BAB IV

PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas skema rangkaian dari sistem alat bantu baca tunanetra secara keseluruhanan, analisa perangkat keras, pengolahan data di perangkat lunak dan analisa perangkat lunak yang digunakan.

4.1 Pengujian Terhadap Perangkat Keras

Mengetahui dan mengukur tingkat keberhasilan suatu sistem yang dibuat dapat dilakukan pengujian terhadap alat yang dibuat. Adapun pengujian yang dilakukan untuk menguji alat yang dibantu adalah sebagai berikut:

4.1.1 Pengujian Kode Braille

Arus yang keluar dari mikrokontroler sangat kecil dan tidak bisa memicu *induktor* yang terdapat dalam *solenoid* sebesar 0.25A, sehingga untuk menguatkan arus dari mikrokontoler maka digunakan IC ULN2803 sebagai *driver solenoid*. Arus normal yang dihasilkan dari ULN2803 tiap *channel* sebesar 500 mA.

Hasil pengujian tersebut terhadap kinerja sistem kode braille dapat dilihat bahwa tegangan *solenoid* berbeda untuk setiap karakter yang diuji. Jika yang diuji berupa huruf maka pada *identifier* yang akan membedakan antara huruf kapital atau kecil. Sedangkan pengujian terhadap angka maupun tanda baca memiliki identifer yang sama dan untuk pengujian pada blok karakter, *solenoid* yang bekerja tergantung huruf, angka dan tanda baca yang diuji. Hasil pengujian dapat dilihat pada tabel 4.1 dan pengujian untuk semua karakter termasuk juga *identifier* dapat dilihat pada lampiran.

Tabel 4.1 Pengujian Kode Braille

NO	Karakter yang Diuji	Nomor Shaft yang Muncul		Tegangan tiap shaft dalam volt dengan beban											
				Identifier						Karakter					
	Karakter	*Identifier*	Karakter	$S0_1$	$S0_2$	$S0_3$	$S0_4$	$S0_5$	$S0_6$	$S0_1$	$S0_2$	$S0_3$	$S0_4$	$S0_5$	$S0_6$
1	A	6	1	3.2	3.2	3.2	3.2	3.2	0	0	1.5	1.5	1.5	1.5	1.5
2	B	6	1, 2	2.5	2.5	2.5	2.5	2.5	0	0	0	1.8	1.8	1.8	1.8
3	C	6	1, 4	2.6	2.6	2.6	2.6	2.6	0	0	1.8	1.8	0	1.8	1.8
4	D	6	1, 4, 5	2.7	2.7	2.7	2.7	2.7	0	0	2.0	2.0	0	0	2.0
5	E	6	1, 5	2.6	2.6	2.6	2.6	2.6	0	0	1.8	1.8	1.8	0	1.8

Pengujian terhadap huruf kapital maupun huruf kecil membutuhkan arus yang lebih besar jika dibandingkan dengan arus pada saat pengujian berupa angka maupun tanda baca, hal ini bisa terjadi karena *solenoid* pada *identifier* untuk menunjukan huruf kapital maupun kecil akan turun akan lebih banyak dibandingkan dengan angka maupun tanda baca, misalnya ketika masukannya adalah "A" maka *solenoid* yang turun adalah *solenoid* 1, 2, 3, 4 dan 6, sedangkan ketika masukanya berupa tanda baca "!" *solenoid* yang turun hanya *solenoid* 1 dan 2. Hal ini membuktikan bahwa *identifier* untuk huruf lebih banyak membutuhkan tegangan dibandingkan dengan tanda baca maupun angka.

Berdasarkan hasil pengujian yang dilakukan, dari 75 buah karakter yang diujikan tanpa beban/*solenoid* pada rangkaian *driver solenoid*, semuannya berhasil ditampilkan baik pada blok *identifier* sebagai pengenalan untuk jenis karakter dan pada blok karakternya. Maka perhitungan *persentase* keberhasilannya yaitu:

$$Persentase \text{ Keberhasilan} = \frac{\text{Karakter yag berhasil tampil}}{\text{Total karakter yang diuji}} * 100\% \text{.....(3.1)}$$

$$= \frac{75}{75} * 100\% \quad = 100\%$$

Pengujian untuk kode Braille dengan memasangkan *solenoid* pada *driver*. Pengujian ini dilakukan secara acak untuk huruf kapital lima huruf, huruf kecil lima huruf, angka diambil lima angka dan untuk tanda baca diambil lima tanda baca untuk

lebih jelas dapat lihat pada lampiran D. Sehingga jumlah karakter yang diujikan adalah 20 karakter. Maka perhitungan *persentase* keberhasilannya yaitu:

$$Persentase \text{ keberhasilan} = \frac{\text{Karakter yang berhasil tampil}}{\text{Total karakter yang diuji}} * 100\% \text{.....(3.2)}$$

$$= \frac{20}{20} * 100\% = 100\%$$

Selain pengujian untuk setiap karakter, pengujian juga dilakukan terhadap sebuah kata yang berbeda-beda sebanyak 10 kata. Kumpulan kata tersebut mewakili semua *alphabet* indonesia serta didalam terdapat kombinasi dari huruf kecil dan huruf besar untuk lebih jelasnya hasil pengujianya dapat dilihat pada lampiran D. Pertama pengujian dilakukan untuk tingkat keberhasilan kemunculan karakter-karakter yang terdapat disetiap kata yang ada dengan perhitungan *persentase* sebagai berikut:

$$Persentase \text{ Keberhasilan} = \frac{\text{Karakter yang berhasil tampil}}{\text{Jumlah karakter pada sebuah kata}} * 100\% \text{.....(3.3)}$$

$$= \frac{5}{5} * 100\% = 100\%$$

Sedangkan *persentase* keberhasilan muncul karater-karakter untuk setiap kata dapat dilihat pada lampiran D, *persentase* ini diperoleh dengan rumus yang pada persamaan 3.4. Merujuk ke hasil pengujian tersebut dapat dicari *persentase* keberhasilan untuk semua kata dengan rumusan sebagai berikut:

$$Persentase \text{ keberhasilan} = \frac{\sum persentase \text{ keberhasilan}}{\sum \text{kata yang diuji}} \% \text{.....(3.4)}$$

$$= \frac{100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\%}{10}$$

$$= \frac{1000\%}{10} = 100\%$$

Pengujian dilakukan dengan dengan dua cara, yaitu melakukan pengujian terhadap tegangan yang keluar tanpa beban (*solenoid*) dan pengujian dengan menggunakan beban (*solenoid*). Hasil kedua pengujian ini memiliki perbedaan pada tegangan yang keluar dari setiap *channel driver* dan pengukuran tegangan pada saat

solenoid dipasang pada *driver solenoid*. Untuk memperjelasnya pernyataan tersebut, misal pengujian untuk huruf "A" maka *solenoid* yang turun, maka *solenoid* yang membutuhkan tegangan yaitu *solenoid* dengan urutan nomor 2,3,4,5,6. Pada *channel driver solenoid* ke-2 mengeluarkan tegangan 6,4 volt dan tegangan pada *solenoid* sebesar 1.5 Volt, dari pengukuran tersebut dapat diambil selisih antara pengukuran tersebut dengan cara, yaitu:

$$\text{Selisih} = V_{\text{Tanpa Beban}} - V_{\text{Beban}} \quad (3.5)$$

$$= 6.4 - 1.5 = 4.9 \text{ Volt}$$

Perbedaan pada kedua pengukuran tersebut menghasilkan selisih tegangan, dari hasil selisih ini, pada saat *solenoid* bekerja sangat membutuhkan tegangan untuk menggerakan kumparannya.

4.1.2 Pengujian Terhadap Perangkat Lunak

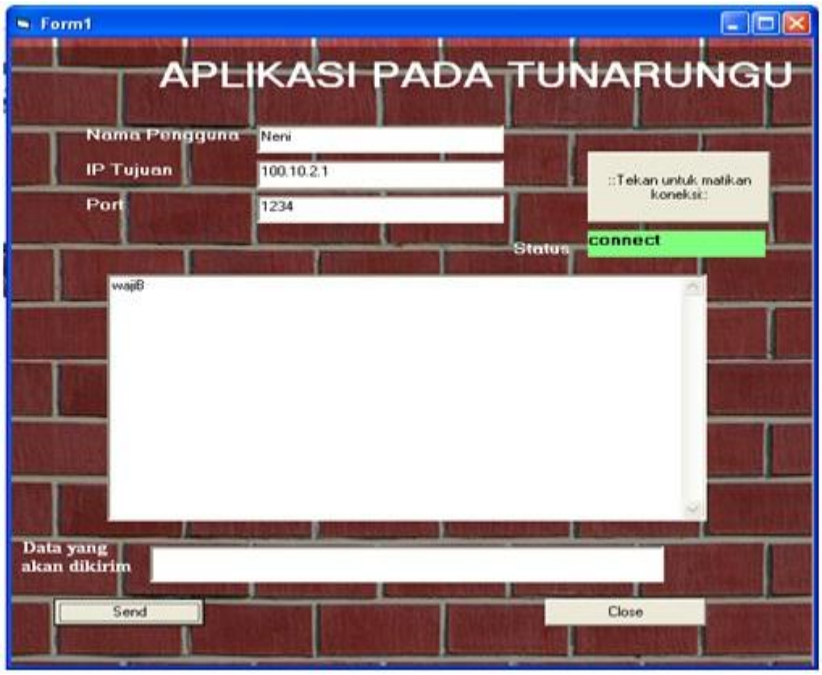
Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan dan kekurangan dari perangkat lunak yang dibuat. Pengujian perangkat lunak dilakukan pada *visual basic* dengan perancangan yang telah dibuat pada bab perancangan.

4.1.2.1 Pengujian pada Aplikasi Tunarungu

Pengujian ini dilakukan dengan cara menjalankan terlebih dahulu program yang telah dibuat pada *visual basic*. Setelah program pada tunarungu dijalankan, maka tunarungu harus menekan tombol "Tekan untuk koneksi", sehingga pada status tampil "connect" pada tampilan *visual basic*, maka komputer tunarungu dan komputer tunanetra telah terhubung secara LAN. Maka kedua aplikasi dapat berkomunikasi dan dapat menjadi media pengajaran.

Pengujian aplikasi tunarungu tidak itu saja. Pengujian dilakukan saat tunarungu akan mengirimkan datanya berupa karakter atau sebuah kata. Pengujian ini dilakukan pada saat tunarungu akan mengirimkan sebuah kata, yaitu "wajiB" ke tunanetra. Pertama-tama tunarungu mengetikan sebuah kata pada kolom pengiriman, kemudian tekan tombol "send" untuk mengirimkan data tersebut ke tunanetra, selain

pengirimaan data tersebut ke tunanetra, data tersebut akan masuk ke kolom percakapan. Pengujian terhadap aplikasi pada tunarungu dapat dilihat pada gambar 4.1



Gambar 4.1 Pengujian Aplikasi Tunarungu

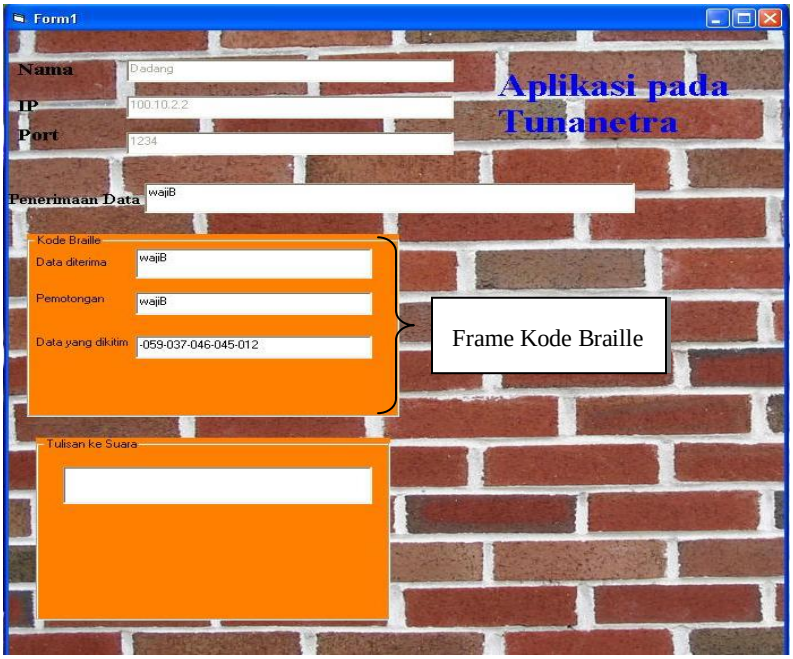
4.1.2.2 Pengujian pada Aplikasi Tunanetra Menerima Data

Setelah pengujian pada tunarungu telah selesai maka pengujian selanjutnya yaitu pengujian pada aplikasi pada tunanetra.

Pengujian pada aplikasi tunanerta, yaitu menerima respon “connect”, karena aplikasi tunanetra pada saat dijalankan hanya menunggu komputer tunarungu menekan tombol “connect”, selain itu yaitu aplikasi tunanetra telah menerima data berupa kata “wajiB”, maka data tersebut dapat dilihat pada kolom peneriman, maka respon yang harus dilakukan oleh tunanetra yaitu membaca dengan cara mesentuh kode braille yang telah dibuat atau mendengar apa yang kirim oleh tunarungu.

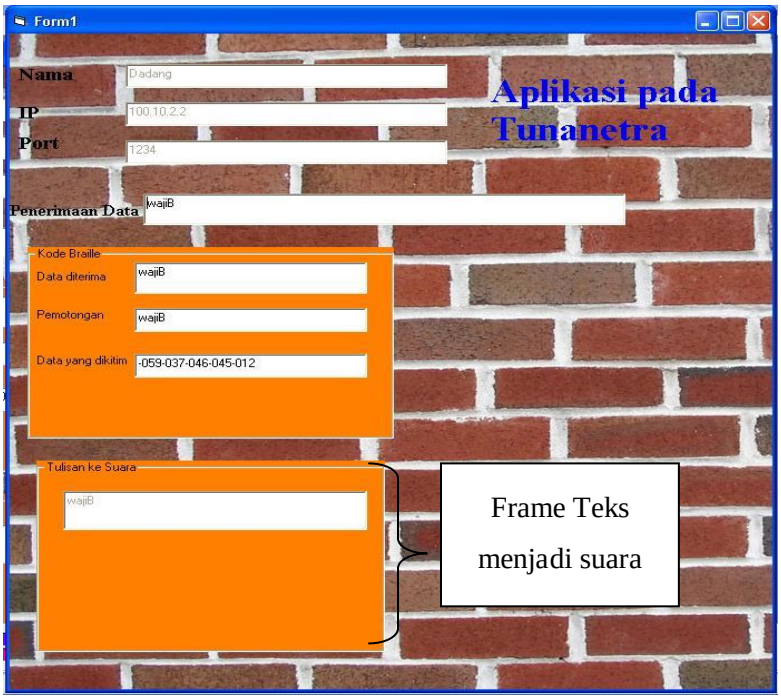
Jika tunanetra akan membaca kata “wajiB” maka tunanetra harus menekan *push button* yang kiri dan pada aplikasi *visual basic* akan muncul *frame* braille seperti

pada gambar 4.2. data yang dikirim berupa kata “wajiB” akan muncul pada kolom dengan label data diterima kemudian data tersebut akan dipecah atau dipotong menjadi sebuah *array*, karakter-karakter yang telah menjadi *array* akan masuk ke kolom pemotongan sesuai dengan kata yang diterima yaitu “wajiB”, lalu pada kolom selanjutnya yaitu kolom data yang dikirim, arti dari data yang dikirim yaitu karakter-karakter pada kata “wajiB” diberi sebuah nilai pada program *visual basic* dan nilai tersebut yang akan dikirim ke mikrokontroler untuk dibandingkan dan diberi nilai kode braille untuk dicetak pada *solenoid* sehingga dapat disentuh oleh tunanetra.



Gambar 4.2 Pengujian Aplikasi Tunanetra untuk Kode Braille

Jika tunanetra ingin berupa suara dari kata “wajiB” yang dikirim oleh tunanetra maka tunanetra harus menekan “enter’ pada keyboard untuk mengaktifkan program *konversi* tulisan ke suara pada *visual basic*. Untuk pengujian dapat dilihat pada gambar 4.3



Gambar 4.3 Pengujian Terhadap Aplikasi Tunanetra untuk Tulisan ke Suara

Pengujian tehadap aplikasi pada tunanetra tahaptulisan menjadi suara diuji dengan masukan tulisan dalam Bahasa Indonesia dan dalam Bahasa Inggris dan kata-kata dasar dipilih secara acak sebanyak 35 kata. Sebagai contoh hasil pengujian dapat dilihat pada tabel 4.1 dan untuk hasil pengujian yang lebih lengkap dapat dilihat pada lampiran D tabel 4.

Tabel 4.2 Tabel Pengujian Aplikasi Tulisan ke Suara

No	Bahasa Indonesia		Bahasa Inggris	
	Kata	Keterangan	Kata	Keterangan
1	Rumah	Gagal	Eat	Berhasil
2	Kayu	Berhasil	Pin	Berhasil
3	Makan	Berhasil	Book	Berhasil
4	Sisir	Berhasil	Leg	Berhasil
5	Layar	Gagal	Down	Berhasil

Dari tabel 4.1 dapat diketahui *persentase* untuk tingkat keberhasilan ketika masukan terhadap tunanetra berupa kata dalam Bahasa Indonesia dan dalam Bahasa Inggris. Hasil *persentase*-nya dapat dilihat sebagai berikut:

Persentase keberhasilan untuk masukan Bahasa Indonesia

$$Persentase \text{ Keberhasilan} = \frac{\text{Kata yang Berhasil}}{\text{Seluruh Kata yang Diuji}} * 100\% \quad \text{.....(3.6)}$$

$$= \frac{21}{35} * 100 = 60\%$$

Karena tingkat keberhasilan untuk masukan Bahasa Indonesia kurang dari 100%, maka dengan masukan Bahasa Indonesia mempunyai *persentase error* untuk 14 kata dasar yang dipilih secara acak.

$$Persentase\ Error = \frac{\text{Kata yang Gagal}}{\text{Seluruh Kata yang Diuji}} * 100\% \quad \text{.....(3.7)}$$

$$= \frac{14}{35} * 100\% = 40\%$$

Tingkat keberhasilan untuk masukan Bahasa Inggris berbeda dengan Bahasa Indonesia. Dengan hasil *persentase* pengujian sebagai berikut

$$Persentase \text{ Keberhasilan} = \frac{\text{Kata yang Berhasil}}{\text{Seluruh Kata yang Diuji}} * 100\% \quad \text{.....(3.8)}$$

$$= \frac{35}{35} * 100\% = 100\%$$

4.1.2.3 Pengujian pada Aplikasi Tunanetra Mengirim Data

Pada saat tunanetra akan mengirim data menggunakan *speech recognitiom* dengan memanfaatkan aplikasi SAPI 5.1 diuji dengan 2 tahapan yaitu pengujian pengenalan suara dengan pelatihan dan pengenalan suara tidak pelatihan dengan jumlah data berupa kata sebanyak 35 kata. Hasil pengujian untuk pengenalan suara dapat dilihat pada lampiran D.

1. Pengujian Pengenalan Suara dengan Tanpa Pelatihan

Pengujian pada tahap ini dilakukan terhadap 2 pria dan 2 wanita yang terlebih dahulu melakukan pelatihan sehingga mesin dapat mengenali pola suara yang melakukan pengenalan pola suara dan hasil pengujian sebagai berikut:

Hasil pengujian pengenalan suara untuk pria ke-1 dengan 35 kata dengan hasil pengujian 15 kata tulisannya muncul pada aplikasi dan 20 kata yang gagal muncul dan tingkat keberhasilan dalam *persentase*nya, yaitu:

$$Persentase \text{ Keberhasilan} = \frac{\text{Jumlah kata yang dikenal i}}{\text{Total kata}} * 100\% \text{ ……..(3.9)}$$

$$= \frac{15}{35} * 100\% = 42.86\%$$

Pada pengujian pria Ke-2 kata yang muncul sebanyak 15 kata dan gagal muncul 20 kata dengan perhitungan *persentase* keberhasilan menggunakan persamaan 3.9 dan hasil perhitungan *persentase* keberhasilan sebesar 42.86%

Persentase keberhasilan untuk wanita ke-1 sebesar 40% menggunakan persamaan 3.9 dengan hasil pengujian yaitu, kata yang muncul sebanyak 14 kata dan 21 kata yang gagal muncul pada aplikasi tunanetra.

Wanita ke-2 yang melakukan pengujian diktasi tanpa pengenalan. Sistem hanya mengenali 17 kata dan 18 kata gagal dikenali dengan *persentase* keberhasilan 48.57%. Hasil *persentase* ini didapat menggunakan persamaan 3.9.

Sehingga dari empat pengujian yang dilakukan, didapatkan rata-rata *persentase* tingkat keberhasilan secara keseluruhan adalah sebesar 43.57% yang didapat dari hasil perhitungan menggunakan persamaan 3.10, untuk hasil pengujian yang lebih lengkap dapat dilihat pada lampiran D pada tabel 5.

$$\text{Rata} - \text{Rata} = \frac{\sum_{i=1}^{n} \text{akurasi}}{\text{Jumlah penguj ian yang dilakukan}} \text{ ……..(3.10)}$$

$$= \frac{42.86\% + 42.86\% + 40\% + 48.57\%}{4}$$

$$= 43.57\%$$

2. Pengujian Pengenalan Suara dengan Pelatihan

Pengujian pada tahap ini dilakukan oleh 2 pria dan 2 wanita yang melakukan pelatihan, pengujian ini dilakukan untuk mengetahui keakuratan sistem dalam mengenali suara dan hasil pengujiannya sebagai berikut:

Hasil pengujian pengenalan suara untuk pria ke-1 dengan 35 kata dengan hasil pengujian 31 kata tulisannya muncul pada aplikasi dan 4 kata yang gagal muncul dan tingkat keberhasilan dalam *persentase*nya,yaitu:

$$Persentase \text{ Keberhasilan} = \frac{\text{Jumlah kata yang dikenali}}{\text{Total kata}} * 100\% \text{ ……..(3.11)}$$

$$= \frac{31}{35} * 100\% = 88.57\%$$

Pada pengujian pria Ke-2 kata yang muncul sebanyak 28 kata dan gagal muncul 7 kata dengan perhitungan *persentase* keberhasilan menggunakan persamaan 3.9 dan hasil perhitungan *persentase* keberhasilan sebesar 80%

Persentase keberhasilan untuk wanita ke-1 sebesar 71.43% menggunakan persamaan 3.9 dengan hasil pengujian yaitu, kata yang muncul sebanyak 25 kata dan 10 kata yang gagal muncul pada aplikasi tunanetra.

Wanita ke-2 yang melakukan pengujian diktasi tanpa pengenalan. Sistem hanya mengenali 24 kata dan 11 kata gagal dikenali dengan *persentase* keberhasilan 68.57% hasil *persentase* ini didapat menggunakan persamaan 3.9.

Sehingga dari empat pengujian yang dilakukan, didapatkan rata-rata *persentase* tingkat keberhasilan secara keseluruhan adalah sebesar 77.14% yang didapat dari hasil perhitungan menggunakan persamaan 3.12, untuk hasil pengujian yang lebih lengkap dapat dilihat pada lampiran D pada tabel 6.

$$\text{Rata} - \text{Rata} = \frac{\sum_{i=1}^{n} \text{akurasi}}{\text{Jumlah pengujian yang dilakukan}} \text{ ……..(3.12)}$$

$$= \frac{88.57 + 80\% + 71.43\% + 68.57\%}{4}$$

$$= 77.14\%$$

4.2 Analisa Perangkat Lunak

Perangkat lunak sangat penting unyuk mengendalikan sebuah alat bantu komunikasi antara tunanetra dan tunarungu. Perangkat lunak yang akan akan dianalisa meliputi program pada *visual basic*, program pada mikrokontroler *basic stamp*, program *konversi* teks ke suara dan *konversi* suara ke teks.

4.2.1 Analisa Program pada Mikrokontroler Basic Stamp

Program utama dari alat bantu komunikasi antara tunanerta dan tunarungu terdiri *inisialisasi serial*, prosedur pengecekan masukan pada mikrokontroler dan meng-*konversi*-kan menjadi kode braille dengan cara membandingkan setiap karakter dengan data yang diterima pada mikrokontroler, prosedur pengecekan *identifier* dengan cara membandingkan setiap karakter yag diterima dan prosedur pengiriman data *identifier* dan karakter ke alat bantu dalam bentuk kode Braille dengan nilai heksadesimal. Program pada mikrokontroler dapat dilihat pada lampiran C.

4.2.2 Analisa Perangkat Lunak untuk Komunikasi Serial

Dibawah ini merupkan penggalan program untuk menggunakan komunikasi *serial* pada mikrokontroler.

```
SERIN RX,240,300,lanjut ,[DEC3 inserial]

SEROUT TX,240,["1"]
```

Pada penggalan program diatas terdapat perintah “SERIN”.Perintah tersebut digunakan untuk menerima data dari *visual basic* ke mikrokontroler melalui *pin* “RX” dengan baudrate 240 pada mikrokontroler BS2P40 sebagai *inisialisasi* dari 9600 baudrate pada komputer selama 300 siklus, lompat ke label “lanjut” dan data yang diterima dirubah ke desimal sebanyak 3 digit. Sedangkan untuk “SEROUT” kirim data serial (“1”) ke visual basic melalui *pin* “TX” dengan baudrate 240.

4.2.3 Analisa Perangkat Lunak untuk Konversi ke Kode Braille

Penggalan program dibawah ini digunakan untuk mengcek data yang diterima dan membandingkan serta memberikan kode braille sebuah karakter.

```
periksa:
IF (inserial = 75) THEN
        biner = $39
ENDIF
GOSUB cetak
RETURN
```

Penggalan program ini akan mengerjakan, pengecekan terhadap data yang diberi dengan program yang telah dibuat, setelah dicek dan sesuai maka diberikan nilai Braille untuk data yang diterima dan akan ke proses berikutnya yaitu ke prosedur "cetak" untuk menampilkan kode braille tersebut ke *solenoid* melalui *driver solenoid*.

4.2.4 Analisa Perangkat Lunak untuk Pengecekan *Identifier*

Program ini sama halnya seperti *konversi* kode braille, yaitu membandingkan dengan kode yang ada pada mikrokontoler memberikan kode braille dan mengirimkan kode braille tersebut ke *driver solenoid* untuk ditampilkan ke *solenoid*. Berikut penggalan programnya:

```
identifier:
IF (inserial >=63) AND (inserial <76)  THEN
      biner = $00
ENDIF
GOSUB cetak1
RETURN
```

Program diatas digunakan untuk pengecekan *identifier* tanda baca. Alur penggalan program ini adalah program akan membandingkan kode data yang ada pada mikrokonroler dengan kode data yang dikirim dari komputer, diberikan kode braille dalam bentuk bilangan heksadesimal terakhir memanggil prosedur "cetak1" untuk memberikan kode braille yang telah didapat ke *driver solenoid*.

4.2.5 Analisa Perangkat Lunak untuk Pemicu Driver Solenoid

Program pemicu *driver solenoid* sangat dibutuhkan karena untuk mencetak kode braille dari karakter yang diterima dari komputer ke *solenoid* melalui *driver solenoid*.

```
cetak:
charpin = 5
FOR i= 0 TO 5
        temp = biner & 1
        IF temp = 1 THEN
          LOW charpin
        ELSE
          HIGH charpin
        ENDIF
biner = biner >> 1
charpin = charpin + 2
NEXT
RETURN
```

Program diawali dengan *inisialisasi pin* awal pada mikrokontoler dengan nama variabel “charpin”, karena yang digunakan yaitu kode braille dengan 6 *dot*. Sehingga pengulangan untuk kode braille yaitu sebanyak 6 kali. Kode braille yang telah ada di “AND” dengan 1 agar mendapatkan nilai “temp’ untuk dibandingkan. Jika nilai “temp” bernilai “1” maka *pin* tersebut akan bernilai *low* atau “0”, jika bernilai 0 maka *pin* tersebut bernilai satu. Setelah pemberian logika selesai maka *bit biner* bergeser ke kiri dan mengulang sebanyak 6 kali, sesuai dengan jumlah *dot* yang digunakan pada kode braille.