

## BAB II LANDASAN TEORI

### 2.1 *Crashbell*

Salah satu peralatan untuk menunjang keselamatan di sisi udara adalah sistem *crashbell*, yaitu sebuah peralatan yang digunakan untuk memberikan peringatan kepada petugas ARFF jika terjadi indikasi kecelakaan pesawat berupa *alarm* yg tombolnya di tekan dari *tower* ATC maupun *Watchroom* ARFF. Sistem *crashbell* ini terdiri dari 2 komponen utama yaitu *server* dan rangkaian *switching unit server* berfungsi sebagai pusat *database* dan tempat *web hosting* dari aplikasi *Human Machine Interface* (HMI) dari sistem *crashbell* yang akan mengirimkan sinyal *order* ke rangkaian *switching unit* [4].

Dalam rangkaian *Server* ini tersusun dari aplikasi XAMPP sebagai basis dasar dari komponen *server* sistem *crashbell*. Sedangkan pada rangkaian *switching unit* fungsi utamanya adalah memproses sinyal yg diterima dari *server* untuk di teruskan ke rangkaian akhir yaitu sebuah rangkain *bell* (lonceng) yang akan menghasilkan bunyi jika mendapat sinyal *trigger* bernilai “1” atau 5 Vdc. Dalam rangkaian *switching unit* ini tersusun oleh modul Raspberry PI dan modul Arduino. Setelah dilakukan proses pengujian dapat disimpulkan bahwa respon *time* rata-rata yang di perlukan sistem *crashbell* untuk mengirim sinyal saat tombol ditekan sampai lonceng bunyi adalah 1,35 detik [4].

Perancangan *crashbell* dari *tower* ke unit ARFF berbasis Arduino dan Raspberry PI bertujuan untuk memperoleh informasi tentang adanya bahaya dengan situasi ketika terjadi bahaya maka tombol panik akan ditekan oleh petugas di kantor kemudian mikrokontroler secara otomatis akan merespon dan menyalakan *alarmnya* dengan cara berbunyi. Pada perancangan ini memiliki beberapa bagian umum yang digunakan, yaitu *power supply*, *bell transmitter* dan *receiver* [4].

Pada gambar 2.1 merupakan proses *maintenance* pada *crashbell*. *Maintenance* ditujukan untuk membenahi susunan kabel serta mengecek apakah *crashbell* berfungsi dengan baik. Lalu pada gambar 2.2 merupakan tampilan *crashbell* dari luar dan pada gambar 2.3 merupakan tampilan *crashbell* dari dalam.



**Gambar 2. 1** Proses *maintenance* pada *crashbell*



**Gambar 2. 2** Tampilan *crashbell* dari luar



**Gambar 2. 3** Tampilan *crashbell* dari dalam

### **2.1.1 Mekanisme *Crashbell***

Berikut mekanisme pada *crashbell* :

- 1) Tombol *crashbell* digunakan dalam keadaan darurat ketika terjadi kecelakaan penerbangan.

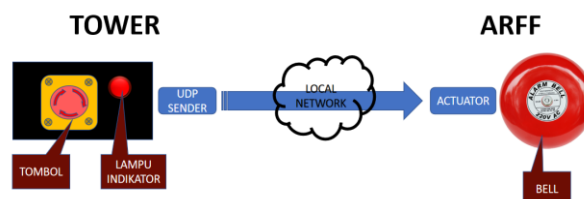
- 2) Tombol *crashbell* yang terpasang di *Tower* merupakan tipe *push lock button*. jadi dengan menekan sekali, tombol akan terkunci (posisi *on*), dan harus di *release* dengan memutar tombol searah jarum jam (arah panah) untuk posisi normal kembali.
- 3) Dalam pelaksanaan tes rutin harian, tombol *crashbell* dapat di tekan sampai lock, kemudian tunggu selama 5 detik (durasi *bell* di pk akan bunyi selama 5 detik) atau sesuai kesepakatan, baru di *release* kembali.
- 4) *Bell* di ARFF (*Airport Rescue & Fire Fighting*) akan terus bunyi, selama tombol *crashbell* belum di *release*.

### 2.1.2 Alur Kerja *Crashbell*

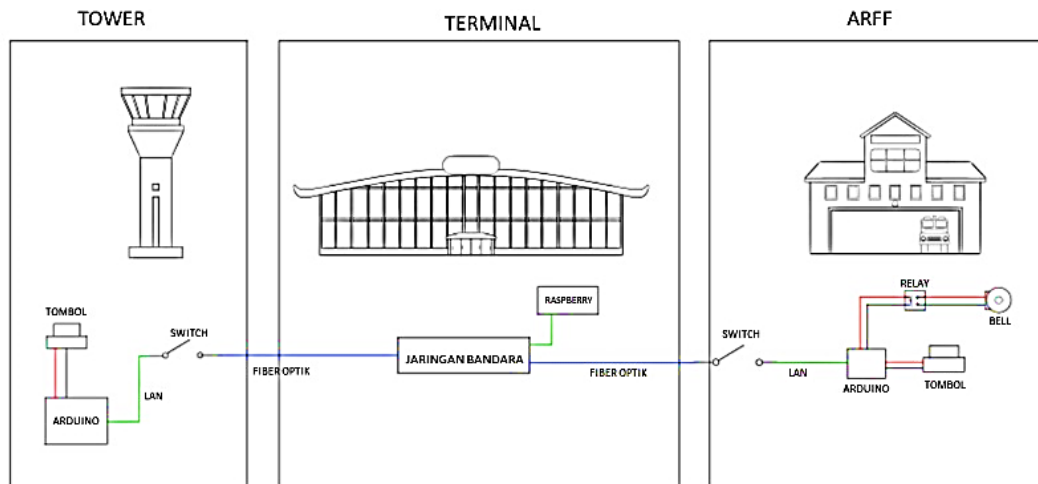
Berikut alur kerja *crashbell* :

- 1) Ketika tombol *crashbell* di tekan (lock), maka perangkat UDP *sender* akan mendeteksi tombol tertekan, menyalakan lampu indikator dan mengirim sinyal “aktifkan *bell*” ke perangkat *actuator* di gedung ARFF (*Airport Rescue & Fire Fighting*).
- 2) *Actuator* akan mengaktifkan *bell* selama tombol belum di *release*
- 3) *Release* (penormalan) tombol *crashbell* dengan memutar tombol searah jarum jam (arah panah). maka UDP *sender* akan mengirim sinyal “matikan *bell*” ke perangkat *actuator*.
- 4) *Actuator* akan me-nonaktifkan *bell*.

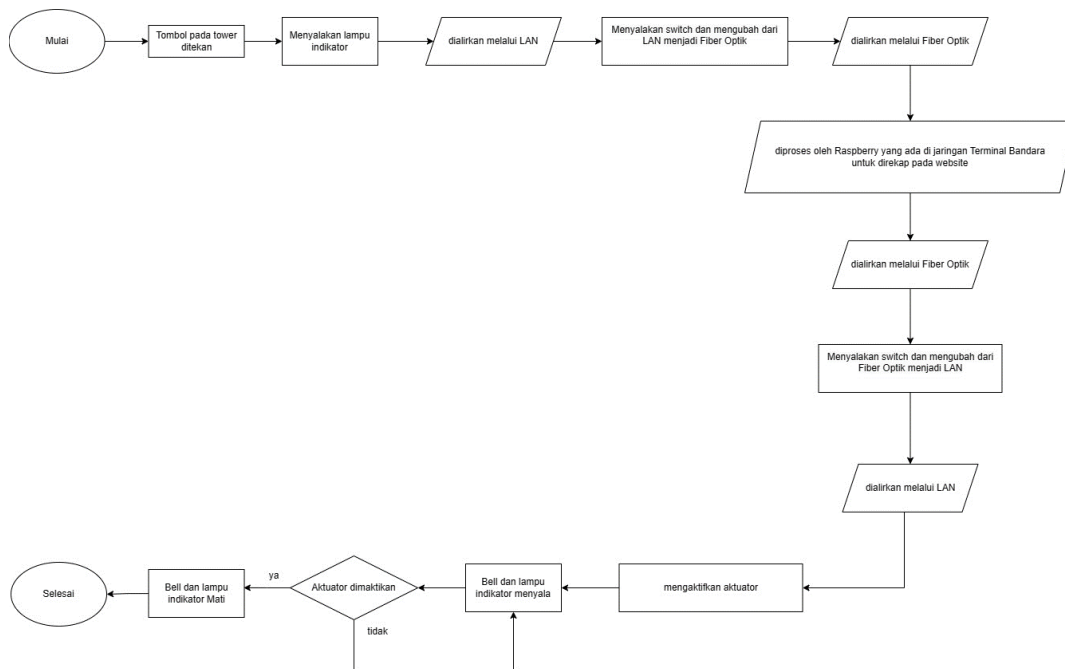
Pada gambar 2.4 merupakan alur kerja dari *crashbell*. Menmpilkan bahwa terdapat tombol dengan lampu indikator yang jika ditekan maka akan mengirim informasi melalui jaringan lokal dan menyalakan aktuator pada ARFF. Pada gambar 2.5 merupakan blok diagram *crashbell*. Merupakan tahapan dari *crashbell* dimana dimulai dari *tower* hingga ARFF. Pada gambar 2.6 menampilkan *flowchart* cara kerja *crashbell*.



**Gambar 2. 4 Alur kerja *crashbell***



Gambar 2. 5 Blok diagram *crashbell*

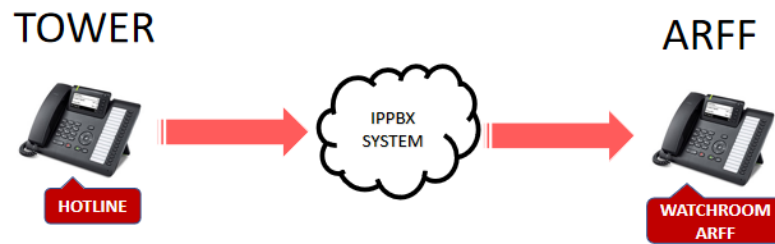


Gambar 2. 6 Flowchart cara kerja *crashbell*

## 2.2 Hotline TOWER - ARFF

*Hotline Tower - ARFF* merupakan pesawat telepon yang di tempatkan di *Tower* sebagai kontijensi keadaan darurat ketika *crashbell* tidak berfungsi. Hal ini sebagai opsi kedua dari penanggulangan terorisme/kecelakaan pesawat. Pesawat telepon ini hanya bisa digunakan untuk *direct call* (sambungan langsung) ke

*Watchroom*/gedung ARFF. Jaringan dipakai sama seperti pada *crashbell* dimana memanfaatkan LAN dan juga kabel fiber optik.



**Gambar 2. 7** Alur kerja *hotline* Tower-ARFF

Pada gambar 2.7 menampilkan alur kerja Alur Kerja *Hotline* Tower-ARFF. Ketika gagang pesawat telepon *hotline* diangkat, pesawat telepon secara otomatis akan melakukan panggilan ke nomor ekstensi gedung ARFF.

### 2.3 Arduino UNO

Arduino Uno adalah *board* mikrokontroler berbasis ATmega328. Memiliki 14 pin input dari *output* digital dimana 6 pin input tersebut dapat digunakan sebagai *output* PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP *header*, dan tombol *reset*. ATmega 328 adalah *chip* mikrokontroler 8-bit berbasis AVR-RISC buatan Atmel yang memiliki 32 KB memori ISP *Flash* dengan kemampuan baca-tulis (*read/write*), 1 KB EEPROM, 2 KB SRAM dan karena kapasitas memori *Flash* sebesar 32 KB inilah kemudian *chip* ini diberi nama ATmega328 [6].

Mikrokontroler adalah *chip* atau IC (*integrated circuit*) yang bisa diprogram menggunakan komputer. Tujuan memberikan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input dan kemudian menghasilkan *output* sesuai yang diinginkan. Mikrokontroler bertugas sebagai otak yang mengendalikan input, proses dan *output* sebuah rangkaian elektronik. Saat ini Arduino sangat populer di seluruh dunia [7].

### 2.4 Bell

*Bell* merupakan komponen peringatan bertipe *Audible* yang berupa perangkat yang akan memberikan peringatan berupa suara sirine, klakson, maupun seperti lonceng [8]. Fungsi dari *bell* adalah sebagai penanda pada petugas pemadam kebakaran yang berada di gedung ARFF (*Airport Rescue & Fire Fighting*) agar

cepat bersiaga dan cepat menanggapi jika ada hal yang bersangkutan dengan keamanan di Bandara. Pada gambar 2.8 dan 2.9 menampilkan *bell* yang terpasang pada gedung ARFF. *bell* ditempatkan pada tempat yang tinggi dan dekat dengan tempat berkumpulnya pada tegugas ARFF agar dapat terdengar baik sehingga respon terhadap ancaman dapat lebih maksimal.



**Gambar 2. 8 Bell terpasang didekat pintu masuk ARFF**



**Gambar 2. 9 Tampilan *bell* dari jauh**

## **2.5 Pengubah Tegangan**

Dalam mengontrol benda yang memiliki tegangan di atas tegangan operasi Arduino (biasanya 5V atau 3.3V) dapat menggunakan beberapa komponen, seperti *relay* atau aktuator. Pilihan antara *relay* dan aktuator bergantung pada jenis perangkat yang akan dikendalikan dan kebutuhan spesifik proyek.

### **2.5.1 Relay**

*Relay* adalah komponen elektronik yang berfungsi sebagai sakelar elektromagnetik yang dapat mengontrol perangkat berdaya tinggi dengan sinyal listrik berdaya rendah dari Arduino. Ketika Arduino mengirim sinyal ke *relay*

(biasanya menggunakan pin *output* digital), *relay* akan menarik elektromagnetnya, yang kemudian mengubah posisi kontaknya dari *off* (mati) menjadi *on* (hidup). Ini akan menghubungkan atau memutuskan sirkuit listrik perangkat yang lebih besar (seperti lampu atau motor) sesuai dengan kebutuhan [9].

### 2.5.1 Aktuator

Aktuator adalah perangkat mekanis atau elektrik yang mengubah energi listrik menjadi gerakan fisik atau tindakan tertentu. Aktuator dapat digunakan untuk mengontrol perangkat yang memerlukan gerakan fisik, seperti servo motor, motor DC, atau *stepper motor*. Arduino mengirim sinyal kontrol ke aktuator, dan aktuator menggerakkan komponen fisik sesuai dengan perintah tersebut. Misalnya, servo motor dapat digunakan untuk menggerakkan sudut sesuai dengan sinyal yang diberikan[10]. Pada gambar 2.10 menampilkan tampilan fisik *relay*. Pada gambar 2.11 menampilkan tampilan fisik aktuator.



**Gambar 2. 10 Relay**



**Gambar 2. 11 Aktuator**

## 2.6 Website Crashbell

Sistem *crashbell* adalah sebuah sistem yang telah diimplementasikan dan dipantau melalui sebuah *website* yang terintegrasi dengan Raspberry Pi. Raspberry Pi berfungsi sebagai perangkat yang memungkinkan akses ke Rednode, yang pada gilirannya menjadi bagian integral dari sistem *crashbell* [11].

Sistem *crashbell* sendiri terdiri dari dua komponen utama yang bekerja bersama-sama untuk menjalankan operasionalnya secara efektif:

### A. Server

Komponen *server* berperan sebagai pusat *database* yang mengumpulkan dan menyimpan data penting terkait dengan sistem *crashbell*. Selain itu, *Server* ini juga berfungsi sebagai host untuk aplikasi *Human Machine Interface* (HMI) dari sistem *crashbell*. Melalui HMI, pengguna dapat mengirimkan sinyal order yang diperlukan untuk mengaktifkan atau mematikan rangkaian *switching unit* [12].

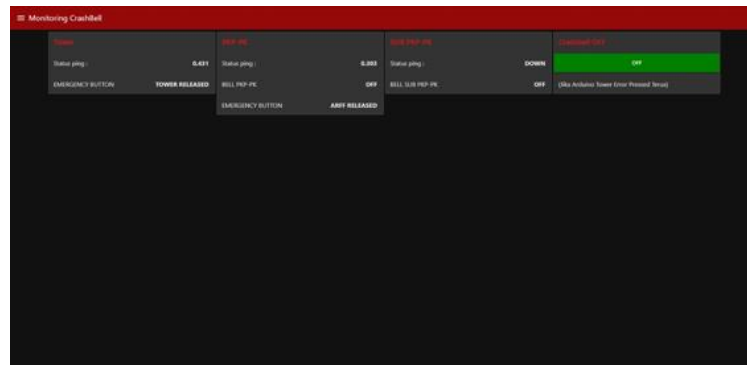
### B. Rangkaian Switching Unit

Rangkaian *switching unit* adalah komponen yang bertanggung jawab untuk menjalankan aksi fisik berdasarkan perintah yang diterima melalui HMI. Ketika HMI diakses melalui *website* terintegrasi, perintah akan diteruskan oleh *Server* ke rangkaian *switching unit*. Rangkaian ini kemudian akan mengendalikan proses fisik yang relevan, seperti mengaktifkan atau mematikan suatu perangkat atau sistem [13].

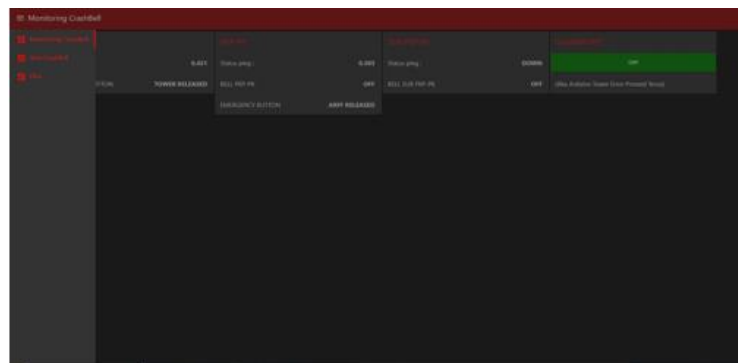
Dengan demikian, sistem *Crashbell* ini menciptakan sebuah ekosistem yang terintegrasi antara *server* sebagai pusat kendali dan pengelolaan data, serta rangkaian *switching unit* sebagai *executor* dari perintah yang diterima. Integrasi Raspberry Pi memungkinkan akses dan pemantauan sistem melalui *website*, menciptakan solusi yang efisien dan efektif dalam operasional *crashbell*. Pada gambar 2.12 menampilkan tampilan *website crashbell* yakni pada halaman utama dimana menampilkan kondisi dari *bell*. Pada gambar 2.13 menampilkan tampilan sub menu yang ada di *website crashbell*. Terdapat menu untuk monitoring kondisi *crashbell*, test *crashbell*, dan rekam jejak *crashbell*. Pada gambar 2.14 menampilkan catatan rekam jejak *crashbell* dimana menampilkan kapan saja *crashbell* menyala. Pada gambar 2.15 menampilkan menu untuk test *crashbell*



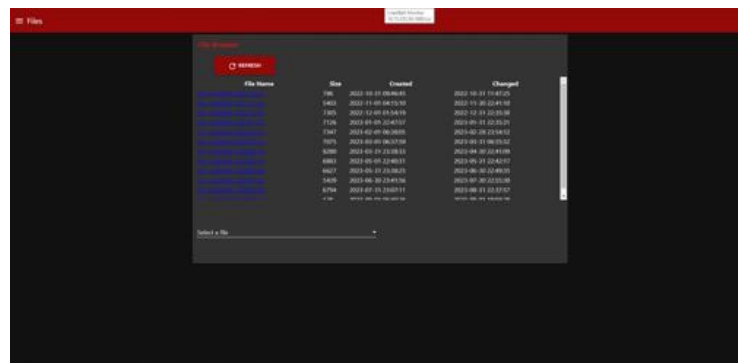
dimana saat tombol “ON” ditekan maka akan menyalakan *bell* yang ada di Gedung ARFF sedangkan dapat dimatikan dengan menekan tombol “OFF”.



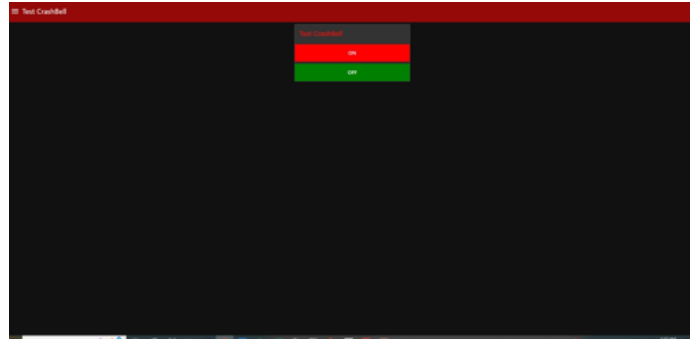
Gambar 2. 12 Tampilan menu utama *website crashbell*



Gambar 2. 13 Tampilan menu yang ada di *website crashbell*



Gambar 2. 14 Tampilan menu catatan *bell* dibunyikan



**Gambar 2. 15** Tampilan menu pengetesan *crashbell*

## 2.7 Raspberry Pi

Raspberry Pi, sering disingkat dengan nama Raspi, adalah komputer papan tunggal (*single-board circuit; SBC*) yang seukuran dengan kartu kredit yang dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, *Raspberry Pi Foundation*, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris[14].

Ide dibalik Raspberry Pi diawali dari keinginan untuk mencetak pemrogram generasi baru. Seperti disebutkan dalam situs resmi *Raspberry Pi Foundation*, waktu itu Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft, dari laboratorium komputer Universitas Cambridge memiliki kekhawatiran melihat kian turunnya keahlian dan jumlah siswa yang hendak belajar ilmu komputer. Mereka lantas mendirikan yayasan *Raspberry Pi* bersama dengan Pete Lomas dan David Braben pada 2009. Tiga tahun kemudian, *Raspberry Pi Model B* memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit. Pada bulan Februari 2016, *Raspberry Pi Foundation* mengumumkan bahwa mereka telah menjual 8 juta perangkat Raspi, sehingga menjadikannya sebagai perangkat paling laris di Inggris [15].