

# Purwarupa Sistem Daftar Pelanggaran Lalulintas Berbasis Mini-Komputer *Raspberry Pi*

Rombang Mathew Raphael Clinton, Rizal Sengkey

Teknik Elektro, Universitas Sam Ratulangi Manado, Jl. Kampus Bahu-Unsrat Manado, 95115

Email: mathew.rombang@gmail.com, rizalsengkey@gmail.com

**Abstract** --- *Research aims as prototype system to prevent traffic jam, by preventing vehicles from being stopped in road that has no stop sign, this system will detect vehicles that stop in excessive time determined to included in the list of traffic violations in the database. The prototype design of this system, the Raspberry Pi-based mini-computer uses RFID as identification sensor running, which is connected in database as vehicle for storing existing vehicle data. And as the basis of the system used Raspberry Pi 3 model B, which has been installed Raspbian as a operating system. So this prototype includes infringement data in the database.*

**Keywords** : *mini-computer Raspberry Pi; Prototype of a traffic violation system; Python webserver; RFID MFRC522.*

**Abstrak** — Penelitian ini bertujuan sebagai purwarupa sistem untuk mencegah kemacetan, dengan cara mencegah kendaraan dilarang berhenti dalam jalan yang terdapat larangan berhenti, sistem ini akan mendeteksi kendaraan yang berhenti dalam waktu yang berlebihan ditentukan akan masuk daftar pelanggaran lalulintas dalam database. Dalam rancangan Purwarupa sistem ini berbasis mini-computer *Raspberry Pi* ini menggunakan *RFID* sebagai sensor identifikasi untuk berjalan, yang terhubung dalam database sebagai penyimpan data kelengkapan kendaraan, yang ada. Dan sebagai basis dari sistem digunakan *Raspberry Pi 3 model B*, yang telah dipasang sistem operasi *Raspbian*. Jadi purwarupa ini memasukan data pelanggaran dalam database.

**KataKunci** : *mini-computer Raspberry Pi; Purwarupa sistem pelanggaran lalulintas; python webserver; RFID MFRC522.*

macet di daerah atau negara tersebut. Kendaraan yang berhenti sembarangan pun menambah masalah kemacetan, dibuatnya rambu di sekitar daerah rawan macet pun bukanlah solusi yang sangat bagus bagi masyarakat yang tanggap aturan tanpa adanya penjagaan polisi. Kini taksi online dan kendaraan umum pun membuat daftar panjang masalah kemacetan di beberapa daerah.

Sulawesi Utara merupakan daerah berkembang di Indonesia yang tidak lari dari masalah ini. Dari data Satuan Lalu Lintas (Satlantas) Polresta Manado sampai akhir tahun 2015, tercatat jumlah panjang kendaraan 1.711,250 kilometer (km), sedangkan panjang jalan hanya 518.830 km. Berarti kekurangan panjang jalan 1.192,42 km. Jumlah kendaraan bermotor di Sulawesi Utara (Sulut) terus meningkat, dimana setiap bulan ketambahan kendaraan bermotor hingga 2.000 unit. Dari data Direktorat Lalu Lintas (Ditlantas) Polda Sulut, jumlah kendaraan hingga bulan Agustus mencapai 1.085.987 unit. Menurut Kepala Sub Direktorat (Kasubdit) Registrasi dan Identifikasi Kendaraan Bermotor (Regident) Ditlantas Polda Sulut, AKBP Denny Situmorang SIK, jumlah kendaraan bermotor ada 1.085.987 unit, yang 40% terdaftar di Kota Manado, sedangkan lainnya terdaftar di 13 Samsat Kabupaten/Kota. Jumlah tertinggi dipegang kendaraan roda dua, sebanyak 836.087 unit, secara keseluruhan terjadi penambahan kendaraan bermotor setiap bulan kurang lebih 0,6% kendaraan.

Berdasarkan masalah di atas, penulis akan membuat/merancang suatu alat untuk membantu mengatasi salah satu masalah kemacetan dengan fungsinya mencegah suatu kendaraan menghabiskan waktu dalam suatu ruas jalan yang padat dengan judul “Purwarupa Sistem Pencegahan Kemacetan Berbasis Mini-Komputer *Raspberry Pi*”

## A. *Radio Frequency Identification (RFID)*

*Radio Frequency Identification (RFID)* adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. *RFID* menggunakan frekuensi radio untuk membaca informasi dari sebuah komponen kecil yang disebut tag atau transponder (*Transmitter + Responder*). Tag *RFID* akan mengenali diri sendiri ketika mendeteksi sinyal dari komponen yang kompatibel, yaitu pembaca *RFID (RFID Reader)*. Komponen yang merupakan *Tag* dan *Reader RFID* yang digunakan dalam purwarupa ini. Modul pembaca/penulis *RFID* ini sangatlah praktis untuk digunakan di rangkaian

## I. PENDAHULUAN

Dewasa ini, negara-negara berkembang mengalami perkembangan yang sangat pesat mengejar perkembangan negara maju. Pertambahan dan kemajuan industri kendaraan bermotor salah satu perkembangannya. Dengan adanya kendaraan bermotor membuat kemudahan dan semakin cepatnya aktivitas daerah tersebut. Namun pertambahan kendaraan bermotor tidak sebanding dengan pertambahan ruas jalan yang bisa dilewati kendaraan membuat banyak titik

elektronika. Pada gambar 1 terlihat gambaran cara kerja dari sensor *RFID*.

Sistem *RFID* terdiri dari empat komponen, di antaranya sebagai berikut :

#### 1) *Tag*

*Tag* *RFID* adalah komponen yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari *tag* *RFID* umumnya memiliki memori sehingga *tag* ini mempunyai kemampuan untuk menyimpan data. Memori pada *tag* secara dibagi menjadi sel-sel. Beberapa sel menyimpan *data Read Only*, misalnya serial number yang unik yang disimpan pada saat *tag* tersebut diproduksi. Sel lain pada *RFID* mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan catu daya *tag*, *tag* *RFID* dapat digolongkan menjadi:

##### a. *Tag Aktif*

*Tag* yang catu dayanya diperoleh dari baterai, sehingga akan mengurangi daya yang diperlukan oleh pembaca *RFID* dan *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe *tag* ini adalah harganya yang mahal dan ukurannya yang lebih besar dan lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh *tag* *RFID* maka rangkaiannya akan semakin kompleks dan ukurannya akan semakin besar.

##### b. *Tag Pasif*

*Tag* yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca *RFID*. Rangkaiannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah *tag* hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca *RFID* harus menyediakan daya tambahan untuk *tag* *RFID*.

#### 2) *Pembaca (Reader)*

Pembaca *RFID* adalah penghubung antara software aplikasi dengan antena yang akan meradiasikan gelombang radio ke *tag* *RFID*. Gelombang radio yang diemisikan oleh antena berpropagasi pada ruangan di sekitarnya yang mengakibatkan data dapat berpindah secara wireless ke *tag* *RFID* yang berada berdekatan dengan antena.

#### 3) *Software Aplikasi*

Software Aplikasi adalah aplikasi pada sebuah workstation atau Personal Computer (PC) yang dapat membaca data dari *tag* melalui pembaca *RFID*. Baik *tag* dan pembaca *RFID* dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.[1]

Spesifikasi MFRC 522 Menggunakan teknologi MIFARE Type A 13.56 MHz (ISO/IEC 14443) A/MIFARE mode yang dirilis oleh NXP Semiconductor dengan sistem keamanan berbasis Crypto-1 (pada seri Classic) dan Triple-DES / AES (pada seri DESFire). Spesifikasinya adalah Chipset MFRC522 C ontactless Reader/Writer IC, dengan Frekuensi 13,56 MHz, Jarak pembacaan kartu: < 50mm, Protokol akses SPI (Serial Peripheral Interface) @ 10 Mbps, Kecepatan transmisi RF 424 kbps (dua arah / bi-directional) / 848 kbps (unidirectional), mendukung kartu MIFARE jenis Classic

S50 / S70, UltraLight, dan DESFire, *Framing & Error Detection (parity+CRC)* dengan 64 byte internal I/O buffer, catu daya 3,3 Volt, konsumsi Arus 13-26 mA pada saat operasi baca/tulis, < 80µA saat modus siaga, suhu operasional: -20°C s.d. +80°C dan dengan dimensi: 40 x 50 mm. Gambar 2 terlihat *device* *RFID* jenis dari MFRC 522.

#### B. *Raspberry Pi*

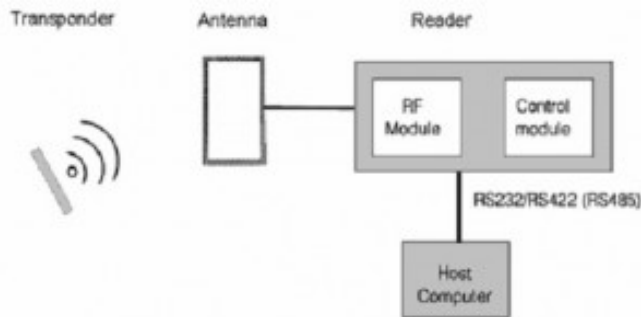
*Raspberry Pi* adalah sebuah komputer papan tunggal (*single-board computer*) atau SBC berukuran kartu kredit. *Raspberry Pi* telah dilengkapi dengan semua fungsi layaknya sebuah komputer lengkap, menggunakan SoC (*System-on-a-chip*) ARM yang dikemas dan diintegrasikan diatas PCB. Perangkat ini menggunakan kartu SD untuk *booting* dan penyimpanan jangka panjang.*Jaringan Sub Transmisi* [2]

##### 1) *Raspberry Pi 3*

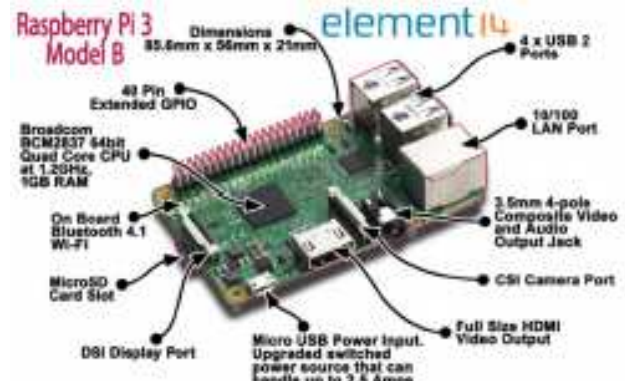
Pada gambar 4 terlihat *Raspberry Pi 3* merupakan generasi ketiga dari keluarga *Raspberry Pi*. *Raspberry Pi 3* memiliki RAM 1GB dan grafis Broadcom VideoCore IV dengan frekuensi *clock* yang lebih tinggi dari sebelumnya yang berjalan pada 250MHz. *Raspberry Pi 3* menggantikan *Raspberry Pi 2* model B pada bulan Februari 2016. Kelebihan *Raspberry Pi 3* dibandingkan dengan *Raspberry Pi 2* adalah: A 1.2GHx 64-bit quad-core ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1, *Bluetooth Low Energy (BLE)* Sama seperti *Pi 2*, *Raspberry Pi 3* juga memiliki 4 USB port, 40 pin GPIO, Full HDMI (*High Definition Multimedia Interface*) port, Port Ethernet, Combined 3.5mm audio jack and composite video, Camera Interface (CSI), Display Interface (DSI), slot kartu Micro SD (Sistem tekan-tarik, berbeda dari yang sebelumnya ditekan-tekan), dan *VideoCore IV 3D graphics core*. *Raspberry Pi 3* memiliki faktor bentuk identik dengan *Raspberry Pi 2* dan memiliki kompatibilitas lengkap dengan *Raspberry Pi 1* dan *2*. *Raspberry Pi 3* juga direkomendasikan untuk digunakan bagi mereka yang ingin menggunakan *Pi* dalam proyek-proyek yang membutuhkan daya yang sangat rendah.GPIO *Raspberry Pi*. Gambar 3 adalah *raspberry pi 3* model B [3]

GPIO merupakan sederet pin yang terdiri dari 40 pin dengan berbagai fungsi. Salah satu fitur yang kuat dari *Raspberry Pi* adalah deretan GPIO (tujuan umum input / output) pin di sepanjang tepi atas pin board. GPIO adalah antarmuka fisik antara *Pi* dan dunia luar. Pada tingkat yang paling sederhana, banyak orang menganggap mereka sebagai *switch* yang dapat mengaktifkan atau menonaktifkan (input) atau bahwa *Pi* dapat mengaktifkan atau menonaktifkan (output). Gambar 4 memperlihatkan GPIO pin dari *RFID* tersebut. [4]

Dari 26 pin GPIO yang dimiliki *Raspberry Pi*, terdapat 2 pin sebagai sumber tegangan 5 V, 2 pin sumber tegangan 3.3 V, 5 pin *ground*, 17 pin input / output. GPIO pada *Raspberry Pi* dapat dikendalikan dan dipicu dengan berbagai cara, bisa dengan terminal menggunakan *bash script* atau dengan bahasa program yang lain. Selain itu dapat memprogram pin untuk berinteraksi dengan cara yang menakjubkan dengan dunia nyata.



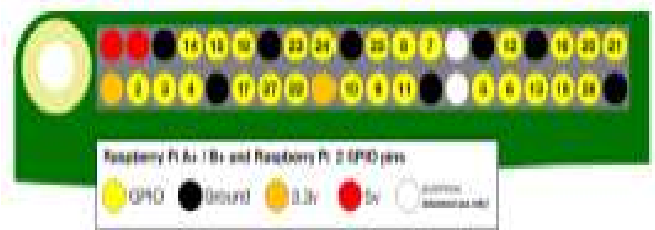
Gambar 1 Sistem RFID



Gambar 3 Tampilan Raspberry Pi 3 Model B



Gambar 2 Sensor RFID jenis MFRC522



Gambar 4 Raspberry Pi GPIO pin

Input tidak harus berasal dari saklar fisik, dapat juga berasal dari sensor atau sinyal dari komputer lain atau perangkat. Pada gambar 5 merupakan gambaran lengkap port GPIO dari Raspberry Pi.[5]

C. Python

Pada awalnya kita akan beranggapan bahwa penamaan bahasa pemrograman ini didasarkan pada nama binatang melata, anggapan tersebut salah. Penamaan bahasa pemrograman ini diilhami ketika pembuatnya menonton acara komedi di televisi di BBC yang bernama Monty Python's Flying Circus. Pembuat bahasa pemrograman ini adalah Guido van Rossum dari Amsterdam, Belanda. Pada awalnya, motivasi pembuatan bahasa pemrograman ini adalah untuk bahasa skrip tingkat tinggi pada sistem operasi terdistribusi Amoeba.

Sejak muncul tahun 1991 di domain public, bahasa pemrograman ini berkembang dengan dukungan komunitas pengguna dan pengembangnya, seperti Python Software Activity, internet newsgroup comp.lang.python, dan organisasi informal lainnya.

Bahasa pemrograman ini menjadi umum digunakan untuk kalangan engineer seluruh dunia dalam pembuatan perangkat lunaknya, bahkan beberapa perusahaan menggunakan python sebagai pembuat perangkat lunak komersial.

Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinan atau mendistribusikannya. Lengkap dengan source code, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya.

Python dapat digunakan dalam beberapa sistem operasi, seperti kebanyakan sistem UNIX, PCs (DOS, Windows, OS/2), Macintosh, dan lainnya. Pada kebanyakan sistem operasi linux, bahasa pemrograman ini menjadi standarisasi untuk disertakan dalam paket distribusinya. [6]

Bahasa pemrograman python mempunyai filosofi antara lain. Coherence. Bahasa pemrograman skrip tidaklah menjadi sulit untuk dibaca, ditulis, dan dimaintain. Power (kekuatan). Bahasa pemrograman ekstensi tidaklah mempunyai fungsi yang terbatas. Scope (jangkauan). Bahasa pemrograman dinamis yang dapat digunakan untuk berbagai macam tugas, dapat menyediakan tanggapan yang cepat selama masa pembuatan sistem aplikasi dan berguna untuk membuat aplikasi tingkat tinggi. Object. Pemrograman berorientasi objek yang efisien, karena mempunyai kekuatan dalam struktur dan penggunaan kembali kode yang telah ada sehingga tidak sulit digunakan. Python didistribusikan dibawah lisensi *OpenSource* yang disetujui *OSI (OpenSource Initiatives)*, sehingga Python bebas digunakan, gratis digunakan, bahkan untuk produk-produk komersil. Yayasan Perangkat Lunak Python – Python Software Foundation (PSF) memegang dan melindungi hak atas kekayaan intelektual dibawah Python, tertuang dalam konferensi PyCon, serta mendanai proyek-proyek pada komunitas Python.

Pengguna Python seringkali menggunakan frasa *“batteries included”* untuk mendeskripsikan pustaka-pustaka standar. Pustaka tersebut mencakup semua dari pemrosesan tidak sinkron kedalam file yang dikompresi. Python sendiri merupakan kumpulan suatu modul-modul yang sangat baik dan dapat menangani secara praktis setiap domain masalah. Kita dapat membuat server web hanya dalam 3 baris kode

saja. Kita juga dapat membangun kode sumber untuk data yang dibangun secara fleksibel menggunakan kemampuan Python dalam hal introspeksi kode sumber. Selain itu Python juga memiliki fitur-fitur bahasa pemrograman tingkat lanjut seperti meta-classes, duck typing, dan decorators.

Integration(integrasi). Tidak ada bahasa pemrograman yang menjadi pulau. Penyediaan dari kekuatan bahasa pemrograman dinamis, dan antarmuka yang terdefinisi dengan baik untuk bahasa pemrograman lainnya, python membantu sistem hibrid yang secara simultan pengungkitan kecepatan perubahan dari python, dan efisiensi dari

Perlu diketahui pula bahasa pemrograman python cenderung menyerupai bahasa pemrograman tradisional seperti *C* dan *Pascal*, daripada bahasa pemrograman skrip seperti *Perl* dan *Tcl*. Pada kenyataannya, *Python* adalah suatu dari bahasa pemrograman skrip dalam samaran dari bahasa pemrograman tradisional. Ada pendapat lain yang menyatakan bahwa Python merupakan jembatan antara bahasa skrip dan *C*.

Beberapa fitur yang dimiliki Python adalah, Memiliki kepastakaan yang luas; dalam distribusi Python telah disediakan modul-modul 'siapa pakai' untuk berbagai keperluan. Memiliki tata bahasa yang jernih dan mudah dipelajari. Memiliki aturan layout kode sumber yang memudahkan pengecekan pembacaan kembali dan penulisan ulang kode sumber. berorientasi obyek. memiliki sistem pengelolaan memori otomatis (garbage collection, seperti *java*) modular, mudah dikembangkan dengan menciptakan modul-modul baru; modul-modul tersebut dapat dibangun dengan bahasa *Python* maupun *C/C++*. Memiliki fasilitas pengumpulan sampah otomatis, seperti halnya pada bahasa pemrograman *Java*, *python* memiliki fasilitas pengaturan penggunaan ingatan komputer sehingga para pemrogram tidak perlu melakukan pengaturan ingatan komputer secara langsung. *IDE* untuk *python* sendiri bisa di dapatkan gratis langsung dari web resminya (<http://python.org/>). Python sampai sekarang sudah realase sampai versi 3.x tapi versi tersebut menggunakan core baru sehingga kebanyakan penggunaannya masih menggunakan versi 2.x dan gw serta tutorial yang bakal gw tulis disini juga pakai versi 2.x. IDE bawaan python namanya *IDLE python*. Setelah program diinstal, ketika dibuka yang pertama muncul adalah jendela interactive shell, interactive shell adalah core awal python, terinstall default pada linux. Seperti yang di singgung di awal, *python* itu interpreter, penterjemahan kode dilakukan perperintah. Dalam interactive shell kita bisa 'mencoba' memasukkan kode apa saja dan hasilnya langsung keluar. Interactive shell berfungsi untuk menampilkan keluaran dari kode yang kita buat. Selain itu, interactive shell juga bisa dipakai untuk mencari kesalahan kode kita, mencoba apakah satu baris kode/perintah yang kita buat akan berjalan semestinya atau tidak. *IDLE python* (Integrated DeveLopment Environment for Python) merupakan Python shell yang dapat digunakan untuk menjalankan perintah-perintah python secara interactive. Aplikasi ini bagus dan berguna terutama untuk yang sedang memulai mempelajari bahasa pemrograman python.

Pada saat menggunakan *IDLE* ini, saatnya dimana membutuhkan cara untuk mengulang command python yang sudah dijalankan sebelumnya (history command). Pada lingkungan shell linux maupun DOS, untuk mengulang

command yang pernah dijalankan sebelumnya caranya cukup sederhana, yaitu dapat dengan menggunakan panah atas (UP Arrow) maupun panah bawah (DOWN Arrow). Khusus pada lingkungan DOS dapat juga dengan menggunakan tombol F7 untuk menampilkan list dari perintah DOS yang sudah pernah dijalankan.

Sedangkan pada *IDLE Python/Python Shell*, cara diatas sayangnya tidak berfungsi, untuk menjalankan ulang command python, cukup arahkan cursor pada baris command yg sudah pernah dijalankan sebelumnya, dilanjutkan dengan menekan ENTER. Maka command pada cursor tersebut akan tercopy dan menjadi command aktif yang siap untuk dijalankan ulang.

atau bisa juga dengan menggunakan shortcut berikut:

ALT+P: History previous command

ALT+N: History next command

7 Elemen Dasar Pemrograman bahasa Python adalah Input berarti membaca sesuatu dari I/O unit, misalkan keyboard. Pada bahasa Python untuk menerima masukan dari pengguna (user), kita dapat menggunakan metode `input()` dan `raw_input()`. Data dapat berupa konstanta, variabel, dan struktur yang berisi bilangan, kalimat, ataupun alamat memori. Penjelasan tentang tipe data ada pada numerik, string, list, tuple, dan dictionary. Operation akan mengubah suatu nilai menjadi nilai lain, mengkombinasikan nilai, dan membandingkan nilai. Yang termasuk operator diantaranya adalah operator Aritmatika dan Assignment. Output berarti menuliskan informasi ke layar, ke disk, atau ke salah satu I/O unit. Untuk menuliskan output program, Python 2.x menggunakan `print`, sedangkan pada Python 3.x menggunakan fungsi `print()`. Conditional Execution berhubungan dengan sejumlah perintah yang akan dijalankan jika kondisi tertentu dipenuhi. Python menggunakan pernyataan `if...elif...else`. Loop berhubungan dengan sejumlah perintah yang akan dikerjakan beberapa kali, selama beberapa kondisi dipenuhi atau sampai suatu kondisi terpenuhi. Python menggunakan pernyataan `for` dan `while` untuk melakukan loop (pengulangan). Subroutine adalah sekumpulan perintah yang bisa dijalankan dari setiap tempat dalam program dengan cara memanggil namanya. Python menyebutnya fungsi atau metode. Tata cara penulisan fungsi atau metode di Python yaitu dengan menggunakan pernyataan `def nama_fungsi()`.

#### 1) Pip

*pip* adalah singkatan dari Pip Installs Python atau *PIP Installs Packages*, kepanjangannya tidak perlu diperhatikan karena memang kalau diartikan terdengar aneh. Bayangkan *pip* adalah sebuah *app store* (atau biasa disebut sebagai *package manager*), kita bisa mencari, menginstall, me-manage modules atau package pada instalasi python kita.

#### 2) Virtual Environment

*Virtualenv* adalah tools untuk membuat lingkungan python virtual yang terisolasi. Terisolasi artinya tertutup dan tidak bisa diakses dari dunia luar. Program Python yang berjalan di dalam *virtualenv* memiliki modul-modulnya sendiri dan program dari luar tidak bisa mengaksesnya. Sedangkan program Python yang berjalan tanpa *virtualenv* hanya bisa menggunakan modul-modul global saja, yang berada di `/usr/lib/python2.7/site-packages`.

Saat menginstal modul/library menggunakan pip, modulnya akan terinstal secara global di `/usr/lib/python2.7/site-packages`. Virtual env pada dasarnya dibuat sebagai tempat pembatas untuk program yang memerlukan modul pada versi yang bersangkutan tanpa mengganggu modul sama dengan versi lain yang terpasang. Karena itu membutuhkan virtualenv, agar masing-masing aplikasi memiliki modulnya sendiri.[7]

#### D. Serial Peripheral Interface

Serial Peripheral Interface (SPI) adalah salah satu protocol komunikasi serial yang sering digunakan karena sangat mudah dan sederhana. Faktanya, komunikasi SPI lebih mudah jika dibandingkan dengan komunikasi USART ataupun UART. Terlihat pada gambar 6 sebagai setup hardware SPI.

Sejak SPI diakui oleh dunia sebagai standar protocol komunikasi, SPI tersedia dan digunakan pada berbagai macam mikrokontroler, seperti: 8051, ARM, PIC, AVR, dll. Hal ini menyebabkan kita dapat menghubungkan dua mikrokontroler dari arsitektur yang berbeda.

SPI banyak diaplikasikan seperti transmisi kabel data, (meski pilihan pertama adalah USART, tapi tidak ada salahnya jika menggunakan SPI karena lebih mudah), transmisi wireless (nirkabel) seperti pada Zigby, 2,4 GHz transmission, dll, pemrograman AVR (upload file \*.hex ke mikrokontroler) juga menggunakan SPI transfer, perangkat komunikasi data dari sensor maupun peripheral device yang lain, seperti: IMU, Joystick, Kompas, dll.

Keuntungan Menggunakan Bus SPI adalah SPI menggunakan empat buah pin untuk berkomunikasi yang nantinya akan dijelaskan dibawah, sedangkan perangkat komunikasi lainnya menggunakan dua/tiga pin seperti I2C, TWI, USART, dll. Berikut adalah Keuntungan menggunakan Bus SPI sangat mudah dalam interfacenya (setup/inisialisasi dan pengiriman data lebih mudah dibandingkan dengan USART atau I2C), komunikasi *Full duplex*, konsumsi daya yang rendah dibanding dengan I2C dan masih banyak lagi, disamping itu, SPI juga memiliki kelemahan antara lain butuh sambungan atau koneksi yang lebih banyak, jadi lebih memakan pin dari mikrokontroler.

##### 1) Master Dan Slave Dalam SPI

Dalam Komunikasi SPI, dua atau lebih *device* yang terhubung yang salah satunya akan bertindak sebagai Master dan yang lainnya akan bertindak sebagai *Slave*.

Master *Device* adalah *device* yang memulai sambungan dan melakukan control transmisi data. Saat kedua *device* sudah terkoneksi, master dapat meminta request data baik mengirim data ataupun menerima data. Seperti pada penjelasan di atas, hal ini dinamakan koneksi *FULL DUPLEX*, yaitu master *device* dapat mengirim data dan juga *slave device* juga dapat mengirim data pada saat yang bersamaan.

Seperti yang terlihat dalam gambar 13 SPI menggunakan empat pin untuk berkomunikasi, berikut ini penjelasan bagaimana PIN dari SPI. SPI biasanya menggunakan empat pin untuk berkomunikasi. Yaitu MOSI, MISO, SCK dan SS.

##### a. MOSI

*Master Output Slave Input*, dimana pin ini digunakan sebagai Master mengirim data dan *Slave* sebagai penerima data.

##### b. MISO

*Master Input Slave Output*, dimana pin ini digunakan sebagai Master bertindak sebagai penerima data dan *Slave* sebagai pengirim data.

##### c. SCK, SPI Clock Line

Berarti *clock* yang digunakan untuk berkomunikasi (Komunikasi synchronous)

##### d. SS

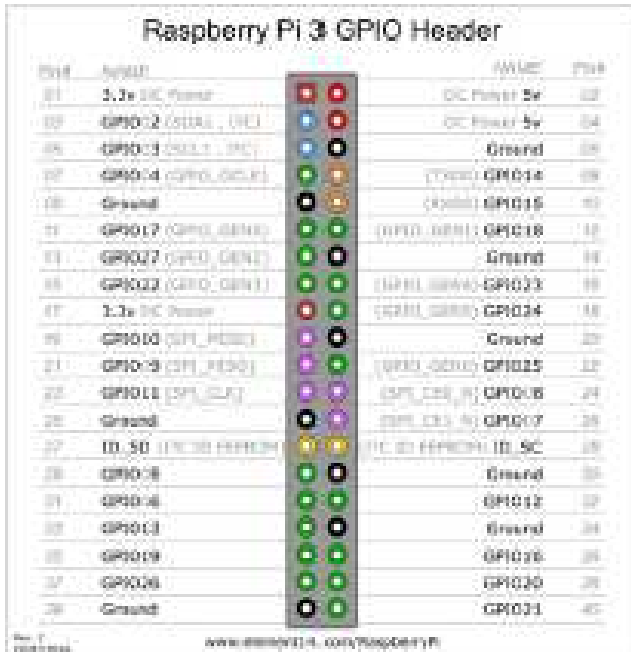
Artinya *Slave Select*. Jika SS bernilai 0 atau *low* maka *Device* bertindak sebagai *slave* dan jika bernilai 1 atau *high* maka *device* bertindak sebagai Master.[8]

#### 2) Transaksi Data Dalam Mode SPI

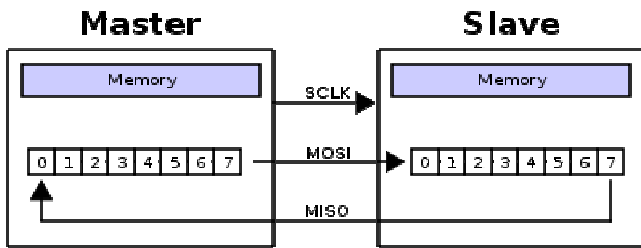
Pertukaran data dalam mode SPI. SPI beroperasi berdasarkan *shift register* baik master *device* maupun *slave device*, keduanya akan mempunyai 8 bit *shift register*. Namun tergantung dari berbagai macam arsitektur mikrokontroler, ada yang bisa memiliki 10 bit ataupun 12 bit *shift register*. Untuk memulai komunikasi, bus master melakukan konfigurasi *clock*, dengan catatan frekuensi atau kecepatan transfer data antara SPI master *device* dan *slave device* harus sama, biasanya bisa mencapai beberapa MHz. Master akan memilih perangkat *slave* dengan mengeluarkan logika 0, lalu master akan menunggu proses yang telah dijadwalkan di master itu sendiri seperti urutan intrupsi timer, konversi analog ke digital (ADC), dll. Lalu setelah periode itu selesai master akan mengeluarkan *clock* yang pertanda akan dimulainya proses komunikasi Serial.

Setiap satu *clock* SPI dilakukan, maka akan terjadi komunikasi *full duplex* antara master *device* dengan *slave device*. Master mengirimkan satu Bit pada line MISO, lalu *slave* akan membacanya. Setelah itu, pada line MISO *slave device* akan mengirimkan data kembali ke master *device* dan master akan membacanya. Urutan atau sekuen ini akan bertahan seperti di atas meskipun kita tidak menggunakan komunikasi *Full Duplex* atau hanya menggunakan satu line komunikasi saja (seperti *simplex*).

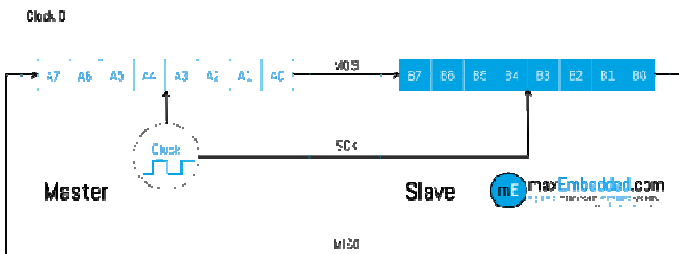
Transmisi data akan melibatkan dua *shift register* dari beberapa ukuran data yang diberikan seperti 8 bit, 10 bit ataupun 12 bit. Namun pada umumnya digunakan 8 bit *shift register*. Keduanya akan terkoneksi dalam topologi ring secara virtual. Data yang dikirimkan biasanya akan bergeser satu per satu dari bit pertama hingga bit kedelapan. Setelah register bergeser keluar, berarti master dan *slave* sudah bertukar data. Lalu selanjutnya akan bergantian *slave* dan master. Jika data yang dikirim banyak, maka *shift register* akan diisi ulang dengan data yang baru. Lalu proses pengirimannya pun diulang. Proses pengiriman akan dihentikan jika master mengirim sinyal toggle untuk mengakhiri pemilihan *slave*. Seperti terlihat pada gambar 7.



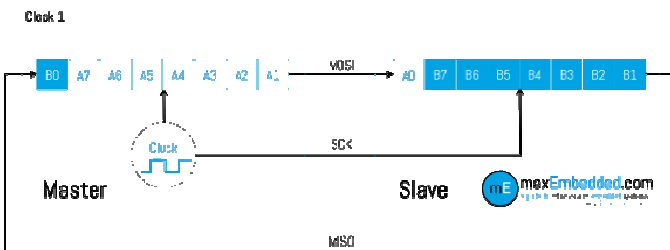
Gambar 5 Raspberry Pi 3 Model B GPIO 40 Pin Block Pinout



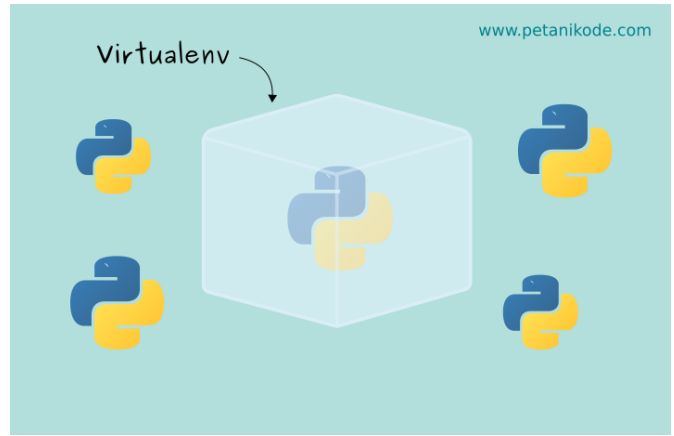
Gambar 6 Setup Hardware SPI



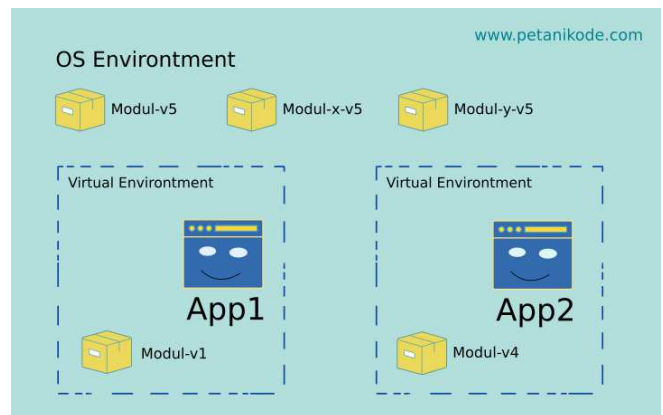
Gambar 7 SPI transfer Data 1



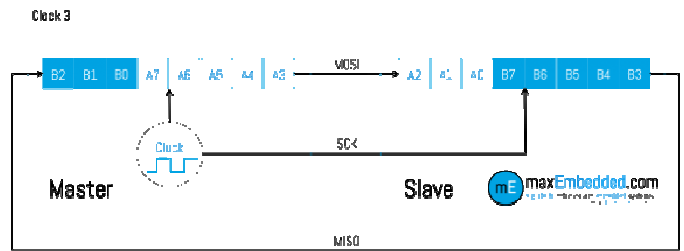
Gambar 8 SPI transfer Data 2



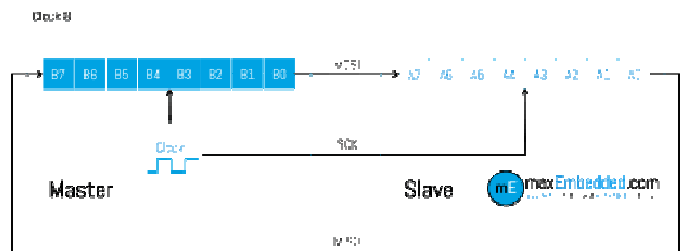
Gambar 9 Gambaran cara kerja virtual env



Gambar 10 Pemisahan modul oleh virtual env



Gambar 11 SPI transfer Data 3



Gambar 12 SPI transfer Data 4

Pada gambar 8, bisa kita lihat saat *clock* dari master memberikan tanda ke *slave device*, *shift register* akan menggeser data di bit A0 dari master, menempati bit ke 7



dari *slave device*. Lalu bit B0 dari *slave device* akan menempati bit ke 7 dari *master device*. Begitulah proses berulang dan terjadi setiap ada *clock* dari master.

Pada gambar ke 11, kita bisa lihat saat *clock* ke 3 dari *master device* yang mengakibatkan *shift register* menggeser nilai dari *Slave* di bit B2 berpindah menempati bit ke 7 dari *master device*. Lalu Bit A2 dari *master device* akan bergeser 1 bit ke bit 7 dari *slave device*. Hal ini akan berjalan sesuai dari perintah *clock* dari master. Data dari tiap bit baik dari master maupun *slave device* akan bergeser 1 bit sesuai dengan *clock* dari master. Selanjutnya kita akan melihat apa yang terjadi saat *clock* mencapai hitungan ke 8.

Bisa kita lihat pada gambar 12, semua data sudah berpindah dari master ke *slave* dan dari *slave* ke master. Hal ini membuktikan bahwa komunikasi SPI adalah komunikasi serial full duplex. Biasanya *clock* akan member tAnda bahwa SPI akan berakhir dan master akan mengulangi untuk memilih *slave device*.

Kedua *device* baik master maupun *slave* akan menempatkan data yang akan ditransfer ke dalam *shift register* mereka sebelum komunikasi serial dimulai.

Master menghasilkan 8 pulsa untuk menggeser nilai setiap bit yang ada pada *shift register* baik *slave* maupun master. Setelah 8 *clock* selesai, master akan memberikan 1 bit informasi sebagai tAnda komunikasi dan sebaliknya dari *slave* ke master.

Setelah 8 *clock* selesai, master akan menerima data dari *slave* yang sudah ada di *shift register* master dan *slave* akan menerima data dari master yang tersimpan di *shift register slave device*.

3) *Multiple Slave SPI Mode*

Seperti yang telah dijelaskan sebelumnya, Bus SPI memungkinkan terdiri dari beberapa *slave device* dan hanya ada 1 master. Namun sangat jarang terjadi mengingat setting *clock* yang kompleks dan sangat susah. Pada *Multiple Slave*, Pin SS lah yang akan berfungsi untuk memilih *slave* mana yang akan menerima dan mengirim data. Pins SS ini mempunyai konfigurasi active

*low*, yang berarti kita harus memberikan pulsa 0 untuk memilih *slave* yang akan kita ajak bertukar data.

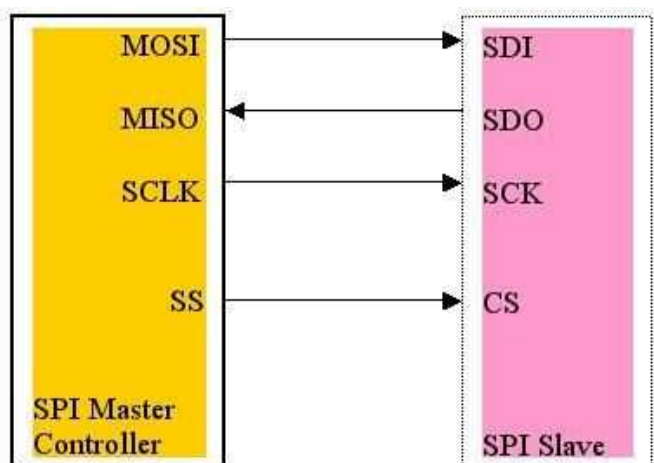
Semua PIN MISO, MOSI dan SCK dari master akan terkoneksi secara parallel ke 3 *Slave device*. Namun ada 3 pin SS dari master yang masing-masing pin nya terpisah dan hanya terkoneksi ke pin SS dari setiap *slave*. Pin SS lah yang akan bergantian memilih *slave* mana yang akan diajak berkomunikasi dengan master. Sebagai contoh, pin SS1 akan memberikan *logic low* ke pin SS dari *Slave 1*. Berarti *Slave device* yang akan bertukar data adalah *slave 1* dengan master, dan seterusnya.

Namun sebagai catatan untuk menghindari tabrakan data, kita tidak boleh memberi *logic low* atau 0 kepada Pin SS1, SS2, SS3 secara bersamaan. Untuk bergantian, kita harus menunggu transfer data telah selesai dari satu *slave*, lalu kita memberikan *logic 1* ke *slave* tersebut yang berarti *slave 1* telah idle. Baru saat komunikasi sudah selesai dengan *slave 1*, maka kita bisa membuat pin SS2 menjadi 0 atau *low* dan memulai komunikasi dengan *slave 2*, dan seterusnya sampai kembali lagi ke *slave 1*. Sesuai gambar 14.[9]

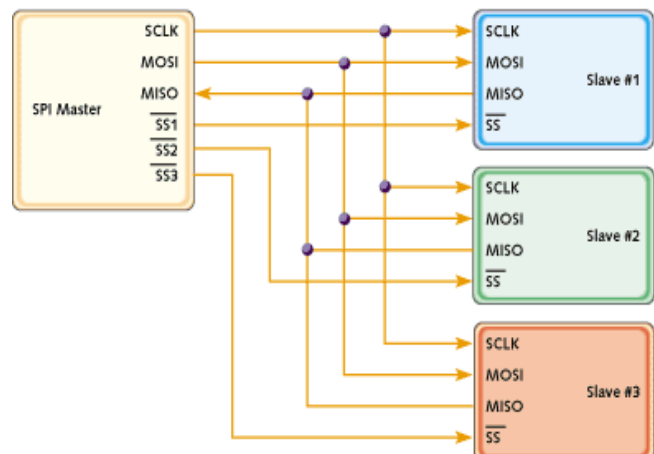
E. *SQLite*

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat ACID-compliant dan memiliki ukuran pustaka kode yang relatif kecil. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. Tidak seperti pada paradigma client-server umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana.

Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi



Gambar 13 single master diagram interface SPI



Gambar 14 SPI bus dengan multiple slave

disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai.

Pustaka SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-93, termasuk transaksi yang bersifat atomic, konsistensi basisdata, isolasi, dan durabilitas (dalam bahasa Inggris lebih sering disebut ACID), trigger, dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe sehingga data bisa dientrikan dalam bentuk string untuk sebuah kolom bertipe integer. Beberapa kalangan melihat hal ini sebagai sebuah inovasi yang menambah nilai guna dari sebuah basisdata, utamanya ketika digunakan dalam bahasa pemrograman berbasis script (PHP, Perl), sementara kalangan lain melihat hal tersebut sebagai sebuah kekurangan.[10]

#### 1) *SQLAlchemy*

SQLAlchemy adalah Python SQL toolkit dan Object Relational Mapper yang memberi pengembang aplikasi kekuatan penuh dan fleksibilitas SQL.

Ini menyediakan rangkaian lengkap pola ketekunan tingkat perusahaan yang terkenal, dirancang untuk akses database yang efisien dan berkinerja tinggi, diadaptasi menjadi bahasa domain yang sederhana dan Pythonic.

Database SQL berperilaku kurang seperti koleksi objek semakin banyak ukuran dan kinerja mulai penting; koleksi objek berperilaku kurang seperti tabel dan baris semakin banyak abstraksi mulai menjadi masalah. SQLAlchemy bertujuan untuk mengakomodasi kedua prinsip ini.

SQLAlchemy menganggap database sebagai mesin aljabar relasional, bukan hanya kumpulan tabel. Baris dapat dipilih dari tidak hanya tabel tetapi juga bergabung dan pernyataan pilih lainnya; salah satu dari unit ini dapat dikomposisikan ke dalam struktur yang lebih besar. Bahasa ekspresi SQLAlchemy membangun konsep ini dari intinya.

SQLAlchemy paling terkenal dengan object-relational mapper (ORM), komponen opsional yang menyediakan pola data mapper, di mana kelas dapat dipetakan ke database secara terbuka, berbagai cara - memungkinkan model objek dan skema database untuk berkembang dalam suatu cara dipisahkan bersih dari awal.

Pendekatan keseluruhan SQLAlchemy terhadap masalah-masalah ini sepenuhnya berbeda dari kebanyakan alat SQL / ORM lainnya, yang berakar pada apa yang disebut pendekatan berorientasi complimentary; alih-alih menyembunyikan SQL dan detail relasional objek di balik dinding otomatisasi, semua proses sepenuhnya terpapar dalam serangkaian alat transparan yang dapat disusun. Perpustakaan melakukan pekerjaan mengotomatisasi tugas-tugas yang berlebihan sementara pengembang tetap mengendalikan bagaimana database diatur dan bagaimana SQL dibangun.[11]

#### F. *Konsep Model Data Entity Relationship Diagram atau ERD*

Entity Relationship Diagram (ERD) untuk mendokumentasikan data perusahaan dengan mengidentifikasi jenis entitas (*entity*) dan hubungannya. ERD merupakan suatu model jaringan yang menggunakan susunan data yang

disimpan pada sistem secara abstrak. ERD juga menggambarkan hubungan antara satu entitas yang memiliki sejumlah atribut dengan entitas yang lain dalam suatu sistem yang terintegrasi. ERD digunakan oleh perancang sistem untuk memodelkan data yang nantinya akan dikembangkan menjadi database. Model data ini juga akan membantu pada saat melakukan analisis dan perancangan database, karena model data ini akan menunjukkan bermacam-macam data yang dibutuhkan dan hubungan antar data.

##### 1) *Entitas (entity)*

Entitas menunjukkan objek-objek dasar yang terkait di dalam sistem. Objek dasar dapat berupa orang, benda, atau hal lain yang keterangannya perlu disimpan dalam database.

##### 2) *Atribut (attribute)*

Atribut sering juga disebut sebagai properti, merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan sebagai database. Atribut berfungsi sebagai penjelas sebuah entitas untuk menggambarkan atribut yang dilakukan dengan mengikuti aturan sebagai atribut dinyatakan dengan simbol ellipse, ama atribut ditulis dalam simbol ellipse, nama atribut berupa kata benda tunggal, nama atribut sebisa mungkin menggunakan nama yang mudah dipahami dan padat menyatakan maknanya dengan jelas, atribut dihubungkan dengan entitas yang sesuai dengan menggunakan garis.

##### 3) *Relasi (relation)*

Relasi atau hubungan adalah kejadian atau transaksi yang terjadi di antara dua entitas yang keterangannya perlu disimpan dalam database.

Variasi relasi terdapat relasi binary merupakan relasi yang terjadi antara 2 himpunan entitas yang berbeda. Relasi ini merupakan relasi yang umum digunakan. Relasi antara mahasiswa mengambil matakuliah yang menunjukkan binary relation. Relasi Unary merupakan variasi relasi yang terjadi dari sebuah himpunan entitas ke himpunan entitas yang sama, dan unary sering disebut dengan relasi tunggal. Relasi N-ary merupakan relasi dari 3 entitas atau lebih. Relasi ini untuk menghubungkan dari tiga entitas yang dimasukkan ke relasi multi entitas.

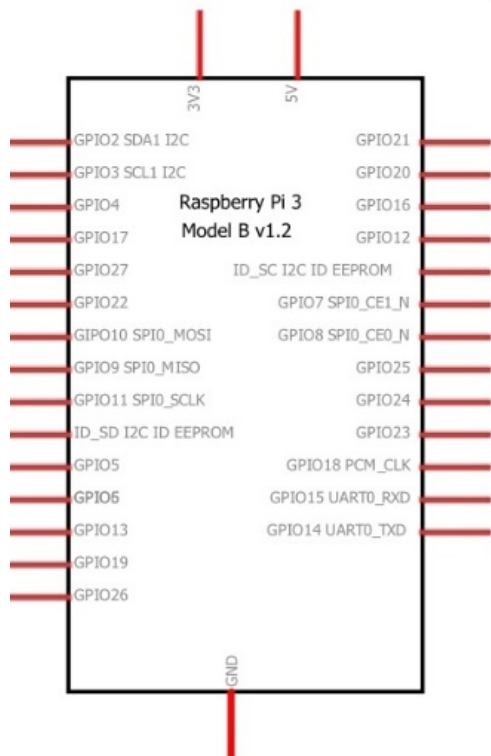
Kardinalitas (Derajat Relasi) Model relasi ini berdasarkan persepsi dunia nyata diantaranya himpunan objek dasar dan relasi antara entitas. Entitas dapat diartikan sebagai objek dan diidentifikasi secara unik, dan objeknya dapat berbentuk orang, barang, dan sebagainya. Kardinalitas relasi menunjukkan maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas relasi yang terjadi di antara dua himpunan entitas dapat berupa satu ke satu (*one to one*), satu ke banyak (*one to many*), dan banyak ke banyak (*many to many*). Hubungan satu ke satu (*one to one*) berarti setiap himpunan entitas hanya boleh berhubungan dengan satu himpunan entitas lainnya. Hubungan satu ke banyak (*one to many*) berarti satu dari setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya. Hubungan banyak ke banyak (*many to many*) berarti setiap himpunan entitas boleh berhubungan dengan banyak himpunan entitas lainnya dan sebaliknya.



G. Flask

Dengan terus melajunya roda perkembangan teknologi, web berkembang menjadi alat bantu yang tidak hanya mampu menyediakan informasi, namun juga mampu untuk mengolah informasi. Proses pengolahan informasi dengan memanfaatkan teknologi web menyebabkan web menjadi media informasi yang dinamis. Hal ini membutuhkan penggunaan sarana teknis yang menentukan. Saranatersebut banyak bergantung pada penggunaan perangkat lunak yang kuat, aman,terpercaya, dan cepat. Perangkat lunak penentu yang dibutuhkan antara lain adalahsebuahserver-side scripting language, atau juga disebut sebagai CGI (CommonGateway Interface), yang banyak tersedia di pasaran seperti: Perl, ASP (Active ServerPage), Cold Fusion, Python, dan PHP.

Flask adalah salah satu web framework dengan menggunakan bahasa python. Flask ini termasuk dalam microframework karena tidak membutuhkan library tambahan maupun aplikasi tertentu. Flask sendiri tidak memiliki lapisan database abstraksi, validasi formulir, atau komponen lain di mana library pihak ketiga yang sudah ada menyediakan fungsi umum. Namun, Flask mendukung ekstensi yang dapat menambahkan fitur aplikasi seolah-olah mereka diimplementasikan dalam Flask itu sendiri. Ekstensi ada untuk pemetaan obyek-relasional, validasi formulir, penanganan unggahan, berbagai teknologi open authentication dan beberapa alat terkait framework. Aplikasi yang menggunakan flask framework termasuk Pinterest, LinkedIn, dan web page community untuk flask sendiri.[12]



Gambar 15 Bentuk Skematik dari Raspberry Pi

II. METODE

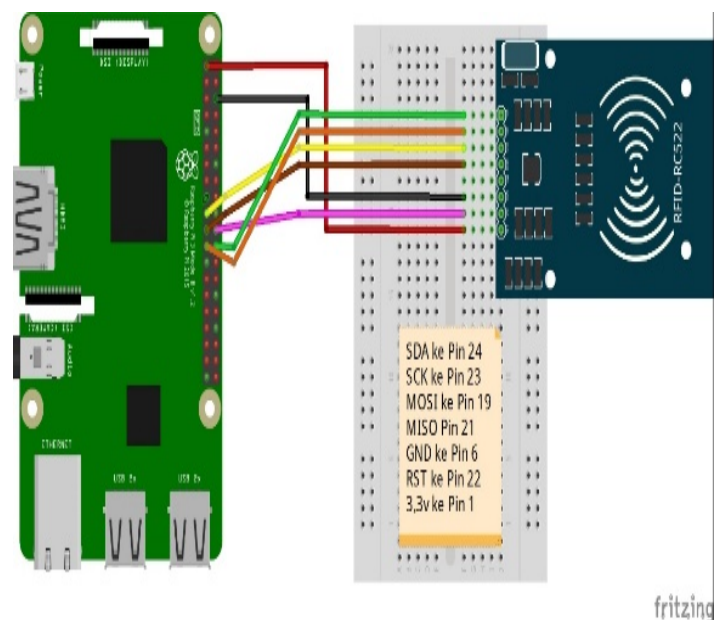
A. Konsep Dasar Perancangan Alat

Pemilihan perangkat keras (hardware) dan perangkat lunak (software) yang merupakan implementasi sistem. Konsep dasar merupakan pedoman untuk merencanakan sesuatu dalam melakukan rancangan (desain), dimana konsep ini memuat langkah-langkah dan petunjuk untuk menentukan sesuatu penunjang yang dibutuhkan dalam mendesain.

B. Desain Purwarupa Pencegah Kemacetan berbasis Raspberry Pi

Dalam membuat desain Purwarupa pencegah kemacetan ini menggunakan aplikasi Fritzing V0.9.3b sebagai aplikasi pembantu. Aplikasi ini memiliki fitur menarik dan menyediakan berbagai komponen lain, dan untuk komponen yang belum tersedia, bisa dicari dalam web forum yang telah disediakan dari Penyedia aplikasi ini.

Dan dengan pembuatan database dipakai SQLite untuk pembuatannya karena software ini mudah digunakan dan kompatibel pada raspberry juga dengan sensor yang digunakan, pada penelitian ini, sensor yang di gunakan adalah RFID. SQLite mempunyai fitur database yang mudah dikelola dan fleksibel, dari fitur yang bisa dengan mudah berjalan dengan sistem versi berapapun, tidak memerlukan konfigurasi banyak, stabil, bisa diperluas dan diintegrasikan dengan berbagai antarmuka pemrograman aplikasi (API), dan sangat cepat. Seperti pada gambar 15 yang merupakan skematik dari Raspberry pi



Gambar 16 Gambar Rangkaian

### 1) Perancangan Perangkat Keras

Pada perancangan perangkat keras ini akan dijelaskan mengenai rangkaian kerja sistem pemindai RFID dengan basis Raspberry Pi dan sistem output.

#### a. Perancangan Sistem Komputer Pengontrol

Pusat Pengontrol alat ini adalah mini-komputer Raspberry pi 3 ver.B. pin pin yang digunakan untuk RFID Reader adalah Pin 1 sebagai catu daya 3.3volt, Pin 22 sebagai Reset, Pin 19 sebagai MOSI, Pin 21 sebagai MISO, Pin 23 sebagai SCK, Pin 24 sebagai SDA dan Pin 6 sebagai grounding. Seperti yang terlihat dalam gambar 16.

#### b. Perancangan RFID Reader

Sensor ini bertugas sebagai pemindai atau reader dari sistem ini, sinyal yang masuk lewat RFID tag yang dipindai akan terbaca dan sebagai identifikasi identitas dari tag tersebut. Dari ID dari tag tersebut akan masuk dalam database jika ID dari tag telah dikenali atau belum.

### 2) Perancangan perangkat lunak

#### a. Instalasi Raspberry pi

*Raspbian* adalah sistem operasi yang digunakan pada *Raspberry Pi* dalam purwarupa tersebut. Sistem operasi ini juga direkomendasikan melalui website resmi *Raspberry Pi*. Cara instalasinya tidak sama dengan cara instalasi sistem operasi pada umumnya seperti di *PC* atau laptop. Karena storage yang digunakan pada *Raspberry* adalah berupa SD card. Flask website dan juga database *SQLAlchemy* digunakan pada sistem ini adalah sebagai penyimpanan data setiap *RFID* yang terdaftar maupun yang terpindai, akses tampilan dari data yang tersimpan.

#### b. Perancangan dan Pembuatan Sistem

Perancangan database dibuat berdasarkan *RFID* Tag yang bersisi nomor identitas dari setiap *RFID*, dari setiap identitas lengkap yang terakses membutuhkan *RFID tag* yang terpindai dalam *RFID* reader untuk setiap masuk, nomor *ID* akan masuk dalam database dan akan mengeluarkan data lengkap beserta presensi waktu setiap kendaraan masuk. *SQLAlchemy* memanfaatkan format database dari *SQLite* sebagai penyimpanan database yang diolah, sehingga menjadi webserver dengan database berbasis python.

## III. HASIL DAN PEMBAHASAN

### A. Pengujian RFID

Pengujian ini dibuat untuk mencoba respon dari RFID Tag dan sensor RFID terhadap mini-komputer, pengujian ini dibuat dengan cara menggunakan tes skrip percobaan, dan dengan cara menempelkan RFID Tag pada sensor RFID. Hasil dari pengujian RFID terbaca didalam terminal Raspberry yang telah diaktifkan skrip untuk percobaan sensor dan tag RFID,

dan diketahui disetiap RFID akan muncul kode unik yang teridentifikasi sebagai identitas dari RFID tag. Adapun karena sensor dan tag yang digunakan adalah jenis MFRC522 maka jarak yang mendukung pembacaan RFID pun terbatas kurang dari 5 centimeter. Jadi dengan begini, Sistem yang dibuat ini adalah purwarupa.

### B. Pengujian secara keseluruhan

Pengujian ini sebagaimana telah dibuat suatu sistem dari purwarupa, dan akan melihat respon dari keseluruhan bagian terhadap sistem yang telah dibuat, pengujian dengan cara memasukan RFID tag pada Sensor RFID tipe MFRC522, untuk melihat respon dari database dan script yang telah dibuat. Perintah berikut untuk mengaktifkan Web Server.

```
flask run --host=0.0.0.0
```

Perhatikan argumen `--host=0.0.0.0` adalah argumen untuk membuat `externally visible server` atau server yang dapat diakses di luar `localhost` sehingga bisa diakses oleh komputer lain.

Dengan adanya gambar 19 bisa diketahui cara registrasi dari setiap *RFID tag* untuk setiap kendaraan.

Dengan melihat Gambar 20, menunjukkan tampilan data dari kendaraan yang telah masuk daftar, tampilan ini merupakan tampilan web yang telah disederhanakan. Data yang termasuk dalam daftar ini dasarnya seperti pada STNK (Surat Tanda Nomor Kendaraan) seperti nomor registrasi, nama pemilik, alamat pemilik, merk kendaraan, tipe, jenis, tahun, isi silinder, nomor rangka, nomor mesin, warna, bahan bakar, warna TNKB (Tanda Nomor Kendaraan Bermotor), tahun registrasi, nomor BPKB (Bukti Pemilik Kendaraan Bermotor), dan kode lokasi, tapi dengan tambahan Nomor ID sebagai nomor identitas RFID tag yang terdaftar dalam 1 kendaraan bermotor.

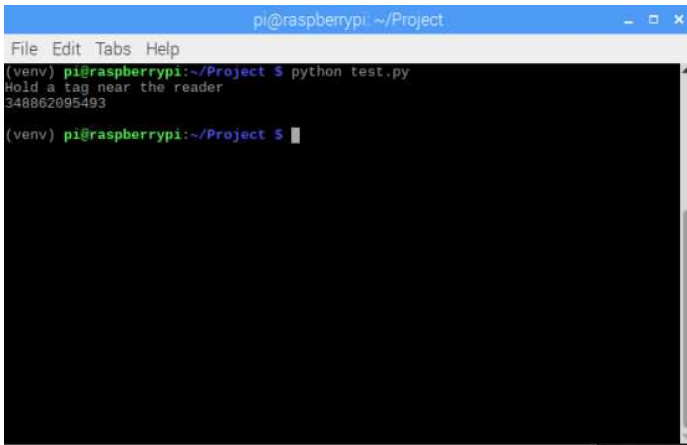
Dari gambar 21 tampilan dari terminal menampilkan respon dari setiap RFID tag mobil yang masuk dan melakukan pelanggaran, setiap RFID yang telah di tampilkan adalah RFID yang telah terdaftar, dan akan menampilkan Waktu pelanggaran, nomor RFID, nomor registrasi atau plat nomor polisi, dan pemilik. Untuk RFID yang belum terdaftar, skrip akan membaca tapi tidak akan menulis pelanggaran, karena belum sama sekali terdaftar dalam database.

Dari gambar 22 ini menampilkan daftar pelanggaran yang terdaftar sesuai waktu pelanggaran, daftar pelanggaran ini merupakan daftar pelanggaran dalam tampilan web sehingga bisa ditampilkan dalam bentuk display yang lebih user-friendly.

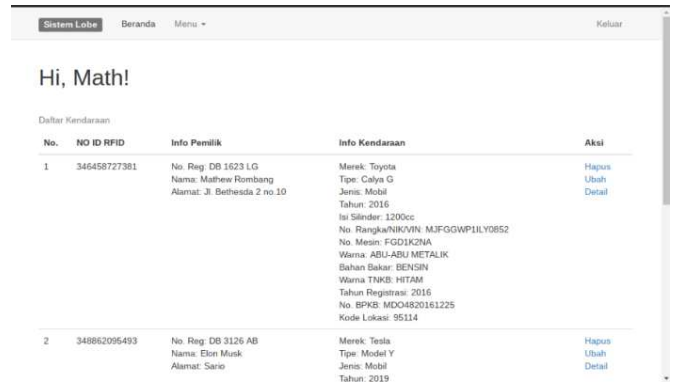
Secara Keseluruhan pengujian yang dibuat telah berjalan sesuai sistem yang dibuat tetapi masih ada beberapa kendala pada hardware maupun software yang memiliki keterbatasan, respon RFID terhadap database kendaraan yang terdaftar telah selaras dengan database pelanggaran, dan ditampilkan dalam daftar yang berupa tampilan web yang didesain sehingga bisa gampang dipahami dan dilihat bagi orang-orang.

## IV KESIMPULAN DAN SARAN

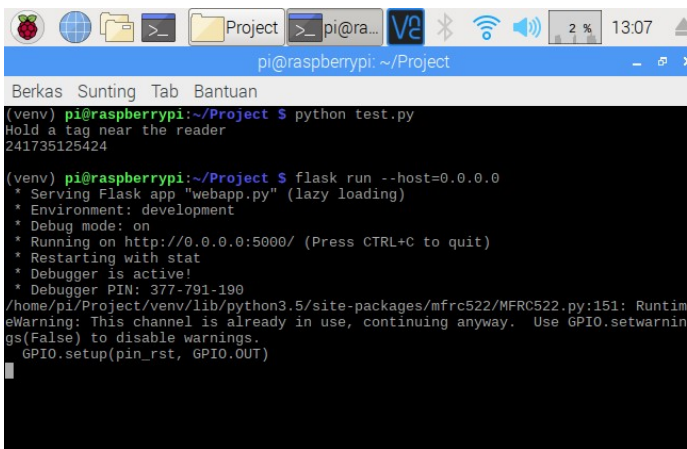
### A. Kesimpulan



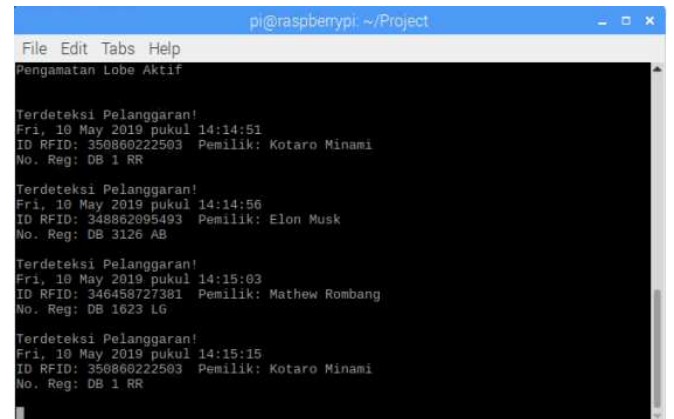
Gambar 17 Pengujian skrip tes beserta RFID tag



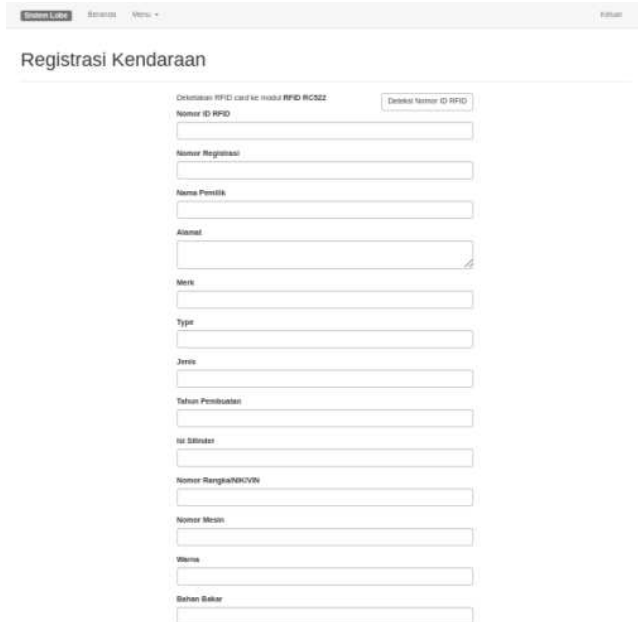
Gambar 20 Tampilan dari kendaraan yang terdaftar dengan RFID



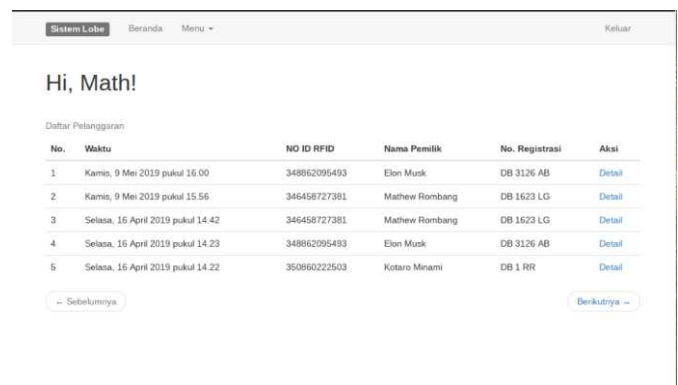
Gambar 18 Tampilan setelah perintah flask run



Gambar 21 Respon skrip terhadap RFID tag yang sudah terdaftar database.



Gambar 19 List Pengisian Registrasi Kendaraan



Gambar 22 Daftar Pelanggaran atau presensi yang terdaftar dalam database

Berdasarkan pengujian dan analisis purwarupa sistem pencegah kemacetan berbasis mini-komputer *Raspberry Pi*, dapat ditarik kesimpulan bahwa *Raspberry Pi* dapat dengan tunggal mengolah database sederhana dan sensor RFID tanpa alat pihak ketiga, hanya dengan library python yang ada. Sistem yang telah dibuat untuk purwarupa ini secara teknis telah berjalan untuk skrip yang telah dibuat, tanpa interruptible. Pengolahan ini bisa sebagai dasar mencegah kemacetan dengan cara mencegah kendaraan berhenti sembarangan dalam area tertentu dalam ruas jalan yang padat kendaraan dan juga terdapat larangan berhenti, adapun bisa sebagai alat yang mempermudah petugas dalam mengatur area-area padat kendaraan..

### B. Saran

Purwarupa sistem pencegah kemacetan berbasis mini-komputer *Raspberry Pi* ini adalah alat yang sangat berguna bagi lalu lintas kota yang rawan macet, namun alat ini masih memiliki beberapa kekurangan terutama pada sensornya yang masih terbatas, port SPI yang terdapat pada raspberry masih belum mendukung multiple *slave* karena perlu adanya tambahan komponen luar untuk menangani sistem SPI multiple *slave*. Untuk pengembangan seterusnya kedepannya disarankan agar menggunakan sensor dengan tipe yang lebih mumpuni dan lebih baik dalam hal akurasi dan jarak jangkauan yang lebih panjang. Juga dalam pengembangan skrip maupun database masih dalam tahap eksperimental jangka panjang.

## V. KUTIPAN

- [1] A. Purnama, "Pengertian Dan Komponen Radio Frequency Identification (RFID)," 2019. [Online]. Available: <http://elektronika-dasar.web.id/pengertian-dan-komponen-radio-frequency-identification-rfid/>. [Accessed: 27-Apr-2019].
- [2] M. R. Nurkamiden, M. E. I. Najoan, and M. D. Putro, "Rancang Bangun Sistem Pengendalian Perangkat Listrik Berbasis Web Server Menggunakan Mini PC Raspberry Pi Studi Kasus Gedung Fakultas Teknik Universitas Sam Ratulangi," vol. 11, no. 1, 2017.
- [3] S. T. Wibisono, D. Darjat, and S. Sudjadi, "IMPLEMENTASI TEKNOLOGI RASPBERRY PI DAN RFID PADA PIRANTI PENYIMPANAN 'DEPOSIT BOX,'" *Transient J. Ilm. Tek. Elektro*, vol. 4, no. 1, pp. 134–140, Jun. 2015.
- [4] I. T. Baskoro, D. Darjat, and S. Sudjadi, "PERANCANGAN PENGONTROLAN NYALA LAMPU DAN KIPAS ANGIN PADA SEBUAH RUANGAN MENGGUNAKAN RASPBERRY PI MODEL B DENGAN WEB GUI," *Transient J. Ilm. Tek. Elektro*, vol. 3, no. 4, pp. 567–571, Feb. 2015.
- [5] I. Taufiqurrohman, N. Widiyasono, and H. Mubarak, "Pemanfaatan Raspberry Pi untuk Hacking dan Forensic dengan metode NIST ( National Institute of Standards and Technology )," *J. Tek. Inform. dan Sist. Inf.*, vol. 3, pp. 231–244, 2017.
- [6] A. Muhardian, "Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal," 2018. [Online]. Available: <https://www.petanikode.com/python-linux/>.
- [7] A. Muhardian, "Mengenal Virtualenv: Apa Saja yang Harus Kamu Ketahui?," 2017. [Online]. Available: <https://www.petanikode.com/python-virtualenv/>. [Accessed: 08-Feb-2019].
- [8] F. Husain, F. Dzulqornain, J. Anthony, and A. Nur, "Serial Peripheral Interface (SPI) pada AVR." [Online]. Available: <http://www.insinyoer.com/serial-peripheral-interface-spi-pada-avr/>. [Accessed: 03-Mar-2019].
- [9] F. Husain, F. Dzulqornain, J. Anthony, and A. Nur, "Dasar-Dasar

- Serial Peripheral Interface (SPI) Mikrokontroler," 2015. [Online]. Available: <http://www.insinyoer.com/dasar-dasar-serial-peripheral-interface-spi-mikrokontroler/>. [Accessed: 02-Jan-2019].
- [10] "About SQLite." [Online]. Available: <https://www.sqlite.org/about.html>. [Accessed: 27-Apr-2019].
- [11] "SQLAlchemy - The Database Toolkit for Python." [Online]. Available: <https://www.sqlalchemy.org/>. [Accessed: 08-Jan-2019].
- [12] D. Lord, "Flask | The Pallets Projects." [Online]. Available: <https://palletsprojects.com/p/flask/>. [Accessed: 08-Dec-2018].



Penulis bernama Mathew Raphael Clinton Rombang, anak kedua dari dua bersaudara. Anak dari Arnold Robby Rombang ( Ayah ) dan Ida Triangka Subagyo ( Ibu ). Lahir di Manado, tanggal 1 Mei 1995. Saya Mulai menempuh Ssekolah Dasat RK 13 St. Paulus Manado (2000-2006). Setelah itu saya melanjutkan pendidikan Sekolah Menengah Pertama Katolik Pax Christi Manado (2006-2009). Selanjutnya saya menempuh pendidikan ke Sekolah Menengah Atas Negeri 1 Manado (2009-2012). Pada Tahun 2012 saya melanjutkan pendidikan S1 di Program Studi Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sam Ratulangi. Pada tahun 2014 saya mengambil konsentrasi minat Sistem Komputer, Penulis melaksanakan kerja praktek di PT. Telkom Manado selama 3 bulan lamanya. Kemudian penulis melaksanakan Kuliah Kerja Terpadu di desa Matabulu Timur kecamatan Nuangan Kabupten Bolaang Mongondow Timur. Selama berada di bangku kuliah penulis tergabung dalam organisasi kemahasiswaan yaitu Himpunan Mahasiswa Elektro (HME), Unit Pelayanan Kerohanian (UPK) Kristen Fakultas Teknik Unsrat, menjadi bagian dari Lab Sistem Komputer. Aktif mengikuti organisasi dan kegiatan non akademik di luar maupun dalam kampus.