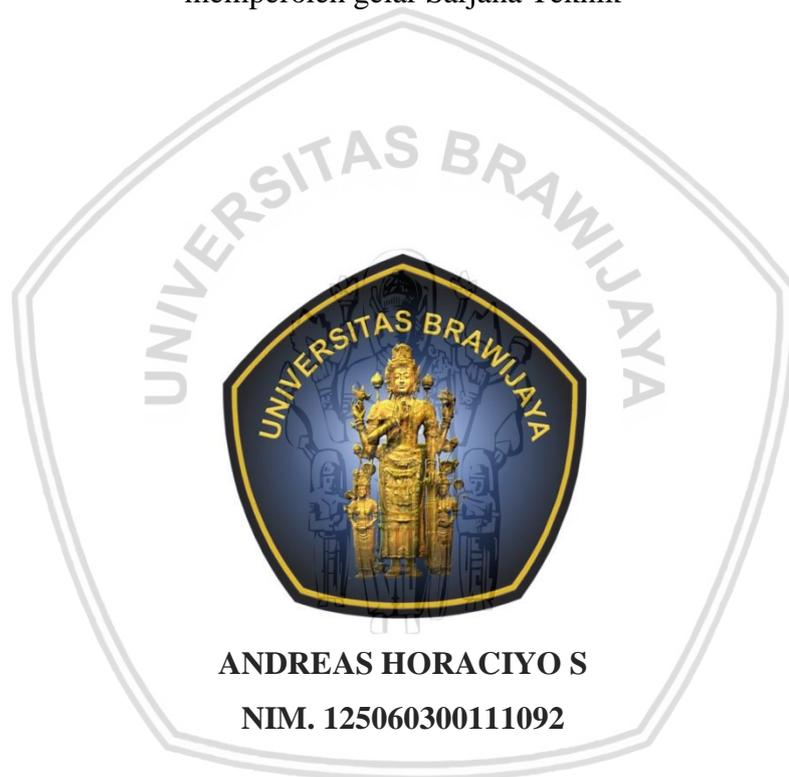


**RANCANG BANGUN SISTEM KOMUNIKASI UNTUK SISTEM
PEMBAYARAN PADA PRINTER MENGGUNAKAN TEKNOLOGI
*NEAR FIELD COMMUNICATION***

SKRIPSI

TEKNIK ELEKTRO KONSENTRASI ELEKTRONIKA

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



ANDREAS HORACIYO S
NIM. 125060300111092

UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2018



repository.ub.ac.id

LEMBAR PENGESAHAN
RANCANG BANGUN SISTEM KOMUNIKASI UNTUK SISTEM
PEMBAYARAN PADA PRINTER MENGGUNAKAN TEKNOLOGI
NEAR FIELD COMMUNICATION

SKRIPSI
TEKNIK ELEKTRO KONSENTRASI ELEKTRONIKA

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



ANDREAS HORACIYO S
NIM. 125060300111092

Skripsi ini telah direvisi dan disetujui oleh dosen pembimbing
Pada 04 Juni 2018

Dosen Pembimbing I

Dosen Pembimbing II

Raden Arief Setyawan, S.T., M.T
NIP : 19750819 199903 1 001

Dr-Ing. Onny Setyawati, S.T., M.T., M.Sc.
NIP : 19740417 200003 2 007

Mengetahui,
Ketua Jurusan/Ketua Program Studi

Hadi Suyono, ST., MT., Ph.D
NIP : 19730520 200801 1 013





I can do all things through HIM who strengthens me

Philippians 4 : 13



PERNYATAAN ORISINALITAS SKRIPSI

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas didalam Naskah Skripsi adalah asli dari pemikiran saya, tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik disuatu perguruan tinggi dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata didalam naskah Skripsi ini dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi dibatalkan, serta diproses sesuai peraturan perundang-undangan yang berlaku (UU No.20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

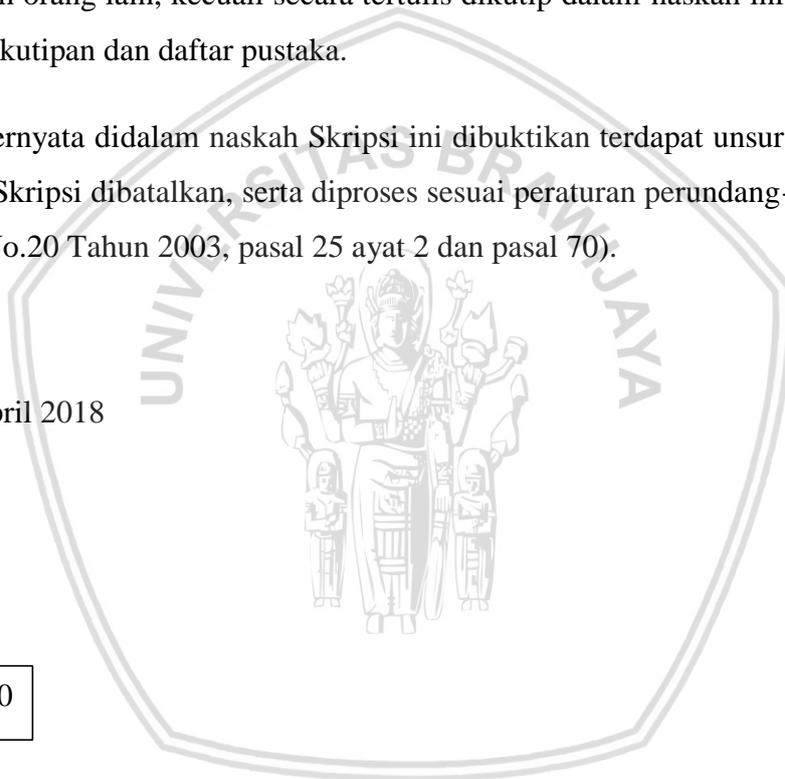
Malang, 23 April 2018

Mahasiswa,

Materai 6000

Andreas Horaciyo S

NIM. 125060300111092



RINGKASAN

Andreas Horaciyo S, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, April 2018, *Rancang Bangun Sistem Komunikasi untuk Sistem Pembayaran pada Printer Menggunakan Teknologi Near Field Communication*, Dosen Pembimbing : Raden Arief Setyawan dan Onny Setyawati.

NFC (*Near Field Communication*) adalah sebuah terobosan baru dalam dunia teknologi berbasis RFID (*Radio Frequency Identification*) dengan frekuensi kerja 13,56 MHz. Jarak komunikasi yang dekat serta kombinasi kode unik yang berbeda-beda setiap *tag* NFC membuat teknologi ini sangat sesuai untuk media yang membutuhkan tingkat keamanan yang tinggi. Sebagai contohnya adalah dapat diaplikasikan sebagai media pembayaran. Penelitian ini membahas mengenai salah satu aplikasi yang mengadopsi teknologi NFC yang merujuk pada sebuah perancangan alat.

Tag NFC yang digunakan adalah *tag* stiker NTAG213 dengan seri ISO14443-A yang sesuai dengan seri *tag* yang dapat dibaca oleh *reader* RC522. NodeMCU yang memiliki kemampuan untuk mengirim data menggunakan wifi digunakan sebagai mikrokontroler pada perancangan ini. Perancangan ini dilengkapi dengan database yang akan mentumpukan tabel yang berisikan nama pengguna, kode unik tag, dan jumlah pulsa yang dimiliki. Komunikasi antara mikrokontroler dan database server menggunakan protokol http untuk mengirim dan menerima data. Dirancang pula web yang akan mempermudah dalam proses pengisian pulsa maupun pendaftaran id.

Hasil pengujian menunjukkan akurasi 100 persen pada saat pengisian maupun pada saat pengurangan pulsa. Pulsa dapat bertambah sesuai keinginan pada saat pengisian pulsa dan dapat berkurang dengan tepat pada saat pengurangan pulsa. Berdasarkan hasil dari penelitian ini menunjukkan bahwa teknologi NFC dapat diaplikasikan untuk sistem pembayaran pada rental printer.

SUMMARY

Andreas Horaciyo S, Department of Electrical Engineering, Faculty of Engineering, Brawijaya University, April 2016, *Design Communication System for Printer Payment System Using Near Field Communication Technology*, Academic Supervisor: Raden Arief Setyawan and Onny Setyawati.

NFC (Near Field Communication) is a new breakthrough technology based on RFID (Radio Frequency Identification) operating at 13.56 MHz. Near communication distance and different combination of NFC *tag* unique codes make this technology very suitable for high security devices. This research developed one of the NFC technology application by designing of tool that can be used for printer payment system.

NFC *tag* used ISO14443-A series supported by RC522 *reader*. NodeMCU is used in this system because of it's feature that make NodeMCU can connect to the internet using WiFi. The system using a database that have collumns that contain user name, unique code, and the credit content. The communication between mikrokontroler and the server database is using the http method to send and receive data. A website is designed to make the process is easier.

The result showed that the system have 100% accuracy when adding charge or substracting charge from the credit. The result of the experiment showed that NFC technology can be applied to the printer payment system.

Keywords: NFC, RFID, *database*, application software, payment system

PENGANTAR

Terimakasih dan segala puji bagi Tuhan Yesus, karena atas segala petunjuk dan karunia yang diberikanNya sehingga skripsi ini dapat diselesaikan. Skripsi dengan judul “Rancang bangun sistem komunikasi untuk sistem pembayaran pada printer menggunakan teknologi *Near Field Communication*” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya. Penulis menyadari bahwa dalam penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

- Papa, Mama dan seluruh keluarga penulis atas segala macam dukungan yang telah diberikan kepada penulis hingga terselesaikannya skripsi ini.
- Bapak Hadi Suyono, ST., MT., Ph.D selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Ibu Ir. Nurussa’adah, MT. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya.
- Bapak Ir. Nanang Sulistiyanto, MT. selaku dosen pembimbing akademik atas segala bimbingan, nasehat dan motivasi yang telah diberikan.
- Bapak Raden Arief Setyawan, S.T., M.T. selaku Ketua Kelompok Jabatan Fungsional Konsentrasi Elektronika (Paket B) dan kedua dosen Pembimbing skripsi ini Bapak Raden Arief Setyawan, S.T., M.T. dan Ibu Onny Setyawati, Dr-Ing., S.T., M.T., M.Sc. atas segala bimbingan, kritik, dan saran yang telah diberikan.
- Bapak Ahmad Dulhadi, ST. selaku pranata di Laboratorium Sistem Digital, yang sudah membantu dalam hal peminjaman alat Laboratorium untuk skripsi ini.
- Seluruh dosen dan karyawan Teknik Elektro Universitas Brawijaya, yang telah memberikan banyak ilmu dan pelajaran berharga selama penulis menempuh ilmu di Teknik Elektro Universitas Brawijaya.
- Seluruh teman-teman asisten Laboratorium Sistem Digital mulai angkatan 2010 hingga 2014, yang telah berbagi suka duka bersama selama menuntut ilmu di Laboratorium.
- Semua teman-teman *Voltage* 2012, terutama konsentrasi Elektronika (Voctron) yang telah berbagi suka dan duka selama ini mulai dari probin maba yang tak terlupakan hingga penulisan skripsi sekarang ini.

- Dan semua orang yang telah membantu dan tidak bisa penulis sebutkan satu persatu, terimakasih banyak atas semua bantuannya.

Pada akhirnya, penulis menyadari skripsi ini masih belum sempurna. Penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat.

Malang, 30 Januari 2018

Penulis



DAFTAR ISI

Halaman

PENGANTAR	i
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vii
DAFTAR LAMPIRAN	vii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	2
BAB II TINJAUAN PUSATAKA	4
2.1. Jaringan Komputer	4
2.1.1. Local Area Network (LAN)	4
2.1.2. Metropolitan Area Network (MAN).....	5
2.1.3. Wide Area Network (WAN).....	5
2.2. Protokol TCP/IP	5
2.2.1. Arsitektur TCP/IP	5
2.2.2. Mekanisme TCP/IP.....	7
2.2.3. UDP (User Datagram Protocol)	8
2.2.4. TCP (Transfer Control Protocol).....	8
2.2.5. IP (Internet Protocol).....	8
2.2.6. HTTP (Hypertext Transfer Protocol)	9
2.3. MySQL.....	11
2.4. Application Programming Interface <i>Server</i>	12
2.4.1. REST (Representational State Transfer)	12
2.4.2. PHP (Hypertext Preprocessor).....	13
2.4.3. JSON (Javascript Object Notation)	13
2.5. Mikrokontroler NodeMCU.....	13
BAB III METODE PENELITIAN	16
3.1. Spesifikasi Perancangan	16
3.2. Perancangan Sistem Komunikasi untuk Sistem Pembayaran pada Printer	16



3.2.1. Perancangan Sistem.....	16
3.2.2. Konfigurasi Jaringan Internet dan Web Server	18
3.2.3. Perancangan <i>Database Server</i>	19
3.2.4. Perancangan <i>Web</i>	19
3.2.5. Perancangan Software Mikrokontroler	22
3.3. Pengujian Alat.....	25
3.3.1. Pengujian Pendaftaran ID Baru	25
3.3.2. Pengujian Pengisian Pulsa.....	26
3.3.3. Pengujian Pengurangan Pulsa	26
BAB IV HASIL DAN PEMBAHASAN.....	27
4.1. Hasil Pengujian Pendaftaran ID baru	27
4.2. Hasil Pengujian Pengisian Pulsa	29
4.3. Hasil Pengujian Pengurangan Pulsa	31
BAB V PENUTUP	33
5.1. Kesimpulan	33
5.2. Saran.....	34
DAFTAR PUSTAKA	35
LAMPIRAN.....	36



DAFTAR TABEL

No	Judul	Halaman
Tabel 2. 1	Spesifikasi Mikrokonroler NodeMCU	14
Tabel 3. 1	Rancangan Database Server	19
Tabel 3. 2	Alokasi pin mikrokontroller	24
Tabel 4. 1	Pengujian pendaftaran ID baru	28
Tabel 4. 2	Hasil pengujian pengisian pulsa	30
Tabel 4. 3	Hasil pengujian pengurangan pulsa.....	31



DAFTAR GAMBAR

No	Judul	Halaman
Gambar 2.1	Arsitektur Protokol TCP/IP	6
Gambar 2.2	Proses Pengalamatan TCP/IP	7
Gambar 2.3	Format paket TCP	8
Gambar 2.4	Format URL pada HTTP	9
Gambar 2.5	Format HTTP Request.....	10
Gambar 2.6	Format HTTP Response	11
Gambar 2.7	Mikrokontroler NodeMCU.....	14
Gambar 2.8	Diagram komponen ESP8266	15
Gambar 3.1	Blok Diagram Sistem	17
Gambar 3.2	<i>Flowchart</i> pendaftaran id	20
Gambar 3.3	Halaman Pendaftaran ID	20
Gambar 3.4	<i>Flowchart</i> pengisian pulsa	21
Gambar 3.5	Halaman pengisian pulsa	21
Gambar 3.6	<i>Flowchart</i> program utama mikrokontroler	23
Gambar 3.7	<i>Flowchart</i> proses print	25
Gambar 4.1	Proses pendekatan <i>tag</i> pada <i>reader</i>	27
Gambar 4.2	Tampilan form database sebelum pengujian	27
Gambar 4.3	Tampilan form database setelah pengujian	29
Gambar 4.4	Tampilan form pengisian pulsa	30
Gambar 4.5	Tampilan database setelah pengisian pulsa	31
Gambar 4.6	Tampilan database sesudah pengurangan pulsa	32

DAFTAR LAMPIRAN

No	Judul	Halaman
Lampiran 1	Listing Program Mikrokontroler	36
Lampiran 2	PHP Halaman Pendaftaran ID	41
Lampiran 3	PHP Halaman Pengisian Pulsa	43



BAB I

PENDAHULUAN

1.1. Latar Belakang

Salah satu pembahasan utama dalam dunia pendidikan sampai saat ini adalah teknologi. Saat ini, kecanggihan teknologi semakin dibutuhkan untuk memenuhi kebutuhan manusia yang semakin berkembang dan semakin rumit. Seiring perkembangan zaman, manusia berusaha melakukan berbagai hal untuk memenuhi hal tersebut. Penelitian semakin dikembangkan untuk mengembangkan teknologi yang sudah ada bahkan juga untuk menciptakan teknologi baru yang lebih mutakhir.

Sistem pembayaran secara garis besar menggunakan alat pembayaran yang diklasifikasikan menjadi dua yaitu alat pembayaran tunai dan non tunai (Hidayat, 2009: 1). Teknologi dalam sistem pembayaran juga semakin berkembang dan mendorong manusia untuk menggunakan mekanisme pembayaran yang lebih mempermudah proses transaksi. Manusia semakin meninggalkan sistem pembayaran tunai yang dinilai kurang efisien. Sistem pembayaran non tunai semakin diminati karena kelebihan yang lebih mudah dan aman saat digunakan. NFC merupakan salah satu teknologi yang dapat digunakan sebagai alat pembayaran non tunai.

NFC (*Near Field Communication*) merupakan teknologi komunikasi radio jarak dekat yang memungkinkan komunikasi antara dua perangkat dalam jangkauan, sehingga dapat digunakan untuk pemodelan sistem pembayaran *contactless* (Johnston dkk., 2015: 1446). Jangkauan kerja yang dekat membuat teknologi ini memiliki tingkat keamanan yang lebih tinggi. Kemampuan NFC untuk berkomunikasi memungkinkan NFC itu sendiri untuk digunakan sebagai alat pembayaran non tunai. Berbagai aplikasi teknologi telah menggunakan NFC di dalamnya, salah satunya dalam sistem pembayaran. Dalam penggunaan NFC untuk alat pembayaran dibutuhkan pula sistem komunikasi yang handal. Sistem komunikasi yang handal akan membuat sistem pembayaran bekerja lebih efisien.

Penelitian untuk membuat sebuah rancangan sistem komunikasi untuk proses transaksi dengan teknologi NFC dapat dijadikan solusi dari beberapa permasalahan tersebut. Perancangan ini, diharapkan kedepannya dapat lebih dikembangkan lagi untuk aplikasi yang lebih luas.

Pada penelitian ini teknologi NFC digunakan sebagai media pembayaran non tunai *card based instrument*. Perancangan yang dilakukan adalah merancang sebuah sistem komunikasi yang diaplikasikan untuk sistem pembayaran pada rental printer.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah diatas, dapat dirumuskan sebuah masalah bagaimana merancang sistem komunikasi yang dapat digunakan untuk sistem pembayaran pada printer. Adapun rincian dari rumusan masalah tersebut yang dijadikan sebagai acuan dalam penelitian ini adalah sebagai berikut :

1. Bagaimana prinsip kerja Rancang Bangun Sistem Komunikasi untuk Sistem Pembayaran pada Printer Menggunakan Teknologi NFC?
2. Bagaimana merancang dan membuat Sistem Komunikasi untuk Sistem Pembayaran pada Printer Menggunakan Teknologi NFC?

1.3. Batasan Masalah

Akibat banyaknya kemungkinan yang akan terjadi dalam penelitian ini, penulis membatasi masalah yang akan dibahas dalam penelitian ini meliputi:

1. Menggunakan PC sebagai *server*.
2. Menggunakan NodeMCU sebagai mikrokontroler dan *transmitter* data.
3. Menggunakan satu koneksi internet.
4. Tidak membahas pengontrolan printer.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dirumuskan, tujuan dari perancangan ini adalah merancang sistem komunikasi untuk sistem pembayaran pada printer dengan menggunakan teknologi NFC yang meliputi perancangan sistem komunikasi mikrokontroler dengan *server*, pembuatan *database* sistem, dan pembuatan *web*.

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini antara lain adalah dapat digunakan sebagai referensi untuk pengembangan aplikasi serupa di masa yang akan datang. Aplikasi ini diharapkan mampu meningkatkan sistem komunikasi yang dapat berjalan dengan cepat dan lebih aman.

Selain itu dengan proses transaksi yang lebih praktis, diharapkan proses bisnis dapat berjalan lancar sehingga mampu meningkatkan pendapatan pengusaha. Pengguna aplikasi ini diharapkan nantinya akan mendapatkan kenyamanan dan kemudahan dalam bertransaksi.



BAB II

TINJAUAN PUSTAKA

Tinjauan pustaka yang digunakan sebagai referensi dalam penelitian ini antara lain adalah teori-teori mengenai jaringan komputer, protokol komunikasi, *Application Programming Interface*, MySQL, serta mengenai mikrokontroler NodeMCU.

2.1. Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri dari kumpulan komputer yang saling terhubung satu dengan lain yang bertujuan untuk sharing informasi dan bertukar informasi komunikasi sumber daya. Setiap komputer atau perangkat peripheral yang terhubung disebut node, dan jaringan komputer yang terhubung paling sedikit memiliki 2 komputer di dalam jaringan yang memiliki banyak node puluhan bahkan jutaan sehingga terhubung satu dengan lain, dengan perantara memakai kabel tembaga, namun ada juga dengan fiber optick, Bluetooth, bahkan dengan teknologi satelit (Todd Lamle, 2013). Topologi jaringan adalah hal yang dapat menjelaskan bagaimana hubungan geometris antara unsur-unsur dasar penyusun jaringan, yaitu node, link, dan station. *Node* adalah perangkat yang menjadi titik penghubung dalam jaringan komputer. *Link* adalah media transisi yang digunakan, dan *station* adalah *client* atau *server* dalam jaringan.

Dalam konsep jaringan komputer, jarak merupakan hal yang penting dalam perancangannya, karena untuk setiap jarak tertentu diperlukan teknik yang berbeda. Berdasarkan jarak dan area jangkauannya, jaringan komputer dibagi menjadi tiga yaitu : *Local Area Network* (LAN), *Metropolitan Area Network* (MAN), *Wide Area Network* (WAN).

2.1.1. *Local Area Network* (LAN)

LAN adalah jaringan komputer yang mencakup area atau jangkauan yang sempit/tidak luas. Jaringan komputer ini biasa digunakan di rumah, kampus, atau kantor untuk menghubungkan komputer-komputer pribadi dan workstation menggunakan peralatan secara bersama-sama dan saling bertukar data dan informasi. LAN dapat digunakan tanpa membutuhkan fasilitas yang disediakan oleh perusahaan telekomunikasi umum.

2.1.2. *Metropolitan Area Network (MAN)*

MAN merupakan jaringan komputer yang mencakup area yang lebih luas daripada LAN. MAN digunakan untuk membangun infrastruktur jaringan komputer antar kantor dalam suatu kota. MAN mencakup perusahaan yang memiliki beberapa kantor dalam suatu kota. Jaringan komputer ini mampu mencakup area dengan radius 10-50 km.

2.1.3. *Wide Area Network (WAN)*

WAN adalah jaringan komputer yang memiliki cakupan area yang sangat luas. Jangkauan area WAN lebih luas daripada satu kota, bias mencakup lintas Negara ataupun lintas benua. WAN menggunakan komponen untuk berkomunikasi yaitu media transmisi dan perangkat *switching*. Media transmisi digunakan untuk menyalurkan sinyal dari pengirim kepada penerima, sedangkan perangkat *switching* digunakan untuk meneruskan sinyal.

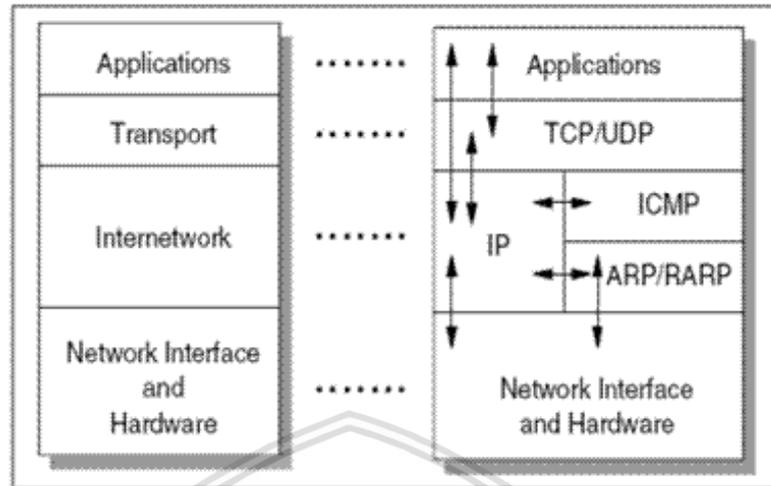
2.2. Protokol TCP/IP

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data antar komputer di internet. Dengan menggunakan protokol yang sama maka perbedaan jenis komputer dan perbedaan sistem operasi bukan menjadi sebuah masalah.

TCP/IP terdiri atas sekumpulan protokol yang masing-masing memiliki fungsi atas bagian tertentu dalam komunikasi data. Dengan pembagian fungsi tersebut sehingga masing-masing protokol memiliki tugas yang jelas dan sederhana. Protokol satu dengan yang lainnya akan tetap bekerja dengan baik apabila masih dapat mengirim dan menerima data.

2.2.1. *Arsitektur TCP/IP*

Seperti pada gambar 2.1, TCP/IP terdiri atas empat *layer*, yaitu *Network Access/Interface Layer*, *Internet Layer*, *Host-to-Host Transport Layer*, dan *Application Layer*.

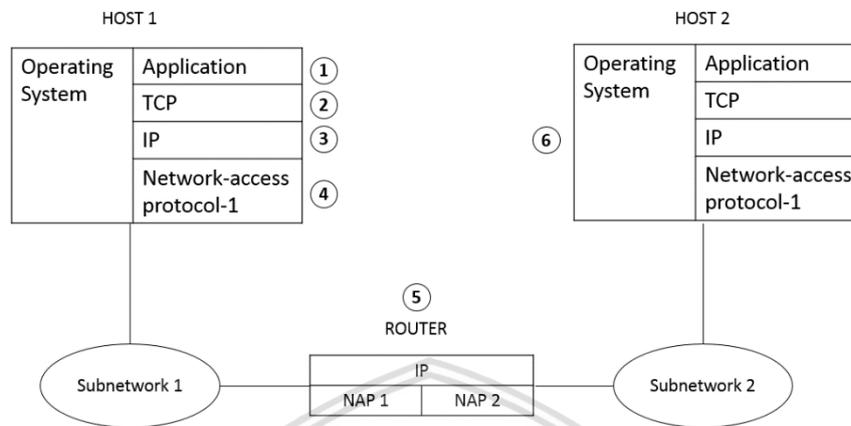


Gambar 2.1 Arsitektur Protokol TCP/IP

Sumber: Hunt, 1992

1. *Network Interface Layer* bertanggung jawab mengirim data ke media fisik dan menerima data dari media fisik. Media fisik dapat berupa kabel, serat optic, atau gelombang radio. Protokol pada *layer* ini bertugas menerjemahkan sinyal listrik menjadi sinyal digital yang dimengerti komputer atau peralatan lain.
2. *Internet Layer* bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol, antara lain: IP, ARP, dan ICMP. IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat. ARP (*Address Resolution Protocol*) berfungsi untuk menemukan alamat hardware dari host komputer yang terletak pada jaringan yang sama. ICMP (*Internet Control Message Protocol*) berfungsi untuk mengirimkan pesan dan melaporkan kegagalan pengiriman data.
3. *Host-to-Host Transport Layer* bertanggung jawab untuk mengadakan komunikasi antara dua host/komputer. Pada *layer* ini terdapat dua protokol, yaitu TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*).
4. *Application Layer* bertanggung jawab atas aplikasi yang menggunakan protokol TCP/IP. Pada *Layer* ini terdapat beberapa protokol seperti HTTP (*Hypertext Transfer Protocol*), Telnet, FTP (*File Transfer Protocol*), dan SMTP (*Simple Mail Transfer Protocol*).

2.2.2. Mekanisme TCP/IP



Gambar 2.2 Proses Pengalamatan TCP/IP

Sumber : Stallings, 2001

Gambar 2.2 menunjukkan proses pengalamatan pada TCP/IP. Proses ini terjadi ketika *host 1* mengirim proses pada *host 2*. Proses-proses tersebut antara lain:

1. Proses pengiriman akan membangkitkan sebuah blok data dan melewatkannya menuju TCP.
2. TCP menambahkan informasi control (*header*) sepanjang 20 *byte* untuk masing-masing potongan data sehingga membentuk segmen TCP.
3. TCP membawa masing-masing segmen ke IP dengan perintah transmisi data ke *host 2*.
4. Masing-masing *datagram* IP dibawa ke *network access layer* untuk ditransmisikan pada tujuan. Pada proses ini terjadi penambahan *network access header*.
5. Data diterima di *host 2* dan terjadi proses kebalikannya. Masing-masing *layer* akan melepaskan *headernya* sebelum melewatkan data ke *layer* di atasnya dan sisanya dilewatkan ke *layer* yang lebih tinggi sampai data asli dikirimkan ke tujuan.

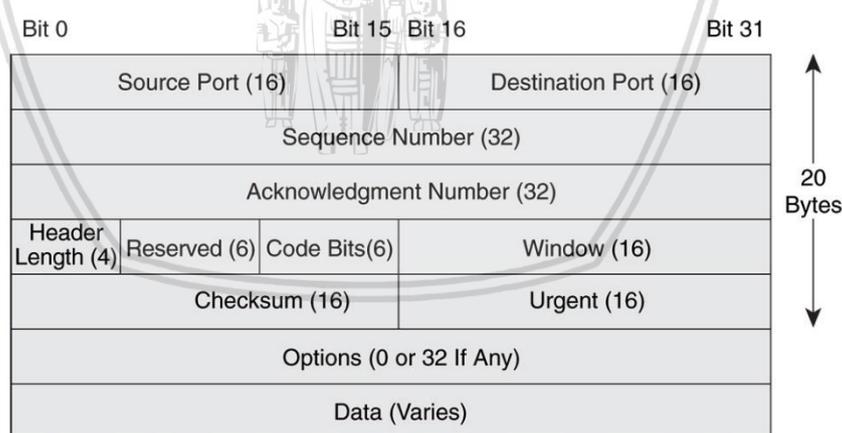
Pada TCP/IP, masing-masing *layer* menambahkan informasi kontrol (*header*) untuk menjamin pengiriman yang sesuai. Informasi control ini diletakkan di depan setiap data yang akan ditransmisikan. Setiap *layer* memperlakukan seluruh data yang diterima olehnya sebagai data dan akan menambahkan header di depan informasi tersebut. Proses penambahan header pada tiap *layer* ini dinamakan proses enkapsulasi.

2.2.3. UDP (*User Datagram Protocol*)

UDP merupakan protokol yang berada di *Host-to-Host Transport Layer*. Protokol ini bersifat *connectionless* dan *unreliable*. Dikarenakan sifatnya yang *connectionless*, maka tidak diperlukan pembentukan suatu hubungan terlebih dahulu untuk pengiriman data. Protokol ini tidak menjamin data yang dikirimkan akan sampai dalam kondisi yang benar karena sifatnya yang *unreliable*. Protokol ini lebih sering diimplementasikan untuk aplikasi seperti multimedia *streaming* dimana dibutuhkan nilai *delay* yang kecil dan rugi-rugi paket data kecil masih bisa ditoleransi.

2.2.4. TCP (*Transfer Control Protocol*)

TCP merupakan protokol yang terletak pada *transport layer* di arsitektur TCP/IP. Berbeda dengan UDP, TCP memiliki sifat *connection oriented*, *reliable*, dan memiliki *byte stream service* bagi *application layer*. TCP memerlukan adanya pembentukan hubungan antara aplikasi pengguna TCP sebelum dapat melakukan pertukaran data. TCP menerapkan proses deteksi kesalahan paket dan transmisi sehingga memiliki sifat yang *reliable*. *Byte stream service* membuat paket yang dikirimkan dan sampai ke tujuan secara berurutan. Format paket TCP seperti pada gambar 2.3.



Gambar 2.3 Format paket TCP

Sumber: RFC 793

2.2.5. IP (*Internet Protocol*)

IP terletak pada *network layer* dan merupakan inti dari protokol TCP/IP. Sama seperti UDP, IP melakukan pengiriman data yang bersifat *connectionless* dan *unreliable*. IP melakukan *best effort delivery service* yang berarti akan melakukan usaha sebaik-baiknya

agar data yang dikirim sampai ke tujuan. Apabila dalam proses pengiriman data tersebut terjadi gangguan (jalur putus, router macet, atau *host* sedang down), IP hanya akan memberitahukan kepada pengirim melalui ICMP bahwa terjadi masalah dalam proses pengiriman data. Untuk meningkatkan keandalan protokol ini, harus disediakan oleh protokol lain yang berada di *layer* atasnya (TCP dan *Application layer*).

IP memiliki *datagram delivery service* yang berarti setiap data yang dikirim adalah independen terhadap paket data yang lainnya. Akibatnya, jalur yang ditempuh masing-masing paket data IP ke tujuannya bisa menjadi berbeda satu dengan lainnya karena jalur yang ditempuh berbeda-beda dan paket yang sama bisa menjadi tidak berurutan.

2.2.6. HTTP (*Hypertext Transfer Protocol*)

HTTP merupakan protokol yang terletak pada *application layer*. HTTP tidak menentukan bagaimana data akan diproses, HTTP hanya bertanggung jawab menentukan bagaimana data tersebut ditransfer. HTTP dapat digunakan untuk mentransfer semua format data. Selain *web browser*, aplikasi lain juga dapat melakukan transfer data menggunakan protokol ini.

HTTP bersifat *request-response*, dimana dalam protokol ini client terlebih dahulu melakukan pesan request ke *server* dan selanjutnya *server* akan memberikan *response* sesuai dengan *request* yang dikirimkan oleh client. *Request* dan *response* dalam HTTP disebut sebagai *request chain* dan *response chain*. Hubungan HTTP yang paling sederhana terdiri atas hubungan *user-agent* dengan *server* asal.

HTTP menggunakan format URL (*Universal Resource Locator*) dalam bentuk seperti pada gambar 2.4.

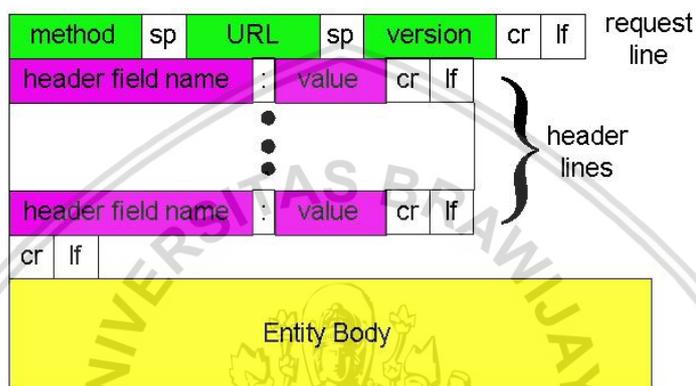
`http : // host [:port] / [resource_path]`

Gambar 2.4 Format URL pada HTTP
Sumber: Purbo, 1998

Host merupakan nama domain yang legal atau *IP address* dari *server*. *Port* adalah bilangan yang menunjukkan *port* HTTP pada *host*, jika *port* tidak disebutkan maka *port*

HTTP akan diasumsikan pada *port* 80. *Resource_path* menyatakan lokasi *resource* dalam host.

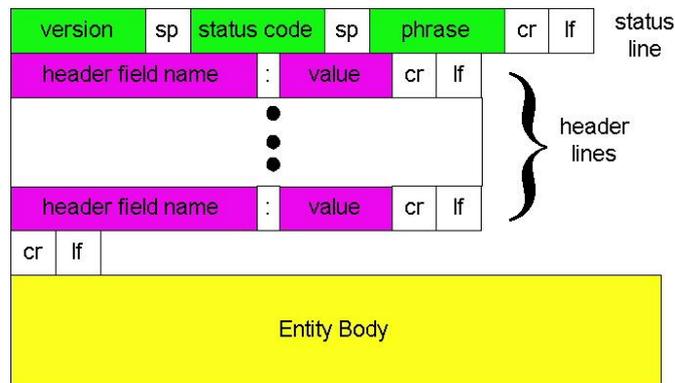
Komunikasi protokol HTTP terdiri dari pesan request yang diberikan oleh *user-agent* dan *response server*. HTTP menggunakan format pesan yang generic dalam setiap *request* dan *response*. Pesan HTTP terdiri atas baris-mulai, *header*, dan isi pesan. Header dan isi pesan dipisahkan oleh sebuah baris kosong yang terdiri dari karakter CRLF.



Gambar 2.5 Format HTTP Request
Sumber: Heidemann, 1997

Seperti pada gambar 2.5, setiap HTTP *request* diawali dengan baris mulai/*request line*. Baris mulai tersebut berisi metode *request*, *request-URL*, dan versi protokolnya. Metode *request* menunjukkan metode apa yang hendak dilakukan atas *resource* yang ditunjuk oleh *request-URL*. Ada beberapa metode yang didefinisikan oleh HTTP/1.1, antara lain:

1. GET, metode GET mengambil informasi apa saja dalam bentuk entity yang didefinisikan oleh *request-URL*.
2. POST, metode POST digunakan untuk meminta *server* menempatkan entity yang dikirim bersama *request* sebagai subordinat dari *request-URL* yang dituju. Metode ini biasa digunakan untuk mengirimkan *form*.



Gambar 2.6 Format HTTP Response
 Sumber: Heidemann, 1997

Setelah menerima *request*, HTTP akan memberikan *response* atas *request* tersebut. Seperti pada gambar 2.6, *response* HTTP terdiri atas baris status, *header* dan isi pesan. Baris status berisi kode status yang berupa kode tiga digit dan frasa alasan yang merupakan penjelasan singkat atas kode status tersebut. Digit pertama kode status menentukan kelas dari *response*. Protokol HTTP/1.1 mendefinisikan 5 nilai digit pertama sebagai berikut;

1. *Informational* (*request* diterima dan proses berlanjut) : 1XX
2. *Success* (*request* diterima dan dimengerti) : 2XX
3. *Redirection* (*request* membutuhkan tindakan lebih lanjut) : 3XX
4. *Client error* (*request* mengandung sintaks yang salah) : 4XX
5. *Server error* (*server* gagal melakukan tindakan sesuai *request*) : 5XX

Server HTTP dapat menghasilkan kode-status selai yang didefinisikan dalam RFC sepanjang digit pertama kode tersebut telah dimengerti oleh aplikasi yang menggunakan HTTP.

2.3. MySQL

MySQL adalah sistem *database relational* (RDBMS) yang bersifat *open source freeware* dan berada dibawah lisensi GPL (General Public License). *Database* pertama kali dirintis seorang *programmer database* bernama Michael Widenius. Pada awalnya MySQL hanya dapat berjalan pada sistem operasi Unix dan Linux dan seiring berjalannya waktu diciptakan pada hampir semua *platform* termasuk *platform* yang umum digunakan yaitu Windows.

MySQL merupakan turunan dari salah satu konsep utama basis data yaitu SQL (*Structured Query Language*). SQL merupakan bahasa terstruktur yang berfungsi sebagai penghubung program *database* dengan bahasa pemrograman. Dalam penggunaan SQL

terdapat aturan-aturan penggunaan yang telah distandarkan dengan standar ANSI (*American National Standards Institute*) atau ISO SQL.

MySQL memiliki fasilitas API (*Application Programming Interface*) yang memungkinkan basis data MySQL dapat diakses oleh bahasa pemrograman lain. Kemampuan ini merupakan salah satu dari beberapa kelebihan MySQL. Kelebihan lainnya adalah MySQL dapat menangani data dalam skala yang besar (lebih dari 50 juta *record*, 60 ribu tabel, dan 5 milyar baris). MySQL juga dapat digunakan oleh beberapa user dalam waktu bersamaan tanpa mengalami masalah (*multiuser*) dan juga mendukung berbagai tipe data yang kompleks dan disertai perlindungan keamanan.

2.4. *Application Programming Interface Server*

Application Programming Interface (API) adalah sekumpulan fungsi, perintah, dan protokol yang dapat digunakan untuk menghubungkan satu aplikasi dengan aplikasi lainnya sehingga dapat berinteraksi. Seiring perkembangan internet, API diimplementasikan pada sisi *server* dan dapat digunakan oleh beberapa aplikasi yang terhubung ke *server* menggunakan protokol tertentu. Pada protokol HTTP, *Application Programming Interface* umumnya disebut sebagai *Web Application Programming Interface Server* atau *Web Service*.

2.4.1. REST (*Representational State Transfer*)

Representational State Transfer adalah sebuah *web architectural style* dan bukan merupakan suatu protokol. Ini disebabkan karena REST menggunakan protokol yang berada pada *application layer* termasuk protokol HTTP. Sebelum REST ditemukan, dalam membuat sebuah *Web Application Programming Interface Server* digunakan SOAP (*Simple Object Access Protocol*). SOAP merupakan protokol untuk memanipulasi resource dengan cara membuat entity pesan dalam format XML (*Extensible Markup Language*) yang memenuhi standart yang dikeluarkan oleh W3C dan mengirimkan pesan request tersebut dengan metode POST ke *server*. Kemudian *server* melakukan *parsing* terhadap *entity* XML tersebut untuk menentukan *request* yang diinginkan client.

Prinsip kerja REST berbeda dengan SOAP. Pada implementasinya, REST jauh lebih cepat dibandingkan dengan SOAP karena REST tidak membutuhkan *entity* pesan dalam format XML, sehingga membuat proses persiapan oleh client tidak membutuhkan proses parsing XML pada *server*. Pada REST, *resource* yang diidentifikasi sebagai *request-*

URL dapat dimanipulasi dengan menggunakan HTTP *method* yang sesuai. *Server* akan mengirim representasi dari *resource* yang diminta client dalam format apapun setelah menerima *request* dari client. Sekarang ini format yang banyak dikenali oleh client adalah JSON.

Berikut merupakan ciri-ciri dari REST :

1. Penggunaan HTTP *method (verb)* yang tepat untuk memanipulasi *resource*
2. Request-URL tidak mengandung *query string* dan hanya terdiri dari elemen *resource* yang dipisahkan dengan karakter “/”.
3. Penggunaan kode status yang tepat pada HTTP *Response*.

2.4.2. PHP (*Hypertext Preprocessor*)

PHP merupakan bahasa berbentuk script yang berada pada *server* dan diproses di *server* pula. PHP mempunyai fungsi yang sama dengan script lainnya seperti ASP (*Active Server Page*), JSP (*Java Server Pages*) dan lainnya.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). Pertama kali dibuat oleh Rasmus Ledorf tahun 1995. Saat itu PHP masih bernama *Form Interpreted (FI)*, yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari *web*. Lalu Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisannya menjadi sumber terbuka, maka banyak pemrogram yang tertarik mengembangkan PHP.

2.4.3. JSON (*Javascript Object Notation*)

JSON merupakan sebuah berkas yang umumnya digunakan untuk pertukaran data pada internet. Berkas JSON berbasis teks membuat JSON mudah dikenali oleh berbagai macam bahasa pemrograman lain. Hal ini membuat JSON menjadi ideal untuk digunakan dalam pertukaran data antar aplikasi yang berbeda bahasa pemrogramannya.

2.5. Mikrokontroler NodeMCU

NodeMCU adalah mikrokontroler dengan mikroprosesor *open source* untuk *platform IoT (Internet of Things)*. Pada NodeMCU juga terdapat *firmware* yang berjalan pada ESP8266 *WiFi SoC* dari *Espressif System* dan perangkat keras yang berbasis ESP-12 *module* sehingga NodeMCU mendukung koneksi dengan *WiFi*. NodeMCU dapat dihubungkan dengan komputer menggunakan USB dan dapat diprogram dengan menggunakan program

editor seperti LUA dan Arduino IDE. NodeMCU dapat diprogram menggunakan bahasa pemrograman C atau C++ yang dicustom menjadi sebuah *software* khusus sebagai *platform* pemrograman. NodeMCU memiliki keunggulan dibanding mikrokontroler lainnya seperti Arduino Uno. NodeMCU memiliki kemampuan untuk terhubung ke jaringan *wireless* tanpa harus menggunakan perangkat tambahan. Tampilan NodeMCU dapat dilihat seperti pada gambar 2.7.



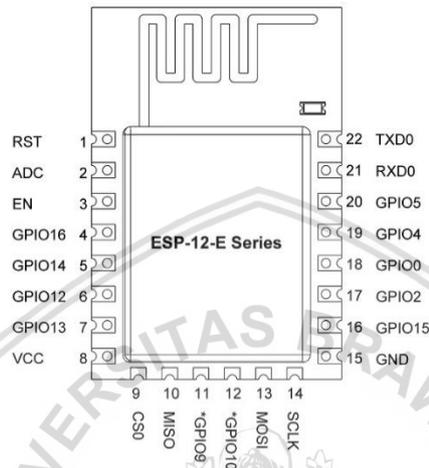
Gambar 2.7 Mikrokontroler NodeMCU
Sumber : NodeMCU, 2014

Pada skripsi ini NodeMCU diprogram menggunakan editor Arduino IDE dengan bahasa C. Dengan teknologi *low power*, perangkat NodeMCU aktif dengan tegangan 3,3V. Spesifikasi NodeMCU dapat dilihat pada tabel 2.1.

Tabel 2. 1 Spesifikasi Mikrokontroler NodeMCU

Protokol <i>WiFi</i>	802.11 b/g/n
Frekuensi Kerja	2.4G-2.5G (2400M-2483,5M)
Tegangan Operasi	3,0V-3,6V
Arus Operasi	80mA
Protokol Jaringan	Ipv4, TCP/UDP/HTTP/FTP
Mode <i>WiFi</i>	Station/softAP/softAP+station

Salah satu kelebihan NodeMCU adalah terdapatnya ESP8266-12 yang berfungsi untuk menyambungkan NodeMCU dengan jaringan *wireless*. Diagram komponen ESP8266 dapat dilihat pada gambar 2.8.



Gambar 2.8 Diagram komponen ESP8266
Sumber: espressif, 2016

ESP8266EX merupakan *WiFi chips* yang paling bisa diintegrasikan dalam dunia industri. Dengan ukurannya yang kecil, ESP8266EX hanya memerlukan sedikit *circuit external* dan terintegrasikan dengan Tensilico MCU 32-bit. Pada skripsi ini, perangkat ESP8266 yang digunakan telah terdapat pada mikrokontroler NodeMCU. Pada skripsi ini ESP8266 berfungsi sebagai modul WiFi agar mikrokontroler dapat terhubung langsung dengan koneksi jaringan WiFi yang tersedia. Modul ini memiliki tiga mode WiFi, yaitu *station*, *access point*, dan *both* (keduanya). Modul ini juga dilengkapi dengan prosesor, memori, dan GPIO.

BAB III

METODE PENELITIAN

Metode penelitian yang digunakan pada skripsi ini adalah metode studi kepustakaan dan penelitian laboratorium. Studi kepustakaan dilakukan sebagai penunjang yang berupa data literatur, informasi dari internet dan konsep-konsep teoritis dari buku penunjang. Penelitian laboratorium berupa perancangann perangkat keras dan perangkat lunak. Langkah-langkah yang dilakukan untuk merealisasikan alat yang dirancang adalah penentuan spesifikasi, perancangan alat dan pengujian alat.

3.1. Spesifikasi Perancangan

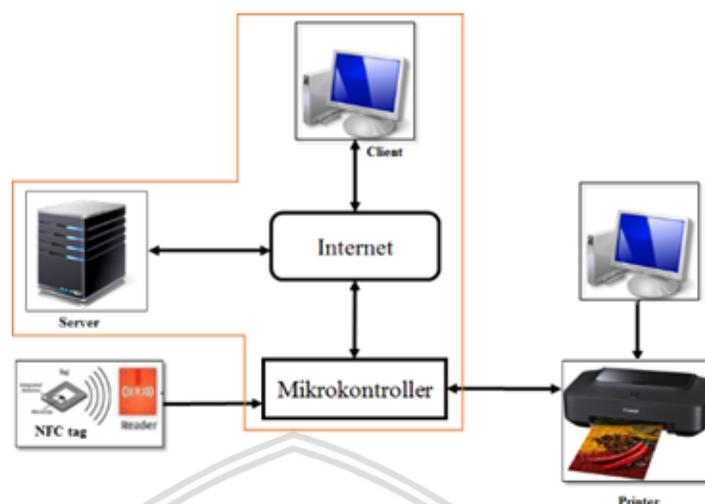
Dalam perancangan ini, pada perancangan sistem komunikasi dibutuhkan alat dan bahan sebagai berikut :

1. Seperangkat PC (Personal Komputer) yang dilengkapi dengan *software* XAMPP.
2. *Software* ArduinoIDE
3. Mikrokontroler NodeMCU V3.
4. *cPanel* (sebagai *web hosting control panel*).

3.2. Perancangan Sistem Komunikasi untuk Sistem Pembayaran pada Printer

3.2.1. Perancangan Sistem

Perancangan sistem ini dibagi menjadi dua bagian utama, yaitu perancangan sistem pada perangkat keras dan perangkat lunak. Perangkat keras terdiri atas mikrokontroler yang terhubung ke perangkat lainnya seperti LCD, keypad, *reader* dan printer, sedangkan perangkat lunak yang dirancang terdiri atas *database* dan *web* aplikasi untuk mengakses *database*. Secara keseluruhan, diagram blok perancangan sistem pembayaran ditunjukkan pada Gambar 3.1. Bagian yang akan dirancang pada penelitian ini adalah bagian sistem komunikasi antara mikrokontroler dan *server* melalui internet (bagian dilingkupi garis merah yang ditunjukkan dalam Gambar 3.1).



Gambar 3.1 Blok Diagram Sistem

Pada diagram blok tersebut dijelaskan bahwa input mikrokontroller terdiri atas tiga bagian yaitu dari *reader*, *keypad*, LCD dan printer. *Tag* NFC dibaca oleh *reader* yang dihubungkan dengan mikrokontroller. Mikrokontroller dengan jaringan internet dihubungkan oleh tanda panah bolak-balik. Hal ini menjelaskan bahwa antara mikrokontroller, *server* dan *client* terdapat komunikasi dua arah, selain menerima data dari *server*, mikrokontroller juga berperan mengirim data ke *server*. Mikrokontroller menerima data ID dan jumlah pulsa dari *server* dan juga berperan mengirimkan data jumlah pulsa yang telah dikurangi setelah proses *printing* dilakukan. Komunikasi antara mikrokontroller, *server* dan *client* berlangsung melalui jaringan internet yang akan diakses melalui *web*. Mikrokontroller juga akan menerima respon dari *server* yang kemudian akan diolah lagi oleh mikrokontroller untuk ditampilkan di LCD.

Prinsip kerja dari alat ini dimulai dari pembacaan kode unik *tag* NFC oleh *reader* yang menghasilkan sebuah kode unik *tag* yang kemudian dikirim ke mikrokontroller untuk diolah lebih lanjut. Hasil pengolahan kode unik oleh mikrokontroller tersebut kemudian melalui jaringan internet dikirim ke *server* untuk dicocokkan dengan kode unik yang sudah terdaftar pada *database*. *Database* yang dimaksud dalam hal ini adalah *database* menggunakan SQL yang implementasinya nanti digunakan sebagai *database server*.

Apabila kode unik yang diterima terdapat pada *database*, maka *server* akan mengirimkan jumlah pulsa yang dimiliki oleh kode unik tersebut, dimana hal ini dijadikan sebagai indikator bahwa *tag* tersebut telah terdaftar pada *database*. *Tag* yang telah terdaftar dapat melakukan proses pengisian pulsa maupun proses *printing*. Pengisian pulsa dilakukan

melalui *web* menggunakan *pc client*. Jumlah pulsa yang akan diisikan kemudian akan ditambahkan kedalam *database*.

Ketika *tag* telah teridentifikasi dan terdaftar dalam *database* maka proses pengisian pulsa dapat dilakukan. Jumlah pulsa yang akan diisikan dapat dimasukkan didalam form pengisian pulsa melalui *pc client*. Setelah jumlah pulsa yang akan diisikan telah ditentukan, proses tahap ini diakhiri dengan tombol Tambah Saldo pada form. Secara otomatis isi pulsa pada id yang sudah terdeteksi tadi akan bertambah sesuai dengan jumlah pulsa yang sudah diisikan pada form.

Berbeda halnya apabila id belum terdaftar, maka kode unik yang diterima tidak terdapat pada *database*, dan *server* akan mengirimkan sebuah karakter yang dijadikan sebagai indikator bahwa *tag* tersebut tidak terdaftar pada *database* dan tidak dapat diisi pulsanya. *Web* akan menampilkan form pendaftaran id untuk menambahkan id tersebut kedalam *database*.

Pada perancangan ini proses sistem dibagi menjadi beberapa bagian yaitu proses perancangan *database*, proses perancangan *web*, dan proses perancangan software mikrokontroler.

3.2.2. Konfigurasi Jaringan Internet dan Web Server

Blok diagram pada Gambar 3.1 menunjukkan *PC client* dapat mengakses *web server* melalui internet. *Server* yang digunakan yaitu *PC* dengan paket perangkat lunak *XAMPP*. *Server* akan menampilkan halaman-halaman yang digunakan seperti halaman pengisian pulsa maupun pendaftaran id.

Konfigurasi Jaringan dan Web Server yang digunakan pada sistem memiliki spesifikasi berikut:

1. Sistem Operasi digunakan untuk menjalankan paket perangkat lunak yang dibutuhkan oleh sistem. Sistem operasi yang digunakan pada server adalah sistem operasi *Linux*.
2. Topologi yang diterapkan pada perancangan merupakan topologi star dimana semua *PC client* akan terhubung secara langsung ke server melalui internet.
3. Jaringan internet yang digunakan merupakan jaringan internet dengan kecepatan download 3,83 Mbps dan kecepatan upload 0,65 Mbps.

4. *Web server* digunakan untuk menampilkan halaman pengisian pulsa dan halaman pendaftaran id. Paket perangkat lunak untuk membangun *web server* adalah XAMPP yang telah terdapat *Apache* dan PHP di dalamnya.
5. *Domain* yang digunakan pada perancangan merupakan domain yang dibuat melalui *web* www.dot.tk. *Web* ini dapat digunakan untuk membuat domain gratis maupun berbayar. *Domain* yang digunakan pada perancangan adalah rfidprint.tk.
6. *CPanel web hosting control panel* digunakan untuk menyimpan data dari *web* sehingga dapat diakses menggunakan internet. *CPanel* menyediakan graphical interface dan tools untuk memaksimalkan kinerja suatu website.
7. MySQL digunakan untuk membuat database pada *server* untuk menyimpan data-data pengguna berupa id dan saldo yang dimiliki.
8. PHP digunakan untuk menuliskan script dari halaman *web* akan dibuat. Pada perancangan, pembuatan file PHP menggunakan CMS yaitu wordpress untuk mempermudah proses pembuatan dan membuat tampilan *web* menjadi lebih menarik,

3.2.3. Perancangan Database Server

Pembuatan *database* pada perancangan ini menggunakan software MySQL dari XAMPP. Perancangan *database server* ditunjukkan oleh Tabel 3.1. *Database* yang dirancang terdiri atas satu tabel dengan empat kolom. Kolom pertama adalah No, kolom kedua ID yang berisi kode unik dari *tag*. Kolom ketiga adalah Nama untuk memasukkan nama pemilik id *tag*. Kolom terakhir adalah Saldo yang berisikan jumlah pulsa yang dimiliki *tag*.

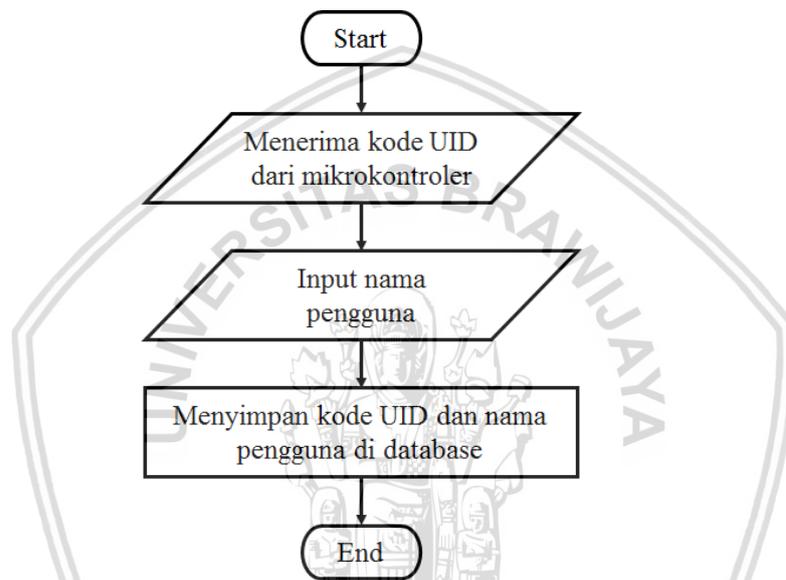
Tabel 3.1 Rancangan *database server*

Nama Kolom	Tipe Data
No	INT (10)
ID_kartu	VARCHAR (16)
Nama	VARCHAR (50)
Saldo	INT (11)

Database yang telah dibuat nantinya akan diupload ke *server hosting* dan dapat diakses melalui internet. *Server hosting* yang digunakan adalah Cpanel.

3.2.4. Perancangan Web

Proses pendaftaran id baru dan proses pengisian pulsa pada sistem dilakukan melalui *web*. Melalui *web* maka id baru dapat dimasukkan ke dalam *database* dan pulsa yang diisikan ditambahkan ke dalam *database*. Terdapat 2 halaman yang dirancang yaitu halaman yang berisi form pendaftaran id baru dan halaman yang berisi form pengisian pulsa. *Web* hasil perancangan dapat diakses di rfidprint.tk.

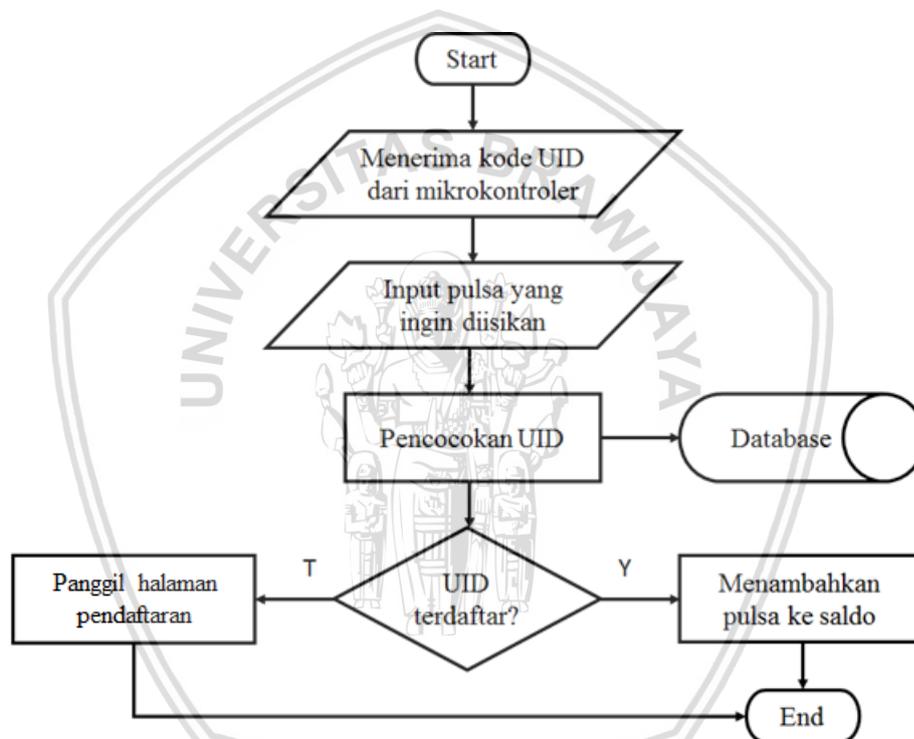


Gambar 3.2 *Flowchart* pendaftaran id

Gambar 3.3 Halaman Pendaftaran ID

Sebelum melakukan proses pengisian pulsa maupun proses printing, pengguna harus mendaftarkan *tag* NFC agar terdata dalam *database server*. Gambar 3.2 menunjukkan

flowchart pendaftaran id. Pada proses pendaftaran ID baru, *server* akan menunggu kode UID yang dibaca oleh *reader* melalui mikrokontroler. Setelah kode unik dikirim oleh mikrokontroler maka kode unik tersebut akan tertera pada form pendaftaran. Setelah proses tersebut, pengguna diminta untuk memasukkan nama pengguna yang akan disimpan di *database*. Apabila nama data UID dan nama pengguna telah dimasukkan maka data tersebut akan disimpan dan terdaftar di dalam *database server*. Tampilan halaman pendaftaran id baru dapat dilihat pada gambar 3.3.



Gambar 3.4 Flowchart pengisian pulsa

The screenshot shows a web browser window with the URL `rfidprint.tk/index.php/Printer/transaction/045fd992`. The page has a navigation bar with 'Printer Otomatis', 'Daftar Baru / Edit', 'Tambah Saldo', and 'Database Pengguna'. The main content area is split into two panels:

- Tambah Saldo (Left Panel):** Contains a form for adding balance. It has three input fields: 'Identitas Kartu' (with value '045fd992'), 'Nama Pengguna' (with value 'coklat'), and 'Tambah Saldo' (with value 'Jumlah Saldo'). A blue 'Tambah Saldo' button is at the bottom.
- Catatan Transaksi (Right Panel):** A table showing transaction history.

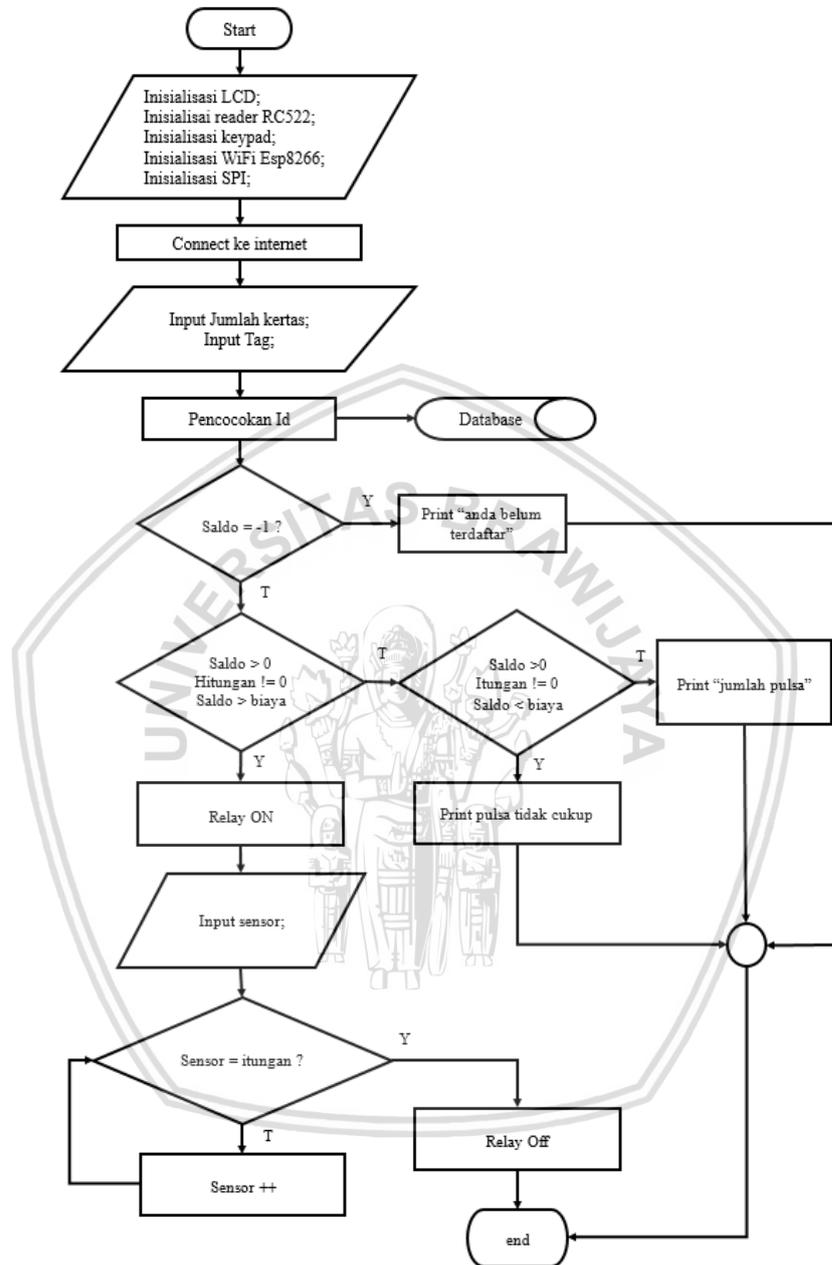
Tanggal Trx	Saldo
19 Mar 2018	Rp 5.000
19 Mar 2018	(Rp 1.000)
19 Mar 2018	(Rp 1.500)
19 Mar 2018	(Rp 500)
Saldo Akhir	Rp 2.000

Gambar 3.5 Halaman pengisian pulsa

Setelah *tag* NFC terdaftar dalam *database* maka proses pengisian pulsa dapat dilakukan. Proses pengisian pulsa dapat dilihat seperti pada Gambar 3.4. Pertama *server* akan menerima kode UID dari mikrokontroler dan pengguna diminta untuk memasukkan jumlah pulsa yang ingin diisikan ke dalam form pengisian pulsa. Apabila kode UID dan jumlah pulsa sudah diterima, maka *server* akan melakukan pencocokan UID dengan *database* apabila UID terdaftar dalam *database* maka *server* akan menambahkan pulsa sejumlah yang ingin diisikan ke dalam saldo *tag* tersebut. Bila UID belum terdaftar maka *server* akan meminta pengguna untuk mendaftarkan kartu terlebih dahulu dan memanggil halaman pendaftaran id. Tampilan halaman pengisian pulsa dapat dilihat pada gambar 3.5.

3.2.5. Perancangan Software Mikrokontroler

Program utama proses yang ada di mikrokontroler ditunjukkan oleh *flowchart* pada Gambar 3.6. Proses di mikrokontroler diawali dengan proses inisialisasi LCD, *reader* RC522, keypad, Esp8266, dan komunikasi SPI. Setelah itu, mikrokontroler akan tersambung ke jaringan internet melalui access point yang telah ditentukan. Ketika terdapat *tag* yang didekatkan kepada *reader* maka mikrokontroler akan menerima kode unik *tag* dari *reader* untuk kemudian dikirimkan ke *server* menggunakan perintah HTTP begin. Kemudian dengan perintah HTTP GET mikrokontroler akan menunggu respon dari *server*. *Server* akan merespon dengan jumlah pulsa yang terdapat pada *tag*. Ketika *tag* belum terdapat dalam *database* maka *server* akan mengirimkan nilai -1 sebagai tanda bahwa *tag* belum terdaftar,



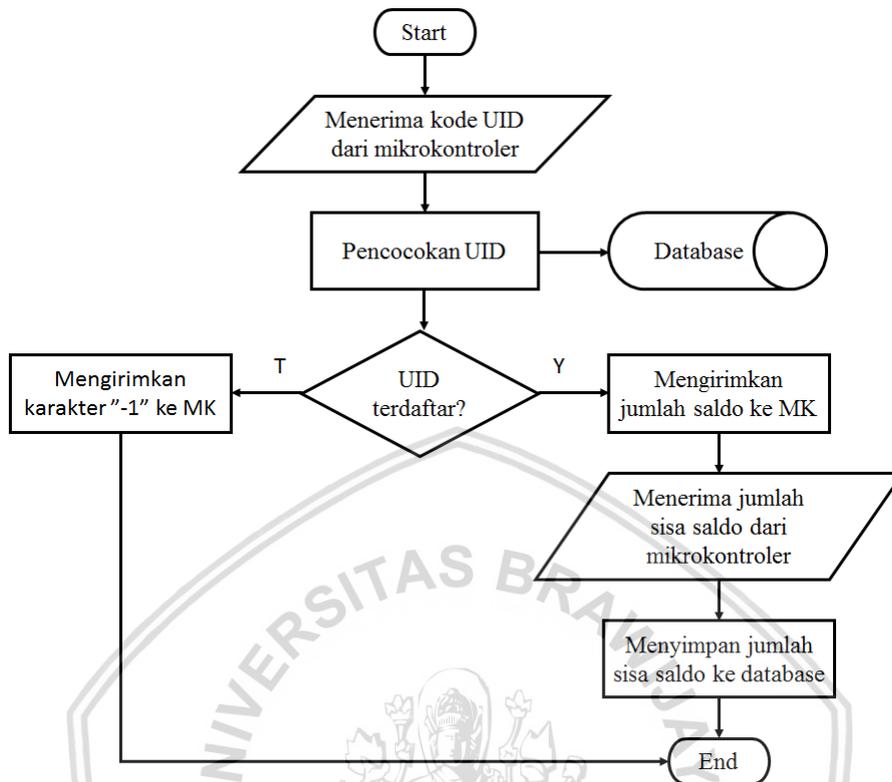
Gambar 3.6 Flowchart program utama mikrokontroler

Proses selanjutnya adalah untuk *tag* yang terdaftar pada *database* yang ingin melakukan proses print, mikrokontroler membaca input dari keypad (hitungan) untuk jumlah kertas yang ingin di print dan menghitung jumlah biaya print. Biaya print disini dihitung dengan mengkalikan jumlah kertas yang di masukkan dengan harga print perlembar. Ketika proses print telah selesai, mikrokontroler akan mengirimkan jumlah pulsa setelah dikurangi biaya print ke *server* dengan menggunakan perintah HTTP begin. Jumlah pulsa yang telah

dikirimkan ke *server* akan disimpan dalam *database*. Tabel 3.1 menjelaskan alokasi pin yang digunakan pada mikrokontroler.

Tabel 3. 1 Alokasi pin mikrokontroller

No	Unit External		Modul Mikrokontroller	Keterangan
	Nama Unit	Pin	Nama Pin	
1	<i>Reader RC522</i>	SS	GPIO15	Pin IRQ tidak digunakan
		SCK	GPIO14	
		MISO	GPIO12	
		MOSI	GPIO13	
		IRQ	-	
		GND	GND	
		RST	GPIO0	
		Vcc	+3,3V	
2	<i>Keypad Matrix 4x4</i>	Row 1	Pin5	-
		Row 2	Pin4	
		Row 3	Pin3	
		Row 4	Pin2	
		Column 1	Pin9	
		Column 2	Pin8	
		Column 3	Pin7	
		Column 4	Pin6	
3	I2C Interface LCD	GND	GND	-
		Vcc	+5V	
		SDA	GPIO4	
		SCL	GPIO5	
4	TCRT5000	Vcc	+5V	-
		GND	GND	
		D0	GPIO16	
5	Relay	Vcc	+5V	
		GND	GND	
		IN	GPIO2	



Gambar 3.7 Flowchart proses print

Sedangkan proses yang terjadi pada *server*, pertama *server* akan menerima kode UID dari mikrokontroler lalu melakukan pencocokan UID dengan *database*. Apabila kode UID terdaftar maka *server* akan mengirimkan jumlah saldo ke mikrokontroler untuk diproses. Setelah pengurangan pulsa dilakukan di mikrokontroler maka sisa saldo akan diterima oleh *server* dan *disimpan* di *database*. Bila UID belum terdaftar maka *server* akan mengirimkan karakter “-1” yang mendandakan bahwa UID tersebut belum terdaftar di *database server*.

3.3. Pengujian Alat

Pengujian dilakukan untuk membuktikan bahwa sistem komunikasi yang dirancang pada penelitian ini bekerja dengan baik. Adapun tahapan dari pengujian tersebut meliputi pengujian pendaftaran ID baru, pengujian pengisian pulsa, dan pengujian pengurangan pulsa.

3.3.1. Pengujian Pendaftaran ID Baru

Pengujian pendaftaran id baru dilakukan dengan proses identifikasi *tag* terlebih dahulu. Kemudian ketika data kartu telah diterima dan ditampilkan pada halaman pendaftaran id, akan dimasukkan nama pengguna pada form pendaftaran. Pengujian ini dilakukan untuk

mengetahui keberhasilan sistem komunikasi pada proses pendaftaran id baru pada sistem. Pada pengujian ini digunakan 6 *tag* yang berbeda-beda. Hasil pengujian membandingkan antara id kartu dan nama pengguna yang diisi pada form identitas dengan id kartu dan nama pengguna yang masuk ke dalam *database*. Apabila data tersebut sama maka dapat dikatakan bahwa sistem komunikasi pada proses pendaftaran id baru dapat bekerja dengan baik.

3.3.2. Pengujian Pengisian Pulsa

Pengujian dilakukan dengan proses identifikasi *tag* terlebih dahulu. Kemudian ketika data kartu telah diterima dan ditampilkan pada halaman pengisian pulsa, akan dimasukkan jumlah pulsa yang ingin diisikan pada form pengisian pulsa. Pengujian ini dilakukan untuk mengetahui keberhasilan sistem komunikasi pada proses pengisian pulsa pada sistem. Pada pengujian ini digunakan 6 *tag* yang berbeda-beda. Tiap *tag* akan diuji dengan pengisian pulsa sebanyak 15 kali. Hasil pengujian membandingkan antara jumlah pulsa didalam *database* sebelum proses pengisian dan jumlah pulsa didalam *database* setelah proses pengisian. Apabila data tersebut sesuai maka dapat dikatakan bahwa sistem komunikasi pada proses pengisian pulsa dapat bekerja dengan baik.

3.3.3. Pengujian Pengurangan Pulsa

Pengujian dilakukan dengan proses identifikasi *tag* terlebih dahulu. Kemudian mikrokontroler akan menghitung jumlah pulsa yang digunakan dalam proses print sesuai dengan jumlah masukan dari keypad. Mikrokontroler akan menghitung jumlah pulsa yang dipakai dan mengirimkan jumlah sisa pulsa ke *server*. Pengujian ini dilakukan untuk mengetahui keberhasilan sistem komunikasi pada proses pengisian pulsa pada sistem. Pada pengujian ini digunakan 6 *tag* yang berbeda-beda. Tiap *tag* akan diuji dengan pengisian pulsa sebanyak 10 kali. Hasil pengujian membandingkan antara jumlah pulsa didalam *database* sebelum proses print dan jumlah pulsa didalam *database* setelah proses print. Apabila data tersebut sesuai maka dapat dikatakan bahwa sistem komunikasi pada proses pengurangan pulsa dapat bekerja dengan baik.

BAB IV

HASIL DAN PEMBAHASAN

Hasil dari penelitian ini diperoleh dari tahapan pengujian. Setelah itu, dilakukan analisis pembahasan dari hasil yang diperoleh. Hasil dari pengujian ini nantinya dijadikan sebagai acuan dalam pengambilan kesimpulan. Adapun tahapan dari pengujian tersebut meliputi pengujian proses pendaftaran ID baru, pengujian proses pengisian pulsa, dan pengujian proses pengurangan pulsa. Pengujian dilakukan dengan menggunakan 6 *tag* yang berbeda.

4.1. Hasil Pengujian Pendaftaran ID baru

Pengujian dilakukan untuk mengetahui bahwa sistem dapat memasukkan data baru ke dalam *database* sesuai dengan perancangan. Pengujian dilakukan dengan cara mendekatkan *tag* ke *reader* seperti pada gambar.



Gambar 4.1 Proses pendekatan *tag* pada *reader*

Setelah *tag* terbaca akan muncul pemberitahuan bahwa *tag* belum terdaftar. *Form* pengisian data akan muncul di PC kemudian memasukkan data nama pengguna. Data UID dan nama pengguna yang dimasukkan akan disimpan di dalam *database*. Tampilan sebelum proses pengujian dapat dilihat pada Gambar 4.2.

Database Pengguna			
ID Kartu	Nama Pengguna	Saldo	Tanggal Daftar

Gambar 4.2 Tampilan form database sebelum pengujian

Tabel 4. 1 Pengujian pendaftaran ID baru

Tag ke-	UID	Tampilan form
1	0415b092	
2	0441a892	
3	045fd992	
4	040d3052	

5	04ce2252	<p>Form Ubah/ Tambah Identitas</p> <p>Identitas Kartu</p> <input type="text" value="04ce2252"/> <p>Nama Pengguna</p> <input type="text" value="kuning"/> <p>Daftar</p>
6	04662652	<p>Form Ubah/ Tambah Identitas</p> <p>Identitas Kartu</p> <input type="text" value="04662652"/> <p>Nama Pengguna</p> <input type="text" value="ungu"/> <p>Daftar</p>

Pada Gambar 4.2 ditunjukkan kondisi *database* sebelum proses pendaftaran ID baru. Sebelum proses pendaftaran ID baru, *database* tidak menyimpan data apapun. Setelah proses pendaftaran ID baru seperti pada Tabel 4.1 maka data ID kartu dan nama pengguna akan disimpan didalam *database*. Terlihat pada Gambar 4.3 *database* telah terisi data sesuai dengan data yang dimasukkan pada form pendaftaran seperti pada Tabel 4.1. Pada *form database* terdapat ID kartu sesuai dengan UID tiap *tag*, nama pengguna yang telah dimasukkan pada proses pendaftaran, jumlah saldo yang dimiliki tiap pengguna dan tanggal pendaftaran tiap ID.

Database Pengguna				
ID Kartu	Nama Pengguna	Saldo	Tanggal Daftar	
0415b092	hijau	Rp 0	10 Jul 2017	Delete
0441a892	hijau tua	Rp 0	10 Jul 2017	Delete
045f9992	coklat	Rp 0	10 Jul 2017	Delete
040d3062	merah	Rp 0	10 Jul 2017	Delete
04ce2252	kuning	Rp 0	10 Jul 2017	Delete
04662652	ungu	Rp 0	10 Jul 2017	Delete

Gambar 4.3 Tampilan form database setelah pengujian

Dari hasil pengujian diatas didapatkan hasil bahwa data ID baru yang didaftarkan tersimpan di dalam *database*. Sistem dapat melakukan proses pendaftaran ID baru dengan baik.



4.2. Hasil Pengujian Pengisian Pulsa

Pengujian dilakukan untuk mengetahui apakah sistem dapat membaca jumlah pulsa dan melakukan pengisian pulsa sesuai dengan jumlah yang ingin diisi dengan tepat. Pengujian dilakukan dengan cara membuka *form* pengisian pulsa lalu mendekatkan *tag* pada *reader* hingga *tag* teridentifikasi. Kemudian memasukkan jumlah pulsa yang ingin diisi dan sistem akan menambahkan pulsa pada *database*. Tampilan *form* pengisian pulsa dapat dilihat seperti pada Gambar 4.4. Pada *form* pengisian pulsa terdapat Identitas kartu yang merupakan kode UID dari *tag* dan nama pengguna yang dimasukkan pada proses pendaftaran. Jumlah pulsa yang akan diisi dapat dimasukkan pada kolom Tambah Saldo.

Gambar 4.4 Tampilan form pengisian pulsa

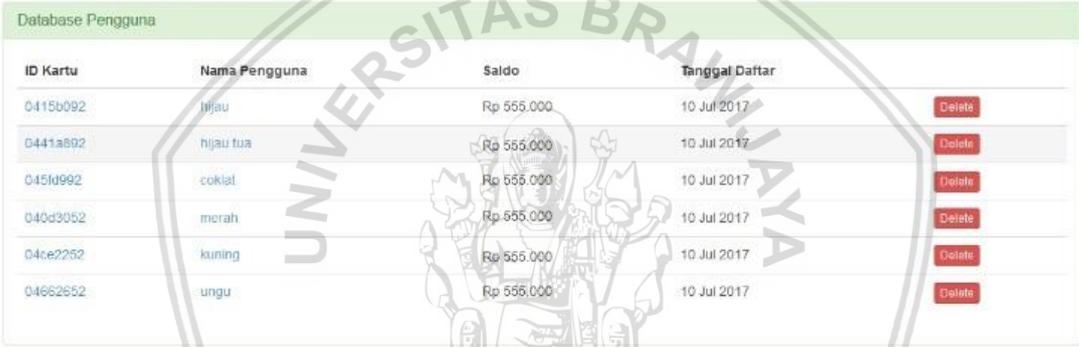
Tabel 4. 2 Hasil pengujian pengisian pulsa

Tag ke-	Nama Pengguna	5.000			10.000			20.000			50.000			100.000			Jumlah pulsa
		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
1	Hijau	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	555.000
2	Hijau tua	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	555.000
3	Coklat	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	555.000
4	Merah	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	555.000
5	Kuning	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	555.000
6	ungu	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	555.000

Keterangan :

√ = pengisian pulsa berhasil

Tabel 4.2 menunjukkan hasil pengujian pengisian pulsa pada 6 tag yang berbeda. Pengisian pulsa dilakukan sebanyak 15 kali dengan rincian 3 kali pengisian pulsa Rp. 5.000, 3 kali pengisian pulsa Rp. 10.000, 3 kali pengisian pulsa Rp. 20.000, 3 kali pengisian pulsa Rp. 50.000 dan 3 kali pengisian pulsa Rp. 100.000. Pengisian pulsa dilakukan bertahap berkelompok harga. Setelah pengisian pulsa Rp. 5.000 sebanyak tiga kali, maka jumlah pulsa pengguna menjadi sebesar Rp 15.000. Pada akhir pengujian pulsa tiap pengguna akan memiliki pulsa sebesar Rp. 555.000. Pada tabel ditunjukkan bahwa seluruh proses pengisian pulsa pada keenam tag berhasil dengan baik. Pulsa yang diisikan tersimpan dalam *database* seperti pada Gambar 4.5. Misalkan untuk nama pengguna Coklat, saldo yang dimiliki sebesar Rp. 555.000.



ID Kartu	Nama Pengguna	Saldo	Tanggal Daftar	
0415b092	hijau	Rp 555.000	10 Jul 2017	Delete
0441a892	hijau tua	Rp 555.000	10 Jul 2017	Delete
0451d992	coklat	Rp 555.000	10 Jul 2017	Delete
040d9052	merah	Rp 555.000	10 Jul 2017	Delete
04ce2252	kuning	Rp 555.000	10 Jul 2017	Delete
04652652	ungu	Rp 555.000	10 Jul 2017	Delete

Gambar 4.5 Tampilan database setelah pengisian pulsa

Dari hasil pengujian didapatkan hasil bahwa pulsa yang ditransaksikan sesuai dengan pulsa yang diisikan ke dalam *database*. Sistem dapat melakukan proses pengisian pulsa dengan baik.

4.3. Hasil Pengujian Pengurangan Pulsa

Pengujian dilakukan untuk mengetahui apakah jumlah pulsa yang digunakan untuk mencetak sesuai dengan jumlah pulsa yang dikurangkan dari *database*. Harga yang ditentukan adalah 500 rupiah/kertas. Pengujian dilakukan dengan cara menentukan jumlah kertas yang akan dicetak lalu mendekatkan tag pada *reader*. Printer akan menyala dan proses akan berakhir saat jumlah kertas yang dicetak sama dengan jumlah kertas yang ditentukan.

Tabel 4. 3 Hasil pengujian pengurangan pulsa

Tag ke-	Nama Pengguna	Jumlah pulsa	Pengujian ke-										Pulsa sesudah proses print
			1	2	3	4	5	6	7	8	9	10	

1	Hijau	555.000	√	√	√	√	√	√	√	√	√	√	516.000
2	Hijau tua	555.000	√	√	√	√	√	√	√	√	√	√	516.000
3	Coklat	555.000	√	√	√	√	√	√	√	√	√	√	516.000
4	Merah	555.000	√	√	√	√	√	√	√	√	√	√	516.000
5	Kuning	555.000	√	√	√	√	√	√	√	√	√	√	516.000
6	Ungu	555.000	√	√	√	√	√	√	√	√	√	√	516.000

Keterangan :

√ = pengurangan pulsa berhasil

Tabel 4.3 menunjukkan hasil pengujian pengurangan pulsa pada 6 tag yang berbeda. Pengurangan pulsa dilakukan sebanyak 10 kali dengan rincian 3 kali pengurangan pulsa Rp. 1.000, 3 kali pengurangan pulsa Rp. 2.000, 2 kali pengurangan pulsa Rp. 5.000, dan 2 kali pengurangan pulsa Rp. 10.000. Setiap pengguna telah memiliki saldo sebesar RP. 555.000. Pengujian dilakukan bertahap berkelompok harga. Setelah pengurangan pulsa sebesar Rp. 1.000 sebanyak tiga kali, maka pulsa pengguna akan berkurang sebanyak Rp. 3.000 menjadi Rp. 552.000. Pada akhir pengujian, setiap pengguna akan memiliki sisa pulsa sebesar Rp. 516.000. Pada tabel ditunjukkan bahwa seluruh proses pengurangan pulsa pada keenam tag berhasil dengan baik. Sisa pulsa yang tersimpan dalam *database* dapat dilihat pada Gambar 4.6. Misalkan untuk nama pengguna hijau memiliki sisa pulsa sebesar Rp.516.000 setelah proses pengujian pengurangan pulsa.



ID Kartu	Nama Pengguna	Saldo	Tanggal Daftar
0415b092	hijau	Rp 516.000	10 Jul 2017
0441a892	hijau tua	Rp 516.000	10 Jul 2017
0451d992	coklat	Rp 516.000	10 Jul 2017
040d3052	merah	Rp 516.000	10 Jul 2017
04ce2252	kuning	Rp 516.000	10 Jul 2017
04662652	ungu	Rp 516.000	10 Jul 2017

Gambar 4.6 Tampilan database sesudah pengurangan pulsa

Dari hasil pengujian didapatkan hasil bahwa pulsa berkurang sesuai dengan harga print yang dilakukan. Sistem dapat melakukan proses pengurangan pulsa dengan baik.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan perancangan dan pengujian yang telah dilakukan selama penelitian ini, dapat ditarik kesimpulan bahwa pemodelan perancangan sistem komunikasi untuk sistem pembayaran dengan menggunakan teknologi NFC ini siap untuk dirancang menjadi sebuah sistem yang utuh dan siap untuk dikembangkan lebih lanjut. Hal ini dapat dibuktikan dengan keberhasilan beberapa tahap pengujian sebagai berikut :

1. Mikrokontroler dapat mengirimkan data kode unik tag ke *server* menggunakan protokol HTTP. *Request* HTTP yang dikirimkan oleh mikrokontroler dapat diterima dan direspon oleh *server*. Setelah mikrokontroler mengirimkan *request* HTTP GET maka mikrokontroler dapat menerima data jumlah saldo yang dimiliki oleh tag setelah mendapatkan respon oleh server. Dengan keberhasilan pada pengiriman data kode unik dan keberhasilan pada penerimaan data jumlah saldo tiap tag, dapat dikatakan bahwa mikrokontroler dapat berkomunikasi dengan *server*.
2. *Web* yang dirancang dapat mengirimkan data ke *server* pada saat proses pendaftaran id baru dan pada saat proses pengisian pulsa. Pada proses pendaftaran id, *web* dapat mengirimkan data nama pengguna dan kode unik untuk disimpan dalam *database*. Pada proses pengisian pulsa, *web* dapat mengirimkan data kode unik dan jumlah pulsa yang akan diisikan untuk disimpan dalam *database*. Dengan keberhasilan pada saat proses tersebut, dapat dikatakan bahwa *web* yang dirancang dapat berkomunikasi dengan *server* dan dapat mengakses *database*.
3. 100 % akurasi pada saat pengisian pulsa setelah pengujian terhadap 6 tag yang berbeda dengan 15 kali pengisian pulsa pada setiap tag. Pulsa bertambah sesuai dengan jumlah yang diinginkan.
4. 100 % akurasi pada saat pengurangan pulsa setelah pengujian terhadap 6 tag yang berbeda dengan 10 kali pengurangan pulsa pada setiap tag. Pulsa berkurang sesuai dengan jumlah saldo yang digunakan.

5.2.Saran

Berikut ini adalah saran untuk penelitian selanjutnya :

1. Mikrokontroler sebaiknya dapat tersambung ke jaringan internet lainnya tanpa harus mengubah program.
2. Sebaiknya diterapkan sistem keamanan pada web dan database.
3. Sebaiknya diterapkan sistem untuk mencegah printer melanjutkan proses mencetak data sebelumnya yang belum selesai dicetak apabila tag baru terdeteksi.



DAFTAR PUSTAKA

- Espressif, 2016. ESP8266EX Datasheet. www.espressif.com (Diakses 25 Juli 2017)
- Heidemann, John.1997. *Modelling the Performance of HTTP Over Several Transport Protocols*. IEEE/ACM Transactions on Networking, Vol. 5, No. 5
- Hidayat, Taufik. 2009. *PayPal untuk Transaksi dan Bisnis Online*. Jakarta: Elex Media Komputindo.
- Hunt, Craig. 1992. *TCP/IP Network Administration, third edition*. O'Reilly Media, Inc.
- Johnston, Lindsay dkk. 2015. *Banking, Finance and Accounting : Concepts, Methodologies, Tools and Applications*. USA: Business Science Reference.
- Karimah, Rohdatul L. 2016. *Rancang Bangun Alat Pengisian Pulsa untuk Sistem Pembayaran Menggunakan Teknologi Near Field Communication*. Skripsi. Malang: Fakultas Teknik, Universitas Brawijaya.
- Kurniawan, Arief. 2017. *Pengembangan Sistem Monitoring Listrik Pada Ruangan Menggunakan NodeMCU dan MQTT*. Skripsi. Malang: Fakultas Ilmu Komputer, Universitas Brawijaya.
- Lammle, Todd. 2012. *CompTIA Network + Study Guide 2nd Edition*. Indianapolis: John Wiley & Sons.
- Mansfield, Niall. 2003. *Practical TCP/IP – Mendesain, Menggunakan dan Troubleshooting Jaringan*. Jogjakarta: Andi.
- NodeMCU, 2014. *NodeMcu Connect Things Easy*. www.nodemcu.com (Diakses 25 Juli 2017)
- Purbo, Onno W. 2001. *TCP/IP-Standard, Desain, dan Implementasi*. Jakarta : PT. Elex Media Komputindo
- RFC 2616. 1999. *Hypertext Transfer Protocol*. R.Fielding.
- RFC 793. 1981. *Transmission Control Protocol*. J. Postel.
- Setyawan, Raden Arief. 2015. *Pengembangan Sistem Presensi Mahasiswa Elektronik Berbasis RFID*. Jurnal EECCIS Vol. 9. Universitas Brawijaya.
- Sritusta Sukaridhoto, 2005, *Panduan belajar Jaringan Komputer*, PENS-ITS , 2007.
- Stallings, William. 2001. "The TCP/IP Protocol Suite, Volume 3". Howard W. Sams.

LAMPIRAN

Program

1. Listing Program Mikrokontroler

```

#include <SPI.h>
#include "MFRC522.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 20, 4);

/* wiring the MFRC522 to ESP8266 (ESP-12)
  RST   = GPIO5
  SDA(SS) = GPIO4
  MOSI  = GPIO13
  MISO  = GPIO12
  SCK   = GPIO14
  GND   = GND
  3.3V  = 3.3V
*/

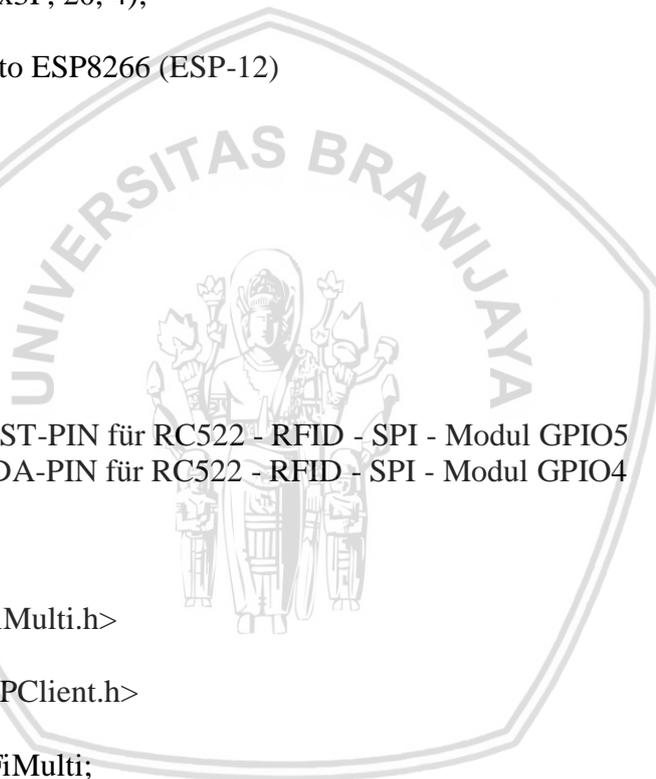
#define RST_PIN 0 // RST-PIN für RC522 - RFID - SPI - Modul GPIO5
#define SS_PIN 15 // SDA-PIN für RC522 - RFID - SPI - Modul GPIO4
#define relay 2
#include <Arduino.h>

#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>

ESP8266WiFiMulti WiFiMulti;
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance
String jumlahprint = "";
int itungan = 0;
String payload;
byte nuidPICC[3];
char setID[10];
int saldo = 0;
int biayaprint = 0;
float hargaperkertas = 500;
int INPUTSENSOR = 16;
int sensorCounter = 0;
int val = 0;
int old_val = 0;
int sisasaldo = 0;

```



```

void setup() {
  lcd.begin();
  lcd.backlight();

  Serial.begin(9600); // Initialize serial communications
  pinMode(INPUTSENSOR, INPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, LOW);
  for (uint8_t t = 4; t > 0; t--) {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
  }

  SPI.begin();
  mfrc522.PCD_Init();
  WiFiMulti.addAP("mahameru 76", "adasyaratnya");

  Serial.println(F("Ready!"));

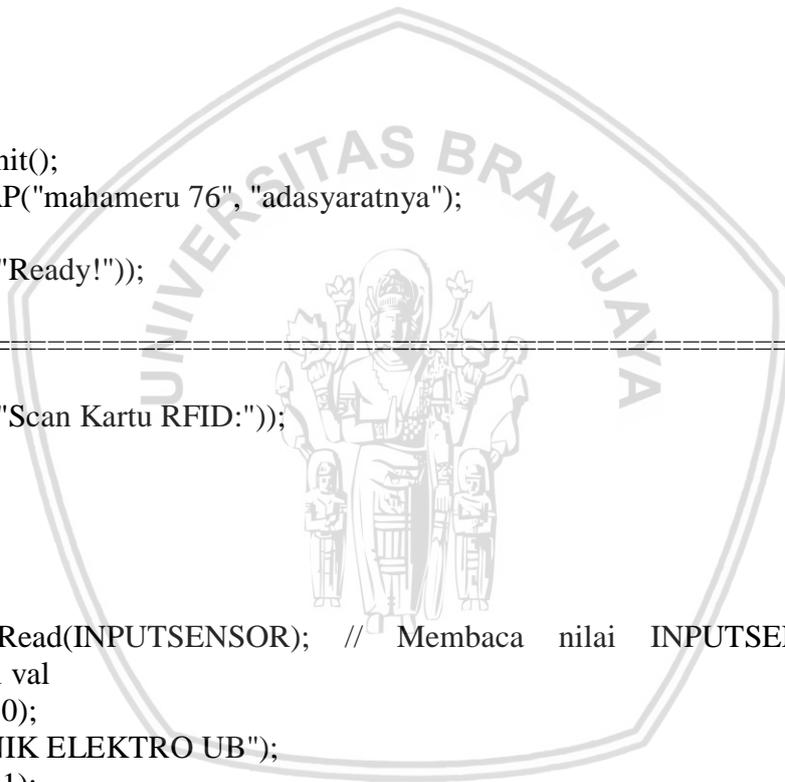
  Serial.println(F("-----"));
  Serial.println(F("Scan Kartu RFID:"));

}

void loop() {

  val = digitalRead(INPUTSENSOR); // Membaca nilai INPUTSENSOR dan
  menyimpannya di val
  lcd.setCursor(1, 0);
  lcd.print("TEKNIK ELEKTRO UB");
  lcd.setCursor(0, 1);
  lcd.print("Scan Kartu Anda....");
  lcd.setCursor(0, 2);
  lcd.print("Jumlah Kertas =");
  lcd.print(itungan);
  lcd.setCursor(0, 3);
  lcd.print("Biaya = Rp.");
  lcd.print(biaya);
  // cek jika terjadi perubahan
  if ((val == LOW) && (old_val == HIGH))
  {
    sensorCounter++;
    Serial.println(sensorCounter);
    delay(200);
  }
}

```



```

if ((sensorCounter) == (itungan) && (sensorCounter) != 0)
{ Serial.println("printer mati ");
  digitalWrite(relay, LOW);
  lcd.setCursor(0, 1);
  lcd.clear();
  lcd.print("Print Selesai...");
  delay(3000);
  lcd.clear();
  itungan = 0;
  biayaprint = 0;
  saldo = 0;
  sensorCounter = 0;
}

```

```

old_val = val; // val sekarang disimpan kedalam old_val
// agar kedepannya perubahan dapat diketahui

```

```

while (Serial.available() > 0) // cek data yang masuk dari pin serial arduino
{
  int inChar = Serial.read();
  if (isDigit(inChar)) {
    // convert the incoming byte to a char
    // and add it to the string:
    jumlahprint += (char)inChar;
  }
  // if you get a newline, print the string,
  // then the string's value:
  if (inChar == '\n') {
    Serial.print("Value:");
    Serial.println(jumlahprint.toInt());
    Serial.print("String: ");
    Serial.println(jumlahprint);
    // clear the string for new input:
    itungan = jumlahprint.toInt();
    biayaprint = itungan * hargaperkertas;
    jumlahprint = "";
  }
}
if (!mfrc522.PICC_IsNewCardPresent()) {
  return;
}
if (!mfrc522.PICC_ReadCardSerial()) {

  return;
}

```

```

for (int i = 0; i < 4; i++) { //
  nuidPICC[i] = mfrc522.uid.uidByte[i];
}
mfrc522.PICC_HaltA();
sprintf(setID, "%02x%02x%02x%02x", nuidPICC[0], nuidPICC[1], nuidPICC[2],
nuidPICC[3]);

```

```
Serial.println(setID);
```

```

if ((WiFiMulti.run() == WL_CONNECTED)) {
  HTTPClient http;
  Serial.println("[HTTP] begin...");
  http.begin("http://rfidprint.tk/index.php/Printer/APIsetCard/MESIN1/" + String(setID));
//perintah untuk mengirim data dari clint (arduino) ke Server
  Serial.println("[HTTP] GET...");
  int httpCode = http.GET();
  if (httpCode > 0) {
    Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    if (httpCode == HTTP_CODE_OK) {
      String payload = http.getString(); // mengambil data saldo dengan tipe data string
      saldo = payload.toInt(); // merubah string menjadi integer
      Serial.println(saldo); // Ganti integer = saldo
      lcd.clear();
      if (saldo > 0)
      {
        lcd.setCursor(0, 1);
        lcd.print("Saldo Anda Rp.");
        lcd.print(saldo);
        delay(3000);
        lcd.clear();
      }
    }
  }
  } else {
    Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
  }
  http.end();
  delay(1000);
}
biayaprint = itungan * hargaperkertas;
if (saldo == -1) // -1 nilai yang didapatmbil dari server karena kartu belum terdaftar
{
  Serial.println("anda belum terdaftar");
  lcd.setCursor(0, 1);
  lcd.print("Anda Belum Terdaftar");
  delay(3000);
  itungan = 0;
}

```

```

    biayaprint = 0;
    saldo = 0;
    lcd.clear();
}
if (saldo > 0 && itungan != 0) // jika saldo ada dan value keypad tidak bernilai 0
{
    if (saldo > biayaprint && saldo != -1)
    { Serial.println("Mulai Print"); // Jika saldo lebih besar dari biaya print dan jika saldo tidak
    = -1, -1 adalah nilai yang dikirimkan oleh server karena ketiadaan data pengguna
    digitalWrite(relay, HIGH);
    sisasaldo = saldo - biayaprint;
    lcd.setCursor(0, 1);
    lcd.print("Mulai Print...");
    delay(3000);

    Serial.println(sisasaldo); // ini sisa saldo yang untuk ngurangin saldo di web server
    HTTPClient http;
    http.begin("http://rfidprint.tk/index.php/Printer/APIinsertTransaction/" + String(setID)
+ '/' + String(biayaprint)); //HTTP perintah untuk mengirim kembali saldo
    Serial.println("[HTTP] GET...");
    int httpCode = http.GET();
    if (httpCode > 0) {
        Serial.printf("[HTTP] GET... code: %d\n", httpCode);

    } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }
}
else {
    Serial.println("Saldo Tidak Cukup");
    digitalWrite(relay, LOW); // Jika saldo tidak cukup relay akan tetap off
    lcd.setCursor(0, 1);
    lcd.print("Saldo Tidak Cukup");
    itungan = 0;
    biayaprint = 0;
    saldo = 0;
    delay(3000);
    lcd.clear();
}
}
}
}

```

2. PHP Halaman Pendaftaran ID

```

<div class="row">
  <div class="col-sm-6 col-sm-offset-3">
    <div class="panel panel-warning">
      <div class="panel-heading">
        <h3 class="panel-title">Form Ubah/ Tambah Identitas</h3>
      </div>
      <div class="panel-body">
        <form id="formGeneral" action="<?php echo base_url(index_page() .
'/Printer/submitCard/'); ?>" method="POST">
          <div class="form-group">
            <label for="id">Identitas Kartu</label>
            <input name="id" type="text" class="form-control" id="id"
placeholder="Scan dengan RFID Reader" disabled>
          </div>
          <div id="usercard">
            <div class="form-group">
              <label for="name">Nama Pengguna</label>
              <input name="name" type="text" class="form-control"
id="name" placeholder="Nama Pemilik Kartu">
            </div>
            <button id="submit" type="submit" class="btn btn-primary btn-
block">Ubah / Daftar</button>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
<script type="text/javascript">
var lastcard = "";

$(document).ready(function() {
  $('#usercard').hide();
  $('.pricelist').priceFormat({prefix: 'Rp ', centsLimit: 0, thousandsSeparator: '.'});

```

```

$('#formGeneral').on('submit', function(e) {
    $("input[type='submit']", this)
    .val("Please Wait...")
    .attr('disabled', 'disabled');

    $("#formGeneral :disabled").removeAttr('disabled');
    $('.pricelist').each(function( index ) {
        $(this).val(parseInt($(this).unmask()));
    });
    return true;
});

var refreshCardReader = setInterval(function() {
function(data) {
    if (data.status != 400) {
        if (data.name != "") {
            $("#id").val(data.name);
            lastcard = data.name;
            $.getJSON("<?php echo base_url(index_page) .
'/Printer/APIgetIdentify/'); ?>" + data.name, function(data2) {
                if (data2.id == 0) {
                    $("#submit").html("Daftar");
                } else {
                    $("#name").val(data2.name);
                    $("#submit").html("Ubah Identitas");
                }
            });
            $("#usercard").show();
            clearInterval(refreshCardReader);
        }
    }
});
}, 2000);

var checker = setInterval(function() {
    if (lastcard != '') {

```

```

function(data) {
    $.getJSON("<?php echo base_url(index_page() . '/Printer/APIgetCard/'); ?>",
        if (data.status != 400) {
            if (data.name != "" && lastcard != data.name) {
                location.reload();
                clearInterval(checker);
            }
        }
    });
}, 1000);
});
</script>

```

3. PHP Halaman Pengisian Pulsa

```

<div class="row">
    <div class="col-sm-7">
        <div class="panel panel-warning">
            <div class="panel-heading">
                <h3 class="panel-title">Tambah Saldo</h3>
            </div>
            <div class="panel-body">
                <form id="formGeneral" action="<?php echo base_url(index_page() .
'/Printer/addBalance/'); ?>{cardid}" method="POST">
                    <div class="form-group">
                        <label for="id">Identitas Kartu</label>
                        <input name="id" disabled type="text" class="form-control"
id="id" placeholder="Scan dengan RFID Reader" value="{cardid}">
                    </div>
                    <a id="userregister" class="btn btn-danger btn-block" href="<?php echo
base_url(index_page() . '/Printer/'); ?>">Daftar Kartu</a>
                    <div id="usercard">
                        <div class="form-group">
                            <label for="name">Nama Pengguna</label>
                            <input name="name" type="text" class="form-control"
id="name" placeholder="Nama Pemilik Kartu" disabled>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="form-group">
            <label for="price">Tambah Saldo</label>
            <input name="price" type="text" class="form-control
pricelist" id="price" placeholder="Jumlah Saldo">
        </div>
        <button type="submit" class="btn btn-primary btn-block">Tambah
Saldo</button>
    </div>
</form>
</div>
</div>
</div>
<div class="col-sm-5">
    <div class="panel panel-info" id="trx">
        <div class="panel-heading">
            <h3 class="panel-title">Catatan Transaksi</h3>
        </div>
        <div class="panel-body">
            <table class="table table-hover table-strip" id="datatable1">
                <thead>
                    <tr>
                        <th>Tanggal Trx</th>
                        <th>Saldo</th>
                    </tr>
                </thead>
                <tbody>
                    {list}
                    <tr class="success g{issell}">
                        <td class="changedate">{created}</td>
                        <td class="pricelist b{issell}">{balance}</td>
                    </tr>
                    {/list}
                </tbody>
                <tfoot>
                    <tr>
                        <th class="text-right">Saldo Akhir</th>
                        <th class="pricelist">{balance}</th>
                    </tr>
                </tfoot>
            </table>
        </div>
    </div>
</div>

```

```
                </tfoot>
            </table>
        </div>
    </div>
</div>
<script type="text/javascript">
    var lastcard = "";

    var month_names_short = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov',
'Dec'];
    function callcheck(card) {
        $.getJSON("<?php echo base_url(index_page()) . '/Printer/APIgetIdentify/'; ?>" + card,
function(data2) {
            if (data2.id == 0) {
                $("#name").val("Daftar Kartu Terlebih Dahulu");
                $('#userregister').show();
            } else {
                $("#name").val(data2.name);
                $('#usercard').show();
                $('#trx').show();
            }
        });
    }
$(document).ready(function() {
    $('#userregister').hide();
    $('#usercard').hide();
    $('#trx').hide();

    $('.pricelist').priceFormat({prefix: 'Rp ', centsLimit: 0, thousandsSeparator: '.'});

    $(".g0").each(function () {
        $(this).removeClass('success');
    });
    $(".b0").each(function () {
        $(this).html('(' + $(this).html() + ')');
    });
});
```

```
$('#formGeneral').on('submit', function(e) {
    $('input[type='submit']", this)
    .val("Please Wait...")
    .attr('disabled', 'disabled');

    $("#formGeneral :disabled").removeAttr('disabled');
    $('.pricelist').each(function( index ) {
        $(this).val(parseInt($(this).unmask()));
    });
    return true;
});

var refreshCardReader = setInterval(function() {
function(data) {
    if (data.status != 400) {
        if (data.name != "") {
            clearInterval(refreshCardReader);
            $("#id").val(data.name);
            lastcard = data.name;
            window.location.href = '<?php echo base_url(index_page() .
'/Printer/transaction'); ?>/' + data.name;
            callcheck(data.name);
        }
    }
    });
}, 2000);

var checker = setInterval(function() {
function(data) {
    if (data.status != 400) {
        if (data.name != "" && lastcard != data.name) {
            window.location.href = "<?php echo base_url(index_page()
. '/Printer/transaction/'); ?>";
            clearInterval(checker);
        }
    }
}
```

```
        });  
    });  
}, 1000);  
  
$(".changedate").each(function () {  
    var d = new Date($(this).html() * 1000);  
    var currentDay = d.getDate();  
    if (currentDay < 10) { currentDay = '0' + currentDay; }  
    var date = currentDay + " " + month_names_short[d.getMonth()] + " " + d.getFullYear();  
    $(this).html(date);  
});  
  
if ('{cardid}' != '') {  
    lastcard = '{cardid}';  
    clearInterval(refreshCardReader);  
    callcheck("{cardid}");  
}  
});  
</script>
```

