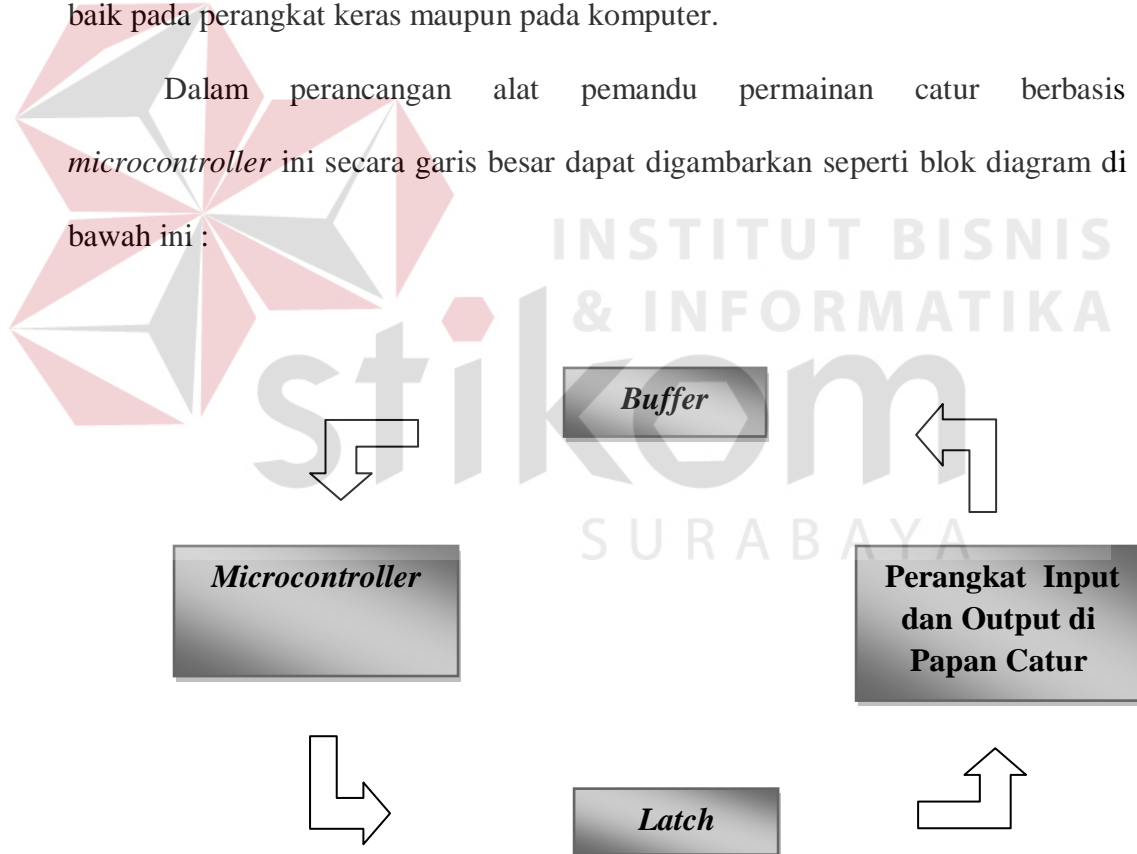


BAB III

METODE PENELITIAN

Metode penelitian yang digunakan dalam perancangan perangkat keras adalah studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari internet dan konsep-konsep teoritis dari buku-buku penunjang. Sedangkan dalam perancangan perangkat lunak berdasarkan teori dan aplikasi dari buku-buku penunjang sehingga dapat dilakukan percobaan langsung baik pada perangkat keras maupun pada komputer.

Dalam perancangan alat pemandu permainan catur berbasis *microcontroller* ini secara garis besar dapat digambarkan seperti blok diagram di bawah ini :



Gambar 3.1. Blok Diagram Keseluruhan Sistem

Dari Gambar 3.1. dapat dilihat bahwa sistem secara keseluruhan terbagi menjadi empat bagian yaitu *Microcontroller*, *Latch 74HC573*, Perangkat input dan output papan catur, serta *Buffer 74HC244*.

Buffer 74HC244 berfungsi untuk membaca data tiap baris dari deteksi letak bidak pada papan catur. *Microcontroller* berfungsi sebagai pengatur jalan dari bidak pada papan catur, dalam *microcontroller* tersebut memuat program-program untuk menentukan pergerakan bidak selanjutnya. Pergerakan bidak pion yang *valid* ditandai dengan menyalnya LED pada papan catur. *Latch 74HC573* untuk mengatur nyala dari LED di papan catur agar sesuai seperti yang dimaksudkan program pada *microcontroller*.

3.1. Perancangan Perangkat Keras

Perangkat keras merupakan rangkaian elektronik yang digunakan sebagai pemroses. Rangkaian-rangkaian berikut terdiri dari rangkaian pemroses (*Minimum System ATmega32*), rangkaian input berupa sakelar PCB (*Printed circuit board*) dan *Buffer 74HC244*, rangkaian output berupa *Latch 75HC573* dan LED.

Pada *Minimum system ATmega32* terdapat beberapa rangkaian pendukung yaitu rangkaian *reset* dan rangkaian *oscillator*. Pada rangkaian *reset* menggunakan *manual reset*. Pada rangkaian *oscillator* menggunakan komponen kristal 11.0592 MHz sebagai *clk (clock)*.

Rangkaian input pada perangkat ini terdiri dari *Buffer 74HC244* sebagai pengendali data input dari papan catur. Kemudian sensor input berupa desain pada papan catur yang cara kerjanya menyerupai sakelar yang dapat mendeteksi keberadaan bidak pada masing-masing kotak.

Rangkaian output pada perangkat ini terdiri dari *Latch 74HC573* sebagai penyangga data output untuk menyalakan LED pada papan catur. Sensor output berupa LED tiga milimeter berwarna merah yang dapat menyala sebagai tanda memandu pergerakan bidak catur.

3.1.1. Rangkaian *Microcontroller*

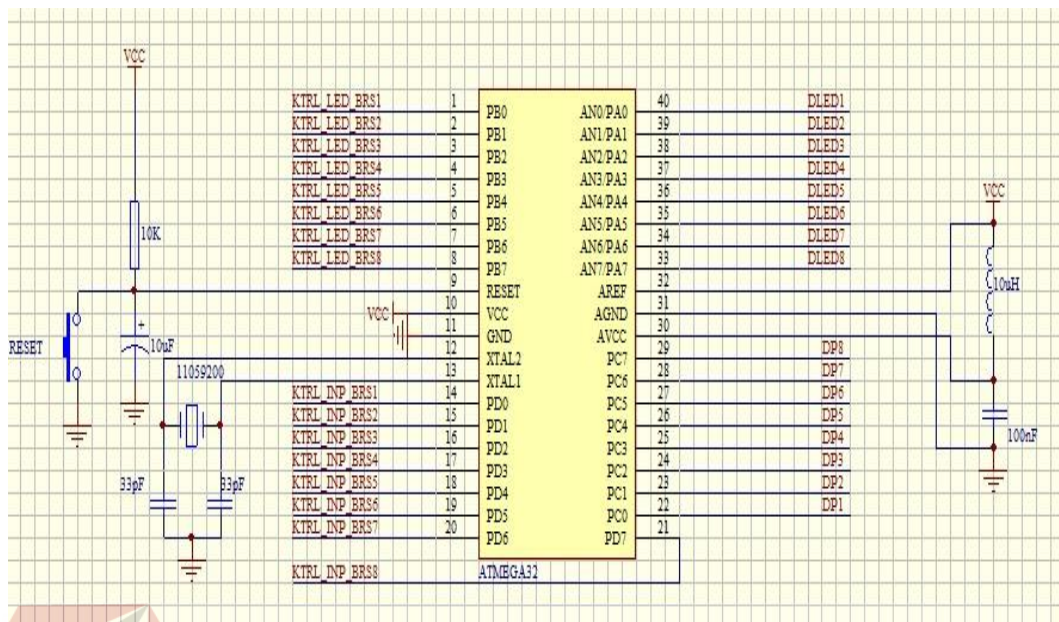
Pada proyek akhir ini dibuat piranti pengendali menggunakan *Microcontroller* keluaran AVR, yaitu ATmega32. Untuk mengaktifkan atau menjalankan *microcontroller* ini diperlukan rangkaian *minimum system*. Rangkaian *minimum system* tersebut terdiri rangkaian reset, rangkaian *oscillator*, rangkaian *power supply* dan rangkaian sistem *microcontroller*.

A. Rangkaian *minimum system microcontroller*

Untuk menjalankan *microcontroller* dibutuhkan sebuah rangkaian minimum agar *microcontroller* tersebut dapat bekerja dengan baik. Rangkaian *minimum microcontroller* terdiri dari rangkaian reset dan rangkaian *oscillator*.

Reset pada *microcontroller* ATmega32 terjadi dengan adanya logika *high* “1” selama dua *cycle* pada kaki RST pada *microcontroller* ATmega32. Setelah kondisi pin RST kembali *low*, maka *microcontroller* akan menjalankan program dari alamat 0000H. Dalam hal ini reset yang digunakan adalah *manual reset*.

Pada pin VCC diberi masukan tegangan operasi berkisar antara 4,5 volt sampai dengan 5,5 volt. Pin RST mendapat *input* dari *manual reset*. Rangkaian *minimum system* dapat dilihat pada Gambar 3.2 berikut:



Gambar 3.2. Rangkaian *Minimum System*

Pin XTAL1 dan XTAL2 merupakan pin *oscillator* bagi *microcontroller* ATmega32. Pin XTAL1 berfungsi sebagai input dan XTAL2 sebagai output *oscillator*. *Oscillator* ini bisa berasal dari kristal atau dari keramik *resonator*. Seperti yang sudah terlihat di atas, pin XTAL1 dan XTAL2 dihubungkan dengan komponen XTAL sebesar 11.0592 MHz. Pada proyek akhir ini dibuat rangkaian *oscillator internal* yang terbuat dari kristal. Nilai C1 dan C2 masing-masing 33 pF.

B. Perancangan *Interface I/O*

Rangkaian I/O dari *microcontroller* mempunyai kontrol direksi yang tiap bitnya dapat dikonfigurasi secara individual, maka dalam perancangan I/O yang digunakan ada yang berupa operasi port ada pula yang dikonfigurasi tiap bit I/O. Berikut ini akan diberikan konfigurasi dari I/O *microcontroller* tiap bit yang ada pada masing-masing port yang terdapat pada *microcontroller*.

1. Port A

Port A digunakan untuk output *Latch* 74HC573.

2. Port B

Port B digunakan untuk output kontrol LED.

3. Port C

Port C digunakan untuk input *Buffer* 74HC244.

4. Port D

Port D digunakan untuk kontrol input sakelar.

Untuk perancangan *interface* input dan output pada *microcontroller* yang lebih mendetil dapat dilihat pada tabel 3.1 dibawah :

Tabel 3.1. Perancangan *interface* input/output

Port	Alokasi Port pada Hardware
PortA.0	Data LED 1
PortA.1	Data LED 2
PortA.2	Data LED 3
PortA.3	Data LED 4
PortA.4	Data LED 5
PortA.5	Data LED 6
PortA.6	Data LED 7
PortA.7	Data LED 8
PortB.0	Kontrol <i>clock Latch</i> 1
PortB.1	Kontrol <i>clock Latch</i> 2
PortB.2	Kontrol <i>clock Latch</i> 3
PortB.3	Kontrol <i>clock Latch</i> 4
PortB.4	Kontrol <i>clock Latch</i> 5
PortB.5	Kontrol <i>clock Latch</i> 6
PortB.6	Kontrol <i>clock Latch</i> 7
PortB.7	Kontrol <i>clock Latch</i> 8
PortC.0	Data input sakelar 1
PortC.1	Data input sakelar 2
PortC.2	Data input sakelar 3
PortC.3	Data input sakelar 4
PortC.4	Data input sakelar 5
PortC.5	Data input sakelar 6
PortC.6	Data input sakelar 7
PortC.7	Data input sakelar 8

PortD.0	Kontrol <i>chip select Buffer 1</i>
PortD.1	Kontrol <i>chip select Buffer 2</i>
PortD.2	Kontrol <i>chip select Buffer 3</i>
PortD.3	Kontrol <i>chip select Buffer 4</i>
PortD.4	Kontrol <i>chip select Buffer 5</i>
PortD.5	Kontrol <i>chip select Buffer 6</i>
PortD.6	Kontrol <i>chip select Buffer 7</i>
PortD.7	Kontrol <i>chip select Buffer 8</i>

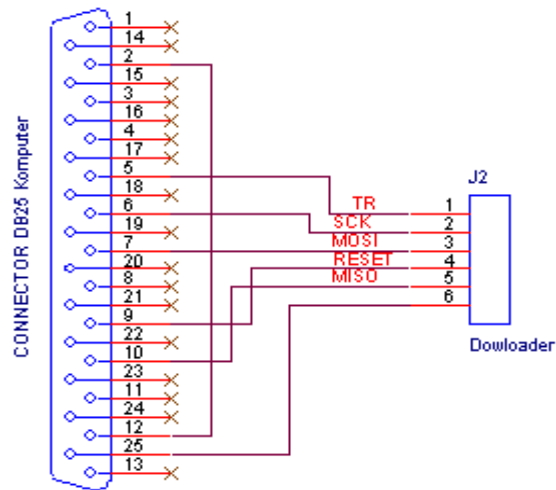
Pada pemrograman bahasa Basic menggunakan *compiler* Bascom AVR, inisialisasi Port pada *microcontroller* cukup dengan beberapa baris pernyataan seperti dibawah ini :

```
$regfile = "m32def.dat"
$crystal = 11059200

Ddra = &B11111111
Ddrb = &B11111111
Ddrc = &B00000000
Ddrd = &B11111111
```

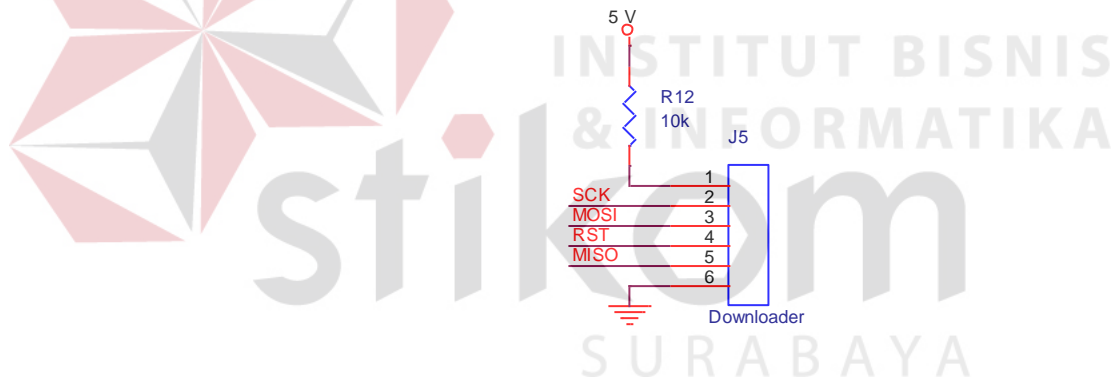
C. Program Downloader

Untuk melakukan proses *downloading* program dalam format *.HEX* dari komputer ke dalam *memory* program *internal microcontroller*, penulis menggunakan kabel *downloader* dengan *interface* DB25 yang dihubungkan pada port LPT1 pada komputer. Konfigurasi kabel *downloader* terdapat pada Gambar 3.3 berikut:



Gambar 3.3. Rangkaian kabel *downloader* pada port LPT

Sedangkan untuk konektor *downloader* pada *microcontroller* ATmega32 dapat dilihat pada Gambar 3.4 berikut:



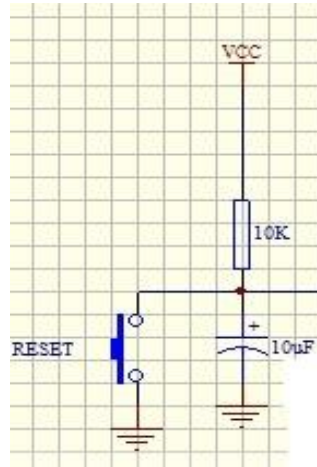
Gambar 3.4. Konektor *Downloader* pada *Microcontroller* ATmega32

D. Rangkaian *Reset*

Reset pada *microcontroller* ATmega32 terjadi dengan adanya logika *high* “1” selama dua *cycle* pada kaki RST pada *microcontroller* ATmega32. Setelah kondisi pin RST kembali *low*, maka *microcontroller* akan menjalankan program

dari alamat 0000H. Dalam hal ini reset yang digunakan adalah *manual reset*.

Rangkaian *reset* dapat dilihat pada Gambar 3.5 berikut :

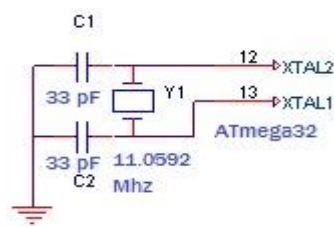


Gambar 3.5. Rangkaian Reset

E. Rangkaian *Oscillator*

Pin XTAL1 dan XTAL2 merupakan pin *oscillator* bagi *microcontroller* ATmega32. Pin XTAL1 berfungsi sebagai input dan XTAL2 sebagai output *oscillator*. *Oscillator* ini bisa berasal dari kristal atau dari keramik *resonator*.

Rangkaian *oscillator* dapat dilihat pada Gambar 3.6 berikut:

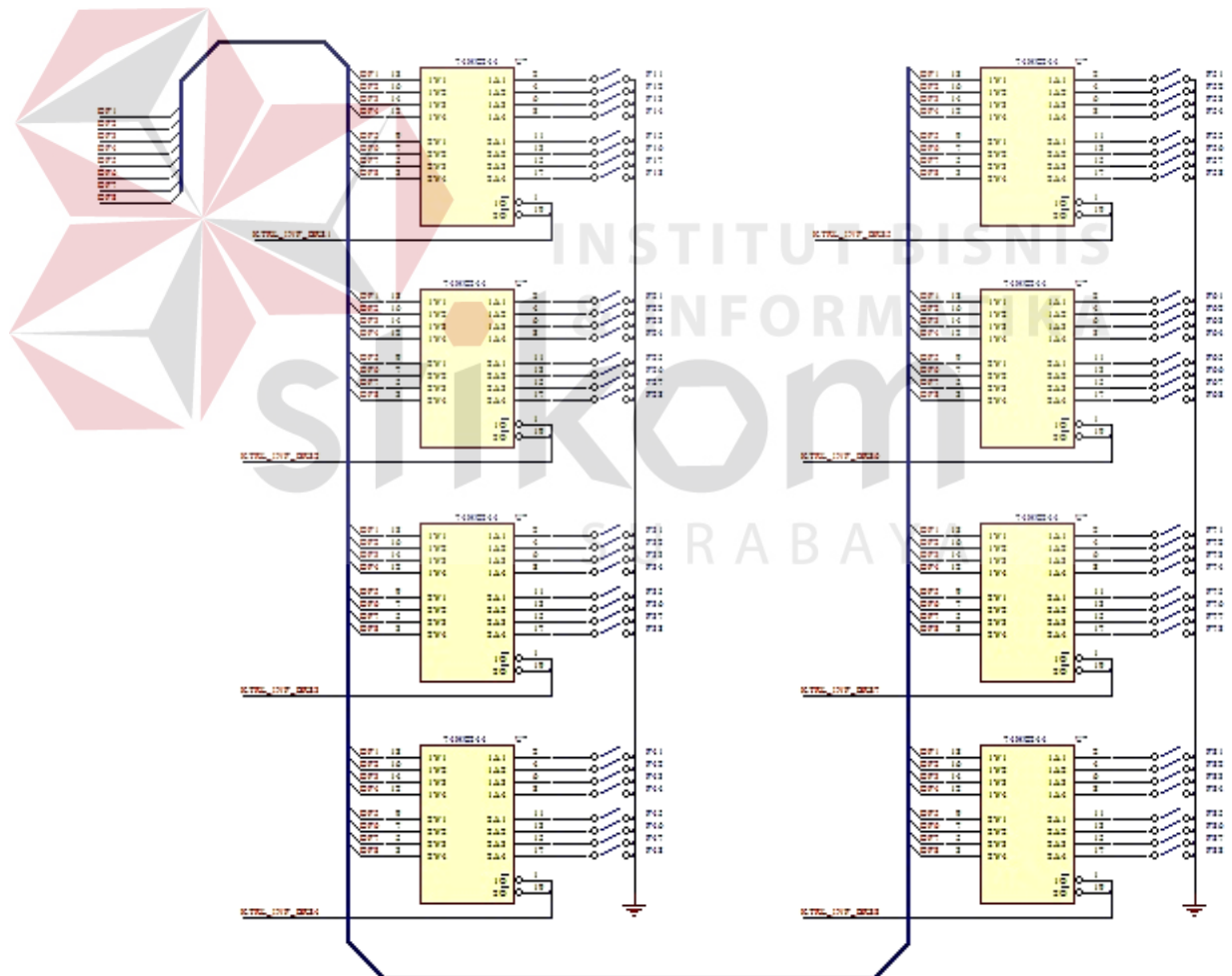


Gambar 3.6. Rangkaian *Oscillator*

Pada proyek akhir ini dibuat rangkaian *oscillator internal* yang terbuat dari kristal. Nilai C1 dan C2 masing-masing 33 pF.

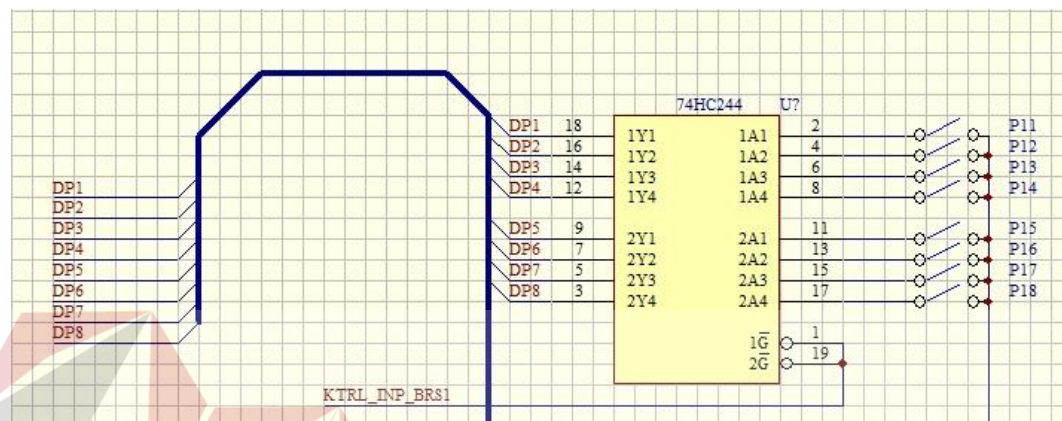
3.1.2. Rangkaian Input

Rangkaian input hal ini berfungsi sebagai pendeteksi keberadaan bidak dan posisi bidak pada saat ini. Rangkaian elektronika untuk perangkat input terdiri dari *Buffer 74HC244* dan serangkaian desain PCB (*printed circuit board*) yang menyerupai sakelar. Rangkaian desain PCB yang berfungsi sebagai pendeteksi keberadaan bidak catur akan dijelaskan lebih detil pada perancangan mekanik di sub-bab selanjutnya. Berikut adalah skematik dari rangkaian kontrol input yang terdiri dari sakelar dan *Buffer 74HC244* seperti pada gambar 3.7 ini :



Gambar 3.7. Rangkaian skematik kontrol input

Skematik di atas terdiri dari delapan *Buffer* 74HC244. Satu *Buffer* digunakan untuk mengontrol input dari satu baris sehingga masing-masing baris memiliki satu *Buffer* 74HC244. Berikut adalah gambar dari kontrol input baris pertama pada rangkaian input.



Gambar 3.8. Rangkaian skematik kontrol input baris satu

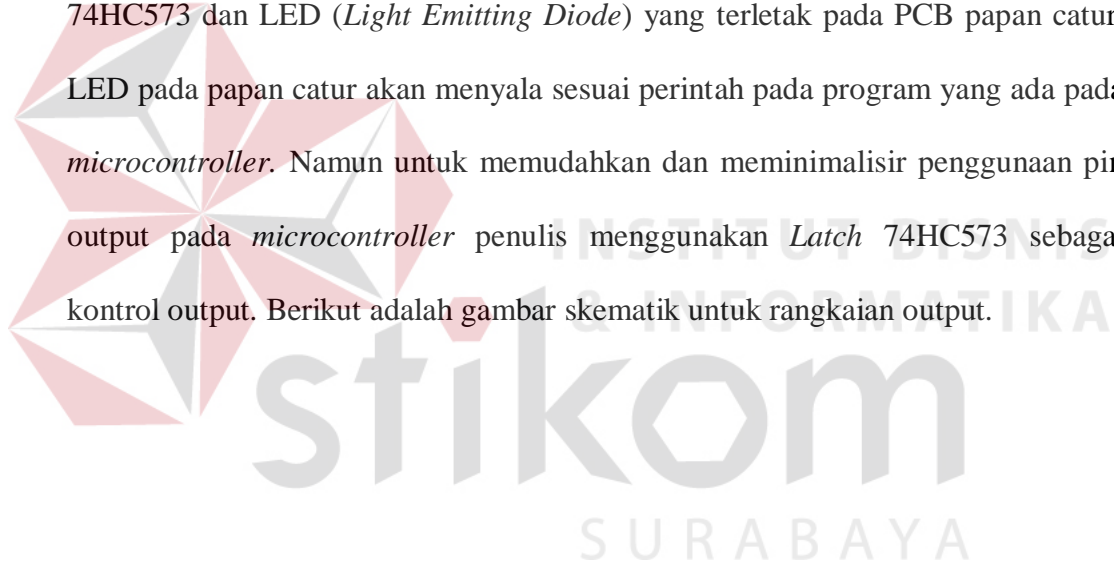
Gambar di atas adalah detail dari gambar 3.7. Pada gambar 3.8 ini terdapat satu *Buffer* 74HC244 untuk kontrol baris pertama begitu juga seterusnya hingga baris kedelapan. *Buffer* 74HC244 dalam hal ini berfungsi untuk membaca perbaris data bidak yang diangkat. *Buffer* ini berguna untuk membaca delapan bit pada baris tersebut selama *chip select* 74HC244 ini diaktifkan. Namun, jika *chip select* tidak diaktifkan data pada baris tersebut akan mengambang atau tidak terdeteksi 1 maupun 0. Dalam project tugas akhir ini, kotak catur yang terdeteksi keberadaan bidak pada kotak tersebut akan bernilai 0, begitu juga sebaliknya saat ada bidak akan bernilai 1.

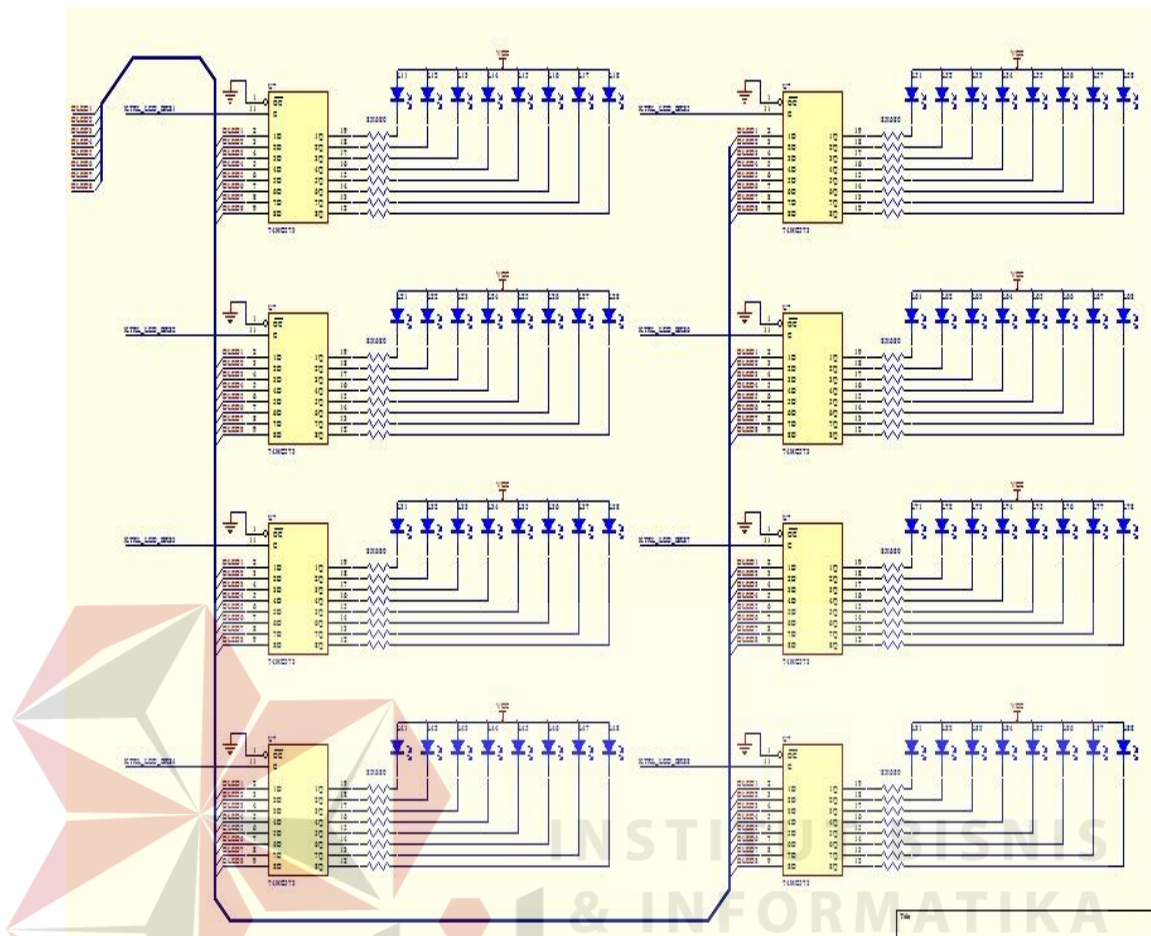
Rangkaian input ini menggunakan 2 port dari *microcontroller* yaitu port C dan port D. Port D digunakan untuk mengambil 8 bit data dari tiap-tiap *Buffer* 74HC244, sedangkan port C digunakan untuk mengendalikan *chip select* dari tiap *Buffer*. Penggunaan *Buffer* 74HC244 pada perangkat ini selain untuk keperluan

pengambilan data, juga untuk mengatasi keterbatasan *port* input dan output pada *microcontroller*, karena satu *chip Buffer* dapat menangani satu baris papan catur atau 8 input sekaligus.

3.1.3. Rangkaian Output

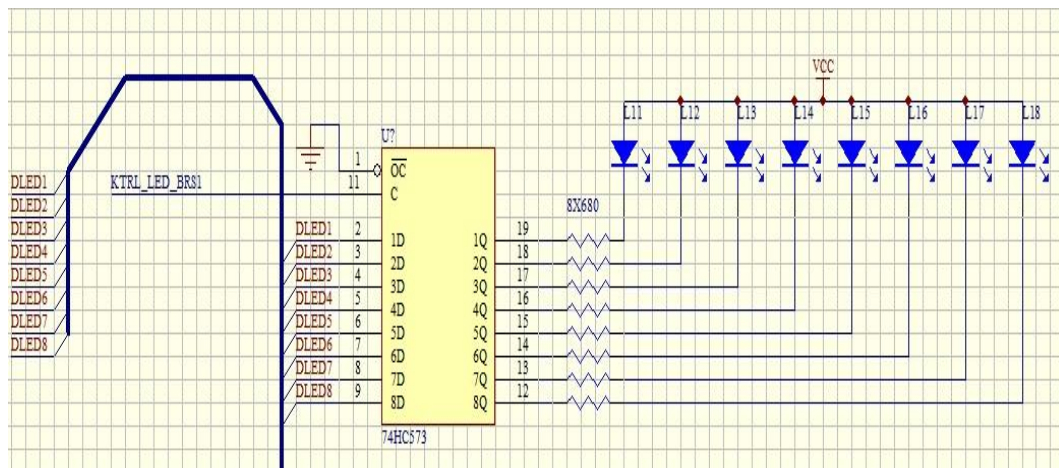
Rangkaian output proyek akhir ini berfungsi sebagai pengatur nyala LED (*Light Emitting Diode*) pada papan catur sebagai penanda langkah *valid* bidak yang diangkat saat itu. Rangkaian elektronika untuk output terdiri dari *Latch* 74HC573 dan LED (*Light Emitting Diode*) yang terletak pada PCB papan catur. LED pada papan catur akan menyala sesuai perintah pada program yang ada pada *microcontroller*. Namun untuk memudahkan dan meminimalisir penggunaan pin output pada *microcontroller* penulis menggunakan *Latch* 74HC573 sebagai kontrol output. Berikut adalah gambar skematik untuk rangkaian output.





Gambar 3.9. Rangkaian kontrol output papan catur

Skematik di atas terdiri dari delapan *Latch* 74HC573. Satu *Latch* digunakan untuk mengontrol output dari satu baris sehingga masing-masing baris memiliki satu *Latch* 74HC573. Berikut adalah gambar dari kontrol output baris pertama pada rangkaian output.

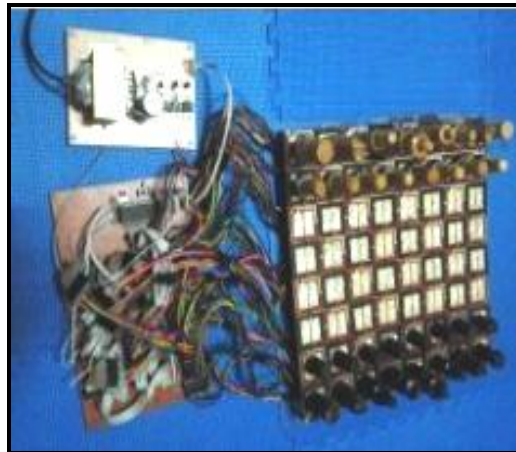


Gambar 3.10. Kontrol output baris pertama

Gambar 3.10 di atas adalah kontrol output baris pertama sebagai detail dari gambar 3.9 di halaman sebelumnya. Pada kontrol output baris pertama ini terdapat satu *Latch* 74HC573 sebagai penahan data yang dikirim dari *microcontroller* begitu juga seterusnya untuk baris berikutnya hingga baris kedelapan. Prinsip kerja dari *Latch* 74HC573 ini adalah mempertahankan data output yang diterima dari *microcontroller* untuk menyalakan LED pada papan catur selama satu *clock* yang diberikan pada *Latch* baris masing-masing. Untuk mengubah data pada baris yang diinginkan cukup dengan memberikan satu *clock* berikutnya pada *Latch* yang sesuai.

Rangkaian output ini menggunakan 2 port dari *microcontroller* yaitu port A dan port B. Port A digunakan untuk memberikan 8 bit data ke tiap-tiap *Latch* 74HC573, sedangkan port B digunakan untuk mengatur *clock* dari tiap *Latch*.

Kemudian pada perancangan *hardware* pada keseluruhan sistem dapat dilihat pada Gambar 3.11 berikut :



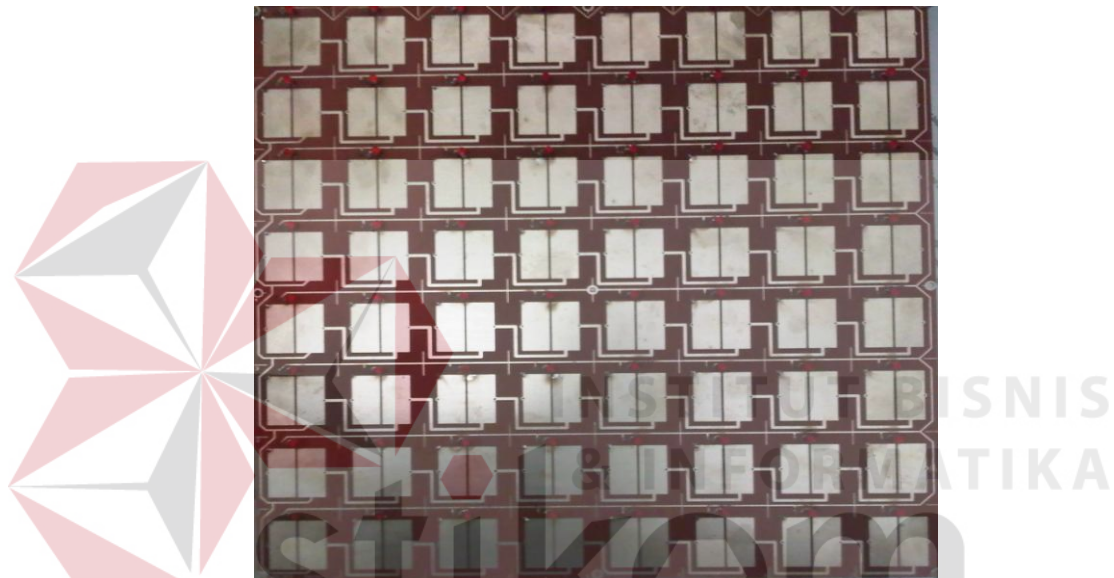
Gambar 3.11. Seluruh perangkat keras pada keseluruhan sistem

Pada gambar di atas, terdapat tiga bagian yaitu bagian papan catur, *control board*, dan juga *power supply board*. Pada bagian papan catur terdapat LED sebagai output dan sakelar PCB sebagai input. Pada bagian *control board* terdapat *Latch 74HC573* dan *Buffer 74HC244* yang tergabung dengan *minimum system ATmega32*. Kemudian *board* terakhir terdapat rangkaian *power supply*.

3.2. Perancangan Mekanik

Untuk perancangan mekanik, sebisa mungkin untuk merancang papan catur yang bisa mendeteksi keberadaan bidak catur pada tiap kotak papan catur tersebut. Untuk merancang mekanik yang dapat memenuhi kriteria tersebut, Papan catur dan bidak catur harus berfungsi seperti sakelar dimana pada saat ada bidak catur pada kotak, maka kotak mengirim data kepada *Buffer 74HC244* untuk diteruskan menuju *microcontroller*. Selain sebagai pendeteksi keberadaan bidak catur, papan catur juga harus dilengkapi indikator di setiap kotak papan catur. Indikator ini berfungsi sebagai penanda bahwa bidak yang diangkat pada saat itu *valid* untuk menempati kotak yang indikatornya menyala.

Sensor yang digunakan sebagai pendeteksi buah catur yang sedang dijalankan pada papan catur adalah serangkaian desain PCB yang menyerupai sakelar. Masing-masing petak pada papan catur terdapat sepasang pelat tembaga PCB, dimana pemasangannya menyerupai *keypad* yang sama-sama menggunakan metode *scanning* sehingga untuk mendapatkan 64 petak dibutuhkan 8 baris dan 8 kolom. Rangkaian input pada papan catur ditunjukkan pada Gambar 3.12 :



Gambar 3.12. Design PCB deteksi input

Pada desain papan catur di atas terdapat indikator berupa LED (*light emitting diode*) berukuran 3 milimeter sebagai penanda *valid* atau tidaknya sebuah bidak menempati suatu kotak pada papan catur. Selain desain sakelar pada papan PCB ini, juga ada perangkat pendukung yaitu logam yang diletakkan dibawah permukaan masing-masing buah catur. Hal ini berfungsi sebagai deteksi letak bidak disaat logam pada bidak catur menyentuh salah satu kotak pada papan PCB maka dua pelat tembaga pada PCB akan tersambung menyerupai cara kerja sakelar. Berikut adalah gambar dari logam yang diletakkan pada permukaan bawah masing-masing bidak :

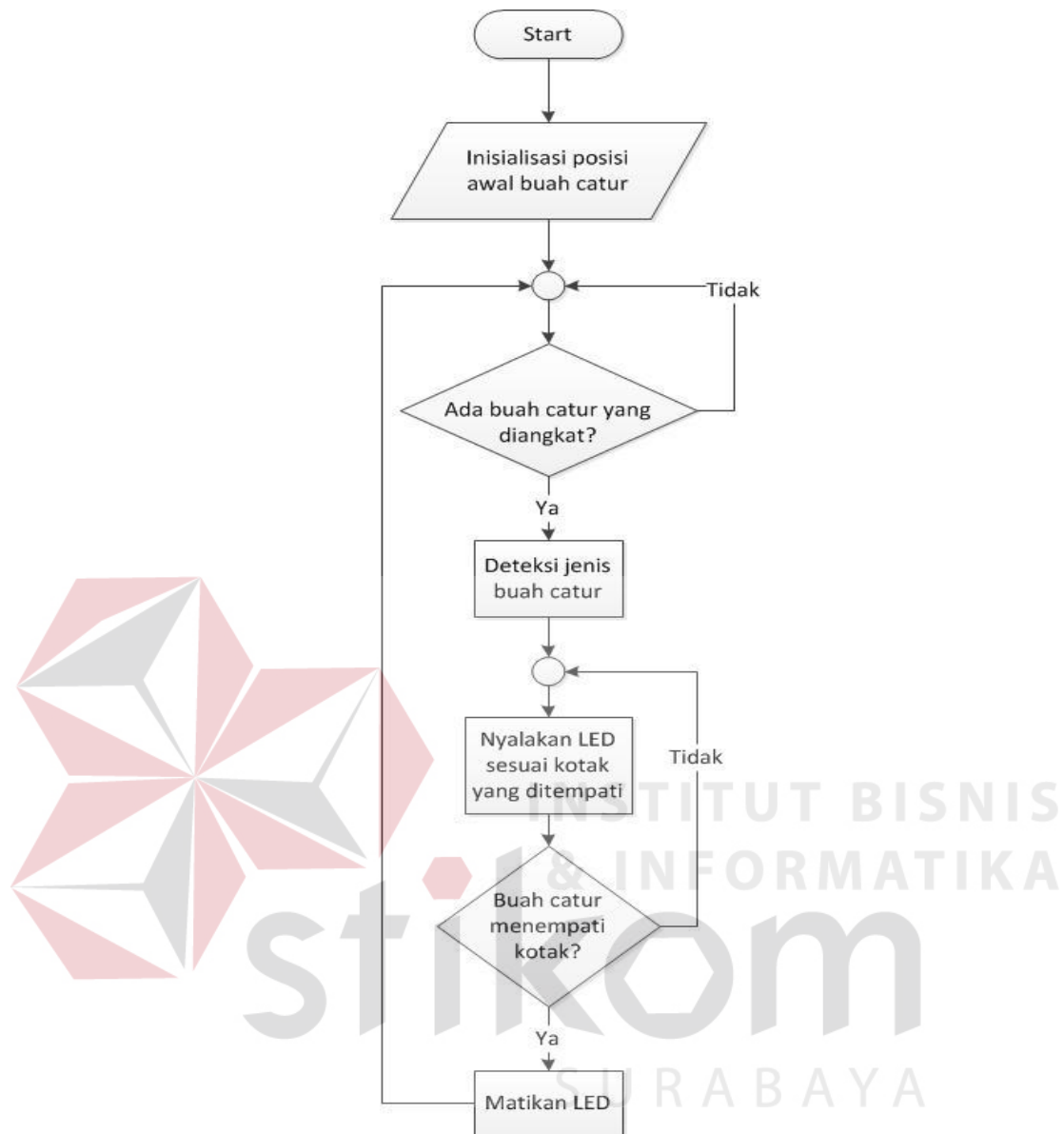


Gambar 3.13. Logam pada permukaan bawah Bidak

Untuk logam yang digunakan pada permukaan bawah bidak adalah sebuah mur dan logam. Penggunaan logam ini digunakan sebagai konduktor pada sistem sakelar sebagai input deteksi bidak.

3.3. Perancangan Perangkat Lunak

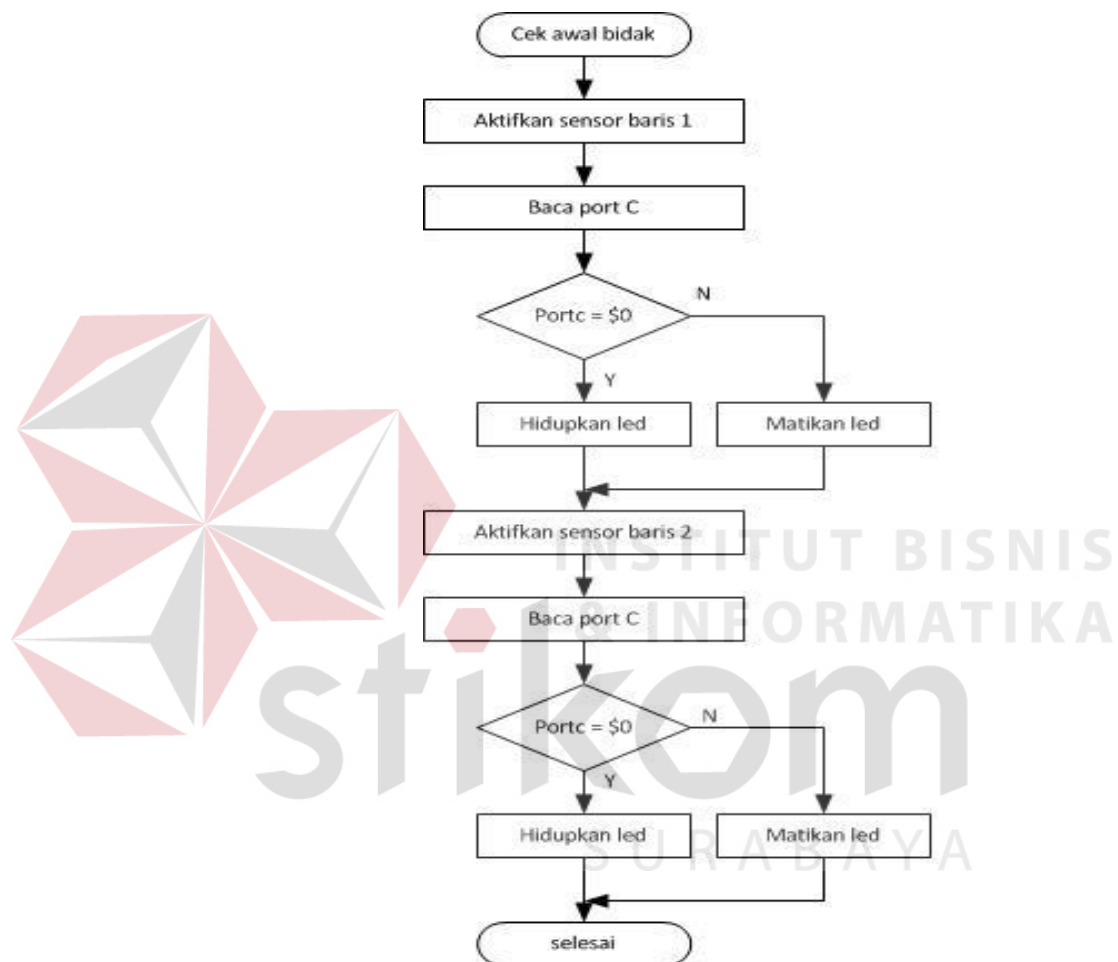
Perangkat lunak pada *microcontroller* bertujuan untuk memudahkan *user* dalam bermain catur. Seluruh perangkat lunak pada sistem pemandu permainan catur ini menggunakan *memory* pada *microcontroller*. Perangkat lunak pada *microcontroller* ini akan menampilkan arah pola pergerakan bidak yang dijalankan *user*. Diagram alir perangkat lunak dapat dilihat pada Gambar 3.14 berikut:



Gambar 3.14. Diagram Alir Perangkat Lunak

Terdapat empat proses inti pada keseluruhan sistem ini yaitu proses inisialisasi, proses pendeteksian bidak yang diangkat, proses menyalakan LED, dan terakhir adalah proses pendeteksi bidak diletakkan. Proses yang pertama adalah inisialisasi bidak pada awal permainan, inisialisasi ini bertujuan untuk memberi identitas pada 16 bidak putih yang terletak di baris pertama dan kedua. Proses inisialisasi ini sangat penting untuk pendeteksian bidak yang dijalankan

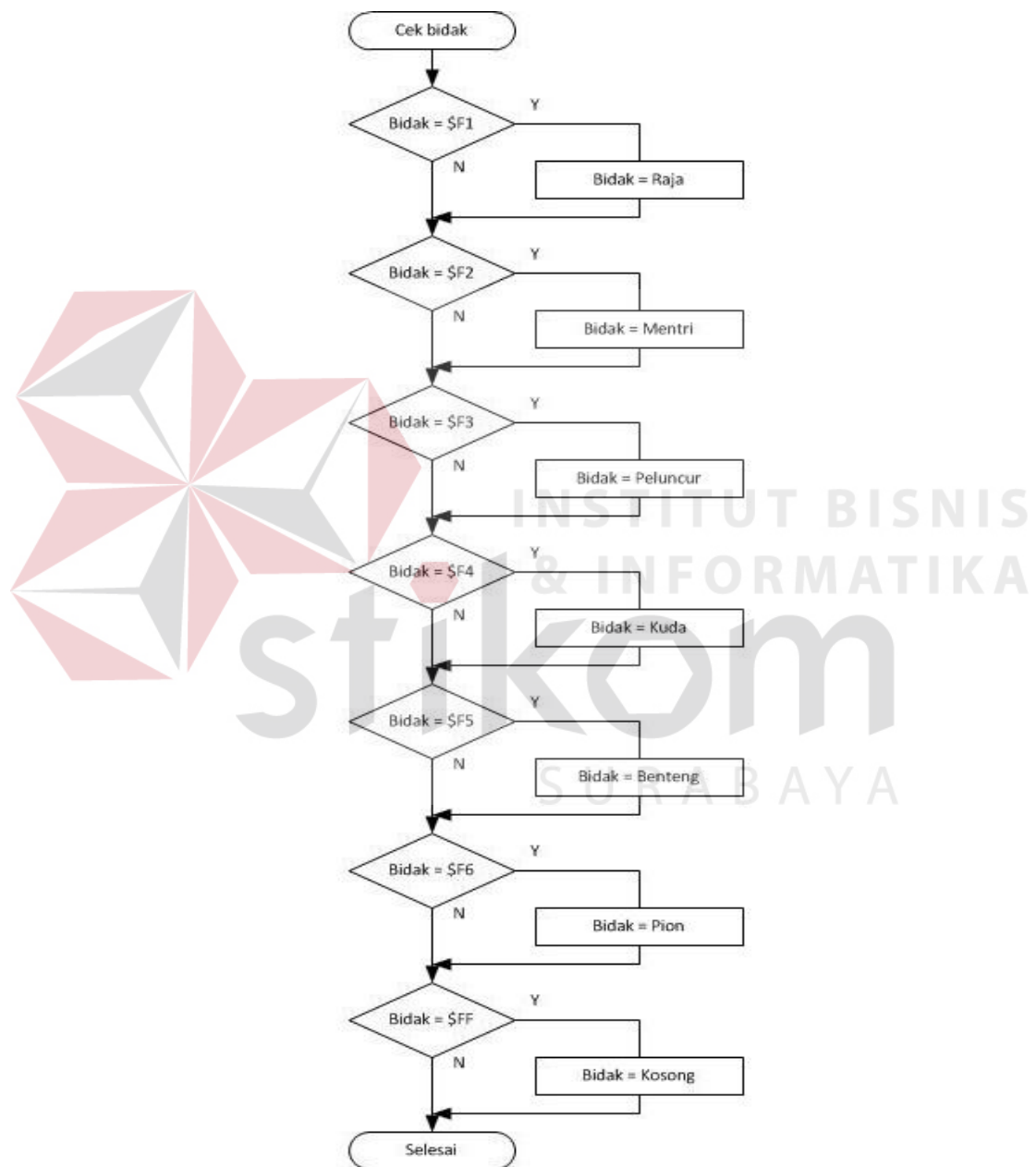
pada papan catur. Pendeteksian bidak ini bertanggung jawab dalam pengambilan keputusan saat menyalakan LED pemandu pola pergerakan bidak catur yang sedang diangkat. Diagram alir untuk proses pendeteksian awal bidak terlihat pada gambar dibawah ini:



Gambar 3.15. Diagram alir inisialisasi awal

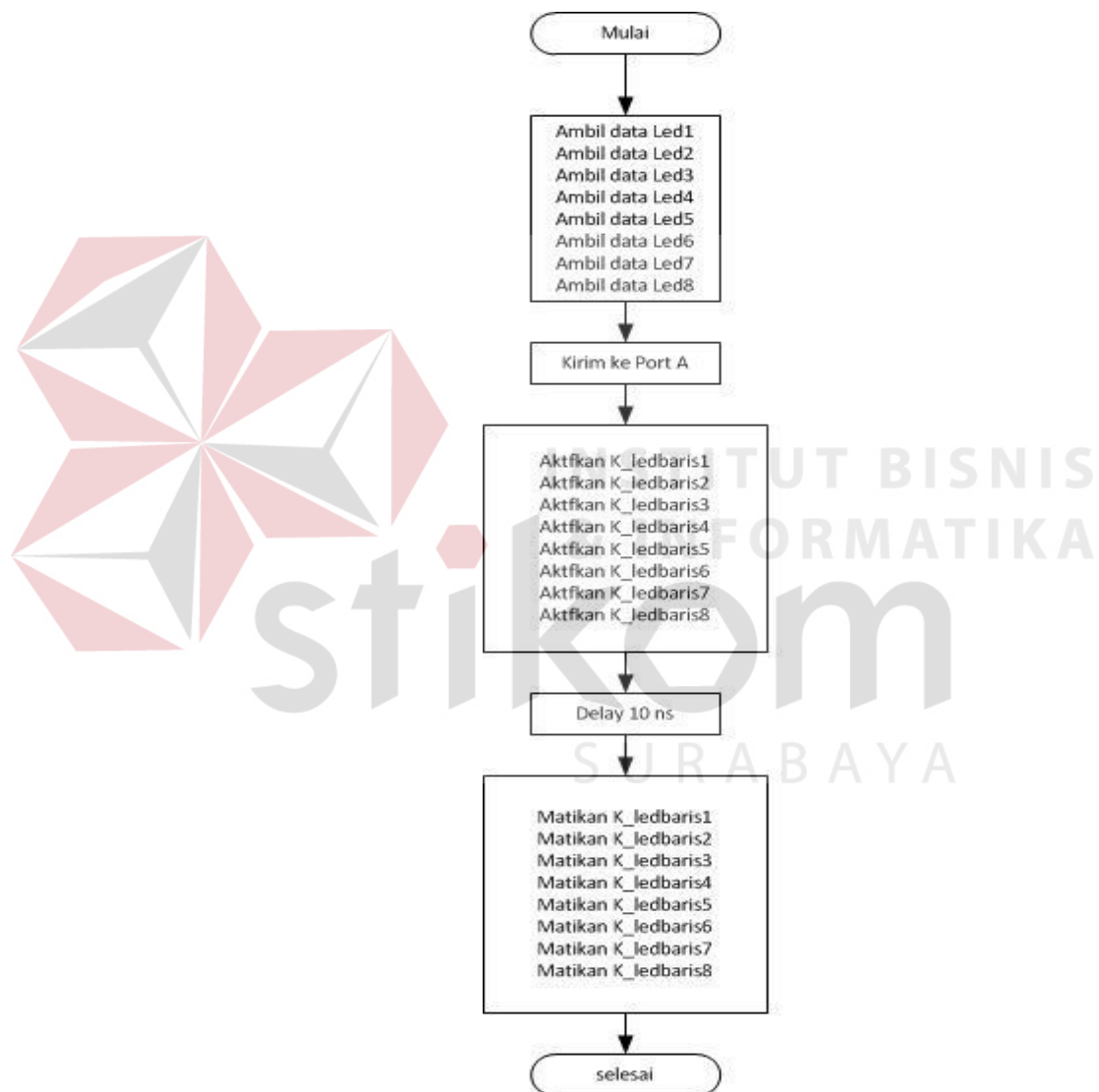
Pada proses inisialisasi ini bidak yang terdeteksi menempati satu kotak papan catur akan menyalakan LED pada kotak tersebut sebagai tanda konfirmasi bidak tersebut telah diberi identitas. Setelah 16 bidak tersebut telah lengkap diinisialisasi, proses selanjutnya adalah menunggu salah satu bidak terangkat. Proses ini berulang hingga terdeteksi ada bidak yang diangkat, begitu ada bidak

yang terangkat, proses pendeteksian jenis bidak yang diangkat berjalan. Proses pembacaan jenis bidak ini akan memutuskan jenis bidak yang diangkat tersebut adalah bidak Raja, Menteri, Peluncur, Kuda, Benteng, ataupun Pion. Gambar berikut adalah diagram alir dari proses pendeteksian bidak:



Gambar 3.16. Diagram alir deteksi jenis bidak

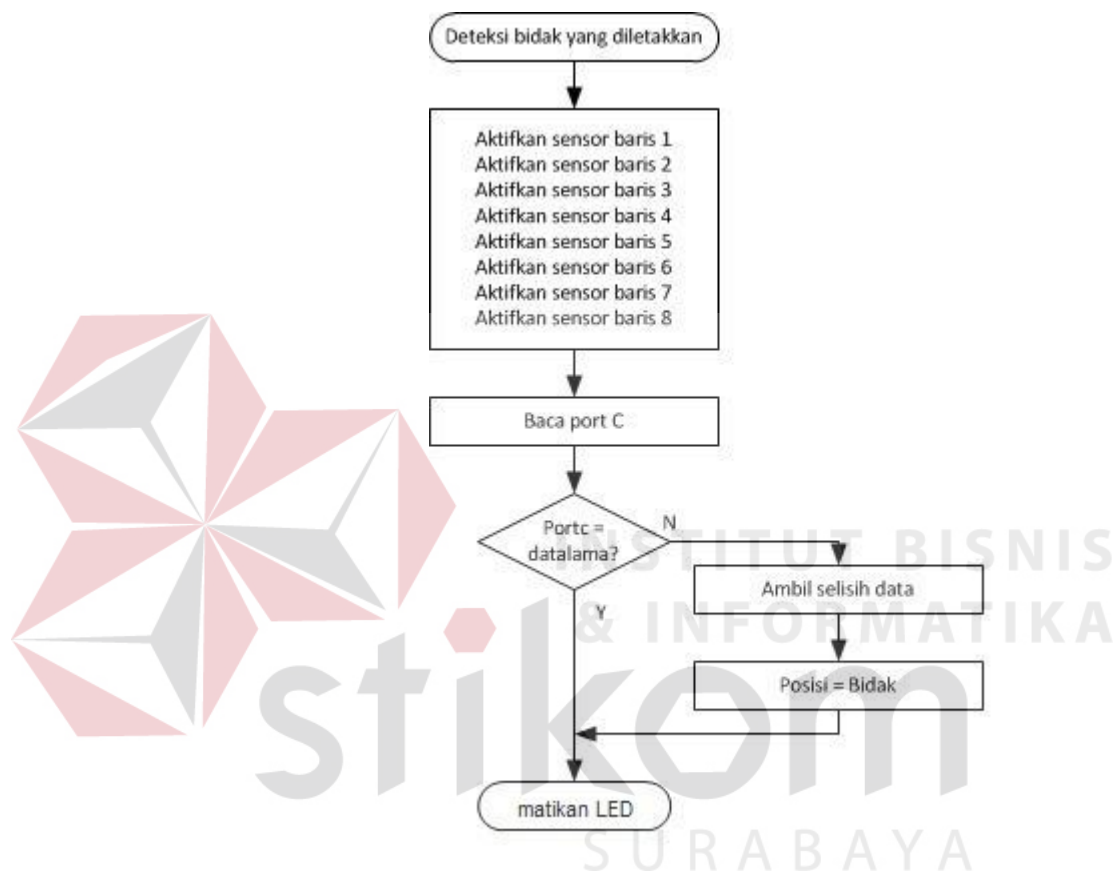
Setelah diketahui jenis bidak yang terangkat, proses berikutnya adalah menjalankan subrutin sesuai jenis bidak yang telah dideteksi tersebut. Pada subrutin ini dilakukan pengolahan data di variabel yang menyimpan informasi posisi bidak tersebut dan menghasilkan nyala LED sesuai pola pergerakan bidak yang diangkat. Berikut diagram alir proses menyalakan LED pada papan catur:



Gambar 3.17. Diagram alir proses menyalakan LED

Pada proses subrutin masing-masing jenis bidak, perangkat pemandu permainan catur akan menampilkan arah pola pergerakan sesuai bidak yang dideteksi pada proses sebelumnya. Kemudian, langkah selanjutnya adalah

menunggu bidak yang terangkat tadi menempati salah satu kotak yang menyala pada papan catur, kondisi ini berulang hingga bidak diletakkan dengan baik pada salah satu kotak tersebut. Gambar berikut adalah diagram alir proses pendeteksian bidak yang diletakkan:



Gambar 3.18. Diagram alir proses bidak diletakkan

Setelah bidak menempati salah satu kotak yang menyala, semua lampu LED dipadamkan kecuali LED pada kotak yang ditempati. Nyala LED pada kotak yang ditempati menyala sedikit lebih lama dari LED pada kotak yang lainnya sebagai tanda konfirmasi bahwa bidak tersebut telah menempati posisi yang baru. Setelah proses ini selesai perangkat pemanduan permainan catur akan kembali menunggu ada bidak selanjutnya yang terangkat.