

# INTERAKSI ROBOT KRAKATAU FC DENGAN MULTICAST SECARA PARALEL (STUDI KASUS KONTES ROBOT SEPAKBOLA INDONESIA HUMANOID 2018)

Satria Darmawan  
Teknik Komputer

\*) satraia.damawan@gmail.com

## Abstrak

Kontes Robot Sepakbola Indonesia (KRSBI) Humanoid merupakan perlombaan robot yang diadakan secara rutin setiap tahun oleh Kemenristek Dikti. Dalam KRSBI, setiap robot diharuskan memiliki kemampuan mandiri untuk bereaksi terhadap lingkungan sekitar, seperti mendeteksi bola, menghampiri bola, berkomunikasi antar robot dan mengeksekusi bola hingga mencetak gol. Permasalahan yang terjadi pada robot sepakbola ini adalah perihal perebutan bola oleh tim yang sama yang tentunya akan mengakibatkan kerugian bagi tim sendiri dan keuntungan bagi tim lawan. Untuk mengatasi hal tersebut maka robot harus memiliki kemampuan untuk berkomunikasi satu sama lain agar tidak terjadi perebutan bola dalam tim. Teknik yang paling cocok untuk melakukan hal tersebut adalah IP Multicast. *IP Multicast* berpotensi untuk digunakan pada pendistribusian data masal semacam ini, sumber daya jaringan. *IP Multicast protocol* dapat menghemat *bandwith* karena data dikirimkan ke sebuah *multicast group*, tidak eksklusif untuk tiap *receiver* seperti halnya yang terjadi dalam pengiriman secara unicast, sehingga penggunaan sumber daya pada *node* pengirim dapat dikurangi secara signifikan.

**Kata Kunci:** kontes robot indonesia, IP multicast dan robot sepak bola.

---

## PENDAHULUAN

Kontes Robot Sepakbola Indonesia (KRSBI) Humanoid merupakan perlombaan robot yang diadakan secara rutin setiap tahun oleh Kemenristek Dikti (Putri et al., 2020). Dalam KRSBI, setiap robot diharuskan memiliki kemampuan mandiri untuk bereaksi terhadap lingkungan sekitar, seperti mendeteksi bola, menghampiri bola, berkomunikasi antar robot dan mengeksekusi bola hingga mencetak gol (Ramdan & Utami, 2020). Proses komunikasi ini hanya diizinkan melalui *Wifi* yang dianggap mewakili telinga dan mulut pada manusia (Jayadi et al., 2021). Untuk dapat berkomunikasi satu sama lain maka diperlukan sebuah *Access Point* sebagai media penghantar untuk komunikasi antar robot (Wijayanto et al., 2021). Untuk menghubungkan 2 atau lebih buah robot yang berbeda maka saya menerapkan suatu teknik pengiriman data secara masal agar data dapat dikirimkan ke banyak *node*, padahal data yang dikirimkan sama (Munandar & Amarudin, 2017).

*IP Multicast* berpotensi untuk digunakan pada pendistribusian data masal semacam ini karena dapat menghemat *bandwith* dan sumber daya jaringan (Fitri, Maulud, et al., 2021). *IP Multicast protocol* dapat menghemat *bandwith* (Fitri et al., 2020). *Node* pengirim dalam sebuah jaringan multicast hanya perlu melakukan satu kali pengiriman untuk seluruh *node* penerima yang tergabung dalam *multicast group* (Samsugi et al., 2020). Selain itu sifat *IP multicast* yang tidak reliabel ini dikarenakan *desain IP multicast* yang menggunakan UDP sebagai *transport protocol* (Yulianti et al., 2021). UDP juga tidak menjamin paket data akan sampai ke tujuan (Sulastio et al., 2021). Maka dari itu dibutuhkan pengiriman lebih

dari 1 informasi, yaitu identitas dan data, yang mana untuk memisahkan 2 buah informasi tersebut diperlukan sebuah *parsing* (Bahrudin et al., 2020). Namun proses komputer masih memiliki keterbatasan (*limitasi*) kemampuan dalam proses perhitungan dalam pengolahan data (Riski et al., 2021). Maka dari itu komunikasi tidak dapat dilakukan secara *serial*, dikarenakan untuk melakukan teknik *IP multicast* fungsi kirim dan terima pada robot haruslah dieksekusi secara bersamaan karena jika dilakukan secara *serial* maka yang terjadi adalah program *stuck* jika kedua kondisi robot dalam keadaan menerima data (Neneng et al., 2021).

Sehingga diperlukan sebuah cara yang secara efektif mampu memproses kedua fungsi tersebut secara bersamaan suatu metode yang disebut dengan teknologi *parallel computing* (Phelia, Pramita, Susanto, Widodo, & Tina, 2021). *Parallel computing* adalah suatu bentuk komputasi dengan membagi proses kalkulasi dan dilakukan secara simultan menggunakan beberapa komputer secara parallel (Asanovic, et al, 2006) (Wajiran et al., 2020). Metode ini biasanya digunakan pada proses perhitungan yang melibatkan data yang sangat banyak dan atau berukuran sangat besar (Iqbal et al., 2018). Agar komputer dapat bekerja secara simultan, komputer-komputer yang menyusun suatu sistem *parallel computing* harus terhubung dalam suatu jaringan agar komputer-komputer tersebut dapat saling berkomunikasi untuk menyelesaikan suatu tugas komputasi (Borman, Putra, et al., 2018). Hal ini dinilai efektif untuk mengatasi masalah tersebut (Silvia et al., 2016). Selain karena dapat melakukan eksekusi program secara cepat, metode ini juga dapat menghindari *stuck* pada komunikasi robot Krakatau FC (Pratama Zanofa & Fahrizal, 2021).

## **KAJIAN PUSTAKA**

### **Komunikasi**

Komunikasi merupakan proses pemindahan dan pertukaran pesan, dimana pesan ini dapat berbentuk fakta, gagasan, perasaan, data atau informasi dari seseorang kepada orang lain (Dita et al., 2021). Proses ini dilakukan dengan tujuan untuk mempengaruhi dan atau mengubah informasi yang dimiliki serta tingkah laku orang yang menerima pesan tersebut (Adhinata et al., 2021). Komunikasi pada dasarnya merupakan suatu proses yang menjelaskan siapa? mengatakan apa? dengan saluran apa? kepada siapa? dengan akibat atau hasil apa? (*who? says what? in which channel? to whom? with what effect?*) (Setiawan et al., 2021). Proses komunikasi adalah bagaimana komunikator menyampaikan pesan kepada komunikannya, sehingga dapat menciptakan suatu persamaan makna antara komunikan dengan komunikatornya (Amarudin & Ulum, 2018). Proses komunikasi ini bertujuan untuk menciptakan komunikasi yang efektif (sesuai dengan tujuan komunikasi pada umumnya) (Fitri, Rossi, et al., 2021). Proses komunikasi, banyak melalui perkembangan (Samsugi & Wajiran, 2020). Komunikasi adalah proses di mana individu mentransmisikan stimulus untuk mengubah perilaku individu yang lain (Arpiansah et al., 2021). Komunikasi yakni bentuk interaksi manusia yang saling memengaruhi satu sama lain secara sengaja dan tidak sengaja (Titin Yulianti, Selamat Samsugi, Prio Agung Nugroho, 2015). Komunikasi dapat menjadi alat untuk menghibur seseorang (Lestari et al., 2021). Misalnya penyampaian rasa simpati ketika seseorang bersedih, buku motivasi yang menghibur, acara televisi yang menyenangkan, juga musik dengan lirik penyemangat, semua merupakan bentuk komunikasi (Febrina & Megawaty, 2021). Tujuan berkomunikasi untuk mengubah sikap, pendapat, perilaku, dan sosial (Suaidah, 2021).

## Jaringan Komputer

Jaringan komputer adalah sebuah sistem yang terdiri atas komputer-komputer yang didesain untuk dapat berbagi sumber daya (*printer, CPU*), berkomunikasi (surel, pesan instan), dan dapat mengakses informasi bersama (Rossi et al., 2017). Jaringan komputer adalah Sebuah sistem di mana komputer yang terhubung untuk berbagi informasi dan sumber daya (Suwarni et al., 2021). Koneksi dapat dilakukan sebagai *peer-to-peer* atau *client / server*, biasanya hubungan antara komputer ini lebih cepat dari umumnya koneksi internet (Puspaningrum et al., 2020). Jaringan Komputer adalah Sebuah sistem jaringan komputer melibatkan dua buah komputer yang dihubungkan dengan menggunakan media online atau online telepon (Titin Yulianti, Selamat Samsugi, Prio Agung Nugroho, 2015). Sedangkan sistem jaringan yang rumit atau kompleks tergantung pada imajinasi masing-masing (Jupriyadi, 2018). Khususnya, jaringan komputer berarti semua *node* seperti *server*, stasiun kerja (*workstation*), *printer* dan sebagainya dihubungkan satu sama lain dengan tujuan untuk berbagi informasi dan bahan (Valentin et al., 2020).

## IP Multicast

*IP Multicast* adalah metode mengirimkan data UDP kepada sejumlah penerima yang tertarik menerima data-data ini (Utama & Putri, 2018). *IP Multicast* ini menggunakan teknik komunikasi *one-to-many* atau *many-to-many* (Novia Utami Putri et al., n.d.). Pengirim hanya mengirimkan suatu data sekali saja untuk diterima banyak penerima (Nurkholis & Susanto, 2020). Pengirim tidak perlu tahu siapa penerimanya maupun jumlah penerimanya (Priyambodo et al., 2020). Salah satu konsep penting dalam *IP Multicast* ini adalah *IP Multicast group address* (Fariyanto et al., 2021). Pemancar atau sumber atau penerima mengirimkan informasi ke *IP Multicast group address*, *IP Multicast group address* kemudian meneruskan ke penerima yang berminat menerima informasi ini (Kristiawan et al., 2021). Jadi *IP Multicast group address* ini merupakan penghubung antara pemancar dan penerima (Tiku Ali & Patombongi, 2016). *Protocol* yang digunakan untuk bergabung dengan *group* ini adalah *Internet Group Management Protocol (IGMP)* (Amarudin et al., 2020). *Protocol* lain terkait dengan *IP Multicast* ini adalah *Protocol Independent Multicast (PIM)* (Rossi & Rahni, 2016). *Multicast filtering* adalah proses yang memastikan bahwa penerima hanya menerima *Multicast traffic* hanya jika dia bergabung dalam suatu *Multicast group* (Rossi et al., 2018). Contoh yang menggunakan pendekatan *Multicast* ini antara lain pembelajaran jarak jauh, pengiriman serentak informasi saham, dan *video konferensi* (Fitri, Chen, et al., 2021).

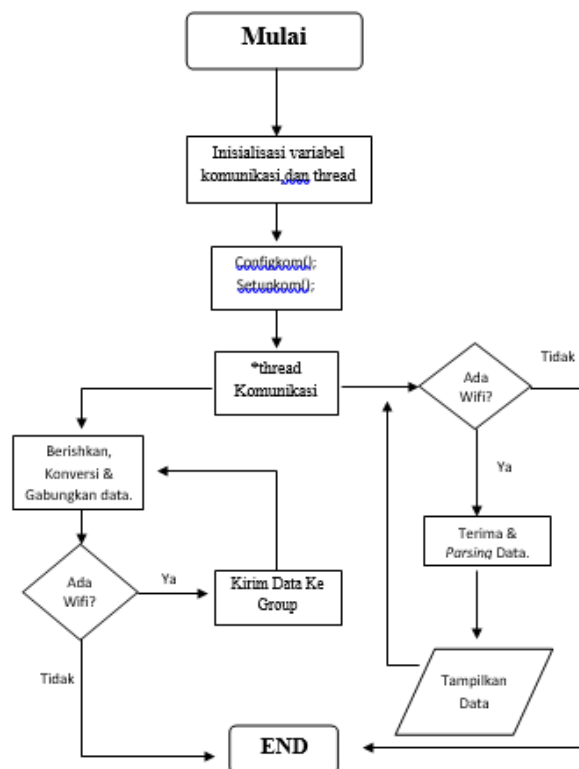
## Multithreading

*Multithreading* adalah kemampuan sebuah program untuk melakukan lebih dari satu pekerjaan sekaligus (Borman, Syahputra, et al., 2018). Keuntungan dari *Multithreading* adalah sifat respons yang interaktif dan *real-time* (Harahap et al., 2020). Arti istilah *Multithreading* dianggap berkaitan erat dengan pengertian *multithread* merupakan jalannya beberapa proses dengan urutan yang cepat (*multitasking*) dalam satu program (Phelia, Pramita, Susanto, Widodo, Aditomo, et al., 2021). *Multithreading* ini merupakan teknik pada manipulasi data dimana *node-node* pada struktur pohon data berisi penunjuk ke *node* yang lebih tinggi untuk membuat lintasan struktur menjadi lebih efisien (Pratama et al., 2021).

## METODE

### Perancangan Arsitektur IP Multicast Secara Paralel

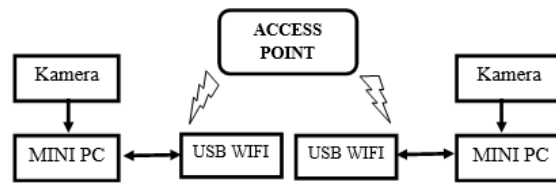
Arsitektur *IP Multicast* secara paralel pada sistem yang akan dibangun adalah arsitektur *IP Multicast* yang terdiri dari sebuah *thread* untuk melakukan pengiriman dan penerimaan data. Karena data yang dikirim masih berupa *string* yang berada dalam 1 variabel maka perlu dilakukan *parsing* dan konversi dari *string* ke integer agar data dapat diolah sebagai nilai numerik. Sebelum melakukan pengiriman atau penerimaan data, Sistem akan melakukan pengecekan koneksi terhadap *wifi*, jika *wifi* tersedia maka sistem akan mengirimkan informasi secara terus menerus, jika koneksi *WiFi* tidak tersedia maka program untuk berkomunikasi dihentikan dan sistem akan menjalankan robot secara offline. Secara umum alur proses Komunikasi dengan *IP Multicast* dapat dilihat pada gambar 1 dibawah ini.



Gambar 1

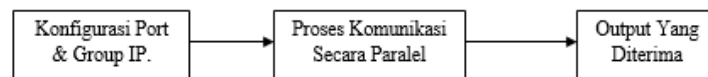
Gambar 1 menjelaskan bahwa *thread* komunikasi akan membuat sebuah *thread* yang mampu menjalankan proses pengiriman dan penerimaan data secara terpisah. Proses sendiri diawali dari proses *inisialisasi*, yang kemudian dilanjutkan dengan proses pembuatan *thread*. Setelah proses pembuatan *thread*, pada fungsi pengiriman variabel akan dikemas kedalam sebuah *array* yang akan dikirim nantinya. Setelah variabel dikemas selanjutnya adalah pengecekan koneksi *Wifi* jika tersedia maka sistem akan mengirimkan data ke *group* dan kembali lagi ke pengemasan data, jika tidak maka komunikasi akan terhenti. Sedangkan diproses penerimaan data pertama dilakukan pengecekan koneksi *WiFi*, jika tersedia maka sistem akan melakukan penerimaan data sekaligus pemisahan data dengan teknik *parsing*. Setelah proses pemisahan data selesai, langkah selanjutnya adalah data akan ditampilkan di *console*. Jika koneksi *Wifi* tidak tersedia maka komunikasi akan terhenti.

## Perancangan Arsitektur Sistem



Gambar 2

## Perancangan Sistem Secara Global



Gambar 3

## Perancangan Platform Robot

Robot sepak bola Krakatau dibangun dengan referensi desain DARWIN-OP. Robot memiliki 20 DOF, dimana 10 sendi tubuh bagian bawah digerakkan oleh servo MX-64T dan 10 sendi tubuh bagian atas dimotori oleh servo MX-28T. Dua sendi tubuh bagian atas digunakan sebagai basis peletakan kamera dengan fitur angguk (Tilt) dan geleng (Pan). Kondisi eksternal robot didapat melalui sensor *passive* berupa digital kamera PS Eye dan USB Wifi TP-Link N150 – TL-WN723N sebagai media komunikasi. Robot ini sendiri di otaki oleh komputer dengan *processor Intel Cherry Trail* yang bekerja pada clock 1.8 GHz..

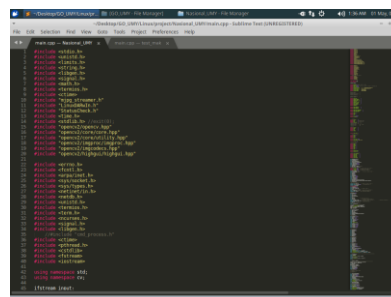


Gambar 4

## HASIL DAN PEMBAHASAN

### Tampilan Text Editor

*Text Editor* yang digunakan pada mini pc robot Krakatau FC adalah sublime text, dikarenakan warna yang *responsive* memudahkan *programmer* untuk pengkodean dan pencarian *error* program pada robot.



Gambar 5

Sublime Text sangat cocok digunakan untuk melakukan pengkodean pada bahasa pemrograman berbasis *console* karena fitur *auto complete* yang ada sangat memudahkan untuk memanggil variabel atau fungsi yang sudah kita buat.

### Program Bagian Pre Processor

```
22 #include <errno.h>
23 #include <fcntl.h>
24 #include <arpa/inet.h>
25 #include <sys/socket.h>
26 #include <sys/types.h>
27 #include <netinet/in.h>
28 #include <netdb.h>
29 #include <unistd.h>
30 #include <termios.h>
31 #include <term.h>
32 #include <ncurses.h>
33 #include <signal.h>
34 #include <libgen.h>
35 // #include "cmd_process.h"
36 #include <ctime>
37 #include <pthread.h>
38 #include <cstdlib>
39 #include <fstream>
40 #include <iostream>
```

Gambar 6

*Pre Processor* adalah bagian dimana program yang pertama kali dieksekusi oleh *compiler*, di bagian ini biasanya digunakan untuk menginclude kan library atau file header lainnya atau file program yang terpisah dari program utama yang berisikan fungsi-fungsi tertentu. Gambar diatas merupakan bagian *pre processor* dari program yang telah dibuat yang berisikan *library-library* yang dibutuhkan untuk melakukan pembuatan program komunikasi.

### Program Bagian Deklarasi Variabel

```
159 int new_message;
160 char recv_str[MAX_LEN+1]; /* buffer to receive string */
161 bool pernahkonek;
162 int robot=0;
163 int robotb;
164 int robotc=0;
165 int waktudie=0;
166 bool bekedaget=false;
167 bool sekali=false;
168 char arahbola[10];
169 char tiltsetku[10];
170 char pansetku[10];
171 int posisibola, posisibolb, posisibolc;
172 int tiltseta, tiltsetb, tiltsetc, panseta, pansetb, pansetc;
173 int langkahnya=0;
174 unsigned int send_len;
175 unsigned short mc_port=5000;
176 pthread_t krakataukom;
177 using namespace std;
178 int valuekom=50;
```

Gambar 7

```
179 char identitas[1024];
180 char send_str[MAX_LEN];
181 int sock;
182 struct sockaddr_in mc_addr;
183 char* mc_addr_str;
184 unsigned char mc_ttl;
185 string data[100];
186 int cekdetik=0;
187 int j;
188 int muterkompas;
189 int detectgoalcount;
190 int x;
191 int konekora;
192 int flag on;
193 int recv_len;
194 struct ip_mreq mc_req;
195 struct sockaddr_in from_addr;
196 unsigned int from len;
197 int waktu=0;
198 //akhir variabel komunikasi
```

Gambar 8

Gambar diatas adalah deklarasi variabel-variabel yang digunakan untuk pembuatan program komunikasi dengan teknik *IP Multicast*, dimana *port* yang digunakan adalah *port* 5000 pada variabel *mc\_port* . Variabel yang digunakan untuk melakukan pengiriman data adalah *send\_str[MAX\_LEN]* dan variabel yang digunakan untuk melakukan penerimaan data adalah *recv\_str[MAX\_LEN+1]*. Pada bagian ini juga pendeklarasian *thread* dibuat dengan nama *pthread\_t krakataukom*, dan variabel yang digunakan untuk penyimpanan *ip group* adalah *struct mc\_addr*.

## Program Bagian Konfigurasi

```

396 void configkom(){
397     flag_on=1;
398     mc_ttl=1;
399     mc_req.imr_multiaddr.s_addr = inet_addr("224.1.0.1");
400     mc_req.imr_interface.s_addr = htonl(INADDR_ANY);
401
402     memset(&mc_addr, 0, sizeof(mc_addr));
403     mc_addr.sin_family = AF_INET;
404     mc_addr.sin_addr.s_addr = inet_addr("224.1.0.1");
405     mc_addr.sin_port = htons(mc_port);
406 }

```

Gambar 9

Pada bagian ini *configkom* berfungsi untuk pengaturan *ip group* pada program dimana *ip group* yang digunakan adalah 224.1.0.1 sebagai konfigurasi dasar sebelum melakukan pengecekan koneksi terhadap *access point* yang tersedia, fungsi ini hanya dieksekusi sekali saja saat program dijalankan.

## Program Bagian Setup

```

407 void setupsock() {
408     //1)Nonmembership(sock, (struct sockaddr *)&mc_addr, sizeof(mc_addr)=0)
409     if ((sock = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) < 0) {
410         perror("socket() failed");
411     }
412     //2)
413     if ((setsockopt(sock, SOL_SOCKET, SO_REUSEADDR, &flag_on,
414         sizeof(flag_on)) < 0) ||
415         perror("setsockopt() failed");
416     }
417     //3)
418     if ((setsockopt(sock, IPPROTO_IP, IP_MULTICAST_TTL,
419         (void *)&mc_ttl, sizeof(mc_ttl)) < 0) ||
420         perror("setsockopt() failed");
421     }
422     //4)
423     if ((setsockopt(sock, IPPROTO_IP, IP_ADD_MEMBERSHIP,
424         (void *)&mc_req, sizeof(mc_req)) < 0) ||
425         perror("setsockopt() failed");
426     }
427     //5)
428     if ((bind(sock, (struct sockaddr *)&mc_addr,
429         sizeof(mc_addr)) < 0) ||
430         perror("bind() failed");
431     }
432 }
433 }
434 }

```

Gambar 10

Pada bagian ini akan dilakukan percobaan ke koneksi *wifi* untuk pengaturan *sock* serta *ip multicast group* yang telah dikonfigurasi sebelumnya, pada bagian *bind sock* dan *ip multicast group* akan diikat sehingga tidak akan keluar dari konfigurasi yang sudah ditetapkan sebelumnya yang membuat komunikasi hanya berjalan di *ip group* yang ditetapkan sebelumnya.

## Program Bagian Pemaketan Data

```

442 strcpy(send_str, "*B");
443 sprintf(identitas, "%d", diameter);
444 sprintf(arahbola, "%d", arah);
445 sprintf(pansetku, "%d", panset);
446 sprintf(tiltsetku, "%d", tiltset);
447 strcat(send_str, identitas);
448 strcat(send_str, " ");
449 strcat(send_str, arahbola);
450 strcat(send_str, " ");
451 strcat(send_str, tiltsetku);
452 strcat(send_str, " ");
453 strcat(send_str, pansetku);
454 strcat(send_str, "#");

```

Gambar 11

Pada bagian ini data yang akan dikirimkan ke dalam *group* akan dipaketkan terlebih dahulu kedalam 1 variabel array bernama *send\_str*, *strcpy* adalah fungsi yang digunakan untuk menyalin sebuah *string* kedalam variabel *send\_str*, pada gambar diatas dilakukan penyalinan sebuah *string* “\*B” yang merupakan identitas robot maka data sementara pada variabel *send\_str* adalah *send\_str[\*B]*. *Strcat* adalah sebuah fungsi yang digunakan untuk menggabungkan dua buah *string* menjadi 1, pada gambar diatas banyak dilakukan penggabungan *string* kedalam variabel *send\_str* yang akan membuat variabel *send\_str* berisikan nilai *send\_str[\*B, identitas, arahbola, tiltsetku, pansetku]* yang nantinya akan dipisahkan kembali pada bagian penerimaan data.

## Program Bagian Pengiriman Data

```

456 send_len = strlen(send_str);
457 if ((sendto(sock, send_str, send_len, 0,
458     (struct sockaddr *)&mc_addr,
459     sizeof(mc_addr)) != send_len) {
460     konekora--;
461 }else
462     new_message=0;
463 }

```

Gambar 12

Pada bagian ini dilakukan pengiriman data dengan fungsi *sendto()* sebelum melakukan pengiriman perlu dilakukan fungsi *strlen* pada variabel yang akan dikirim untuk mengetahui berapa jumlah karakter yang ada dalam variabel tersebut yang nantinya akan dikirim juga kedalam sebuah grup yang akan membantu untuk melakukan *parsing*.

### Program Bagian Penerimaan Data

```

465     if (konekora > 1) {
466         if ((setsockopt(sock, IPPROTO_IP, IP_MULTICAST_TTL,
467             (void *) &mc_ttl, sizeof(mc_ttl)) < 0) {
468             perror("setsockopt() failed");
469         }
470         from_len = sizeof(from_addr);
471         memset(&from_addr, 0, from_len);
472     }
473     if ((recv_len = recvfrom(sock, recv_str, MAX_LEN, 0,
474         (struct sockaddr *)&from_addr, &from_len) < 0) {
475         perror("recvfrom() failed");
476     }
477     else {
478         new_message=1;
479     }
480 }

```

Gambar 13

Pada bagian ini penerimaan data dilakukan, hal yang pertama dilakukan adalah melakukan pengecekan koneksi, jika ada koneksi terhadap *wifi* maka akan dilakukan *setsockopt* yang berfungsi untuk melakukan pengecekan pada *group* yang dikonfigurasi sebelumnya apakah masih *exist* atau tidak dengan memanfaatkan fitur *mc\_ttl* yang berarti *multicast time to live*. Untuk mengetahui jumlah karakter yang diterima maka memerlukan sebuah variabel *recv\_len* yang akan menampung nilai jumlah karakter dalam variabel *from\_len*. Fungsi yang digunakan untuk melakukan penerimaan data adalah *recvfrom()* yang mana *recvfrom* akan menampung data kedalam variabel *recv\_str*. Variabel *recv\_str* akan menampung nilai informasi yang dikirim dalam bentuk paket data yang artinya data-data yang dikirim masih belum dipisahkan dalam sebuah *array*. Jika digambarkan maka variabel *recv\_str* akan berisikan nilai *array* seperti *paketdata* yang dikirim sebelumnya yaitu *recv\_str[\*B,identitas,arahbola,tiltsetku,pansetku]* dimana *identitas* akan dimasukan di index 1, arah bola berada diindex ke 2, *tiltsetku* berada di index ke 3, dan *pansetku* berada di index ke 4.

### Program Bagian Parsing Data

```

483     data[j] = "";
484     data[j] = "";
485     for (i=0; i<strlen(recv_str); i++) {
486         if ((recv_str[i] == '#' || (recv_str[i] == ','))) {
487             data[j] = "";
488         }
489         else {
490             data[j] = data[j] + recv_str[i];
491         }
492     }
493     if (data[j] == "A") {
494         robot=atoi(data);
495         posisi=atoi(data);
496         tiltset=atoi(data);
497         panset=atoi(data);
498     }
499     if (data[j] == "B") {
500         robot=atoi(data);
501         posisi=atoi(data);
502         tiltset=atoi(data);
503         panset=atoi(data);
504     }
505     else if (data[j] == "C") {
506         robot=atoi(data);
507         posisi=atoi(data);
508         tiltset=atoi(data);
509         panset=atoi(data);
510     }
511     memset(recv_str, 0, sizeof(recv_str));
512 }
513 }

```

Gambar 14

Pada bagian ini dilakukan *parsing* data terhadap variabel *recv\_str* dimana akan dilakukan *iterasi* sebanyak jumlah karakter yang ada dalam variabel *recv\_str* yang mencari karakter string “#” sebagai penutup dan “,” sebagai pemisah data. Data dipisahkan dan ditampung kedalam variabel *string* baru bernama variabel *data* yang mana di tiap *elemen* mewakili 1 nilai data yang nanti akan dipisahkan kembali dan dikonversikan menjadi *integer* kedalam variabel lain seperti variabel *robot* jika identitas robot adalah “A” dan *robotb* jika identitas robot adalah “B” setelah data berhasil dipisahkan semua maka perlu dilakukan peresetan memori dengan fungsi *memset*.



## Program Bagian Multithreading

```
518 void *Threadkomunikasi(void*)
519 {
520     kirim();
521     terima();
522     usleep(250);
523 }
524
525 void komunikasi(){
526     pthread_create(&krakataukom, NULL, Threadkomunikasi, NULL);
527     pthread_join(krakataukom, NULL);
528 }
529
530 }
```

Gambar 15

Pada bagian ini dilakukan pembuatan *thread* komunikasi agar dapat melakukan eksekusi pengiriman dan penerimaan data secara terpisah, *usleep* digunakan untuk memberikan jeda pengiriman dan penerimaan selama 0.25 detik agar data yang dikirim tidak terlalu cepat dan membebani memori.

## Program Bagian While

```
2521 hitung++;
2522 printf("%i. Diameter Bola Robot A : %i Posisi Robot A :%i\n",hitung, robotA, posisiBola);
2523 cout << "Waktu Eksekusi : " << duration << " second" << endl;
2524
2525 if(konekoraWi){
2526     if ((sendto(sock, send_str, send_len, 0,
2527         (struct sockaddr *)&send_addr,
2528         sizeof(send_addr)) != send_len) {
2529         cout<<"Komunikasi Wi...Wi";
2530         robotA++;
2531         robotWi++;
2532     }
2533     else{
2534         setuptom();
2535     }
2536     while(1)
2537     {
2538         komunikasi();
2539     }
2540 }
```

Gambar 16

Pada bagian ini fungsi-fungsi yang telah dibuat akan dijalankan terus menerus selama robot berjalan dengan memperhatikan koneksi *wifi*, nilai dari hasil komunikasi akan ditampilkan dengan fungsi *printf*, nilai yang akan ditampilkan yaitu diameter bola dan posisi robot.

## Output Hasil Implementasi Program Komunikasi

```
%: 10. Diameter Bola Robot A : 258 Posisi Robot A :339
%: 11. Diameter Bola Robot A : 258 Posisi Robot A :339
%: 12. Diameter Bola Robot A : 258 Posisi Robot A :339
%: 13. Diameter Bola Robot A : 237 Posisi Robot A :340
%: 14. Diameter Bola Robot A : 237 Posisi Robot A :341
%: 15. Diameter Bola Robot A : 237 Posisi Robot A :341
%: 16. Diameter Bola Robot A : 237 Posisi Robot A :341
%: 17. Diameter Bola Robot A : 237 Posisi Robot A :341
%: 18. Diameter Bola Robot A : 237 Posisi Robot A :340
%: 19. Diameter Bola Robot A : 237 Posisi Robot A :337
%: 20. Diameter Bola Robot A : 237 Posisi Robot A :338
%: 21. Diameter Bola Robot A : 225 Posisi Robot A :338
%: 22. Diameter Bola Robot A : 225 Posisi Robot A :339
%: 23. Diameter Bola Robot A : 336 Posisi Robot A :339
%: 24. Diameter Bola Robot A : 336 Posisi Robot A :338
%: 25. Diameter Bola Robot A : 336 Posisi Robot A :344
%: 26. Diameter Bola Robot A : 336 Posisi Robot A :346
```

Gambar 17

Implementasi program komunikasi ini bisa dikatakan telah berhasil dan *output* yang ditampilkan sesuai dengan nilai sesungguhnya pada saat komunikasi berjalan pada robot.

## Hasil Pengujian IP Multicast Secara Paralel

```
%: 2. Diameter Bola Robot A : 0 Posisi Robot A :339
Waktu Eksekusi : 0.000377 second
%: 3. Diameter Bola Robot A : 0 Posisi Robot A :339
Waktu Eksekusi : 0.001109 second
%: 4. Diameter Bola Robot A : 0 Posisi Robot A :339
Waktu Eksekusi : 0.000325 second
%: 5. Diameter Bola Robot A : 0 Posisi Robot A :339
Waktu Eksekusi : 0.000264 second
%: 6. Diameter Bola Robot A : 129 Posisi Robot A :339
Waktu Eksekusi : 0.000366 second
%: 7. Diameter Bola Robot A : 129 Posisi Robot A :339
Waktu Eksekusi : 0.000315 second
%: 8. Diameter Bola Robot A : 129 Posisi Robot A :340
Waktu Eksekusi : 0.000559 second
%: 9. Diameter Bola Robot A : 129 Posisi Robot A :340
Waktu Eksekusi : 0.000326 second
%: 10. Diameter Bola Robot A : 129 Posisi Robot A :340
Waktu Eksekusi : 0.000345 second
```

Gambar 18

Tabel 1

Pengujian	Data Berhasil Diterima	Waktu Eksekusi
Data Ke 2	Berhasil	0.000377 detik
Data Ke 3	Berhasil	0.001109 detik
Data Ke 4	Berhasil	0.000325 detik
Data Ke 5	Berhasil	0.000264 detik
Data Ke 6	Berhasil	0.000366 detik
Data Ke 7	Berhasil	0.000315 detik
Data Ke 8	Berhasil	0.000559 detik
Data Ke 9	Berhasil	0.000326 detik
Data Ke 10	Berhasil	0.000345 detik
<b>Rata – Rata Waktu Eksekusi</b>		<b>0.00043844 detik</b>

Pada pengujian pertama didapat kesimpulan bahwa data ke-2 hingga ke-9 berhasil diterima seluruhnya dengan rata-rata waktu eksekusi 0.00043844 detik.

```

%: 51. Diameter Bola Robot A : 201 Posisi Robot A :339
Waktu Eksekusi : 0.000296 second
%: 52. Diameter Bola Robot A : 201 Posisi Robot A :339
Waktu Eksekusi : 0.000086 second
%: 53. Diameter Bola Robot A : 201 Posisi Robot A :339
Waktu Eksekusi : 0.000239 second
%: 54. Diameter Bola Robot A : 201 Posisi Robot A :339
Waktu Eksekusi : 0.000568 second
%: 55. Diameter Bola Robot A : 201 Posisi Robot A :340
Waktu Eksekusi : 0.000925 second
%: 56. Diameter Bola Robot A : 201 Posisi Robot A :339
Waktu Eksekusi : 0.000557 second
%: 57. Diameter Bola Robot A : 201 Posisi Robot A :339
Waktu Eksekusi : 0.000572 second
%: 58. Diameter Bola Robot A : 417 Posisi Robot A :339
Waktu Eksekusi : 0.000529 second
%: 59. Diameter Bola Robot A : 417 Posisi Robot A :340
Waktu Eksekusi : 0.000366 second

```

Gambar 19

Pengujian kedua dilakukan pada data ke 51 hingga 59 pada data yang diterima oleh robot b dari robot a.

Tabel 2

Pengujian	Data Berhasil Diterima	Waktu Eksekusi
Data Ke 51	Berhasil	0.000296 detik
Data Ke 52	Berhasil	0.000086 detik
Data Ke 53	Berhasil	0.000239 detik
Data Ke 54	Berhasil	0.000568 detik
Data Ke 55	Berhasil	0.000925 detik
Data Ke 56	Berhasil	0.000557 detik
Data Ke 57	Berhasil	0.000572 detik
Data Ke 58	Berhasil	0.000529 detik
Data Ke 59	Berhasil	0.000366 detik
<b>Rata – Rata Waktu Eksekusi</b>		<b>0.00045977 detik</b>

Pada pengujian kedua didapat kesimpulan bahwa data ke-51 hingga ke-59 berhasil diterima seluruhnya dengan rata-rata waktu eksekusi 0.00045977 detik.

```

s: 111. Diameter Bola Robot A : 390 Posisi Robot A :340
Waktu Eksekusi : 0.000298 second
s: 112. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.000289 second
s: 113. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.000259 second
s: 114. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.000042 second
s: 115. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.000506 second
s: 116. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.000577 second
s: 117. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.000423 second
s: 118. Diameter Bola Robot A : 390 Posisi Robot A :339
Waktu Eksekusi : 0.001097 second
s: 119. Diameter Bola Robot A : 390 Posisi Robot A :340
Waktu Eksekusi : 0.000272 second
    
```

Gambar 20

Pengujian ketiga dilakukan pada data ke 111 hingga 119 pada data yang diterima oleh robot b dari robot a.

Tabel 3

Pengujian	Data Berhasil Diterima	Waktu Eksekusi
Data Ke 111	Berhasil	0.000298 detik
Data Ke 112	Berhasil	0.000289 detik
Data Ke 113	Berhasil	0.000259 detik
Data Ke 114	Berhasil	0.000042 detik
Data Ke 115	Berhasil	0.000506 detik
Data Ke 116	Berhasil	0.000577 detik
Data Ke 117	Berhasil	0.000423 detik
Data Ke 118	Berhasil	0.001097 detik
Data Ke 119	Berhasil	0.000366 detik
<b>Rata – Rata Waktu Eksekusi</b>		<b>0.00042855 detik</b>

Pada pengujian ketiga didapat kesimpulan bahwa data ke-111 hingga ke-119 berhasil diterima seluruhnya dengan rata-rata waktu eksekusi 0.00042855 detik.

```

s: 1085. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.000431 second
s: 1086. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.001006 second
s: 1087. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.000343 second
s: 1088. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.00034 second
s: 1089. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.000241 second
s: 1090. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.00028 second
s: 1091. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.001016 second
s: 1092. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.000434 second
s: 1093. Diameter Bola Robot A : 450 Posisi Robot A :347
Waktu Eksekusi : 0.000401 second
    
```

Gambar 21

Pengujian keempat dilakukan pada data ke 1085 hingga 1093 pada data yang diterima oleh robot b dari robot a.

Tabel 4

Pengujian	Data Berhasil Diterima	Waktu Eksekusi
Data Ke 1085	Berhasil	0.000431 detik
Data Ke 1086	Berhasil	0.001006 detik
Data Ke 1087	Berhasil	0.000343 detik
Data Ke 1088	Berhasil	0.000034 detik
Data Ke 1089	Berhasil	0.000241 detik
Data Ke 1090	Berhasil	0.000028 detik
Data Ke 1091	Berhasil	0.001016 detik
Data Ke 1092	Berhasil	0.000434 detik
Data Ke 1093	Berhasil	0.000401 detik
<b>Rata – Rata Waktu Eksekusi</b>		<b>0.00043711 detik</b>

Pada pengujian keempat didapat kesimpulan bahwa data ke-1085 hingga ke-1093 berhasil diterima seluruhnya dengan rata-rata waktu eksekusi 0.00043711 detik.

## SIMPULAN

Berdasarkan hasil penelitian dan pembahasan maka dapat disimpulkan bahwa *IP Multicast* sangat efektif dan efisien untuk digunakan sebagai teknik komunikasi pada robot Krakatau FC karena *IP Multicast* hanya mengirimkan data 1 kali ke banyak penerima dan teknik *IP Multicast* yang dilakukan secara paralel mendapatkan hasil yang sangat optimal yaitu rata-rata waktu eksekusi tiap 1 kali melakukan pengiriman dan penerimaan data adalah sekitar 0.0004 detik ,dikarenakan proses pada pengiriman dan penerimaan data dieksekusi secara terpisah.

## REFERENSI

- Adhinata, F. D., Rakhmadani, D. P., Wibowo, M., & Jayadi, A. (2021). A Deep Learning Using DenseNet201 to Detect Masked or Non-masked Face. *JUITA: Jurnal Informatika*, 9(1), 115. <https://doi.org/10.30595/juita.v9i1.9624>
- Amarudin, A., Saputra, D. A., & Rubiyah, R. (2020). Rancang Bangun Alat Pemberi Pakan Ikan Menggunakan Mikrokontroler. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 1(1), 7–13.
- Amarudin, A., & Ulum, F. (2018). Desain Keamanan Jaringan Pada Mikrotik Router OS Menggunakan Metode Port Knocking. *Jurnal Teknoinfo*, 12(2), 72–75.
- Arpiansah, R., Fernando, Y., & Fakhrurozi, J. (2021). GAME EDUKASI VR PENGENALAN DAN PENCEGAHAN VIRUS COVID-19 MENGGUNAKAN METODE MDLC UNTUK ANAK USIA DINI. *Jurnal Teknologi Dan Sistem Informasi*, 2(2), 88–93.
- Bahrudin, A., Permata, P., & Jupriyadi, J. (2020). Optimasi Arsip Penyimpanan Dokumen Foto Menggunakan Algoritma Kompresi Deflate (Studi Kasus: Studio Muezzart). *Jurnal Ilmiah Infrastruktur Teknologi Informasi*, 1(2), 14–18.
- Borman, R. I., Putra, Y. P., Fernando, Y., Kurniawan, D. E., Prasetyawan, P., & Ahmad, I. (2018). Designing an Android-based Space Travel Application Trough Virtual Reality

- for Teaching Media. *2018 International Conference on Applied Engineering (ICAE)*, 1–5.
- Borman, R. I., Syahputra, K., Jupriyadi, J., & Prasetyawan, P. (2018). Implementasi Internet Of Things pada Aplikasi Monitoring Kereta Api dengan Geolocation Information System. *Seminar Nasional Teknik Elektro, 2018*, 322–327.
- Dita, P. E. S., Al Fahrezi, A., Prasetyawan, P., & Amarudin, A. (2021). Sistem Keamanan Pintu Menggunakan Sensor Sidik Jari Berbasis Mikrokontroler Arduino UNO R3. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 121–135.
- Fariyanto, F., Suaidah, S., & Ulum, F. (2021). PERANCANGAN APLIKASI PEMILIHAN KEPALA DESA DENGAN METODE UX DESIGN THINKING (STUDI KASUS: KAMPUNG KURIPAN). *Jurnal Teknologi Dan Sistem Informasi*, 2(2), 52–60.
- Febrina, C. A., & Megawaty, D. A. (2021). APLIKASI E-MARKETPLACE BAGI PENGUSAHA STAINLESS BERBASIS MOBILE DI WILAYAH BANDAR LAMPUNG. *Jurnal Teknologi Dan Sistem Informasi*, 2(1), 15–22.
- Fitri, A., Chen, H., Yao, L., Zheng, K., Susarman, Rossi, F., & Yin, Y. (2021). Evaluation of the Groundsill's stability at downstream of "Citorek" Bridge in Cimadur River, Banten Province. *IOP Conference Series: Earth and Environmental Science*, 880(1), 012029. <https://doi.org/10.1088/1755-1315/880/1/012029>
- Fitri, A., Maulud, K. N. A., Pratiwi, D., Phelia, A., Rossi, F., & Zuhairi, N. Z. (2020). Trend Of Water Quality Status In Kelantan River Downstream, Peninsular Malaysia. *Jurnal Rekayasa Sipil (JRS-Unand)*, 16(3), 178–184.
- Fitri, A., Maulud, K. N. A., Rossi, F., Dewantoro, F., Harsanto, P., & Zuhairi, N. Z. (2021). Spatial and Temporal Distribution of Dissolved Oxygen and Suspended Sediment in Kelantan River Basin. *4th International Conference on Sustainable Innovation 2020–Technology, Engineering and Agriculture (ICoSITEA 2020)*, 51–54.
- Fitri, A., Rossi, F., Suwarni, E., & Rosmalasari, D. (2021). *Pelatihan Pembuatan Video Pembelajaran Bagi Guru MA Matha ' ul Anwar Lampung Pada Masa Pandemi COVID-19*. 2(3), 189–196. <https://doi.org/10.23960/jpkmt.v2i3.50>
- Harahap, A., Sucipto, A., & Jupriyadi, J. (2020). Pemanfaatan Augmented Reality (Ar) Pada Media Pembelajaran Pengenalan Komponen Elektronika Berbasis Android. *Jurnal Ilmiah Infrastruktur Teknologi Informasi*, 1(1), 20–25.
- Iqbal, M., Gani, R. A., Ahdan, S., Bakri, M., & Wajiran, W. (2018). Analisis Kinerja Sistem Komputasi Grid Menggunakan Perangkat Lunak Globus Toolkit Dan MPICH-G2. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(2).
- Jayadi, A., Susanto, T., & Adhinata, F. D. (2021). Sistem Kendali Proporsional pada Robot Penghindar Halangan (Avoider) Pioneer P3-DX. *Majalah Ilmiah Teknologi Elektro*, 20(1), 47. <https://doi.org/10.24843/mite.2021.v20i01.p05>
- Jupriyadi, J. (2018). Implementasi Seleksi Fitur Menggunakan Algoritma Fvbrm Untuk Klasifikasi Serangan Pada Intrusion Detection System (Ids). *Prosiding Semnastek*.
- Kristiawan, N., Ghafaral, B., Borman, R. I., & Samsugi, S. (2021). Pemberi Pakan dan Minuman Otomatis Pada Ternak Ayam Menggunakan SMS. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 93–105.

- Lestari, F., Susanto, T., & Kastamto, K. (2021). PEMANENAN AIR HUJAN SEBAGAI PENYEDIAAN AIR BERSIH PADA ERA NEW NORMAL DI KELURAHAN SUSUNAN BARU. *SELAPARANG Jurnal Pengabdian Masyarakat Berkemajuan*, 4(2), 427–434.
- Munandar, G. A., & Amarudin, A. (2017). Rancang Bangun Aplikasi Kepegawaian Pegawai Negeri Sipil Dan Pegawai Honoror pada Badan Kepegawaian dan Diklat Kabupaten. *Jurnal Teknoinfo*, 11(2), 54–58.
- Neneng, N., Putri, N. U., & Susanto, E. R. (2021). Klasifikasi Jenis Kayu Menggunakan Support Vector Machine Berdasarkan Ciri Tekstur Local Binary Pattern. *CYBERNETICS*, 4(02), 93–100.
- Novia Utami Putri, V., Wiryono, W., & Gunggung, S. (n.d.). *KEANEKARAGAMAN JENIS TANAMAN, PEMANFAATAN DAN POTENSI CADANGAN KARBON PADA SISTEM AGROFORESTRI PEKARANGAN DUSUN II DESA HARAPAN MAKMUR KECAMATAN PONDOK KUBANG KABUPATEN BENGKULU TENGAH*. Fakultas Pertanian, UNIB.
- Nurkholis, A., & Susanto, T. (2020). Rancangan Media Pembelajaran Hewan Purbakala Menggunakan Augmented Reality. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(5), 978–987.
- Phelia, A., Pramita, G., Susanto, T., Widodo, A., Aditomo, R., Indonesia, U. T., Za, J., Alam, P., Ratu, L., Kedaton, K., & Lampung, K. B. (2021). *JURNAL CEMERLANG : Pengabdian pada Masyarakat PENINGKATAN PENGETAHUAN ANIMASI VIDEO DAN ROBOTIK DALAM PENERAPAN PROJECT BASE LEARNING DI SMA IT BAITUL JANNAH*. 4(1), 98–108.
- Phelia, A., Pramita, G., Susanto, T., Widodo, A., & Tina, A. (2021). *IMPLEMENTASI PROJECT BASE LEARNING DENGAN KONSEP ECO-GREEN DI*. 5, 670–675.
- Pratama, M. A., Sidhiq, A. F., Rahmanto, Y., & Surahman, A. (2021). Perancangan Sistem Kendali Alat Elektronik Rumah Tangga. *Jurnal Teknik Dan Sistem Komputer*, 2(1), 80–92.
- Pratama Zanofa, A., & Fahrizal, M. (2021). Penerapan Bluetooth Untuk Gerbang Otomatis. *Portaldata.Org*, 1(2), 1–10.
- Priyambodo, T. K., Dhewa, O. A., & Susanto, T. (2020). Model of Linear Quadratic Regulator (LQR) Control System in Waypoint Flight Mission of Flying Wing UAV. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 12(4), 43–49.
- Puspaningrum, A. S., Firdaus, F., Ahmad, I., & Anggono, H. (2020). Perancangan Alat Deteksi Kebocoran Gas Pada Perangkat Mobile Android Dengan Sensor Mq-2. *Jurnal Teknologi Dan Sistem Tertanam*, 1(1), 1–10.
- Putri, N. U., Oktarin, P., & Setiawan, R. (2020). Pengembangan Alat Ukur Batas Kapasitas Tas Sekolah Anak Berbasis Mikrokontroler. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 1(1), 14–22. <https://doi.org/10.33365/jimel.v1i1.189>
- Ramdan, S. D., & Utami, N. (2020). Pengembangan Koper Pintar Berbasis Arduino. *Journal ICTEE*, 1(1), 4–8. <https://doi.org/10.33365/jictee.v1i1.699>
- Riski, M., Alawiyah, A., Bakri, M., & Putri, N. U. (2021). Alat Penjaga Kestabilan Suhu Pada Tumbuhan Jamur Tiram Putih Menggunakan Arduino UNO R3. *Jurnal Teknik*

- Dan Sistem Komputer*, 2(1), 67–79.
- Rossi, F., Aizzuddin, A., & Rahni, A. (2018). *Joint Segmentation Methods of Tumor Delineation in PET – CT Images : A Review*. 7, 137–145.
- Rossi, F., Mokri, S. S., & Abd. Rahni, A. A. (2017). Development of a semi-automated combined PET and CT lung lesion segmentation framework. *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*, 10137, 101370B. <https://doi.org/10.1117/12.2256808>
- Rossi, F., & Rahni, A. A. A. (2016). Combination of low level processing and active contour techniques for semi-automated volumetric lung lesion segmentation from thoracic CT images. *ISSBES 2015 - IEEE Student Symposium in Biomedical Engineering and Sciences: By the Student for the Student*, 26–30. <https://doi.org/10.1109/ISSBES.2015.7435887>
- Samsugi, S., & Wajiran, W. (2020). IOT: Emergency Button Sebagai Pengaman Untuk Menghindari Perampasan Sepeda Motor. *Jurnal Teknoinfo*, 14(2), 99–105.
- Samsugi, S., Yusuf, A. I., & Trisnawati, F. (2020). Sistem Pengaman Pintu Otomatis Dengan Mikrokontroler Arduino Dan Module Rf Remote. *Jurnal Ilmiah Mahasiswa Kendali Dan Listrik*, 1(1), 1–6.
- Setiawan, M. B., Susanto, T., & Jayadi, A. (2021). PENERAPAN SISTEM KENDALI PID PESAWAT TERBANG TANPA AWAK UNTUK KESETABILAN ROLL, PITCH DAN YAW PADA FIXED WINGS. *The 1st International Conference on Advanced Information Technology and Communication (IC-AITC)*.
- Silvia, A. F., Haritman, E., & Muladi, Y. (2016). Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino Dan Android. *Electrans*, 13(1), 1–10.
- Suaidah, S. (2021). Teknologi Pengendali Perangkat Elektronik Menggunakan Sensor Suara. *Jurnal Teknologi Dan Sistem Tertanam*, 02(02). <https://ejurnal.teknokrat.ac.id/index.php/jtst/article/view/1341>
- Sulastio, B. S., Anggono, H., & Putra, A. D. (2021). SISTEM INFORMASI GEOGRAFIS UNTUK MENENTUKAN LOKASI RAWAN MACET DI JAM KERJA PADA KOTA BANDARLAMPUNG PADA BERBASIS ANDROID. *Jurnal Teknologi Dan Sistem Informasi*, 2(1), 104–111.
- Suwarni, E., Rosmalasar, T. D., Fitri, A., & Rossi, F. (2021). Sosialisasi Kewirausahaan Untuk Meningkatkan Minat dan Motivasi Siswa Mathla'ul Anwar. *Jurnal Pengabdian Masyarakat Indonesia*, 1(4), 157–163. <https://doi.org/10.52436/1.jpmi.28>
- Tiku Ali, J., & Patombongi, A. (2016). Perancangan Game Edukasi Pembelajaran Membaca Berbasis Android. *Simtek : Jurnal Sistem Informasi Dan Teknik Komputer*, 1(1), 1–8. <https://doi.org/10.51876/simtek.v1i1.1>
- Titin Yulianti, Selamat Samsugi, Prio Agung Nugroho, H. A. (2015). Rancang Bangun Alat Pengusir Hama Babi Menggunakan Mikrokontroler Arduino Dengan Sensor Gerak. *Jtst*, 3(4), 21–27.
- Utama, S., & Putri, N. U. (2018). Implementasi Sensor Light Dependent Resistor (LDR) Dan LM35 Pada Prototipe Atap Otomatis Berbasis Arduino. *CIRCUIT: Jurnal Ilmiah Pendidikan Teknik Elektro*, 2(2).

- Valentin, R. D., Diwangkara, B., Jupriyadi, J., & Riskiono, S. D. (2020). Alat Uji Kadar Air Pada Buah Kakao Kering Berbasis Mikrokontroler Arduino. *Jurnal Teknik Dan Sistem Komputer*, 1(1), 28–33.
- Wajiran, W., Riskiono, S. D., Prasetyawan, P., & Iqbal, M. (2020). Desain Iot Untuk Smart Kumbang Thinkspeak Dan Nodemcu. *POSITIF: Jurnal Sistem Dan Teknologi Informasi*, 6(2), 97–103.
- Wijayanto, D., Firdonsyah, A., Adhinata, F. D., & Jayadi, A. (2021). Rancang Bangun Private Server Menggunakan Platform Proxmox dengan Studi Kasus: PT.MKNT. *Journal ICTEE*, 2(2), 41. <https://doi.org/10.33365/jictee.v2i2.1333>
- Yulianti, T., Samsugi, S., Nugroho, P. A., & Anggono, H. (2021). Rancang Bangun Pengusir Hama Babi Menggunakan Arduino dengan Sensor Gerak. *JTST*, 2(1), 21–27.