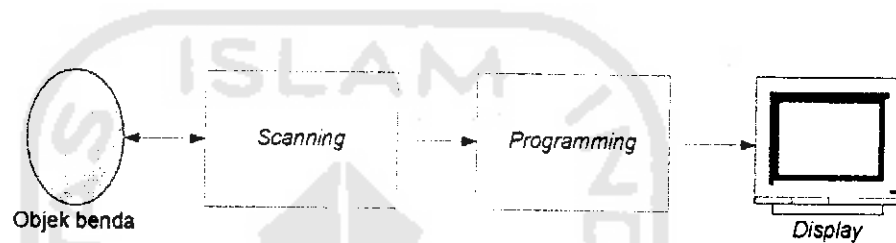


### BAB III

## PERANCANGAN SISTEM

Perancangan sistem secara garis besar diperlihatkan pada gambar 3.1 dibawah ini:



Gambar 3.1 Diagram blok proses

Objek berupa benda akan diambil datanya. Data tersebut adalah waktu pantul dari gelombang ultrasonik untuk diubah menjadi tingkatan warna abu – abu yang kemudian ditampilkan menjadi citra yang bisa dilihat pada layar monitor komputer.

Proses awal dalam diagram blok adalah *scanning*. Pengambilan data dilakukan dengan proses *scanning* terhadap objek benda. *Scanning* yaitu proses untuk pengalamatan lokasi yang spesifik pada sebuah objek. Pengalamatan tersebut menentukan letak piksel – piksel yang kemudian akan ditampilkan pada komputer.

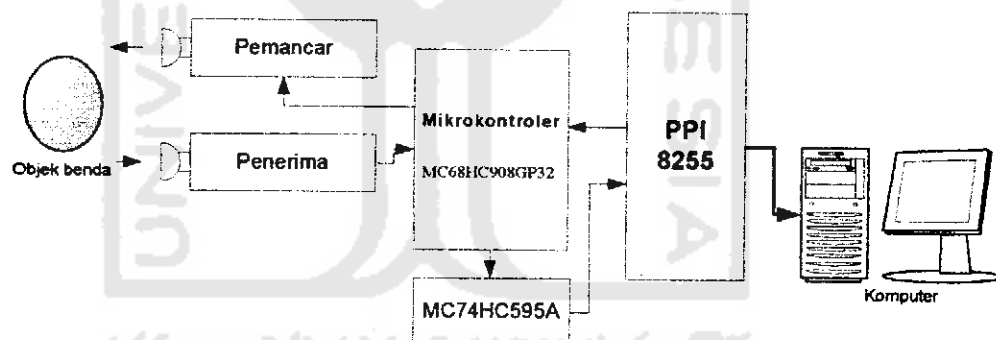
Bagian kedua dari proses adalah *programming*. *Programming* dibagi menjadi dua yaitu pemrograman pada mikrokontroler dan pemrograman untuk

tampilan visualisasi agar hasil akhir dari proses yaitu citra dapat ditampilkan dalam sebuah *display* yaitu layar monitor komputer.

### 3.1 Perancangan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah:

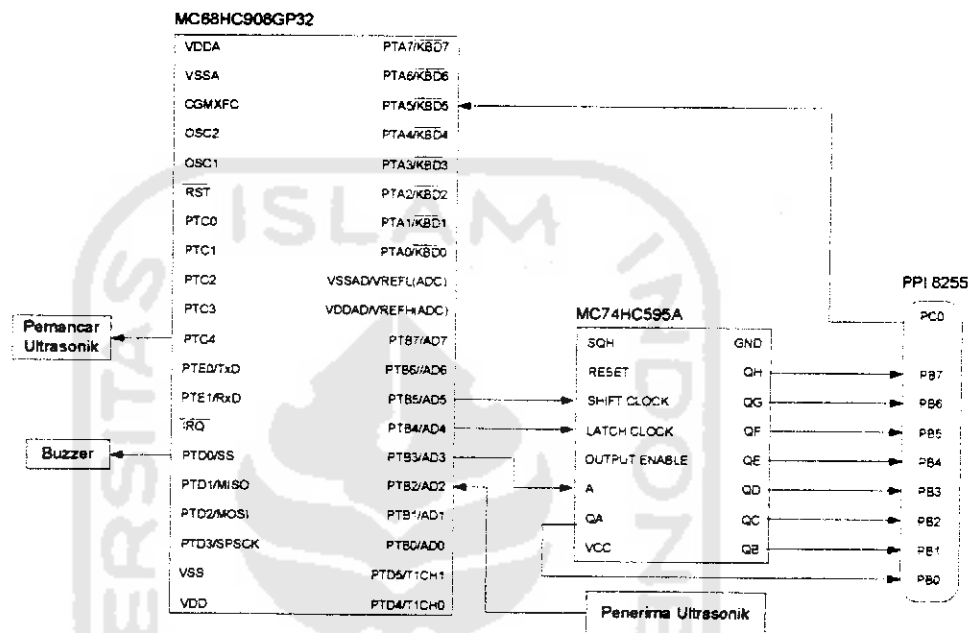
1. Pemancar dan penerima ultrasonik.
2. Mikrokontroler MC68HC908GP32.
3. MC74HC595A.
4. PPI 8255.
5. Satu unit komputer dengan prosesor Pentium II 333 MHz yang dilengkapi dengan perangkat lunak Borland C++ Builder versi 4.



Gambar 3.2 Diagram blok perangkat keras

Berdasarkan gambar 3.2, objek benda akan di-*scanning* oleh perangkat keras yang terdiri atas rangkaian pemancar dan penerima ultrasonik. Pemancar dan penerima tersebut memiliki sensor ultrasonik untuk pengiriman dan penerimaan sinyal. Sinyal pantul dari proses *scanning* akan diterima mikrokontroler sebagai

data. Data yang telah diproses akan dikirim ke komputer melalui *shift register* MC74HC595A dan antarmuka PPI 8255. Skema rangkaian MC68HC908GP32, MC74HC595A dan PPI 8255 ditunjukkan pada gambar 3.3.



Gambar 3.3 Skema rangkaian mikrokontroler, *shift register* dan antarmuka

Gambar 3.3 menunjukkan bahwa untuk mengaktifkan program atau memberi *keyboard interrupt* dari komputer diperlukan satu kaki port dari PPI 8255 yaitu port C0. Setelah terjadi *keyboard interrupt*, PTC4 dari mikrokontroler akan mengaktifkan pemancar ultrasonik. Jika pada PTB2 menerima masukan dari penerima ultrasonik atau dalam waktu yang sudah ditentukan sesuai jumlah *counter* maka *buzzer* akan berbunyi dan data dari perhitungan *counter* akan dikirim ke komputer melewati *shift register* MC74HC595A dan PPI 8255. Untuk menghubungkan *shift register* dibutuhkan tiga pin dari mikrokontroler yaitu pin

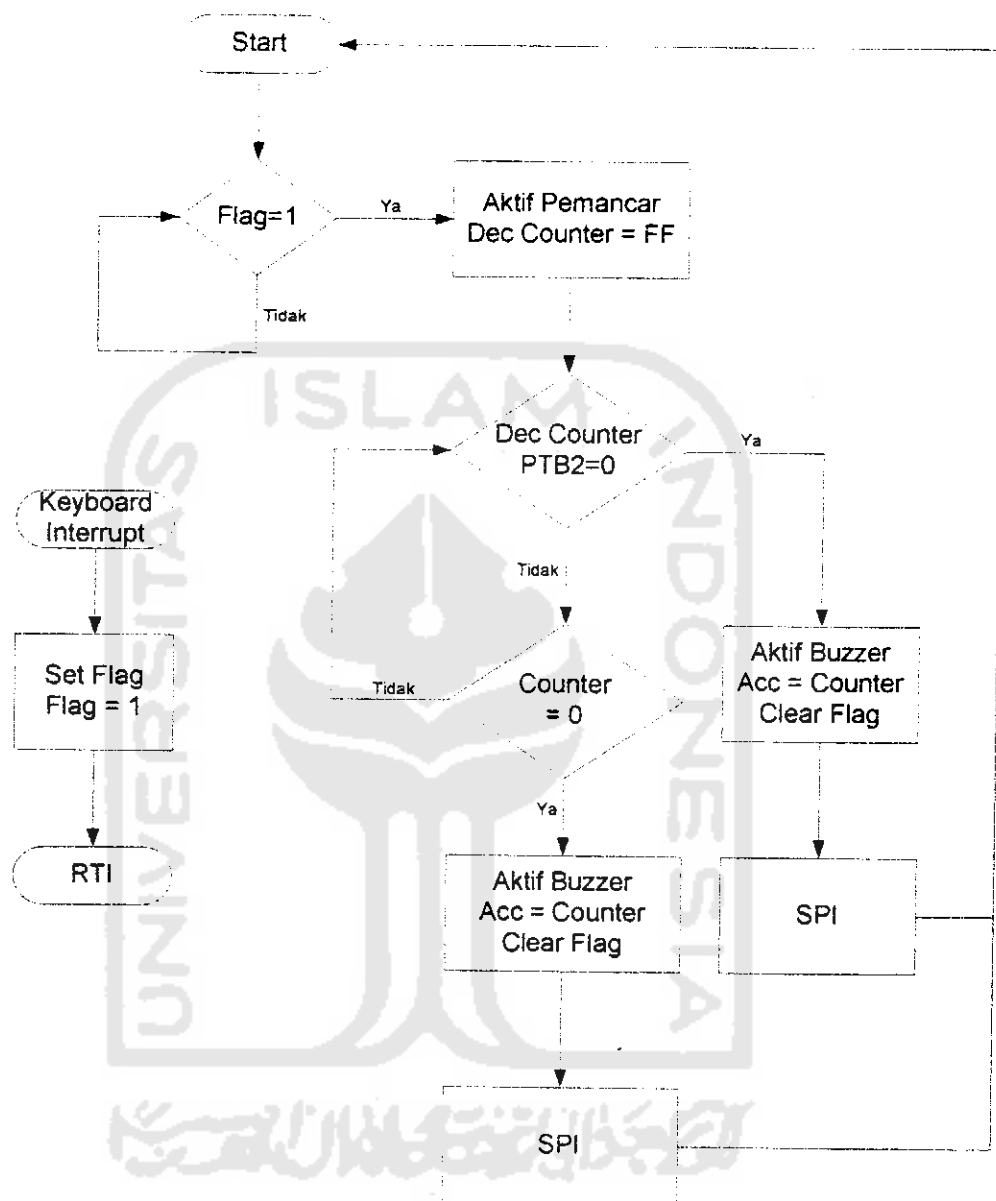
PTB3, PTB4 dan PTB5 yang masing - masing terhubung pada serial data input A, *latch clock*, dan *shift clock*. Data serial yang masuk pada MC74HC595A kemudian diubah menjadi data paralel 8 bit dan dikirimkan ke komputer melewati antarmuka PPI 8255 pada port B.

### **3.2 Perancangan Perangkat Lunak**

Pengoperasian perangkat keras sangat bergantung pada perangkat lunaknya. Ada dua macam perangkat lunak yang digunakan yaitu WinIDE\_GP32 dengan bahasa pemrograman assembly untuk pemrograman mikrokontroler dan Borland C++ Builder 4 dengan bahasa pemrograman C++ untuk pengaktifan PPI 8255 dan tampilan visualisasi citra pada komputer.

#### **3.2.1 Pemrograman Mikrokontroler MC68HC908GP32**

Pemrograman mikrokontroler MC68HC908GP32 menggunakan perangkat lunak WinIDE\_GP32 dengan bahasa pemrograman assembly. Mikrokontroler MC68HC908GP32 adalah alat yang berfungsi untuk mengatur sistem dan memproses data sebelum dikirim ke komputer. Fungsi mikrokontroler tersebut diantaranya mengaktifkan dan menonaktifkan pemancar ultrasonik, menghitung waktu pantul ultrasonik, mengaktifkan *buzzer* dan mengirimkan hasil proses ke dalam komputer. Diagram alir pemrograman ditunjukkan pada gambar 3.4.



Gambar 3.4 Diagram alir pemrograman mikrokontroler

Program mikrokontroler terdiri atas dua rutin utama yaitu rutin program utama dan rutin *keyboard interrupt*. Program utama akan selalu aktif untuk

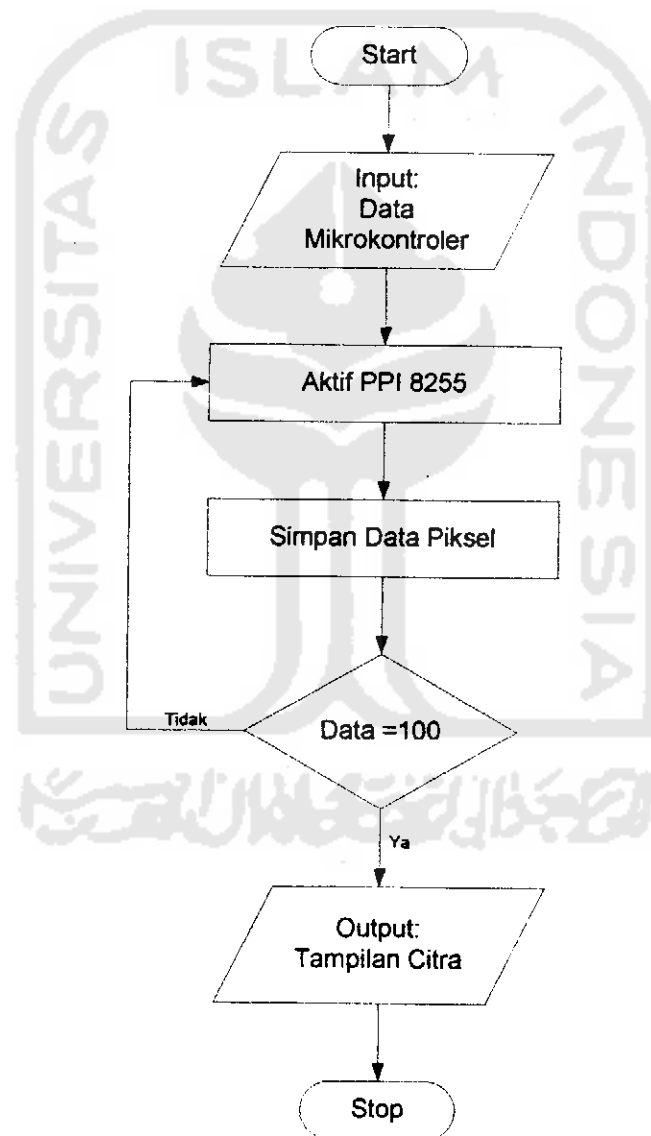
menunggu adanya *keyboard interrupt* dari komputer. *Interrupt* berfungsi untuk mengaktifkan program mikrokontroler selanjutnya yaitu mengaktifkan pemancar ultrasonik, mengaktifkan *counter* penghitung, mengaktifkan *buzzer*, mengirim data ke komputer melewati SPI (*Serial Peripheral Interface*) dan mematikan pemancar ultrasonik. Jika tidak ada *keyboard interrupt* maka program akan selalu tetap menunggu hingga terjadi adanya *keyboard interrupt*. Terjadinya *keyboard interrupt* ditandai dengan berubahnya bit Flag pada register menjadi 1.

Pengaktifan *counter* penghitung pada mikrokontroler berfungsi untuk mengukur seberapa lama gelombang ultrasonik diterima oleh penerima ultrasonik. *Counter* akan berhenti jika ada input pada PTB2 dari rangkaian penerima ultrasonik yang menandakan adanya pantulan. *Counter* akan mengalami penurunan nilai dari 255 sampai dengan 0 sesuai dengan lama waktu pantul. Hasil dari *counter* yang berhenti akan dikirim ke komputer melalui SPI. Jika tidak ada input pada PTB2, *counter* akan tetap berhenti jika pengurangan nilai sudah sampai 0. Hal ini menandakan bahwa tidak ada sinyal pantul yang diterima oleh rangkaian penerima ultrasonik. Nilai 0 tersebut juga akan dikirim ke komputer melalui SPI sebagai warna hitam.

*Counter* yang berhenti ditandai dengan bunyi *buzzer*. Satu kali bunyi *buzzer* menandakan pengurangan *counter* sudah mencapai nilai 0 atau tidak ada sinyal yang diterima oleh sensor penerima. Sedangkan dua kali bunyi *buzzer* menandakan bahwa ada sinyal pantul yang diterima oleh sensor penerima.

### 3.2.2 Pemrograman C++

Perangkat lunak yang digunakan yaitu Borland C++ Builder versi 4. Pemrograman dibagi dalam dua bagian utama. Bagian pertama adalah program pengaktifan PPI 8255 pada slot ISA dan bagian kedua adalah program penampilan citra 2 dimensi pada layar monitor komputer.



Gambar 3.5 Diagram alir pemrograman C++

Gambar 3.5 merupakan diagram alir program C++. Pemrograman dimulai dengan pengaktifan PPI 8255 sebagai antarmuka yang mendapatkan input data dari mikrokontroler. Data yang dikirim ke komputer merupakan data setiap piksel yang harus ditampilkan kembali dalam bentuk warna yang tersusun sesuai urutan piksel. Jika data piksel belum mencapai 100 maka akan dilakukan pengambilan data lagi sesuai perintah *user* sampai mencapai 100 data agar dapat masuk ke program selanjutnya yaitu penampilan citra. Penampilan citra pada layar monitor merupakan tahap akhir dari pemrograman C++.

### 3.2.2.1 Pemrograman PPI 8255

Pemrograman PPI 8255 dilakukan sebagai proses inialisasi port – port penghubung antara perangkat keras dengan komputer. Ada 3 hal pokok yang perlu diperhatikan yaitu penentuan alamat port, mode, dan pengalamatan *control word*.

Untuk modul PPI 8255 dibutuhkan empat buah alamat. Dalam perancangan ini alamat yang digunakan adalah:

Port A	= <i>base address</i>	280h = 1010000000
Port B	= <i>base address</i> +1	281h = 1010000001
Port C	= <i>base address</i> +2	282h = 1010000010
<i>Control Word</i>	= <i>base address</i> +3	283h = 1010000011

Penentuan mode / protokol komunikasi dipilih memakai mode 0 (protokol sederhana / *basic input - output*) karena hubungan perangkat keras hanya sebagai input atau output saja pada setiap portnya.



Hasil rancangan perangkat keras, dibutuhkan 9 bit (dari 24 bit) saluran input / output pada PPI 8255. Dalam perancangan ini Port B digunakan sebagai input data mikrokontroler 8 bit dan Port C digunakan sebagai output 1 bit untuk *keyboard interrupt* pada mikrokontroler. Dari perancangan letak port tersebut maka konfigurasi untuk *control word* PPI 8255 dipilih konfigurasi 9A (hexadesimal).

### 3.2.2.2 Pemrograman Penampilan Citra

Tahap akhir dari pemrograman yaitu penampilan citra agar citra tersebut dapat dianalisa. Tampilan citra dibuat pada sebuah *canvas* berukuran 10 x 10 piksel. Setiap piksel berukuran 200 dots. Susunan dari piksel – piksel tersebut ditunjukkan pada gambar 3.6.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Gambar 3.6 Susunan letak piksel

Proses *scanning* / penembakan ultrasonik dilakukan sesuai dengan urutan piksel pada gambar 3.6 yaitu dari arah kiri ke kanan dan dari atas sampai ke bawah.

Dalam setiap piksel mengandung 16 unsur piksel sebelumnya yang dikonvolusi. Hal ini dimaksudkan agar data dari setiap piksel akan lebih valid dan citra menjadi halus. Piksel – piksel yang berada pada posisi awal dimungkinkan datanya kurang valid karena hanya dikonvolusi kurang dari 16 piksel atau bahkan tidak sama sekali sehingga piksel – piksel tersebut sebenarnya dapat dihilangkan.

