

# Rancang Bangun Prototype Node dan Protokol Komunikasi untuk Pengambilan Data Pada Sistem Penyelamatan Korban Kebakaran Gedung Bertingkat.

Anna Widya Kusuma#1, Ali Husein Alasiry, S.T., M.Eng.#2, Edi Satriyanto, S.Si., M.Si#3

#Jurusan Teknik Elektronika, Politeknik Elektronika Negeri Surabaya  
Kampus PENS-ITS Sukolilo, Surabaya  
an2.tio@gmail.com

## ABSTRAK

Kebakaran gedung bertingkat pada kota besar masih sering terjadi dengan tingkat korban yang cukup besar. Alarm kebakaran pada gedung bertingkat berfungsi dengan baik, namun hal itu hanya untuk pertolongan pertama. Jika api telah membesar maka alarm tersebut sudah tidak berfungsi. Dengan kondisi kebakaran yang cukup besar akan menyebabkan korban kebingungan untuk mencari jalan keluar. Oleh karena itu, proyek akhir ini akan memberikan sebuah sistem informasi berupa tanda yang mempermudah korban untuk meloloskan diri.

Komunikasi serial *singlehop* dengan memanfaatkan serial RS 485 mampu mengambil data tiap *node* baik dalam kondisi normal maupun saat terjadi kebakaran. Untuk melakukan monitoring, *slave* dilengkapi dengan sensor LM 35, AF 30, dan *water detector*. Proses pengambilan data *slave* dilakukan oleh *sub master* setelah mendapatkan *request* dari *master*. Data dari *slave* yang diterima oleh *sub master* akan dikumpulkan terlebih dahulu oleh *master* untuk kemudian dikirimkan ke *server*. Selanjutnya *master* akan menunggu respon dari *server*, jika tidak ada respon maka selama waktu 20 detik *master* akan siklus komunikasi dengan *node* yang lain. Sedangkan jika *server* merespon maka data tersebut akan dikirimkan ke *sub master* untuk dilakukan eksekusi.

Sistem yang diterapkan mampu memberikan informasi jalan keluar dengan tingkat keberhasilan sebesar 90%. Sistem komunikasi antar *node* mendukung proses pengambilan data dengan waktu pengambilan dalam satu siklus sebesar 1.5 second. Proses pengambilan data antar *node* menggunakan *baud rate* sebesar 19200. Pengecekan data antar *node* menggunakan FCS mampu memberikan keakuratan data yang dikirimkan ke *server*.

**Kata Kunci :** Kebakaran, LM 35, AF 30, *water detector*, RS 485, *singlehop*, FCS

## I. PENDAHULUAN

Di kota besar seperti Jakarta, tiap tahun, petugas pemadam kebakaran harus memadamkan 700-800 kejadian dengan kerugian rata-rata sekitar Rp. 100 miliar tiap tahun. Antara tahun 1996-2000, total kejadian kebakaran di Jakarta mencapai 4.210 kali mencakup wilayah seluas 275 ha dengan kerugian ditaksir mencapai Rp. 557,48 miliar. Sedangkan keluarga yang kehilangan tempat tinggal dan harta benda dalam kurun waktu tersebut mencapai 22.784 KK atau 107.182 jiwa, tercatat 208 jiwa di antaranya meninggal dan 307 luka-luka akibat kebakaran tersebut.

Pada proyek akhir ini dibuat suatu sistem jaringan sensor untuk membantu korban mencari jalan keluar yang tepat saat terjadi kebakaran pada suatu gedung. Dengan memberikan sensor pada titik-titik tertentu (jalan keluar) maka akan dapat diketahui seberapa besar kebakaran yang terjadi lokasi tersebut. Dengan mengolah dan membandingkan data sensor tiap titik maka akan dihasilkan sebuah output yang menunjukkan bahwa lokasi tersebut memiliki tingkat kebakaran tertentu (tinggi, sedang, dan rendah).

Kemudian tingkat kebakaran tersebut akan diinformasikan kepada korban untuk memilih jalan keluar yang tepat untuk meloloskan diri. Selain itu, informasi tentang kondisi lokasi kebakaran tersebut bisa diketahui melalui PC yang berada di *security room*. Informasi tersebut divisualisasikan mengikuti konstruksi bangunan gedung sehingga memudahkan petugas pemadam kebakaran. Pihak pemadam kebakaran dapat menganalisa dan mengambil keputusan yang tepat untuk menyelamatkan korban dan memadamkan api.

## II. RUMUSAN MASALAH

Rumusan masalah proyek akhir ini adalah

1. Topologi jaringan bagaimana yang sesuai untuk diterapkan pada jaringan sensor untuk sistem informasi kebakaran gedung bertingkat
2. Bagaimana mengatasi permasalahan komunikasi antar *node* baik dalam kondisi normal maupun ketika ada *node* yang mengalami kerusakan akibat kebakaran.

3. Bagaimana protokol komunikasi untuk pengiriman data yang sesuai dalam aplikasi sistem informasi kebakaran gedung bertingkat.

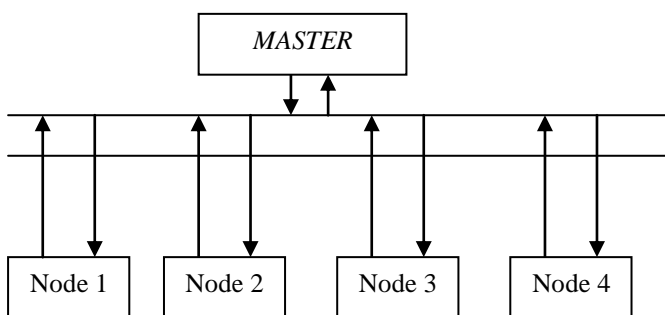
### III. BATASAN MASALAH

Agar pembahasan pada proyek akhir ini tidak melebar, maka diambil batasan masalah yang akan dibahas sebagai berikut :

1. Jumlah node dibatasi sebanyak 10 buah.
2. Sistem hanya melakukan monitoring terhadap plant tanpa adanya pengendalian atau kontrol terhadap *plant*.
3. Sistem yang dibuat hanya akan diuji pada gedung PENS-ITS dengan batasan jumlah lantai 2 dengan masing-masing lantai memiliki tiga jalan keluar.
4. Komunikasi menggunakan serial RS-485.
5. Parameter yang dideteksi hanya perubahan temperatur dan ada tidaknya asap.

### IV. TINJAUAN PUSTAKA

Pada [2] telah digunakan integrasi jaringan multi mikrokontroler untuk system proteksi bahaya kebakaran gedung. Komunikasi data antar mikrokontroler menggunakan RS-485. Sistem yang dibuat terdiri atas satu *Master* dan beberapa *Slave*. Aktuator yang digunakan berupa penyemprot air. Penekanan pada [2] adalah upaya sistem dalam memadamkan api. Tidak adanya komunikasi antara *Slave* 1 dengan *Slave* yang lain.



Gambar 1. Multi Drop

### V. PERENCANAAN DAN PEMBUATAN SISTEM

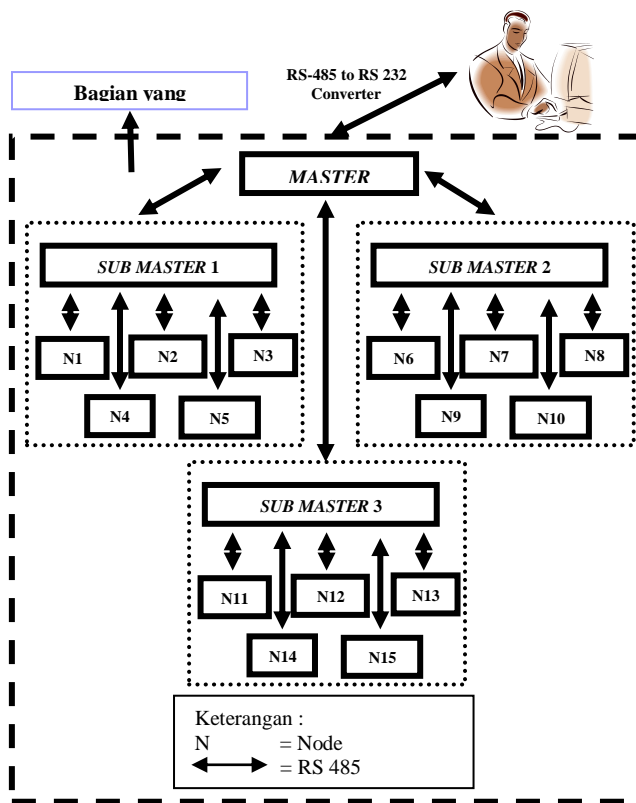
Sistem secara keseluruhan terdiri dari 19 *node* dibagi menjadi 3 yaitu *master*, *sub master* dan *slave*. Satu *node* sebagai *master*, tiga *node* sebagai *sub master*, dan lima belas *node* sebagai *slave*.

Topologi jaringan yang digunakan adalah topologi jaringan pohon (*tree*). Topologi ini kami gunakan karena sesuai dengan karakteristik dari serial RS 485. Komunikasi

max 485 selalu menggunakan topologi *bus* dalam penggunaannya.

Kelemahan dari topologi jaringan pohon yaitu jika terjadi kerusakan pada hierarki teratas maka sistem komunikasi secara keseluruhan tidak akan berjalan, kami atasi dengan peletakan *node* untuk hierarki teratas pada titik aman sehingga sangat kecil kemungkinan untuk terjadi kerusakan pada *node* hierarki teratas. Saat terjadi kebakaran setiap *node* akan memberikan data kondisi sekitar jalan keluar. Kemudian data tersebut diolah dan dibandingkan antara satu dengan yang lain hingga ditemukan jalan yang paling aman untuk dilewati.

Berikut blok diagram sistem keseluruhan :



Gambar 2. Blok diagram system

Terdiri dari empat bagian utama, yaitu

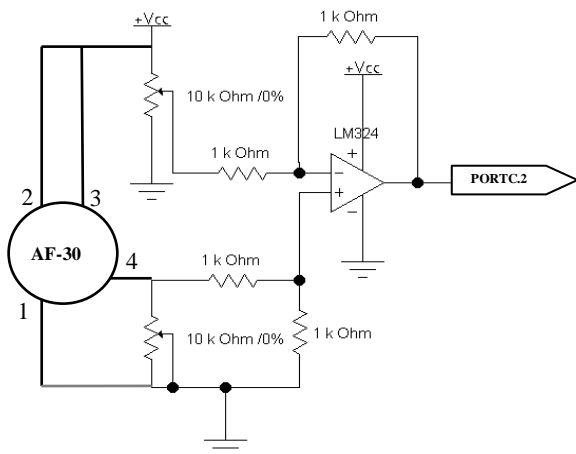
#### 1. Bagian *Slave*

Berfungsi sebagai sensor interaktif yang mendeteksi keadaan lingkungan dari kebakaran. Informasi yang diambil antara lain kondisi temperatur dengan menggunakan LM35, kondisi asap dengan menggunakan AF-30, dan sensor deteksi intensitas air. Setiap data yang diambil berukuran 8 bit. Pengambilan data dilakukan setelah mendapatkan perintah, kemudian menunggu pengambilan data hingga selesai. Setelah itu data akan diolah oleh mikrokontroler dan dikirim menuju *Sub master* dengan menggunakan serial 485.

Setiap lantai pada prototype diletakkan tiga *Slave*, pada prototype memiliki lima lantai sehingga digunakan *Slave* sebanyak lima belas buah.



menggunakan rangkaian *subtractor* maka sensor dapat dikalibrasi.



Gambar 7. Sensor AF-30 dan Op-amp

$$V_o = \left(1 + \frac{R}{R}\right) \cdot \left(\frac{R}{R+R}\right) \cdot V_1 - \frac{R}{R} \cdot V_2$$

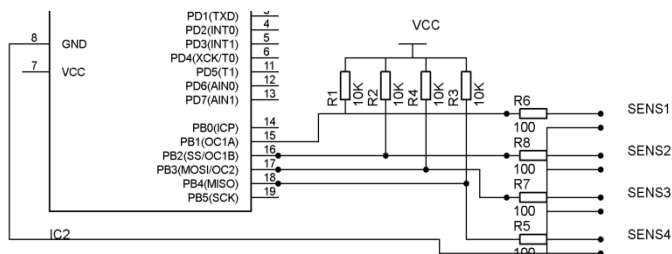
$$V_o = \left(\frac{2R}{R}\right) \cdot \left(\frac{R}{2R}\right) \cdot V_1 - \frac{R}{R} \cdot V_2$$

$$V_o = V_1 - V_2$$

### 3. Sensor Intensitas air

Berfungsi untuk mendeteksi banyak air dalam ruangan tersebut. Dengan banyaknya air maka diasumsikan wilayah tersebut semakin aman.

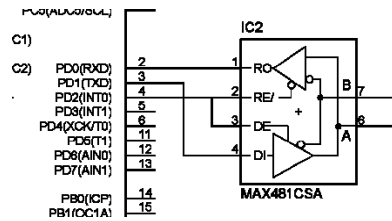
Pada rangkaian terdapat 4 level yang membaca keberadaan air dalam ruangan. Rangkaian ini langsung terintegrasi dengan minimum sistem AT MEGA 8. Sensor berbentuk konduktor yang terdiri dari ground dan vcc. Air merupakan konduktor yang baik. Jadi jika antara ground dan vcc bersentuhan maka tegangan yang masuk pada AT MEGA 8 akan turun mendekati 0 karena tegangan vcc terbagi dua (*voltage divider*).



Gambar 9. Sensor Intensitas air

### 4. Rangkaian Serial RS 485

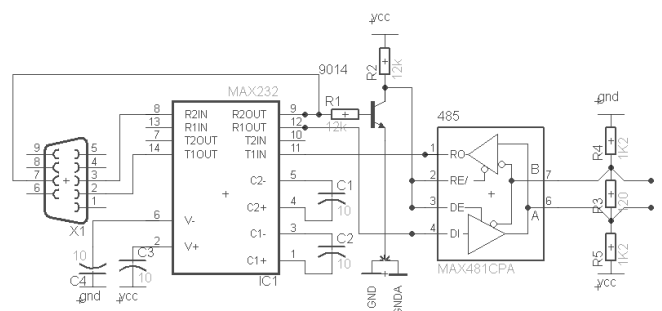
RS 485 berfungsi untuk melakukan komunikasi antara satu lantai dengan lantai yang lain. Sistem yang digunakan adalah sistem multidrop. Pada proyek akhir ini serial RS 485 dihubungkan dengan AT MEGA 8. Berikut adalah rangkaian RS 485 :



Gambar 10. Rangkaian RS 485

### 5. Rangkaian Converter RS 485 to RS 232

Untuk rangkaian converter RS 485 to RS 232 secara keseluruhan seperti pada gambar 3.9. Dari gambar tersebut dapat dilihat bahwasanya rangkaian converter RS 485 to RS 232 terdiri dari dua rangkaian, yaitu rangkaian RS 485 dan rangkaian RS 232. Fungsi dari rangkaian Converter RS 232 to 485 adalah agar node dapat komunikasi dengan PC. Oleh karena tegangan level RS 485 harus diubah terlebih dahulu menjadi tegangan level TTL. Kemudian dari level TTL tegangan tersebut diubah menjadi level tegangan RS 232.



Gambar 11. Rangkaian Converter RS 485 to RS 232

Pada rangkaian tersebut terdapat dua rangkaian, yaitu rangkaian RS 485 dan rangkaian RS 232.

#### Rangkaian RS 232

Rangkaian RS 232 berfungsi untuk mengubah level tegangan RS 485 ke level tegangan RS 232 agar dapat berkomunikasi dengan komputer. *Transmitter out 2* IC max 232 dihubungkan ke transistor. Hal tersebut berfungsi untuk mengontrol RE/ dan DE pada max 485.

#### Rangkaian RS 485

Rangkaian RS 485 berfungsi untuk mengubah level RS 232 menjadi level RS 485 sehingga data dari PC dapat

diterima oleh node. Untuk nilai resistansi bias dan nilai resistansi terminasi sama dengan nilai sebelumnya yaitu  $R_t = 120 \Omega$  (bisa berubah disesuaikan dengan impedansi kabel) dan nilai dari  $R_{b1} = 1.4 K\Omega$  dipasang *pull down* dengan terminal B, untuk  $R_{b2} = 1.4 K\Omega$  dipasang *pull up* dengan terminal A.

### Cara Kerja Rangkaian Converter

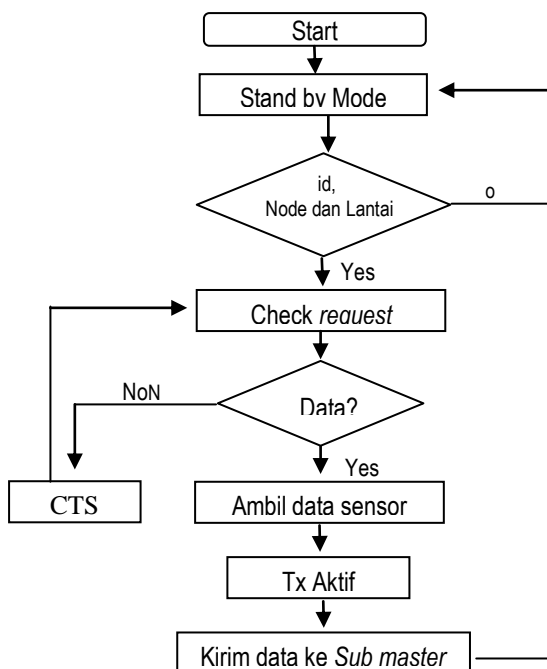
Pada dasarnya *RS232 to RS485 Converter* ini hanya merubah *level* tegangan saja, apabila ada sinyal yang masuk dari PC ke IC MAX232, maka IC ini akan merubah level tegangan yang masuk menjadi *level* tegangan TTL. *Level* tegangan TTL tersebut akan diterima oleh MAX 485, dari MAX 485 tegangan *level* TTL akan diubah menjadi tegangan *level* 485.

Saat PC dalam kondisi *idle* maka *Receiver output* pada IC MAX 232 akan ber-logic *high* ("1"). Hal tersebut menyebabkan transistor 9014 akan *saturasi*. Kondisi tersebut menyebabkan *Receiver Enable* (RE/) menjadi aktif sehingga *Receiver Output* (RO) pada IC MAX 485 siap untuk mengirim data kepada T1in pada IC MAX 232. Kondisi ini akan terus berlangsung hingga transistor MAX 485 dalam kondisi *cut-off*.

Saat PC mengirim start bit maka *Receiver output* pada MAX 232 berlogic low ("0"). Dengan kondisi ini, maka transistor 9014 akan dalam kondisi *cut-off* sehingga *Driver Output Enable* (DO) akan aktif. Dengan begitu *Driver input* siap menerima data dari IC MAX 485. Kondisi tersebut bertahan hingga *Receiver Output* pada IC MAX 232 kembali ke kondisi high (5 Volt).

### PERENCANAAN SOFTWARE

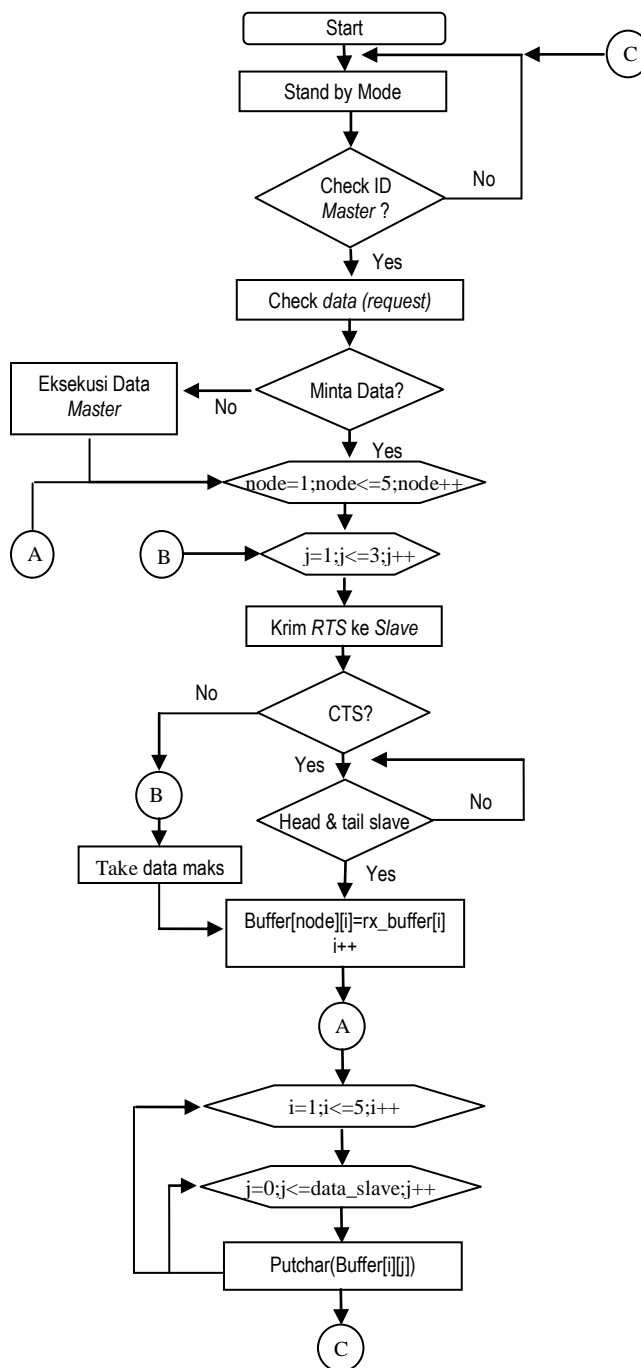
#### 1. Flowchart Slave



Gambar 12. Flowchart slave

Pada flowchart tersebut, node *Slave* dalam kondisi *stand by* untuk menunggu *request* dari *Sub master*. Saat *request* dari *Sub master* datang, maka node akan mengoreksi protocol ( berupa: id (start bit), node serta lantai ) terlebih dahulu. Jika cocok atau sesuai, maka transmitter pada node *Slave* akan diaktifkan kemudian mengirimkan data ADC berupa sensor asap, temperature, dan intensitas air dalam bentuk desimal ke *Sub master*.

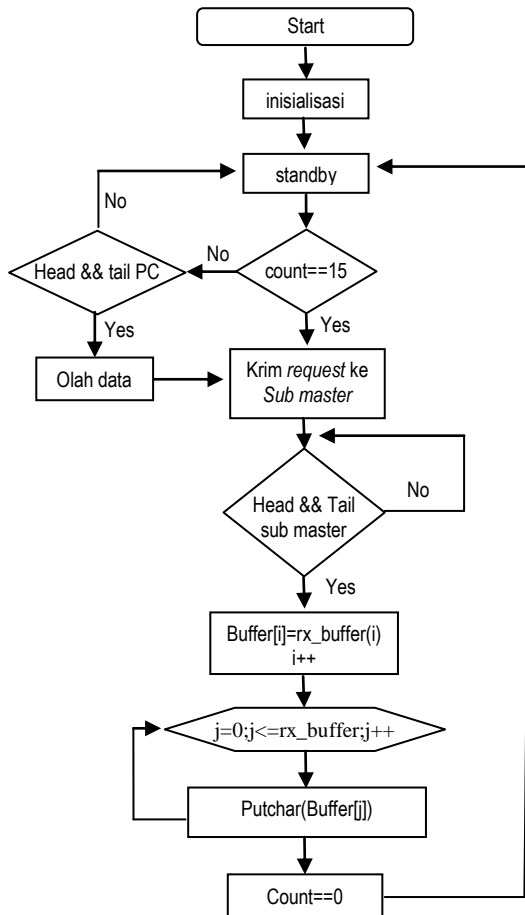
#### 2. Flowchart Sub Master



Gambar 13. Flowchart sub master

Metode pengiriman yang dilakukan dari *sub master* ke *slave* adalah berupa RTS dan CTS. Hal ini bertujuan agar master mengetahui apakah kabel antara *sub master* dengan *slave* tidak terputus. Jika terputus maka *sub master* akan melakukan permintaan ke *slave* lagi. Hal ini dilakukan sebanyak tiga kali. Jika selama tiga kali permintaan tersebut *slave* tetap tidak mengirimkan CTS maka *sub master* akan memutuskan bahwa kabel *slave* terputus. Dan diasumsikan bahwa daerah tersebut kebakaran maksimal. Hal tersebut dilakukan secara terus menerus sebanyak lima kali sesuai dengan banyak *slave* pada tiap *sub master*.

### 3. Flowchart Master



Gambar 14. Flowchart master

Dengan menggunakan interrupt per 1 detik, *count* akan dijumlah secara terus menerus mulai dari 0 hingga 15 (16 second). Saat mencapai 15, maka *master* akan mengirimkan *request* kepada *sub master* untuk meminta data. Setelah itu, *master* akan menunggu hingga terjadi *rx\_interrupt*.

Saat seluruh data dari *sub master* sampai pada *master* maka *master* akan melakukan *buffer* terlebih dahulu terhadap seluruh data. Setelah seluruh data selesai di-*buffer* maka *master* akan mengirimkan data ke PC dengan *header* dan *tail* yang berbeda. Setelah itu, *count* akan di-nolkan kembali.

## PERENCANAAN PROTOKOL

### 1. Protokol Pengecekan Node

Protokol ini digunakan untuk mengetahui kondisi sensor apakah aktif / tidak untuk menentukan proses berikutnya.

Format request protokol

Del	SB	N	R	FCS1	FCS2	T
-----	----	---	---	------	------	---

Keterangan :

Tabel 1. Komponen protokol pengecekan node

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	\$	8 bit
SB	ID sub master	1-3	8 bit
N	ID node	1-5	8 bit
R	Request	R	8 bit
FCS1	FCS	Byte pertama FCS	8 bit
FCS2	FCS	Byte kedua FCS	8 bit
T	Terminator	^	8 bit

Format respon Protokol aktif

Del	A	N
-----	---	---

Keterangan :

Tabel 2. Komponen protokol respon node

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	@	8 bit
A	Ack	A	8 bit
T	Terminator	&	8 bit

### 2. Protokol Pengiriman Data Sensor

Protokol ini digunakan untuk mengambil data sensor pada *slave*.

Del	SB	N	D	FCS1	FCS2	T
-----	----	---	---	------	------	---

Keterangan :

Tabel 3. Komponen protokol pengambilan data node

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	\$	8 bit
SB	ID sub master	Urutan jalan keluar	8 bit
N	ID node	Urutan lantai	8 bit
D	Request data	D	8 bit
FCS1	FCS	Byte pertama FCS	8 bit
FCS2	FCS	Byte kedua FCS	8 bit
T	Terminator	^	8 bit

Format respon *node/slave*

Del	D	SB	N	D1	D2	D3
-----	---	----	---	----	----	----

FCS1	FCS2	T
------	------	---

Keterangan :

**Tabel 4.** Komponen Protokol data *node*

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	@	8 bit
D	keterangan	D	8 bit
SB	<i>ID sub master</i>	Urutan jalan keluar	8 bit
N	<i>ID node</i>	Urutan lantai	8 bit
D1	Sensor temperatur	Nilai sensor	8 bit
D2	Sensor asap	Nilai sensor	8 bit
D3	Sensor <i>water det</i>	Nilai sensor	8 bit
FCS1	FCS	Byte pertama FCS	8 bit
FCS2	FCS	Bye kedua FCS	8 bit
T	Terminator	^	8 bit

### 3. Protokol Pengiriman data *sub master*

Protokol ini digunakan untuk mengirimkan data sensor (dari *sub master*) menuju *master*.

Del	SB	.	D1	.	D2	.	D3	.
-----	----	---	----	---	----	---	----	---

D4	.	D5	FCS1	FCS2	T
----	---	----	------	------	---

Keterangan :

**Tabel 5.** Komponen Protokol data *sub master*

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	@	8 bit
SB	<i>ID sub master</i>	Urutan jalan keluar	8 bit
.	Tanda pemisah	.	8 bit
D1-D5	Data Sensor <i>node</i>	Data sensor tiap <i>node</i>	45 char
FCS1	FCS	Byte pertama FCS	8 bit
FCS2	FCS	Bye kedua FCS	8 bit
T	Terminator	#, !, %	8 bit

### 4. Protokol Pengiriman data *master*

Protokol ini digunakan untuk melakukan *request* data dari *master* menuju *sub master*

Del	M	R	T
-----	---	---	---

Keterangan :

**Tabel 6.** Komponen Protokol *request sub master*

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	\$	8 bit
M	Master	M	8 bit
R	Request	R	8 bit
T	Terminator	^	8 bit

Berikut format protokol pengiriman data *master* menuju ke *server*

Del	.	D1	.	D2	.	D3	FCS1	FCS2	T
-----	---	----	---	----	---	----	------	------	---

Keterangan :

**Tabel 7.** Komponen Protokol pengiriman data dari *master* ke *server*

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	(	8 bit
SB	<i>ID sub master</i>	Urutan jalan keluar	8 bit
.	Tanda pemisah	.	8 bit
D1-D3	Data Sensor	Data sensor dari <i>sub master</i>	135 char
FCS1	FCS	Byte pertama FCS	8 bit
FCS2	FCS	Bye kedua FCS	8 bit
T	Terminator	)	8 bit

### 5. Protokol Eksekusi Data dari *Server*

Del	T	N	D1	T	N	D2	T	N	D3	T
-----	---	---	----	---	---	----	---	---	----	---

**Tabel 8.** Komponen Protokol pengiriman data dari *master* ke PC

Simbol	Keterangan	Isi	Ukuran
Del	Delimiter	P,C	16 bit
T	<i>Tangga</i>	T	8 bit
N	<i>ID sub master</i>	Urutan <i>sub master</i>	8 bit
D1	Eksekusi tangga 1	E0-FF	8 bit
D2	Eksekusi tangga 2	E0-FF	8 bit
D3	Eksekusi tangga 3	E0-FF	8 bit

## VI. DATA DAN ANALISA DATA

Data percobaan

Pengujian ke-		Titik Api					Data kirim PC	Out	TIMER (sec)
		L1	L2	L3	L4	L5			
1	Tangga 1	■	■	■	■	■	PC1E51F388 <sup>^</sup>	Tidak sesuai	16.5
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
2	Tangga 1	■	■	■	■	■	PC1E11E3FF <sup>^</sup>	sesuai	15.6
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
3	Tangga 1	■	■	■	■	■	PC1E11E3BB <sup>^</sup>	sesuai	15.4
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
4	Tangga 1	■	■	■	■	■	PC1E01F8E7 <sup>^</sup>	sesuai	15.5
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
5	Tangga 1	■	■	■	■	■	PC1E01F9E6 <sup>^</sup>	sesuai	15.6
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
6	Tangga 1	■	■	■	■	■	PC1F01F81E7 <sup>^</sup>	sesuai	15.3
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
7	Tangga 1	■	■	■	■	■	PC1E01FCE3 <sup>^</sup>	sesuai	15.5
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
8	Tangga 1	■	■	■	■	■	PC1F01ECF3 <sup>^</sup>	sesuai	15.4
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
9	Tangga 1	■	■	■	■	■	PC1F01ECE3 <sup>^</sup>	sesuai	15.4
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			
10	Tangga 1	■	■	■	■	■	PC1FC1E0E3 <sup>^</sup>	sesuai	15.5
	Tangga 2	■	■	■	■	■			
	Tangga 3	■	■	■	■	■			

**Keterangan :** ■ = kebakaran maksimal  
 ■ = Kebakaran sedang  
 ■ = Aman

### Analisa

Pengujian dilakukan dalam *prototype* yang telah dibuat dengan jalan keluar yang telah disediakan. *Prototype* memiliki 3 jalan keluar dan di sekitar jalan keluar terdapat node. Setiap node mewakili daerah yang telah ditentukan dan setiap node mewakili ketiga jalan keluar. *Master* memulai siklus untuk mengambil data pada sensor pada saat detik ke-20. Kemudian *sub master* akan *request* ke slave dengan melakukan pengecekan terlebih dahulu apakah kabel terputus atau tidak. Jika kabel terputus maka *sub master* akan melanjutkan *request* ke *slave* yang lain. Jika seluruh pengiriman node telah selesai diterima oleh sub master, maka data tersebut akan dikirim ke *master* dengan metode pengiriman dari sub master 1, sub master 2, dan terakhir adalah sub master 3. Data dari *sub master* terlebih dahulu akan disimpan oleh *master* pada *eprom* sebelum dikirim ke *server*. *Master* akan menunggu respon dari *server*, jika hingga 20 detik *server* tidak merespon maka *master* akan melakukan *request* ulang ke *sub master*, namun sebaliknya jika terdapat respon dari *server* maka data *server* akan diolah terlebih dahulu oleh *master* sebelum dikirimkan menuju *sub master*. Dari *sub master* data tersebut dieksekusi sesuai dengan permintaan dari *server*.

*Baud rate* yang digunakan untuk melakukan komunikasi dengan *server* sebesar 9600 bps. Sedangkan untuk komunikasi antar *node baud rate* yang digunakan sebesar 19200 bps. Penggunaan *baud rate* yang berbeda dikarenakan dari pengujian yang telah dilakukan penggunaan *baud rate* 19200 menyebabkan data yang terkirim ke *server* mengalami banyak mengalami *error*. Tercatat bahwa pengiriman data yang berhasil dilakukan oleh *sub master* dengan *baud rate* 19200 hanya 6 kali pengiriman data dalam satu siklus sehingga perlu dilakukan perubahan pada *baud rate*. Waktu rata-rata yang dibutuhkan dalam satu siklus pengiriman dan penerimaan data sebesar 1.5 *second*. Pada program, dibutuhkan delay minimal 2 ms dalam mengirimkan setiap karakter. Jika tidak, maka data yang terkirim tidak akan terbaca oleh *server* atau data akan mengalami kerusakan. Penggunaan FCS pada setiap pengiriman data memberikan keakuratan pada data yang dikirim. Jika terdapat data yang mengalami *error* maka FCS pada penerima dan pengirim tidak akan sama, maka *master* akan meminta ulang data tersebut ke pihak *sub master*. Hal tersebut terbukti dengan tidak adanya kesalahan data yang dikirim kepada *server* dari sepuluh kali pengujian yang dilakukan.

Dari sepuluh pengujian yang dilakukan hanya satu kali kegagalan dalam menemukan jalan keluar. Hal ini disebabkan karena pengiriman data dari *server* tidak sesuai dengan protokol yang telah direncanakan sehingga *master* tidak dapat membacanya. Sistem komunikasi yang dibangun cukup berhasil untuk mengirimkan data menuju ke *server* melalui *master* dengan menggunakan serial RS 485 dengan tingkat keberhasilan sebesar 90%.

## VII. KESIMPULAN

Dari pengujian yang telah dilakukan maka dapat disimpulkan :

1. Sistem informasi yang digunakan untuk aplikasi penyelamatan korban kebakaran gedung bertingkat memiliki tingkat keberhasilan sebesar 90 % dan rata-rata waktu yang dibutuhkan mulai dari pengambilan data hingga melakukan eksekusi penyelamatan sebesar 1.5 detik.
2. Topologi jaringan *tree* dapat diterapkan pada aplikasi sistem informasi kebakaran gedung bertingkat dengan waktu rata-rata satu siklus pengiriman sebesar 1.1 detik.
3. Hasil penerapan algoritma *singlehop* untuk sistem komunikasi data antara *sub master* dengan setiap *node/slave* serta *sub master* dengan *master* pada sistem penyelamatan korban kebakaran gedung bertingkat ini berjalan dengan optimal dengan tingkat keberhasilan 100%. Rata-rata waktu yang dibutuhkan untuk mengambil data dari 5 *node* atau *slave* sebesar 1.1 *second*.



## VII. SARAN

1. Untuk mendeteksi penyebaran titik api dibutuhkan sensor penunjuk arah angin untuk mendeteksi penyebaran titik api sehingga informasi yang diberikan kepada pemadam kebakaran akan lebih baik.

## VIII. DAFTAR PUSTAKA

- [1] Marais, Hein. 2008. *RS-485/RS-422 Circuit Implementation Guide*. ANALOG DEVICE.
- [3] Marais, Hein. 2008. *RS-485/RS-422 Circuit Implementation Guide*. ANALOG DEVICE.
- [4] Usman. 2008. *Teknik Antarmuka + Pemrograman Mikrokontroler AT89S52*. ANDI : Yogyakarta
- [5] Winoto, Ardi. 2008. *Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*. INFORMATIKA : BANDUNG
- [6] W.Hughes, Fredireick. 1997. *Panduan OP-AMP*. PT Elex Media Komputindo Gramedia : Jakarta