

## 2. LANDASAN TEORI

Bab ini berisi mengenai teori-teori dan informasi dasar yang mendukung perancangan dan pembuatan sistem akses *database* prediksi nilai saham. Di dalamnya terdapat informasi mengenai sistem sejenis yaitu Yahoo Finance. Selain itu juga terdapat informasi cara penggunaan PayPal API, informasi seputar saham di Indonesia, dan informasi pendukung lainnya.

### 2.1. Yahoo Finance

Yahoo Finance adalah sebuah *website* dari Yahoo yang menyediakan banyak informasi seputar ekonomi dan bisnis. Yahoo Finance telah menjadi sebuah *website* nomor satu di Amerika Serikat, yang menyediakan informasi berita ekonomi dan penelitian mengenai perekonomian sejak Januari 2008. Pada tahun 2012, pengunjung Yahoo Finance sudah mencapai 37,5 juta pengunjung tiap bulannya. (“Yahoo Finance”, para. 1)

Banyak layanan informasi yang disediakan oleh Yahoo Finance, dan bisa dikatakan bahwa Yahoo Finance memiliki informasi keuangan yang lengkap mulai dari berita, nilai saham, nilai tukar mata uang, dan laporan keuangan dari perusahaan-perusahaan. Kini Yahoo Finance tidak hanya melayani informasi di Amerika Serikat saja, namun juga beberapa Negara besar di Amerika Latin, Eropa, dan Asia. Indonesia menjadi salah satu negara Asia yang terdapat layanan Yahoo Finance di dalamnya. Salah satu contohnya adalah informasi mengenai saham di Indonesia.

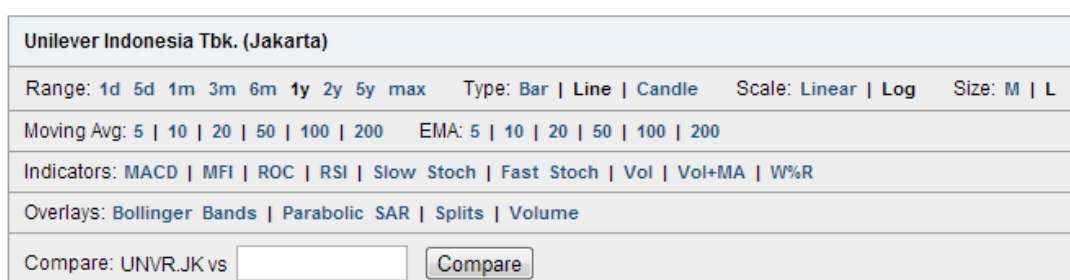
Informasi mengenai nilai harga saham pada Bursa Efek Jakarta terdapat di dalam Yahoo Finance, sehingga pengunjung yang ingin mengetahui perubahan nilai harga saham dapat mengakses secara *online*. Informasi saham yang ada di dalam Yahoo Finance yaitu informasi nilai indeks harga saham gabungan (IHSG), serta informasi nilai harga saham untuk masing-masing perusahaan. Fitur *Company Profile* tidak tersedia untuk semua perusahaan yang ada di Indonesia. Beberapa komponen yang digunakan untuk analisa, yaitu *Analyst Option*, *Analyst Estimates* juga tidak *up-to-date*.



Gambar 2.1. Tampilan dari informasi saham perusahaan di dalam Yahoo Finance.

Sumber: Yahoo Finance - Univeler Indonesia Tbk (2013, para. 1)

Untuk membuka informasi dari saham perusahaan, maka *user* harus memasukkan kode dari saham tersebut. Kode saham perusahaan yang ada di dalam bursa efek Jakarta diberi akhirkan .JK. Hal ini untuk membedakan antara saham perusahaan di Indonesia dengan yang ada di negara lain, karena ada saham perusahaan yang memiliki kode saham yang sama tetapi berbeda bursa efek. Informasi dari saham perusahaan yang ditampilkan sangat lengkap. Mulai dari nilai *close*, *open*, *volume*, hingga informasi nilai *bid* dan *ask*, serta informasi lain yang menunjukkan kesimpulan dari saham seperti *day's range*, *52wk range*, dan *average volume*. Contoh tampilan dari saham perusahaan di dalam Yahoo Finance terlihat pada Gambar 2.1.



Gambar 2.2. Pilihan menu dari grafik yang disediakan oleh Yahoo Finance.

Sumber: Yahoo Finance - Univeler Indonesia Tbk (2013, para. 1)

Selain menyediakan informasi secara tertulis, Yahoo Finance juga menyediakan grafik untuk menganalisa saham perusahaan. Grafik pada Yahoo Finance mempunyai banyak pilihan untuk mengatur tampilan dari grafik seperti yang terlihat pada Gambar 2.2. Pilihan tersebut antara lain,

- *Range*, digunakan untuk mengatur rentang waktu.
- *Type*, digunakan untuk memilih jenis grafik yang diinginkan.
- *Scale*, digunakan untuk memilih skala dari grafik.
- *Size*, untuk memilih ukuran dari grafik.
- *Moving Avg*, digunakan untuk menampilkan *Simple Moving Average* (SMA).
- *EMA*, digunakan untuk menampilkan *Exponential Moving Average* (EMA).
- *Indicators*, digunakan untuk memilih fitur-fitur analisa dari saham, seperti *Relative Strength Index* (RSI) dan *Moving Average Convergence Divergence* (MACD)
- *Overlays*, digunakan untuk menampilkan pilihan yang tersedia secara *overlay* pada grafik.
- *Compare*, digunakan untuk membandingkan saham yang satu dengan yang lain.

| Prices       |           |           |           |           |            |            |
|--------------|-----------|-----------|-----------|-----------|------------|------------|
| Date         | Open      | High      | Low       | Close     | Volume     | Adj Close* |
| May 6, 2013  | 27,000.00 | 27,000.00 | 26,100.00 | 27,000.00 | 2,555,000  | 27,000.00  |
| May 3, 2013  | 25,900.00 | 26,250.00 | 25,350.00 | 25,900.00 | 2,870,500  | 25,900.00  |
| May 2, 2013  | 25,250.00 | 25,500.00 | 25,200.00 | 25,250.00 | 2,647,000  | 25,250.00  |
| May 1, 2013  | 25,700.00 | 25,850.00 | 24,800.00 | 25,700.00 | 2,122,000  | 25,700.00  |
| Apr 30, 2013 | 26,250.00 | 26,250.00 | 22,900.00 | 26,250.00 | 13,435,000 | 26,250.00  |
| Apr 26, 2013 | 22,750.00 | 23,000.00 | 22,750.00 | 22,750.00 | 1,804,500  | 22,750.00  |
| Apr 25, 2013 | 23,000.00 | 23,000.00 | 22,850.00 | 23,000.00 | 1,390,000  | 23,000.00  |
| Apr 24, 2013 | 23,000.00 | 23,000.00 | 22,600.00 | 23,000.00 | 1,380,500  | 23,000.00  |

Gambar 2.3. Daftar data perubahan harga dari saham.

Sumber: Yahoo Finance - Univeler Indonesia Tbk (2013, para. 1)

Yahoo Finance juga menyediakan informasi mengenai perubahan harga saham setiap harinya. Informasi tersebut dapat ditampilkan secara tabel, seperti

yang terlihat pada Gambar 2.3. Selain melihat data dalam bentuk tabel, *user* dapat mengunduh data tersebut berupa *file* csv. *File* unduhan inilah yang nantinya akan dimanfaatkan oleh sistem di dalam Tugas Akhir sebagai sumber data.

Untuk mengambil *file* tersebut, caranya adalah dengan melakukan klik terhadap tombol donwload yang tersedia, atau dengan memasukkan URL tertentu pada browser. URL tersebut memiliki beberapa parameter yang digunakan untuk mengatur banyaknya data yang ingin diunduh. Contoh URL yang digunakan untuk mengunduh data saham adalah sebagai berikut

```
http://ichart.finance.yahoo.com/table.csv?s=BBCA.JK&a=1&b=1&c=2007&d=1&e=1&f=2013&g=d&ignore=.csv.
```

Gambar 2.4. URL untuk mengunduh data saham dari Yahoo Finance.

Parameter-parameter yang digunakan antara lain,

- Parameter S digunakan untuk menentukan kode dari saham yang ingin diunduh. Kode saham adalah kode saham dari Yahoo Finance sehingga harus diberi akhiran .JK
- Parameter A, B, C, D, E, dan F merupakan rentang waktu dari data yang ingin diunduh. A adalah bulan awal, B adalah hari awal, C adalah tahun awal. D adalah bulan akhir, E adalah hari akhir, dan F adalah tahun akhir. Data bulan dimulai dari 0 untuk Januari dan 11 untuk Desember. Sehingga pada URL yang ditunjukkan di atas, data yang di ambil adalah data dari tanggal 1 Februari 2007 hingga 1 Februari 2013.
- Parameter G merupakan jenis data yang ingin diunduh. Apakah data tersebut harian, mingguan, atau tahunan.

## 2.2. Saham di Indonesia

Saham (*stock*) dapat didefinisikan sebagai tanda penyertaan atau pemilikan seseorang atau badan dalam suatu perusahaan atau perseroan terbatas. Saham berwujud selembur kertas yang menerangkan bahwa pemilik kertas tersebut adalah pemilik perusahaan yang menerbitkan surat berharga tersebut. Porsi

kepemilikan ditentukan oleh seberapa besar penyertaan yang ditanamkan di perusahaan tersebut. (Darmadji, 5)

Saham di Indonesia dapat diperjual-belikan di dalam pasar saham yaitu biasa disebut dengan Bursa Efek. Bursa Efek adalah lembaga atau perusahaan yang menyelenggarakan atau menyediakan fasilitas sistem (pasar) untuk mempertemukan penawaran jual dan beli efek antar berbagai perusahaan atau perorangan yang terlibat dengan tujuan memperdagangkan efek perusahaan-perusahaan yang telah tercatat di Bursa Efek. (Darmadji, 34)

Di Indonesia terdapat dua buah Bursa Efek, yaitu Bursa Efek Jakarta dan Bursa Efek Surabaya. Masing-masing bursa memiliki jam kerja atau jam buka saham sebagai berikut,

- Senin-Kamis  
Sesi I jam 09.30-12.00  
Sesi II Jam 13.30-16.00
- Jumat  
Sesi I Jam 09.30-11.30  
Sesi II Jam 14.00-16.00

Bursa saham akan tutup pada hari Sabtu dan Minggu, serta pada hari-hari libur nasional dan keagamaan yang ditetapkan oleh Pemerintah Indonesia. Bursa juga akan libur pada hari Cuti Bersama yang juga ditetapkan oleh Pemerintah.

Di pasar traditional, sebelum adanya transaksi atau sebelum harga terbentuk terjadi suatu proses tawar-menawar. Hal tersebut juga terjadi di pasar saham, harga dari saham juga melalui proses tawar-menawar yang disebut dengan istilah *bid* dan *offer*. Harga saham yang ditawarkan oleh penjual untuk dijual disebut dengan *offer*. Harga saham yang ditawarkan oleh pembeli untuk dibeli disebut dengan *bid*. Proses permintaan akan saham inilah yang mengakibatkan terjadinya perubahan harga saham tertentu di dalam Bursa Efek.

Harga sebuah saham dapat berubah naik dan turun dalam hitungan menit maupun dalam hitungan detik. Hal ini dimungkinkan karena banyaknya pesanan yang dimasukkan ke dalam sistem JATS (*Jakarta Automated Trading Systems*). Di lantai perdagangan BEJ terdapat lebih dari 400 terminal komputer dimana para *floor trader* dapat memasukkan pesanan yang ia terima dari nasabah. Masukkan

pesanan-pesanan tersebut baik jual maupun beli akan berpotensi terjadinya transaksi pada harga tertentu. Di BEJ (Bursa Efek Jakarta) terdapat lebih dari 330 saham yang tercatat dan dapat diperdagangkan oleh investor lokal maupun investor mancanegara. (Darmadji, 102)

Istilah-istilah penting yang berkaitan dengan saham adalah sebagai berikut,

- *Previous Price* menunjukkan harga pada penutupan hari sebelumnya.
- *Open* atau *Opening Price* menunjukkan harga pertama kali pada saat pembukaan Sesi I perdagangan.
- *High* atau *Highest Price* menunjukkan harga tertinggi atas suatu saham yang terjadi sepanjang perdagangan pada hari tersebut.
- *Low* atau *Lowest Price* menunjukkan harga terendah atas suatu saham yang terjadi sepanjang perdagangan pada hari tersebut.
- *Last Price* menunjukkan harga terakhir yang terjadi atas suatu saham.
- *Change* menunjukkan selisih antara harga pembukaan dengan harga terakhir yang terjadi.
- *Close* atau *Closing Price* menunjukkan harga penutupan suatu saham. *Closing Price* suatu saham dalam satu hari perdagangan pada akhir Sesi II, yaitu Pukul 16.00 (pukul 4.00 sore).

Keuntungan membeli saham yaitu, mendapatkan deviden (*long term*) atau mendapatkan capital gain (*short term*) membeli saham ketika harga saham rendah kemudian menjualnya kembali saat harga tinggi. Biasanya orang-orang lebih memilih melakukan *trading* saham, atau memainkan saham dalam waktu pendek (*short term*) untuk mendapatkan *capital gain*.

### **2.3. Paypal**

PayPal adalah sebuah perusahaan yang bergerak dalam bidang *e-commerce* yang menyediakan jasa pembayaran dan pengiriman uang antar pemilik *account* PayPal. Ada beberapa manfaat yang akan didapatkan bila kita mempunyai *account* PayPal yaitu, dengan menggunakan PayPal memungkinkan orang yang berada di belahan dunia yang lain dapat melakukan transaksi pembayaran ataupun pengiriman uang secara instan melalui internet. Selain itu, PayPal juga mempunyai beberapa fasilitas API (*Application Programmable Interface*) yang

dapat digunakan oleh orang atau organisasi yang hendak mendirikan sebuah bisnis *online*, yang menggunakan PayPal untuk proses pembayarannya.

Menurut (“PayPal Developer”, par. 1) Ada banyak tipe API yang dimiliki oleh PayPal, beberapa contoh dari API tersebut adalah sebagai berikut,

- Adaptive Payment API
- Express Checkout API
- Mass Pay
- PayPal Payment Pro

Masing-masing API tersebut memiliki fungsi yang berbeda-beda. Penjual dapat memilih salah satu dan menyesuaikannya sesuai dengan kebutuhan bisnis mereka. Dalam dunia ini ada banyak sekali mekanisme pembayaran secara *online*, Oleh karena itu PayPal menyediakan beragam API untuk menjawab kebutuhan dari penjual yang mempunyai mekanisme pembayaran yang beragam.

### **2.3.1. Express Checkout API for Digital Goods**

*Express Checkout API for Digital Goods* adalah salah satu fasilitas API yang disediakan oleh PayPal yang dikhususkan untuk transaksi pembayaran barang yang bersifat *digital*. Barang *digital* yang dimaksudkan adalah seperti, *file*, musik, *video* yang bisa diunduh, ataupun barang lain yang bersifat semu (tidak terlihat fisiknya). Yang membedakan API ini dengan API yang lainnya adalah *Express Checkout API for Digital Goods* ini lebih sederhana. *Field-field* yang ada dalam API ini lebih sedikit karena *field-field* yang memuat informasi mengenai shipping tidak tersedia dalam API ini. Barang *digital* tidak memerlukan *shipping* (pengiriman barang ke alamat tertentu), pembeli yang membeli barang *digital* dapat melakukan unduh terhadap barang yang dipesannya setelah selesai melakukan transaksi pembayaran. Hal ini juga membuat resiko barang yang dibeli tidak sampai ke tangan pembeli menjadi lebih kecil.

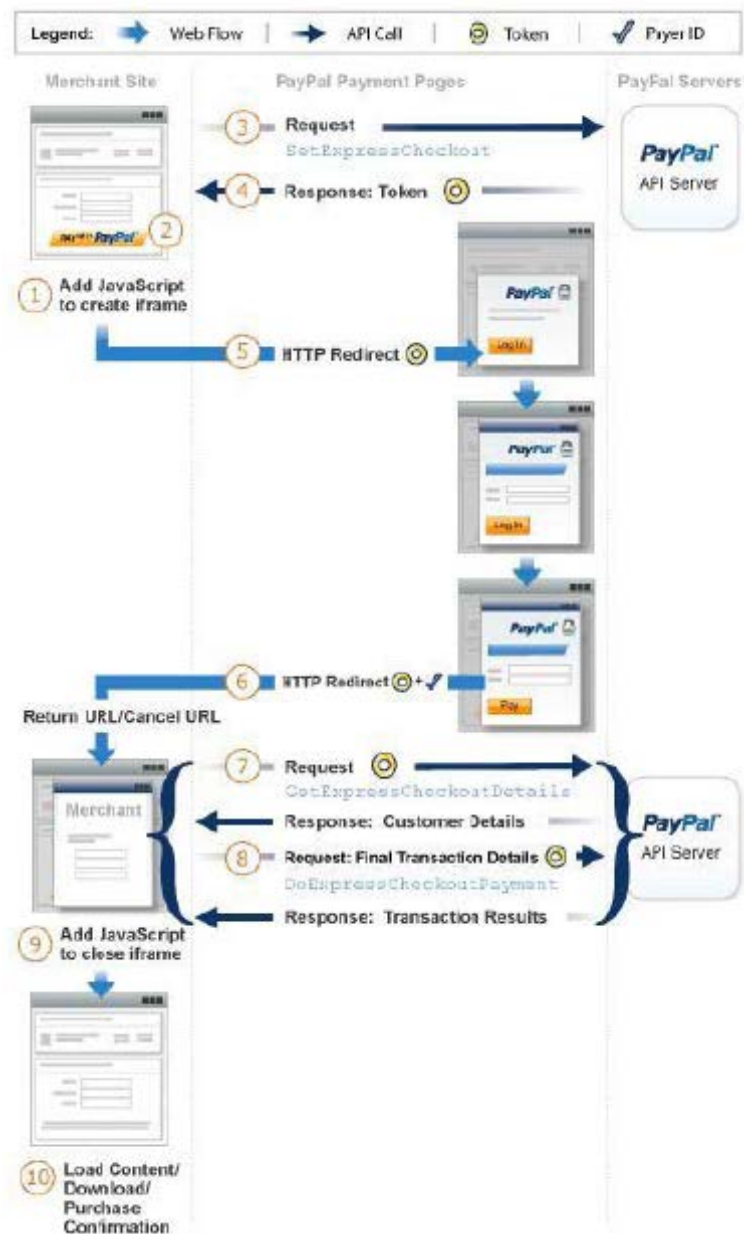


Gambar 2.5. Cara kerja *Express Checkout API* secara umum.

Sumber: Paypal (2012, p. 9)

Cara kerja Express Checkout API secara umum meliputi 3 tahap,

1. Pembeli yang berada di dalam *website* penjual melakukan click terhadap tombol *Pay with PayPal*. Di dalam tahap ini sistem akan melakukan persiapan untuk melakukan transaksi pembayaran melalui PayPal. Sistem mengumpulkan informasi dan mengirimkannya ke PayPal dengan parameter *SetExpressCheckout* (SetEC pada gambar).
2. Kemudian akan muncul form untuk melakukan pembayaran melalui PayPal. Pada form tersebut pembeli melakukan *login* terhadap *account* PayPal mereka, kemudian melakukan *review* terhadap transaksi pembayaran yang akan dilakukannya. Sistem melakukan hal tersebut dengan menggunakan parameter *GetExpressCheckout* (GetEC pada gambar).
3. Setelah pembeli selesai melakukan *review*, sistem melakukan pembayaran dengan menggunakan parameter *DoExpressCheckout* (DoEC pada gambar). Bila pembayaran berhasil maka pembeli dapat mendapatkan pesanan yang mereka beli. Penjelasan lebih lanjut tentang parameter-parameter tersebut akan dibahas pada bagian selanjutnya.



Gambar 2.6. Alur kerja dari Express Checkout API secara keseluruhan

Sumber: Paypal (2012, p. 20)

Alur kerja dari Express Checkout API adalah sebagai berikut,

1. Menambahkan *javascript* ke dalam *website*. *Javascript* diletakkan pada halaman *website* yang terdapat tombol untuk melakukan pembayaran dengan PayPal di dalamnya. *Javascript* yang diletakkan adalah *javascript dg.js* pada URL <https://www.paypalobjects.com/js/external/dg.js> yang disediakan oleh PayPal. *Javascript* ini akan mengatur alur pembayaran

melalui PayPal. Alur tersebut akan muncul dalam bentuk *lightbox* atau mini browser.

2. Meletakkan tombol yang bertuliskan *Pay with PayPal*. Tombol tersebut diperoleh dari *website* PayPal, yaitu pada URL [https://www.paypal.com/en\\_US/i/btn/btn\\_dg\\_pay\\_w\\_paypal.gif](https://www.paypal.com/en_US/i/btn/btn_dg_pay_w_paypal.gif). Penjual disarankan untuk menggunakan tombol ini karena gambar tersebut menjadi ciri khas dari PayPal, pembeli akan lebih yakin untuk membeli bila ada gambar ini dalam *website* penjual.
3. Ketika pembeli melakukan click terhadap tombol tersebut, maka sistem akan memanggil operasi API yang bernama *SetExpressCheckout*. Di dalam operasi tersebut berisi *field-field* yang harus disiapkan untuk melakukan transaksi. Penjelasan lebih lanjut ada pada bagian selanjutnya. Sistem harus menggunakan PayPal API dengan versi 65,1 ke atas agar semua proses pemanggilan fungsi API dapat berjalan dengan baik.
4. Bila operasi tersebut berhasil berjalan maka sistem akan mendapatkan balasan dari PayPal berupa *Token*. *Token* adalah sebuah *id* sementara yang khusus dan bersifat unik, yang akan digunakan selama proses transaksi pembayaran berlangsung.
5. Sistem menggunakan *token* yang telah didapat dan mengarahkan pembeli ke proses transaksi berikutnya. Pada tahap ini, sebuah mini browser akan muncul dan pembeli melakukan proses log in ke dalam *account* PayPal miliknya.
6. Setelah pembeli melakukan click terhadap tombol Pay, pembeli akan diarahkan menuju return URL yang disiapkan pada saat memanggil *SetExpressCheckout*. Bila pembeli melakukan click terhadap tombol cancel, pembeli diarahkan ke cancel URL.
7. Jika pembeli setuju melakukan pembayaran, sistem menggunakan operasi API *GetExpressCheckout* untuk memperoleh informasi-informasi yang berkaitan dengan proses pembayaran tersebut.
8. Setelah itu sistem memanggil operasi API *DoExpressCheckout* untuk melakukan pembayaran.

9. Di dalam sistem harus ada sebuah *javascript* untuk menutup *lightbox* atau mini browser yang digunakan untuk melakukan transaksi pembayaran.
10. Setelah semuanya berhasil, sistem melanjutkan proses tersebut dengan memenuhi pesanan dari pembeli, misalnya memberikan layanan unduh, dan sebagainya.

### 2.3.2. SetExpressCheckout

Sebelum memulai sebuah transaksi dengan menggunakan PayPal, sistem harus terlebih dahulu mempersiapkan beberapa informasi yang dibutuhkan. Informasi tersebut berkaitan dengan proses transaksi yang akan berlangsung dan informasi tersebut akan diisikan ke dalam *field-field* yang ada di dalam operasi API *SetExpressCheckout*. Cara kerja operasi API tersebut adalah “request and response” maksudnya sistem harus mengirimkan pesan tertentu kepada PayPal dan PayPal akan membalas dengan pesan tertentu pula. Pesan yang dikirim bukanlah sembarang pesan, melainkan kumpulan *field-field* yang harus benar format penulisannya. Bila terjadi salah penulisan maka sistem tidak akan berjalan dengan baik.

Ada beberapa field yang penting yang harus ada di dalam sistem dan bersifat penting, misalnya,

- *PAYMENTREQUEST\_0\_AMT* adalah informasi mengenai jumlah uang yang akan dibayarkan dengan menggunakan PayPal.
- *PAYMENTREQUEST\_0\_CURRENCYCODE* berisi mengenai informasi mata uang yang digunakan di dalam transaksi.
- *PAYMENTREQUEST\_0\_TAXAMT* berisi tentang jumlah pajak yang harus dibayarkan.
- *RETURNURL* berisi URL yang akan digunakan untuk mengarahkan pembeli ketika pembeli sudah selesai melakukan transaksi pembayaran.
- *CANCELURL* berisi URL yang akan digunakan untuk mengarahkan pembeli ketika pembeli membatalkan proses transaksi.
- *PAYMENTREQUEST\_0\_PAYMENTACTION* bernilai “Sale”.
- *PAYMENTREQUEST\_0\_ITEMAMT* untuk memberi informasi mengenai item yang dijual. Didalamnya terdapat *field-field* seperti,

*L\_PAYMENTREQUEST\_0\_NAME*<sub>n</sub> berisi nama dari item,  
*L\_PAYMENTREQUEST\_0\_AMT*<sub>n</sub> berisi harga dari item,  
*L\_PAYMENTREQUEST\_0\_QTY*<sub>n</sub> berisi banyaknya item,  
*L\_PAYMENTREQUEST\_0\_ITEMCATEGORY*<sub>n</sub> bernilai *Digital* karena barang yang dijual adalah barang yang bersifat *digital*.

- *REQCONFIRMSHIPPING* bernilai “0” karena tidak memerlukan proses shipping.
- *NOSHIPPING* bernilai “1” karena tidak menggunakan shipping

Semua *field-field* yang telah diisi dengan nilai tertentu tadi kemudian disambung menjadi satu dengan menggunakan karakter &. Setelah itu, baru kemudian pesan tersebut disambung ke PayPal. Pesan yang berhasil akan mendapat balasan dari PayPal berupa *token*. Contoh pesan dapat dilihat pada gambar berikut,

#### Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=http://...
&CANCELURL=http://...
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_AMT=300
&PAYMENTREQUEST_0_ITEMAMT=200
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&L_PAYMENTREQUEST_0_ITEMCATEGORY0=Digital
&L_PAYMENTREQUEST_0_ITEMCATEGORY1=Digital
&L_PAYMENTREQUEST_0_NAME0=Kitty Antics
&L_PAYMENTREQUEST_0_NAME1=All About Cats
&L_PAYMENTREQUEST_0_QTY0=1
&L_PAYMENTREQUEST_0_QTY1=1
&L_PAYMENTREQUEST_0_AMT0=100
&L_PAYMENTREQUEST_0_AMT1=100
```

#### Response Parameters

```
[successResponseFields]
&TOKEN=EC-17C76533PL706494P
```

Gambar 2.7. Contoh pesan yang dikirim dan diterima oleh sistem ketika melakukan operasi *SetExpressCheckout*.

Sumber: PayPal (2012, p. 24)

Selain *field-field* yang dibahas diatas ada banyak field lain yang dapat digunakan ketika menjalankan operasi API SetExpressCheckout. *Field-field* yang tercantum diatas adalah *field-field* yang umum digunakan dan yang bersifat penting saja. Untuk melihat *field-field* yang lain dapat dilihat pada panduan PayPal *Express Checkout API for Digital Goods* (PayPal 53).

### **2.3.3. Instant Payment Notification**

*Instant Payment Notification* (IPN) adalah sebuah fitur yang dimiliki oleh PayPal yang memberikan sebuah fasilitas notifikasi kepada penjual bila terjadi transaksi pembayaran yang melibatkan penjual tersebut. Layanan ini dapat digunakan oleh sistem otomatis *back-office* penjual untuk melayani pembeli yang telah melakukan transaksi pembayaran dengan penjual, misalnya untuk memenuhi order yang diinginkan pembeli, melakukan pelacakan terhadap pembeli, menampilkan data-data yang berkaitan dengan proses transaksi pembayaran yang dilakukan oleh pembeli.

IPN memberikan notifikasi kepada penjual setiap ada transaksi yang berkaitan dengan penjual. Transaksi tersebut bisa bermacam-macam misalnya, transaksi pembayaran, pembayaran yang berulang, transaksi pengembalian uang, dan sebagainya. IPN juga akan memberikan notifikasi dari sebuah event yang berasal dari *Instant Payment*, *Instant Payment* ini menyangkut Express checkout API, Adaptive Payment API, dan transaksi pembayaran langsung dari kartu kredit. Selain itu IPN juga akan memberikan notifikasi bila terjadi pembayaran dari *e-check*, *e-check* adalah sebuah pembayaran yang dilakukan langsung dari *account* bank pembeli, sama seperti check biasanya, *e-check* membutuhkan 3 sampai 5 hari kerja untuk memproses *e-check* tersebut dan mencairkannya, setelah cair maka pembayaran tersebut akan muncul di *account* PayPal penjual.

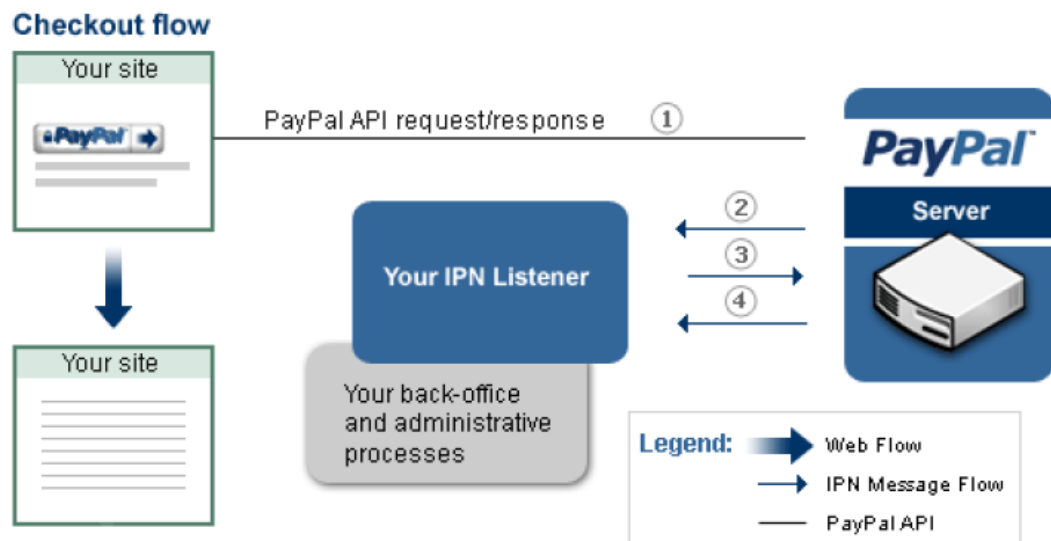
Mengapa IPN sangat penting? Sebenarnya hanya dengan menggunakan Express Checkout API, sudah bisa dibuat sebuah sistem pembayaran melalui PayPal. Bila pembeli sudah menyelesaikan langkah-langkah pembayaran yang ada di dalam Express Checkout maka penjual memberikan order yang diminta oleh pembeli tersebut. Namun masalahnya, bagaimanakah mengetahui apakah pembayaran tersebut sudah berhasil dilakukan. Apakah benar langkah-langkah

tersebut sudah diikuti dengan baik dan pembeli sudah membayar, karena bisa saja pembeli melakukan hack terhadap *website* tersebut sehingga langkah-langkah tersebut dilewati tanpa melalui pembayaran. Disinilah IPN sangat berperan, dengan menggunakan IPN sistem menjadi lebih aman dari ancaman-ancaman yang mencoba memanipulasi pembayaran. Karena cara kerjanya yang bersifat *asynchronous* maka penjual menjadi aman.

Selain karena faktor keamanan IPN juga sangat penting digunakan karena transaksi pembayaran bukan hanya berasal dari *website* penjual. Memang benar bahwa kebanyakan transaksi pembayaran yang akan muncul adalah berasal dari *website* penjual, namun ada juga transaksi yang diluar itu yaitu bila pembeli mengirimkan uang kepada penjual secara langsung melalui PayPal *account* miliknya, tidak melalui *website* penjual, atau adanya pembayaran yang dilakukan oleh PayPal sendiri. Semua transaksi tersebut dapat diketahui dengan menggunakan IPN.

Menurut (Paypal 8) cara kerja dari IPN secara umum adalah melalui tahap-tahap berikut ini, yaitu:

- Adanya transaksi pembayaran yang dilakukan oleh pembeli terhadap penjual.
- Di dalam sistem yang dimiliki oleh penjual, dibuat sebuah *listener* atau ada yang menyebutnya dengan handler, yang bertugas untuk menangani pesan notifikasi yang dikirim oleh PayPal bila terjadi pembayaran.
- Setelah pesan notifikasi berhasil diverifikasi, sistem melanjutkan dengan tahap selanjutnya sesuai dengan kebutuhan misalnya, memenuhi kebutuhan pembeli, memberi layanan download terhadap barang yang dibeli oleh pembeli, memasukkan data-data transaksi ke dalam database.



Gambar 2.8. Bagan alur kerja dari PayPal IPN

Sumber: PayPal (2012, p. 12)

Gambar di atas merupakan alur kerja dari proses IPN secara keseluruhan. Bila melihat gambar tersebut maka akan terlihat bahwa PayPal IPN bekerja secara asynchronous artinya bekerja secara terpisah dengan *website* yang dimiliki oleh penjual. Di dalam *website* penjual, pembeli akan melakukan pembelian secara normal dengan menggunakan Express Checkout API. Pembeli tidak mengetahui tentang IPN karena IPN bekerja pada bagian yang berbeda dari sistem pembayaran yang dilakukan oleh pembeli. Ketika transaksi sudah melalui tahap *DoExpressCheckout* maka PayPal API akan meminta pesan notifikasi untuk dikirimkan ke sistem *listener* yang dimiliki penjual (langkah nomor 1). Setelah menerima pesan dari PayPal API, PayPal akan mengirimkan pesan notifikasi kepada sistem *listener* penjual yang memberitahu bahwa telah terjadi transaksi pada *account* PayPal penjual (langkah nomor 2). *Listener* kemudian mengirimkan ulang pesan tersebut ke PayPal, pesan yang dikirim ulang tersebut harus berisi pesan yang sama seperti yang dikirim pada langkah 2, kemudian dilakukan proses encoding dengan cara yang sama pula (langkah nomor 3). PayPal akan menerima pesan tersebut dan akan membalas dengan satu kata yaitu *VERIFIED* bila pesan yang dikirimkan sama, dan *INVALID* bila pesan yang dikirimkan tidak sama (langkah nomor 4). Hal ini dilakukan untuk memastikan bahwa pesan IPN sudah dikirim ke alamat yang tepat, dalam hal ini yaitu *listener* dari penjual.

Langkah nomor 2, 3, dan 4 sangat penting dilakukan. Bila di dalam langkah nomor 3 PayPal tidak menerima balasan pesan yang dikirim oleh PayPal pada langkah nomor 2, maka PayPal akan terus mengirim pesan IPN ke *listener* sampai PayPal menerima balasannya. Jika dalam waktu 4 hari PayPal tetap tidak menerima balasan maka PayPal akan berhenti mengirim pesan IPN tersebut. Pesan IPN yang dikirim oleh PayPal bisa terjadi sewaktu-waktu jadi harus dipastikan bahwa *listener* dapat bekerja selama 24 jam. Hal selanjutnya yang perlu diperhatikan adalah Internet tidak selalu bisa diandalkan 100%, bisa saja ketika pembeli selesai melakukan transaksi pembayaran, pesan IPN yang seharusnya dikirim tidak terkirim ke *listener*. Jika hal ini terjadi maka penjual harus mengirimkan ulang pesan IPN tersebut secara manual, penjual masuk ke dalam *account* PayPal miliknya dan mengirimkan ulang pesan IPN di dalam menu IPN History di dalam *account* miliknya.

Di dalam pesan IPN yang dikirim oleh PayPal, banyak berisi informasi-informasi mengenai pembayaran yang telah dilakukan oleh pembeli. Berikut ini adalah contoh dari pesan respon IPN yang dikirimkan kepada sistem *back-office* dari penjual terhadap pembayaran \$19,95 melalui Express Checkout.

```
mc_gross=19.95&protection_eligibility=Eligible&address_status=confirmed&pay
er_id=LPLWNMTBWMFAY&tax=0.00&address_street=1+Main+St&payment_date=20%3A12%
3A59+Jan+13%2C+2009+PST&payment_status=Completed&charset=windows-
1252&address_zip=95131&first_name=Test&mc_fee=0.88&address_country_code=US&
address_name=Test+User&notify_version=2.6&custom=&payer_status=verified&add
ress_country=United+States&address_city=San+Jose&quantity=1&verify_sign=Atk
OfCXbDm2hu0ZELryHFjY-Vb7PAUvS6nMXgysbE1En9v-
1XcmSoGtf&payer_email=gpmac_1231902590_per%40paypal.com&txn_id=61E67681CH32
38416&payment_type=instant&last_name=User&address_state=CA&receiver_email=g
pmac_1231902686_biz%40paypal.com&payment_fee=0.88&receiver_id=S8XGHL9T3S
&txn_type=express_checkout&item_name=&mc_currency=USD&item_number=&residenc
e_country=US&test_ipn=1&handling_amount=0.00&transaction_subject=&payment_g
ross=19.95&shipping=0.00
```

Gambar 2.9. Contoh pesan IPN yang dikirim oleh PayPal

Sumber : PayPal (2012, p. 15)

Pada gambar tersebut terlihat bahwa di dalam pesan IPN ada berbagai macam informasi mengenai transaksi pembayaran, misalnya informasi *mc\_gross* merupakan informasi besar jumlah transaksi pembayaran yaitu 19,95. Selain itu

ada juga informasi tentang tanggal transaksi (*payment\_date*), *id* dari transaksi (*txn\_id*), status dari transaksi (*payment\_status*), mata uang yang digunakan (*mc\_currency*), dan juga email milik pembeli (*payer\_email*). Informasi-infromasi ini sangat penting dan dapat digunakan untuk keperluan melacak setiap transaksi yang ada, dan memenuhi kebutuhan pembeli bila transaksi yang dilakukan oleh pembeli sudah berhasil atau *payment\_status=COMPLETED*.

## 2.4. Highstock

Highstock adalah sebuah modul *javascript* yang digunakan untuk membuat grafik berdasarkan data yang disediakan. Highstock menggunakan kode program *javascript* murni, dan tidak memerlukan tambahan plug-in atau instalasi apapun sebelumnya. Highstock ini bersifat gratis untuk *website-website* yang bersifat personal dan non-profit. Untuk menggambar grafik di dalam *website* dengan menggunakan Highstock, perlu dilakukan konfigurasi Highstock dan menulis kode program terlebih dahulu. Kode program yang digunakan tidaklah sulit, karena tinggal mengatur isi dari parameter-parameter yang telah disediakan oleh Highstock.

Menurut dokumentasi dari situs Highstock (“Highstock Documentation”, par. 1) Untuk menggunakan Highstock, hal pertama yang harus dilakukan adalah menyertakan *javascript* dari Highstock dan jQuery. Kedua *javascript* tersebut didapatkan dari URL yang tertera pada Gambar 2.9.

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></script>
<script src="http://code.highcharts.com/highcharts.js"></script>
```

Gambar 2.10. URL dari *javascript* yang digunakan untuk Highstock.

Sumber: Highstock (2013, p. 1)

```
<div id="container" style="width:100%; height:400px;"></div>
```

Gambar 2.11. Tempat untuk menampilkan grafik.

Sumber: Highstock (2013, p. 1)

Hal berikutnya yang dilakukan adalah menyiapkan sebuah tempat di dalam HTML untuk menampilkan grafik dari Highstock. Tempat tersebut dibuat dengan

menggunakan *id* tertentu, contoh pada Gambar 2.10. Tempat untuk menampilkan gambar diberi nama container dengan ukuran lebar 100% atau maksimal dan tinggi sebesar 400 *pixel*.

```
$(document).ready(function() {
  var chart1 = new Highcharts.Chart({
    chart: {
      renderTo: 'container',
      type: 'bar'
    },
    title: {
      text: 'Fruit Consumption'
    },
    xAxis: {
      categories: ['Apples', 'Bananas', 'Oranges']
    },
    yAxis: {
      title: {
        text: 'Fruit eaten'
      }
    },
    series: [{
      name: 'Jane',
      data: [1, 0, 4]
    }, {
      name: 'John',
      data: [5, 7, 3]
    }
  ]
});
```

Gambar 2.12. Kode program dasar untuk menampilkan Highstock.

Sumber: Highstock (2013, p. 1)

Setelah tempat siap digunakan. Langkah berikutnya adalah membuat kode program untuk menampilkan grafik. Untuk membuat grafik, dibuatlah kode program yang memanggil kelas chart dari Highstock seperti pada gambar di atas. Kemudian setelah kelas dipanggil, dilakukan proses konfigurasi grafik sehingga grafik bisa menampilkan gambar sesuai dengan yang diinginkan oleh sistem. Kode program yang berwarna merah adalah kode program untuk melakukan konfigurasi. Parameter-parameter yang digunakan dalam gambar adalah,

- *renderTo* digunakan untuk memberikan informasi lokasi dimana grafik akan digambar, dalam hal ini grafik akan digambar pada container pada Gambar 2.10.

- *type* memberikan informasi jenis dari grafik. *Bar* adalah grafik batang. Selain *bar*, terdapat berbagai macam jenis grafik seperti *line*, *area*, *ohlc*, *candlestick*, dsb.
- *text* pada title digunakan untuk memberikan judul dari grafik.
- *categories* pada *xAxis* memberikan label pada sumbu x dari grafik.
- *Text* pada *yAxis* untuk memberikan judul dari sumbu y.
- *Series* berisi informasi data yang akan digambar di dalam grafik. Highstock bisa menggambar lebih dari dua series secara sekaligus. *name* merupakan informasi nama dari series, dan *data* berisi data dari series tersebut yang akan digambar dalam grafik.



Gambar 2.13. Struktur dasar bagian-bagian dari grafik Highstock.

Sumber: Highstock (2013, p. 1)

Parameter-parameter lain, yang merupakan bagian dasar grafik, yang sering digunakan oleh *user* untuk mengoperasikan grafik Highstock ditunjukkan pada Gambar 2.12.

- *rangeSelector* untuk mengatur rentang waktu yang akan ditampilkan di dalam *website*, *rangeSelector* menentukan besar kecilnya range dari sumbu x pada grafik. *Zoom* digunakan untuk memilih rentang waktu, misalnya 1m (satu bulan), 1y (satu tahun), dsb.
- *Navigator* untuk menentukan rentang waktu, hanya saja beda cara pengoperasiannya. Untuk mengubah rentang waktu maka *user* harus melakukan klik pada navigator untuk memperlebar atau mempersempit rentang waktu.
- *Scrollbar* digunakan untuk menggeser data yang ditampilkan pada grafik.
- *Crosshair* merupakan garis yang menunjukkan posisi *mouse* saat ini di dalam grafik. *Crosshair* digambar tegak lurus dengan sumbu x untuk menunjukkan data yang ditunjuk dengan *mouse* berada pada posisi mana di dalam sumbu x.
- *Tooltip* merupakan tampilan kotak yang menunjukkan informasi data yang ditunjuk dengan *mouse*. Informasi tersebut berupa nilai dari sumbu x dan sumbu y.

## 2.5. Bank Central Asia

BCA merupakan salah satu bank besar yang ada di Indonesia, BCA merupakan kepanjangan dari Bank Central Asia. Karena bank BCA merupakan bank yang dikenal oleh banyak masyarakat Indonesia, dan memiliki nasabah yang banyak. Pada akhirnya dipilihlah bank BCA sebagai bank yang digunakan oleh sistem.

Seiring perkembangan teknologi yang ada, hal tersebut memungkinkan transaksi pembayaran dilakukan dengan mudah. BCA menyediakan layanan yang disebut dengan BCA Klik Pay. Dengan layanan tersebut nasabah dapat membeli barang dengan mudah melalui internet. Namun, untuk menggunakan layanan tersebut penjual harus menjadi *merchant* dari BCA. Salah satu syarat untuk bisa menjadi *merchant* adalah perusahaan atau usaha bisnis sudah berbadan hukum, dengan mengajukan proposal ke BCA. (“Cara Menjadi Merchant”, para. 1) Oleh karena itu, sistem tidak bisa menggunakan layanan ini. Sehingga sistem membuat skenario pembayaran BCA tersendiri.

## 2.6. MySQL

MySQL adalah sebuah *database* yang bersifat *open source* yang dilindungi GNU General Public License. MySQL merupakan *database* yang paling sering digunakan di dunia, Organisasi-organisasi besar seperti Facebook, Google, Adobe, Alcatel Lucent and Zappos juga menggunakan *database* MySQL. MySQL ini dapat digunakan di 20 platform OS yang berbeda seperti, Linux, Windows, Mac OS, Solaris, IBM AIX, dsb. (MySQL, par. 1)

Di dalam Tugas Akhir ini, sistem juga menggunakan MySQL sebagai *database*. Untuk mengoperasikan data di dalam *database* MySQL, diperlukan sebuah perintah-perintah khusus. Beberapa perintah umum yang digunakan di dalam Tugas Akhir ini adalah,

- *INSERT* merupakan perintah MySQL yang digunakan untuk memasukkan data-data ke dalam *database*.
- *SELECT* digunakan untuk mengambil data di dalam *database*.
- *ORDER BY* merupakan perintah yang digunakan untuk mengurutkan data di dalam *database*. Secara standard, *ORDER BY* akan mengurutkan data secara *ascending* (dari kecil ke besar). Bila sistem ingin mengurutkan secara *descending* (dari besar ke kecil), maka ditambahkan perintah *DESC*.
- *LIMIT* merupakan perintah yang digunakan untuk membatasi jumlah data yang akan diambil.
- *UPDATE* digunakan untuk merubah nilai dari data yang ada di dalam *database*.

## 2.7. CRON

CRON adalah sebuah sistem *daemon* pada Ubuntu Linux yang digunakan untuk menjalankan sebuah tugas khusus pada waktu sudah yang ditentukan. CRON menggunakan *file* crontab untuk menjalankan tugasnya. Crontab merupakan sebuah *file* text yang sangat sederhana dimana di dalamnya terdapat daftar dari perintah-perintah yang ingin dijalankan oleh server dengan menggunakan CRON. Perintah-perintah tersebut akan dijalankan di belakang sistem, artinya perintah tersebut akan berjalan dengan sendirinya tanpa

mempengaruhi atau dipengaruhi oleh *website*. Setiap *user* di dalam server memiliki crontab-nya masing-masing sehingga crontab hanya berjalan pada bagian-bagian yang bisa diakses oleh *user* tersebut. Untuk menjalankan crontab *user* memberikan perintah *crontab -e*. Ada juga *root* crontab yaitu crontab yang digunakan untuk menjalankan perintah-perintah yang memerlukan akses administratif. Crontab ini hanya bisa dilakukan oleh *root* atau administrator dengan memberikan perintah *sudo crontab*. (“Ubuntu”, para. 1)

Secara umum, format penulisan perintah di dalam crontab terdiri dari dua bagian yaitu bagian yang menunjukkan waktu perintah akan dijalankan dan bagian yang menunjukkan perintah apa yang dijalankan. Masing-masing bagian dipisahkan dengan spasi. Pada Gambar 2.13 ditunjukkan contoh penulisan perintah pada crontab.

```
01 04 1 1 1 /usr/bin/somedirectory/somecommand
01 04 * * * /usr/bin/somedirectory/somecommand
```

Gambar 2.14. Contoh perintah crontab.

Sumber: Ubuntu (2013. para. 3)

01 merupakan informasi menit, 04 merupakan informasi jam, 1 merupakan informasi mengenai tanggal (1-31), 1 berikutnya merupakan informasi mengenai bulan (1-12), dan 1 berikutnya merupakan informasi mengenai hari (0-6) dimana 0 adalah Minggu. Sehingga perintah pada gambar bagian atas artinya adalah sebagai berikut, sistem akan menjalankan *somecommand* pada jam 04:01 pagi pada tanggal 1 Januari dan setiap hari Senin pada bulan Januari.

Selain dengan menggunakan angka, crontab juga bisa ditulis dengan menggunakan tanda bintang (\*). Tanda bintang memiliki arti setiap, jadi jika ingin melakukan perintah crontab setiap satu jam maka angka 04 diganti dengan tanda bintang tersebut. Pada gambar bagian bawah, perintah crontab memiliki arti sebagai berikut, sistem akan menjalankan *somecommand* pada jam 04:01 pagi setiap harinya. Hal ini terjadi karena informasi tanggal, bulan, dan hari diganti dengan tanda bintang.

## 2.8. RSS

RSS merupakan singkatan dari *Really Simple Syndication*, yaitu sebuah format XML sederhana yang berisi informasi terkini dari sebuah *website* yang dibagikan kepada publik. RSS sudah berkembang dan saat ini banyak digunakan oleh *website-website* untuk saling berbagi informasi satu dengan yang lain. RSS juga berguna bagi *user* yang selalu ingin mengetahui informasi terkini dari *website* yang diikutinya. Dengan menggunakan RSS, *user* tersebut tidak perlu berkunjung ke *website* secara langsung, namun cukup mengikuti RSS *Feed* yang disediakan oleh *website* tersebut.

Setiap *file* RSS bisa mempunyai banyak berita di dalamnya, dalam satu *file* bisa mencakup lebih dari 15 berita. Masing-masing berita di dalam *file* tersebut dipisahkan oleh `<item>` tag. Di dalam tag item tersebut ada juga tag lain yaitu *title*, *link*, *description*, dan *image*. Contoh format RSS dapat dilihat pada Gambar 2.14.

```
...
<item>
  <title>RSS Resources</title>
  <link>http://www.webreference.com/authoring/languages/xml/rss/</link>
  <description>Defined in XML, the Rich Site Summary (RSS) format has
  quietly become a dominant format for distributing headlines on the Web.
  Our list of links gives you the tools, tips and tutorials you need to get
  started using RSS. 0323</description>
</item>
...
```

Gambar 2.15. Contoh format penulisan RSS.

Sumber: Webreference (2013, para. 3)

Setiap berita ditandai dengan `<item>` tag dan ditutup dengan `</item>` tag. Di dalam berita tersebut ada tag lain seperti `<title>` yang merupakan judul dari berita, `<link>` merupakan URL dari mana berita tersebut diambil, `<description>` merupakan penjelasan singkat mengenai berita yang disajikan. Jika *user* ingin membaca lebih detail tentang berita tersebut maka *user* harus masuk ke dalam *link* yang tersedia.