BAB III
METODOLOGI PENELITIAN

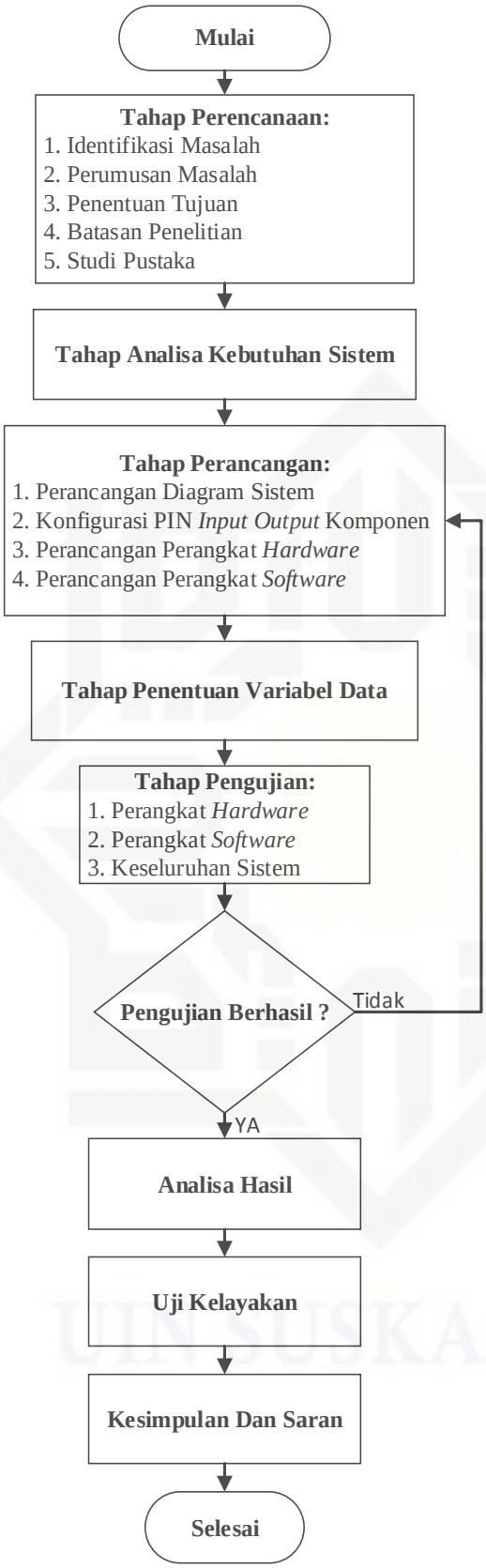
3.1. Jenis Penelitian

Jenis penelitian ini yaitu kuantitatif, penelititan kuantitatif yang dilakukan adalah eksperimental dengan melakukan uji coba sistem dan selanjutnya menganalisa terhadap variabel *input* dan variabel *output* yang dihasilkan untuk mengetahui kemampuan dan kualitas kinerja sistem. Eksperimen ini dilakukan dengan rancang bangun sistem peringatan dini kebakaran hutan dalam bentuk diagram, perangkat *hardware*, dan perangkat *software*. Dengan melakukan eksperimen terhadap perancangan dan pembuatan perangkat, diharapkan akan didapatkan rangkaian serta program yang sesuai dengan fungsi serta tujuan dari pembuatan perangkat.

3.2. Prosedur Penelitian

Penelitian ini dilakukan dalam beberapa siklus tahapan yang saling berkaitan. Tahapan prosedur penelitian dimulai dari tahap perencanaan yang bertujuan untuk menentukan permasalahan yang akan diteliti, perumusan masalah, menentukan tujuan penelitian, batasan penelitian yang akan dilakukan dan studi pustaka yang bertujuan untuk mencari teori-teori terkait penelitian yang akan dilakukan berdasarkan data sekunder yang relevan. Kemudian tahapan selanjutnya yaitu tahap analisa kebutuhan komponen yang akan digunakan dalam perancangan perangkat. Dilanjutkan ketahap perancangan, adapun kegiatan yang dilakukan pada tahap ini yaitu perancangan diagram sistem, penentuan rangkaian alat, perancangan perangkat *hardware* stasiun *transceiver* dan *receiver*, serta perancangan software. Dilanjutkan ketahap menentukan variabel data yang dibutuhkan dalam proses pengumpulan data, untuk keperluan analisa data. Kemudian setelah itu melakukan tahapan uji coba, selanjutnya analisa hasil dan terakhir melakukan tahapan uji kelayakan yang bertujuan untuk mengetahui kemampuan dan kualitas seluruh perangkat. Prosedur penelitian yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 3.1 diagram alur prosedur penelitian.

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Sultan Syarif Kasim Riau State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU



Gambar 3.1. Diagram alur prosedur penelitian

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3.3. Tahap Perencanaan

Tahap perancanaan adalah tahap dalam menentukan permasalah penelitian, mulai dari identifikasi masalah yang akan diteliti, hingga tujuan akhir yang ingin dicapai dari penelitian ini. Adapun kegiatan yang dilakukan pada tahap identifikasi adalah:

1. Identifikasi Masalah

   Tahap identifikasi masalah dilakukan berdasarkan fenomena yang sedang terjadi dan disertai dengan data skunder yang relevan.

2. Perumusan Masalah

   Pada tahap perumusan masalah penulis membuat pertanyaan yang akan dijawab melalui penelitian ini berdasarkan fenomena yang terjadi.

3. Penentuan Judul Penelitian

   Berdasarkan perumusan masalah yang telah ditetapkan, maka penulis menentukan judul penelitian yaitu "Rancang Bangun Sistem Peringatan Dini Kebakaran Hutan Menggunakan Sistem *Wireless Sensor Network* (WSN) Berbasis Arduino".

4. Penentuan Tujuan

   Penentuan tujuan adalah tahap untuk memperjelas apa saja yang menjadi sasaran dari penelitian ini terkait dengan topik permasalahan yang sedang dilakukan dalam penelitian.

5. Studi Literatur

   Studi literatur merupakan tahapan mencari, mengumpulkan dan menganalisa data masalah yang terjadi, dari berbagai sumber, baik dari data statistik pemerintah maupun *website* resmi terkait dengan topik permasalahan, serta melakukan *riview* pada beberapa jurnal terkait untuk membedakan penelitian sebelumnya dengan penelitian yang akan dilakukan.

3.4.Analisa Kebutuhan Sistem

Analisa kebutuhan sistem merupakan tahapan menentukan kebutuhan komponen perangkat, dengan melakukan analisa atau menjabarkan fitur-fitur yang diinginkan dari komponen yang akan digunakan nantinya dalam penelitian ini sangat diperlukan untuk mendapatkan hasil yang optimal, berdasarkan topik permasalahan dilakukan analisa kebutuhan komponen yang akan digunakan, adapun data-data komponen yang dibutuhkan dalam pengerjaan sistem adalah sebagai berikut:

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

a. Suhu dan kelembaban seperti yang telah dijelaskan sebelumnya merupakan salah satu parameter indikasi adanya kebakaran, untuk mengetahui kondisi suhu dan kelembaban lingkungan di sekitar *node* , dibutuhkan perangkat yang mampu mengukur suhu dan kelembaban dengan sensitivitas yang tinggi dan tingkat akurasi yang bagus. Sensor DHT22 adalah sensor suhu dan kelembaban yang memiliki tingkat akurasi yang cukup bagus serta memiliki sensitivitas yang tinggi. DHT22 mampu mengukur suhu dari -40° – 80°C dan mampu mengukur kelembaban dalam rentang 0-100%, serta memiliki kesetabilan pengukuran dengan kesalahan pengukuran (*error*) ± *0.5%* per tahun, dan juga dengan biaya yang sangat terjangkau.

b. Karbon monoksida adalah gas dari hasil pembakaran tidak sempurna, salah satu sumber pembentuk karbon monoksida berasal dari pembakaran berbahan dasar karbon seperti kayu, dan bahan organik lainnya. Untuk mengetahui kandungan gas CO, sistem ini akan menggunakan sensor MQ7. Fitur dari sensor gas MQ7 ini mempunyai sensitivitas yang tinggi terhadap karbon monoksida (CO), stabil, usia pakai yang lama dengan biaya yang terjangkau.

c. PM10 adalah partikulat kasar yang terbentuk dari bahan padat atau cair yang berasal dari ratusan senyawa yang berbeda seperti debu, jelaga, abu, ataupun logam. Dalam konsentrasi tertentu PM10 mampu menyebabkan serangan asma dan bronkitis akut dan juga dapat menyebabkan kematian. Menurut Keputusan Kepala Badan Pengendalian Dampak Lingkungan No. 107 Tahun 1997 Tanggal 21 November 1997, PM10 termasuk parameter dasar untuk Indeks Standar Pencemar Udara (ISPU). Salah satu sumber pembentuk partikulat PM10 adalah kebakaran hutan. Dalam sistem ini digunakan sensor masa udara tipe GP2Y1010AU0F, sensor ini dirancang khusus untuk mendeteksi dan mengukur partikel sangat kecil yang ada di udara dengan sensitivitas tinggi dan penggunaan arus yang kecil ±20mA.

d. Berdasarkan tujuan awal pada perancangan sistem *monitoring* dan peringatan dini kebakaran hutan untuk dapat mengirim dan menerima paket data informasi dari hasil pengukuran dan pendeteksian antar *node* menuju stasiun *receiver* dengan cepat dan akurat, sistem ini menggunakan modul radio SI4432 sebagai media komunikasi untuk mengirim dan menerima paket data antar *node* dan stasiun penerima. Modul radio SI4432 memiliki fitur yang bagus dengan *range* frekuensi 240 – 930 MHz, dengan

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

kecepatan pengiriman data 0.123 - 256 kbps, tegangan listrik yang digunakan 1.8-3.6V dan sudah dilengkapi dengan teknologi CSOM dan ISM sehingga frekuensi radio tidak mengganggu populasi makhluk hidup yang berada di area frekuensi.

e. Perancangan sistem ini nantinya dilengkapi media penyimpanan cadangan, dan pembacaan waktu selama sistem beroperasi. Arduino *Data Logger Shield* yang dilengkapi RTC (*Real Time Clock*) dan Modul memori SD *card* serta modul yang kompatibel terhadap mikrokontroler Arduino seperti Arduino Mega sehingga sangat cocok untuk digunakan dalam sistem ini sebagai modul tambahan untuk penyimpanan cadangan dan pencatat waktu pada data pengukuran.

f. Seluruh proses kerja sistem dikendalikan oleh perangkat lunak dengan menggunakan mikrokontroler, digunakan untuk sistem ini adalah mikrokontroler Arduino, untuk lebih tepatnya mikrokontroler yang digunakan adalah Arduino Mega dan Arduino Nano. Bahasa pemrograman Arduino sederhana, pin I/O (*input* dan *output*) Arduino Mega dan Arduino Nano sesuai dengan kebutuhan sistem selain itu mikrokontroler ini mudah ditemukan dan juga dengan harga yang terjangkau.

g. Sumber listrik pada sistem ini harus mampu memenuhi kebutuhan kelistrikan sistem secara *real time.* Sumber listrik yang digunakan pada sistem ini menggunakan catu daya dari baterai 12 volt, dengan *Solar Cell* sebagai pengisi daya listrik pada baterai.

h. Paket data yang telah dikirim oleh MCU stasiun *receiver* ke komputer *server* akan diolah untuk ditampilkan dalam bentuk tampilan GUI (*Graphical User Interface*). Perancangan aplikasi *Pemantau* sistem peringatan dini kebakaran hutan pada komputer *server* menggunakan *Microsoft Visual Studio express 2013* dengan bahasa pemrograman C#. *Microsoft Visual Studio* sangat cocok digunakan dalam membangun aplikasi *Pemantau* sistem peringatan dini kebakaran hutan, karena *Microsoft Visual Studio* merupakan perangkat lunak IDE (*Integrated Development Environment)* dari *Microsoft* yang didesain khusus untuk dapat digunakan dalam membangun aplikasi antarmuka dengan menggunakan bahasa pemrograman C#.

i. Data hasil pengukuran sensor disetiap *node* serta waktu pengambilan data yang diterima oleh komputer akan disimpan pada *database*. Tujuan pembuatan *database* adalah untuk memudahkan dalam mengakses data. *Database* berperan penting dalam sistem ini, karena *database* mampu menghindari terjadinya duplikasi data. *Database* yang

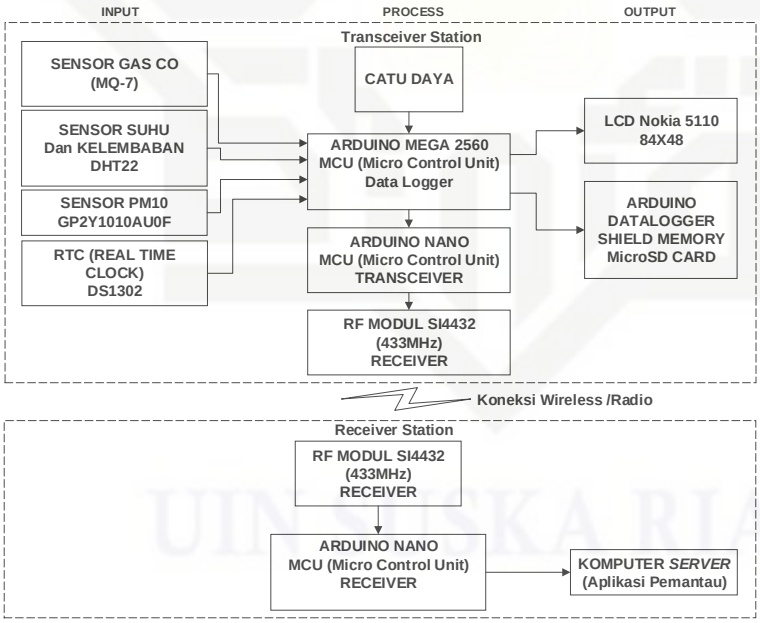
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

digunakan dibangun menggunakan *Microsoft SQL Server 2012*. *Microsoft SQL Server* 2012 merupakan IDE (*Integrate Develop Environment*) yang didesain khusus dalam membangun sebuah aplikasi *database*.

3.5.Tahap Perancangan

Setelah melakukan analisa kebutuhan perangkat, maka perlu perancangan susunan komponen-komponen berdasarkan dengan fitur-fitur yang dibutuhkan sistem agar perangkat dapat digunakan sesuai dengan fungsi yang diinginkan. Langkah awal sebelum melakukan perancangan sistem adalah membuat blok diagram rangkaian sistem, yang dapat dilihat pada Gambar 3.2. Blok diagram rangkaian sistem merupakan gambaran dasar sebelum melakukan perancangan perangkat. Perancangan sistem terdiri dari perancangan *hardware* dan software. Perancangan *hardware* dilakukan dengan membuat konfigurasi pin *input output* komponen *hardware* terlebih dahulu kemudian dilanjutkan dengan membuat model perangkat stasiun *transceiver* dan model perangkat stasiun *receiver*. Sedangkan perancangan *software* dilakukan dalam empat bagian yang berbeda. Perancangan software untuk MCU data logger, MCU modul radio *transceiver*, MCU modul radio *receiver*, dan aplikasi *Pemantau*.

3.5.1. Perancangan Blok Diagram Sistem



Gambar 3.2. Blok diagram rangkaian sistem

Blok diagram rangkaian sistem merupakan gambaran proses kerja sistem secara keseluruhan, diagram rangkaian terbagi menjadi tiga bagian utama. Blok *input* merupakan

Hak Cipta dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

sebagai pemberi sinyal masukan yang berasal dari sensor MQ7, sensor DHT22, sensor PM10, dan RTC DS1302. *Micro Control Unit* (MCU) dan modul radio SI4432 sebagai blok proses. Blok *output* terdiri dari LCD Nokia 5110 data *logger shield* dan komputer. Adapun fungsi dari setiap bagian blok rangkaian diagram sebagai berikut:

1. Sensor MQ7

Sensor gas MQ7 berfungsi untuk mengukur besaran konsentrasi gas CO. Gas CO yang terdeteksi oleh sensor MQ7 dikonversi menjadi tegangan listrik yang berupa sinyal analog yang kemudian akan diterjemahkan oleh MCU menjadi nilai desimal agar dapat dilihat kandungan CO yang terukur.

2. Sensor PM10

Sensor PM10 tipe *Sharp* GP2Y1010AU0F berfungsi sebagai input pengukur kandungan partikel PM10 dan mengolahnya menjadi sinyal analog yang kemudian diterjemahkan oleh MCU sebagai salah satu variabel pencemaran udara yang dipantau.

3. Sensor DHT22

Sensor DHT22 merupakan sensor Suhu dan kelembaban yang menyajikan besaran pengukuran suhu dan kelembaban dengan tingkat akurasi yang baik. Sensor DHT22 berfungsi mengukur suhu dengan satuan *Celcius* dan *Relative Humidity* dalam satuan persen (%).

4. RTC DS1302

RTC DS1302 berfungsi sebagai pencatat waktu detik, menit, jam, tanggal, bulan, hari, serta tahun dari pengukuran sensor dan waktu komunikasi pengiriman data pengukuran dengan batas waktu penggunaan hingga tahun 2100, RTC juga berfungsi untuk keperluan program MCU.

5. LCD Nokia 5110

LCD Nokia 5110 merupakan bagian *output* sistem yang berfungsi menampilkan waktu sistem berjalan, data hasil pengukuran sensor, dan juga sebagai *display* untuk keperluan pengaturan MCU melalui *serial* USB (*Universial Serial BUS*) dalam bentuk teks.

6. Modul Radio SI4432

Modul radio berfungsi sebagai media untuk pengiriman dan penerimaan data pengukuran sensor antara satu *node* sensor dengan *node* lainnya menuju *server*.

UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

7. Arduino *Shield Data Logger*

Arduino *shield data logger shield* berfungsi untuk mencatat data pembacaan sensor pada memori *MicroSD*. Bertujuan sebagai memori penyimpan cadangan dan juga agar mempermudah ketika ingin digunakan untuk kepentingan lebih lanjut.

8. Arduino Mega 2560 dan Arduino Nano pada Modul Radio *Transceiver*

Arduino Mega 2560 merupakan *Micro Control Unit* (MCU) untuk perangkat data *logger* terinterkoneksi dengan Arduino Nano yang merupakan MCU untuk modul radio *transceiver*. Arduino Mega bekerja mengendalikan proses pengukuran sensor serta keperluan komunikasi I2C dengan Arduino Nano, dan Arduino Nano berfungsi untuk proses pengiriman data pengukuran sensor ke *server* melalui modul radio *transceiver*.

9. Arduino Nano pada Modul Radio *Receiver*

Arduino Nano sebagai *Micro Control Unit* (MCU) modul radio *receiver* berfungsi untuk mengendalikan proses penerimaan pesan data pengukuran dari *transceiver* dan mengirim pesan data tersebut ke komputer.

10. Komputer

Komputer berfungsi untuk mengolah data yang diterima dari *receiver* melalui serial USB untuk kemudian data tersebut disimpan di *database* dan ditampilkan ke aplikasi *Pemantau* dalam bentuk GUI (*Graphical User Pemantau*).

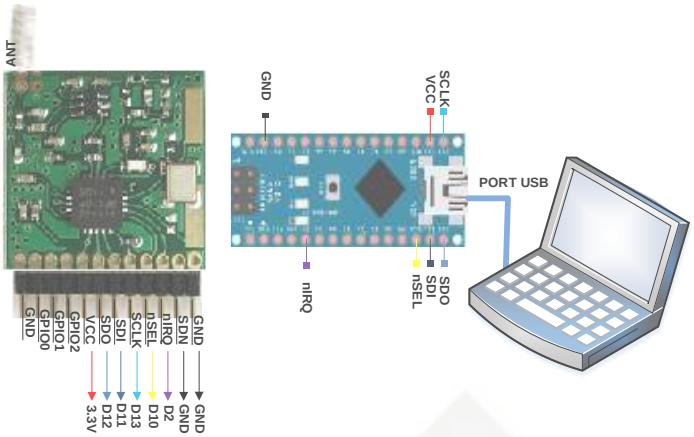
11. Catu Daya

Catu daya berfungsi sebagai sumber energi listrik untuk Arduino Mega. Catu daya yang digunakan yaitu berasal dari baterai 12 volt dengan *Solar Cell* yang berfungsi sebagai penyuplai tegangan pengisian daya pada baterai.

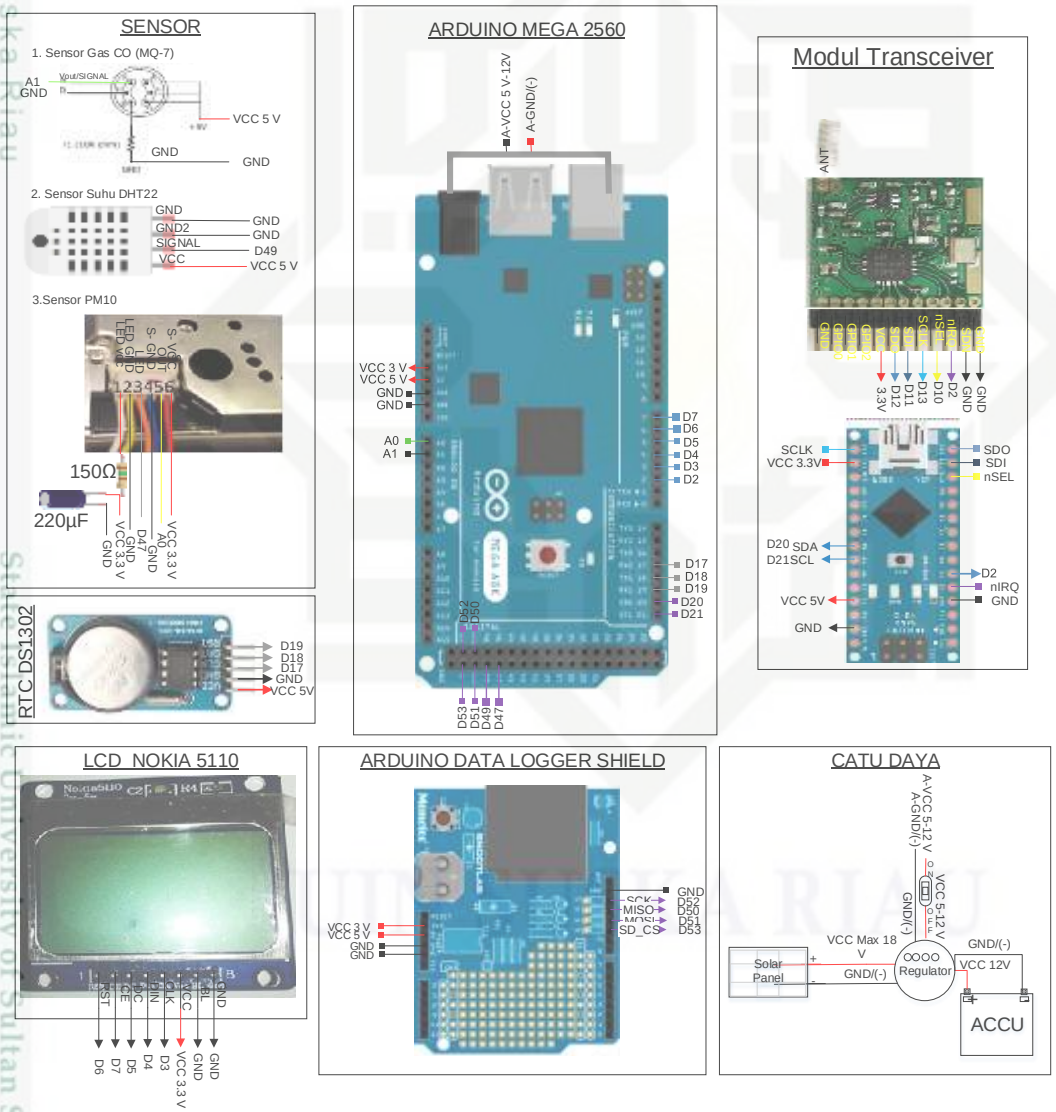
3.5.2. Konfigurasi Komponen-Komponen Terhadap PIN I/O Arduino

Konfigurasi komponen-komponen terhadap PIN *input* dan *output* betujuan untuk menentukan PIN pada arduino yang digunakan sesuai dengan fitur yang dibutuhkan komponen agar sistem dapat berfungsi sesuai dengan skenario yang diinginkan, dan juga untuk mempermudah dalam perancangan, baik itu perancangan *hardware* ataupun *software*. Berikut adalah konfigurasi komponen terhadap PIN I/O yang digunakan.

Hak Cipta Milik UIN Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 3.3.Konfigurasi komponen pada stasiun *reveiver* terhadap PIN I/O arduino

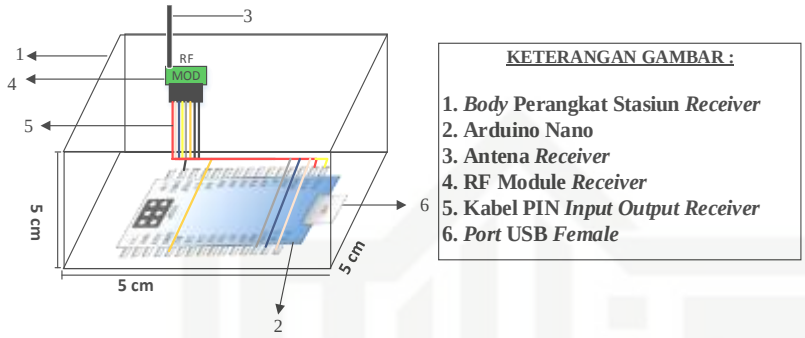


Gambar 3.4. Konfigurasi komponen stasiun *transceiver* terhadap PIN I/O arduino

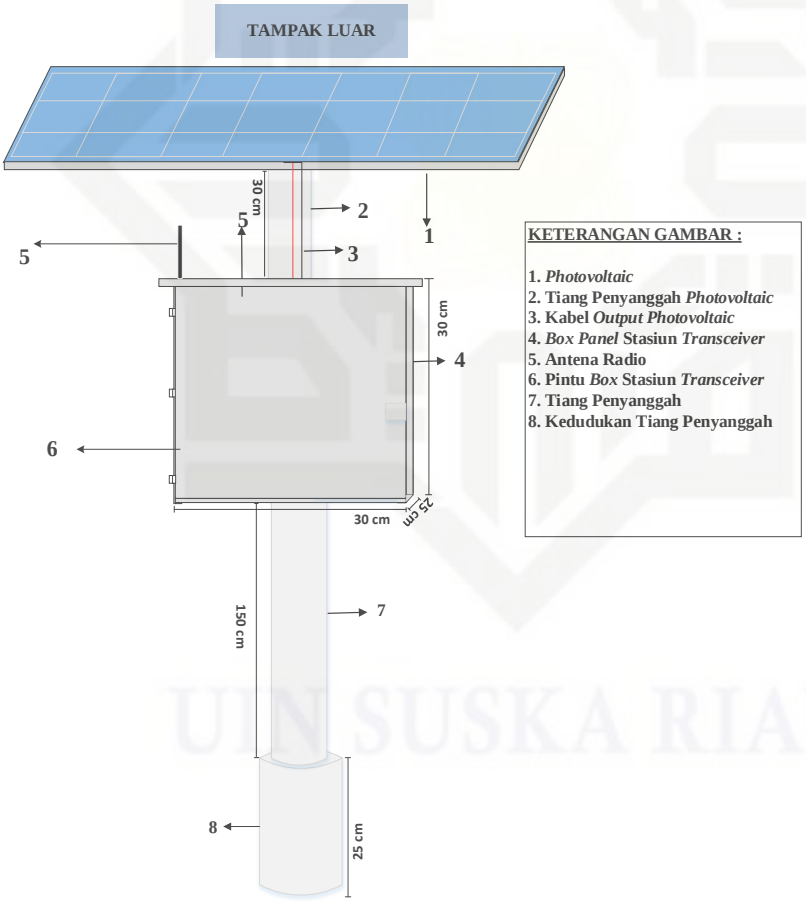
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau

3.5.3. Desain Perangkat *Hardware* (Perangkat Keras)

Perangkat sistem peringatan dini kebakaran hutan yang akan diterapkan di lapangan harus tersusun rapi dan setiap komponen mampu bekerja secara maksimal, untuk mendapatkan hasil tersebut, maka perlu dilakukan desain perangkat *hardware*. Berikut adalah perancangan tata letak komponen:

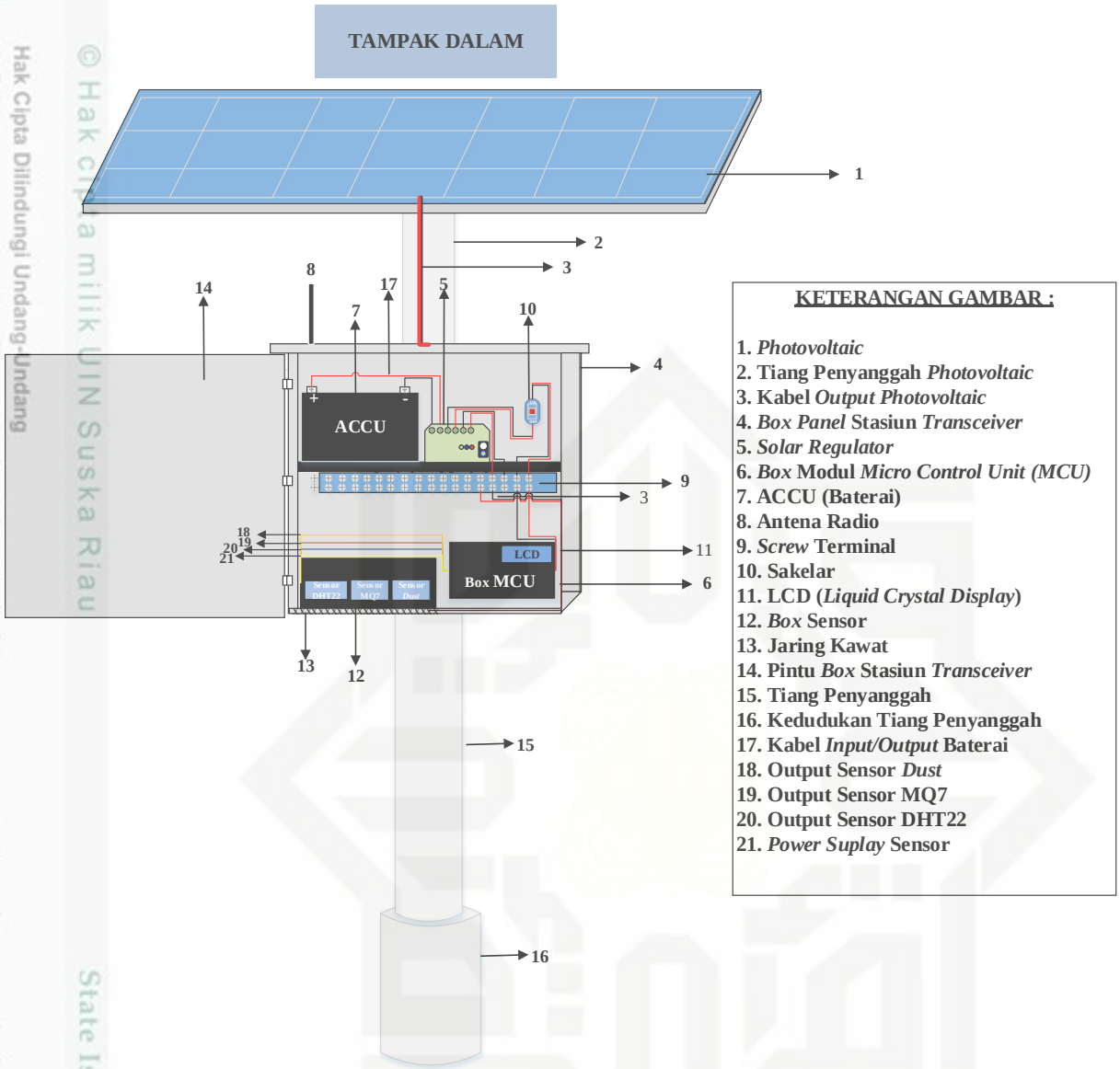


Gambar 3.5. Desain perangkat stasiun penerima (*Receiver*)

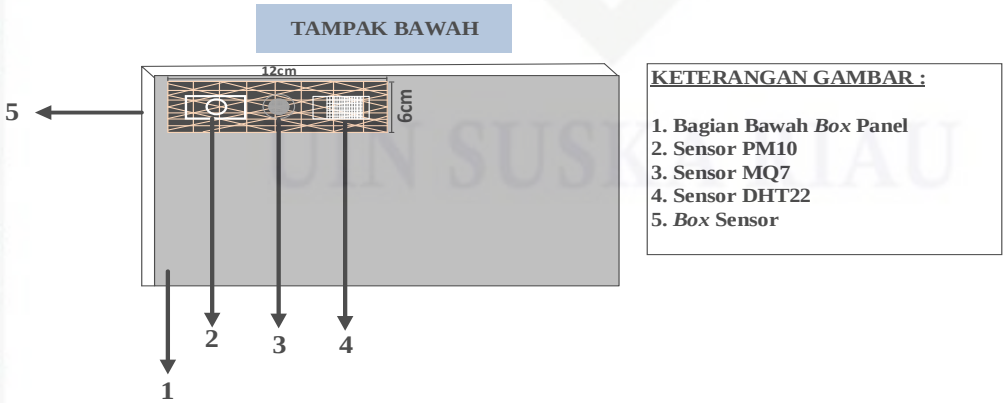


Gambar 3.6. Desain perangkat stasiun *transceiver* tampak luar

Hak Cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 3.7. Desain perangkat stasiun *transceiver* tampak dalam



Gambar 3.8. Desain perangkat stasiun *transceiver* tampak bawah

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau

Penggunaan *box* panel pada perancangan model perangkat dan juga perangkat asli nantinya berfungsi sebagai tempat penyimpanan dan menjaga komponen baterai, *solar* regulator, dan komponen lainnya dari kerusakan baik itu karena faktor alam atau faktor lainnya. Sedangkan *box* plastik digunakan sebagai tempat penyimpanan sensor, RTC, modul Arduino dan komponen lainnya, dengan tujuan melindungi komponen dari korosi dan juga agar penyusunan komponen lebih rapi. *Box* plastik juga digunakan pada perangkat stasiun *receiver* yang berfungsi sebagai tempat penyimpanan komponen modul Arduino dan modul radio dari korosi serta agar penyusunan komponen lebih rapi.

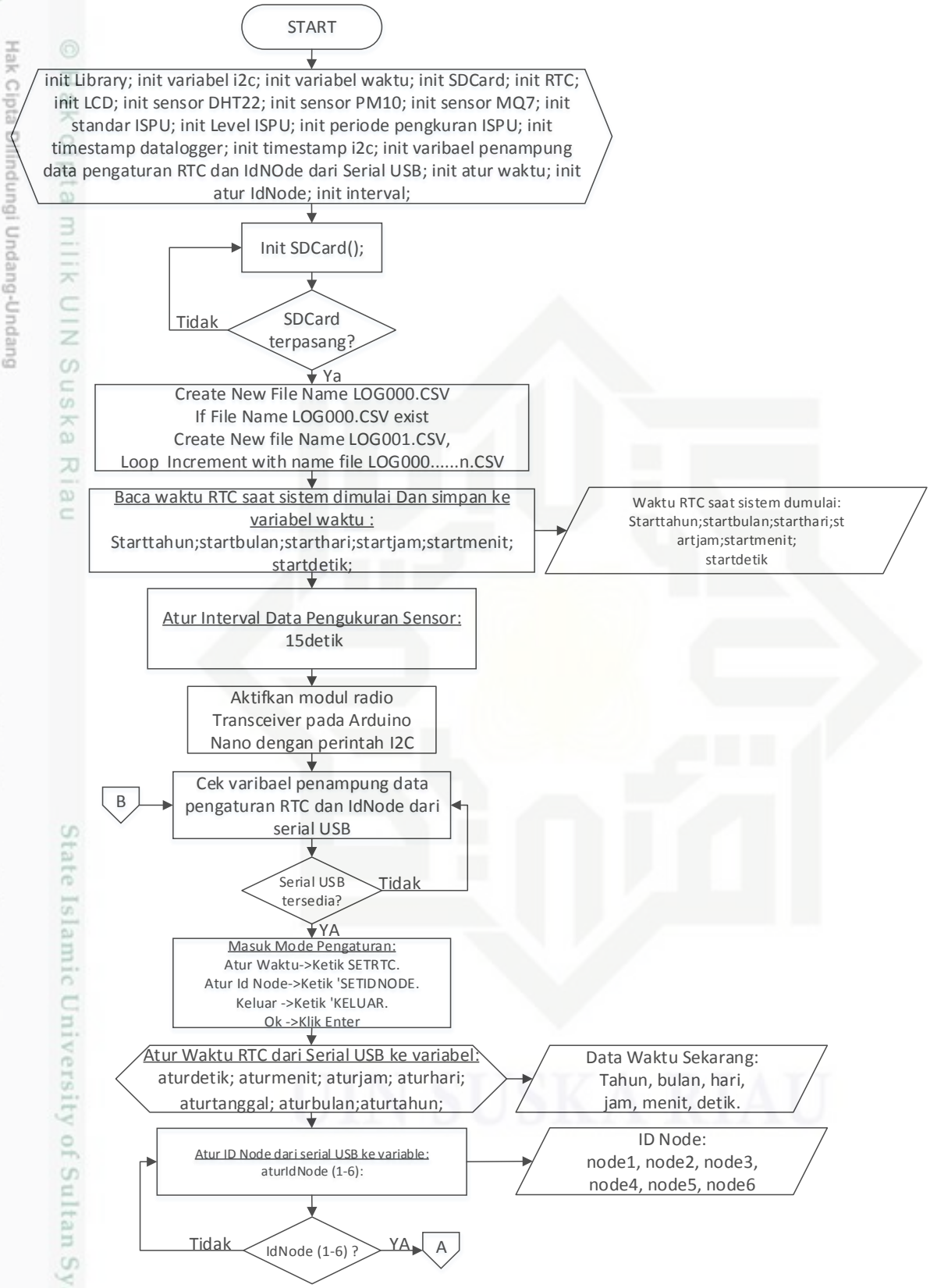
3.5.4. Perancangan *Software* (Perangkat Lunak)

Penelitian ini menggunakan perangkat Arduino Mega 2560 yang berperan sebagai MCU pada sistem data *logger*, dan Arduino Nano sebagai MCU RF modul radio pengiriman data di stasiun *transceiver* dan *receiver*. Maka pemrograman Arduino dilakukan dalam sisi berbeda. Untuk menjalankan dan mengontrol proses kerja sistem, ditanamkan program yang dapat mengatur dan memproses kerja sistem sesuai dengan tujuan yang diinginkan. Perancangan perangkat lunak lainnya juga dilakukan dengan membuat aplikasi *Pemantau* pada komputer *server* yang berperan sebagai pemantau hasil pengukuran sensor dan kondisi dari seluruh *node* di lapangan serta sebagai *database*. Sebelum perancangan perangkat lunak hal yang terpenting yaitu membuat skenario sistem perangkat lunak dengan membuat *flowchart*, pembuatan skenario dilakukan agar perancangan lebih terarah dan lebih jelas.

3.5.4.1. Pemrograman MCU Data *Logger*

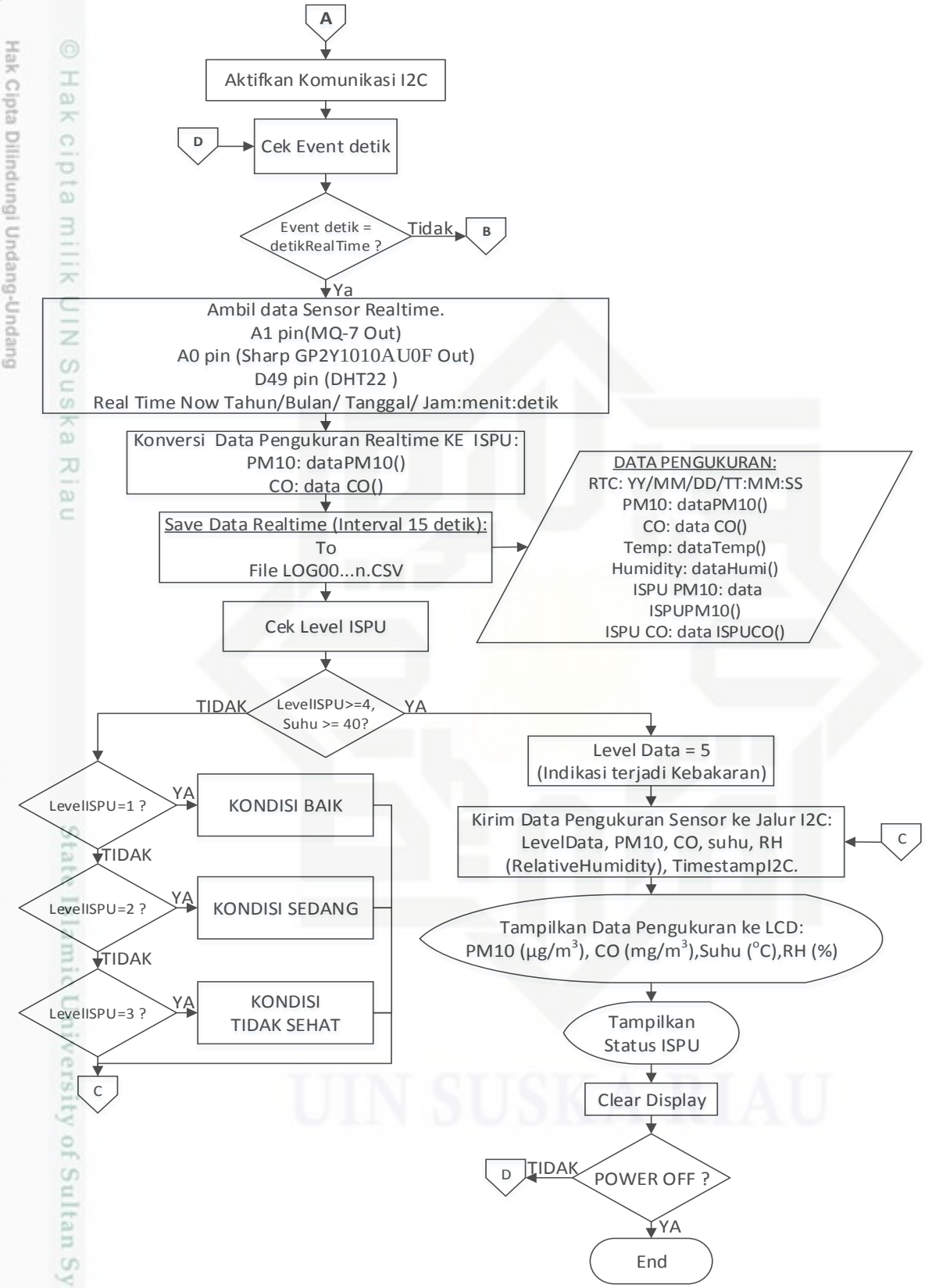
Pemrograman pada MCU data *logger* bertujuan untuk mengendalikan proses pengukuran sensor serta keperluan komunikasi I2C dengan Arduino Nano. Proses kerja program MCU data *logger* dilakukan sesuai dengan *flowchart* pada Gambar 3.9 dan Gambar 3.10.

Hak Cipta dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU



Gambar 3.9. Diagam alur program MCU data *logger*

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau



Gambar 3.10. Diagam alur program MCU data *logger* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Diagram alur program MCU data *logger* dapat dijelaskan sebagai berikut. Terdapat sebuah tombol *switch* disetiap stasiun *transceiver* sistem peringatan dini kebakaran hutan yang berfungsi sebagai penghubung atau pemutus sumber catu daya menuju MCU, jika tombol *switch* diposisikan ON maka sistem akan aktif secara keseluruhan, MCU kemudian melakukan inisial *file library* dan variabel tipe data yang digunakan dalam program, selanjutnya MCU akan memanggil fungsi khusus SD *card*, apabila SD *card* terpasang maka sistem akan membuat sebuah *file* yang dimulai dengan nama file LOG00..n.csv dan jika nama file tersebut sudah ada maka sistem akan membuat file baru LOG001.csv sesuai dengan program yang dibuat, nama *file* akan otomatis berlanjut hingga . Namun jika SD *card* tidak terpasang MCU akan melakukan inisial ulang SD *card* hingga MCU membaca *SDcard* terpasang.

Selanjutnya MCU membaca data waktu dari modul RTC (*Real Time Clock*) DS1302 saat sistem dimulai dan menyimpan data waktu kedalam variabel: starttahun = tahun RTC dimulai, startbulan = bulan RTC dimulai, starthari = hari RTC dimulai, startjam = jam RTC dimulai, startmenit = menitRTC dimulai, dan startdetik = detik RTC dimulai. Kemudian MCU mengatur waktu interval data pengukuran sensor. Setelah itu MCU akan mengaktifkan modul radio *transceiver* pada Arduino Nano melalui jalur komunikasi I2C. Selanjutnya masuk proses cek variabel penampung data pengaturan RTC dan ID node, proses ini berfungsi untuk mengatur waktu RTC sesuai dengan waktu yang sedang berlangsung dan pengaturan ID node dilakukan jika ingin merubah ID node melalui serial USB. Jika variabel penampung data pengaturan RTC dan ID node tersedia maka MCU akan masuk ke mode pengaturan. Selanjutnya masuk ke proses atur RTC dan menyimpannya kedalam variabel: aturdetik = pengaturan waktu detik terkini, aturmenit = pengaturan waktu menit terkini, aturjam = pengaturan waktu jam terkini, aturhari= pengaturan waktu hari terkini, aturtanggal = pengaturan waktu tanggal terkini, aturbulan = pengaturan waktu bulan terkini, aturtahun = pengaturan waktu tahun terkini, Kemudian dilanjutkan masuk proses atur ID node dan menyimpannya kedalam variabel: Idnode1 = ID node 1, IdNode2 = ID node 2, IdNode3 = ID node 3, IdNode4 = ID node 4, IdNode5 = ID node 5, IdNode6 = ID node 6 selanjutnya cek atur IdNode apakah ID node yang diatur 1 sampai 6 jika tidak atur kembali karena node yang digunakan hanya 1 sampai 6 jika ya selanjutnya simpan pengaturan ID node. Kemudian MCU data *logger* akan mengaktifkan komunikasi I2C dengan MCU modul radio *transceiver*.

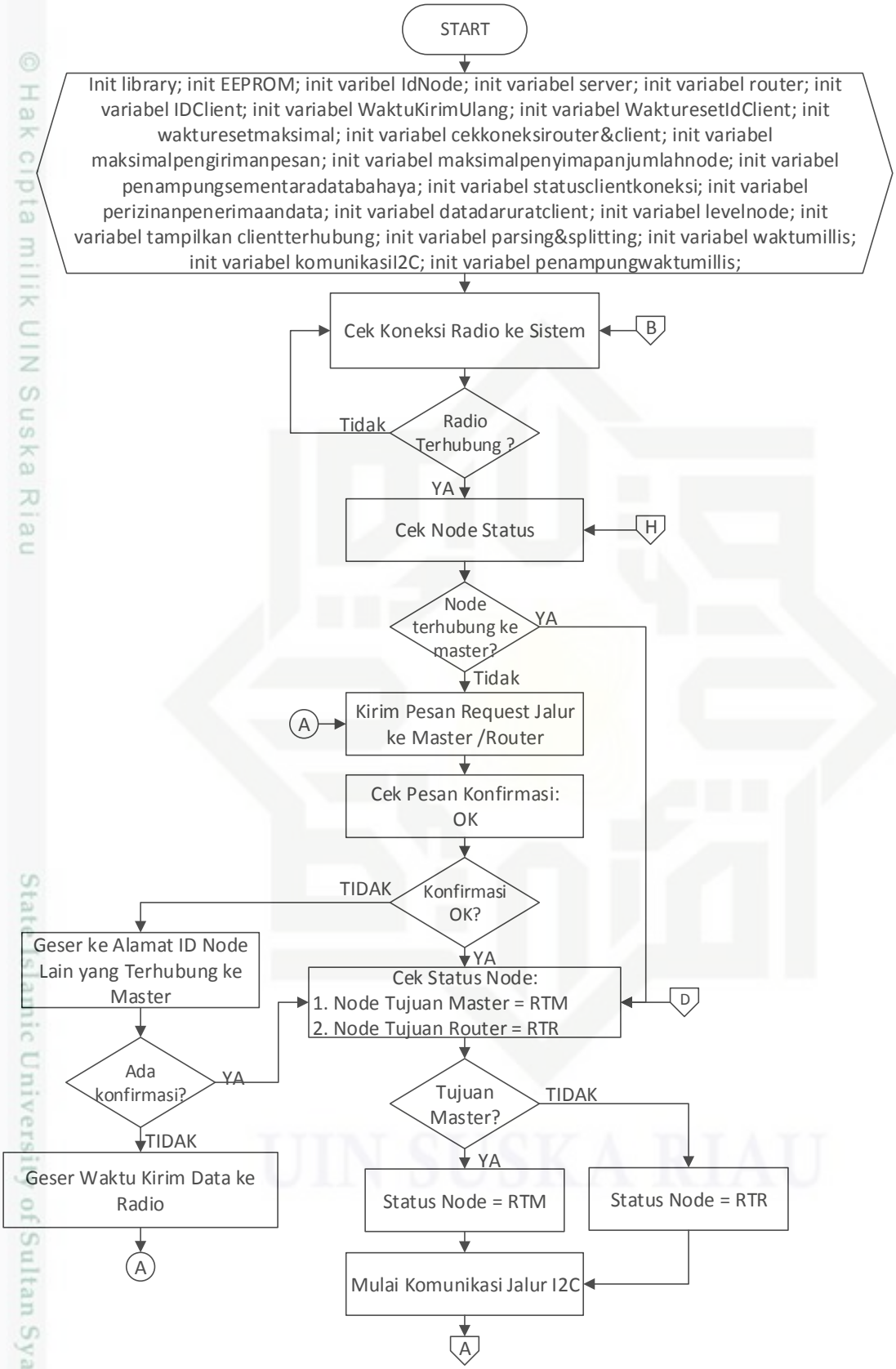
Hak Cipta milik UIN Sultan Syarif Kasim Riau
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Proses selanjutnya yaitu cek waktu event untuk pengambilan data pengukuran sensor, kemudian selanjutnya data pengukuran sensor tersebut dikonversi menjadi data ISPU dan dicek apakah data ISPU > 4 atau suhu >= 40 jika ya berarti level data = 5 yang mengindikasikan terjadinya kebakaran dan level data adalah bahaya. Data selanjutnya dikirim ke MCU modul radio *transceiver* melalui jalur I2C untuk diproses kembali dengan format pengiriman data berikut "LevelData, PM10, CO, suhu, RH, TimestampI2C" dan menyimpan data pengukuran kedalam file LOG00…n.CSV. Namun jika level data pengukuran tidak bahaya, data pengukuran akan langsung dikirim ke jalur I2C dan tampilkan hasil pengukuran ke LCD. Proses akan berhenti hanya jika sistem mati dan akan terus berlanjut jika sistem dalam keadaan hidup.

3.5.4.2. Pemrograman MCU Modul *Transceivers*

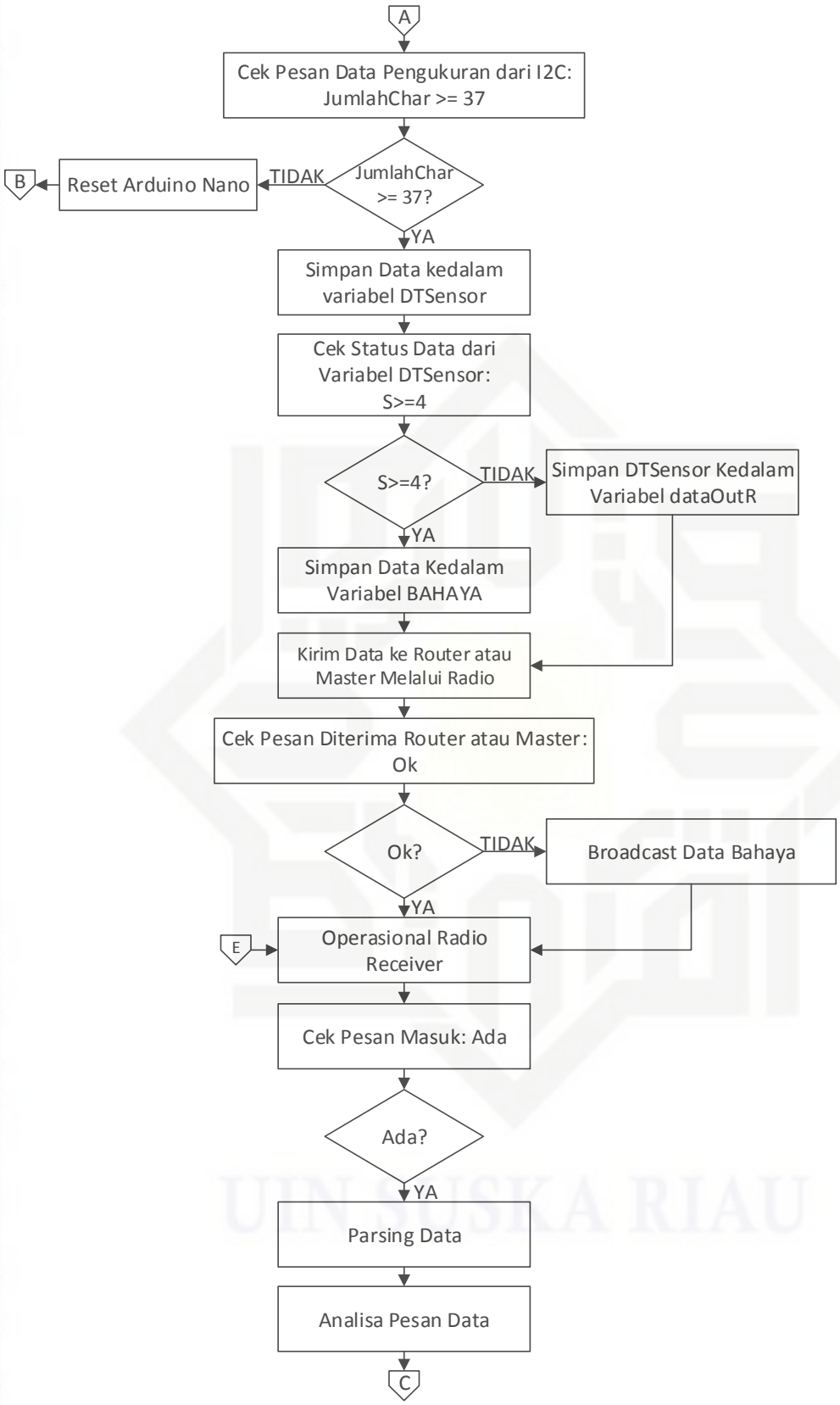
Arduino Nano sebagai *Micro Control Unit* (MCU) untuk modul radio *transceiver* bertugas memproses lebih lanjut hasil pengukuran sensor yang dikirim dari jalur I2C oleh sistem data *logger*. Proses kerja program untuk MCU modul *transceiver* sesuai dengan *flowchart* pada Gambar 3.11, Gambar 3.12, Gambar 3.13, Gambar 3.14 dan Gambar 3.15.

UIN SUSKA RIAU
Hak Cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



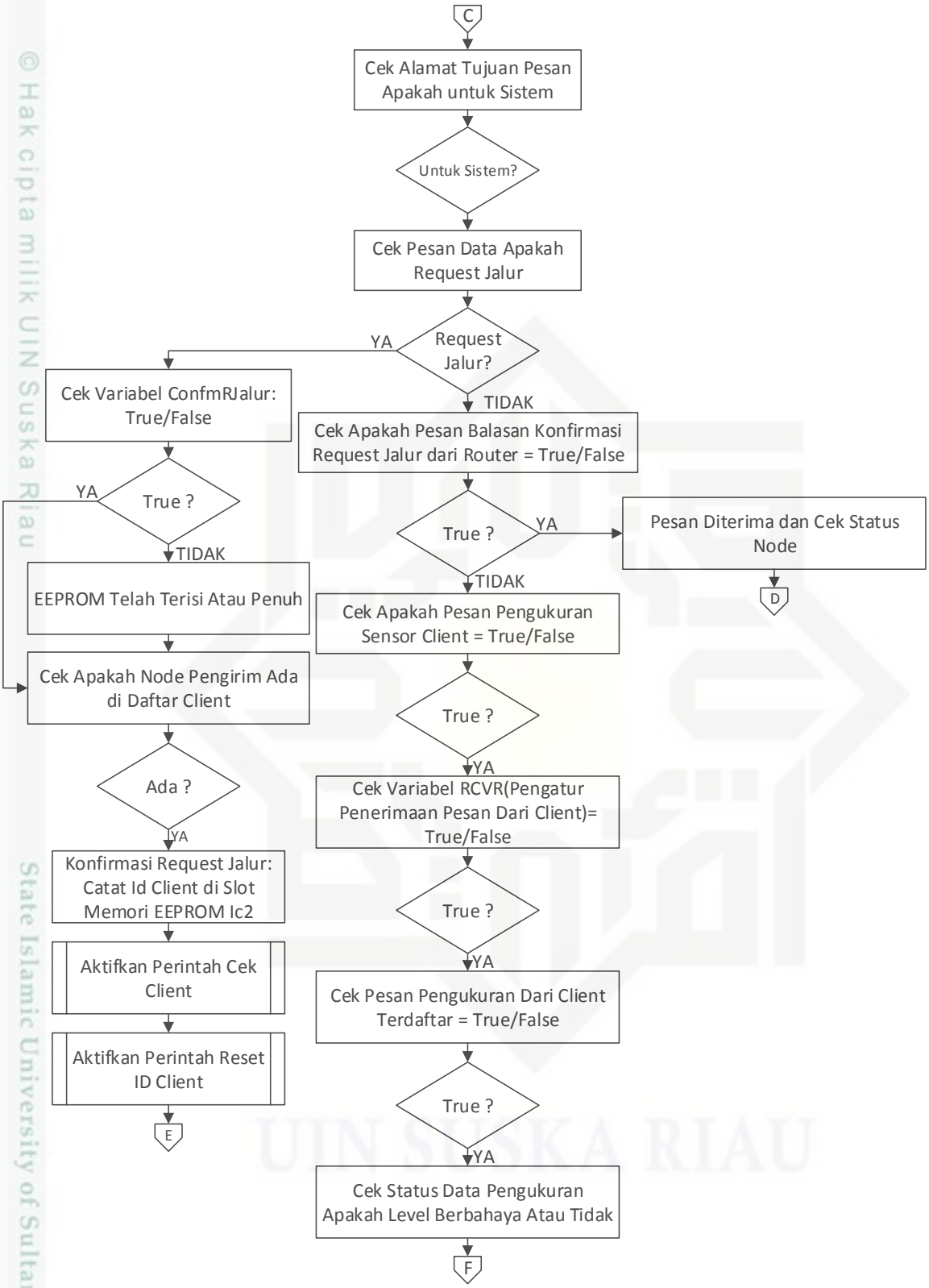
Gambar 3.11. Diagram alur program MCU modul *transceiver*

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



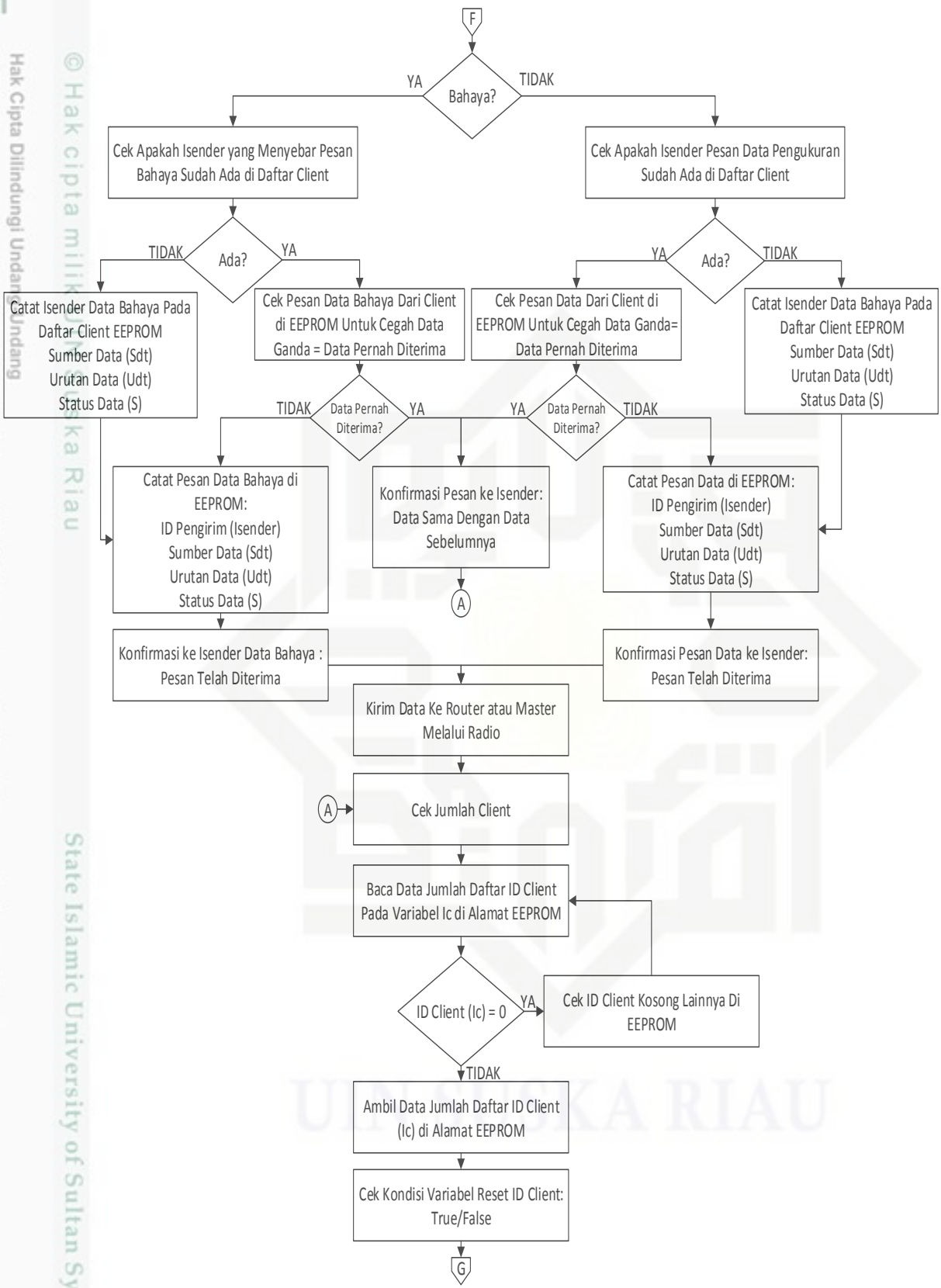
Gambar 3.12. Diagram alur program MCU modul *transceiver* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



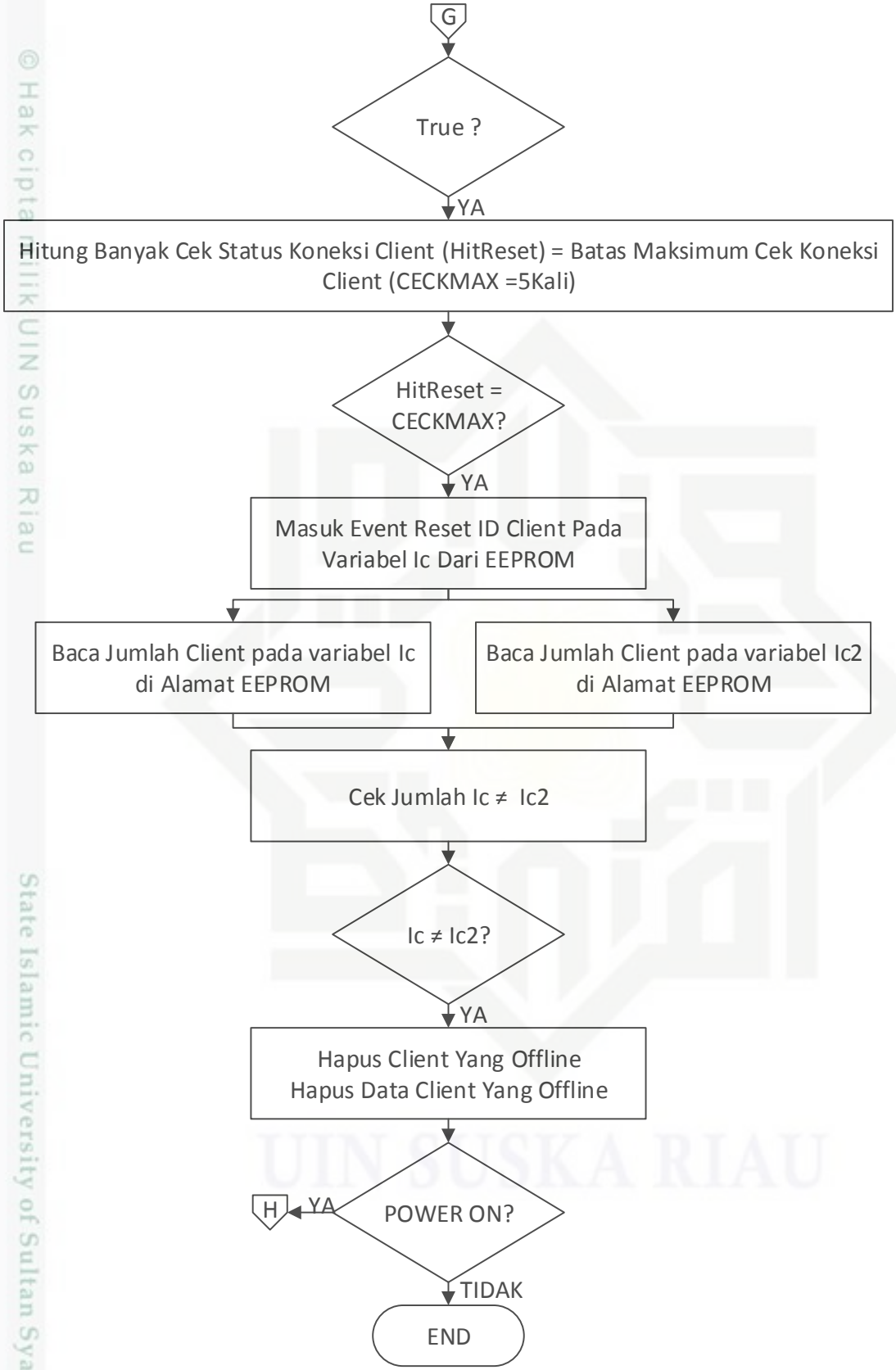
Gambar 3.13. Diagram alur program MCU modul *transceiver* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 3.14. Diagram alur MCU modul *transceiver* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 3.15. Diagram alur program MCU modul *transceiver* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Pada saat sistem dinyalakan, MCU akan melakukan inisialisasi *library* dan variabel data yang digunakan dalam pemrograman kemudian MCU mengecek apakah modul radio terhubung dengan sistem atau tidak. Jika modul radio tidak terhubung maka akan muncul pesan di LCD "*init radio failed*" dan sistem tidak akan melanjutkan proses selanjutnya. Dan jika modul radio terhubung sistem akan melakukan cek *node* status, apakah *node* terhubung dengan stasiun *receiver*, jika ya selanjutnya cek status *node* apakah terhubung ke stasiun *receiver* langsung atau ke *router*. Apabila terhubung ke stasiun *receiver* langsung maka status *node* adalah *Router to Master* (RTM), namun jika terhubung ke *router* maka status *node* adalah *Router to Router*. Dan apabila *node* tidak terhubung ke stasiun *receiver* maka sistem akan mengirim pesan *request* jalur menuju stasiun *receiver* dan apabila ada pesan konfirmasi ok selanjutnya akan melakukan cek status *node* dan jika tidak ada pesan konfirmasi ok berarti koneksi ke stasiun *receiver* tidak terjangkau dan *request* jalur akan digeser ke alamat ID *node* lain yang terhubung ke stasiun *receiver* dan apabila tidak ada konfirmasi maka akan melakukan geser waktu kirim pesan *request* jalur, perintah geser waktu dilakukan untuk keperluan mencegah terjadinya *floading* pesan ke *router* atau *stasiun receiver* dan jika ada konfirmasi ok selanjutnya akan melakukan cek status *node*.

Setelah *node* memiliki jalur koneksi untuk mengirim pesan data menuju stasiun *receiver* sistem akan memulai komunikasi jalur I2C dengan perangkat data *logger*. Apabila ada pesan data pengukuran dari jalur I2C sistem akan melakukan cek pesan data pengkuran apakah jumlah karakter >= 37 jika tidak maka pesan akan diabaikan dan jika pesan data pengukuran >= 37 karakter maka pesan data disimpan ke dalam penampung sementara variabel DTSensor kemudian cek status data dari variabel DTSensor apakah >= 4 jika tidak berarti data bukan data bahaya, simpan data dari variabel DTSensor ke penampung sementara untuk keperluan kirim data melalui radio pada variabel dataOutR, kemudian kirim data ke *router* atau stasiun *receiver* melalui radio. Namun jika status data >= 4 maka pesan data adalah bahaya, kemudian simpan pesan data ke dalam variabel BAHAYA dan kirim data ke *router* atau stasiun *receiver* melalui radio. Apabila tidak ada pesan konfirmasi dari *router* atau stasiun *receiver* maka pesan data bahaya di *broadcast* ke seluruh stasiun,dan jika ada pesan konfirmasi pesan telah diterima *router* atau *receiver*, selanjutnya sistem akan memulai proses operasional radio *receiver* untuk keperluan menerima data dari *node* lainnya.

Hak Cipta milik UIN Suska Riau
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Operasional radio *receiver* merupakan variabel yang digunakan di dalam pemrograman perangkat *transceiver* yang di dalamnya terdapat beberapa variabel penampung sementara pesan data yang diterima dari *node* lainnya. Berikut adalah proses yang dilakukan pada operasional radio *receiver*. Ketika perangkat *transceiver* menerima pesan data dari *node* lainnya data akan diparsing atau dicek setiap karakter paket data kemudian pesan data tersebut dianalisa. Dalam proses analisa pesan data yang masuk pertama kali sistem akan melakukan pengecakan apakah pesan data untuk sistem atau bukan, jika bukan maka pesan data akan diabaikan. Namun jika pesan data untuk sistem pesan data akan dicek kembali, apakah pesan data tersebut adalah pesan *request* jalur.

Jika pesan data adalah *request* jalur sistem akan melakukan pengecekan apakah variabel konfirmasi jalur dalam kondisi *true* jika kondisi *true* sistem akan melakukan pengecekan apakah *node* pengirim pesan *request* ada didaftar *client* dalam slot EEPROM Ic jika ada maka pesan *request* jalur akan dikonfirmasi ok dan ID *client* tidak dicatat kembali di slot EEPROM Ic namun di slot EEPROM Ic2 atau daftar *client Bufer* yang bertujuan untuk keperluan pengecekan koneksi. Dan apabila variabel konfirmasi jalur dalam kondisi *false* kemungkinan slot EEPROM telah terisi atau penuh, maka sistem akan kembali melakukan pengecekan apakah *node* pengirim pesan *request* ada didaftar *client* dalam slot EEPROM Ic jika ada maka pesan *request* jalur akan dikonfirmasi ok dan ID *client* tidak dicatat kembali di slot EEPROM Ic namun di slot EEPROM Ic2 atau daftar *client bufer* yang bertujuan untuk keperluan pengecekan koneksi. Setelah pesan *request* jalur dikonfirmasi sistem akan mulai mengaktifkan perintah cek *client* dan perintah reset ID *client*. Yang bertujuan untuk keperluan hapus *client* dan akan kembali ke proses awal operasional radio *receiver*.

Namun jika pesan data bukan *request* jalur sistem akan melakukan pengecekan apakah pesan balasan konfirmasi *request* jalur dari *router* atau stasiun *receiver*. Jika ya, maka sistem akan kembali menuju proses cek status *node*. Dan jika pesan data yang masuk bukan pesan balasan konfirmasi konfirmasi *request* maka kemungkinan pesan data yang masuk adalah pesan data pengukuran dari *node* lainnya, sistem akan melakukan pengecekan kembali apakah pesan pengukuran sensor. Jika pesan data adalah pesan pengukuran sensor sistem akan mengecek apakah variabel RCVR dalam kondisi *true* atau *false*. Variabel RCVR berfungsi sebagai pengatur penerimaan pesan data dari *client* terdaftar. Jika RCVR dalam kondisi *false* maka pesan data pengukuran diabaikan karena sistem sedang melakukan komunikasi dengan *node*

UIN SUSKA RIAU
Hak Cipta milik UIN Suska Riau State Islamic University of Sultan Syarif Kasim Riau
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

lainnya, dan jika RCVR dalam kondisi *true*, sistem akan mengecek status data pengukuran apakah level berbahaya atau tidak.

Apabila pesan data yang diterima merupakan data bahaya maka akan menjadi pesan yang diprioritaskan oleh sistem untuk diproses terlebih dulu. Sistem akan melakukan pengecekan apakah ID *client* yang mengirimkan pesan bahaya sudah ada didaftar *client*. Jika ada, sistem akan melakukan pengecekan kembali apakah pesan bahaya tersebut pernah diterima atau tidak, untuk menghindari terjadinya data ganda dan juga untuk menghindari terjadinya *floading* penerimaan pesan data dari *client* lainnya. Jika data sudah pernah diterima maka sistem akan mengirim konfirmasi kepada *client* yang mengirim pesan data bahaya bahwa pesan data tersebut sama dengan sebelumnya. Namun jika pesan data bahaya belum pernah diterima data bahaya akan disimpan kedalam memori EEPROM dan sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data bahaya bahwa data bahaya telah diterima dan selanjutnya data bahaya dikirim menuju stasiun *receiver* atau *router*. Dan jika ID *client* yang mengirimkan pesan data bahaya belum ada pada daftar *client* sistem akan menyimpan ID *client* dan data bahaya di memori EEPROM kemudian sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data bahaya dan selanjutnya data bahaya dikirim menuju stasiun *receiver* atau *router*.

Namun apabila pesan data yang diterima bukan data bahaya, sistem akan memproses pesan data pengukuran setelah pesan data bahaya, sistem akan mengecek apakah ID *client* yang mengirim pesan data pengukuran ada sudah ada didaftar *client* atau tidak. Jika ID *client* yang mengirimkan pesan data pengukuran belum ada pada daftar *client* sistem akan menyimpan ID *client* dan data pengukuran di memori EEPROM kemudian sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data pengukuran dan selanjutnya data pengukuran dikirim menuju stasiun *receiver* atau *router*. Dan jika ada, sistem akan melakukan pengecekan kembali apakah pesan data pengukuran tersebut pernah diterima atau tidak, untuk menghindari terjadinya data ganda dan juga untuk menghindari terjadinya *floading* penerimaan pesan data dari *client* lainnya. Jika data sudah pernah diterima maka sistem akan mengirim konfirmasi kepada *client* yang mengirim pesan data pengukuran bahwa pesan data tersebut sama dengan sebelumnya. Namun jika pesan data pengukuran belum pernah diterima data pengukuran akan disimpan kedalam memori EEPROM dan sistem akan mengirim pesan

Hak Cipta dilindungi undang-undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

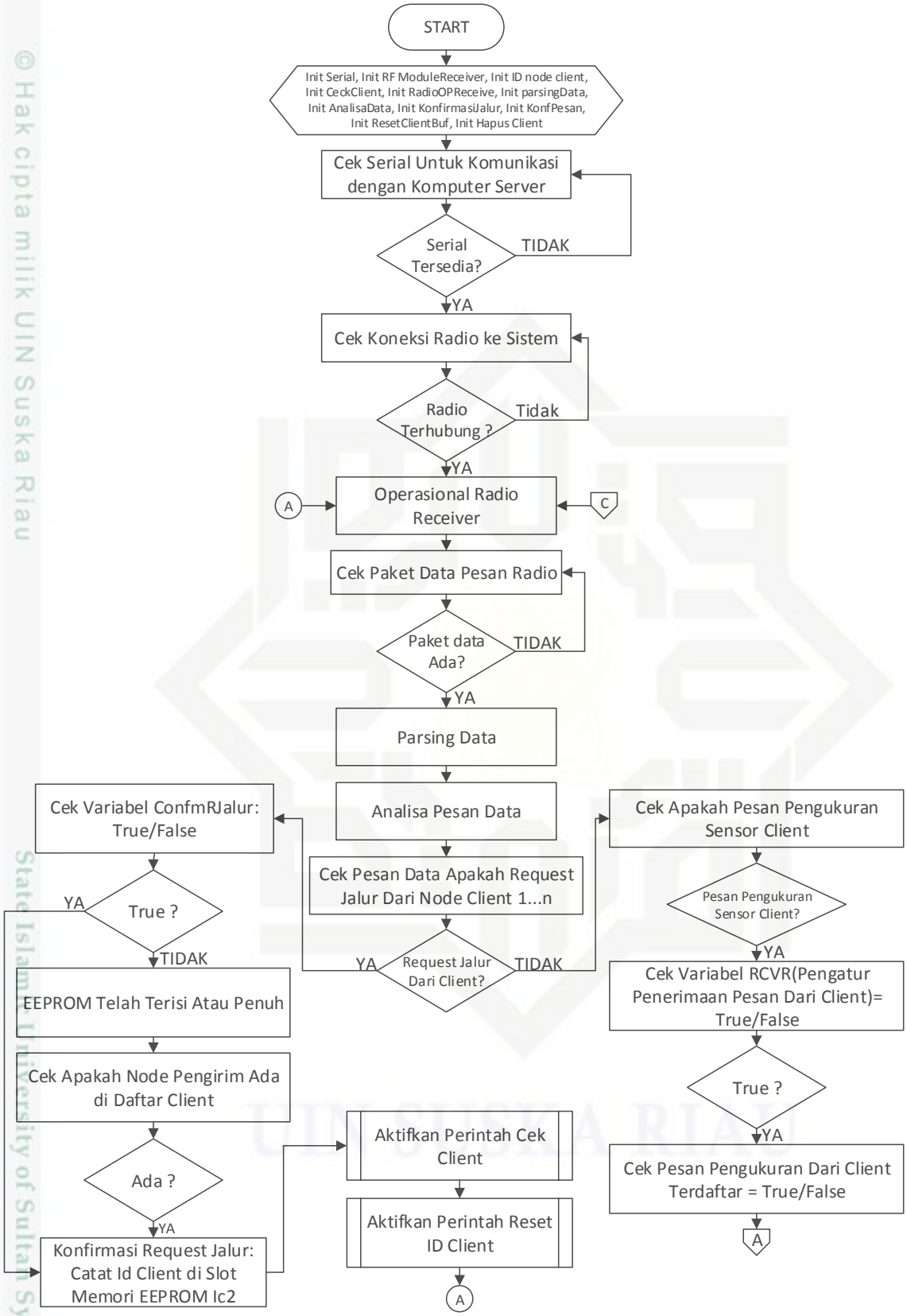
konfirmasi kepada *client* yang mengirim pesan data pengukuran bahwa data telah diterima dan selanjutnya data pengukuran dikirim menuju stasiun *receiver* atau *router*.

Proses selanjutnya adalah cek *client*, proses ini adalah proses yang rutin atau terus menerus dilakukan oleh sistem, proses cek *client* dilakukan dengan mengecek daftar ID *client* pada variabel Ic di alamat EEPROM, apabila daftar ID *client* pada alamat EEPROM = 0, sistem akan terus melakukan pengecekan setiap lot memori EEPROM, namun apabila daftar ID *client* pada alamat EEPROM ≠ 0, menandakan ada slot memori EEPROM yang terisi, sistem akan mengambil data jumlah ID *client* yang terdaftar di alamat EEPROM dan akan memulai cek kondisi variabel reset ID *client*. Apabila variabel cek status koneksi (HitReset) = batas maksimum cek koneksi *client* (CECKMAX) maka sistem akan masuk *event* reset ID *client* pada variabel Ic dan Ic2 di alamat EEPROM. Setelah sistem masuk proses *event* reset ID *client* sistem akan membaca jumlah *client* pada variabel Ic dan variabel Ic2 di alamat EEPROM. Apabila jumlah ID *client* yang terbaca di Ic ≠ Ic2 maka sistem akan menghapus daftar *client* yang *offline*, dan menghapus data *client* yang *offline*. Proses akan berhenti hanya jika sistem mati dan akan terus berlanjut jika sistem dalam keadaan hidup.

3.5.4.3. Pemrograman MCU Modul *Receiver*

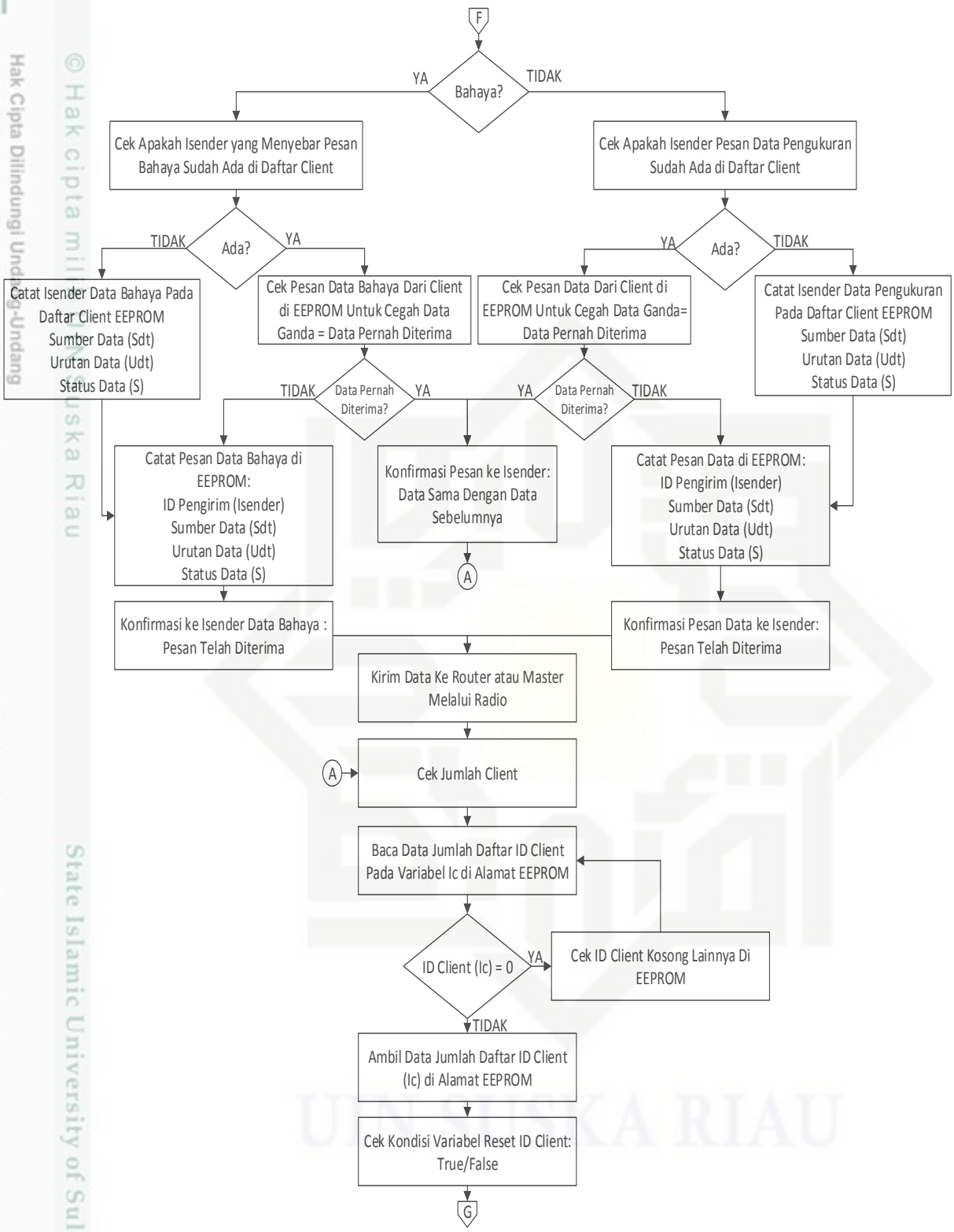
Stasiun *receiver* bekerja sebagai penerima pesan data dari stasiun *transceiver* dan mengirimkan pesan data tersebut ke komputer *server*. Untuk menjalankan proses tersebut dibutuhkan program untuk mengendalikan MCU dalam proses penerimaan pesan data serta proses lebih lanjut. Proses kerja program untuk MCU modul *receiver* sesuai dengan *flowchart* pada Gambar 3.16, Gambar 3.17 dan Gambar 3.18.

Hak Cipta dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU



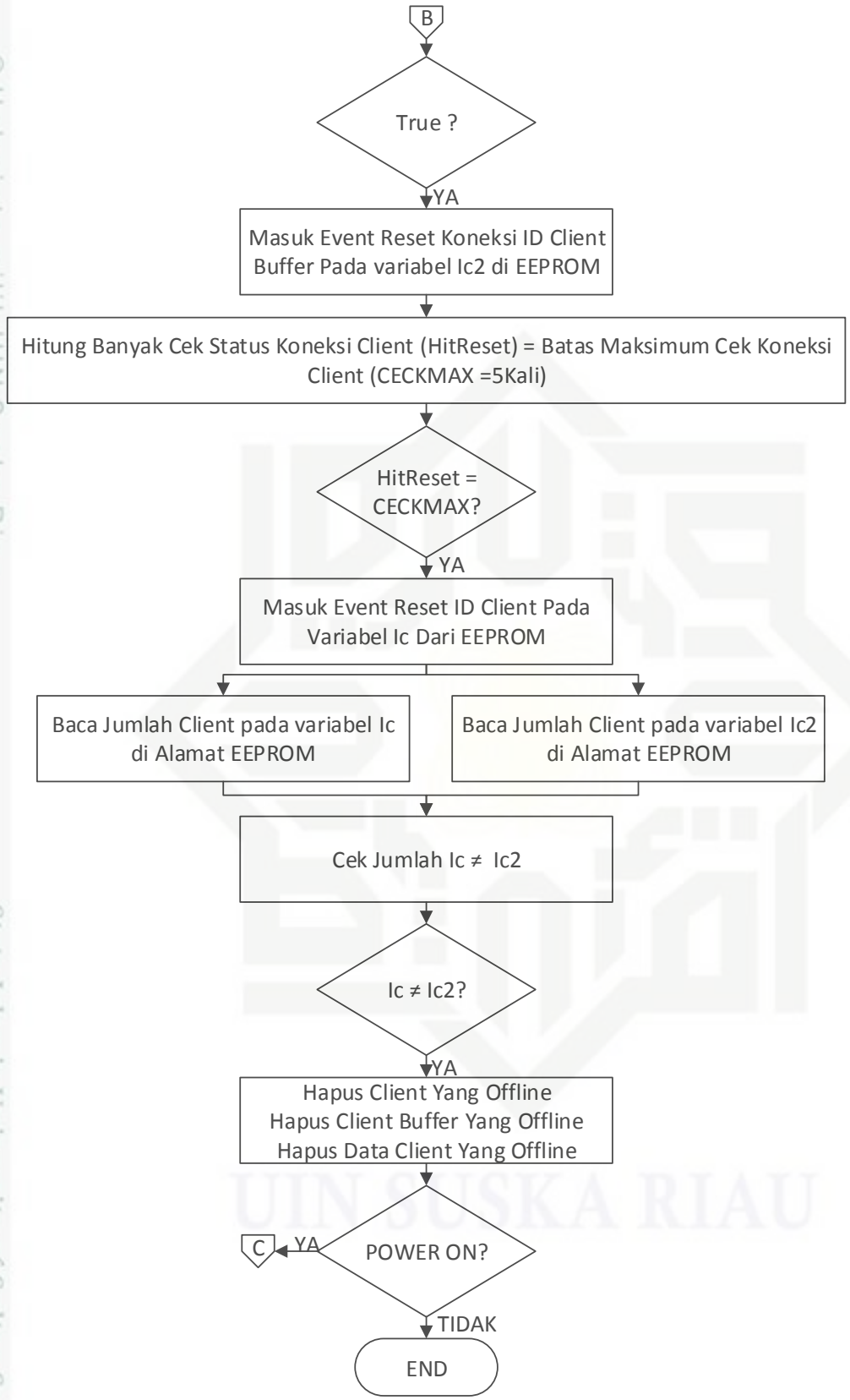
Gambar 3.16. Diagram alur program MCU modul *receiver*

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 3.17. Diagram alur program MCU modul *receiver* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 3.18. Diagram alur program MCU modul *receiver* (lanjutan)

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Berikut adalah penjelasan proses kerja diagram alur program pada MCU modul *receiver*. Pada saat sistem dinyalakan, MCU akan melakukan inisialisasi *library* dan variabel data yang digunakan dalam pemrograman kemudian MCU mengecek serial komunikasi dengan komputer *server* apakah serial komunikasi tersedia atau tidak, jika tidak MCU akan melakukan pengulangan cek kembali hingga serial komunikasi antara MCU modul *receiver* dengan komputer *server*. Selanjutnya MCU akan melakukan pengecekan kembali pada modul radio, apakah terhubung dengan sistem atau tidak. Jika modul radio tidak terhubung maka akan muncul pesan di LCD "*init radio failed*" dan sistem tidak akan melanjutkan proses selanjutnya. Dan jika modul radio terhubung sistem akan memulai proses operasional radio *receiver*.

Proses operasional radio *receiver* merupakan variabel yang digunakan di dalam pemrograman perangkat *transceiver* yang di dalamnya terdapat beberapa variabel penampung sementara pesan data yang diterima dari *node* lainnya. Berikut adalah proses yang dilakukan pada operasional radio *receiver*. Ketika perangkat *transceiver* menerima pesan data dari *node* lainnya data akan diparsing atau dicek setiap karakter paket data kemudian pesan data tersebut dianalisa. Dalam proses analisa pesan data yang masuk, sistem akan melakukan pengecekan apakah pesan data tersebut adalah pesan *request* jalur. Jika pesan data adalah *request* jalur sistem akan melakukan pengecekan apakah variabel konfirmasi jalur dalam kondisi *true* jika kondisi *true* sistem akan melakukan pengecekan apakah *node* pengirim pesan *request* ada didaftar *client* dalam slot EEPROM Ic jika ada maka pesan *request* jalur akan dikonfirmasi ok dan ID *client* tidak dicatat kembali di slot EEPROM Ic namun di slot EEPROM Ic2 atau daftar *client Bufer* yang bertujuan untuk keperluan pengecekan koneksi. Dan apabila variabel konfirmasi jalur dalam kondisi *false* kemungkinan slot EEPROM telah terisi atau penuh, maka sistem akan kembali melakukan pengecekan apakah *node* pengirim pesan *request* ada didaftar *client* dalam slot EEPROM Ic jika ada maka pesan *request* jalur akan dikonfirmasi ok dan ID *client* tidak dicatat kembali di slot EEPROM Ic namun di slot EEPROM Ic2 atau daftar *client bufer* yang juga bertujuan untuk keperluan pengecekan koneksi. Setelah pesan *request* jalur dikonfirmasi sistem akan mulai mengaktifkan perintah cek *client* dan perintah reset ID *client*. Yang bertujuan untuk keperluan hapus *client* dan akan kembali ke proses awal operasional radio *receiver*.

Namun jika pesan data bukan *request* jalur sistem akan melakukan pengecekan apakah pesan data pengukuran sensor. Jika pesan data adalah pesan data pengukuran sensor sistem akan mengecek apakah variabel RCVR dalam kondisi *true* atau *false*. Variabel RCVR berfungsi

Hak Cipta milik UIN Sultan Syarif Kasim Riau
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

sebagai pengatur penerimaan pesan data dari *client* terdaftar. Jika RCVR dalam kondisi *false* maka pesan data pengukuran diabaikan karena sistem sedang melakukan komunikasi dengan *node* lainnya, dan jika RCVR dalam kondisi *true*, sistem akan memproses pesan data pengukuran tersebut, selanjutnya sistem mengecek status data pengukuran apakah level berbahaya atau tidak.

Apabila pesan data yang diterima merupakan data bahaya maka akan menjadi pesan yang diprioritaskan oleh sistem untuk diproses terlebih dulu. Sistem akan melakukan pengecekan apakah ID *client* yang mengirimkan pesan bahaya sudah ada didaftar *client*. Jika ada, sistem akan melakukan pengecekan kembali apakah pesan bahaya tersebut pernah diterima atau tidak, untuk menghindari terjadinya data ganda dan juga untuk menghindari terjadinya *floading* penerimaan pesan data dari *client* lainnya. Jika data sudah pernah diterima maka sistem akan mengirim konfirmasi kepada *client* yang mengirim pesan data bahaya bahwa pesan data tersebut sama dengan sebelumnya. Namun jika pesan data bahaya belum pernah diterima data bahaya akan disimpan kedalam memori EEPROM dan sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data bahaya bahwa data bahaya telah diterima dan selanjutnya data bahaya dikirim menuju komputer *server* untuk ditampilkan di aplikasi *Pemantau*. Dan jika ID *client* yang mengirimkan pesan data bahaya belum ada pada daftar *client* sistem akan menyimpan ID *client* dan data bahaya di memori EEPROM kemudian sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data bahaya dan selanjutnya data bahaya dikirim menuju komputer *server* untuk diproses dan ditampilkan di aplikasi *Pemantau*.

Namun apabila pesan data yang diterima bukan data bahaya, sistem akan memproses pesan data pengukuran setelah pesan data bahaya, sistem akan mengecek apakah ID *client* yang mengirim pesan data pengukuran sudah ada didaftar *client* atau tidak. Jika ID *client* yang mengirimkan pesan data pengukuran belum ada pada daftar *client* sistem akan menyimpan ID *client* dan data pengukuran di memori EEPROM kemudian sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data pengukuran dan selanjutnya data pengukuran dikirim menuju komputer *server* untuk diproses dan ditampilkan di aplikasi *Pemantau*. Dan jika ada, sistem akan melakukan pengecekan kembali apakah pesan data pengukuran tersebut pernah diterima atau tidak, untuk menghindari terjadinya data ganda dan juga untuk menghindari terjadinya *floading* penerimaan pesan data dari *client* lainnya. Jika data

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak Cipta milik UIN Sultan Syarif Kasim Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

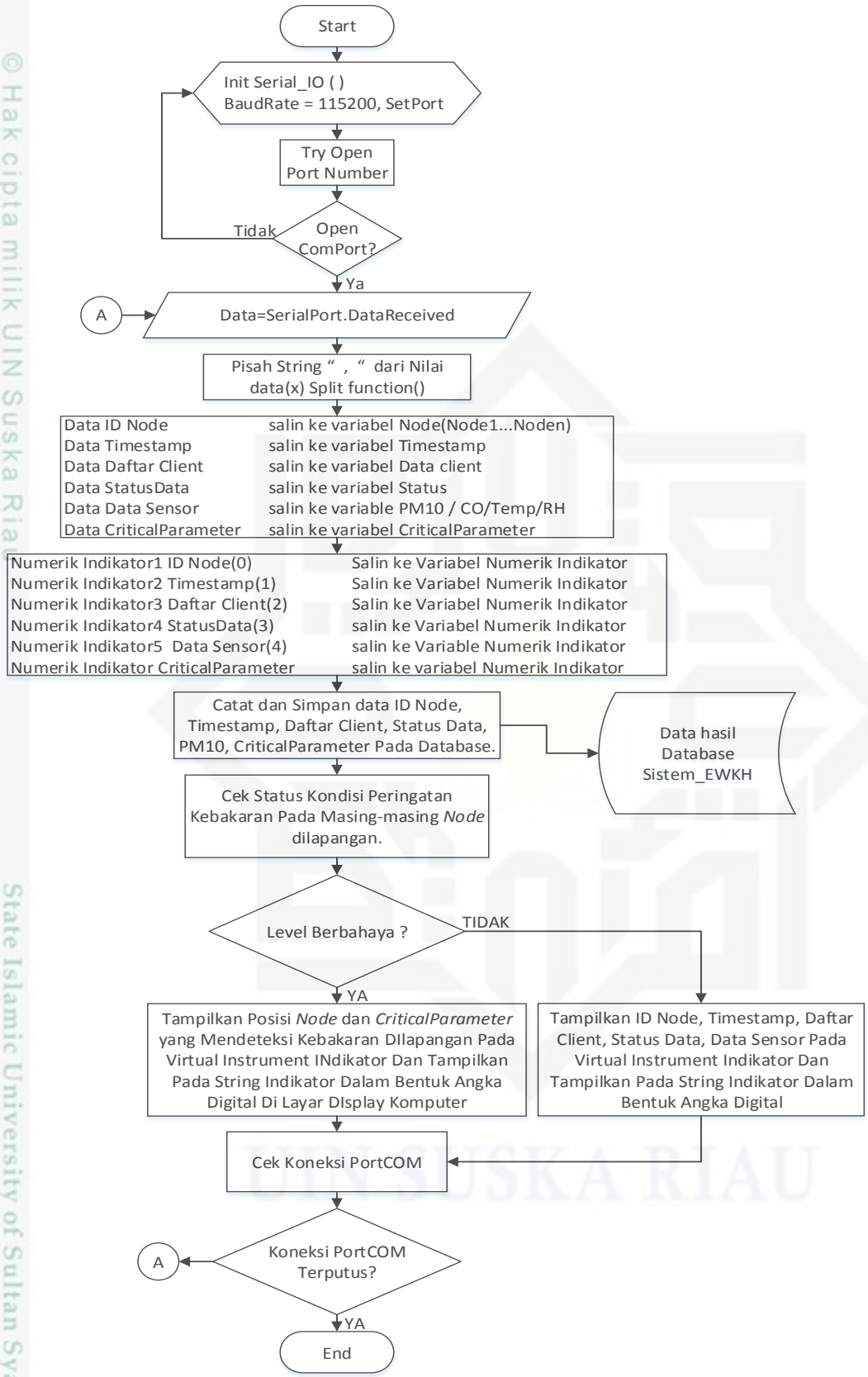
sudah pernah diterima maka sistem akan mengirim konfirmasi kepada *client* yang mengirim pesan data pengukuran bahwa pesan data tersebut sama dengan data sebelumnya. Namun jika pesan data pengukuran belum pernah diterima data pengukuran akan disimpan kedalam memori EEPROM dan sistem akan mengirim pesan konfirmasi kepada *client* yang mengirim pesan data pengukuran bahwa data telah diterima dan selanjutnya data pengukuran dikirim menuju komputer *server* untuk diproses dan ditampilkan di aplikasi *Pemantau*.

Proses selanjutnya adalah cek *client*, proses ini adalah proses yang rutin atau terus menerus dilakukan oleh sistem, proses cek *client* dilakukan dengan mengecek daftar ID *client* pada variabel Ic di alamat EEPROM, apabila daftar ID *client* pada alamat EEPROM = 0, sistem akan terus melakukan pengecekan setiap lot memori EEPROM, namun apabila daftar ID *client* pada alamat EEPROM ≠ 0, menandakan ada slot memori EEPROM yang terisi, sistem akan mengambil data jumlah ID *client* yang terdaftar di alamat EEPROM dan akan memulai cek kondisi variabel reset ID *client*. Apabila variabel cek status koneksi (HitReset) = batas maksimum cek koneksi *client* (CECKMAX) maka sistem akan masuk *event* reset ID *client* pada variabel Ic dan Ic2 di alamat EEPROM. Setelah sistem masuk proses *event* reset ID *client* sistem akan membaca jumlah *client* pada variabel Ic dan variabel Ic2 di alamat EEPROM. Apabila jumlah ID *client* yang terbaca di Ic ≠ Ic2 maka sistem akan menghapus daftar *client* yang *offline*, dan menghapus data *client* yang *offline*. Proses akan berhenti hanya jika sistem mati dan akan terus berlanjut jika sistem dalam keadaan hidup.

3.5.4.4. Pemrograman Aplikasi *Interface*

Paket data yang telah dikirim oleh MCU modul *receiver* melalui *serial* USB menuju komputer selanjutnya akan diolah untuk ditampilkan dalam besaran data digital berbasis *Graphical User Pemantau* (GUI) pada aplikasi *interface*. Data hasil pengukuran oleh *node* serta waktu pengambilan data akan disimpan pada *database*. Data hasil pengukuran nantinya akan disajikan dalam bentuk tampilan gambar dan numerik. Proses kerja aplikasi *Pemantau* dapat dilihat pada Gambar 3.19.

UIN SUSKA RIAU
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU



Gambar 3.19. Diagram alur program aplikasi pemantau

© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Berikut adalah penjelasan diagram alur program aplikasi *Pemantau* di stasiun pemantau. Pada saat aplikasi pemantau dioperasikan dan koneksi *serial* USB stasiun *receiver* dihubungkan pada komputer *server*, aplikasi *Pemantau* akan melakukan inisial *port* I/O stasiun *receiver*, jika *serial port* terhubung, kemudian pesan data dari *node* yang diterima oleh stasiun *receiver* dikirim oleh MCU modul *receiver* menuju komputer dan disimpan kedalam variabel penampung berikut: data ID node simpan ke variabel node (node1...noden), data timestamp simpan ke variabel timestamp, data daftar client simpan ke variabel daftar client, data status data simpan ke variabel status, data DTSensor simpan ke variable PM10, CO, TEMP, RH, data criticalparameter simpan ke variabel criticalparameter. Setelah data disimpan kedalam variabel penampung kemudian data disalin ke variabel numerik di virtual instrument indikator, untuk selanjutnya disimpan dalam *database*. Setelah data disimpan aplikasi akan mengecek status kondisi peringatan kebakaran pada masing-masing *node* di lapangan apakah berbahaya atau tidak. Apabila status tidak berbahaya maka aplikasi hanya akan menampilkan data ID *node*, Timestamp, daftar *client*, status data, DTSensor pada virtual Instrument Indikator. Namun apabila status berbahaya, maka aplikasi akan menampilkan posisi *node* dan *criticalparameter* yang mendeteksi indikasi terjadinya kebakaran pada *virtual instrument* indikator dan juga pada *string* indikator dalam bentuk angka digital di layar komputer *server*. Proses kerja program aplikasi *Pemantau* akan terus berlanjut apabila koneksi *serial* PortCOM antara MCU modul *receiver* dengan komputer terhubung, dan proses hanya akan berhenti apabila koneksi *serial* PortCOM terputus.

3.6. Variabel Data yang Dibutuhkan

Variabel data yang dibutuhkan dalam proses pengumpulan data untuk mengetahui kinerja alat adalah sebagai berikut:

1. Waktu yang dibutuhkan sistem memberikan peringatan dan lokasi kebakaran, yang terhitung dari awal sistem mendeteksi adanya indikasi kebakaran sampai sistem memberikan peringatan di *server*.
2. Kemampuan sistem memberikan peringatan dan lokasi kebakaran hutan.
3. Jumlah data yang dikirim dari semua *node* dan jumlah data yang diterima *server*.
4. Nilai data yang dikirim dari semua *node* dan nilai data semua *node* yang diterima *server*

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

3.7.Uji Kelayakan

Uji kelayakan merupakan tahapan pengujian perangkat secara keseluruhan untuk mengetahui apakah alat yang telah dirancang dapat menjawab rumusan masalah dalam penelitian ini dan mencapai tujuan dari rancang bangun peringatan dini kebakaran hutan. Uji kelayakan dilakukan dengan menggunakan metode kuantitatif dan kualitatif, metode kualitatif digunakan untuk mengetahui kemampuan dan kualitas sistem yang telah dirancang melalui pengujian keseluruhan dan menganalisa sistem berdasarkan variabel yang telah ditentukan. Sedangkan metode kuantitatif digunakan untuk mengetahui kemampuan dan kualitas sistem yang telah dirancang melalui penilaian responden. Dalam proses uji kelayakan ini, diharapkan dapat menjawab parameter *usability, simplicity* dan *interactivity*.

3.7.1. Kemampuan Sistem Memberi Peringatan Dini dan Lokasi Kebakaran Hutan.

Uji kemampuan sistem memberi peringatan dini kebakaran dan lokasi kebakaran kepada *server* merupakan pengujian sistem untuk mengetahui kemampuan masing-masing stasiun *trasnceiver* melakukan *transfer* data menuju *server* serta untuk melihat kemampuan aplikasi *interface* memberikan informasi kebakaran hutan yang terjadi. Untuk mendapatkan hasil tersebut, uji kelayakan dilakukan dengan membuat kebakaran buatan dilokasi semua *node*. Hasil yang didapat pada uji kelayakan ini berupa respon sistem peringatan dini kebakaran hutan, data hasil uji kelayakan dicatat pada tabel. Pada tabel status peringatan di *server* akan diberi tanda (✓) pada kolom tabel yang telah tersedia sesuai dengan kemampuan sistem.

Berikut adalah contoh tabel uji kelayakan kemampuan sistem dalam memberikan informasi peringatan dini dan lokasi kebakaran hutan menuju *server*.

Tabel 3.1 Contoh tabel uji kelayakan kemampuan sistem peringatan dini kebakaran hutan

Lokasi Kebakaran Buatan		Status Peringatan di Aplikasi *Interface*	
		Sampai	Tidak Sampai
ID *Node*	*Node* 1		
	Node 2		
	Node 3		
	Node 5		
	Node 4		

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

3.7.2. Waktu Yang Dibutuhkan Sistem Memberi Peringatan Dan Lokasi Kebakaran

Uji kelayakan selanjutnya yaitu untuk melihat waktu yang dibutuhkan masing-masing *node* saat melakukan *transfer* data peringatan dini dan lokasi kebakaran, yang terhitung dari awal sistem mendeteksi adanya indikasi kebakaran sampai sistem memberikan peringatan di stasiun pemantau. Untuk mendapatkan hasil tersebut, uji kelayakan dilakukan dengan membuat pembakaran buatan dilokasi semua *node* secara bergantian. Hasil yang didapat pada uji kelayakan ini berupa waktu, data hasil uji kelayakan dicatat pada tabel. Berikut adalah contoh tabel hasil uji waktu yang dibutuhkan masing-masing *node* dalam memberikan informasi peringatan dini dan lokasi kebakaran hutan menuju stasiun pemantau.

Tabel 3.2 Uji waktu yang dibutuhkan sistem memberi peringatan dan lokasi kebakaran

ID *Node*	Waktu tercepat (Detik)	Waktu terlama (Detik)	Rata – rata (Detik)
Node 1			
Node 2			
Node 3			
Node 5			
Node 6			

3.7.3. Uji Kemampuan Transmisi Data Modul Radio SI4432

Proses pengujian modul radio dilakukan untuk mengetahui data yang tidak sampai menuju *server* serta untuk mengetahui kesesuaian antara data yang dikirim dari semua *node* dengan data yang diterima *server*. Pengujian pertama dilakukan dengan membandingkan jumlah data hasil pengujian sistem keseluruhan antara jumlah data pengukuran sensor yang tersimpan di dalam memori *micro*SD seluruh *node* dengan jumlah data hasil pengukuran sensor yang tersimpan pada *database* di komputer *server*. Pengujian kedua dilakukan dengan menghitung nilai data pengukuran sensor dari masing-masing *node* yang tersimpan didalam memori *card* dan nilai data pengukuran sensor dari masing-masing *node* yang tersimpan di *database*. Perhitungan untuk menganalisa data pengukuran sensor seluruh *node* dengan jumlah data yang diterima *server* menggunakan aplikasi *Microsoft Excel*. Pengujian ini dilakukan bertujuan untuk

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
© Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

mengetahui kemampuan modul radio SI4432 melakukan pengiriman data, apakah terjadi *lossed data* dan *error* data pada saat melakukan pengiriman data menuju *server* atau tidak.

3.7.4. Uji Kelayakan Sistem Menggunakan Kuisioner

Uji kelayakan tidak hanya dilakukan terhadap pengujian kemampuan sistem saja, tetapi uji kelayakan juga akan dilakukan ke pengguna dengan metode kuantitatif menggunakan kuesioner. Pengujian kelayakan ini dilakukan dengan mengumpulkan data dari sampel pada masyarakat awam. Populasi diambil dari masyarakat awam karena sistem peringatan dini kebakaran hutan ini diharapkan akan diimplementasikan langsung tidak hanya kepada instansi terkait namun juga dapat digunakan oleh masyarakat awam. Uji kelayakan sistem peringatan dini kebakaran hutan dengan cara menunjukkan dan menjelaskan perangkat *hardware* dan perangkat *software* yang telah dibuat, kemudian meminta kepada responden untuk menjawab pertanyaan dari daftar pertanyaan kuesioner, terdapat 9 pertanyaan yang diajukan kepada responden dapat dilihat pada Tabel 3.5. Hasil dari kuesioner ini nantinya diharapkan mampu mengetahui nilai parameter *usability*, *simplicity*, dan *interactivity* dari perancangan yang telah dibuat berdasarkan skor dari jawaban responden dengan skor tertinggi 5 dan skor terendah bernilai 1. Rata-rata skor penilaian responden dicatat pada tabel seperti pada Tabel 3.4. Untuk mengetahui skor pada setiap jawaban dari responden, masing-masing diukur dengan model Likert yaitu, mengukur sikap dengan menyatakan sangat setuju, setuju, kurang setuju, tidak setuju, sangat tidak setuju dengan skor tertinggi bernilai 5 dan skor terendah bernilai 1. Berikut adalah skor pernyataan sikap responden terhadap pertanyaan yang diajukan:

Sangat Setuju (SS) = 5
Setuju (S) = 4
Kurang Setuju (KS) = 3
Tidak Setuju (S) = 2
Sangat TIdak Setuju (SS) = 1 Penentuan skor pernyataan sikap responden untuk setiap pertanyaan pada masing-masing parameter menggunakan skala Likert seperti berikut:

$$\text{Skor Setiap Pernyataan sikap} = Jumlah\ Responder\ x\ Skor\ Pernyataan\ Sikap$$

$$\text{Jumlah Skor Responden} = Skor\ (SS) + Skor\ (S) + Skor\ (KS) + Skor\ (TS) + Skor\ (STS)$$

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
UIN SUSKA RIAU

Selanjutnya mencari rata-rata skor setiap jawaban responden. Berikut adalah rumus yang digunakan.

$$\text{Rata} - \text{Rata Skor} = \frac{Jumlah\ Skor\ Responder}{Jumlah\ Responder}$$

Sebagai contoh untuk pertanyaan dengan jawaban seperti pada Tabel 3.3 berikut.Tabel 3.3 Contoh tabel kuesioner untuk parameter *simplicity*

No.	Daftar Pertanyaan	SS	S	KS	TS	STS
1.	Proses mengoperasikan stasiun *transceiver* sistem peringatan dini kebakaran hutan mudah dilakukan	3	1	1	-	-

Sangat Setuju (SS) = 3 x 5 = 15

Setuju (S) = 1 x 4 = 4

Kurang Setuju (KS) = 1 x 3 = 3

Tidak Setuju (S) = 0 x 2 = 0

Sangat TIdak Setuju (SS) = 0 x 1= 0

Jumlah = 15 + 4 + 3 + 0 + 0 = 21

$$\text{Skor} = \frac{21}{5} = 4.2$$

Hasil contoh skor rata-rata 4.2, yang menandakan pernyataan sikap responden dari pertanyaan nomor 1 untuk parameter *simplicity* adalah setuju (S).

Tabel 3.4 Rata-rata skor penilaian responden

No. Pertanyaan	**Faktor (Rata-rata skor)**		
	Usability	**Simplicity**	**Interactivity**
1	-	-	-
2	-	-	-
3	-	-	-
4	-	-	-
5	-	-	-
Rata-rata			

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
Hak cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Berikut adalah daftar pertanyaan dari kuesioner "Rancang Bangun Sistem Peringatan Dini Kebakaran Hutan Menggunakan *Wireless Sensor Network* (WSN) Berbasis Arduino" seperti terlihat pada Tabel 3.5.

A. Identitas Responden

Nama Responden :

Alamat :

Pekerjaan :

B. Daftar Pertanyaan

Petunjuk pengisian:

Pilihlah jawaban yang dianggap paling sesuai menurut anda, dengan cara memberi tanda (✓) pada kolom jawaban yang telah tersedia. Penilaian dapat dilakukan berdasarkan skala berikut:

Jawaban "Sangat Tidak Setuju (STS)"

Jawaban "Tidak Setuju (TS)"

Jawaban "Kurang Setuju (KS)"

Jawaban "Setuju (S)"

Jawaban "Sangat Setuju (SS)"

Tabel 3.5 Daftar pertanyaan kuesioner

Simplicity						
No	Daftar Pertanyaan	STS	TS	KS	S	SS
1	Proses mengoperasikan stasiun *transceiver* sistem peringatan dini kebakaran hutan mudah dilakukan					
2	Aplikasi *interface* di stasiun pemantau mudah untuk digunakan					
3	Proses untuk mengoperasikan sistem peringatan dini kebakaran hutan mudah, dan tidak membutuhkan keahlian khusus.					

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
Hak Cipta milik UIN Suska Riau
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU

Tabel 3.6 Daftar pertanyaan kuesioner (lanjutan)

4	Tidak membutuhkan waktu lama untuk mengetahui adanya kebakaran hutan berdasarkan lokasi *node* yang mendeteksi adanya indikasi kebakaran.					
Interactivity						
5	Pengguna dapat mengetahui kondisi dilokasi berdasarkan parameter data pengukuran sensor melalui aplikasi *interface* secara *realtime*					
6	Jika terjadi kebakaran hutan, pengguna dapat mengetahui parameter yang mengindikasikan adanya kebakaran hutan beserta waktu terjadinya.					
7	Pengguna dapat mengetahui stasiun *transceiver* yang sedang aktif dan tidak aktif.					
8	Pengguna dapat mengambil data hasil pengukuran sensor untuk keperluan lebih lanjut melalui SDcard dan *database*.					
Usability						
9	Sistem peringatan dini kebakaran hutan sangat membantu untuk mencegah terjadinya kebakaran hutan yang lebih luas. Karena mampu memberikan peringatan dini dan lokasi secara cepat dan tepat ketika ada indikasi kebakaran hutan, berdasarkan titik koordinat pemasangan *node* sensor.					

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.
State Islamic University of Sultan Syarif Kasim Riau
UIN SUSKA RIAU