya dengan komponen yang sesuai dapat terkoneksi ke jaringan. (Tri Kuntoro, 2005)

3.8 Hotspot

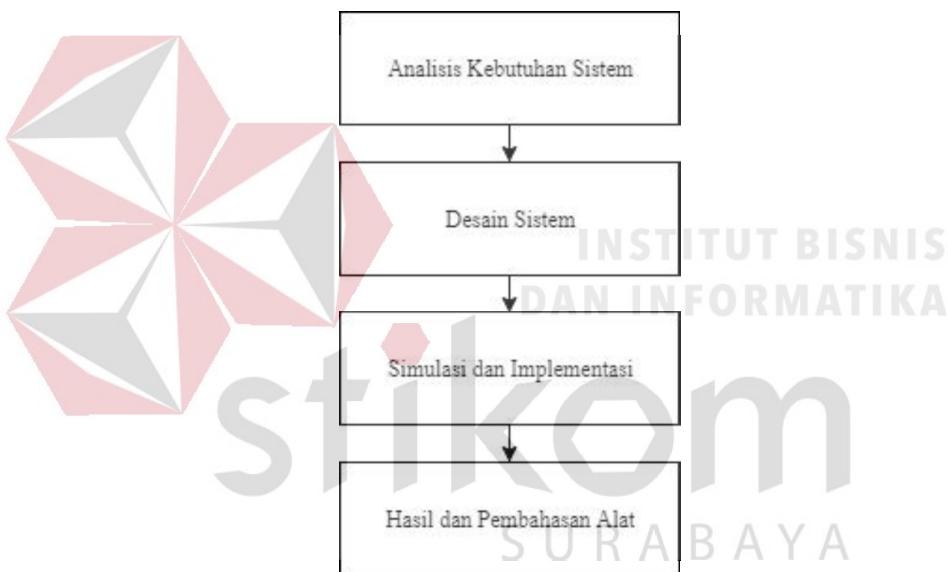
Hotspot merupakan wilayah yang terdapat *access point* sehingga *user* dapat menggunakannya untuk mengkoneksikan jaringan *wireless* dengan menggunakan PC, laptop, atau perangkat lainnya yang terdapat fitur wifi. *Hotspot* adalah definisi untuk daerah yang dilayani oleh satu *Access Point Wireless LAN* standar 802.11a/b/g, dimana pengguna (*user*) dapat masuk ke dalam *Access Point* secara bebas dan mobile menggunakan perangkat sejenis *notebook*, PDA atau lainnya (Stiawan, 2009). Sedangkan menurut (Tri Kuntoro, 2005). *Hotspot* adalah koneksi jaringan internet yang menggunakan *wireless networking* atau jaringan tanpa kabel.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Prosedur Penelitian

Prosedur penelitian merupakan tahap awal dari pengerjaan ini dengan menentukan seluruh tahapan yang akan dilalui, dibawah ini adalah tahapan dari Implementasi RFID (*Radio Frequency Identifier*) untuk *Tracking* Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya.



Gambar 4.1 Tahapan Pengerjaan

Pembahasan dari setiap langkah pada prosedur penelitian akan dijelaskan dibawah ini:

1. Analisis Kebutuhan Sistem

Pada tahap ini dilakukan analisis kebutuhan sistem sebagai bagian dari studi awal bertujuan untuk mengidentifikasi masalah dan kebutuhan spesifik sistem.

Kebutuhan spesifik sistem adalah spesifikasi mengenai hal-hal yang akan

dilakukan sistem ketika diimplementasikan seperti metode dan kebutuhan sistem berupa *software* dan *hardware*.

2. Desain Sistem

Dari data-data yang sudah didapatkan sebelumnya dari analisis kebutuhan, pada tahap desain ini akan dibuat gambar desain alur sistem kerja yang akan dibangun, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada. Desain bisa berupa desain struktur topologi jaringan, alur sistem kerja dan sebagainya yang akan memberikan gambaran yang jelas tentang *project* yang akan dibangun.

3. Simulasi dan Implementasi

Tahap simulasi bertujuan untuk melihat kinerja awal dari penelitian yang akan dilakukan pada aplikasi simulasi sebagai bahan pertimbangan awal dari penelitian yang akan dilakukan sebagai bahan pertimbangan sebelum sistem diterapkan, sehingga dalam tahap implementasi rancangan yang dibuat akan diterapkan pada Balai Riset dan Standardisasi Industri Surabaya.

4. Hasil dan Pembahasan

Tahap yang terakhir adalah analisis terhadap hasil dari semua yang telah dilakukan pada proses implementasi. Hasil analisis berupa nilai yang telah ditentukan menjadi point penting/tolak ukur dari keberhasilan. Tolak ukur yang digunakan untuk menganalisis adalah keberhasilan implementasi dari sistem yang dirancang.

4.2 Analisis Kebutuhan

Tahap analisis kebutuhan sistem pada *project* ini yaitu kebutuhan model aplikasi yang akan digunakan dan kebutuhan perangkat yang menunjang berjalannya sistem. Aplikasi digunakan pada loket untuk melakukan pendataan barang yang akan melakukan proses pengujian di Balai Riset dan Standardisasi Industri Surabaya, sedangkan kebutuhan perangkat meliputi perangkat lunak yang digunakan untuk perancangan sistem aplikasi dan desain *database* maupun kebutuhan perangkat keras. Adapun kebutuhan perangkat dapat dilihat pada tabel 4.1 dan tabel 4.2.

Tabel 4.1 Kebutuhan Perangkat Lunak / *Software*

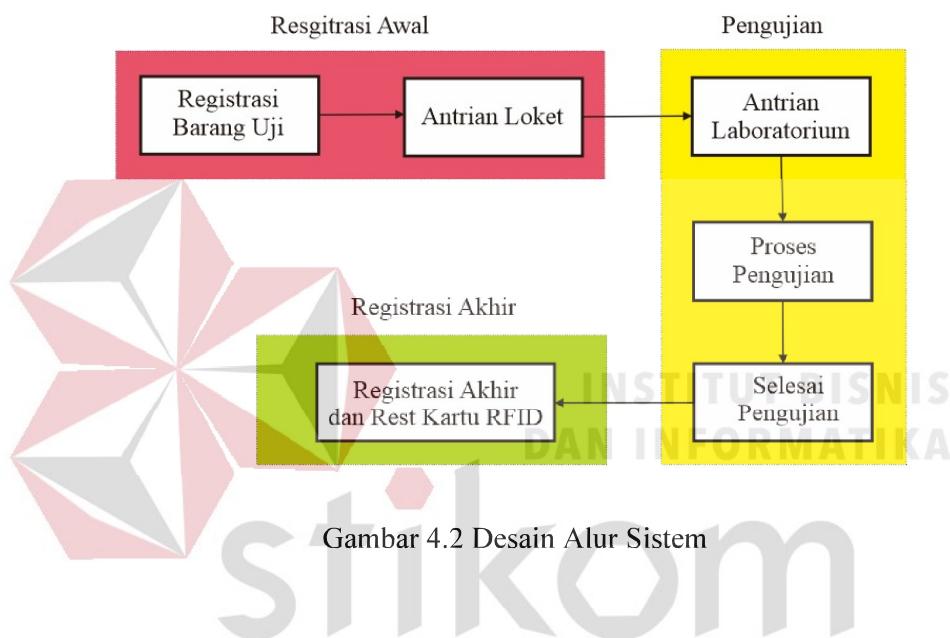
No	Nama	Fungsi
1	Microsoft Visual Studio 2017	Membangun aplikasi pendataan barang dengan C# Programming.
2	Microsoft SQL Server 2017	Sebagai penyimpanan data (<i>database</i>) barang uji.
3	Arduino IDE	Sebagai media untuk menuliskan <i>code</i> program pada Arduino Nano dan NodeMCU ESP8266 V3.
4	Windows 10	Sebagai sistem operasi yang digunakan
5	Xampp	Sebagai <i>web server localhost</i>
6	Sublime	Sebagai <i>text editor</i> untuk menuliskan <i>code php</i>

Tabel 4.2 Kebutuhan Perangkat Keras

No	Nama/Jenis Perangkat	Jumlah
1	NodeMCU	2
2	Arduino Nano	1
3	Kabel Serial USB2	1
4	LCD 16x2	3
5	Modul RFID RC522	3

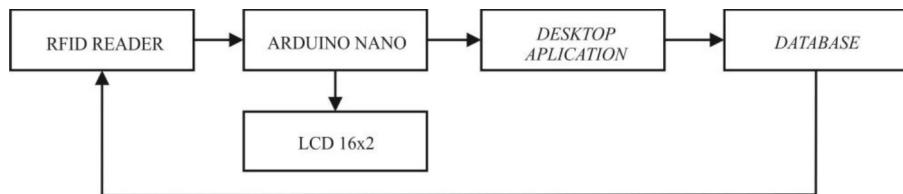
No	Nama/Jenis Perangkat	Jumlah
6	<i>Push Button</i>	3
7	<i>Switch</i>	2
8	<i>Sticker RFID /RFID Card</i>	3
9	<i>Chassing 3D Printing</i>	3

4.3 Desain Sistem



Pada gambar 4.2 telah digambarkan beberapa alur yang harus dilalui oleh barang uji ketika melakukan pengujian di Balai Riset dan Standardisasi Industri Surabaya, dari gambar tersebut secara umum terdapat tiga alur sistem yang digunakan untuk mengimplementasikan RFID untuk *taracking* barang pada loket dan laboratorium kimia di Balai Riset dan Standardisasi Industri Surabaya, tiga alur sistem yang disebut antara lain alur sistem pada loket untuk registrasi awal, alur sistem pada laboratorium, dan alur sistem pada loket untuk registrasi akhir.

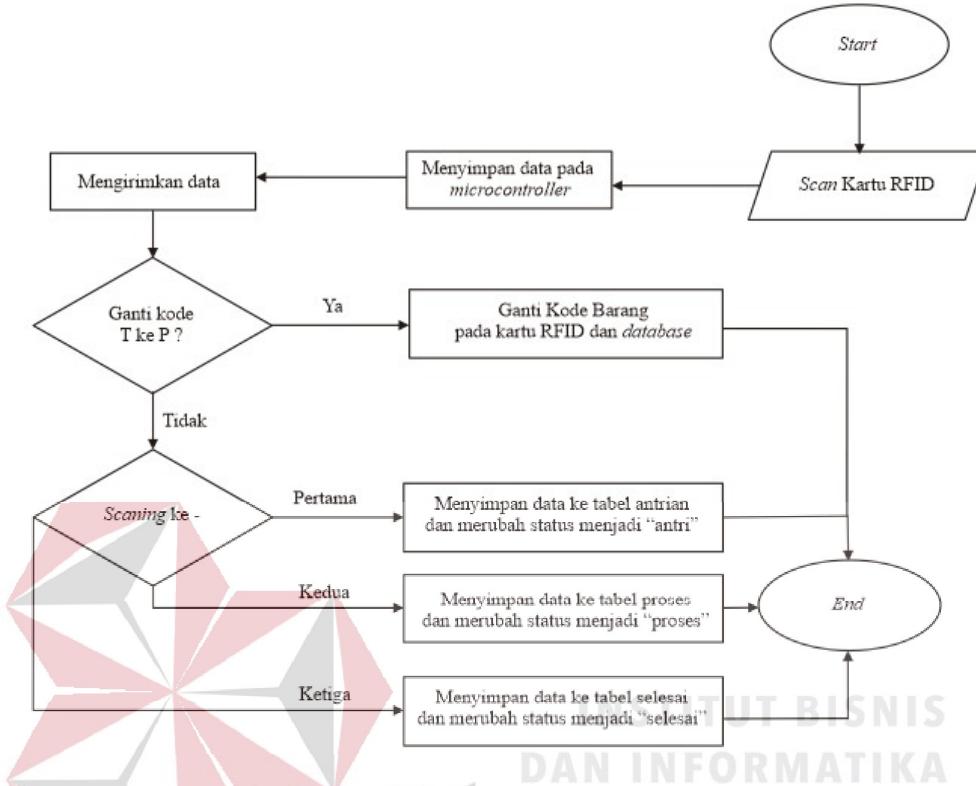
4.3.1 Alur Sistem Registrasi Awal



Gambar 4.3 Alur Sistem Registrasi Awal

Pada gambar 4.3 dijelaskan bahwa kartu RFID yang berbentuk stiker nantinya akan ditempelkan ke barang uji akan dibaca oleh RFID *reader*-RC 522 yang dihubungkan dengan Arduino Nano sebagai *microcontroller* dari alat yang ditempatkan di loket untuk registrasi awal. RFID *reader* akan membaca kode unik pada setiap RFID *card* yang disebut UID. Kode unik tersebut akan ditampilkan melalui LCD 16x2 dan juga ditampilkan pada desktop *application*. Karyawan loket akan memberikan beberapa data yang diperlukan melalui desktop *application* sekaligus memberikan identitas barang berupa kode barang dengan diawali huruf "T" yang berarti data tersebut belum melakukan pembayaran dan belum dilakukan pengujian terhadap barang uji tersebut. Data yang telah ditulis pada desktop application akan disimpan pada *database*. Setelah melakukan registrasi awal, barang uji akan ditempatkan pada antrian loket hingga mendapatkan informasi bahwa barang uji telah melakukan pembayaran. Selanjutnya barang uji yang telah melakukan pembayaran akan ditempatkan pada antrian laboratorium untuk selanjutnya dilakukan proses pengujian pada bagian laboratorium.

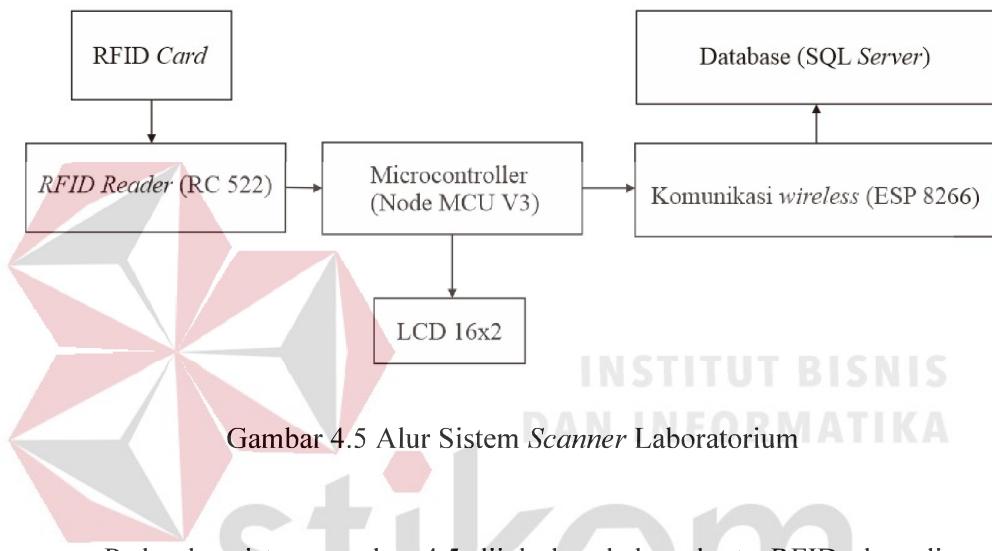
4.3.2 Alur Sistem Proses Pengujian



Gambar 4.4 Alur Pendataan Barang Pada Laboratorium Kimia

Pada proses pengujian sesuai pada gambar 4.4, barang yang telah masuk di antrian loket dan telah melakukan pembayaran dipindahkan ke antrian laboratorium untuk melakukan perubahan status dari kode “T” menjadi “P” menggunakan *scanner* laboratorium. Proses perubahan status tersebut menandakan bahwa barang uji telah melakukan pembayaran dan telah siap melakukan pengujian. Melakukan proses *scanning* dengan *scanner* laboratorium untuk menyimpan status barang uji, bahwa saat ini barang uji berada di antrian pengujian. Setelah terjadi perubahan kode dan status barang uji berada pada tahap antrian pengujian, maka barang uji akan melakukan proses pengujian dan harus dilakukan proses *scan* yang kedua untuk menyimpan status barang uji, yakni

sedang melakukan proses pengujian. Setelah barang uji selesai melakukan proses pengujian maka harus dilakukan scan yang ketiga untuk memperbarui data status barang uji, bahwa barang uji telah selesai melalui tahap pengujian barang. Adapun alur sistem desain alat (*scanner*) yang ditempatkan pada laboratorium kimia adalah seperti terlihat pada gambar 4.5.



Pada alur sistem gambar 4.5 dijelaskan bahwa kartu RFID akan di *scan* melalui RFID *reader* yang menggunakan modul RFID *reader* RC 522. Data UID dan kode barang yang didapat pada setiap kartu RFID akan ditampilkan pada LCD 16x2. Selanjutnya data yang telah terbaca akan disimpan pada EEPROM NodeMCU yang berfungsi sebagai mikrokontroller *scanner* laboratorium, kemudian data yang tersimpan di EEPROM tersebut akan dikirimkan ke *database* melalui koneksi *wireless* dengan menggunakan Modul ESP 8266 yang terdapat pada NodeMCU dan *hotspot / wifi* sebagai *transmiter*. Pengiriman data pada *scanner* dibagi menjadi tiga kondisi, pengiriman data pertama untuk kode barang yang sama menandakan bahwa barang tersebut telah siap melakukan pengujian

dan berlokasi di antrian laoratorium. Pengiriman data kedua menandakan bahwa barang uji sedang melakukan proses pengujian di laboratorium kimia. Pengiriman data ketiga menandakan bahwa barang uji telah melakukan selesai melalui proses pengujian.

4.3.3 Alur Sistem Registrasi Akhir

Pada alur ini kartu RFID yang digunakan sebagai identitas pada setiap barang uji akan melakukan proses *reset* yakni dengan menghapus seluruh data yang tersimpan pada kartu RFID dan database. Adapun alur kerja dari alat yang dirancang adalah dengan mendekatkan kartu RFID pada *scanner* dengan menggunakan modul RC 522 sebagai RFID *reader*, kemudian *scanner* akan menampilkan UID dan kode barang pada LCD 16x2. Pengguna harus menekan tombol yang terdapat pada *scanner* untuk menyimpan data UID pada EEPROM NodeMCU yang digunakan sebagai mikrokontroller *scanner*. Data UID yang tersimpan pada EEPROM NodeMCU akan dibandingkan dengan data yang tersimpan pada *database server* melalui komunikasi wireless menggunakan ESP 8266 pada NodeMCU. Sistem akan menghapus seluruh data yang tersimpan di *database* yang ada keterkaitan dengan penggunaan UID kartu yang akan di *reset*. Setelah penghapusan data pada *database* selesai selanjutnya sistem akan menghapus data yang tersimpan di kartu RFID agar kartu tersebut dapat digunakan kembali.

4.4 Simulasi dan Implementasi

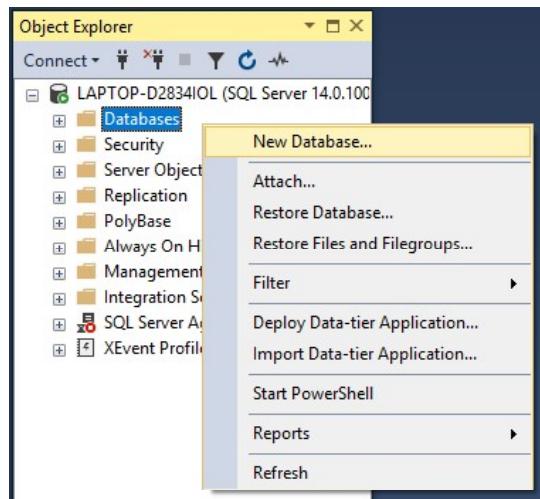
Tahap simulasi dan implementasi dibagi menjadi tiga, yaitu implementasi desain database sistem, perancangan dan implementasi aplikasi registrasi awal, dan perancangan dan implementasi alat *scanner*.

4.4.1 Implementasi Desain Database Sistem

Pada proses perancangan *database* penulis menggunakan Microsoft SQL Server 2017, adapun langkah-langkah perancangan database dengan Microsoft SQL Server 2017 akan dijelaskan sebagai berikut:

A. Membuat Database Baru

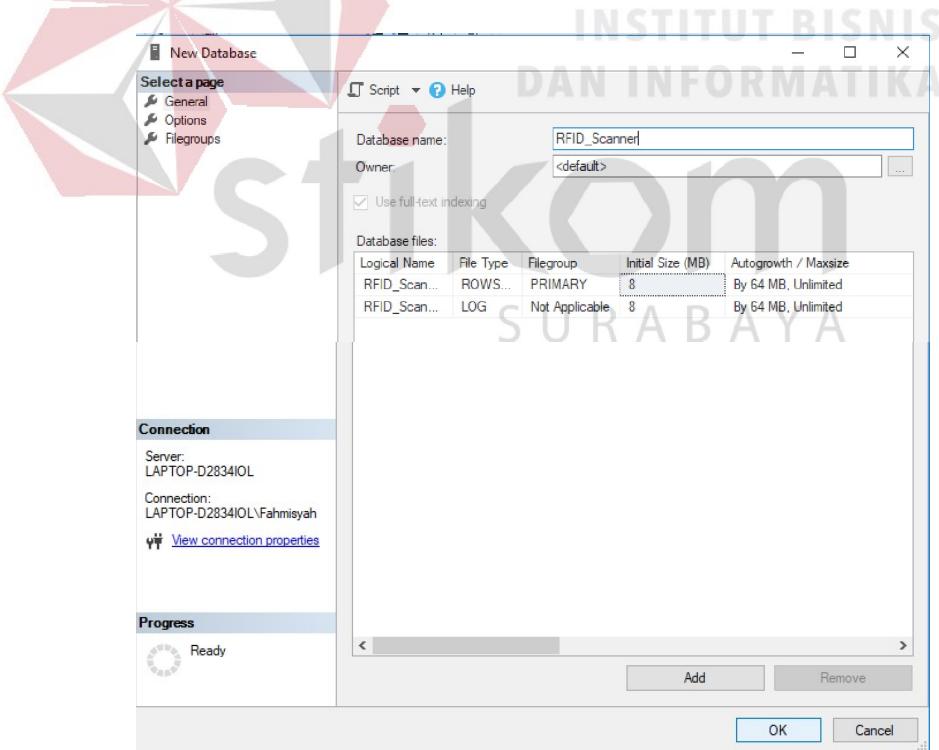
Membuka terlebih dahulu aplikasi *Microsoft SQL Server Management Studio 17*, kemudian meng-klik kanan pada folder *database* yang terdapat pada *object explorer* yang terdapat disisi kiri *interface Microsoft SQL Server Management Studio 17* lalu memilih menu *New Database* seperti pada gambar 4.6.



Gambar 4.6 New Database

Selanjutnya memberikan nama pada *database* yang telah dibuat dan menekan

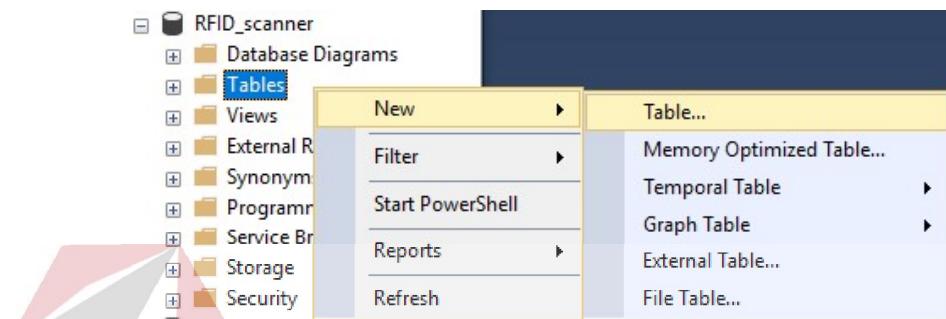
Ok untuk melanjutkan proses perancangan *database* seperti gambar 4.7.



Gambar 4.7 Konfigurasi New Database

B. Membuat Tabel Master Loket

Setelah berhasil membuat database maka akan muncul *icon database* pada folder *database SQL Sever*. Selanjutnya melakukan *double click* pada *icon database* tersebut lalu meng-klik kanan pada folder *table* dan memilih menu *new table* seperti pada gambar 4.8.



Gambar 4.8 Membuat Tabel Baru

Selanjutnya mengisi tabel dengan beberapa atribut yang diperlukan seperti gambar 4.9 berikut, dan menjadikan *UID* sebagai *primary key* dari tabel master loket. Setelah tabel selesai dibuat maka menyimpan tabel dengan nama *Master_Loket*.

Column Name	Data Type	Allow Nulls
uid	varchar(50)	<input type="checkbox"/>
tanggal_masuk	varchar(50)	<input checked="" type="checkbox"/>
nama_barang	varchar(50)	<input checked="" type="checkbox"/>
kode_barang	varchar(50)	<input checked="" type="checkbox"/>
nama_perusahaan	varchar(50)	<input checked="" type="checkbox"/>
nama_pengirim	varchar(50)	<input checked="" type="checkbox"/>
jenis_sample	varchar(50)	<input checked="" type="checkbox"/>
merk	varchar(50)	<input checked="" type="checkbox"/>
volume	varchar(50)	<input checked="" type="checkbox"/>
bentuk	varchar(50)	<input checked="" type="checkbox"/>
parameter_uji	varchar(50)	<input checked="" type="checkbox"/>
biaya_uji	varchar(50)	<input checked="" type="checkbox"/>
metode_uji	varchar(50)	<input checked="" type="checkbox"/>
lokasi	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Gambar 4.9 Desain Tabel Master_Loket

C. Membuat Tabel User

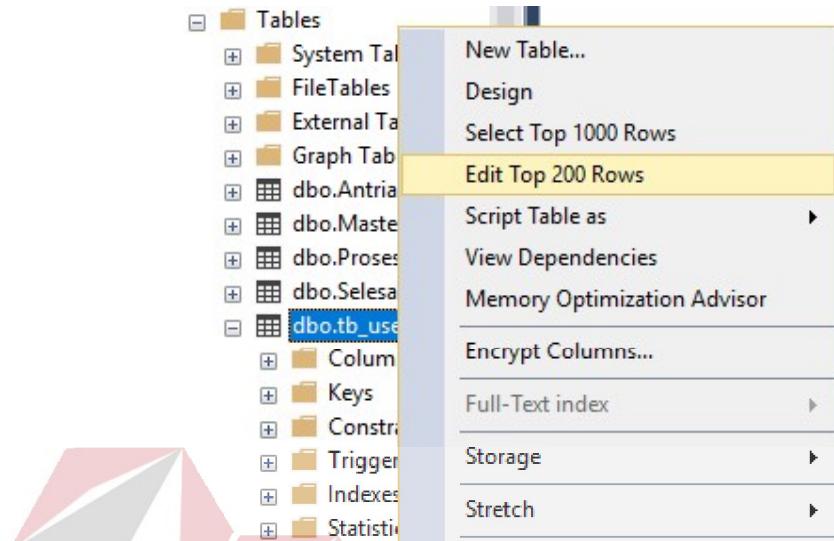
Tabel *user* digunakan untuk menyimpan data *user* beserta *password* yang digunakan untuk *login* pada aplikasi yang digunakan pada loket. Cara membuat tabel *user* sama halnya dengan membuat tabel *Master_Loket*, hanya saja terdapat perbedaan atribut yang tersimpan. Adapun atribut yang disimpan pada tabel *user* adalah *username* dan *password*, berikut adalah desain dari tabel *user* dapat dilihat pada gambar 4.10. Setelah tabel dibuat, menyimpan dengan nama *tb_user*.

Column Name	Data Type	Allow Nulls
username	varchar(50)	<input type="checkbox"/>
password	varchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Gambar 4.10 Desain Tabel tb_user

Setelah tabel berhasil dibuat maka mengisikan *username* dan *password* secara manual pada database dengan cara meng-klik kanan pada menu *Tables*

pada *Object Explorer* kemudian memilih *Edit Top 200 Rows* seperti pada gambar 4.11.



Gambar 4.11 Cara Mengisi tb_user Secara Manual

Kemudian mengisikan data *username* dan *password* pada tabel *user* yang nantinya digunakan untuk *login* pada aplikasi *desktop* seperti pada gambar 4.12.

	username	password
▶	admin	admin123
*	NULL	NULL

Gambar 4.12 Isi Data tb_user

D. Membuat Tabel Antrian Laboratorium Kimia

Tabel antrian laboratorium kimia digunakan untuk menyimpan data barang yang telah melakukan pembayaran dengan ditandai huruf “P” pada awal digit pertama kode barang sekaligus menginformasikan ke *database* bahwa status barang uji berada pada tahap antrian pengujian Laboratorium Kimia. Cara membuat tabel antrian Laboratorium Kimia sama halnya dengan membuat tabel

Master_Loket, hanya saja terdapat perbedaan atribut yang tersimpan. Adapun atribut yang disimpan pada tabel antrian laboratorium adalah UID, kode barang, dan lokasi barang uji. UID berlaku sebagai *primary key* dari tabel antrian Laboratorium Kimia. Desain dari tabel antrian Laboratorium Kimia dapat dilihat pada gambar 4.13. Setelah tabel dibuat simpanlah dengan nama Antrian_LKimia.

Column Name	Data Type	Allow Nulls
uid	varchar(20)	<input type="checkbox"/>
kode_barang	varchar(10)	<input type="checkbox"/>
lokasi	varchar(50)	<input type="checkbox"/>

Gambar 4.13 Desain Tabel Antrian_Lkimia

E. Membuat Tabel Proses Pengujian Laboratorium Kimia

Tabel proses pengujian laboratorium kimia digunakan untuk menyimpan data barang uji yang sedang melakukan proses pengujian. Selanjutnya perubahan status disimpan ke *database*, dengan keterangan barang uji berada pada tahap proses pengujian Laboratorium Kimia. Adapun atribut yang disimpan pada tabel proses pengujian laboratorium adalah UID, kode barang, dan lokasi barang uji. UID berlaku sebagai *primary key* dari tabel antrian Laboratorium Kimia. Desain dari tabel proses pengujian Laboratorium Kimia dapat dilihat pada gambar 4.14. Setelah tabel dibuat selanjutnya menyimpan dengan nama Proses_LKimia.

Column Name	Data Type	Allow Nulls
uid	varchar(20)	<input type="checkbox"/>
kode_barang	varchar(10)	<input type="checkbox"/>
lokasi	varchar(50)	<input type="checkbox"/>

Gambar 4.14 Desain Tabel Proses_LKimia

F. Membuat Tabel Selesai Proses Pengujian

Tabel selesai pengujian laboratorium kimia digunakan untuk menyimpan data barang uji yang telah selesai melakukan proses pengujian. Selanjutnya perubahan status disimpan ke *database*, dengan keterangan barang uji berada telah selesai melakukan proses pengujian di Laboratorium Kimia. Adapun atribut yang disimpan pada tabel selesai proses pengujian laboratorium adalah UID, kode barang, dan lokasi barang uji. UID berlaku sebagai *primary key* dari tabel antrian Laboratorium Kimia. Desain dari tabel selesai proses pengujian Laboratorium Kimia dapat dilihat pada gambar 4.15. Setelah tabel dibuat selanjutnya menyimpan dengan nama Selesai_LKimia.

Column Name	Data Type	Allow Nulls
uid	varchar(20)	<input type="checkbox"/>
kode_barang	varchar(10)	<input type="checkbox"/>
lokasi	varchar(50)	<input type="checkbox"/>

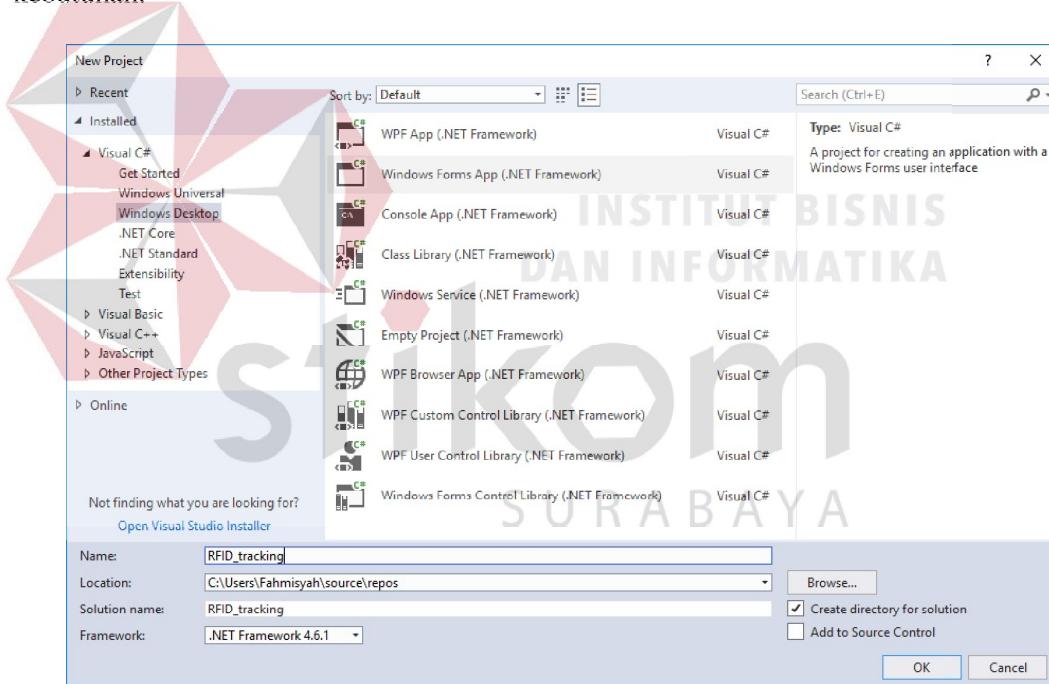
Gambar 4.15 Desain Tabel Selesai_LKimia

4.4.2 Perancangan dan Implementasi Aplikasi Pada Loket

Pada proses perancangan aplikasi penulis menggunakan Microsoft Visual Studio 2017 dengan memilih C# sebagai bahasa pemrograman yang digunakan, adapun langkah-langkah perancangan aplikasi pada loket dengan Microsoft Visual Studio 2017 akan dijelaskan sebagai berikut:

A. Membuat Project Baru

Untuk membuat *project* baru terlebih dahulu membuka aplikasi Microsoft Visual Studio 2017 lalu memilih menu *file* pada pojok kiri atas, dan memilih *tab new*, selanjutnya memilih opsi *new project*. Setelah memilih *menu new project* maka akan muncul tampilan seperti pada gambar 4.16. Selanjutnya memilih *windows desktop* pada menu bagian kiri dan pilih Windows Form App sebagai fitur yang digunakan untuk membuat aplikasi. Mengganti nama *project* sesuai yang diinginkan dan pilih direktori penyimpanan pada bagian *location* sesuai kebutuhan.



Gambar 4.16 Konfigurasi New Project Visual Studio 2017

B. Membuat Form Login

Tampilan awal setelah membuat project baru yaitu berupa form kosong, maka membuat beberapa *button*, *label* dan *text box* yang digunakan untuk mengisiusername dan password oleh user seperti pada gambar 4.17.



Gambar 4.17 Desain Aplikasi Loket

Selanjutnya setelah desain dibuat seperti pada gambar 4.17 maka konfigurasi nama button, label, dan text box sesuai dengan kebutuhan. Untuk selanjutnya *double click* pada bagian *form*, kemudian membuat *source code* seperti dibawah ini:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace ComputerToArduino
{
    public partial class login : Form
    {
        int i;
        public login()
        {
            InitializeComponent();
        }
        private void btnLogin_Click(object sender, EventArgs e)
        {
            string connectionString;
            SqlConnection cnn;
            connectionString = @"Data Source=LAPTOP-D2834IOL;Initial Catalog=RFID_Scanner;Integrated Security=True";
            cnn = new SqlConnection(connectionString);
```

```
cnn.Open();
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.SelectCommand = new SqlCommand(sql, cnn);

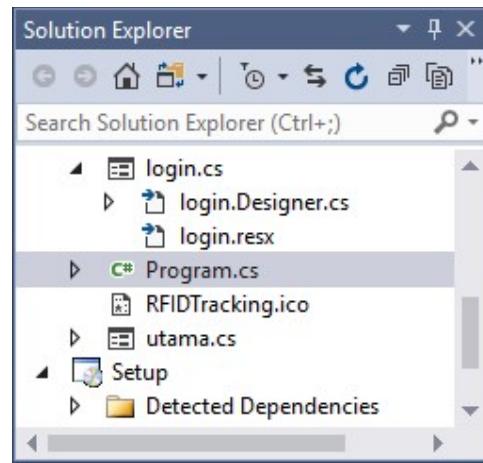
adapter.SelectCommand.ExecuteNonQuery();
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(command);
da.Fill(dt);
i = Convert.ToInt32(dt.Rows.Count.ToString());
if (i == 0)
{
    MessageBox.Show("Username atau password yang anda
masukan salah !");
}
else
{
    this.Hide();
    utama form = new utama();
    form.Show();
}
cnn.Close();
}

private void button1_Click_1(object sender, EventArgs e)
{
    this.Close();
    Application.Exit();
}

private void login_FormClosed(object sender, FormClosedEventArgs e)
{
    Application.Exit();
}

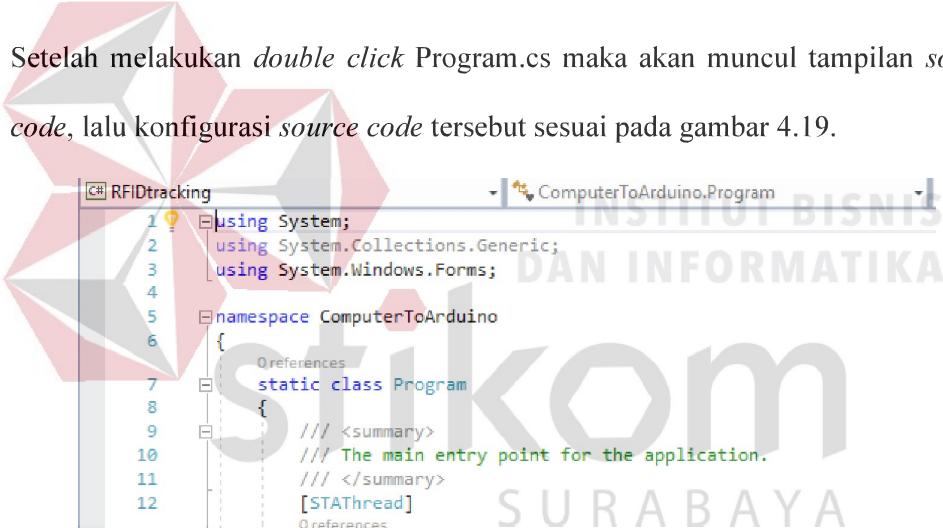
}
```

Setelah menuliskan source code diatas maka menyimpan form tersebut dengan nama login. Selanjutnya melakukan konfigurasi *form login* sebagai tampilan awal saat aplikasi di *running* dengan cara *double klik* Program.cs pada menu *Solution Explorer* yang terdapat pada sisi kanan atas Visual Studio 2017 seperti pada gambar 4.18.



Gambar 4.18 Tampilan *Solution Explorer*

Setelah melakukan *double click* *Program.cs* maka akan muncul tampilan *source code*, lalu konfigurasi *source code* tersebut sesuai pada gambar 4.19.



```

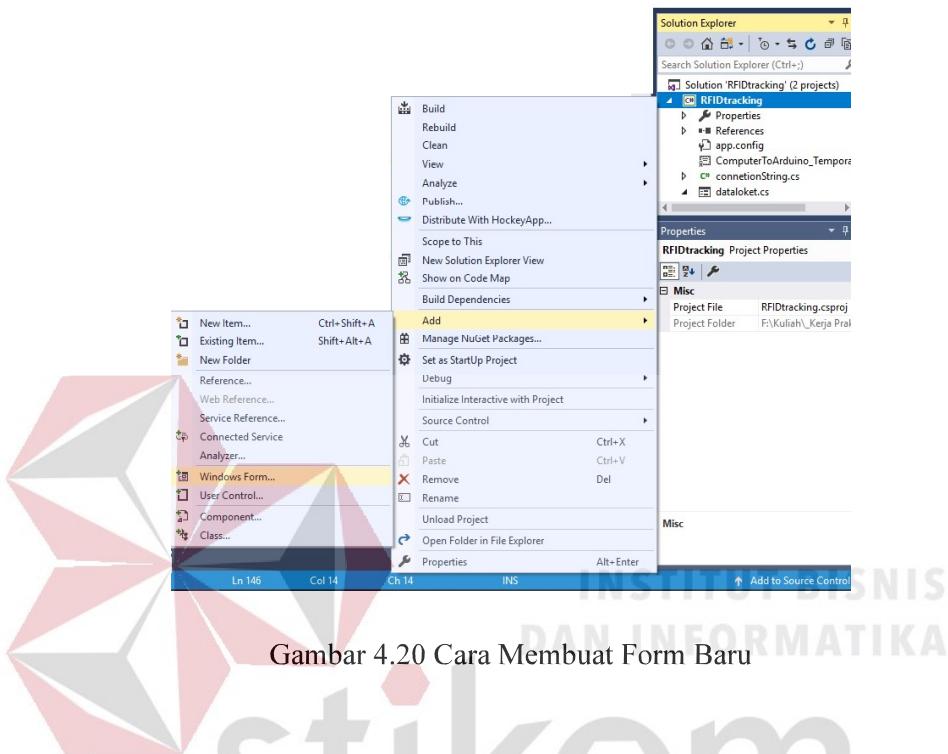
C# RFIDtracking ComputerToArduino.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Windows.Forms;
4
5  namespace ComputerToArduino
6  {
7      static class Program
8      {
9          /// <summary>
10         /// The main entry point for the application.
11         /// </summary>
12         [STAThread]
13         static void Main()
14         {
15             Application.EnableVisualStyles();
16             Application.SetCompatibleTextRenderingDefault(false);
17             Application.Run(new login());
18         }
19     }
20 }
21 
```

Gambar 4.19 Tampilan *Source Code* Program.Cs

C. Membuat Form Utama Loket

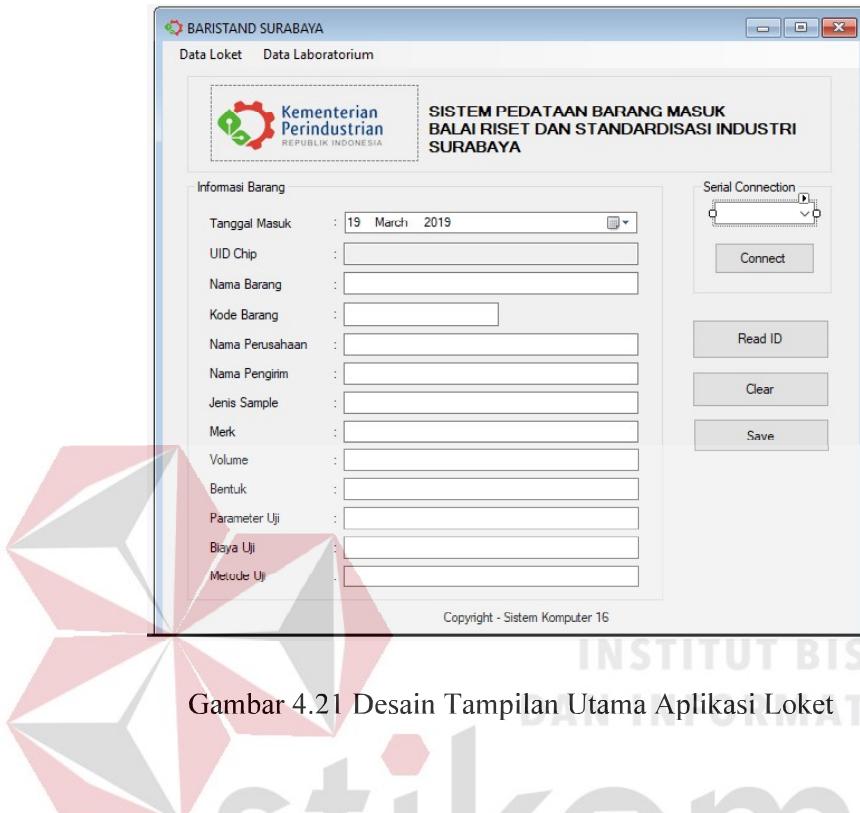
Form utama loket digunakan sebagai tampilan awal setelah *user login* pada aplikasi. Cara membuat *form* tambahan yaitu dengan meng-klik kanan pada *item*

project yang terdapat pada menu *solution explorer*, lalu memilih opsi *add* kemudian memilih fitur *windows form* seperti pada gambar 4.20. Kemudian memberikan nama “utama” pada windows form baru tersebut.



Setelah membuat form utama maka tampilan pertama form tersebut adalah jendela kosong. Selanjutnya memasangkan beberapa *text box* dan *label* untuk mengisikan data yang ingin disimpan pada *database*. menambahkan *combo box* untuk melihat *port serial* aktif yang digunakan untuk komunikasi data serta tambahkan *button connect* untuk mengaktifkan/ menggunakan *port* yang dipilih sesuai yang terdaftar pada *combo box*. Menambahkan *button read id* yang digunakan untuk membaca data UID kartu RFID. Menambahkan *button clear* untuk mengosongkan semua data yang tertulis pada *text box* dan juga *button save* untuk menyimpan data kedalam *database* serta menyimpan kode barang kedalam

kartu RFID dan juga menambahkan *menuStrip* untuk membuat model menu seperti yang terdapat pada gambar 4.21.



Gambar 4.21 Desain Tampilan Utama Aplikasi Loket

Setelah melakukan konfigurasi desain *form* seperti pada gambar 4.21 maka selanjutnya mengisikan *source code* pada form tersebut seperti dibawah ini:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using MySql.Data.MySqlClient;
using System.Data.SqlClient;

namespace ComputerToArduino
{
    public partial class utama : Form
    {
        String reading;
        bool isConnected = false;
        String[] ports;
        SerialPort port;
        bool bekas=false;
```

```
public static bool dataloketform = false;
public static bool labkimiaform = false;
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];
        }
    }
}
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;
    }
}
private void button1_Click(object sender, EventArgs e)
{
    if (!isConnected)
    {
        connectToArduino();
    } else
    {
        disconnectFromArduino();
    }
}
void getAvailableComPorts()
{
    ports = SerialPort.GetPortNames();
}
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();
}
private void disconnectFromArduino()
{
```

```
        isConnected = false;
        port.WriteLine(" ");
        port.Close();
        button1.Text = "Connect";
        disableControls();
        resetDefaults();
    }
    private void enableControls()
    {
        btnDB.Enabled = true;
        btnClear.Enabled = true;
        textBox1.Enabled = true;
        groupBox3.Enabled = true;
        button2.Enabled = true;
    }
    private void disableControls()
    {
        btnClear.Enabled = false;
        btnDB.Enabled = false;
        textBox1.Enabled = false;
        groupBox3.Enabled = false;
        button2.Enabled = false;
    }
    private void resetDefaults()
    {
        textBox1.Text = "";
    }
    private void btnClear_Click(object sender, EventArgs e)
    {
        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();
    }
    private void btnDB_Click(object sender, EventArgs e)
    {
        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.");
        }
        else {
            port.WriteLine(textBox1.Text);
            string connetionString;
            SqlConnection cnn;
```

```
        connetionString = @"Data Source=LAPTOP-D2834IOL;Initial
Catalog=RFID_Scanner;Integrated Security=True";
        cnn = new SqlConnection(connetionString);
        cnn.Open();
        SqlCommand command;
        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,loka
si) values ('" + txtUid.Text + "','" + datepicker1.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);
        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 !");
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        command.Dispose();
        cnn.Close();
    }
}
private void button2_Click(object sender, EventArgs e)
{
    if ((txtUid.Text == "")&&(reading==""))
    {
        reading = port.ReadLine();
        SetText(reading);
    }
    else
    {
```

```
        txtUid.Text = "";
        reading = port.ReadLine();
        SetText(reading);
    }
}
private void dataLoketToolStripMenuItem_Click(object sender,
EventArgs e)
{
    dataloket lkt = new dataloket();
    if (!dataloketform)
    {
        dataloketform = true;
        lkt.Show();
    }
    else
    {
        MessageBox.Show("Data Loket sudah terbuka.", "Attention !",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}
private void labKimiaToolStripMenuItem_Click(object sender,
EventArgs e)
{
    labkimia kimia = new labkimia();
    if (!labkimiaform)
    {
        labkimiaform = true;
        kimia.Show();
    }
}
private void exitToolStripMenuItem_Click(object sender,
EventArgs e)
{
    this.Close();
    login lg = new login();
    lg.Show();
}
private void utama_FormClosed(object sender, FormClosedEventArgs e)
{
    login lg = new login();
    lg.Show();
}
private void utama_FormClosing(object sender,
FormClosingEventArgs e)
{
    if (dataloketform||labkimiaform)
    {
        MessageBox.Show("Please, close all window", "Attention !",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        e.Cancel = true;
    }
}
}
```

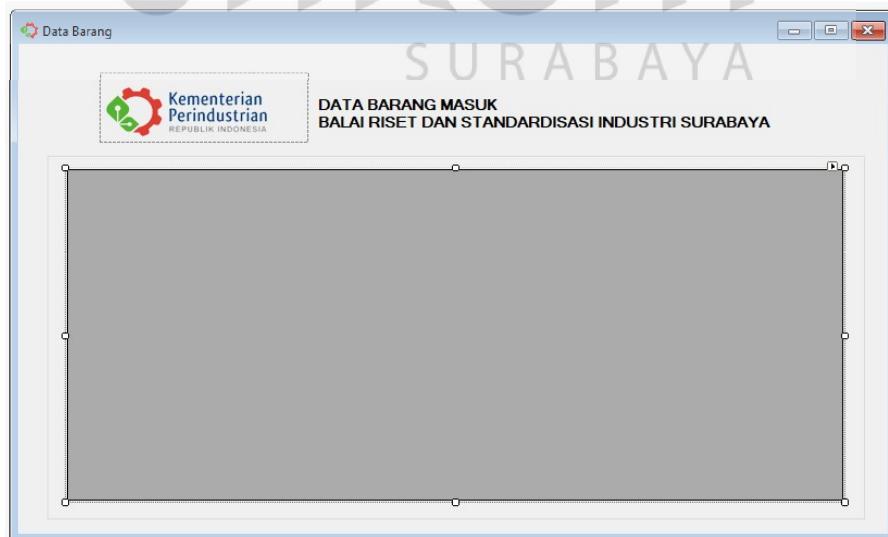
Pada perancangan aplikasi loket penulis menggunakan kode berikut untuk mengambil data UID dari scanner agar dapat ditampilkan melalui aplikasi:

```
reading = port.ReadLine();
```

Kode tersebut digunakan agar data yang dituliskan pada aplikasi berupa data tetap satu baris dan tidak ada tambahan data input yang lainnya.

D. Membuat Form Data Loket

Pada form ini data yang pernah melakukan registrasi diloket akan ditampilkan dengan cara dikoneksikan dengan *database sql server*. Cara membuat form data loket sama saja dengan membuat *windows form* pada *form* utama hanya saja pada form ini data dari database akan ditampilkan melalui *data grid view* yang ada pada Visual Studio 2017. Selanjutnya simpan nama *form* baru dengan nama dataloket. Desain dari tampilan *form* dataloket seperti pada gambar 4.22.



Gambar 4.22 Desain Tampilan *Form* Dataloket

Setelah membuat desain data loket seperti pada gambar 4.22, agar aplikasi dapat menampilkan data barang pada *data grid view* maka konfigurasi *list* program sebagai berikut:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
namespace ComputerToArduino
{
    public partial class dataloket : Form
    {
        SqlConnection connection = new SqlConnection("Data
Source=LAPTOP-D2834IOL;Initial Catalog=RFID_Scanner;Integrated
Security=True");
        public dataloket()
        {
            InitializeComponent();
        }
        private void groupBox1_Enter(object sender, EventArgs e)
        {
            string selectQueryloket = "SELECT * from Master_Loket";
            DataTable table_loket = new DataTable();
            SqlDataAdapter adapter_loket = new
SqlDataAdapter(selectQueryloket, connection);
            adapter_loket.Fill(table_loket);
            dataGridView1.DataSource = table_loket;
        }
        private void dataloket_FormClosed(object sender,
FormClosedEventArgs e)
        {
            utama.dataloketform = false;
        }
    }
}
```

Dari list program diatas terdapat *source code* yang digunakan untuk mengintegrasikan aplikasi dengan *database*, *source code* yang digunakan untuk mengintegrasikan aplikasi dengan database dapat dilihat pada gambar 4.23.

```

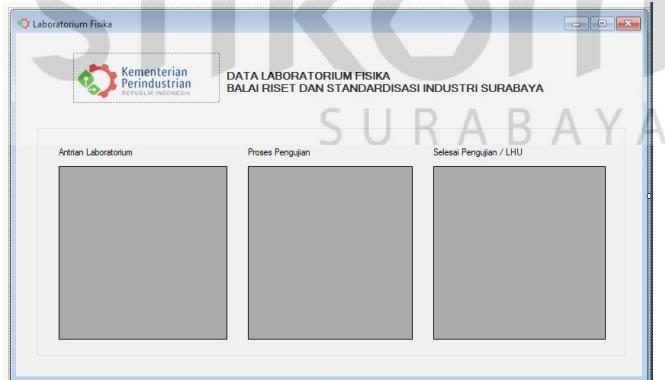
SqlConnection connection = new SqlConnection
    ("Data Source=LAPTOP-D2834IOL;Initial Catalog=RFID_Scanner;Integrated Security=True");
1 reference
public dataloket()
{
    InitializeComponent();
}

```

Gambar 4.23 *Source Code Integrasi Ke Database*

E. Membuat Form Lab Kimia

Pada form ini data yang sedang melakukan proses pengujian dapat ditampilkan pada tiga *data grid view*, yaitu data antrian laboratorium kimia, data proses pengujian dan data selesai pengujian. Data dikoneksikan dengan *database sql server*. Cara membuat form data loket sama saja dengan membuat *windows form* pada *form* utama hanya saja pada form ini data dari database akan ditampilkan melalui *data grid view* yang ada pada Visual Studio 2017. Selanjutnya simpan nama *form* baru dengan nama labkimia. Desain dari tampilan *form* dataloket seperti pada gambar 4.24.



Gambar 4.24 Desain *Form* labkimia

Setelah membuat desain labkimia seperti pada gambar 4.24, agar aplikasi dapat menampilkan data yang sedang melakukan proses pengujian pada *data grid view* maka konfigurasi *list* program sebagai berikut:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
namespace ComputerToArduino
{
    public partial class labkimia : Form
    {
        //MySqlConnection connection = new
        MySqlConnection("datasource=localhost;port=3306;Initial
Catalog='dbscan';username=root;password=");
        SqlConnection cnn = new SqlConnection ("Data Source=LAPTOP-
D2834IOL;Initial Catalog=RFID_Scanner;Integrated Security=True");
        public labkimia()
        {
            InitializeComponent();
        }
        private void groupBox1_Enter(object sender, EventArgs e)
        {
            cnn.Open();
            string selectQueryantri = "SELECT kode_barang, nama_barang
FROM Master_Loket WHERE EXISTS (SELECT uid from Antrian_LKimia WHERE
Master_Loket.uid = Antrian_LKimia.uid)";
            DataTable table_antri = new DataTable();
            SqlDataAdapter adapter_antri = new
SqlDataAdapter(selectQueryantri, cnn);
            adapter_antri.Fill(table_antri);
            dg_antrikimia.DataSource = table_antri;
            string selectQueryproses = "SELECT kode_barang, nama_barang
FROM Master_Loket WHERE EXISTS (SELECT uid from Proses_LKimia WHERE
Master_Loket.uid = Proses_LKimia.uid)";
            DataTable table_proses = new DataTable();
            SqlDataAdapter adapter_proses = new
SqlDataAdapter(selectQueryproses, cnn);
            adapter_proses.Fill(table_proses);
            dg_proseskimia.DataSource = table_proses;
            string selectQueryselesai = "SELECT kode_barang, nama_barang
FROM Master_Loket WHERE EXISTS (SELECT uid from Selesai_LKimia WHERE
Master_Loket.uid = Selesai_LKimia.uid)";
            DataTable table_selesai = new DataTable();
            SqlDataAdapter adapter_selesai = new
SqlDataAdapter(selectQueryselesai, cnn);
            adapter_selesai.Fill(table_selesai);
            dg_selesaikimia.DataSource = table_selesai;
        }
    }

```

```
        private void labfisika_FormClosed(object sender,  
FormClosedEventArgs e)  
        {  
            utama.labkimiaform = false;  
        }  
    }  
}
```

4.4.3 Implementasi Alat *Scanner*

Pada implementasi alat *scanner* terdapat tiga proses perancangan, yaitu proses perancangan alat *scanner* pada loket untuk pendataan barang masuk, perancangan alat scanner laboratorium kimia untuk melakukan pendataan barang yang sedang melakukan proses pengujian untuk tahap antrian, proses pengujian dan selesai, yang ketiga adalah perancangan alat scanner yang digunakan untuk mengatur ulang kartu RFID agar dapat digunakan kembali.

A. Scanner Loket

Scanner loket dirancang dengan menggunakan arduino nano sebagai *microcontroller* yang dilengkapi dengan modul RFID RC 522 untuk membaca UID kartu RFID dan menuliskan identitas pada barang berupa kode barang yang awali dengan kode “T” yang menandakan barang uji belum melakukan proses apapun dan baru saja melalui tahap registrasi. Kartu RFID yang digunakan dalam *project* ini berupa *sticker* RFID yang dapat menyimpan data maksimal 16 bit. Skema rangkaian PCB *scanner* loket dapat dilihat pada gambar 4.25.

B. *Scanner Laboratorium*

Scanner laboratorium dirancang dengan menggunakan NodeMCU sebagai *microcontroller* yang dilengkapi dengan modul RFID RC 522 untuk membaca

UID kartu RFID dan merubah identitas pada barang berupa kode barang yang awali dengan kode “T” menjadi “P” yang menandakan barang uji telah melakukan proses pembayaran dan siap untuk melakukan proses pengujian. *Scanner* laboratorium juga berfungsi untuk merubah status barang uji pada *database*. Skema rangkaian PCB *scanner* laboratorium dapat dilihat pada gambar 4.26.

C. *Scanner* Reset

Scanner reset digunakan untuk melakukan *reset* pada kartu RFID agar dapat digunakan kembali sekaligus menghapus data yang tersimpan pada *database* yang menandakan barang uji telah selesai menjalani proses pengujian di Balai Riset dan Standardisasi Industri Surabaya. *Scannerreset* menggunakan NodeMCU sebagai *microcontroller* yang memungkinkan *update database* dengan komunikasi *wireless*. Skema rangkaian PCB *scanner* reset dapat dilihat pada gambar 4.25

4.4.4 Koneksi *Database* Melalui PHP

Pengiriman data yang dilakukan oleh *scanner* pada laboratorium dan pada *scanner reset* dilakukan melalui koneksi *wireless* menggunakan *hotsport / Wifi* untuk melakukan *update* data pada *database* sesuai alur dari proses pengujian. Pengiriman data tidak dapat dilakukan secara langsung karena perlu adanya penjembatan antara *database* dengan *scanner*, maka php difungsikan sebagai penghubung *database* dengan *scanner* melalui koneksi *wireless* dengan memanfaatkan modul ESP8266 yang terdapat pada *microcontroller* NodeMCU V3.

A. Koneksi PHP dengan *Database Laboratorium*

Koneksi PHP dengan *database* laboratorium digunakan agar data yang discan oleh *scanner* pada laboratorium dapat disimpan dan di *update* oleh *database*, sehingga *user* dapat memantau alur dan status barang uji melalui aplikasi yang telah disediakan di loket. PHP digunakan sebagai penjembatan untuk mengintegrasikan antara alat *scanner* dengan *database* melalui koneksi *wireless* dengan *hotspot/Wifi* menggunakan web *server* lokal, yaitu XAMPP, sehingga alur dari proses *scanning* di laboratorium yaitu dengan mengirimkan data menggunakan komunikasi *wireless* yang dijembatani oleh PHP agar bisa terintegrasi dengan database SQL Server 2017. Adapun *source code* PHP yang digunakan adalah sebagai berikut:

```
<?php
$uid = $_GET['uid'];
$kode_barang = $_GET['kode_barang'];
$lokasi = $_GET['lokasi'];
$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>";
}
$sqlAntrian_LFisika = "SELECT * FROM Antrian_LFisika WHERE uid =
'$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 );
$sqlProses_LFisika = "SELECT * FROM Proses_LFisika WHERE uid = '$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( $stmtSelesai_LFisika );
if ($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);
    $hapustri = "DELETE FROM Antrian_LFisika WHERE
Antrian_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn,$hapustri);
}
echo $row_count_Antrian_LFisika. "<br>";
echo $row_count_Proses_LFisika ;
?>

```

B. Koneksi PHP dengan *Database Reset*

Koneksi PHP dengan *database reset* digunakan sebagai penjembatan antara data yang dikirimkan oleh *scanner reset* kepada *database reset*. Fungsi *database reset* hanya digunakan untuk sementara waktu, dalam artian data yang dikirimkan oleh *scanner reset* hanya digunakan sebagai pembanding dengan data yang tersimpan pada *database reset*. Dengan anggapan apabila data dari kartu RFID yang dikirimkan oleh *scanner reset* terdapat pada penyimpanan *database*

berarti kartu RFID masih belum dapat digunakan kembali, maka dengan melakukan *reset* data pada kartu RFID, data yang disimpan pada kartu beserta data yang tersimpan di *database* akan dihapus, sehingga kartu RFID dapat digunakan kembali sebagai identitas barang uji yang berbeda. Adapun *source code* PHP yang digunakan adalah sebagai berikut:

```
<?php
$uid = $_GET['uid'];
$kode_barang = $_GET['kode_barang'];
$lokasi = $_GET['lokasi'];
$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>";
}
$sqlAntrian_LFisika = "SELECT * FROM Antrian_LFisika WHERE uid =
'$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 );
$sqlProses_LFisika = "SELECT * FROM Proses_LFisika WHERE uid = '$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 );
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);
    $hapustri = "DELETE FROM Antrian_LFisika WHERE
Antrian_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn,$hapustri);
}
if(($row_count_Antrian_LFisika == 0) && ($row_count_Selesai_LFisika !=
0))
{
    $reset = "INSERT INTO Reset_LFisika (uid, kode_barang, lokasi)
VALUES('$uid', '$kode_barang', '$lokasi')";
    $result = sqlsrv_query($conn, $selesai);
    $hapussel = "DELETE FROM Selesai_LFisika WHERE
Selesai_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn, $hapussel);
    $hapuslok = "DELETE FROM Master_Loket WHERE
Master_Loket.uid='$uid'";
    $result = sqlsrv_query($conn,$hapuslok);
    $hapus = "DELETE FROM Antrian_LFisika WHERE
Antrian_LFisika.uid='$uid'";
    $result = sqlsrv_query($conn,$hapus);
}
echo $row_count_Antrian_LFisika. "<br>";
echo $row_count_Proses_LFisika ;
?>

```

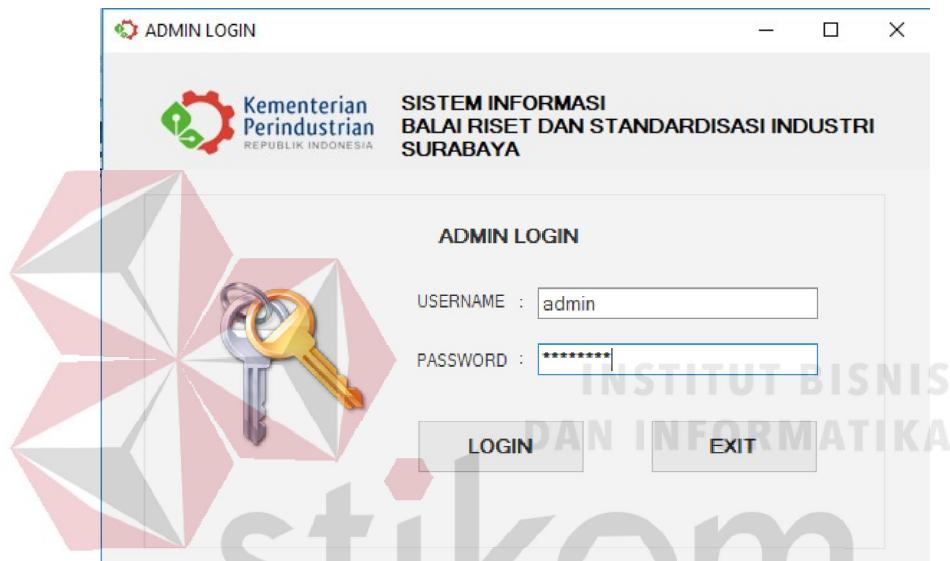
4.5 Hasil dan Pembahasan Alat

Pada tahap ini dilakukan pembahasan secara detail mengenai hasil yang didapatkan dari implementasi yang sudah dilakukan. Terdapat dua pengujian yang akan dilakukan untuk mendapatkan hasil yang diperlukan, yaitu pengujian dari implementasi sistem RFID untuk *tracking* barang pada Loket Balai Riset dan Standardisasi Industri Surabaya.

4.5.1 Pengujian Aplikasi dan *Scanner* Loket

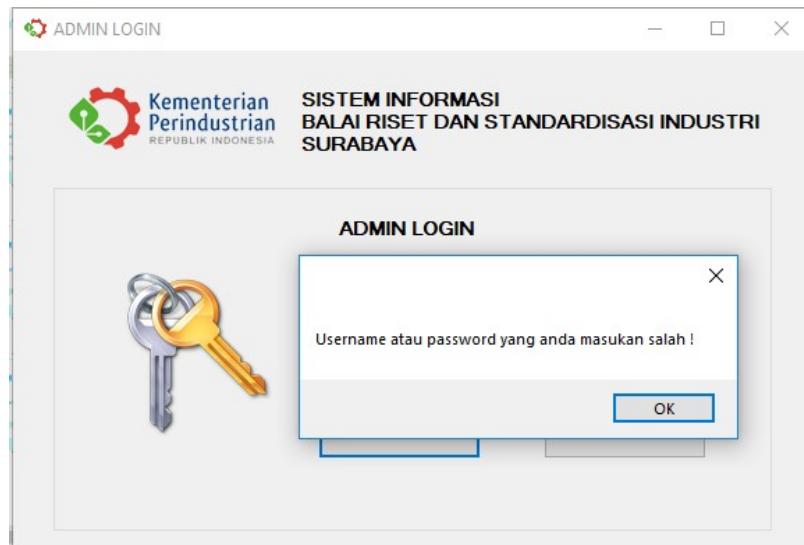
Pengujian Aplikasi dan *Scanner* Loket dilakukan untuk mengetahui efisiensi dan keberhasilan dari perancangan sistem *tracking* barang menggunakan RFID. Berikut adalah pengujian aplikasi dan *scanner* yang ditempatkan di loket.

A. Pengujian Login Aplikasi



Gambar 4.25 Form Login

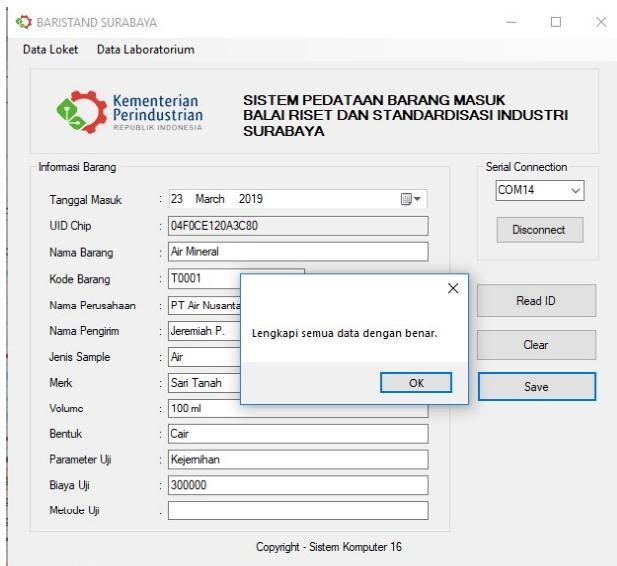
Pada pengujian *form login*, user dapat memasukkan data user sesuai data yang tersimpan pada *database user*. Apabila terjadi kesalahan memasukkan user dan password maka akan muncul peringatan seperti pada gambar 4.26



Gambar 4.26 Kesalahan Memasukkan Username Atau Password

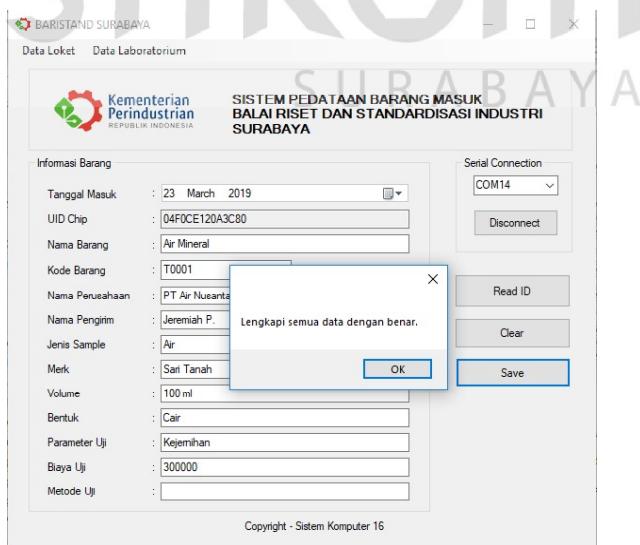
B. Pengujian Registrasi Barang Uji

Pengujian registrasi barang uji dilakukan untuk menganalisis apakah kode barang yang telah disisipkan pada kartu RFID dapat tersimpan pada kartu RFID dan tersimpan pada *database* sistem. Sebelum memasukkan data terlebih dahulu mengatur konfigurasi *port* serial yang digunakan untuk melakukan komunikasi serial dari *scanner* ke aplikasi, kemudian menuliskan data sesuai *form* pada aplikasi yang dapat dilihat pada gambar 4.27.



Gambar 4.27 Pengisian Data Registrasi

Apabila pengisian data salah ataupun kurang lengkap akan muncul notifikasi untuk melengkapi data yang diisikan seperti pada gambar 4.28. Analisis keberhasilan penyimpanan data barang uji pada *database* dapat dilihat pada gambar 4.29.



Gambar 4.28 Kesalahan Pengisian Data

uid	tanggal_masuk	nama_barang	kode_barang	nama_perush...	nama_pengirim	jenis_sample	merk	volume	bentuk	parameter_uji	biaya_uji	metode_uji	lokasi
04F0CE120A3C80	23 March 2019	Air Mineral	T0001	PT Air Nusantara	Jeremiah P.	Air	Sari Tanah	100 ml	Cair	Kejernihan	300000	Mineral Air	loket
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 4.29 Data Yang Tersimpan Pada Tabel Master_Loket

C. Memberikan Kode Barang Melalui *Scanner Loket*

Pemberian kode barang pada kartu RFID dilakukan dengan menempelkan kartu RFID pada *scanner* selama proses *registrasi* barang uji, sehingga kode barang dapat terdimpam pada kartu RFID dan data keseluruhan dapat tersimpan pada *database*. Gambar *scanner* loket dapat dilihat pada gambar

4.30.

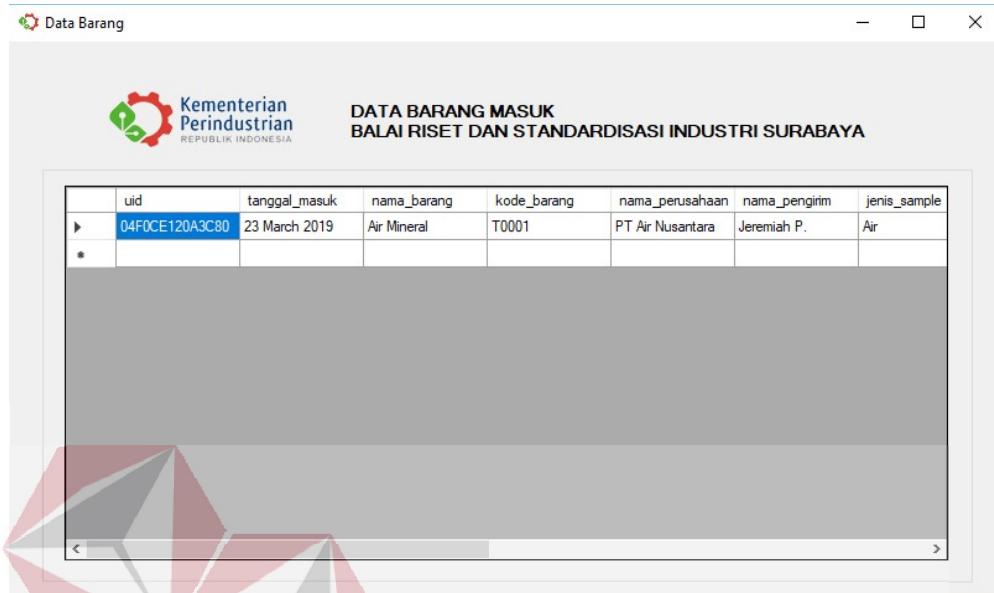


Gambar 4.30 *Scanner Loket*

D. Memantau Data Barang Uji Keseluruhan

Pemantauan barang uji secara keseluruhan dapat dilakukan melalui aplikasi loket pada menu data loket, sehingga semua data barang uji yang pernah melakukan *registrasi* dan belum menyelesaikan proses pengujian dapat dipantau

dan di data melalui menu ini. Tampilan menu data loket dapat dilihat pada gambar 4.31.



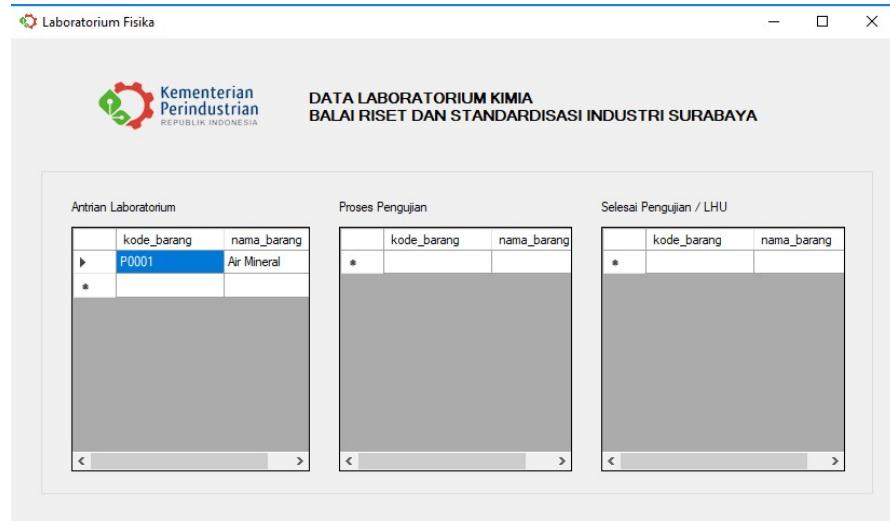
The screenshot shows a Windows application window titled 'Data Barang'. At the top, there is a logo for 'Kementerian Perindustrian REPUBLIK INDONESIA' and the text 'DATA BARANG MASUK BALAI RISET DAN STANDARDISASI INDUSTRI SURABAYA'. Below this is a table with the following columns: uid, tanggal_masuk, nama_barang, kode_barang, nama_perusahaan, nama_pengirim, and jenis_sample. One row is visible in the table, containing: 04F0CE120A3C80, 23 March 2019, Air Mineral, T0001, PT Air Nusantara, Jeremiah P., and Air. The table has a light gray background and black borders for the cells.

	uid	tanggal_masuk	nama_barang	kode_barang	nama_perusahaan	nama_pengirim	jenis_sample
▶	04F0CE120A3C80	23 March 2019	Air Mineral	T0001	PT Air Nusantara	Jeremiah P.	Air
*							

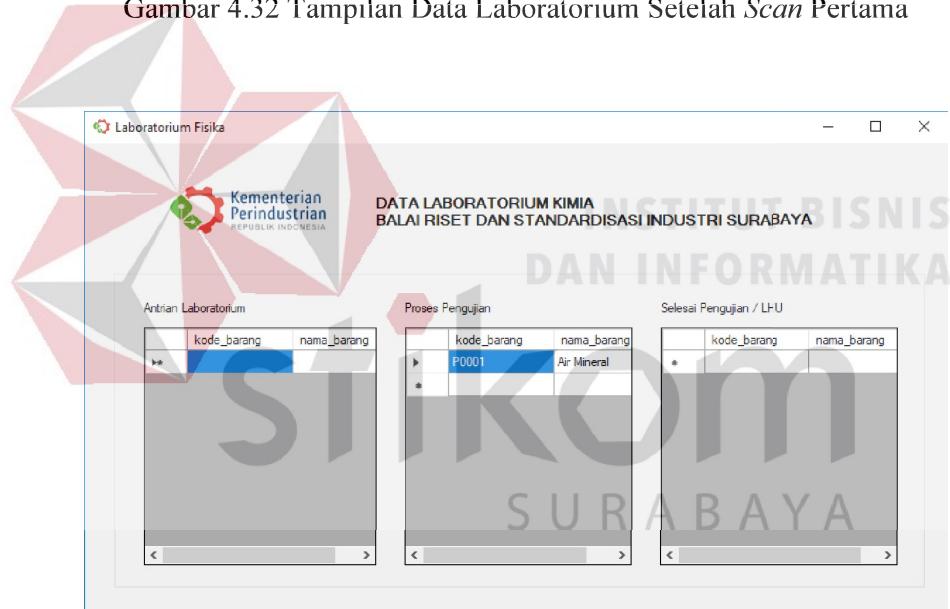
Gambar 4.31 Tampilan Menu Data Loket

E. Memantau Mekanisme Pengujian Barang Uji

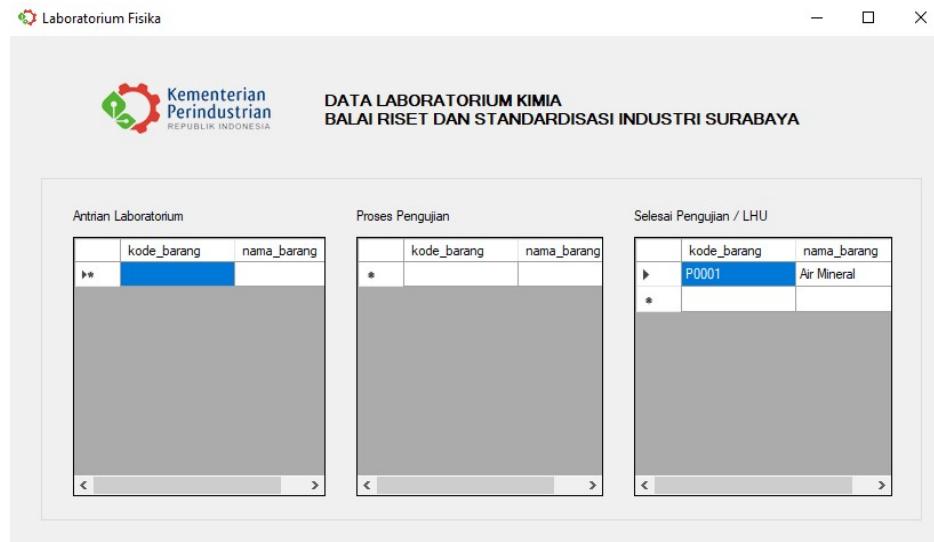
Pemantauan mekanisme pengujian dilakukan untuk mengetahui status terkini dari setiap barang uji. Pemantauan dapat dilakukan melalui aplikasi web pada menu data laboratorium kimia. Terdapat tiga poin utama pada tampilan menu data laboratorium, yaitu tampilan untuk data barang uji yang masih mengantre di antrian laboratorium, barang uji yang sedang melakukan proses, dan barang uji yang telah selesai melakukan pengujian. Mekanisme kerjanya yaitu dengan menambahkan data barang uji pada status terkini dan menghapus status sebelumnya, sehingga tidak terjadi penumpukan data yang sama. Tampilan menu data laboratorium kimia pada aplikasi loket dapat dilihat pada gambar 4.32, gambar 4.33, dan gambar 4.34.



Gambar 4.32 Tampilan Data Laboratorium Setelah *Scan* Pertama



Gambar 4.33 Tampilan Data Laboratorium Setelah *Scan* Kedua



Gambar 4.34 Tampilan Data Laboratorium Setelah *Scan* Ketiga

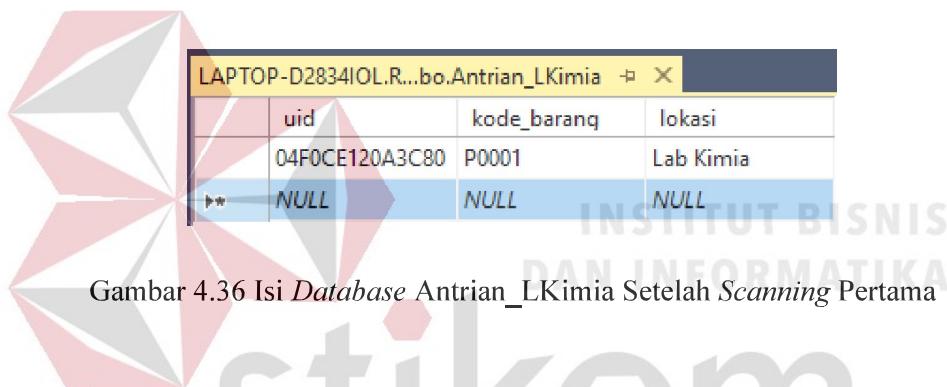
Dari beberapa pengujian yang telah dilakukan, maka dapat disimpulkan bahwa aplikasi dan *scanner* pada loket dapat berjalan dengan lancar dan data yang dikirimkan dapat tersimpan pada *database* maupun kartu RFID.

4.5.2 Pengujian *Scanner* Laboratorium Kimia

Pengujian *scanner* Laboratorium Kimia dilakukan untuk menganalisis keberhasilan pengiriman data melalui komunikasi *wireless* untuk *update* status dari proses pengiriman data dari *scanner* menuju *database*. Tahap yang dilakukan untuk *scanner* yaitu menguji koneksi antara *scanner* dengan *database*, mengganti digit pertama kode barang uji dari “T” ke “P”, mengirimkan status terbaru dari *scanner* ke *database*. Desain *scanner* laboratorium dapat dilihat pada gambar 4.35

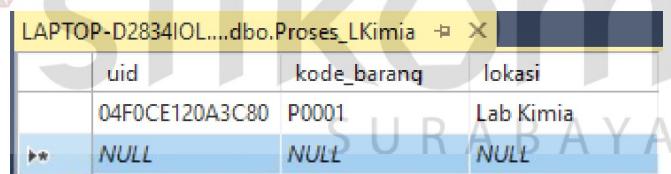


Gambar 4.35 Desain Scanner Laboratorium Kimia



	uid	kode_barang	lokasi
▶*	04F0CE120A3C80	P0001	Lab Kimia
▶*	NULL	NULL	NULL

Gambar 4.36 Isi Database Antrian_LKimia Setelah Scanning Pertama



	uid	kode_barang	lokasi
▶*	04F0CE120A3C80	P0001	Lab Kimia
▶*	NULL	NULL	NULL

Gambar 4.37 Isi Database Proses_LKimia Setelah Scanning Kedua



	uid	kode_barang	lokasi
▶	04F0CE120A3C80	P0001	Lab Kimia
*	NULL	NULL	NULL

Gambar 4.38 Isi Database Selesai_LKimia Setelah Scanning Kedua

4.5.3 Pengujian Reset Kartu RFID

Pengujian *scanner reset* dilakukan untuk menganalisis keberhasilan *scanner* dalam menghapus data pada kartu RFID dan *database*. Cara melakukan *reset* yaitu dengan menempelkan kartu RFID dan menekan tombol yang terdapat pada *scanner reset*. Alur proses reset yaitu dengan mencari koneksi *wireless* / koneksi *server*, jika koneksi telah ditemukan maka *scanner reset* siap digunakan.



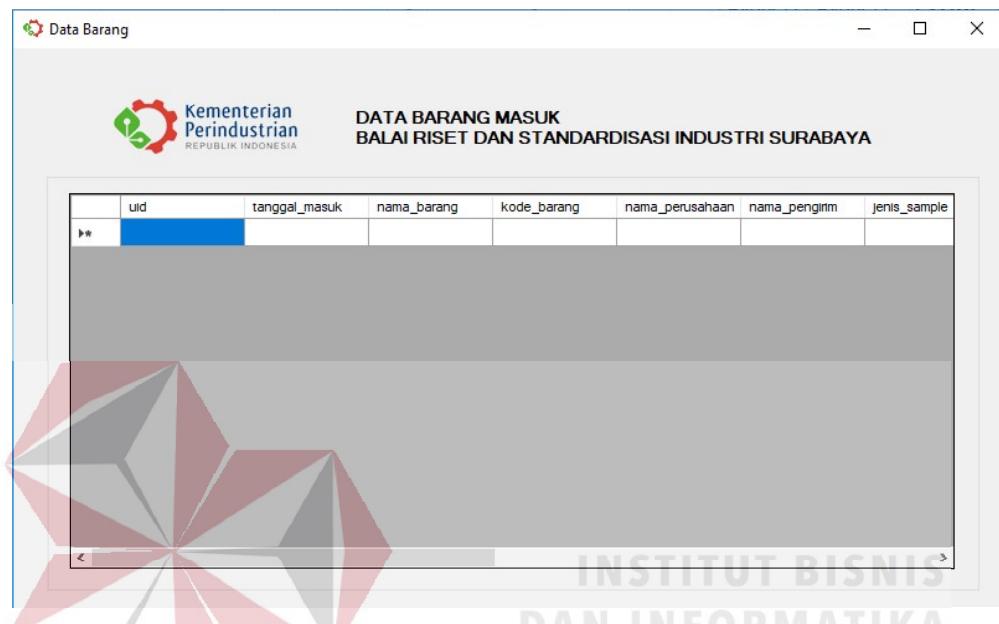
Gambar 4.39 Mencari Koneksi Pada Scanner Reset



Gambar 4.40 Mendapatkan Koneksi Wireless

uid	tanggal_masuk	nama_barang	kode_barang	nama_perusaha...	nama_pengirim	jenis_sample	merk	volume	bentuk	parameter_uji	biaya_uji
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Gambar 4.41 Berhasil Menghapus Data Barang Uji Pada Database



Gambar 4.42 Tidak Bisa Menampilkan Data Karena Telah Terhapus



BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil Implementasi RFID (Radio Frequency Identifier) untuk Tracking Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi pada loket dapat berjalan dengan baik yang dibuktikan dengan pengujian aplikasi untuk menampilkan data dan status pengujian.
2. *Scanner* loket dapat berjalan dengan baik dibuktikan dengan *scanner* dapat memasukan kode barang pada kartu RFID, dan membaca data kode barang yang tersimpan.
3. *Scanner* pada laboratorium kimia dapat berjalan dengan baik yang dibuktikan dengan dapat merubah status antrian, proses dan selesai pada *database* yang ditampilkan melalui aplikasi.
4. Karena kartu RFID yang digunakan berupa *sticker*, maka yang dapat tersimpan pada kartu RFID adalah 16 bit.
5. Untuk kelancaran konektivitas *wireless* maka menyediakan *Wifi* atau *hotspot* yang fokus digunakan untuk *scanner*.
6. Perlu adanya penjembanan berupa program PHP untuk mengatur komunikasi *scanner* dengan *database* melalui komunikasi *wireless*.
7. *Update* data barang uji pada laboratorium dilakukan dengan mekanisme menghapus *log* sebelumnya untuk menghindari data ganda.

8. Implementasi dari sistem ini masih menggunakan penyimpanan *localhost* sehingga diperlukan pengembangan lebih lanjut agar dapat diintegrasikan dengan sistem sebenarnya.

Berdasarkan hasil pengujian yang dilakukan maka dapat disimpulkan bahwa Implementasi RFID (Radio Frequency Identifier) untuk Tracking Barang pada Loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya dapat berjalan dengan baik dan dapat dimanfaatkan sebagaimana mestinya.

5.2 Saran

Saran untuk pengembangan RFID (*Radio Frequency Identifier*) untuk *tracking* barang pada loket dan Laboratorium Kimia di Balai Riset dan Standardisasi Industri Surabaya yaitu:

1. Untuk dapat dikembangkan dengan mengkombinasikan berbagai macam metode perancangan sistem *scanner* maupun sistem komunikasi data agar kapasitas penyimpanan maupun alur komunikasi data dapat diimplementasikan dalam *range* yang luas dan dalam jumlah besar.
2. Pergunakan *microcontroller* yang memiliki kecepatan *transfer* data yang tinggi dan memiliki memori penyimpanan yang besar misalnya Raspberry Pi yang dapat digunakan sebagai *microcontroller*, agar penyimpanan data tidak terbatas pada beberapa data saja.
3. Disisi lain sistem ini juga dapat dikembangkan dengan kombinasi desain *database*, sehingga penyimpanan data dapat lebih terstruktur dengan baik.

DAFTAR PUSTAKA

- Ahson, S., & Mohammed, I. (2008). *RFID Handbook Applications, Technology, Security and Privacy*. United States of America: CRC Press.
- Alan. (2017). *Alan Creative*. Dipetik Februari 29, 2019, dari Pengertian dan Fungsi Pemrograman PHP: <https://alan.co.id/pengertian-dan-fungsi-pemrograman-php/>
- Djapri, D. (2015). *Arduino*. Dipetik Februari 26, 2019, dari Arduino Nao: <https://djukarna4arduino.wordpress.com/2015/01/19/arduino-nano/>
- Narayanan, A. (2005). *Implementing RFID in Library*. Dipetik Maret 12, 2019, dari Implementing RFID in Library: Methodologies, Advantages, and Disadvantages: library.igcar.gov.in/readit-2005/conpro/lgw/s5-8.pdf
- Rahastri. (2015). Perancangan dan Implementasi Sistem Informasi Sekolah. *e- Proceeding of Applied Science : Volume 1 (ISSN : 2442-5826)*, 2660.
- Saputra, A. (2012). *WebTips: PHP, HTML5, dan CSS3*. Jakarta: JASAKOM.
- Stiawan, D. (2009). *Artikel Fasilkom Universitas Sriwijaya*. Dipetik Maret 14, 2019, dari Wireless Fundamental, Instalation & Implemetations.
- Syahwil, M. (2013). *Panduan Mudah Simulasi & Praktek Mikrokontroler Arduino*. Yogyakarta: Andi.
- Tri Kuntoro, P. (2005). *Jaringan Wifi, Teori dan Implementasi*. Yogyakarta: Andi.
- Wikipedia. (2019). *Microsoft Visual Studio*. Dipetik Maret 15, 2019, dari Microsoft Visual Studio- Wikipedia bahasa Indonesia, Ensiklopedia bebas: id.wikipedia.org