

Pengembangan Aplikasi Pencari Harga Terbaik Berbasis Selenium Web Crawler

The Best Price Finder Application Development Based on Selenium Web Crawler

Lina Andriyani¹, Sepyan Purnama Kristanto^{2*}, Lutfi Hakim³, I Wayan Suardinata⁴

Teknik Informatika, Politeknik Negeri Banyuwangi, Banyuwangi, Indonesia¹

Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Banyuwangi, Banyuwangi, Indonesia^{2,3}

Bisnis Digital, Politeknik Negeri Banyuwangi, Banyuwangi, Indonesia⁴

Email: lina.andriyani@gmail.com¹, *sepyan@poliwangi.ac.id², lutfi@poliwangi.ac.id³

*Corresponding Author

Abstract

Online shopping or commonly known as an online store or online shop is a place to purchase an item through the Internet media. With so many e-commerce websites now, we can easily access information on goods both from prices, specifications, and purchase transactions. However, e-commerce websites have now been built and used by many sellers, this sometimes makes it difficult for buyers to choose the items they are looking for. That way we definitely do the manual method, namely visiting one by one e-commerce website to find out price comparison information about the item being sought. Therefore, an application is made that aims to facilitate users to compare fashion prices from three online stores namely Tokopedia, Bukalapak and Blanja.com. The method used is Extreme Programming (XP) and this application was built using the PHP framework laravel and using web crawling techniques. Shows that the application developed is able to provide significant benefits to users in facilitating the process of comparing prices of fashion products from various online stores. Users can access price information, specifications, and transaction options from the three online stores quickly and efficiently through this application. This saves time and effort for buyers in searching and selecting the desired products. In addition, users can also utilize additional features in the app to get product recommendations that suit their preferences and needs.

Keywords: Online shop, Web crawling, Laravel, Price appeal, Fashion.

Abstrak

Belanja Online atau biasa dikenal dengan Toko online atau Online Shop merupakan tempat pembelian suatu barang melalui media Internet. Dengan banyaknya website e-commerce sekarang, kita dapat dengan mudah mengakses informasi barang baik dari harga, spesifikasi, serta transaksi pembelian. Akan tetapi website e-commerce sekarang telah banyak dibangun dan digunakan oleh para penjual, hal tersebut terkadang membuat para pembeli sulit untuk memilih barang yang dicari. Dengan begitu sudah pasti kita melakukan cara manual yaitu mengunjungi satu per-satu website e-commerce untuk mengetahui informasi perbandingan harga mengenai barang yang dicari. Oleh karena itu dibuat sebuah aplikasi yang bertujuan untuk memudahkan pengguna membandingkan harga fashion dari tiga toko online yaitu tokopedia, Bukalapak dan Blanja.com. Metode yang digunakan yaitu Extreme Programming (XP) serta aplikasi ini dibangun menggunakan PHP framework laravel dan menggunakan teknik web crawling. Menunjukkan bahwa aplikasi yang dikembangkan mampu memberikan manfaat signifikan bagi pengguna dalam mempermudah proses perbandingan harga produk fashion dari berbagai toko online. Pengguna dapat mengakses informasi harga, spesifikasi, dan opsi transaksi dari ketiga toko online dengan cepat dan efisien melalui aplikasi ini. Hal ini menghemat waktu dan usaha bagi pembeli dalam mencari dan memilih produk yang diinginkan. Selain itu, pengguna juga dapat memanfaatkan fitur-fitur tambahan dalam aplikasi untuk mendapatkan rekomendasi produk yang sesuai dengan preferensi dan kebutuhan mereka.

Kata Kunci: Toko online, Web crawling, Laravel, Banding harga, Fashion.

I. PENDAHULUAN

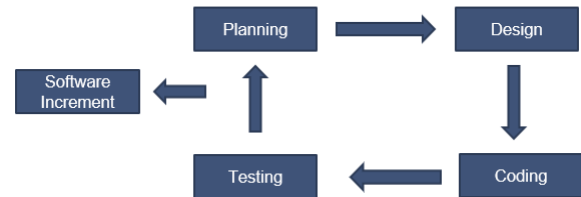
Belanja Online atau biasa dikenal dengan Toko online atau Online Shop merupakan tempat pembelian suatu barang melalui media Internet. Melalui belanja lewat Internet seorang pembeli bisa melihat dan memilih terlebih dahulu barang yang hendak dibeli melalui *website* yang dipromosikan oleh penjual (Nurhayati, 2017). Setelah pembeli memilih barang yang diinginkan kemudian pembeli melakukan pembayaran kepada penjual melalui rekening bank yang bersangkutan. Setelah proses pembayaran di terima, kemudian penjual mengirim barang pesanan pembeli ke alamat tujuan. Toko *Online* di Indonesia mengalami perkembangan yang cukup pesat. Mulai dari *website* penjualan *handphone*, buku, makanan, fashion bahkan hingga ke alat elektronik pun mulai dirambah oleh layanan belanja *online*.

Dengan banyaknya *website e-commerce* sekarang, kita dapat dengan mudah mengakses informasi barang baik dari harga, spesifikasi, serta transaksi pembelian. Hal ini dapat menghemat waktu dan biaya bagi calon pembeli, karena tidak perlu datang ke toko langsung. Akan tetapi *website e-commerce* sekarang telah banyak dibangun dan digunakan oleh para penjual, hal tersebut terkadang membuat para pembeli sulit untuk memilih barang yang dicari. Dengan begitu sudah pasti kita melakukan cara manual yaitu mengunjungi satu per-satu *website e-commerce* untuk mengetahui informasi perbandingan harga mengenai barang yang dicari, serta terkait masalah kebijakan dan aturan tersendiri pada masing-masing *website e-commerce* dalam lingkup penyajian informasi (Yan Watequlis, 2018).

Dengan adanya permasalahan tersebut maka perlu dibangun sebuah aplikasi yang dapat memudahkan pembeli dalam membandingkan harga barang. Maka pada proyek akhir ini diangkat sebuah judul “PENERAPAN METODE WEB CRAWLING PADA APLIKASI PEMBANDING HARGA FASHION DI TOKO ONLINE” yang berasal dari tiga toko online *Tokopedia*, *Bukalapak*, dan *Blanja.com*.

II. METODE PENELITIAN

Penelitian ini dalam pelaksanaannya menggunakan metode *Extreme Programmings* (XP). Metode ini menjadi sebuah pendekatan yang cocok untuk pengembangan awal sebuah aplikasi, dengan melihat batas waktu pengerjaannya. Metode XP dikembangkan dengan tujuan untuk menghasilkan perangkat lunak yang berkualitas tinggi dan lebih produktif. XP juga bertujuan untuk mengurangi biaya selama ada perubahan dalam pengembangan perangkat lunak menggunakan siklus (tahapan) pengembangan perangkat lunak yang singkat. Metode ini dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan-perubahan *requirement* yang sangat cepat (Irmawati, 2019).



Gambar 2. Kerangka Kerja Extreme Programming

Pada Gambar 2 diatas menunjukkan kerangka kerja dari metode XP. Berikut penjelasannya:

A. Planning (Perencanaan)

Tahapan ini dimulai dengan pemahaman konteks dari aplikasi, mendefinisikan keluaran (output), fitur yang ada pada aplikasi, fungsi dari aplikasi yang dibuat, penentuan waktu, serta alur proses pengembangan aplikasi.

B. Design (Perancangan)

Tahap ini melakukan pembuatan desain aplikasi, meliputi pembuatan basis data dan *layout* aplikasi.

C. Coding (Pengkodean)

Pada tahapan ini kegiatan penerapan pemodelan yang sudah dibuat kedalam bentuk *user interface* dengan menggunakan bahasa pemrograman *Python* untuk teknik *web crawling* dengan *library selenium* dan *PHP* untuk *website* dengan *framework Laravel*. Dalam kegiatan ini nantinya menghasilkan sebuah aplikasi yang siap untuk dilakukan pengujian.

D. Testing (Pengujian)

Tahap ini memfokuskan pada pengujian fitur – fitur yang ada pada aplikasi dengan melibatkan calon pengguna sehingga tidak ada kesalahan (*error*), serta aplikasi yang dibuat akan dapat digunakan.

E. Software Increment

Melakukan proses produksi aplikasi atau disebut juga dengan rilis aplikasi. Dimana aplikasi ini dipublikasikan ke khalayak umum dengan cara pengaksesan aplikasi melalui jaringan internet.

III. HASIL DAN PEMBAHASAN

Pengembangan aplikasi pencari harga terbaik berbasis *selenium web crawler* merupakan aplikasi yang dapat digunakan oleh pengguna untuk mencari harga terbaik untuk fashion wanita dan pria. Aplikasi ini memiliki 2 level pengguna yakni pelanggan dan admin. Dalam aplikasi ini terdapat fitur login, fitur filter toko online, fitur kategori, fitur cari dan fitur rentang harga. Aplikasi ini sangat mudah digunakan, karena tanpa login pengguna dapat melakukan pencarian harga tetapi hasil yang di tampilkan terbatas hanya

delapan item yang tampil dan dapat melakukan filter toko online saja. Untuk mengetahui lebih banyak hasil pencarian harga terbaik dan dapat melihat semua fitur yang ada maka pengguna di sarankan untuk login terlebih dahulu. Dalam Pengembangan Aplikasi Pencari Harga Terbaik Berbasis Selenium Web Crawler ini data yang diambil adalah data realtime.

Pengembangan Aplikasi Pencari Harga Terbaik Berbasis Selenium Web Crawler adalah sebuah aplikasi yang mengambil data berdasarkan tiga toko online yakni Tokopedia, Bukalapak dan Blanja.com. Aplikasi ini dapat memudahkan pengguna yang ingin mencari harga pakaian termurah dengan cara mencari harga termurah dari ketiga toko online. Sistem ini dibangun menggunakan dua bahasa pemrograman yakni *python* dan *PHP*, memakai *Framework Laravel* dengan basisdata *MySQL* dan menggunakan *library selenium* untuk pengambilan data di toko online. Selenium adalah sebuah *library* yang menjalankannya menggunakan web driver.

A. Penggunaan Laravel

Aplikasi pencarian harga termurah di toko online di bangun menggunakan *framework laravel*. Untuk membuat proyek baru di laravel ketikkan perintah di cmd. *Command* yang dijalankan sebagai berikut:

```
Composer create-project --prefer-dist
laravel/laravel nama_projek yang akan dibuat
```

Kemudian untuk memastikan instalasi sudah selesai dan siap untuk di gunakan maka ketikkan perintah pada cmd untuk masuk ke dalam file crawling. Perintah yang dijalankan sebagai berikut:

```
Cd C:\xampp\htdocs\crawling
```

Kemudian ketikkan perintah di bawah ini untuk menjalankan laravel

```
Php artisan serve
```

Setelah dijalankan maka akan muncul tampilan laravel yang berarti laravel sudah dapat digunakan. Kemudian dalam aplikasi pencarian harga termurah terdapat fitur filter toko online. Dibawah ini adalah source code untuk filter toko online.

```
if(isi == "tokopedia" || isi == "bukalapak" ||
isi == "belanjacom"){
window.location.href =
"http://localhost:8000/hasil? toko="+isi;
}
```

Pada source code diatas terdapat fungsi yang memanggil isi toko online. Ketika pengguna mengklik tombol tampilan maka akan diarahkan ke halaman home controller yang source codenya dibawah ini:

```
if(isset($_GET['toko'])) &&
!isset($_GET['hrg_awal']) &&
!isset($_GET['hrg_akhir'])){
    $kri = $_GET['toko'];
```

```
$data = DB::table('scrap')
->where('sumber', $kri)
->orderBy('hrg', 'asc')
->paginate(35);
```

Pada source code diatas merupakan fungsi jika toko online yang di cari dapat ditemukan atau ada pada database. Sehingga dengan kode di atas, maka akan menampilkan harga awal dan harga akhir yang mengambil data dari database yang bernama scrap sesuai kriteria sumber dan berdasarkan harga termurah. Kemudian hasil scrap ditampilkan 35 item perhalaman. Kemudian terdapat fitur kategori atasan wanita dan atasan pria. Ketika pengguna mengklik tombol cari berdasarkan kategori maka akan memanggil fungsi di source code dibawah ini:

```
function kirim_kategori(){
var
    category
=
document.getElementsByName('kategori');
for(i = 0; i < category.length; i++) {
    if(category[i].checked)
    var isi = category[i].value
}
window.location.href=
"http://localhost:8000/jalankan?kategori="+isi;
}
```

Fungsi source code diatas untuk memastikan kategori ada kemudian data akan diarahkan ke home controller seperti source code dibawah ini:

```
function jalankan(){
    $this->middleware('auth');
    if(isset($_GET['kategori'])){
        $data = $_GET['kategori'];
        echo $data;
    } else {
        $data = $_POST['data'];
    }
}
```

Pada source code diatas jika kategori ada maka akan menampilkan data sesuai kategori yang di cari. kemudian mulai melakukan scrap data yang dicari.

Tabel 1. Judul tabel, gunakan sentence case (huruf awalnya besar)

No	Baris ini	Italic
1	Ini isi tabel, jika tidak mencukup, Anda bisa mengecilkan ukuran huruf sampai 8 points. Jangan lebih kecil dari ini, kecuali jika Anda menginginkan pembaca tulisan Anda sakit mata. :-)	Font isi tabel Regular

B. Scrap toko online

Dalam pengambilan data di toko online aplikasi ini menggunakan anaconda sebagai alternatif yang lebih baik dalam instalasi python beserta *library*nya. Dan di haruskan mendownload chrome driver untuk menjalankan *library* selenium saat melakukan proses scrap data. Jadi langkah awal yang dilakukan sebelum menscrap toko online adalah menginstal anaconda terlebih dahulu yang didalamnya akan otomatis terinstal *python*. Kemudian gunakan perintah dibawah ini untuk instal *library* selenium dengan cara membuka cmd dan mengetikkan:

```
Pip install selenium
```

Setelah sukses kemudian masukkan source code tools yang digunakan untuk pengambilan data dengan *library* selenium pada file yang digunakan untuk menyimpan source code yang bernama *scrap.py* seperti dibawah ini:

```
from selenium.webdriver.common.keys import Keys
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions as EC
import requests
from selenium.common.exceptions import TimeoutException
```

Pada source code diatas terdapat *library* selenium yang berfungsi untuk menjalankan web driver. Kemudian Dibawah ini terdapat *source code* daftar untuk menyimpan data sesuai data yang ingin diambil pada toko online yakni nama produk, harga produk, gambar produk, dan link produk.

```
nama_produk_tokopedia = []
hrg_produk_tokopedia = []
gbr_produk_tokopedia = []
link_produk_tokopedia = []

nama_produk_bukalapak = []
hrg_produk_bukalapak = []
gbr_produk_bukalapak = []
link_produk_bukalapak = []

nama_produk_belanjacom = []
hrg_produk_belanjacom = []
gbr_produk_belanjacom = []
link_produk_belanjacom = []
```

Setelah daftar selesai dibuat kemudian memulai scrap toko online seperti source code di bawah ini :

```

baca_inputan_user=open("C:/xampp/htdocs/crawling/public/nama_produk.txt", "r")
isi_inputan = baca_inputan_user.read()
chrome_path = r"C:\xampp\htdocs\chromedriver.exe"
driver = webdriver.Chrome(chrome_path)
driver.get("https://www.tokopedia.com/")
driver.maximize_window()
satu=driver.find_element_by_xpath('//*[@id="search-container"]/form/div/div/input')
satu.send_keys(isi_inputan)
satu.send_keys(Keys.ENTER)
driver.get(driver.current_url)
WebDriverWait(driver,25).until(EC.presence_of_element_located((By.CLASS_NAME, 'css-lehqh5q'))))
nama_produk_1 =
driver.find_elements_by_class_name("css-18c4yhp")
hrg_produk_1 =
driver.find_elements_by_class_name("css-rhd610")
gbr_produk_1 =
driver.find_elements_by_class_name("css-lehqh5q")

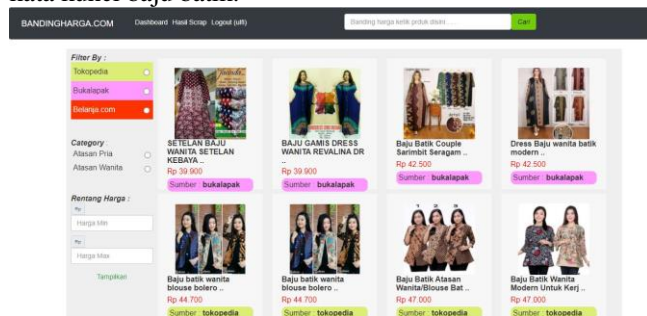
num = 0
for nm_1 in nama_produk_1:
    kon_money_1 =
    hrg_produk_1[num].text.replace("Rp","")
    kon_money_2 = kon_money_1.replace(".", "")
    nama_produk_tokopedia.append(nm_1.text)
    hrg_produk_tokopedia.append(kon_money_2)
    num = num + 1
for gb_1 in gbr_produk_1:
    image = gb_1.find_element_by_tag_name("img")
    linkx = gb_1.find_element_by_tag_name("a")
    img_src = image.get_attribute("src")
    linkx_href = linkx.get_attribute("href")
    link_produk_tokopedia.append(linkx_href)
    gbr_produk_tokopedia.append(img_src)
ulang = 0
for nm_prod_tokopedia in nama_produk_tokopedia:
    masuk={'nama':nm_prod_tokopedia,'hrg':hrg_produk_tokopedia[ulang], 'gbr':gbr_produk_tokopedia[ulang], 'sumber':"tokopedia", 'linkx':link_produk_tokopedia[ulang]}

    Resp=requests.post('http://localhost/webcrawling.php', params=masuk)
```

```
ulang = ulang + 1
driver.quit()
```

Pada source code diatas user menginputkan kata kunci pencarian nama produk, kemudian kata kunci tersebut di simpan sementara didalam file ("C:/xampp/htdocs/crawling/public/nama_produk.txt","r"). Kemudian sistem membaca isi file dari inputan setelah itu sistem memanggil chrome driver sesuai tempat file chromedriver. Kemudian sistem menjalankan chrome driver dengan alamat toko online. Setelah itu chrome driver memaximaize jendela tampilan chrome driver. Kemudian membuka link ke dua secara otomatis menuliskan inputan sesuai yang dituliskan user kemudian di enter. Setelah itu tampil hasil yang di cari sesuai inputan. Saat proses menampilkan hasil chromedriver diberikan waktu scrap 25 detik sampai gambar tertampilkan semua. Kemudian sistem mengambil data berdasarkan nama kelas yakni nama produk, harga produk, gambar produk.

Kemudian sistem memfilter hasil scrap dengan menghilangkan tanda "Rp" dan "." pada harga di karenakan tidak dapat masuk database apabila harga yang di ambil terdapat tanda tersebut. Kemudian memasukkan nama produk dan harga produk ke list yang telah di sediakan. Kemudian sistem mencari tag "img" untuk mengambil link gambar dan mencari tag "a" untuk mengambil link. Kemudian sistem mengambil isi dari "src" yang berisi link gambar dan mengambil isi dari "href" yang berisi link toko online. kemudian data yang di peroleh dimasukkan kedalam list. Setelah itu hasil data yang diperoleh berupa nama produk, harga produk, gambar produk dan sumber toko onlinenya di masukkan kedalam database. Kemudian hasil scrap di tampilkan pada hasil. Dibawah ini terdapat gambar hasil percobaan aplikasi pembanding harga fashion dengan kata kunci baju batik.



Gambar 3. Hasil percobaan kata kunci baju batik wanita

Pada hasil percobaan diatas pengguna memasukkan kata kunci pencarian pakaian yang ingin di cari. kemudian sistem melakukan crawling tiga toko online setelah proses pengambilan data di toko online telah selesai maka sistem akan menampilkan hasil dari harga termurah sesuai pencarian. Pengguna juga dapat memfilter toko online yang diinginkan, memilih category pakaian dan menentukan range

harga yang diinginkan.

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan di atas, dapat diambil kesimpulan bahwa 1) sistem ini dibangun menggunakan dua Bahasa pemrograman yaitu python dan PHP dengan menggunakan *framework Laravel* dan basis data menggunakan dan basis data menggunakan *MySQL*, 2) Pengambilan data dari toko online dilakukan dengan mengambil data dari tiga toko online yakni Tokopedia, bukalapak, dan blanja.com. Data yang diambil berupa nama, harga, gambar dan link produk, 3) Sistem melakukan pengambilan data menggunakan *library selenium* yang menjalankannya menggunakan *webdriver* yang mana didalam Bahasa pemrograman python mendukung dalam pemrosesan data dan banyak *library* yang tersedia, 4) aplikasi ini digunakan untuk pencarian harga termurah yang diambil berdasarkan tiga toko online. Hasil penelitian ini menunjukkan bahwa aplikasi yang dikembangkan mampu memberikan manfaat signifikan bagi pengguna dalam mempermudah proses perbandingan harga produk fashion dari berbagai toko online. Pengguna dapat mengakses informasi harga, spesifikasi, dan opsi transaksi dari ketiga toko online dengan cepat dan efisien melalui aplikasi ini. Hal ini menghemat waktu dan usaha bagi pembeli dalam mencari dan memilih produk yang diinginkan. Selain itu, pengguna juga dapat memanfaatkan fitur-fitur tambahan dalam aplikasi untuk mendapatkan rekomendasi produk yang sesuai dengan preferensi dan kebutuhan mereka. Dengan demikian, pengguna dapat memperoleh hasil yang maksimal dalam proses pembelian produk fashion secara online melalui aplikasi ini.

REFERENSI

- [1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
- [2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
- [3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
- [4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
- [5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.
- [6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.
- [7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
- [8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [Dig. 9th Annu. Conf. Magnetism Japan, 1982, p. 301].
- [9] M. Young, *The Technical Writers Handbook*. Mill Valley, CA: University Science, 1989.

-
- [10] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.
 - [11] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul. 1993.
 - [12] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.
 - [13] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.
 - [14] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. 1995 IEEE Int. Conf. Communications*, pp. 3–8.
 - [15] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.
 - [16] G. W. Juette and L. E. Zeffanella, "Radio noise currents in short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, Jun. 22–27, 1990, Paper 90 SM 690-0 PWRs.
 - [17] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.
 - [18] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
 - [19] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.
 - [20] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.
 - [21] *IEEE Criteria for Class IE Electric Systems* (Standards style), IEEE Standard 308, 1969.
 - [22] *Letter Symbols for Quantities*, ANSI Standard Y10.5-1968.
 - [23] R. E. Haskell and C. T. Case, "Transient signal propagation in lossless isotropic plasmas (Report style)," USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.
 - [24] E. E. Reber, R. L. Michell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (420-46)-3, Nov. 1988.
 - [25] (Handbook style) *Transmission Systems for Communications*, 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.
 - [26] *Motorola Semiconductor Data Manual*, Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.
 - [27] (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume (issue). Available: [http://www.\(URL\)](http://www.(URL))
 - [28] J. Jones. (1991, May 10). *Networks* (2nd ed.) [Online]. Available: <http://www.atm.com>
 - [29] (Journal Online Sources style) K. Author. (year, month). *Title. Journal* [Type of medium]. Volume(issue), paging if given. Available: [http://www.\(URL\)](http://www.(URL))
 - [30] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. 21(3). pp. 876–880. Available: <http://www.halcyon.com/pub/journals/21ps03-vidmar>