BAB II
LANDASAN TEORI

2.1 Internet

Istilah Internet berasal dari kata *internetworking*. *Internetworking* sendiri bisa diartikan sebagai *network* dari *network*, yang berarti kumpulan dari jaringan-jaringan yang menghubungkan komputer dari sistem yang berbeda-beda. Jadi, secara singkat dapat dikatakan bahwa Internet adalah kumpulan berbagai macam sistem jaringan komputer di dunia yang terkoneksi satu sama lain dan dapat saling berkomunikasi satu sama lain.

Menurut sejarahnya, Internet lahir pada era 60-an atau tepatnya tahun 1969. Internet berawal dari proyek risek yang disponsori oleh DARPA (*Defense Advanced Research Projects Agency*). Riset ini bertujuan untuk mengembangkan suatu jaringan komputer yang:

- Bekerja secara transparan, melalui bermacam-macam jaringan komunikasi data yang terhubung satu dengan lainnya.
- Tahan terhadap berbagai gangguan (bencana alam, serangan nuklir, dan lain-lain).

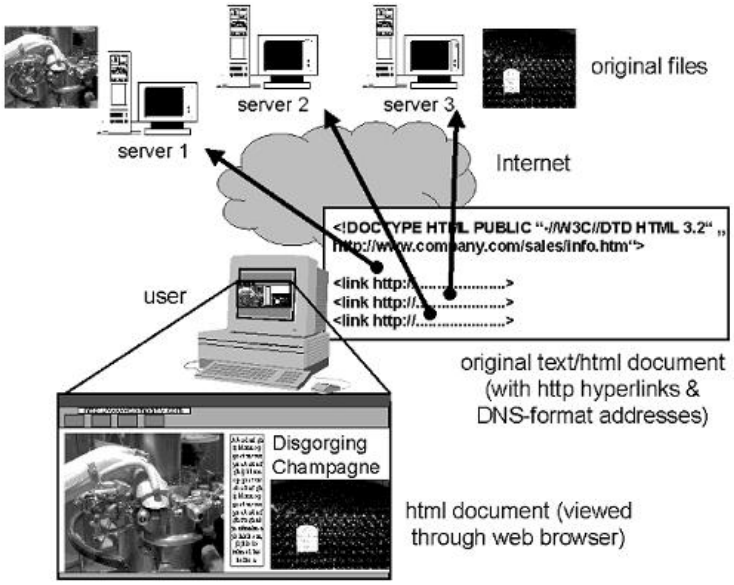
Pengembangan jaringan ini ternyata suskses dan melahirkan ARPANET (*Advanced Research Agency Network*). Jadi, ARPANET dapat dikatakan sebagai jaringan komputer yang pertama di dunia. [18]

2.2 *World Wide Web* (www)

World Wide Web (www) adalah ‘perpustakaan bersama’ dengan informasi yang sangat banyak, yang tersimpan di jutaan komputer yang berbeda di seluruh dunia dan mudah diakses dari tempat lain di dunia melalui internet.

World Wide Web (www) merupakan kombinasi dari empat teknologi yang berbeda yang memungkinkan pencarian dan penjelajahan informasi di komputer lain dengan mudah. Empat teknologi yang muncul pada tahun 1990 untuk menciptakan *World Wide Web* adalah: *domain name system* (*DNS*), *hypertext transfer protocol* (*http*), *hypertext markup language* (*html*) dan peramban web. [12]

Pada Gambar 2.1 berikut merupakan ilustrasi beserta penjelasan hal-hal yang menyangkut *World Wide Web.*



Gambar 2.1. Worldwide Web (www): kombinasi dari DNS, http, html dan peramban web.

[Sumber: Data Networks, IP and the Internet P.453]

- Domain name system (DNS) – yang memungkinkan alamat dari Worldwide Web (www) direpresentasikan dalam bentuk ‘ramah pengguna’ (misal www.company.com) daripada menampilkannya sebagai angka string dan ‘titik-titik’ yang membentuk alamat Internet (misalnya 37.168.153.1);
- *Hypertext transfer protocol* (*http*) – yang memungkinkan dokumen dan file komputer yang berada pada *server* yang berbeda dapat dihubungkan satu sama lain dengan cara *hiperlinks*;

- *Hypertext markup language* (*html*) – yang memungkinkan *hyperlinks* untuk ditulisi kedalam serupa dokumen teks komputer dimaksudkan untuk dilihat oleh manusia;
- Web browser– yang memungkinkan dukomen *web* berbasis *html* yang kompleks dilihat sebagai halaman *web*, walaupun blok-blok individual teks dan gambar yang tampil pada halaman *web* sudah tersusun dengan *hyperlinks* dari komputer yang berbeda yang tersebar di seluruh dunia.

2.3 *Server* Web

Server web dapat merujuk baik pada perangkat keras ataupun perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web dalam layanan ke pengguna dengan menggunakan aplikasi tertentu seperti peramban *web*.

Fungsi utama sebuah *server web* adalah untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Disebabkan sebuah halaman *web* dapat terdiri atas berkas teks, gambar, video, dan lainnya pemanfaatan *server web* berfungsi pula untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web yang terkait; termasuk di dalamnya teks, gambar, video, atau lainnya.

Pengguna, biasanya melalui aplikasi pengguna seperti peramban *web*, meminta layanan atas berkas ataupun halaman *web* yang terdapat pada sebuah *server web*, kemudian *server* sebagai manajer layanan tersebut akan merespon balik dengan mengirimkan halaman dan berkas-berkas pendukung yang dibutuhkan, atau menolak permintaan tersebut jika halaman yang diminta tidak tersedia.

Saat ini umumnya *server web* telah dilengkapi pula dengan mesin penerjemah bahasa skrip yang memungkinkan server web menyediakan layanan situs web dinamis dengan memanfaatkan pustaka tambahan seperti PHP, ASP. [9][10]

2.4 Layanan Web Hosting

Layanan *web hosting* adalah salah satu jenis layanan internet hosting yang memungkinkan individu dan organisasi untuk membuat website mereka dapat diakses melalui *World Wide Web*. *Web host* adalah perusahaan yang menyediakan ruang pada *server* yang dimiliki atau disewa untuk digunakan oleh klien, serta menyediakan konektivitas internet, biasanya di sebuah pusat data.

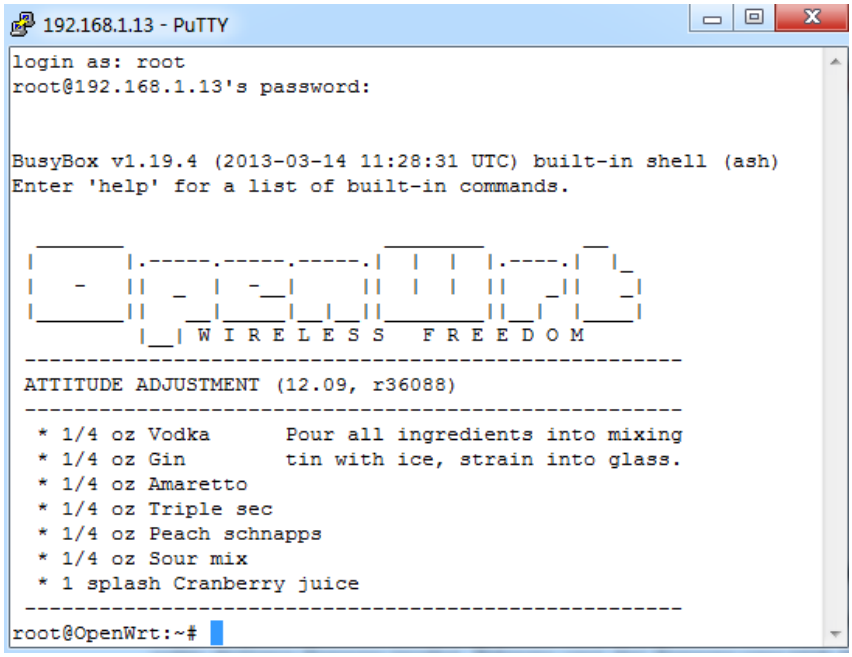
Ruang lingkup layanan *web hosting* sangat bervariasi. Yang paling mendasar adalah halaman *web* dan *hosting file* skala kecil, di mana *file* dapat di-unggah melalui *File Transfer Protocol* (FTP) atau antarmuka *Web*. [11]

2.5 OpenWrt

OpenWrt adalah sistem operasi / sistem operasi tertanam berbasis kernel Linux, dan utamanya digunakan pada perangkat tertanam untuk merutekan lalu lintas jaringan. Komponen utamnya adalah kernel Linux, util-linux, uClibc dan BusyBox. Semua komponen telah dioptimasi dalam hal ukuran, agar cukup kecil untuk pas pada ketersediaan media penyimpanan dan memori yang terbatas dalam *router* rumah.

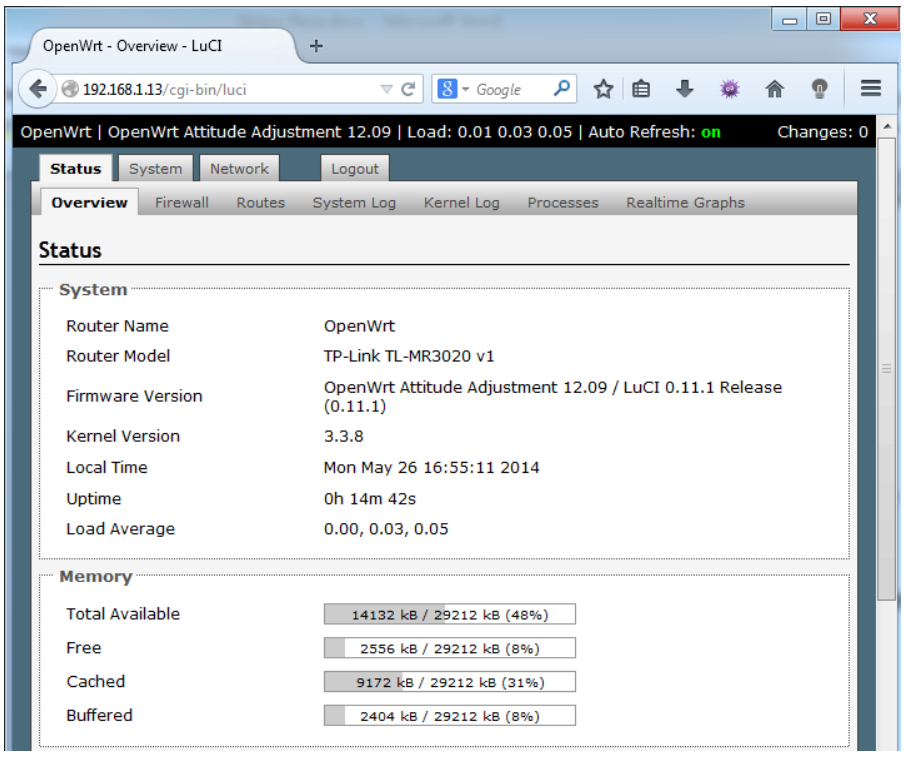
Openwrt dikonfigurasi menggunakan antarmuka baris perintah (ash shell), atau antarmuka *web* (LuCI). Terdapat sekitar 3500 paket perangkat lunak tambahan tersedia untuk instalasi melalu paket sistem manajemen paket *opkg*. [6]

Pada Gambar 2.2 berikut merupakan tampilan antarmuka baris perintah (ash shell) pada openWrt.



Gambar 2.2. Antarmuka Baris Perintah (ash shell)

Pada Gambar 2.3 berikut adalah tampilan antarmuka sistem Openwrt menggunakan web (LuCi).



Gambar 2.3. Antarmuka Web (LuCI)

Terdapat beberapa versi dari OpenWRT dibedakan berdasarkan waktu rilis versi tersebut diantaranya adalah sebagai berikut. [7]

1. White Russian

   White Russian 0.9 adalah versi OpenWRT yang stabil pertama yang dirilis pada bulan Januari 2007.

2. Kamikaze

   Kamikaze dirilis pertama kali dengan versi Kamikaze 7.06 pada bulan Juni 2007 dan mengalami beberapa penambahan fitur dan pengurangan bug pada sistem sehingga terakhir dirilis dengan veri Kamikaze 8.09.2 pada bulan Januari 2010.

3. Backfire

   Backfire dirilis dengan dua versi awalnya dirilis dengan versi Backfire 10.03 pada bulan April 2010 dan diperbaharui dengan versi Backfire 10.03.1 pada bulan Desember 2011.

4. Attitude Adjustment

   Attitude Adjustment 12.09 adalah OpenWRT versi stabil terbaru dirilis pada bulan April 2013.

5. Barrier Breaker

   Barrier Breaker adalah versi terakhir dan dapat digunakan tetapi masih dalam tahap pengembangan.

2.6 Aplikasi Berbasis Web

Dalam rekayasa perangkat lunak, suatu aplikasi web (bahasa Inggris: *web application* atau sering disingkat *webapp*) adalah suatu aplikasi yang diakses menggunakan web browsermelalui suatu jaringan seperti Internet atau intranet.

Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung web browser(seperti HTML, JavaScript, AJAX,

Java, dll) dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi. [2]

2.7 HTML

HyperText berarti teks yang saling terhubung dengan cara *hyperlink* sedangakan *markup* berarti penanda dalam file html dengan *tag*. HTML adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. [4]

Pada saat peramban menampilkan suatu halaman web, peramban akan membaca teks-teks pada dokumen HTML dan mencari suatu kode yang khusus yang disebut Tag. Tag ini dinyatakan dengan tanda lebih kecil dan lebih besar (<>). Tag ini biasanya merupakan suatu pasangan yang disebut tag awal dan tag akhir. Tag awal dinyatakan dalam bentuk <nama tag> sedangkan akhir tag dinyatakan dalam bentuk </nama tag>.

Dalam dokumen HTML, elemen dibagi menjadi dua kategori utama yaitu elemen-elemen <HEAD> yang memberikan informasi tentang dokumen tersebut, sepeti judul dokumen atau hubungan dengan dokumen lain serta elemen-elemen <BODY> yang menentukan bagaimana isi suatu dokumen ditampilkan oleh peramban, seperti paragraf, list, tabel, dll. Struktur standar file HTML adalah sebagai berikut**.** [19]

```
<!doctype html>
<html>
   <head>
   </head>
   <body>
   </body>
</html>
```

2.8 PHP

PHP adalah kependekan dari *PHP: HyperText Preprocessor*, merupakan bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML, dikenal juga sebagai bahasa pemrograman pada sisi server (*server-side scripting*), dan membuat halaman web lebih dinamis.

PHP membuat dokumen HTML secara *'on the fly'* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan *editor* teks atau *editor* HTML. Dengan menggunkan PHP maka *maintenance* suatu situs web menjadi lebih mudah. Proses *update* data dapat dilakukan dengan menggunakan aplikasi yang dibuat dengan menggunakan *script* PHP.

Nama *file script* PHP sejak versi 4.0 menggunakan ekstensi *file* **.php** sebagai standar. *Script* PHP diawali dengan tag **<*?php*** kemudian diakhiri dengan ***?***>. Setiap perintah/statement harus diakhiri dengan menggunakan tanda titik koma (;). Umumnya setiap statement dituliskan dalam satu baris. Berikut contoh Program Hello World yang ditulis menggunakan PHP. [8][17]

```
<?php
   echo "Hello World";
?>
```

2.9 CSS

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemograman.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. [3]

Kode CSS dapat dituliskan di dalam dokumen HTML maupun terpisah dengan menyimpan semua kode CSS kedalam file berekstensi *.css* lalu dikaitkan nama file tersebut di bagian Header HTML contohnya ***<link rel="stylesheet" href="css/style.css">***. Penulisan kode CSS terbagi dalam tiga bagian yaitu:

```
selector { property: value }
```

- *Selector* : menunjukkan target elemen HTML yang akan diberi *style*
- *Property* : aspek dari elemen yang akan dimodifikasi tampilannya
- *Value* : nilai yang diisikan pada *property*

Contoh berikut memberi *style* untuk elemen <*p*> pada dokumen HTML

```
p { color:  #5A5A5A }
```

2.10 jQuery

Jquery merupakan salah satu dari sekian banyak *framework* yang digunakan dan memiliki pengguna yang paling banyak. JQuery merupakan pustaka (*library*) Javascript yang dibangun untuk mempercepat dan memperingkas serta menyederhanakan manipulasi dokumen HTML, penanganan event, animasi, dan interaksi Ajax untuk mempercepat pengembangan web. [14]

Dengan Jquery, developer akan dimanjakan dengan suatu pemrograman JavaScript yang sangat sederhana jika dibandingkan dengan *native* Javascript. Adapun kemampuan yang dimiliki library Jquery antara lain:

- Mampu mengubah tampilan bagian halaman tertentu
- Mampu mengubah isi dari halaman

- Mampu merespon interaksi user dalam halaman
- Mampu menambahkan animasi ke halaman
- Mampu mengambil informasi dari server tanpa me-*refresh* seluruh halaman
- Mampu menyederhanakan kode JavasScript lainnya.

Gambar 2.4 berikut merupakan logo dari framework Jquery.



Gambar 2.4. Logo JQuery

[http://id.wikipedia.org/wiki/Berkas:JQuery_logo.svg]

2.10.1 Sintaks Jquery

Sintaks secara umum Jquery terdiri dari nama fungsi JQuery, parameter, dan *event*. Fungsi JQuery secara umum diwakili oleh tanda dolar ($), sebagaimana *framework* lain. Parameter merupakan sebuah s*elector*, yaitu suatu petunjuk untuk memilih elemen HTML. Sedangkan *event* adalah aksi yang diberikan pada elemen yang dipilih. Berikut bentuk umum dari penulisan fungsi JQuery.

```
$("selector").action()
```

Keterangan:

- $: tanda dolar untuk mendefinisikan fungsi JQuery
- *selector* : menunjukkan elemen yang dipilih, bisa berupa nama *tag*, nama *class*, dan *id* elemen
- *action* : aksi yang akan diberikan terhadap elemen yang dipilih

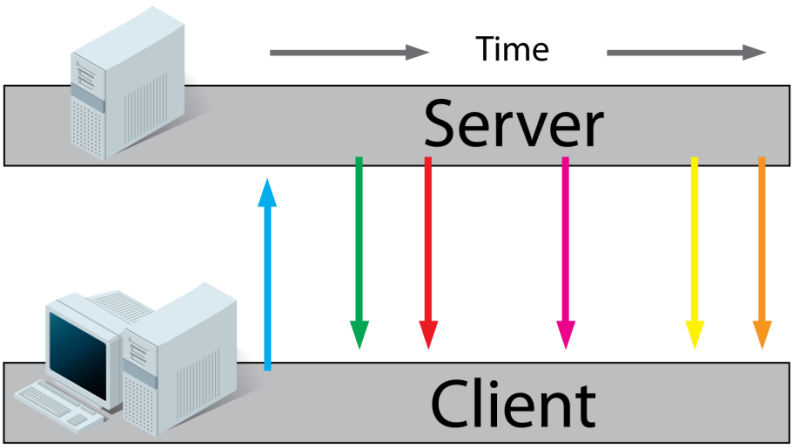
Contoh sederhana menyembunyikan semua paragraf atau konten dari *tag* <***p***> :

```
$("p").hide()
```

2.11 Server-Sent Events

Server-Sent Events adalah teknologi pada browser untuk mendapatkan *update* otomatis dari *server* melalui koneksi HTTP. *Server-Sent Events* memungkinkan aliran event data berbasis text dari server ke klien secara efisien contohnya notifikasi atau update yang dihasilkan di server secara *real-time*. Untuk mencapai tujuan ini, SSE memperkenalkan dua buah komponen: EventSource yaitu *interface* terbaru di dalam *browser*, yang memungkinkan klien menerima *push notification* dari server sebagai event DOM, dan format data "*stream event*", yang digunakan untuk mengirimkan update individual.

Kombinasi dari API Eventsource pada browser dan format data event stream yang didefinisakan dengan baik merupakan hal yang membuat SSE merupakan alat yang efisien dan sangat diperlukan untuk menangani data yang real-time di browser. [13]



Gambar 2.5. Ilustrasi SSE

[http://stackoverflow.com/questions/11077857/what-are-long-polling-websockets-server-sent-events-sse-and-comet]

Gambar 2.5 di atas merupakan ilustrasi aliran data dimana *client* mula-mula melakukan *request* ke server, kemudian server secara terus menerus mengalirkan data ke *client*.

2.11.1 API Eventsource

Antarmuka pemrograman aplikasi (Inggris: *application programming interface* disingkat API) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. [1]

Dalam permrograman komputer, sebuah API menetapkan bagaimana suatu komponen *software* harus berinteraksi dengan yang lain.

Antarmuka eventsource meringkaskan semua pembentukan koneksi low-level dan menguraikan pesan dibelakang sebuah API browser yang sederhana. Untuk memulai menggunakan API eventsource pertama yang dibutuhkan adalah menentukan URL sumber SSE event stream dan mendaftarkan *listeners event* pada sebuah objek.

```
var source=new EventSource("/path/to/stream-url");  1

source.onopen = function () { ... };  2
source.onerror = function () { ... };  3

source.addEventListener("foo", function (event) {  4
processFoo(event.data);
});

source.onmessage = function (event) {  5
      log_message(event.id, event.data);

      if (event.id == "CLOSE") {  6
          source.close();
      }
}
```

1. Membuka koneksi SSE terbaru
2. Callback opsional, dipanggil jika koneksi berhasil terhubung
3. Callback opsional, dipanggil jika koneksi gagal
4. Berlangganan (subscribe) pada event yang bertipe “foo”
5. Berlangganan (subscribe) pada semua event tanpa tipe yang eksplisit
6. Tutup koneksi SSE jika sever mengirimkan ID pesan “CLOSE”

2.12 Pemrograman Shell

Shell adalah program (penterjemah perintah) yang menjembatani *user* dengan sistem operasi dalam hal ini kernel (inti sistem operasi), umumnya shell menyediakan prompt sebagai *user interface*, tempat dimana *user* mengetikkan perintah-perintah yang diinginkan baik berupa perintah *internal shell* (*internal command*), ataupun perintah eksekusi suatu file progam (*external command*), selain itu shell memungkinkan *user* menyusun sekumpulan perintah pada sebuah atau beberapa file untuk dieksekusi sebagai program.

Pemrograman shell yaitu menyusun atau mengelompokkan beberapa perintah shell (*internal* ataupun *external command*) menjadi kumpulan perintah yang melakukan tugas tertentu sesuai tujuan penyusunnya.

Kelebihan shell di linux dibanding sistem operasi lain adalah bahwa shell di linux memungkinkan kita untuk menyusun serangkaian perintah seperti halnya bahasa pemrograman (interpreter language), melakukan proses I/O, menyeleksi kondisi, looping, membuat fungsi, dsb. proses-proses yang umumnya dilakukan oleh suatu bahasa pemrograman Jadi dengan shell di linux kita dapat membuat program seperti halnya bahasa pemrograman, untuk pemrograman shell pemakai unix atau linux menyebutnya sebagai script shell. [15]

2.12.1 Macam-macam Shell

Tidak seperti sistem operasi lain yang hanya menyediakan satu atau 2 shell, sistem operasi dari keluarga unix misalnya linux sampai saat ini dilengkapi oleh banyak shell dengan kumpulan perintah yang sangat banyak, sehingga

memungkinkan pemakai memilih shell mana yang paling baik untuk membantu menyelesaikan pekerjaannya, atau dapat pula berpindah-pindah dari shell yang satu ke shell yang lain dengan mudah, beberapa shell yang ada di linux antara lain:

- Bourne shell(sh),
- C shell(csh),
- Korn shell(ksh),
- Bourne again shell(bash),
- dsb.

2.12.2 Contoh Shell Scrpit

Pada contoh berikut akan menampilkan penggunaan salah satu internal shell (internal command) *date* yang berfungsi menampilkan waktu sistem saat ini.

```
root@OpenWrt:~# date
Wed Apr 23 01:17:03 WIT 2014
```

Berikut merupakan contoh untuk menampilkan karakter apapun dengan perintah *echo* diikuti dengan karakter yang diapit tanda petik dua (“”).

```
root@OpenWrt:~# echo "Teknik Komputer"
Teknik Komputer
```

Perintah-perintah tersebut dapat juga disimpan ke dalam sebuah file dan dapat dijalankan dengan hanya memanggil file berisi perintah-perintah tersebut. Langkahnya adalah sebagai berikut.

1. Masuk ke editor, dapat menggunakan vi, pico, emacs, dsb.
2. Isi dengan perintah-perintah yang akan dijalankan

```
#!/bin/sh
date
echo "Teknik Komputer"
```

3. Simpan dengan nama *tes.sh* pada direktori /root/sh/

4. Ubah permission file tes menggunakan *chmod*

```
root@OpenWrt:~# chmod 755 /sh/tes.sh
```

5. Jalankan file tersebut maka akan manghasilkan output

```
root@OpenWrt:~# tes.sh
Wed Apr 23 01:17:03 WIT 2014
Teknik Komputer
```

2.13 Cronjob

Software utilitas cronjob bertugas untuk menjadwalkan pekerjaan berdasarkan waktu pada sistem operasi komputer *Unix*. Cron digerakan oleh file crontab (tabel cron), file konfigurasi yang menspesifikasikan perintah-perintah shell apa saja yang akan dijalankan secara berkala pada jadwal tertentu. Cronjob dibutuhkan untuk menjalankan tugas tertentu pada dalam interval waktu tertentu. [16]

Format penulisan dalam file crontab ditulis dengan field-field yang dipisahkan dengan spasi, format penulisannya adalah sebagai berikut

```
MIN HOUR DOM MON DOW CMD
```

Tabel 2.1. Keterangan Field Crontab

Field	Definisi	Isi
MIN	Menit	0-59
HOUR	Jam	0-23
DOM	Hari dalam bulan	1-31
MON	Bulan	1-12 atau JAN-DEC
DOW	Hari dalam minggu	0-6 atau SUN-SAT
CMD	Perintah (*command*)	Perintah apa saja yang akan dieksekusi

Isi dari field selain dengan nilai-nilai yang terdapat pada tabel 1, dapat juga diisi dengan karakter bintang (*) yang berarti semua unit. Misal diisikan pada field *MIN* maka berarti "setiap menit".

Berikut ini merupakan contoh untuk menjalankan perintah setiap menit secara terus menerus. Dalam hal ini kumpulan perintah disimpan di dalam file *perintah.sh*.

```
* * * * * perintah.sh
```

Untuk mengerjakan perintah secara terus-menerus setiap lima menit maka pada field *MIN* dapat ditambahkan /5 setelah karakter bintang (*) contohnya sebagai berikut.

```
*/5 * * * * perintah.sh
```

Contoh berikut menjadwalkan pekerjaan pada waktu yang spesifik. Misalnya melakukan backup data setiap tanggal 10 Juni jam 08:30 AM.

```
30 08 10 06 * backup.sh
```

2.14 CURL

Curl adalah alat bantu (*tool*) untuk mentransfer data dari atau ke server, menggunakan salah satu protokol yang didukung (HTTP, HTTPS, FTP, FTPS, GOPHER, DICT, TELNET, LDAP atau FILE). Perintah ini dirancang untuk bekerja tanpa interaksi pengguna. [5]

Curl terdiri dari kumpulan perintah (*shell command*) yang dapat diketikan pada prompt. Format penggunaan curl adalah dengan mengetikkan curl diikuti option(bisa tidak digunakan) dan URL.

```
curl [option] [URL...]
```

Berikut ini adalah contoh penggunaan curl untuk mengambil data yang berada di *web server*. Hasilnya curl akan menampilkan isi dari URL webpage yang telah di-*request*.

```
root@OpenWrt:~# curl http://myhome.web.id/data/data_dari_user.txt
1off2off
```

Sedangkan untuk mengirim data ke webserver maka perlu ditambahkan option --*data* diikuti data yang akan dikirimkan.

```
root@OpenWrt:~# curl --data "suhu=24"
http://www.myhome.web.id/proses_data_rumah.php
```