

2. TEORI PENUNJANG

2.1. APACHE Web Server

Web Server adalah *server* internet yang mampu melayani koneksi transfer data dalam protokol HTTP (*Hypertext Transfer Protocol*). *Web server* saat ini merupakan inti dari *server-server* internet selain *e-mail server* maupun FTP *server*. Hal ini dapat dimaklumi karena *web server* telah dirancang untuk dapat melayani beragam jenis data dan informasi, mulai dari teks, *hypertext*, gambar (*image*), suara, gambar tiga dimensi, *plug-in* dan sebagainya. Keunggulan ini membuat *web* dapat diterima tidak saja dikalangan institusi Universitas, melainkan juga pada hampir semua perusahaan komersial dan lembaga saat ini telah memiliki sebuah atau beberapa *web server* sekaligus di internet.

APACHE adalah salah satu *web server* yang dapat dijalankan pada beberapa macam sistem operasi. APACHE merupakan turunan dari *web server* yang dikeluarkan NCSA yaitu NCSA HTTPd pada sekitar tahun 1995-an. Pada dasarnya APACHE adalah “A PatCHy” (patch) dan pengganti NCSA HTTPd.

Beberapa keuntungan penggunaan APACHE antara lain :

- a. Sangat mudah dikonfigurasi dan dengan mudah digabungkan dengan beberapa modul digabungkan dengan beberapa modul tambahan lainnya (contoh : modul PHP dan modul SSL)
- b. Karena APACHE merupakan *web server* yang gratis dan *open source*, maka tidak perlu melakukan registrasi atau membayar ijin (*license*) untuk memakai dan pemakai diperbolehkan merubah kode asli dari program *web server*
- c. Dapat dijalankan di beberapa sistem operasi, antara lain: WINDOWS 2000/NT/9x, Netware 5.x, OS/2, dan UNIX/LINUX
- d. Secara aktif terus menerus dilakukan pengembangan terhadap beberapa fitur-fitur yang sangat dibutuhkan.

Meskipun beberapa *web server* lain yang merupakan *web server* komersial telah menyatakan bahwa kecepatannya melebihi APACHE, tetapi kecepatan dari APACHE *web server* masih dapat diandalkan. Dan yang lebih menarik perhatian

karena APACHE merupakan *web server* yang gratis (*free*) dan APACHE telah digunakan pada beberapa situs yang memiliki jumlah pengunjung jutaan per hari.

2.2. Hypertext Markup Language (HTML)

HTML berfungsi untuk membuat situs *web* yang dapat ditampilkan pada program *web browser*. Hal ini dilakukan dengan menggunakan *tag* (perintah khusus). HTTP (*Hypertext Transfer Protocol*) merupakan protokol yang digunakan untuk mentransfer data antara *web server* ke *web browser*. Protokol ini men-*transfer* dokumen-dokumen *web* yang ditulis dalam bentuk HTML (*Hypertext Markup Language*).

2.2.1. Tag pada HTML

Tag HTML biasanya berupa *tag* yang berpasangan dan ditandai dengan simbol ‘<’ dan ‘>’. Pasangan atau penutup perintah dari sebuah *tag* ditandai dengan tanda ‘/’.

Setiap dokumen HTML memiliki struktur atau susunan *file* sebagai berikut:

```
<HTML>
<HEAD>
<TITLE>Judul pada title bar web server</TITLE>
</HEAD>

<BODY>
Text, gambar, atau segala sesuatu yang akan
ditampilkan pada dokumen
</BODY>

</HTML>
```

Seperti terlihat pada struktur *file* HTML selalu diawali dengan sebuah *tag* <HTML> dan ditutup dengan *tag* </HTML>. Di dalam *tag* ini terdapat dua bagian besar, yaitu yang diapit oleh *tag* <HEAD>...</HEAD> dan *tag* <BODY>...</BODY>.

Bagian yang diapit oleh *tag* HEAD adalah merupakan *header* dari halaman HTML dan tidak ditampilkan pada *browser*. Bagian ini berisi *tag-tag* seperti <TITLE>...</TITLE> yang berfungsi untuk mengeluarkan judul pada *title bar window browser*.

Bagian kedua yang diapit oleh tag `<BODY>` merupakan bagian yang akan ditampilkan pada halaman *web browser*. Pada bagian ini merupakan tempat untuk menuliskan semua jenis informasi berupa teks maupun gambar dengan bermacam-macam format yang ingin disampaikan kepada pengguna.

2.2.2. Form pada HTML

Form HTML merupakan tag yang paling penting khususnya dalam pembuatan aplikasi berbasis web. Form menyediakan elemen-elemen masukan (*input*) yang dapat berupa *textbox*, *check box*, *radio button*, dan tombol.

Pembuatan *form* pada HTML menggunakan tag `<form>...</form>` dan `<input>...</input>`.

Contoh pembuatan *form* pada HTML adalah sebagai berikut:

```
<html>
<body>

<form>
First name:
<input type="text" name="firstname">
<br>
Last name:
<input type="text" name="lastname">
</form>

</body>
</html>
```

2.2.3. Paragraf pada HTML

Dalam pembuatan paragraf pada HTML digunakan tag `<p>...</p>`. Tag `<p>...</p>` menandai sekumpulan teks sebagai suatu paragraf. Tag `<p>` menyatakan awal dari paragraf, sedangkan `</p>` menyatakan akhir dari sebuah paragraf. Kedua tag tersebut harus ditulis berpasangan.

Contoh penulisan pembuatan paragraf menggunakan tag `<p>...</p>` pada HTML adalah sebagai berikut:

```
<HTML>
<BODY>
  <p>Ini adalah sebuah paragraf </p>
  <p>Ini merupakan paragraf baru</p>
</BODY>
</HTML>
```

2.2.4. Format Teks pada HTML

HTML dapat menampilkan teks secara tebal, miring, ataupun dengan garis bawah. Pada dasarnya, terdapat dua pengelompokan dalam memformat teks, yaitu *Physical Style* yang memformat teks secara fisik teks dan *Logical Style* yang memformat teks berdasarkan kandungan teks yang telah disepakati secara internasional.

a. *Physical Style*

Physical Style adalah jenis format yang diberikan pada teks berdasarkan *tag* yang memang digunakan untuk mengatur tebal huruf, miring, maupun garis bawah.

Tag-tag yang bisa digolongkan ke dalam *Physical Style* adalah:

- ‘...’ berfungsi untuk menebalkan teks.
- ‘<i>...</i>’ berfungsi untuk membuat teks terlihat miring.
- ‘<u>...</u>’ berfungsi untuk memberi garis bawah pada teks.
- ‘<big>...</big>’ berfungsi untuk membesarkan teks.
- ‘<small>...</small>’ berfungsi untuk mengecilkan teks.

b. *Logical Style*

Logical Style adalah jenis format yang diberikan pada teks berdasarkan kesepakatan internasional terhadap kandungan teks tertentu. Saat ini untuk memberikan penekanan teks menggunakan huruf miring. Jika suatu saat standar internasional mengubah penekanan otomatis teks yang diformat menggunakan *Logical Style* akan berubah menyesuaikan standar yang baru.

Tag-tag yang bisa digolongkan ke dalam *Logical Style* adalah:

- ‘...’ berfungsi untuk memberikan penekanan pada teks.
- ‘...’ berfungsi untuk menyatakan teks dengan penekanan yang kuat.

2.2.5. Link pada HTML

Membuat *link* dalam HTML dilakukan dengan menggunakan *tag* ‘<a>...’. Atribut yang digunakan di dalam *tag* ‘<a>...’ adalah atribut *href*=”URL”.

Contoh penggunaan tag '`<a>...</a>`' pada pembuatan *link* HTML adalah sebagai berikut:

```
<html>
<body>

<p>
<a href="lastpage.htm">
This text</a> is a link to a page on
this Web site.
</p>

<p>
<a href="http://www.microsoft.com/">
This text</a> is a link to a page on
the World Wide Web.
</p>

</body>
</html>
```

2.3. Bahasa C

C berukuran kecil namun merupakan bahasa pemrograman yang sangat efektif. Bahasa C mudah untuk dipelajari namun sulit untuk dikuasai. Eksekusi program C sangat cepat. Bahasa C digunakan dalam pembuatan program-program aplikasi bahkan sistem operasi. Sistem operasi DOS maupun UNIX ditulis dengan bahasa C, sehingga C menjadi bahasa yang populer di kalangan *programmer*.

C telah didesain sebagai bahasa tingkat rendah (*low level*), sehingga C dapat dipergunakan para *programmer* untuk membuat sistem operasi dan program yang mengontrol perangkat keras. Hal ini dikarenakan bahasa C adalah bahasa yang berkecepatan sangat tinggi. Namun C juga memberikan fleksibilitas dan struktur data seperti bahasa pemrograman tingkat tinggi. C mempunyai fitur-fitur terbaik baik dari bahasa pemrograman tingkat tinggi dan rendah. Karena alasan ini, C disebut sebagai bahasa tingkat menengah.

Program C mempunyai struktur yang cukup sederhana. Sebuah program C terdiri dari kumpulan fungsi, termasuk fungsi khusus yaitu `main()` sebagai titik awal dari pengeksekusian program. Semua fungsi dalam program C mempunyai tingkatan yang sama, sehingga tidak ada fungsi yang tersembunyi dalam yang lain. Sebagai contoh berikut ini menunjukkan sebuah program C sederhana:

```
#include <stdio.h>
```

```
main()
{
    printf ("Hello world\n");
}
```

2.4. PHP (Hypertext Preprocessor)

PHP adalah bahasa berbentuk skrip yang ditempatkan dan diproses di *server*, yang kemudian dikirimkan ke *client* (tempat pemakai menggunakan *browser*). Secara khusus, PHP dirancang untuk membentuk *web* dinamis. Artinya, ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Pada prinsipnya, PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), *Cold Fusion*, ataupun Perl.

Kelahiran PHP bermula saat Rasmus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya (pada tahun 1994). Skrip-skrip ini selanjutnya dikemas menjadi *tool* yang disebut "*Personal Home Page*". Paket inilah yang menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrogram dapat menempel kode terstruktur dalam *tag* HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan.

Pada saat ini, PHP cukup populer sebagai peranti pemrograman *web*, terutama di lingkungan LINUX. Walaupun demikian, PHP juga dapat berfungsi pada *server-server* yang berbasis UNIX, Windows 2000/98/95/NT, dan Macintosh.

Pada awalnya, PHP dirancang untuk diintegrasikan dengan *web server* Apache. Namun, belakangan PHP juga dapat bekerja dengan *web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami.

2.4.1. PHP Bersifat Bebas

PHP bersifat bebas dipakai. Tidak diperlukan pembayaran apapun untuk menggunakan perangkat lunak ini. PHP dapat di-*download* melalui situs (www.php.net). Untuk versi Windows, dapat diperoleh kode binernya, untuk versi Linux dapat diperoleh kode sumber secara lengkap.

2.4.2. Skrip PHP

Skrip PHP berkedudukan sebagai *tag* dalam bahasa HTML. Sebagaimana diketahui, HTML (*Hypertext Markup Language*) adalah bahasa standar untuk membuat halaman-halaman *web*. Adapun kode berikut adalah contoh kode PHP yang berada di dalam kode HTML:

```
<HTML>
<HEAD>
<TITLE>Pertama</TITLE>
</HEAD>
<BODY>
Selamat menulis PHP.<BR>
<?php
printf("Tgl. Sekarang: %s", Date ("d F Y"));
?>
</BODY>
<?HTML>
```

Kode di atas disimpan dengan ekstensi “.php”. Perhatikan baris-baris yang ada berikut:

```
<?php
printf("Tgl. Sekarang: %s", Date ("d F Y"));
?>
```

Kode tersebutlah yang disebut kode PHP. Kode PHP diawali dengan ‘<?php’ dan diakhiri dengan ‘?>’. Pasangan kedua kode inilah yang berfungsi sebagai *tag* kode PHP. Berdasarkan *tag* inilah, pihak *server* dapat memahami kode PHP dan kemudian memprosesnya. Hasilnya dikirim ke *browser*.

2.5. WAP (Wireless Application Protocol)

WAP adalah suatu protokol aplikasi yang memungkinkan internet dapat diakses oleh ponsel dan perangkat *wireless* lainnya. WAP membawa informasi secara *online* melewati internet langsung menuju ke ponsel atau *client* WAP lainnya. Dengan adanya WAP, berbagai informasi dapat diakses setiap saat hanya dengan menggunakan ponsel.

Protokol ini awalnya dikembangkan oleh WAP Forum (<http://www.wapforum.org>) pada tahun 1997 yang didirikan oleh Ericsson, Motorola, Nokia, dan *Unwired Planet* (sekarang www.phone.com). WAP Forum merupakan organisasi nirlaba yang bertujuan untuk menetapkan standar dalam memberikan akses Internet ke kelas konsumen melalui alat *wireless*. Standar ini

membantu *platform* global untuk menciptakan jalur yang berbeda tetapi sejalan dengan internet.

Kini sudah lebih dari 200 perusahaan tergabung pada WAP Forum yang terdiri atas *vendor* ponsel, *service provider*, dan *software developer*. Karena itu, diperkirakan perkembangan teknologi yang dilahirkan oleh WAP Forum akan lebih dahsyat lagi, mengingat jumlah pemilik yang jauh lebih banyak dibandingkan pemilik komputer yang saat ini diperkirakan 380 juta berbanding 200 juta.

Cara kerja WAP hampir sama dengan cara kerja internet saat ini. Dibutuhkan *WAP Gateway* untuk menjembatani ponsel dengan internet dalam mengirim dan menerima data. Hal ini sama halnya dengan pengguna komputer yang membutuhkan ISP (*Internet Service Provider*) sebagai *gateway* dalam menjembatani komputer dengan internet. Disamping itu, ponsel yang digunakan juga harus *WAP-enabled*, yaitu sudah dilengkapi dengan teknologi WAP yang bisa digunakan untuk mengakses internet.

Model pemrograman WAP terdiri atas tiga bagian, yaitu *WAP Client* yang mengirim permintaan informasi, *WAP Gateway* sebagai penterjemah antara WAP dan HTTP, serta *server* yang memproses permintaan dari *gateway* untuk kemudian menjawabnya.



Gambar 2.1. Alur Komunikasi melalui WAP

Sumber: Sanjaya, Ridwan, & W. Purbo, Onno. *Membuat Aplikasi WAP dengan PHP*. Jakarta: Elex Media Komputindo. 2001. hal 8.

2.5.1. WAP Application Server

Agar suatu aplikasi internet dapat berjalan pada *web server*, maka tipe atau format data aplikasi tersebut harus dapat dikenali oleh *web server*. Format data yang dapat dilayani oleh internet disebut MIME (*Multipurpose Internet Mail*

Extension). Tipe MIME ini dikirim oleh *server* bersama data dan melalui MIME ini maka *browser* mengetahui data apa yang dikirim. Data MIME untuk aplikasi WAP antara lain adalah:

- a. *Wireless Markup Language (text/vnd.wap.wml)*, MIME ini berekstensi wml.
- b. *Compiled WML (application/vnd.wap.wmlc)*, MIME ini berekstensi wmlc.
- c. *WMLScript (text/vnd.wap.wmlscript)*, MIME ini berekstensi wmls.
- d. *Compiled WMLScript (application/vnd.wap.wmlscript)*, MIME ini berekstensi wmlsc.
- e. *Wireless Bitmap (image/vnd.wap.wbmp)*, MIME ini berekstensi wbmp.

2.5.2.WAP Client

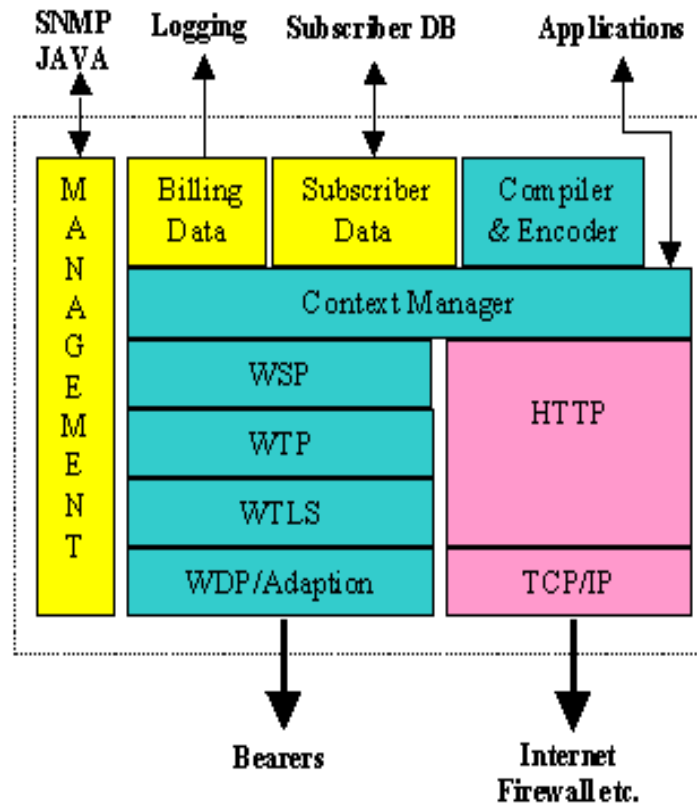
Spesifikasi *WAP Client* antara lain adalah *WAE user agent*, *WTA user agent* dan *WAP Stack*.

- a. *WAE user agent* adalah suatu *microbrowser* yang membaca *content*. Bagian ini menerima WML yang sudah di-*compile*, WMLScript dan semua gambar dari *WAP Gateway*, dan menampilkannya. *WAE user agent* juga termasuk mengatur interaksi dengan *user*, seperti *input* teks, dan pesan *error*.
- b. *WTA user agent* adalah bagian yang mengatur akses ke *interface* ponsel, fungsi *network* seperti *number dialling*, *call answering*, *phonebook*, *message management*, dan *lokasi indikasi servis*.
- c. *WAP stack* adalah bagian yang memperbolehkan ponsel berhubungan dengan *WAP gateway* menggunakan *Wireless Application Protocol*.

2.5.3.WAP Gateway

WAP Gateway bertindak sebagai penterjemah antara internet dan *mobile device*, sehingga memungkinkan keduanya saling berkomunikasi. *WAP Gateway* ini hanya bagi *network operator*, sehingga dalam mensimulasikan aplikasi WAP melalui komputer tidak diperlukan *WAP Gateway*.

WAP Gateway berfungsi untuk meneruskan permintaan informasi dari ponsel menuju server lewat *HTTP Request*, dan sebaliknya, dari *server* menuju ponsel lewat *HTTP Response*.



Gambar 2.2. Arsitektur WAP Gateway

Sumber: Sanjaya, Ridwan, & W. Purbo, Onno. *Membuat Aplikasi WAP dengan PHP*. Jakarta: Elex Media Komputindo. 2001. hal 9.

WAP Gateway memiliki susunan sebagai berikut:

- **WDP**: *WAP Datagram Protocol* adalah *layer transport* yang digunakan untuk mengirim dan menerima pesan atau data melalui segala macam pembawa pesan di jaringan termasuk SMS, USSD, CSD, CDPD, IS-136 paket data dan GPRS.
- **WTLS**: *Wireless Transport Layer Security* adalah sebuah *layer* keamanan yang menyediakan kemampuan enkripsi sehingga keamanan transaksi yang dibutuhkan oleh aplikasi dapat terpenuhi
- **WTP**: *WAP Transaction Protocol* adalah *layer* pendukung transaksi. *Layer* ini menambahkan *reliability* ke *diagram service* yang disediakan oleh WDP.
- **WSP**: *WAP Session Protocol* adalah *layer* yang menyediakan sesi *layer lightweight* untuk efisiensi pertukaran data diantara aplikasi.

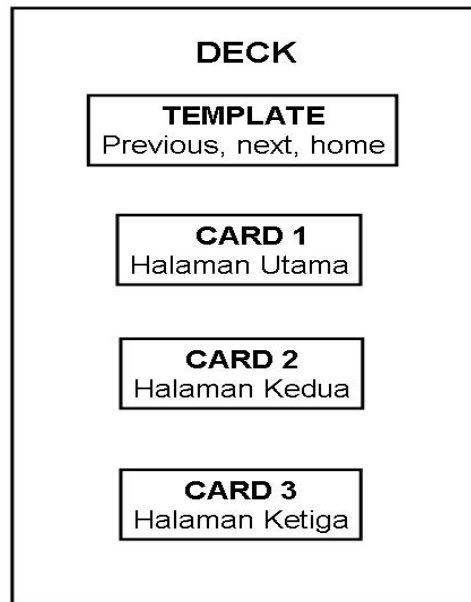
- HTTP *Interface* adalah *interface* yang digunakan untuk melayani penerimaan *WAP Content* dari internet menggunakan ponsel.

2.6. WML (Wireless Markup Language)

WAP menggunakan bahasa komputasi yang dikenal sebagai *Wireless Markup Language* (WML) yang mirip dengan HTML. Bahasa komputer yang pintar ini mengubah informasi berupa teks dari halaman situs dan menampilkannya ke layar ponsel. Aturan-aturan standar dari penulisan struktur bahasa ini dapat ditemukan pada *website* WAP Forum (www.wapforum.com). WML merupakan subset dari XML (*extensible markup language*). Namun tidak perlu mempelajari XML terlebih dahulu untuk memahami WML.

Jika HTML mempunyai *javascript* untuk membuat halaman-halaman di dalamnya menjadi interaktif, WML juga mempunyai versi sendiri dari *javascript* yang disebut *WML Script*. Namun berbeda dengan *javascript* yang bisa dimasukkan ke dalam HTML, *WML Script* tidak bisa dijadikan satu dengan WML dan harus berdiri sendiri dengan satu *file* skrip khusus yang berekstensi *wmls*. Perbedaan lainnya adalah gambar yang tampil di layar ponsel haruslah gambar yang telah dikonversi ke dalam format *wbmp* 1 bit (yang saat ini masih terdiri dari warna hitam dan warna latar belakang saja).

Secara umum beberapa perintah WML terlihat mirip dengan HTML. Namun terdapat beberapa perbedaan dalam struktur penulisan dokumen WML. Jika sebuah dokumen HTML hanya terdiri atas dua bagian utama (yaitu *header* dan *body*), dokumen WML mempunyai *header*, *template (optional)*, dan beberapa *body* yang disebut *cards*. Bagian *header* berfungsi untuk menyatakan versi XML yang dari suatu dokumen WML, *template* berfungsi untuk memberikan tambahan pilihan pada menu *options* atau tambahan tombol di beberapa *browser*, dan *card* berfungsi untuk mendefinisikan halaman-halaman yang berada dalam suatu *file* *wml*. Susunan dokumen secara lengkap ini disebut dengan *deck*. Sama seperti halnya HTML, untuk menyusun kode WML diperlukan kode khusus yang dinamakan dengan *tag*. Penulisan *template* dan *card* berada dalam *tag* `<wml>...</wml>`.



Gambar 2.3. Hirarki dalam WML

Sumber: Sanjaya, Ridwan, & W. Purbo, Onno. *Membuat Aplikasi WAP dengan PHP*. Jakarta: Elex Media Komputindo. 2001. hal 33.

2.6.1. Tag pada WML

Penulisan *tag* dilakukan agar dokumen WML dapat ditampilkan pada *browser* sesuai dengan keinginan. Biasanya, sebuah *tag* diawali dengan tanda '<' dan diakhiri dengan tanda '>'. *Tag* biasanya merupakan suatu pasangan, yang disebut sebagai *tag* awal dan *tag* akhir. *Tag* awal dinyatakan dengan bentuk `<nama_tag>` dan diakhiri dengan *tag* akhir yang dinyatakan dengan bentuk `</nama_tag>`. Jika diperhatikan, terdapat kemiripan dalam penulisan keduanya, namun jelas terdapat perbedaan. *Tag* akhir diawali dengan garis miring '/' dan kemudian dilanjutkan dengan penulisan nama *tag* yang digunakan. Format penulisannya adalah sebagai berikut:

```
<nama_tag>kalimat</nama_tag>
```

Sebagai contoh, jika menginginkan kalimat yang ditulis dapat ditampilkan dengan huruf tebal, yang harus dilakukan adalah menulis perintah seperti berikut ini:

```
<b>kalimat ini dicetak dengan huruf tebal</b>
```

Tidak semua *tag* ditulis berpasangan. Ada pula yang dinamakan *single tag* atau *tag* tunggal, yang hanya terdiri atas satu *tag* awal saja. Untuk penulisan *tag* tunggal, sebelum menuliskan penutup *tag*, lebih dahulu ditulis tanda garis miring. Format penulisan untuk *tag* tunggal adalah sebagai berikut:

```
<nama_tag/>
```

Contoh penggunaan *tag* tunggal adalah pada perintah ganti baris, perintah yang harus ditulis adalah sebagai berikut:

```
Kalimat lalu ganti baris<br/>
```

Komentar dapat ditulis di dalam dokumen WML. Meskipun tidak akan diproses oleh *browser*, tulisan tersebut dapat dijadikan sebagai alat dokumentasi. Berikut ini format penulisan komentar:

```
<!-- komentar ditulis disini-->
```

2.6.2. Atribut pada WML

Di dalam beberapa *tag* terdapat atribut yang menyertai dalam pengaturan fungsinya dalam mengatur isi dokumen. Penulisan atribut biasanya diletakkan pada *tag* awal dan berisi nilai-nilai tertentu. Misalkan dalam *tag* ‘<p>’, yang digunakan untuk membuat paragraf baru, terdapat atribut *align* yang digunakan untuk mengatur perataan teks. Nilai dari atribut dapat diisi dengan “*center*”, “*left*”, dan “*right*”. Contoh penulisannya adalah sebagai berikut:

```
<p align = "left">kalimat</p>
```

2.6.3. Card pada WML

Dalam sebuah dokumen WML bisa saja diisi dengan sejumlah *card* yang cukup banyak. Namun mengingat kapasitas *memory* ponsel yang cukup terbatas, penggunaan *card* lebih baik secukupnya saja. Atribut-atribut di dalam *tag* <card>...</card> yang sering digunakan antara lain:

a. *title*="judul_card"

Berfungsi untuk menampilkan judul atau topik dari halaman WAP.

b. *id*="nama_card"

Berfungsi untuk mengidentifikasi sebuah *card* untuk keperluan tujuan navigasi dari *card* lain.

Berikut adalah contoh dokumen WML sederhana yang mempunyai tiga *card* di dalamnya, yaitu:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
http://www.wapforum.org/DTD/wml_1.1.xml>
<wml>
<tempate>
</template>
<card title="Selamat Datang" id="utama">
<p>
<!-- Kartu pertama-->
Halaman Utama ini merupakan card pertama <br/>
<a href="#kedua">Klik</a>Untuk ke halaman berikutnya
</p>
</card>
<card title="Halaman Kedua" id="kedua">
<p>
Halaman Kedua<br/>
<a href="#ketiga">Klik</a>Untuk ke halaman berikutnya
</p>
</card>
<card title="ketiga id="ketiga">
<p>
Halaman Ketiga
</p>
</card>
</wml>
```

2.6.4. Paragraf pada WML

Tag ‘<p>..*</p>*’ menandai sekumpulan teks sebagai suatu paragraf. *Tag* ‘<p>’ menyatakan awal dari paragraf, sedangkan ‘</p>’ menyatakan akhir dari sebuah paragraf. Kedua *tag* tersebut harus ditulis berpasangan. Jika salah satu tidak ditulis, pada beberapa *browser* mengakibatkan halaman WAP tidak bisa dibaca.

Tabel 2.1. Tag dan Atribut Paragraf pada WML

Tag WML	Keterangan
...	Paragraf
	Atribut: <i>align = right/left/center</i> <i>mode = wrap/nowrap</i>

Sumber: Sanjaya, Ridwan, & W. Purbo, Onno. *Membuat Aplikasi WAP dengan PHP*. Jakarta: Elex Media Komputindo. 2001. hal 38.

2.6.5. Format Teks pada WML

Seperti halnya HTML, WML dapat menampilkan teks secara tebal, miring, ataupun dengan garis bawah. Pada dasarnya, terdapat dua pengelompokan dalam memformat teks, yaitu *Physical Style* yang memformat teks secara fisik teks dan *Logical Style* yang memformat teks berdasarkan kandungan teks yang telah disepakati secara internasional.

a. *Physical Style*

Physical Style adalah jenis format yang diberikan pada teks berdasarkan *tag* yang memang digunakan untuk mengatur tebal huruf, miring, maupun garis bawah.

Tag-tag yang bisa digolongkan ke dalam *Physical Style* adalah:

- ‘...’ berfungsi untuk menebalkan teks.
- ‘<i>...</i>’ berfungsi untuk membuat teks terlihat miring.
- ‘<u>...</u>’ berfungsi untuk memberi garis bawah pada teks.
- ‘<big>...</big>’ berfungsi untuk membesarkan teks.
- ‘<small>...</small>’ berfungsi untuk mengecilkan teks.

b. *Logical Style*

Logical Style adalah jenis format yang diberikan pada teks berdasarkan kesepakatan internasional terhadap kandungan teks tertentu. Saat ini untuk memberikan penekanan teks menggunakan huruf miring. Jika suatu saat standar internasional mengubah penekanan otomatis teks yang diformat menggunakan *Logical Style* akan berubah menyesuaikan standar yang baru.

Tag-tag yang bisa digolongkan ke dalam *Logical Style* adalah:

- ‘...’ berfungsi untuk memberikan penekanan pada teks.
- ‘...’ berfungsi untuk menyatakan teks dengan penekanan yang kuat.

2.6.6. Link pada WML

Dalam membuat *link* antar dokumen pada WML menggunakan *tag* ‘<anchor>...</anchor>’ dan bisa juga menggunakan *tag* ‘<a>...’.

Tabel 2.2. Tag dan Atribut Link.

Tag WML	Keterangan
<code><anchor>...</anchor></code>	Atribut: <i>title</i> ="judul" di dalam tag ' <code><anchor>...</anchor></code> ' terdapat tag ' <code><go href="URL"/></code> ' sebagai penunjuk alamat atau dokumen tertentu
<code><a>...</code>	Atribut: <i>href</i> ="URL"

Sumber: Sanjaya, Ridwan, & W. Purbo, Onno. *Membuat Aplikasi WAP dengan PHP*. Jakarta: Elex Media Komputindo. 2001. hal 43.

2.6.7. Form pada WML

Fasilitas *form* memungkinkan interaksi antara suatu situs WAP dengan pengunjungnya. *Form* dapat digunakan untuk berbagai keperluan, di antaranya untuk mengumpulkan informasi, saran, dan penawaran melalui situs WAP tersebut.

Struktur penulisan *form* dalam sebuah *card* adalah sebagai berikut:

```

<card id="contoh" title="Form">
<do type="accept" label="Hasil">
  <go href="tujuan"/>
</do>
<p>
  <input type="text" name="nama_variabel"/>
</p>
</card>

```

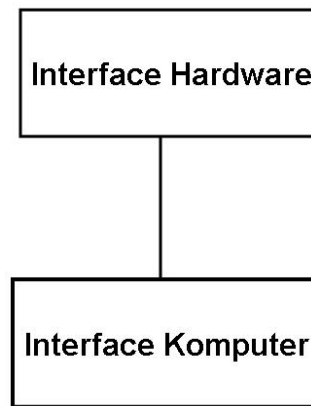
Terlihat pada contoh di atas, elemen *form* terdiri dari dua tag utama yaitu tag '`<do>`' yang menunjukkan tujuan jika *input* dieksekusi dan tag untuk isian (yang bisa berupa *input*, *select*, atau keduanya).

Untuk lebih jelasnya, elemen-elemen yang digunakan dalam membuat form adalah sebagai berikut

- a. Tag '`<do>...</do>`'. Di dalam tag ini menggunakan atribut *type*="accept".
Di dalam tag ini terisi elemen yang digunakan untuk memberikan tujuan setelah isian dieksekusi, yaitu:
 - '`<go href="tujuan"/>`'
 - '`<go href="tujuan" method="post/get">..</go>`'
- b. Tag '`<input/>`' berfungsi untuk membuat kotak isian. Atribut dalam tag '`<input/>`' adalah:
 - *name*="nama"
 - *value*="nilai"
- c. Tag '`<select>...</select>`' berfungsi untuk membuat kotak pilihan (*select box*). Atribut tag ini adalah: *name*="nama".
- d. Tag '`<option>...</option>`' berfungsi untuk membuat pilihan dalam kotak pilihan.

2.7. Komunikasi Serial

Pada komunikasi *serial*, data dikomunikasikan antara komputer dan suatu alat (*hardware*) melalui *interface serial* per 1 bit data dalam 1 waktu. Pada *interface serial* yang paling sederhana, data dapat dikomunikasikan melalui sebuah kabel *single*. Walaupun penggunaan *interface serial* yang sederhana cukup diinginkan, tapi hal ini kurang praktis dalam dunia nyata. Dibutuhkan lebih dari satu kabel untuk melakukan komunikasi *serial* yang baik. *Interface serial* yang umum digunakan adalah *interface serial RS 232*.



Gambar 2.4. Komunikasi Serial

Sumber: Goodwin, Mark. *Serial Communication in C and C++*. New York: Management Information Source, Inc. 1992. hal 34.

2.8. Interface Serial RS 232 9-Pin

Interface serial RS 232 adalah berasal dari istilah *Recommended Standard Number 232, Revision C*, dari *Engineering Departement Electronic Industries Association*. *Standard* ini mencakup konektor *female* pada *Data Communication Equipment* (DCE) dan konektor *male* pada *Data Terminal Equipment* (DTE). Singkatnya, DTE adalah komputer dan DCE adalah alat atau *hardware* yang akan berhubungan dengan komputer. *Standard* RS 232 dapat dibagi dua tipe, yaitu yang menggunakan 25 *pin* dan 9 *pin*. Komputer kelas IBM menggunakan 9 *pin*. Tabel 2.3. menunjukkan nama dan singkatan tiap *pin* pada RS 232 yang menggunakan 9 *pin*.

Tabel 2.3. Interface Serial RS 232 9-pin

Pin	Singkatan	Nama
1	DCD	<i>Data Carrier Detect</i>
2	RD	<i>Received Data</i>
3	TD	<i>Transmit Data</i>
4	DTR	<i>Data Terminal Ready</i>
5	-	<i>Signal Ground</i>
6	DSR	<i>Data Set Ready</i>

Tabel 2.3. Interface Serial RS 232 9-pin (sambungan)

Pin	Singkatan	Nama
7	RTS	<i>Request To Send</i>
8	CTS	<i>Clear To Send</i>
9	RI	<i>Ring Indicator</i>

Sumber: Goodwin, Mark. *Serial Communication in C and C++*. New York: Management Information Source, Inc. 1992. hal 78.

Berikut adalah penjelasan pin-pin tersebut:

1. *Pin 1: Data Carrier Detect (DCD)*, line DCD akan memiliki logika 1 apabila terdapat *data link* sedang dalam kemajuan (*progress*)
2. *Pin 2: Received Data (DCD)*, line RD digunakan oleh DCE untuk mengirimkan data ke DTE
3. *Pin 3: Transmitted Data (TD)*, line TD digunakan oleh DTE untuk mengirimkan data ke DCE.
4. *Pin 4: Data Terminal Ready (DTR)*, line DTR akan diberikan logika 1 oleh DTE (apabila DTE siap untuk berkomunikasi dengan DCE).
5. *Pin 5: Signal Ground*, line ini dihubungkan ke *ground*.
6. *Pin 6: Data Set Ready (DSR)*, line DSR akan diberi logika 1 oleh DCE (apabila DCE telah siap untuk berkomunikasi dengan DTE).
7. *Pin 7: Request To Send (RTS)*, line RTS akan diberi logika 1 oleh DTE (apabila DTE menginginkan untuk *men-transmit* data kepada DCE).
8. *Pin 8: Clear To Send (CTS)*, line CTS akan diberi logika 1 oleh DCE saat DCE siap untuk menerima data.
9. *Pin 9: Ring Indicator (RI)*, line RI akan diberi logika 1 oleh DCE (apabila sebuah *ring* terdeteksi).

2.9. Alamat Serial Port

Untuk penggunaan *serial port* pada komputer dibutuhkan pengalamatan, agar dapat mengaksesnya. Tabel 2.4. menunjukkan nama *serial port* dan alamat *port*-nya.

Tabel 2.4. Alamat Serial Port

<i>Serial Port</i>	<i>Alamat Port</i>
COM1	3F8 _H
COM2	2F8 _H
COM3	3E8 _H
COM4	2E8 _H

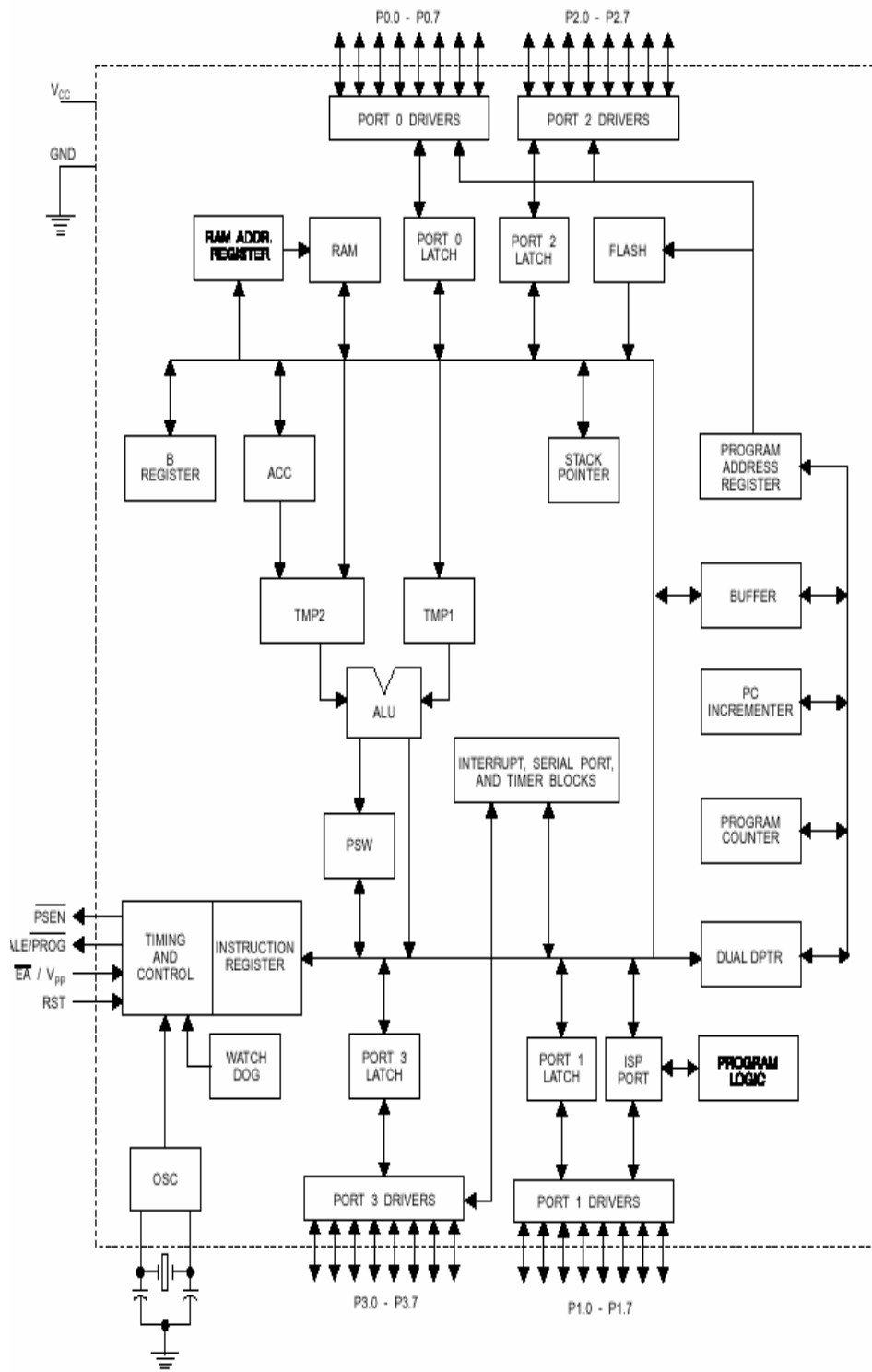
Sumber: Goodwin, Mark. *Serial Communication in C and C++*. New York: Management Information Source, Inc. 1992. hal 82.

2.10. Mikrokontroler AT89S51

Mikrokontroler AT89S51 adalah mikrokontroler CMOS 8-bit dengan 4K byte *In-System Programmable Flash Memory*. Mikrokontroler AT89S51 memiliki fitur-fitur sebagai berikut:

- 8 bit CMOS mikrokomputer
- 4K bytes *In-System Programming Flash Memory*
- 128 bytes internal RAM
- 32 jalur *input-output*
- *Watchdog Timer*
- Dua *pointer* data
- Dua 16-bit *timer/counter*
- Memiliki arsitektur lima *two-level vector interrupt*
- Sepasang *full duplex serial port*
- *On chip oscillator*
- *Clock circuitry*
- Pemrograman ISP yang fleksibel

Struktur dasar AT89S51 digambarkan secara blok diagram seperti pada gambar 2.5.

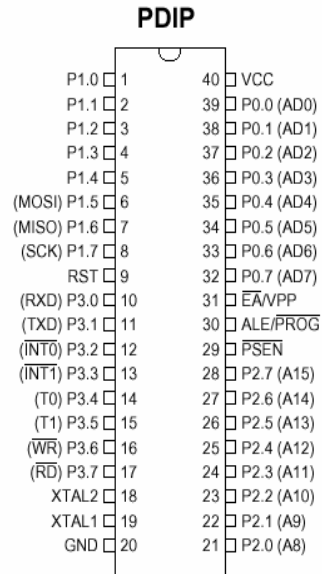


Gambar 2.5. Blok Diagram AT89S51

Sumber: *Atmel Corporation AT89S51 Microcontroller Data Sheet*, hal 3.

2.10.1. Penjelasan Fungsi Pin AT89S51

Mikrokontroler AT89S51 dengan 40 *pin* dapat digambarkan sebagai berikut:



Gambar 2.6. Konfigurasi Pin AT89S51

Sumber: *Atmel Corporation AT89S51 Microcontroller Data Sheet*. hal 2.

AT89S51 memiliki konfigurasi *pin* seperti tampak pada gambar dengan fungsi masing-masing *pin* adalah sebagai berikut:

- a. VCC
 - Tegangan Sumber +5V
- b. GND
 - *Ground*
- c. Port 0
 - Port 0 adalah 8-bit I/O yang bersifat *open drain bidirectional*. Jika difungsikan sebagai *output*, tiap *pin* dapat dihubungkan dengan 8 buah TTL *input*. Saat '1' ditulis pada *pin port 0*, *pin* tersebut dapat digunakan sebagai *input* yang bersifat *high impedance*. Apabila difungsikan sebagai *port*, *port* ini memerlukan *external pull-ups*.
 - Port 0 juga dapat difungsikan untuk memultipleks *data/address bus* (pada bit 0-7). Pada mode ini *port 0* memiliki *internal pull-ups*.

d. *Port 1*

- *Port 1* adalah 8-bit port I/O yang bersifat *bidirectional* dengan *internal pull-ups*. Jika difungsikan sebagai *output*, *port 1* dapat dihubungkan dengan empat *input* TTL. Saat '1' ditulis pada *pin port 1*, maka *port 1* dapat digunakan sebagai *input*. Jika secara eksternal *pin port 1* di *pull down* maka *port 1* akan mengeluarkan arus sebagai akibat dari *internal pull-ups* yang dimilikinya.
- *Port 1* juga menerima *low address bytes* selama pemrograman *flash* dan verifikasi.

Tabel 2.5. Fungsi Khusus Pin 1

<i>Port Pin</i>	Fungsi Alternatif
P1.5	MOSI (<i>used for In-System Programming</i>)
P1.6	MISO (<i>used for In-System Programming</i>)
P1.7	SCK (<i>used for In-System Programming</i>)

Sumber: *Atmel Corporation AT89S51 Microcontroller Data Sheet*. hal 4.

e. *Port 2*

- *Port 2* adalah 8-bit port I/O yang bersifat *bidirectional* dengan *internal pull-ups*. Jika difungsikan sebagai *output*, *port 1* dapat dihubungkan dengan empat *input* TTL. Saat '1' ditulis pada *pin port 1*, maka *port 1* dapat digunakan sebagai *input*. Jika secara eksternal *pin port 1* di *pull down* maka *port 1* akan mengeluarkan arus sebagai akibat dari *internal pull-ups* yang dimilikinya.
- *Port 2* juga menerima *high-order address bits* dan beberapa sinyal kontrol saat pemrograman *flash* dan verifikasi.

f. *Port 3*

Port 3 apabila difungsikan sebagai I/O memiliki konfigurasi yang sama dengan *port 1*. disamping itu tiap-tiap *pin port 3* juga memiliki fungsi khusus seperti yang tertera pada tabel 2.4. dibawah ini.

- *Port 3* menerima beberapa sinyal kontrol pada pemrograman *flash* dan verifikasi.
- *Port 3* juga melayani beberapa fungsi seperti yang terdapat pada tabel.

Tabel 2.6. Fungsi Khusus Port 3

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Sumber: *Atmel Corporation AT89S51 Microcontroller Data Sheet*. hal 5.

g. Reset

- *Input reset*. Jika *pin* ini mendapat logika “high” selama dua *machine cycle* saat *oscillator* aktif, maka alat ini akan ter-reset.

h. ALE/PROG

- *Address Latch Enable* mengeluarkan pulsa untuk me-latch byte rendah saat pengaksesan eksternal *memory*. *Pin* ini juga difungsikan sebagai *input* pulsa rendah saat pemrograman *flash*.
- Pada operasi normal, ALE mengeluarkan pulsa konstan sebesar 1/16 dari frekuensi *oscillator*, dan dapat digunakan untuk penggunaan *timing* eksternal atau untuk *clocking*. Jika diinginkan *pin* ALE dapat dinonaktifkan dengan men-set “0” pada alamat 8EH pada SFR, akan tetapi mode ini tidak mempengaruhi saat akses eksternal *memory*.

g. PSEN

- PSEN adalah sinyal baca pada saat pengaksesan eksternal *program memory*.

h. EA/Vpp

- EA (*Eksternal Access*) digunakan untuk pengaktifan *program memory*. Apabila EA di-set ke '0', maka *program memory* sepenuhnya dibaca dari eksternal ROM. Apabila EA di-set ke '1', maka *program memory* internal akan dipakai dan kemudian dilanjutkan ke *program memory* eksternal.
- *Pin* ini juga menerima *input* tegangan 12 Volt saat pemrograman *flash*.

i. XTAL1

- Merupakan *pin input* untuk *inverting oscillator amplifier* dan *input* dari sirkuit internal clock.

j. XTAL 2

- Merupakan *output* dari *inverting oscillator amplifier*.

2.11. LCD Dot Matrix

LCD *dot matrix* yang digunakan dalam tugas akhir ini adalah *LCD Display Module M1632* buatan Seiko. LCD ini terdiri dari dua bagian. Bagian pertama merupakan panel LCD sebagai penampil informasi dalam bentuk karakter sebanyak 2 baris (masing-masing bisa menampilkan 16 karakter). Bagian kedua merupakan sebuah sistem yang ditempelkan dibalik panel LCD, berfungsi untuk mengatur tampilan informasi dan mengatur komunikasi LCD dengan mikrokontroler yang memakai tampilan LCD tersebut.

Untuk berhubungan dengan mikrokontroler, LCD ini dilengkapi dengan 8 jalur data (D0...D7), dilengkapi pula dengan E, R/W, RS.

Tabel 2.7. Jalur LCD M1632

No	Simbol	Level	Fungsi
1	Vss	-	<i>Power Supply, 0V (GND)</i>
2	Vcc	-	<i>Power Supply, 5V $\pm$ 10%</i>
3	Vee	-	<i>Power Supply, for LCD Drive</i>
4	RS	H/L	<i>H: Data Input L: Instruction input</i>
5	R/W	H/L	<i>H: Read L: Write</i>
6	E	H, 	<i>Enable Signal</i>
7	D0	H/L	<i>Data Bus</i>
8	D1	H/L	<i>Data Bus</i>
9	D2	H/L	<i>Data Bus</i>
10	D3	H/L	<i>Data Bus</i>
11	D4	H/L	<i>Data Bus</i>
12	D5	H/L	<i>Data Bus</i>
13	D6	H/L	<i>Data Bus</i>
14	D7	H/L	<i>Data Bus</i>
15	V+ BL	-	<i>Back Light Supply, 4-4.2V, 50-200mA</i>
16	V- BL	-	<i>Back Light Supply, 0V (GND)</i>

Sumber: *Seiko LCD M1632 User Manual*. Japan: Seiko Instrument Inc. 1987.
hal 2.