

BAB III METODE PENELITIAN

Penyusunan proposal ini didasarkan pada permasalahan yang telah dijelaskan diatas, yakni perencanaan dan perealisasiian alat agar dapat bekerja sesuai dengan yang telah dijelaskan pada rumusan masalah. Langkah-langkah yang perlu dilakukan dalam perealisasiian meliputi: spesifikasi alat, perancangan alat, pengujian alat, dan pengambilan kesimpulan.

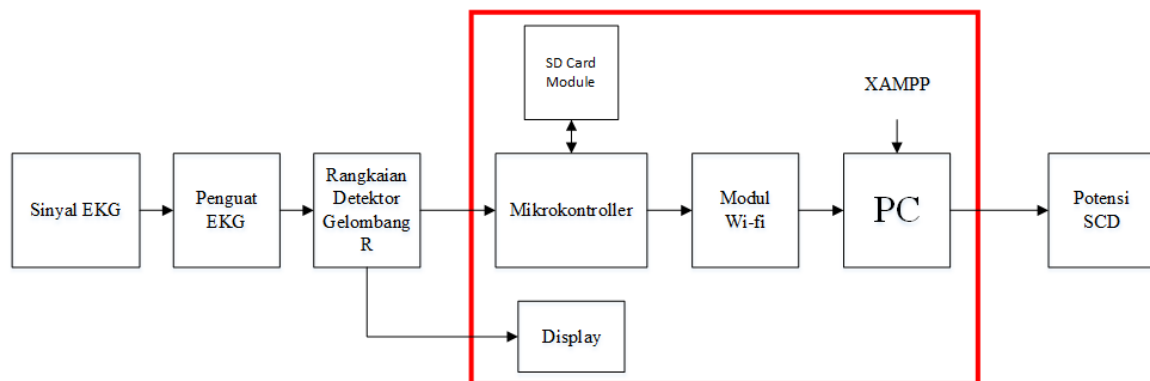
3.1. Spesifikasi Alat

Spesifikasi alat pada perancangan ini ditentukan terlebih dahulu agar didapatkan hasil yang sesuai dan alat bekerja dengan efektif. Adapun spesifikasi alat terdiri dari:

1. Arduino Mega 2560 sebagai mikrokontroler untuk mengolah data masukan sistem dengan Memori *Flash* 256 KB, 8 KB telah digunakan untuk *bootloader*, EEPROM 4 KB, dan *Clock speed* 16 Mhz.
2. *Display* LCD TFT dengan ukuran layar 2.4 *inch* diagonal, resolusi tampilan 240x320 *pixels*, dan kedalaman 65K.
3. Modul SD *Card* dengan kartu *Micro SD* 2GB.
4. Modul ESP8266 sebagai modul *wifi* dengan spesifikasi 802.11 b/g/n, mendukung protokol TCP/IP, *Wifi* 2.4 GHz, mendukung WPA/WPA2, Mendukung mode operasi STA/AP/STA+AP, dan tahan pada suhu -40°C ~ 125°C.

3.2. Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Blok diagram keseluruhan alat ditunjukkan pada Gambar 3.1 dan blok diagram penelitian ini ditandai dengan kotak berwarna merah.



Gambar 3. 1 Blok Diagram Perancangan

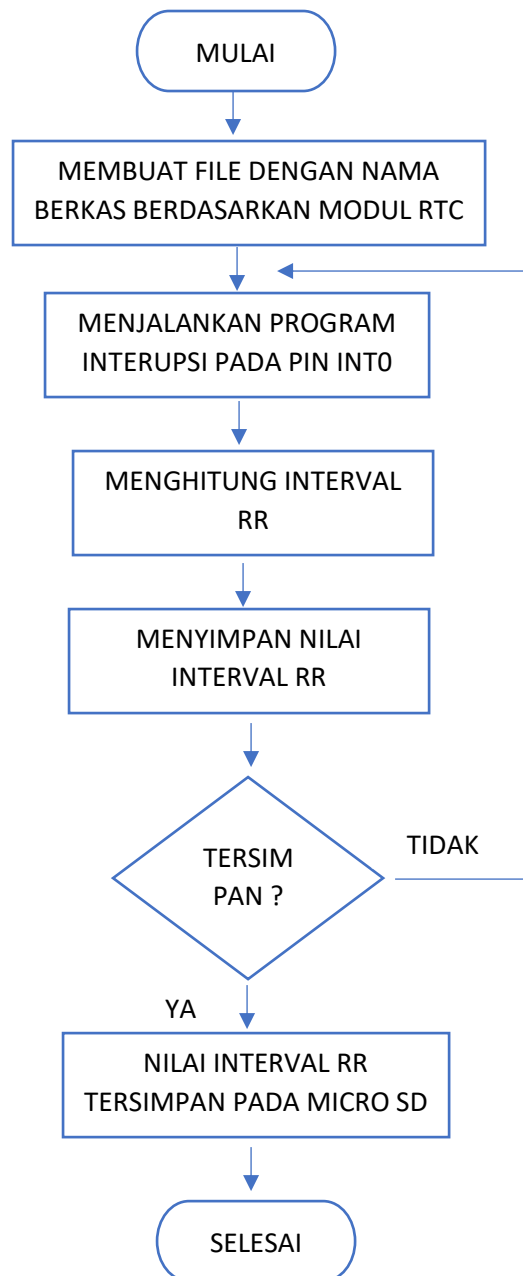
Pada gambar diatas terdapat 5 bagian utama yang memiliki fungsinya masing-masing, diantaranya:

1. *Display*: sebagai penampil sinyal keluaran rekaman EKG.
2. Mikrokontroler: sebagai pengolah data sinyal rekaman EKG agar didapatkan hasil komputasi berupa nilai interval RR.
3. *SD Card*: sebagai tempat penyimpanan data hasil komputasi pada Arduino.
4. Modul *Wifi*: sebagai pentransmisi data yang telah disimpan pada micro SD ke *database*.
5. PC: sebagai *server* untuk menyimpan data yang telah ditransmisikan oleh modul Wifi ke *database* yang telah disediakan.

3.3. Pengolahan Data

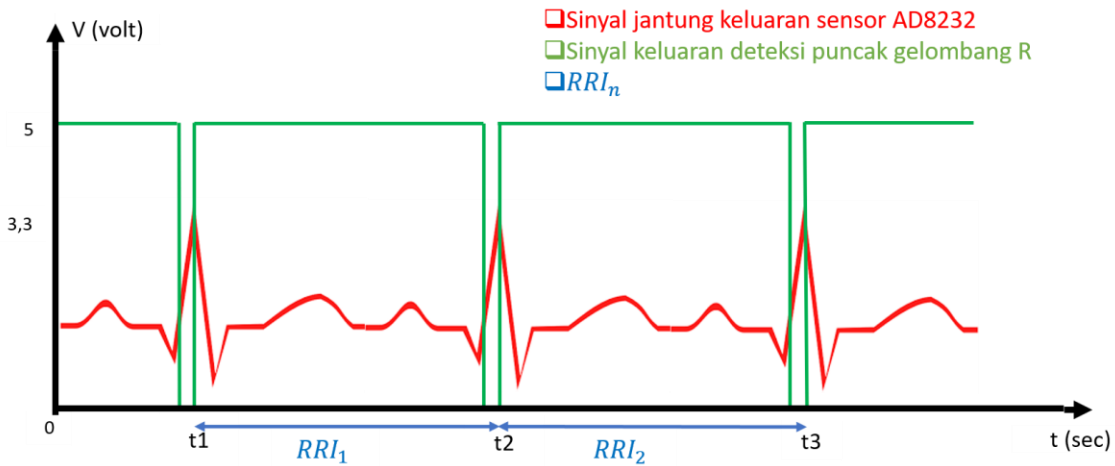
3.3.1. Komputasi Interval RR

Sinyal keluaran Elektrokardiograf berupa sinyal jantung P, Q, R, S, dan T. Elektrokardiograf yang digunakan dalam perancangan ini merupakan sebuah elektrokardiograf yang telah dikondisikan dengan suatu rangkaian khusus sehingga mengeluarkan keluaran berupa gelombang R saja. Gelombang R keluaran elektrokardiograf dimasukkan ke dalam mikrokontroler Arduino pin INT0 agar didapatkan nilai R dalam satuan waktu. Diagram alir pengolahan data dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Diagram alir perancangan pengolahan data

Blok diagram pada Gambar 3.1 menunjukkan bahwa masukan sistem berupa sinyal R yang telah dikondisikan oleh rangkaian deteksi puncak gelombang R. Oleh karena itu, masukan sistem berupa gelombang R dalam interval waktu. Sinyal rekaman EKG ini lah yang akan diproses pada mikrokontroler agar didapatkan nilai interval RR sebagai hasil pengolahan data yang nanti nya akan dikirim pada proses selanjutnya. Ilustrasi sinyal rekaman EKG dari sensor AD8232 dan rangkaian deteksi puncak gelombang R dapat dilihat pada Gambar 3.3.



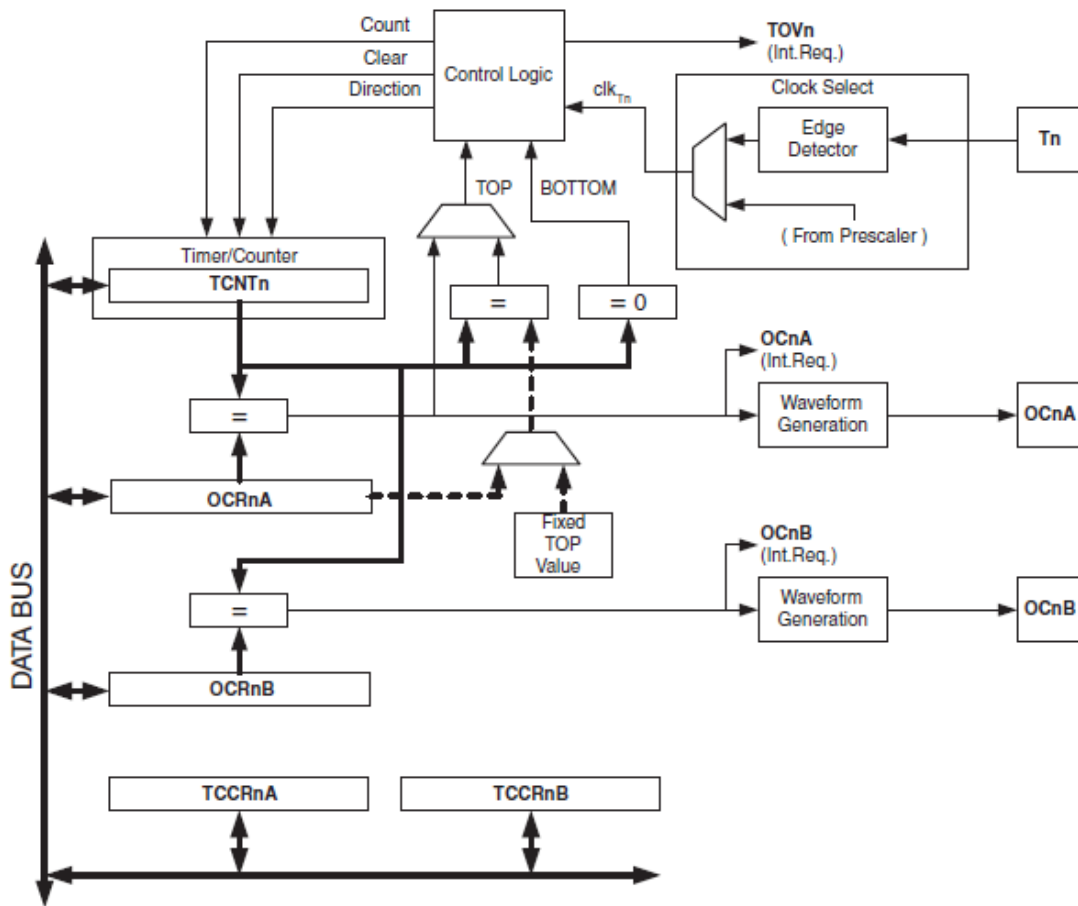
Gambar 3. 3 Ilustrasi sinyal keluaran deteksi puncak gelombang R

Setiap sinyal R yang terdeteksi akan ditandai dengan kenaikan tepi sinyal (*rising edge*). Oleh karena itu, setiap *rising edge* ditandai dengan variabel t yang merepresentasikan waktu puncak gelombang R terjadi. Setelah posisi setiap gelombang R dalam waktu diperoleh, maka dicari nilai interval setiap gelombang R yang muncul. Interval RR didapatkan dari selisih antara gelombang R yang berdekatan, yang dilakukan secara berurutan pada setiap gelombang R. Nilai interval RR (RRI) dapat diselesaikan dengan model matematis:

$$RRI_n = (t_{n+1} - t_n) \dots\dots\dots(3-1)$$

dengan RRI_n merupakan nilai interval RR ke n, t_n adalah waktu kejadian posisi puncak gelombang R ke n, dan t_{n+1} merupakan waktu kejadian posisi puncak gelombang R ke n+1.

Sinyal keluaran EKG dimasukkan ke pin Interrupt INT0 pada Arduino, sehingga masukan akan dijalankan fungsi *interrupt* berupa *rising edge*. Untuk menghitung interval RR, maka perlu dilakukan perhitungan berdasarkan timer/counter pada mikrokontroler yang dipicu oleh interupsi eksternal dari *rising edge* setiap sinyal R yang terekam. Blok diagram timer/counter yang digunakan pada penelitian ini dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Blok Diagram Timer/Counter0 8-bit

Komponen utama pada fungsi *timer/couter* mikrokontroler avr ialah *register* yang akan menghitung mulai dari 0 sampai batas maksimumnya. *Register* avr ini disebut TCNT. Penelitian ini menggunakan *timer/couter0* 8-bit, sehingga register TCNT nya ialah TCNT0. Ada beberapa hal yang perlu diatur untuk menggunakan TCNT0, diantaranya:

3.3.1.1. Register TCCR0 (Timer/Counter Register 0)

Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

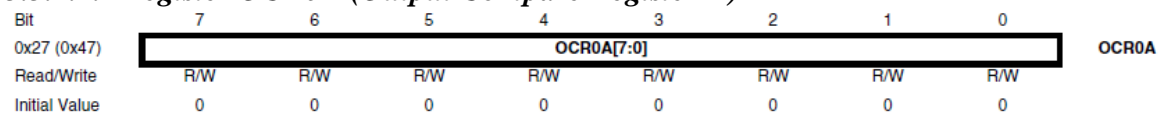
Register TCCR0 diperlukan untuk mengatur *prescaler divider*. *Prescale* digunakan untuk menentukan batas maksimal cacahan yang dilakukan oleh *counter* setiap detik nya. Pada register TCCR0, untuk mengatur nilai *prescaler* yang digunakan, diperlukan pengaturan nilai *clock select bit* (CS00, CS01, dan CS02) berdasarkan *prescaler* yang digunakan. *Clock select bit* yang perlu diatur untuk pemilihan *prescaler* dapat dilihat pada Tabel 3.1.

Tabel 3. 1 *Clock Select Bit Description*

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk _{I/O} /(No prescaling)
0	1	0	clk _{I/O} /8 (From prescaler)
0	1	1	clk _{I/O} /64 (From prescaler)
1	0	0	clk _{I/O} /256 (From prescaler)
1	0	1	clk _{I/O} /1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge
1	1	1	External clock source on T0 pin. Clock on rising edge

Mikrokontroler Arduino yang digunakan memiliki *clock* 16Mhz. *Timer/counter* yang digunakan pada penelitian ini ialah 8-bit *Timer/Counter0* (TCNT0) dengan *prescale* 8. Oleh karena itu dari tabel di atas, nilai CS02=0, CS01=1, dan CS00=0.

3.3.1.2. Register OCR0A (Output Compare Register A)



Nilai TCNT0 akan terus naik sampai dengan nilai maksimumnya dan kembali lagi ke 0. Begitu seterusnya selama diberikan sinyal clock. Nilai OCR diatur untuk pembentukan sinyal PWM. Jika nilai TCNT > OCR maka OC akan memberikan sinyal *high*, begitu sebaliknya. Sistem akan mencacah berdasarkan frekuensi PWM dari persamaan berikut:

$$f_{OCn \times PWM} = \frac{f_{clk_I/O}}{256 \times N} \dots\dots\dots(3-2)$$

dengan f_{clk_I/O} merupakan frekuensi yang didapat dari *clock* Arduino yakni 16Mhz, N ialah *prescaler* yang digunakan, yakni 8. Dari persamaan diatas, dapat diketahui nilai maksimal cacahan yang akan dilakukan oleh mikrokontroler.

$$f = \frac{16 \times 10^6}{256 \times 8}$$

$$f = 7812,5 \text{ Hz}$$

Dengan frekuensi 7812,5 Hz, sistem akan mencacah sebanyak 7812,5 kali setiap 1 detik. Satu kali cacahan membutuhkan waktu 0,0000128 detik.

Pada pengolahan data secara otomatis dengan memanfaatkan fitur *timer/counter* pada mikrokontroler avr, nilai interval RR (RRI) dapat diperoleh melalui persamaan berikut:

$$RRI = \frac{\text{Nilai cacahan yang terbaca}}{7812,5} \dots\dots\dots(3-3)$$

Nilai RRI yang diperoleh dari persamaan di atas merupakan RRI dalam satuan detik.

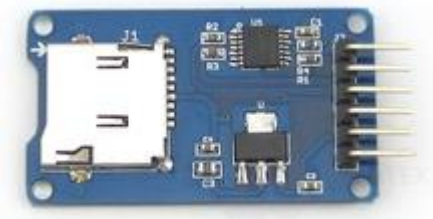
3.3.1.3. Register TIMSK (Timer/counter Interrupt Mask)

Bit	7	6	5	4	3	2	1	0	
(0x6E)	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Register TIMSK digunakan oleh timer0, timer1 dan timer2. Pada timer0, bit 1 (OCIE0A) diberikan nilai 1, untuk mengaktifkan Register OCR0A.

3.3.2. Penyimpanan Data

Nilai hasil komputasi berupa interval RR akan disimpan terlebih dahulu pada sebuah memori eksternal yang dihubungkan langsung ke Arduino melalui modul *SD Card*. Modul *SD Card* merupakan modul untuk yang dapat digunakan untuk mengakses memori dengan tipe *SD Card* untuk melakukan penulisan (*write*) maupun pembacaan (*read*) dengan menggunakan komunikasi *Serial Peripheral Interface* (SPI). Modul *SD Card* ini telah dilengkapi *voltage regulator* 3,3V, sehingga dapat dihubungkan dengan VCC Arduino 5V.



Gambar 3. 5 Modul Micro SD

Sumber : (www.amazon.com, 2017)

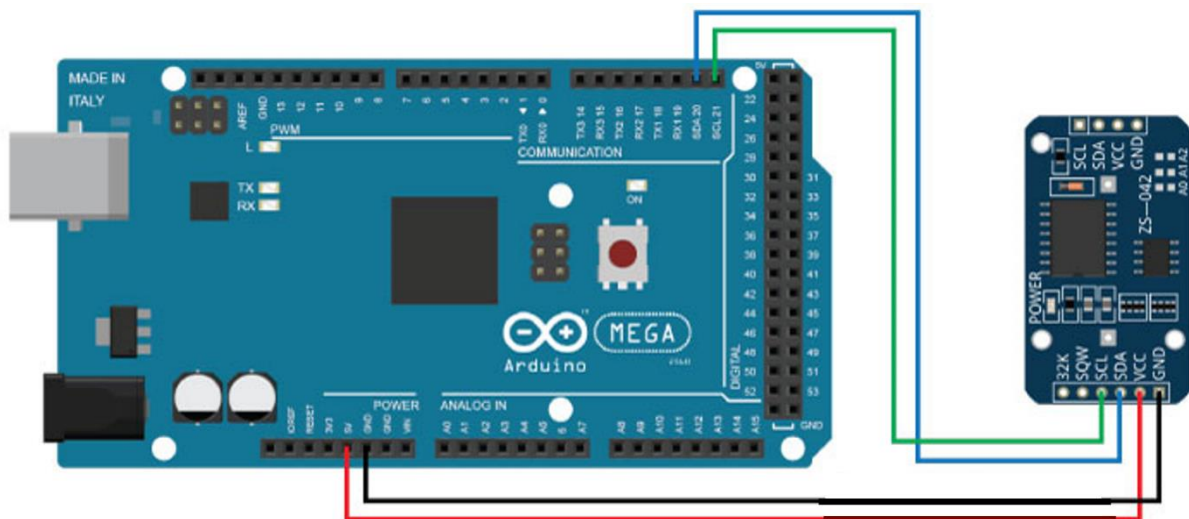
Tabel 3. 2 Konfigurasi pin modul *SD Card* ke Arduino

Modul Micro SD	Arduino
GND	GND
3,3V	3,3V atau 5V
MISO	MISO
MOSI	MOSI
SCK	SCK
CS	CS

Penyimpanan data pada modul *SD Card* ini dibutuhkan sebagai tindakan pencegahan apabila terjadi *error* pada proses pengiriman data ke *database* sehingga masih memiliki *file*

cadangan jika diperlukan untuk sistem pengambilan keputusan. Program akan menyimpan data interval RR setiap pembacaan yang dilakukan. Pembacaan dilakukan sampai dengan 800 data RRI, hal ini dikarenakan sistem selanjutnya, yakni sistem pengambilan keputusan, membutuhkan data rekaman interval RR sebanyak 800 data. RRI akan disimpan ke dalam *file* dengan ekstensi *.csv.

Nama *file* dibuat berdasarkan kapan rekaman EKG tersebut diambil. Hal ini dilakukan agar setiap ingin melakukan rekaman EKG yang baru, *file* otomatis akan dibuat dengan nama *file* yang berbeda sehingga tidak terjadi *error* berupa *file* yang bertumpuk. Untuk mengatasi hal ini, dibutuhkannya komponen berupa modul RTC (*Real Time Clock*), yakni suatu chip yang dapat menyimpan waktu dan tanggal. Maka dari itu, nama *file* akan disesuaikan dengan waktu dilakukannya rekaman EKG dan didapat dari modul RTC. Modul RTC yang digunakan pada penelitian ini ialah RTC 3231. Adapun konfigurasi pin modul RTC 3231 pada Arduino dapat dilihat pada Gambar 3.4.



Gambar 3. 6 Konfigurasi pin Modul RTC 3231 dengan Arduino

Sumber : (www.howtomechatronics.com, 2016)

Jumlah karakter yang dapat ditampung oleh variabel nama *file* maksimal 12 karakter termasuk nama ekstensi nya. Oleh sebab itu, hanya terdapat 8 karakter nama *file* dan 3 karakter digunakan untuk nama ekstensi. Nama *file* yang didapat dari modul RTC 3231 dapat ditentukan melalui program berikut:

```
char filename[]="00:00:00.CSV";// nama file maksimal 12 karakter

//nama file berdasarkan waktu
void getFileName(){
DateTime now = rtc.now();
filename[0]= now.hour()/10+'0;//untuk mendapat digit pertama dari hour()
```



```

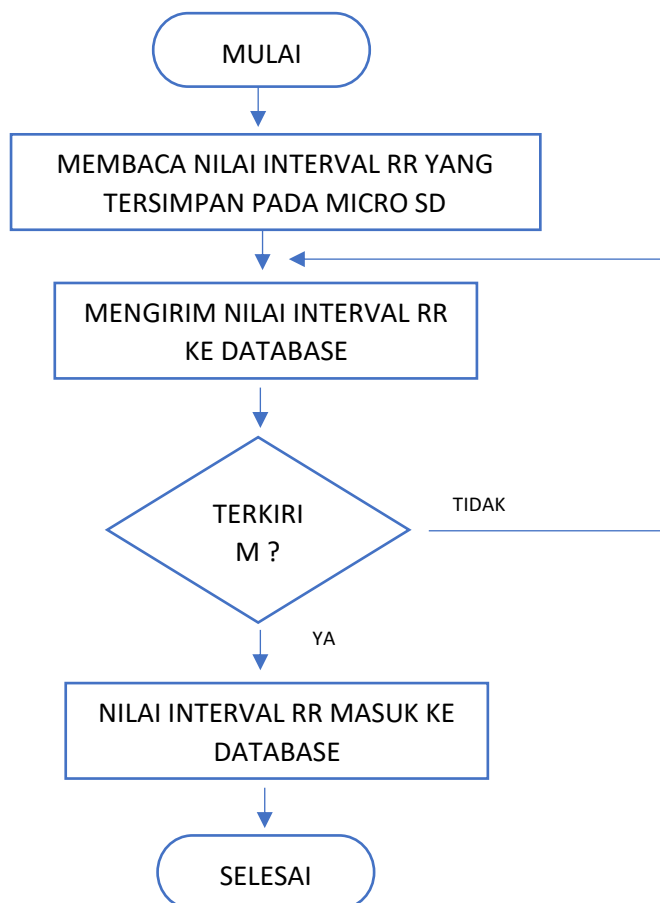
filename[1]= now.hour()%10 + '0';// untuk mendapat digit kedua dari hour()
filename[3]= now.minute()/10 + '0';// untuk mendapat digit pertama minute()
filename[4]= now.minute()%10 + '0';// untuk mendapat digit kedua minute()
filename[6]= (now.second()/10)%10 + '0';//untuk mendapat digit pertama second()
filename[7]= now.second()%10 + '0';// untuk mendapat digit kedua dari second()
}

```

Karakter pada *array*[2] dan *array*[5] diisi dengan karakter titik dua (:) sebagai pemisah antara jam, menit, dan detik pada nama *file*.

3.4. Transmisi Data

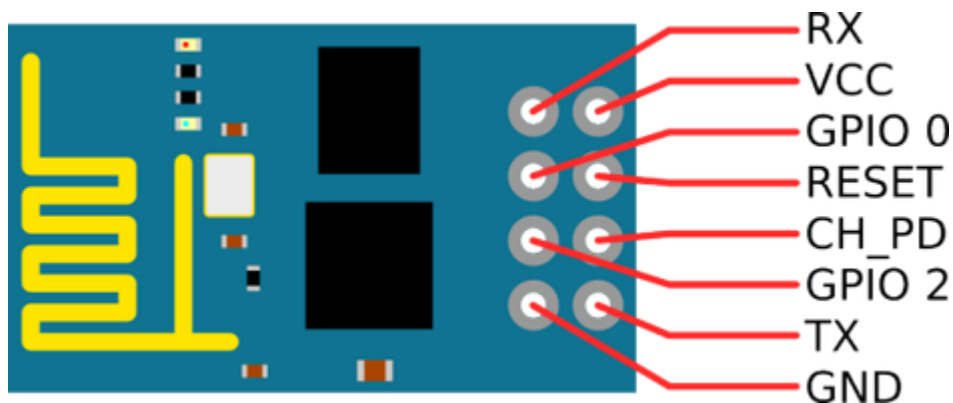
Dibutuhkannya transmisi data hasil komputasi yang sudah diolah di mikrokontroler agar dapat disimpan langsung di PC. Transmisi data yang digunakan adalah transmisi data tanpa kabel (*wireless*) sehingga dibutuhkan *device* yang dapat mentransmisikan data tanpa menggunakan kabel. Pada perancangan ini digunakan modul ESP8266 sebagai modul *wifi* dan *database* MySQL pada *server* XAMPP. Diagram alir transmisi data dapat dilihat pada Gambar 3.6.



Gambar 3. 7 Diagram alir perancangan transmisi data

3.4.1. Modul Wifi ESP8266

ESP8266 merupakan sebuah modul yang berfungsi sebagai modul *Wifi* dan membuat koneksi TCP/IP. Modul ini dapat dijadikan perangkat tambahan pada mikrokontroler seperti Arduino. Modul ini membutuhkan catu daya 3,3V dan memiliki 3 mode, yakni *Stasion*, *Access Point*, dan keduanya. Modul ini dapat diprogram menggunakan *software* Arduino IDE dengan *library* yang telah disesuaikan. Modul dan pin *out* nya dapat dilihat pada Gambar 3.5.



Gambar 3. 8 ESP8266 pin out module

Sumber: (singh, 2016)

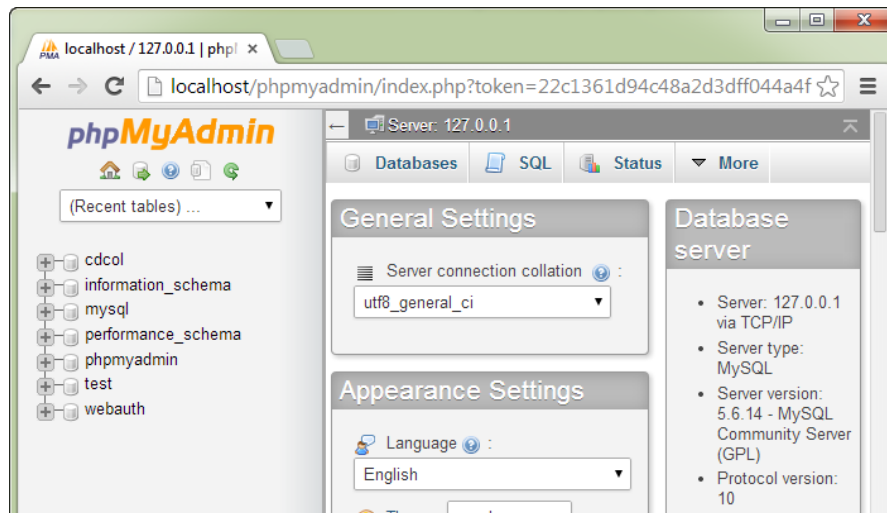
Tabel 3. 3 Konfigurasi pin esp8266 dan Arduino

ESP8266	Arduino
VCC	3,3V
CH_PD	3,3V
GND	GND
Rx	Tx
Tx	Rx

Pada penelitian ini, esp8266 bekerja pada mode 1 atau mode *station*, di mana modul ini akan menerima sinyal *wifi* dari jaringan yang sama pada PC di mana *server* XAMPP dipasang. Untuk mengetahui apakah esp8266 sudah berada pada mode *station* dapat dilakukan pengecekan melalui AT COMMAND.

3.4.2. Database MySQL

Perancangan ini memanfaatkan *database* MySQL pada *phpMyAdmin* yang diakses dari *software* XAMPP. *Database* ini hanya dapat diakses pada jaringan yang sama. Data yang tersimpan dalam *database* dapat disimpan dengan format *.csv. Tampilan *database* MySQL pada *phpMyAdmin* dapat dilihat pada Gambar 3.6.



Gambar 3. 9 Tampilan utama *phpmyadmin* pada *software* XAMPP

Sumber: (Andre, 2014)

Koneksi *database* dibuat dengan menggunakan pemrograman PHP, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan *database*.

