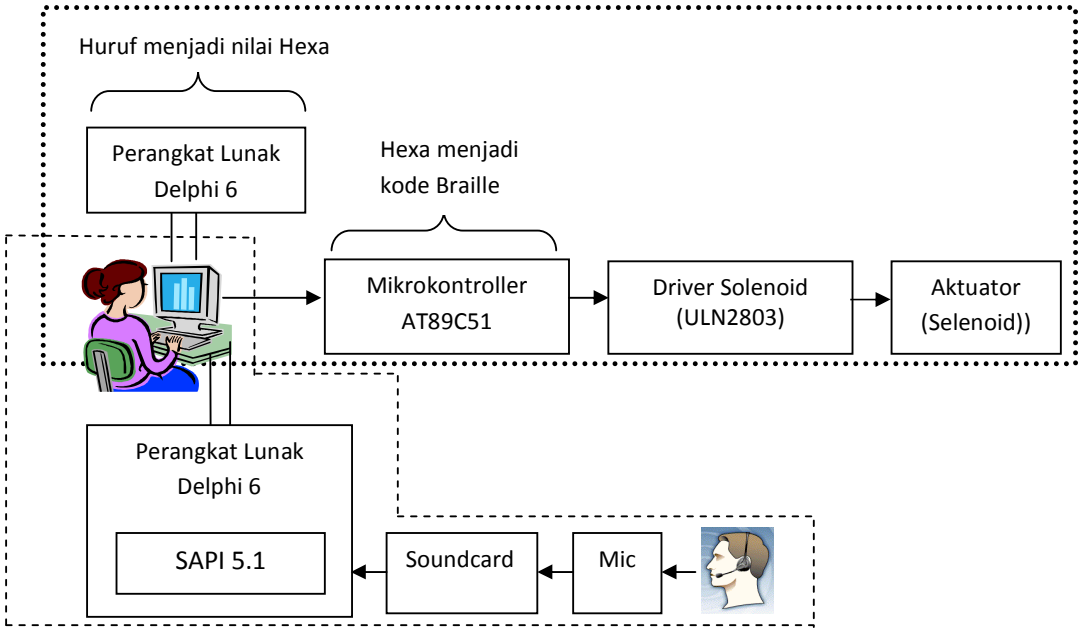
BAB III
PERANCANGAN

Pada bab perancangan sistem ini akan membahas tentang prinsip kerja sistem, perancangan perangkat lunak (*Delphi*), perancangan perangkat keras, algoritma program *assembler* dan algoritma program *delphi.*

3.1 Prinsip Kerja Sistem

Objektif dari alat bantu komunikasi tuna netra dan tuna rungu adalah membuat sebuah alat bantu baca yang dapat mengkonversi karakter ASCII menjadi kode *braille*, yang pada akhirnya kode *braille* tersebut dapat ditampilkan pada alat yang dibuat dan membuat suara per-karakter yang diucapkan diubah menjadi bentuk tulisan pada aplikasi yang telah dibuat. Adapun sistem yang dirancang diperlihatkan pada gambar 3.1.



Gambar 3.1. Perancangan Sistem Alat Bantu Komunikasi Tuna Netra dan Tuna Rungu

Pada saat akan menjalankan alat bantu komunkasi tuna netra dan tuna rungu dengan menggunakan kode braille dan pengenalan suara per-karakter, maka komputer harus dinyalakan terlebih dahulu lalu memasang penghubung untuk koneksi antara komputer dengan alat tersebut dan memasang *microphone.* Setelah semuanya terpasang dengan baik, maka alat tersebut sudah dapat dijalankan dan proses pengenalan suara per-karakter sudah dapat berjalan. Untuk penyandang tuna rungu dapat mengetikkan di komputer karakter dengan menggunakan *software* yang ada, karakter atau kalimat tersebut dikonversi menjadi bilangan *hexa* ASCII. Setelah menjadi bilangan *hexa* maka karakter itu siap dikirimkan ke alat (mikrokontroler AT89C51) melalui *port* serial sebagai antarmuka. Di dalam mikrokontroler bilangan *hexa* ASCII tersebut dikonversi lagi menjadi kode *braille*. Kode *braille* yang didapatkan dari hasil konversi di dalam mikrokontroler tersebut berupa bilangan *hexa* (kode *braille*), yang kemudian digunakan mikrokontroler sebagai V_{IN} untuk mengatur *solenoid* melalui *driver* solenoid IC ULN2803 untuk menaikkan tombol maupun menurunkan tombol. Tombol yang naik tersebut yang digunakan oleh tuna netra dalam membaca karakter. Untuk penyandang tuna netra dapat mengucapkan kata per-karakter yang akan dikenali oleh software yang telah dibuat dan hasilnya akan ditampilkan pada aplikasi.

3.2. Perancangan Perangkat Keras

Alat bantu komunikasi tuna netra dan tuna rungu dengan menggunakan kode braille yang dirancang terdiri dari subsistem utama perangkat keras, yaitu sistem mekanik, mikrokontroler, *solenoid*, *driver*, *power supply*, *port* serial. Gambar rangkaian alat bantu kode Braille dapat dilihat pada lampiran A.

3.2.1. Mekanik Alat Bantu Tuna Netra

Desain mekanik merupakan salah satu faktor penting dalam membangun alat bantu baca tuna netra. Di samping berfungsi untuk menyatukan keseluruhan subsistem dalam satu kesatuan yang utuh, juga akan memberikan kesan estetis dan kenyamanan bagi pengguna alat bantu baca tersebut.

Sistem mekanik dari alat bantu baca tuna netra terdiri dari: *shaf* dan tombol. *Shaf* digunakan untuk menaikkan ataupun menurunkan tombol ketika *solenoid* bekerja, sedangkan tombol itu sendiri digunakan untuk

menampilkan karakter dalam kode *braille* agar tuna netra dapat membaca karakter ASCII. Penyangga bawah digunakan untuk mengunci *solenoid*, perangkat elektronik dan *power supply* sedangkan penyangga atas digunakan sebagai pembatas tombol yang naik atau turun ketika *solenoid* bekerja.

3.2.2. Mikrokontroler

Jenis mikrokontroler yang digunakan adalah AT89C51 keluarga dari MCS-51 keluaran Atmel. Pada pembuatan alat bantu baca untuk tuna netra, mikrokontroler digunakan untuk mengimplementasikan pengaturan *solenoid*, mengkonversi bilangan hexa ASCII menjadi bilangan hexa *braille* dan mengecek tipe karakter yang dikirimkan dari komputer. Berikut ini pin-pin pada port yang digunakan:

a. Port P0

   Port P0.0 sampai dengan P0.5 (pin 39 sampai dengan pin 34) digunakan untuk mengirimkan output ke IC ULN2803 yang hasilnya akan membentuk jenis karakter yang telah dikirmkan oleh komputer.

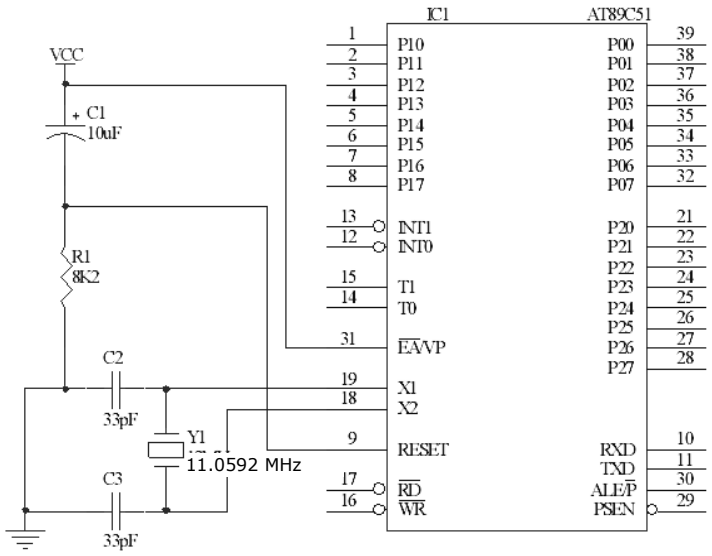
b. Port P2

   Port P2.0 sampai dengan P2.5 (pin 21 sampai dengan pin 26) digunakan untuk mengirimkan output ke IC ULN2803 yang hasilnya akan membentuk jenis karakter yang telah dikirmkan oleh komputer.

c. Port P3

   Port P3.0 dan Port Port P3.1 digunakan untuk menerima data dari IC MAX232

Pin-pin di atas digunakan sebagai pin utama dalam perancangan, selain itu masih terdapat beberapa pin lainnya yang digunakan untuk keperluan khusus, misalnya pin untuk osilator, EA/VP, VCC, GND dan untuk *reset* seperti terlihat pada gambar 3.3.



Gambar 3.2. Rangkaian Minimum AT89C51

Pin EA/VP dihubungkan dengan VCC berfungsi untuk menjalankan program yang ada di memori internal dari AT89C51. Keterangan mengenai bagian osilator dan *reset* adalah sebagai berikut :

a. Osilator

Bagian ini merupakan subsistem dari mikrokontroler yang berfungsi untuk membangkitkan *clock* pada mikrokontroler. *Clock* diperlukan oleh mikrokontroler untuk mensinkronkan proses yang sedang berlangsung dalam mikrokontroler. Rangkaian ini tersusun atas komponen kristal dan dua buah kapasitor. Frekuensi kristal yang biasa digunakan antara 6 sampai 24 MHz. Pada pembuatan alat bantu baca tuna netra ini digunakan kristal 11.0592 Mhz agar mempermudah perhitungan *baudrate* komunikasi serial. Adapun kapasitor yang dapat dipergunakan berdasarkan *data sheet* antara 20pF sampai 40pF untuk osilator yang menggunakan kristal. Kapasitor yang digunakan adalah jenis *monocap* bernilai 33 pF. Kapasitor jenis ini dipilih karena cocok untuk aplikasi frekuensi tinggi. Fungsi kapasitor adalah untuk menstabilkan osilasi yang dihasilkan oleh kristal. Penempatan kristal dan kapasitor diusahakan dekat dengan bagian mikrokontroler untuk menghindari kemungkinan terjadinya *noise*.

b. Reset

Rangkaian *reset* dibuat untuk me-*reset* sistem sehingga proses bisa dijalankan mulai dari awal program. Untuk membuat rangkaian *reset* ada hal-hal yang perlu diperhatikan:

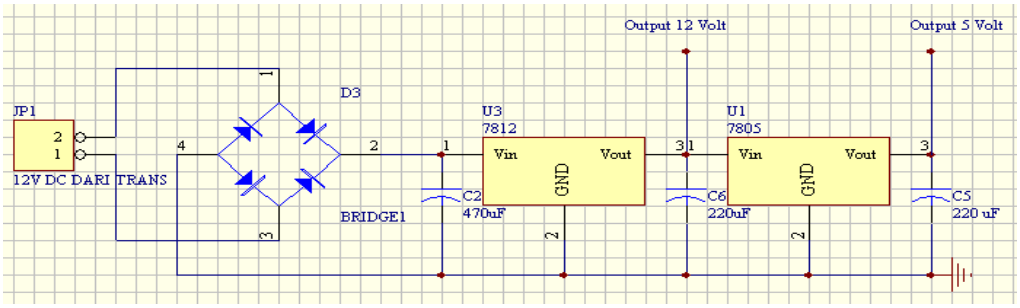
- Lebar pulsa untuk sinyal *reset* harus lebih besar dari 24 *periode* osilasi atau dua kali *machine cycle.*
- Waktu *start up* dari osilator tidak lebih dari 10 ms.
- Tegangan *reset* yang diperlukan adalah lebih dari atau sama dengan 3,5 volt.

3.2.3. Solenoid dan Driver

Solenoid merupakan komponen penting dari sistem yang membangun alat bantu baca tunanetra berbasis jaringan. Untuk menggerakan solenoid tersebut diperlukan sebuah *driver*. *Driver* diperlukan karena arus yang keluar dari mikrokontroler kecil, sehingga perlu dikuatkan. Rangkaian ini menggunakan dua buah *driver* yang masing-masing d*river* digunakan untuk menggerakkan enam buah solenoid. *Driver* yang digunakan dalam pembuatan sistem alat bantu baca ini adalah IC ULN2803 dan arus keluaran pada setiap *chanel*-nya sebesar 500mA. Pada gambar di lampiran A diperlihatkan suatu rangkaian antarmuka enam buah *Solenoid* dengan *driver* ULN2803.

3.2.4. Power Supply

Bagian *power supply* merupakan bagian penting bagi semua rangkaian. Tegangan yang dibutuhkan untuk rangkaian adalah 5 V DC untuk mikrokontroler dan IC ULN2803, sementara *Solenoid* membutuhkan tegangan 9 V sampai 15 V.

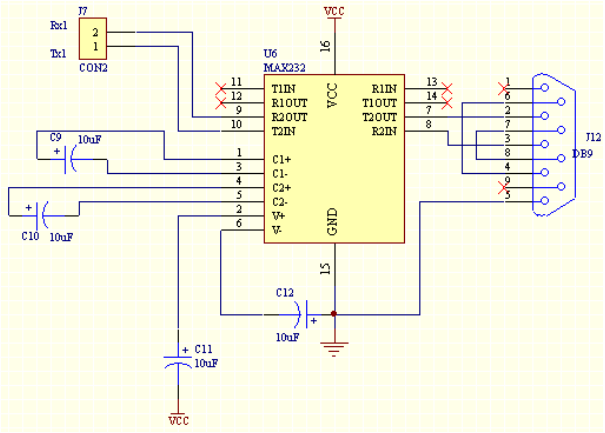


Gambar 3.3. Rangkaian *Power Supply*

Pada gambar 3.4 digunakan *transformator* sebagai sumber tegangan AC dengan arus 2A. Untuk diubah menjadi tegangan DC yang dibutuhkan, digunakan penyearah gelombang penuh dioda *bridge*. Untuk menghasilkan tegangan 5 Volt dan 12 Volt digunakan IC regulator LM7805 dan LM78012 untuk meregulasi tegangan mendekati 5 Volt untuk LM7805 dan 15 Volt untuk LM78012. Sebagai *indikator* rangkaian ini dilengkapi dengan LED. Input untuk IC *regulator* yang digunakan adalah 12 Volt AC dari *transformator*.

3.2.5. Port Serial

Pemanfaatan *port* serial sebagai jalur komunikasi antara alat bantu baca tuna netra dengan komputer karena tidak diperlukan banyak kabel untuk transmisi dibanding dengan *port* paralel. Selain itu pembuatan program juga lebih sederhana dan pengkabelannya lebih panjang. Dibawah ini merupakan gambar rangkaian RS 232.



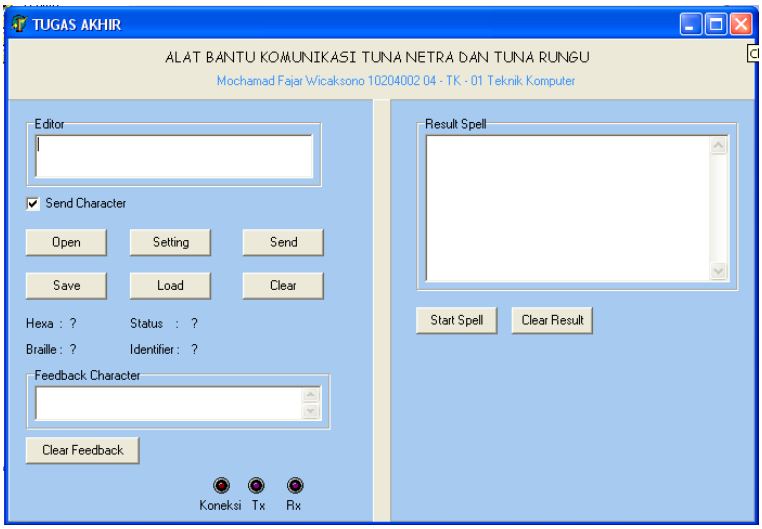
Gambar 3.4. Rangkaian *RS 232*

Pada skema diatas menggunakan IC MAX232 untuk komunikasi serial antara PC dan mikrokontroler. Pin-pin yang digunakan untuk pengiriman dan penerimaan data dalam perancangan alat ini adalah pin 8 (R2IN)untuk menerima data dari PC yang selanjutnya akan dikirimkan ke mikrokontroler malalui pin 9 (R2OUT) dan pin 10 (T2IN) digunakan untuk menerima data dari mikrokontroler yang selanjutnya akan dikirimkan ke PC melalui pin 7 (T2OUT). Rentang tegangan yang

dibutuhkan IC MAX232 agar dapat bekerja adalah dari 4,5V sampai dengan 5V.

3.3. Perancangan Perangkat Lunak (*Delphi*)

Perangkat lunak dari alat Bantu baca tunanetra dibuat menggunakan *Delphi* 6, dengan tampilan seperti gambar 3.2. Untuk pengantarmukaan *software* dengan alat maka digunakan program *Delphi* yang aturan penulisannya didasari dengan aturan penulisan program *Pascal.*



Gambar 3.5. Perancangan *Software*

Keterangan Gambar:

Send	: Mengirimkan string yang telah dipecah per-karakter ke alat melalui port serial.
Open	: Akan mengaktifkan port serial ketika tombol open ditekan.
Clear	: Menghapus semua karakter yang telah di ketik pada memo.
Setup	: Akan membuka kotak dialog setting port serial.
Save	: Akan menyimpan pengaturan port serial.
Load	: Akan memanggil setting port serial yang telah disimpan.
Start Spell	: Mempersilakan pada penyandang tuna netra untuk mulai berbicara per-karakter.
Clear Result	: Menghapus semua karakter yang telah di ucapkan oleh penyandang tuna netra.

Clear Feedback : Menghapus karakter yang dikirimkan kembali oleh mikrokontroler.

Label *Hexa* : Menunjukkan nilai *hexadecimal* dari karakter.

Label Braille : Menunjukkan nilai kode Braille dari karakter.

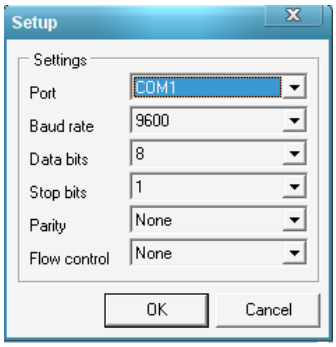
Label Status : Menunjukkan jenis dari karakter yang dikirim.

Label *Identifier* : Menunjukkan nilai *identifier* dari karakter yang menunjukkan jenis karakter kepada tuna netra.

Lampu Koneksi : Menunjukkan bahwa alat telah terkoneksi ke komputer.

Lampu Tx : Lampu ini akan menyala saat terjadi pengiriman data.

Lampu Rx : Lampu ini akan menyala saat terjadi penerimaan data.



Gambar 3.6. *Setup Alat*

3.3.1. Perancangan Perangkat Lunak Untuk Alat Bantu Kode Braille (*Delphi*)

Pada penelitian ini, perangkat lunak yang dirancang hanya mengubah input string yang diketikkan pada memo menjadi bentuk Hexa. Nilai Hexa yang didapat akan dikirimkan ke port serial.

3.3.2. Perancangan Perangkat Lunak Untuk Pengenalan Suara (Delphi)

Pada penelitian pengenalan suara per-karakter ini, menggunakan diktasi. Mode diktasi ini menggunakan metode pemisahan kata dengan cara *discrete speech, speaker dependent, Pencocokan kata whole word dan large vocabulary.*

Discrete speech, mengharuskan untuk mengucapkan kalimat secara terpenggal dengan adanya jeda sejenak diantara kata. Jeda tersebut

digunakan oleh sistem untuk mendeteksi awal dan akhir sebuah kata. *Speaker dependent* sistem membutuhkan pelatihan suara untuk setiap pengguna yang akan menggunakan sistem tersebut. Sistem tidak dapat mengenali suara pengguna yang belum pernah melakukan pelatihan. *Pencocokan kata dengan whole word,* sistem akan mencari di basis data kata yang sama persis dengan kata hasil ucapan. Algoritma pengenalan suara dapat dilihat pada lampiran B.

Diktasi dengan cara di atas membutuhkan minimal 8MB RAM dan processor Pentium. Sedangkan untuk sistem operasi yang diperlukan agar sistem dapat berjalan adalah Windows 95, Windows NT 3.5 dan yang lebih besar lagi (Windows XP).

3.3.2.1 Langkah Pembuatan Sistem Pengenalan Suara

Langkah pertama pembuatan sistem pengenalan suara adalah membuat antar muka utama bagi aplikasi. Antar muka utama yang digunakan oleh aplikasi ialah ISpRecoContext. Antar muka ini merupakan sebuah ISpEventSource, yang berarti antar muka aplikasi untuk menerima notifikasi dari sebuah even pengenalan pembicaraan. Aplikasi mempunyai dua pilihan untuk menggunakan e*ngine* pengenalan pembicaraan (ISpRecognizer), yaitu *shared recognizer* dan *inproc recognizer*. *Shared recognizer* memungkinkan untuk berbagi resource dengan aplikasi pengenalan pembicaraan yang lain, sedangkan *inproc recognizer* merepresentasikan sebuah *engine* pengenalan suara hanya untuk satu aplikasi saja. Untuk membuat ISpRecoContext untuk *shared* ISpRecognizer, aplikasi hanya memerlukan pemanggilan terhadap CoCreateInstance (COM) pada komponen CLSID_SpSharedRecoContext dan SAPI akan mengatur *audio input stream* sesuai dengan pengaturan standard SAPI.

Langkah kedua adalah pengaturan notifikasi untuk even yang dibutuhkan oleh aplikasi. Karena ISpRecognizer merupakan ISpNotifySource, maka aplikasi dapat memanggil salah satu metode IspNotify Source dari ISpRecoContext untuk

memberitahukan sebuah even dari ISpRecoContext. Kemudian aplikasi juga harus memanggil ISpEventSource::SetInterest untuk menyatakan even mana yang perlu untuk dinotifikasikan. Even yang paling penting adalah SPEI_RECOGNITION, yang berfungsi untuk menyatakan bahwa ISpRecognizer telah mengenali suara tertentu dari ISpRecoContext.

Langkah ketiga adalah mengaktifkan ISpRecoGrammar, sesuai dengan mode pengenalan suara, misalkan diktasi atau *command and control.* Langkah pertama, aplikasi harus membuat ISpRecoGrammar menggunakan IspReco Context::CreateGrammar. Kemudian aplikasi harus memanggil ISpRecoGrammar::Load Dictation untuk mode diktasi atau IspRecoGrammar::LoadCmdxxx untuk mode *command and control*. Aplikasi harus memanggil ISpRecoGrammar::SetDictationState untuk mode diktasi atau IspRecoGrammar::SetRuleState/ISpRecoGrammar::SetRuleIdStat e untuk mode *command and control* untuk mengaktifkan *grammar* sehingga sistem pengenalan pembicaraan dapat bekerja. Ketika terjadi notifikasi pada saat pengenalan suara, maka lParam yang merupakan variabel anggota dari struktur SPEVENT akan menjadi ISpRecoResult, yang kemudian digunakan oleh aplikasi untuk dapat menentukan apa yang telah dikenali.

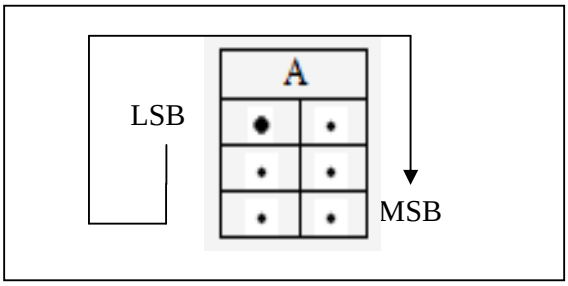
3.4. Algoritma Program Assembler

Perangkat lunak ini berfungsi mengendalikan keseluruhan sistem elektronika alat bantu baca tunanetra. Secara umum, algoritma pada mikrokontroler dimulai dari inisialisasi port dan alamat-alamat yang akan digunakan, selanjutnya jika ada interupsi serial, maka penerimaan data bisa dimulai dan dibandingkan untuk mengetahui hasil kode Braille dan *indentifier* Braille dari nilai yang didapat. *Flowchart* dari alat bantu kode Braille dapat dilihat pada lampiran B.

3.4.1. Tabel Karakter Program Assembler

Tabel karakter dalam kode *Braille* didapat dengan cara mengubah setiap karakter menjadi nilai *Hexa*, lalu diubah menjadi kode *Braille* sesuai

dengan bentuk huruf *Braille* standard dan urutan pada perancangan solenoid yang dibuat. Contohnya dapat dilihat pada gambar di bawah ini:



Gambar 3.7.a. Urutan *LSB* dan *MSB* pada huruf *Braillle*

Tabel 3.1. Tabel karakter dalam kode braille

No.	Karakter	ASCII	Braille
1	Spasi	20H	18H
2	!	21H	16H
3	"	22H	00H
4	$	24H	3CH
5	*	2AH	28H
6	+	2BH	16H
7	,	2CH	02H
8	-	2DH	24H
9	.	2EH	32H
10	0	30H	01H
11	1	31H	03H
12	2	32H	09H
13	3	33H	19H
14	4	34H	11H
15	5	35H	0BH
16	6	36H	1BH
17	7	37H	13H
18	8	38H	0AH
19	9	39H	1AH
20	:	3AH	26H
21	;	3BH	06H
22	=	3DH	30H
23	?	3FH	26H
24	A	41H	01H
25	B	42H	03H
26	C	43H	09H
27	D	44H	19H
28	E	45H	11H
29	F	46H	0BH
30	G	47H	1BH
31	H	48H	13H

No.	Karakter	ASCII	Braille
46	W	57H	3AH
47	X	58H	2DH
48	Y	59H	3DH
49	Z	5AH	35H
50	a	61H	01H
51	b	62H	03H
52	c	63H	09H
53	d	64H	19H
54	e	65H	11H
55	f	66H	0BH
56	g	67H	1BH
57	h	68H	13H
58	i	69H	0AH
59	j	6AH	1AH
60	k	6BH	05H
61	l	6CH	07H
62	m	6DH	0DH
63	n	6EH	1DH
64	o	6FH	15H
65	p	70H	0FH
66	q	71H	1FH
67	r	72H	17H
68	s	73H	0EH
69	t	74H	1EH
70	u	75H	25H
71	v	76H	27H
72	w	77H	3AH
73	x	78H	2DH
74	y	79H	3DH
75	z	7AH	35H

Tabel 3.1. Lanjutan

32	I	49H	0AH
33	J	4AH	1AH
34	K	4BH	05H
35	L	4CH	07H
36	M	4DH	0DH
37	N	4EH	1DH
38	O	4FH	15H
39	P	50H	0FH
40	Q	51H	1FH
41	R	52H	17H
42	S	53H	0EH
43	T	54H	1EH
44	U	55H	25H
45	V	56H	27H