penyimpan data (*zip drive*), *flash disk*, kamera digital dan perangkat lainnya. USB sangat mendukung transfer data sebesar 12 Mbps dan biasanya disediakan minimal 2 *port* pada tiap *motherboard*. Jika dibandingkan dengan *port* paralel dan *port* serial, penggunaan *port* USB lebih mudah dalam penggunaannya.

Spesifikasi USB versi 1.0 dirilis pertama kali tahun 1996. kemudian versi 1.1 dirilis oleh Intel tahun 1998 dengan tambahan kemampuan transfer data pada kecepatan rendah (1,5 MBits/s) dan kecepatan penuh (12 MBit/s). Versi 2.0 diumumkan tahun 1999, ditambah dengan kemampuan transfer data pada kecepatan tinggi (480 MBit/s).

Selain digunakan untuk menghubungkan periperal lain ke PC maka kegunaan dari *port* USB tersebut dapat dikembangkan lagi. Salah satu ide pengembangan dari penggunaan *port* USB yaitu menggerakkan motor *stepper*. Motor *stepper* akan digerakkan dengan menggunakan perangkat lunak Delphi 7.0 dan mikrokontroler sebagai penyimpan data instruksi yang akan digunakan untuk menggerakkan motor tersebut.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat diambil suatu rumusan masalah sebagai berikut:

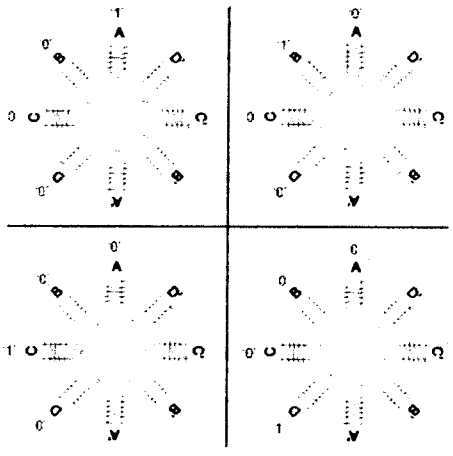
1. Bagaimana mengendalikan motor *stepper* melalui USB *interface* (antaramuka) dengan perantara mikrokontroler ATMega8535 melalui mikrokontroler AT90S2313.

**BAB V Penutup**

Berisi kesimpulan dan saran-saran dari proses perancangan, simulasi sistem, serta keterbatasan-keterbatasan yang ditemukan dan juga asumsi-asumsi yang dibuat selama melakukan penelitian.

yang bersebelahan diberi tegangan dan catu tegangan sebelumnya dilepas, maka kutub magnet tetap pada rotor itu akan perpindah posisi menuju kutub magnet lilitan yang dihasilkan. Berarti telah terjadi gerakan 1 *step*. Bila langkah ini diulang terus-menerus, dengan memberikan tegangan secara bergantian ke lilitan-lilitan yang bersebelahan, maka rotor akan "berputar".

Sedangkan untuk membalikkan putaran motor *stepper* cukup membalikkan urutan pemberian arus pada lilitan. Untuk memperlambat atau mempercepat putaran, cukup mengatur waktu urutan pemberian arus saja. Akan tetapi, terlalu lambat akan menyebabkan motor *stepper* bergetar dan jika terlalu cepat akan mengakibatkan motor tidak mau berputar (slip).



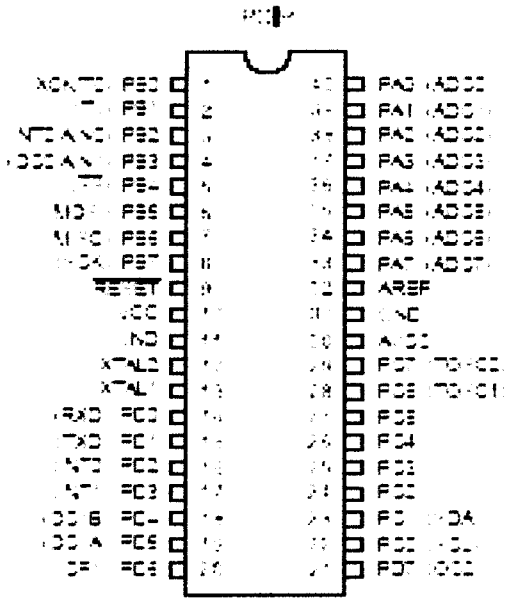
**Gambar 2.1.** Konstruksi motor *stepper*

Logika perputaran rotor tersebut dapat dianalogikan secara langsung dengan data '0' atau '1' yang diberikan secara serentak terhadap semua lilitan stator motor. Hal ini memudahkan bagi '*system designer*' dalam hal menciptakan putaran-putaran motor *stepper* secara bebas dengan hanya mempermainkan bit-bit

4. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* menghemat penggunaan daya listrik.

### 2.3.3 Konfigurasi Pin ATMega8535

Konfigurasi pin ATMega8535 bisa dilihat pada Gambar 2.6 berikut ini.



**Gambar 2.7** Pin ATMega8535

Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATMega8535 sebagai berikut:

1. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND merupakan pin *ground.*
3. *Port* A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
4. *Port* B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu *Timer/Counter,* komparator analog dan SPI

melampaui kecepatan transfer *port* paralel dan serial karena telah menyempurnakan proses transfer tersebut. Kabel USB mengurangi derau dan distorsi selama data dikirim, sehingga data dapat diterima dengan sedikit kesalahan

### 2.5.1 USB *Function*

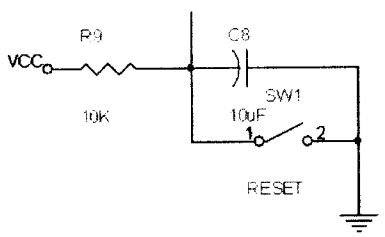
Suatu piranti USB dapat dikatakan sebagai sebuah alat *transceiver* (pengirim sekaligus penerima), baik *host* maupun peralatan USB itu sendiri. Sebuah istilah baru diperkenalkan, yakni USB *function* yang maksudnya adalah peralatan USB yang memiliki kemampuan khusus seperti *printer, scanner*, modem, dan perangkat lainnya..

Penerapan koneksi USB terhadap alat lain (*device*) saat ini dipecahkan dengan 2 (dua) jalan, yaitu:

1. Menggunakan mikrokontroler untuk alat (*device*) yang mempunyai *port* USB. Adalah penting untuk mengetahui bagaimana cara kerja USB dan menulis *firmware* ke dalam mikrokontroler secara tepat. Sebagai tambahan, penting juga untuk menginstall *driver* pada komputer (selama sistem operasi tidak termasuk dalam kelas USB standart). Kerugian utama pada pabrik-pabrik kecil dan amatir adalah kurang tersedianya jenis mikrokontroller dan tingginya harga mikrokontroler tersebut dibandingkan dengan mikrokontroler RS232.
2. Menggunakan konverter universal antara USB dan *interface* lainnya. *Interface* lainnya ini dapat berupa RS232, data *bus* 8 bit atau *bus* TWI.

Pada rangkaian osilator ini digunakan dua buah kapasitor 22pF. Sedangkan rangkaian *power on reset* berfungsi untuk menjaga agar pin RST mikrokontroler selalu berlogika rendah saat mikrokontroler mengeksekusi program. Mikrokontroler direset pada transisi tegangan rendah ke tegangan tinggi, oleh karena itu pada pin RST dipasang kapasitor yang terhubung ke VCC dan resistor ke *ground* yang akan menjaga RST bernilai 1 (satu) saat pengisian kapasitor dan bernilai 0 (nol) saat kapasitor penuh.

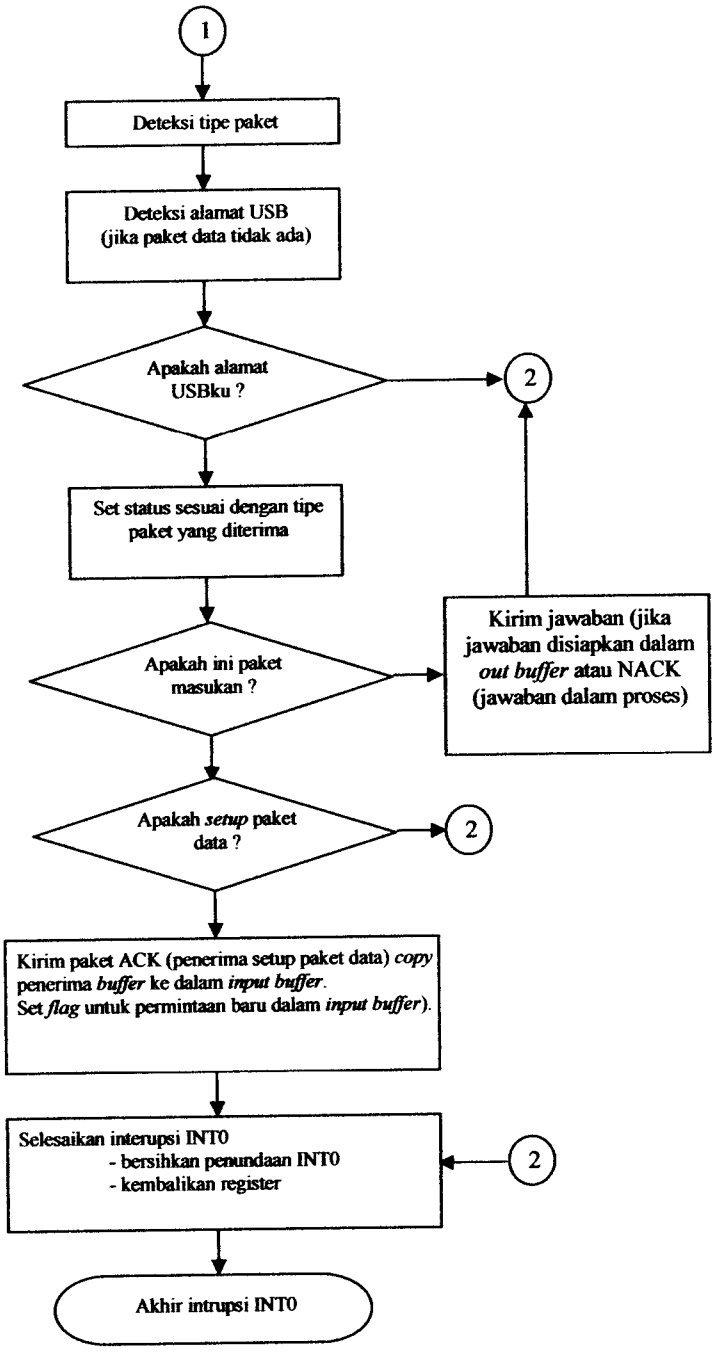
Pada saat sumber tegangan diaktifkan kapasitor terhubung singkat sehingga arus mengalir dari VCC langsung ke kaki RST sehingga reset berlogika 1, kemudian kapasitor terisi hingga tegangan pada kapasitor sama dengan VCC. Pada saat itu kapasitor terisi penuh. Dengan demikian tegangan reset akan turun menjadi 0 sehingga kaki RST berlogika 0.




**Gambar 3.5** Rangkaian Power On Reset

### *Port* B Mikrokontroler AT90S2313

*Port* B merupakan *port* I/O 8-bit *bi-directional*. Pin-pin pada *port* ini dapat diberi resistor *pull-up* internal secara individu. PB0 dan PB1 juga dapat digunakan untuk melayani *input* sebagai komparator analog. *Buffer port* B dapat mencatu




**Gambar 3.7** Diagram alir mikrokontroler

Adapun penjelasan diagram *flowchart* tersebut adalah sebagai berikut: Interupsi eksternal 0 aktif sepanjang waktu, sementara *firmware* bekerja. Rutin ini

*FinishReceiving:*

Merupakan pengkopian baris data dari paket penerimaan USB untuk penguraian kode paket (untuk penguraian kode NRZI dan *bitstuffing*).

*USB Reset:*

Inisialisasi *interface* USB terhadap nilai bawaannya (sejak dinyalakan)

*SendPreparedUSBAnswer:*

Mengirimkan kandungan *output buffer* yang telah disiapkan ke *line* USB, pengkodean NRZI dan *bitstuffing* dijalankan selama transmisi. Paket dihentikan dengan EOP.

*ToggleDATAPID:*

Identifikasi paket *Toggle* Data PID antara DATA0 dan DATA1 PID. *Toggling* ini penting selama transmisi untuk setiap spesifikasi USB.

*ComposeZeroDATA1PIDAnswer:*

Penyusunan jawaban *zero* selama transmisi. Jawaban *zero* tidak ada datanya dan digunakan untuk beberapa kasus sebagai jawaban ketika tidak ada data yang tersedia pada alat.

*InitACKBuffer:*

Memulai *buffer* dalam RAM dengan data ACK. *Buffer* ini secara periodik mengirim jawaban sehingga menjaga alat dalam memori.

*SendACK:*

Mengirimkan paket ACK ke *line* USB.