

## **BAB III ANALISIS**

Dalam bab ini akan dibahas tentang tata cara mengendarai mobil yang sebenarnya, dan tata cara mengontrol mobil R/C secara manual. Selain itu dalam bab ini membahas pula tentang kendala yang ditimbulkan akibat mobil R/C dikendalikan oleh komputer dan pemecahannya. Dalam kendaraan yang sebenarnya memiliki beberapa kontrol yang harus dikuasai agar kendaraan bisa melaju lurus, berbelok, mundur ataupun berhenti. Begitu juga pada mobil R/C, untuk lebih jelasnya kita lihat sebagai berikut :

### **3.1 Mengendarai Kendaraan Yang Sebenarnya**

Pada kendaraan yang sesungguhnya terdapat beberapa fungsi kontrol yang harus diperhatikan dengan baik dan tata cara aturan menggunakan kendaraan tersebut. Fungsi kontrol itu terdiri dari gas, rem kaki, kopling, gear, rem tangan, semua ini digunakan untuk menjalankan kendaraan roda empat. Dalam mengendarai kendaraan tersebut kita harus mengontrol fungsi-fungsi yang telah disebutkan diatas.

Pada saat kita akan menjalankan kendaraan tentu saja kita harus menjalankan mesin terlebih dahulu, setelah mesin menyala maka kita tinggal menggerakkan mobil tersebut. Tapi ada aturan yang harus dilakukan oleh kita harus merubah gear normal ke gear satu, tapi sebelum itu kita harus menginjak kopling terlebih dahulu. Setelah gear pada posisi satu kopling tersebut di lepas perlahan dan kita langsung menginjak gas sedikit demi sedikit sesuai kecepatan yang kita inginkan. Ada beberapa yang penulis lihat bahwa selain kita pertama berangkat dari posisi berhenti ke posisi melaju dengan posisi awal gear satu, tapi ada juga yang menggunakan posisi gear dua.

Kemudian pada saat kita sedang melaju terjadinya sesuatu atau terdapat halangan di depan kita, maka kendaraan dimungkinkan harus berhenti ada

beberapa aturan yang penulis ketahui pertama pada saat jarak tertentu terhadap halangan atau rintangan yang ada kita menginjak terlebih dahulu kopling kemudian kendaraan dari jalan yang cukup cepat menjadi pelan akibat pengaruh kopling tadi kemudian kita menginjak rem kaki tentunya agar mobil bisa berhenti tapi pada saat kita menginjak rem kopling pun masih pada posisi terinjak. Yang kedua kita tinggal menginjak rem kaki agar mobil bisa berhenti.

Begitulah kondisi pengontrolan ketiga fungsi tersebut akan terus seperti itu, untuk fungsi rem tangan jika kita mengendarai pada tempat yang menanjak agar supaya tidak terjadinya melaju mundur ke belakang pada saat berhenti.

Ada satu lagi yang sebagai fungsi kontrol dinama dia bisa membelokkan mobil dan lurus yaitu stirl kita bisa belok kiri atau kanan atau menginginkan lurus kita tinggal atur stirlnya saja. Jika kita menginginkan belok kanan maka stirl kita putar ke kanan sesuai belokan jalan.

### **3.2 Mengontrol mobil R/C ( Remote Control )**

Pada bagian ini kita akan melihat bagaimana mobil R/C ini bisa bergerak lurus, belok kanan, belok kiri dan mundur. Memang pada mobil yang sesungguhnya terdapat fungsi-fungsi yang harus kita kendalikan, tapi di sini fungsi yang kita kendalikan tidak terlalu rumit dan cukup banyak.

Kalau kita menginginkan mobil R/C ini bergerak lurus, maka kita tinggal menekan tombol kontrol yang berfungsi untuk maju, berbeda dengan kendaraan yang telah kita singgung di atas, pada saat waktu tertentu kita menginginkan untuk berbelok maka tombol maju tetap ditekan dan tombol untuk berbelok di tekan juga. Tombol berbelok ada dua yaitu tombol kiri dan kanan, kalau kita tekan tombol kanan serta tombol maju maka mobil R/C kita akan bergerak sambil berbelok ke kanan, begitu juga sebaliknya. Jika kita juga menginginkan untuk mundur pada mobil R/C kita maka kita tinggal menekan tombol mundur saja.

Sensor untuk melihat adanya rintangan masih secara manual yaitu oleh kita sendiri, jika didepan mobil R./C kita terdapat rintangan maka kita harus berusaha menghindari rintangan tersebut. Ini memang ada kemiripan dengan

mengendarai kendaraan yang sesungguhnya hanya saja fungsi kontrolnya yang berbeda.

### 3.3 Komunikasi Serial Pada Program

Sebelum kita menjelaskan bagaimana komputer mengirim sebuah pesan melalui serial port serta dan kemudian diterima oleh *microcontrollers* dan diperiksa pesan yang dikirimnya. Serial port yang akan digunakan untuk berkomunikasi antara komputer dengan *microcontrollers* adalah serial port yang berkaki 9 atau juga disebut DB9, kaki serial port ini yang digunakan adalah kaki pengiriman data serta kaki yang menuju ke ground. Tabel 3.1 memperlihatkan kesembilan kaki serial port tersebut.

**Tabel 3.1 : Kaki DB9**

Pin	Sinyal
1	Data Carrier Detect
2	Received Data
3	Transmitted Data
4	Data Terminal Ready
5	Sinyal Ground
6	Data Set Ready
7	Request to Send
8	Clear to Send
9	Ring Indikator

Pada tabel diatas bahwa kaki yang digunakan adalah kaki yang mengirim data serta kaki sinyal ground, kaki tersebut adalah kaki atau pin 3 dan pin 5. Tapi sebelum pengiriman pesan atau karakter oleh program, kita harus inisialisasikan *bound rate* serial portnya kemudian port berapa yang harus digunakan. Sebagai contoh *bound rate* 9600 dan port 2 yang digunakan, dan selain ini adapula yang harus disesuaikan apakah serial port digunakan untuk mengirim atau menerima dalam program.

Pada saat mobil yang dibuat dalam program digerakan, secara hampir bersamaan kita kirim sebuah pesan karakter untuk menjalankan mobil R/C tersebut, Yang pasti inisial diatas sudah ditentukan. Karakter yang dikirim mengandung sebuah perintah untuk menjalankan, membelokkan mobil R/C, sebagai contoh karakter “a” digunakan untuk menjalankan mobil R/C, kemudian karakter “ab” digunakan untuk menjalankan serta membelokan ke kanan. Pada tabel 3.2 diperlihatkan keseluruhan karakter yang mengandung sebuah perintah. Dengan kaki atau pin 3 pada serial port karakter yang ada pada tabel 3.2 tersebut bisa dikirim.

**Tabel 3.2 : Perintah karakter**

Karakter	Perintah
a	Mobil R/C berjalan
ab	berjalan dan berbelok ke kanan
ac	berjalan dan berbelok ke kiri
d	Berhenti

### 3.4 Komunikasi serial Pada Kontroller

Setelah karakter dikirim oleh program yang ada pada komputer melalui serial port, maka karakter tersebut yang dikirim diterima oleh kontroller dengan serial port juga. Penerimaan karakter tersebut melalui kaki atau pin 2 yang menerima karakter serta pin 3 yang sebagai ground.

Sebelum karater tersebut diterima oleh kontroller, maka sinyal tersebut harus dirubah terlebih dahulu menjadi sinyal TTL. Dikarenakan *microcontrollers* hanya bisa membaca sinyal TTL. Setelah dirubah karater tersebut diterima oleh salah satu kaki kontroller yang khusus untuk menangani serial port.

Kaki tersebut bernama pin RXD atau P3.0 ( port 3.0 ). Kemudian karakter tersebut diperiksa apakah karakter tersebut sebagai perintah untuk menjalankan mobil R/C atau memberhentikan mobil R/C, sebagai contoh “a” perintah untuk menjalankan mobil R/C kemudian kontroller dengan menggunakan biner untuk mengfungsikan button jalan. Output pin yang digunakan untuk mengfungsikan

button tersebut P1.0 sampai P1.3, adapun perintah selengkapnya pada tabel 3.3 sebagai berikut :

**Tabel 3.3** : Pengambilan perintah pada *microcontrollers*

Karater	Perintah biner	Keterangan
A	0000 0001	Maju
ab	0000 0101	Maju belok kanan
ac	0000 0011	Maju belok kiri
d	0000 0000	Berhenti

### 3.5 Kendala Yang Timbul Pada Otomatisasi

Setelah kita tahu bagaimana tata cara mengendarai kendaraan yang sebenarnya dan mobil R/C tersebut, serta mengetahui perbedaannya kedua tata cara tersebut. Maka kita tahu bagian mana saja yang harus dirubah untuk bisa dikendalikan oleh komputer. Fungsi-fungsi yang harus dirubah adalah fungsi tombol yang digunakan untuk mengendalikan mobil R/C secara manual.

Adapun kendala atau permasalahan yang ditimbulkan pada saat komputer mengendalikan mobil R/C tersebut adalah sebagai berikut :

1. Kesulitan komputer untuk mengendalikan fungsi tombol yang ada di remote control.
2. Kesulitan komputer untuk mengetahui arah mana yang harus dituju agar tidak terjadinya tabrakan pada mobil R/C.
3. Pada mobil R/C ini untuk sebagai rintangan yang ada adalah sebuah benda-benda yang ada pada suatu ruangan.

### 3.6 Pemecahan Masalah

Dengan kondisi permasalahan atau kendala yang ada maka pengendalian mobil R/C oleh komputer dibagi dua, agar bisa mengatasi kendala tersebut. Pengendalian yang dibagi dua tersebut adalah pengendalian hardware dan pengendali software, pengendalian hardware memiliki beberapa pekerjaan atau fungsi yang harus dilakukan, yang terdiri dari :

1. Kendali tombol, dimana kendali tombol ini menggunakan sebuah relay yang sebagai kendali agar mobil R/C bisa bergerak.
2. Kendali relay, agar supaya relay bisa dikendalikan maka harus adanya sebuah chip yang menerima data dari serial port dan kemudian menjalankan relay yang sesuai dari data yang diterima oleh chip.

Pengendalian software dimana bagian ini untuk mengontrol pada pengendali hardware, bagian software ini terdiri dari :

1. Peta, dimana peta ini berfungsi untuk menentukan jalur yang akan dituju.
2. Komunikasi yang digunakan adalah komunikasi serial agar pengendali hardware ini bisa sesuai dengan software.
3. Pengendali transmit yang digunakan agar terjadinya komunikasi antara pengendali hardware dan pengendali software menggunakan RS232 yang mengendalikan pengiriman maupun penerimaan data.

### **3.5 Hasil Analisis**

Dalam memindahkan kontrol manual ke kontrol otomatis dibutuhkan suatu pemahaman dalam segi mengendarai atau mengontrolnya, tentunya dibutuhkan kepekaan pada analisa agar fungsi kontrol tersebut bisa menjadi otomatis. Dalam analisa tersebut diungkapkan tata cara mengendarai baik pada kendaraan yang sebenarnya maupun pada mobil R/C, begitu juga permasalahan yang ditimbulkan pada proses pembuatan serta pemecahannya.

Permasalahan dalam yang ditimbulkan akibat pemindahan kontrol manual ke kontrol otomatis ada tiga kendala yang mengacu pada pengendalian mobil R/C secara manual ada sebagai berikut : pertama kesulitan komputer untuk mengendalikan fungsi tombol yang ada di remote control, kedua kesulitan komputer untuk mengetahui arah mana yang harus dituju agar mobil R/C bisa mencapai tujuan atau posisi yang telah ditentukan, dan yang ketiga kesulitan komputer menghadapi rintangan yang ada pada benda-benda di suatu ruangan tertentu agar tidak terjadinya tabrakan pada mobil R/C. Tentu saja, tiga permasalahan ini dipecahkan dengan dua pemecahan, yang terdiri dari pengendalian software dan pengendalian hardware yang masing-masing memiliki

beberapa pekerjaan atau fungsi yang terpisah tapi dalam satu tujuan. Pekerjaan atau fungsi yang dimiliki masing-masing sudah diungkapkan diatas.

Pada intinya analisa yang telah dibuat serta aplikasi yang akan dibangun ini, hanya untuk pemindahan remote control yang manual menjadi kontrol otomatis. Dan aplikasi yang digunakan oleh komputer menggunakan peta pada suatu ruangan tertentu untuk mengendalikan mobil R/C.

## **BAB IV**

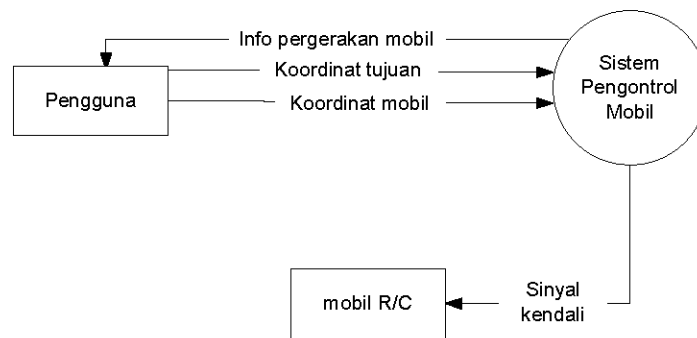
### **PERANCANGAN**

Pada tahapan perancangan ini dijelaskan secara mendetail mengenai perancangan perangkat lunak, dengan tool-tool yang terkait didalamnya. Tool-tool tersebut membantu kita dalam rangka merancang perangkat lunak yang akan dibangun. Tahapan ini merupakan tahap lanjutan dari tahapan analisis, adapun tool yang akan digunakan sebagai berikut ini :

1. Data Flow Diagram (DFD), pembahasan cara kerjanya sistem yang dimulai dari proses besar atau umum sampai kepada peroses yang tekecil.
2. Penjelasan proses, pembahasan mengenai cara kerja proses pada DFD level terakhir.
3. Hubungan fisik, pembahasan tentang bagaimana hubungan antara komputer dengan *microcontrollers*.
4. Schematic elektronik, pembahasan mengenai jalur untuk rangkaian elektronika.
5. Perancangan tampilan, pembahasan ini mengenai tampilan program yang akan dibuat tool-tool apa saja yang digunakan.

#### **4.1 Data Konteks Diagram**

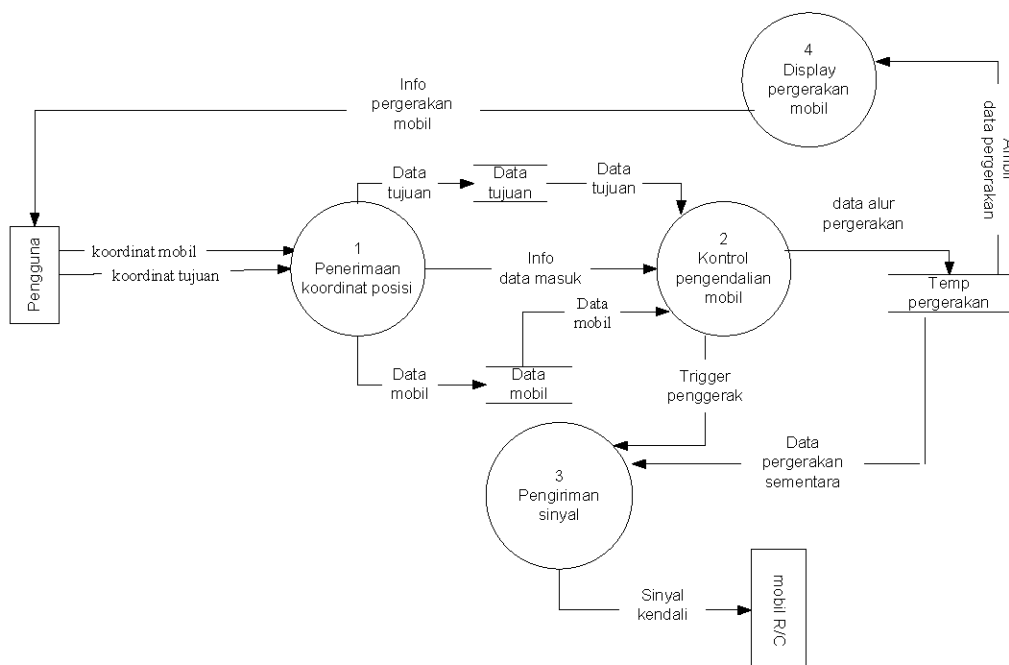
Dalam diagram ini menggambarkan proses aplikasi yang dibuat secara umum, serta menggambarkan pula keterkaitan dengan entitas luar seperti pengguna, data yang dimasukan oleh pengguna seperti koordinat mobil, koordinat tujuan.



**Gambar 4.1 :** DCD data kontek diagram level 0

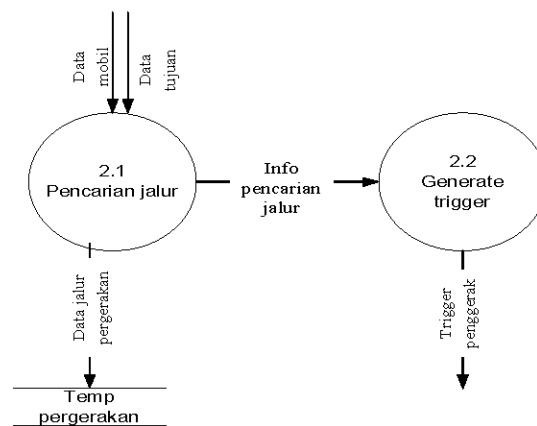
## 4.2 Data Flow Diagram

Diagram ini adalah suatu diagram lanjutan dari data konteks diagram. Maksud dari diagram lanjutan yaitu menjelaskan proses secara terperinci pada aliran data. Proses ini terdiri dari empat proses untuk level 1 yang terdiri dari *penerimaan koordinat, kontrol pengendalian mobil, pengiriman siny, dan display penggerak mobil*. Untuk level 2 terdiri dari dua proses yang terdiri dari *pencari jalur dan generate trigger*.



**Gambar 4.2 :** Data flow diagram level 1

Dalam level 2 inilah, sebagai proses perhitungan serta pengecekan yang menentukan mobil agar bisa belok atau lurus. Proses perhitungan dan pengecekan akan berhenti jika mobil tersebut sampai pada tujuan. Dalam level 2 juga terdapat sebuah proses yang sebagai pemicu gerak, untuk menggerakkan mobil R/C yang melalui sinyal kendali.



**Gambar 4.3 :** DFD level 2 proses 2

### 4.3 Pejelasan Proses

Dalam level 1 memiliki empat proses yang setiap proses fungsi tertentu, proses-proses itu adalah *penerimaan koordinat posisi*, *kontrol pendalian mobil*, *pengirim sinyal*, *display penggerak mobil*. PSPEC untuk level 1 adalah sebagai berikut :

1. Proses 1, proses ini menerima data dari operator atau pengguna dan data yang telah diterima disimpan dalam sebuah penyimpanan koordinat awal, penyimpanan ini terdiri dari dua bagian. Bagian pertama penyimpanan data mobil didalamnya berisi data koordinat awal mobil jika diinput. Bagian kedua adalah penyimpanan data tujuan, data ini digunakan untuk tujuan mobil bergerak.

**Judul**

{penerimaan input data koordinat awal yang kemudian dilanjutkan dengan penentuan jalur yang akan digunakan untuk menggerakkan mobil sampai tujuan serta menggerakkan mobil}

**Kamus Lokal**

x1,x2,y1,y2 : integer {koordinat mobil, tujuan}  
 datX1,dax2,daxY1,daxY2 : integer {penyimpanan koordinat awal}  
 tempX, tempY : integer {penyimpanan koordinat sementara}  
 procedure penerima(); {menerima input koordinat awal dari pengguna}  
 procedure cari\_jalur(x1,y1,sudut,x2,y2); {prosedur penentuan jalur}  
 procedure pengirim(); {prosedur pengiriman sinyal kendali}  
 procedure display\_penggerak(); {informasi pergerakan mobil}  
 procedure penggerak(x1:integer,y1:integer,x2:integer,y2:integer);  
 procedure kirim(); {pengiriman sinyal pengendali}

**Algoritma**

penerima();  
 cari\_jalur(x1,y1,x2,y2);  
 display\_penggerak();  
 penggerak(x1,y1,x2,y2);  
 kirim();

procedure penerima(); {untuk menerima data koordinat awal dan disimpan dalam penyimpan atau storage koordinat awal}

**Kamus**

kor\_mobilX1 : integer; kor\_mobilY1 : integer;  
 kor\_tujuanX2 : integer; kor\_tujuanY2 : integer;  
 st : integer;

**Algoritma**

Input(kor\_mobilX1);  
 Input(kor\_mobilY1);  
 Input(kor\_tujuanX2);  
 Input(kor\_tujuanY2);  
 If kor\_mobilX1  $\neq$  0 and kor\_mobilY1  $\neq$  0 then  
   Output('refresh');  
 Else  
   datX1 $\leftarrow$  kor\_mobilX1;  
   datY1 $\leftarrow$  kor\_mobilY1;  
 Endif  
 If kor\_tujuanX2  $\neq$  0 and kor\_tujuanY2  $\neq$  0 then  
   Output('refresh');  
 Else  
   datX2 $\leftarrow$  kor\_tujuanX2;  
   datY2 $\leftarrow$  kor\_tujuanY2;  
 Endif  
 cari\_jalur(x1,y1,x2,y2);

2. Proses 3, proses ini pada saat menerima trigger penggerak kemudian sebuah perintah yang berisi karakter yang sesuai dengan data penyimpanan temp penggerak.

Procedure kirim(); { mengirim sinyal kendali ke kontroler }
<p><b><u>Kamus</u></b>  bound_rate:integer;  port_open : boolean;  msg : char;</p>
<p><b><u>Algoritma</u></b>  Input(bound_rate);  If port_open = true then    If msg = 'a' then      Output('perintah maju');    Else if msg = 'b' then      Output('perintah maju sambil belok kanan');    Else if msg = 'c' then      Output('perintah maju sambil belok kiri');    Else if msg = 'd' then      Output('perintah mundur');    Else if msg = 'e' then      Output('perintah berhenti');    Endif    Endif    Endif    Endif  endif</p>

3. Proses 4, proses ini mengambil data pergerakan sementara yang akan dijadikan informasi bagi pengguna. Informasi ini untuk mengetahui posisi sekarang. Setiap perubahan data yang berubah akan langsung diinformasikan oleh proses ini.

Procedure display_penggerak(); {menampilkan informasi pergerakan mobil}
<p><b><u>Kamus</u></b>  g : integer;  h : integer;</p>

**Algoritma**

```

g←tempX;
h←tempY;
output(g);
output(h);

```

Pada level 2 adalah sebuah level lanjutan dari proses 2 level 1. level 2 ini memiliki dua proses yang terdiri dari pencarian jalur dan generate trigger, kedua proses ini berfungsi sebagai otak pengendali untuk mengetahui jalur mana yang digunakan. Dalam penjelasan proses ini akan menjelaskan PSPEC untuk level 2 ini sebagai berikut :

1. Proses 2.1, proses ini mengambil data dari sebuah penyimpanan koordinat awal misalnya disebut buffer. Buffer ini terdiri dari dua bagian yang pertama adalah data mobil, yang menyimpan data koordinat mobil, yang kedua adalah data tujuan. Setelah mengambil data tersebut dan terjadinya sebuah pergerakan mobil maka proses ini akan mencari jalur yang akan digunakan sebagai pergerakan, dan jalur yang ditentukan akan simpan di penyimpanan sementara. Proses ini akan berulang sampai kondisi dimana posisi mobil akan sama dengan posisi tujuan.

```

Procedure cari_jalur(x1,y1,x2,y2);
{pengecekan jalur }

```

**Kamus****Algoritma**

```

x1 ←datX1;
y1 ←datY1;
x2←datX2;
y2←datY2;
If x1 ≠ 0 and y1 ≠ 0 then
  repeat
    x1←x1+1;
    tempX←x1;
    y1←y1+1;
    tempY←y1;
    Penggerak(x1,y1,x2,y2);
  until x1= x2 or y1 =y2
endif
endif

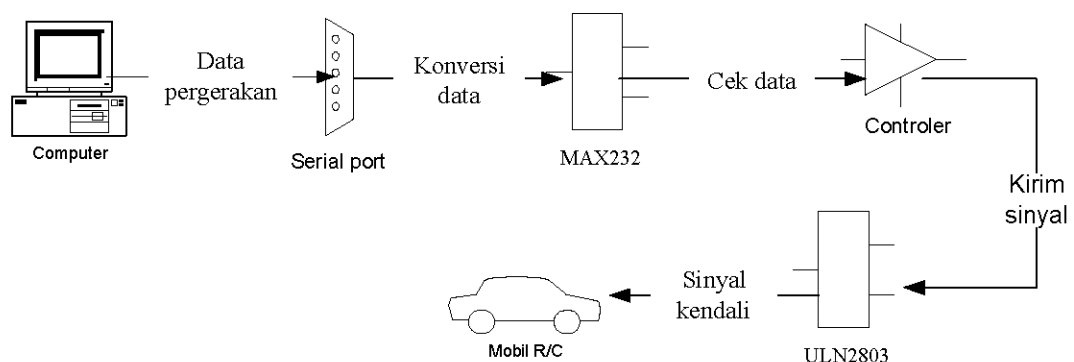
```

2. Proses 2.2, setelah mendapatkan jalur yang ditentukan maka proses ini menerima informasi bahwa proses 2.1 menetapkan jalur yang akan digunakan, dari informasi ini maka proses ini mengirimkan perintah secara langsung untuk menjalankan mobil

<pre> Procedure penggerak(x1:integer,y1:integer,x2:integer,y2:integer); {untuk menggerakkan mobil} </pre>
<p><b><u>Kamus</u></b>  gerak : boolean;</p>
<p><b><u>Algoritma</u></b>  gerak ← false;  if x1 ≠ x2 and y1 ≠ y2 then    gerak ← true;     kirim();  else    gerak ← false;  endif</p>

#### 4.4 Hubungan Fisik

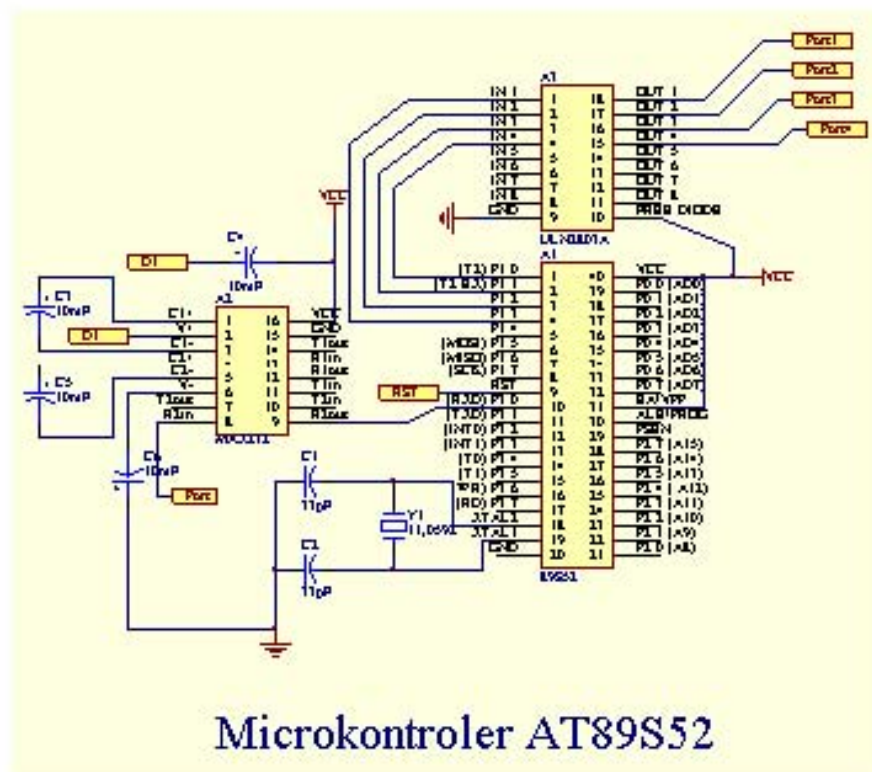
Dalam gambar 4.4 yang dibawah ini diperlihatkan bagaimana hubungan antara komputer dengan pengontrol. Disana kita bisa lihat bahwa komputer mengirimkan data pergerakan melalui serial port, selanjutnya data tersebut kirim ke sebuah drive untuk mengkonversi data menjadi data TTL, agar kotroller bisa membaca data yang diterima hingga sampai mengirimkan sinyal kendali ke pada mobil.



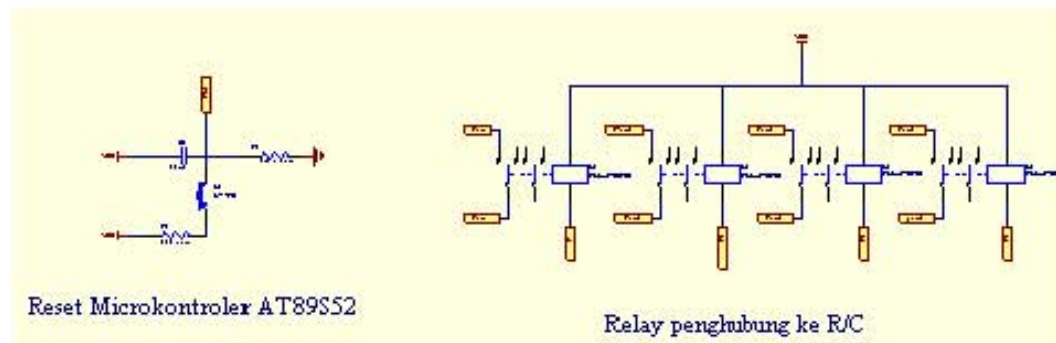
**Gambar 4.4 :** Hubungan fisik

#### 4.5 Schematic Elektronika

Pada bagian ini, kita akan melihat jalur elektronika yang akan mengontrol sinyal kedali, yang terdiri dari beberapa IC dan beberapa relay. IC tersebut terdiri dari satu buah *microcontroller*, satu buah driver MAX232, satu buah driver ULN2803A, dan empat buah relay yang bertegangan 5V. disamping itu juga komponen lain seperti pada fungsi reset, pada fungsi clock generator. Dibawah ini adalah gambar schematic elektronika sebagai berikut :



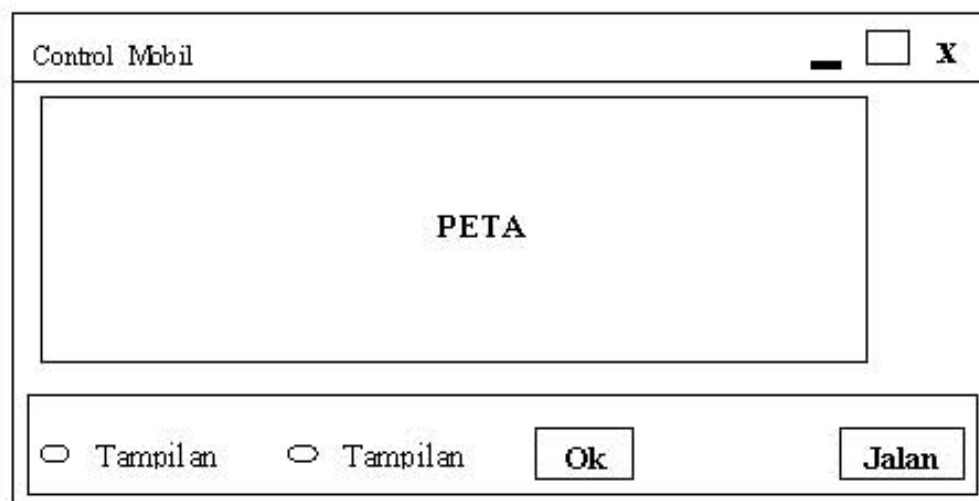
Gambar 4.5a : Schematic mikrokontroler AT89S52



**Gambar 4.5b :** *Skematik reset dan relay*

#### 4.6 Perancangan Tampilan

Tampilan sangatlah penting untuk dirancang agar pada saat pengguna merasa mudah untuk menggunakannya. Dalam segi pewarnaan tampilan yang dirancang tidak mencolok, agar mata kita yang melihat tidak merasa silau sehingga tidak terjadinya kesulitan untuk melihat tampilan dalam menggunakannya. Pada perancangan tampilan kali ini memiliki satu tampilan yang digunakan, selain itu memiliki beberapa tool yang digunakan seperti *button*, *radio button*, seperti yang kita lihat gambar di bawah ini :



**Gambar 4.6 :** *Perancangan tampilan*

Dalam tampilan yang diperlihatkan pada gambar 4.6, memiliki beberapa warna dan beberapa fungsi, yakni :

1. Warna putih digunakan sebagai warna dasar tampilan, agar tampilan tidak terlalu terang dan juga sebagai kombinasi warna pada peta. Karena objek-objek yang dijadikan peta menggunakan warna hitam.
2. Warna kotak yang sebagai batas peta digunakan warna hitam. Penggunaan warna hitam ini agar supaya tidak terlalu terang karena warna dasar tampilan berwarna putih.
3. *Radio button* (Tampilan mobil), teks yang digunakan berukuran 12 dan type teks yang digunakan *Times New Roman*, *radio button* ini berfungsi untuk menampilkan sebuah kotak yang digunakan sebagai mobil dan warna kotak tersebut berwarna merah.
4. *Radio button* (Tampilan Tujuan), teks yang digunakan berukuran 12 dan type teks yang digunakan *Times New Roman*, *radio button* ini berfungsi untuk menampilkan sebuah kotak kecil yang digunakan sebagai tujuan dan warna kotak ini berwarna biru.
5. *Button* (OK), *button* ini memiliki warna sesuai tampilan komputer yang digunakan, dan berfungsi untuk tidak mengaktifkan kedua *radio button* dan *button* putar agar pada saat dijalankan tidak bisa dirubah-rubah.
6. *Button* (Jalan), *button* ini memiliki warna sesuai tampilan komputer yang digunakan, dia berfungsi untuk menggerakkan mobil. Mobil akan berhenti pada saat mobil sudah berada pada posisi tujuan.