

---

# Rancang Bangun *Chrome Extension* dengan *Page Synthesis*

<sup>1)</sup>Roger Victor Liem, <sup>2)</sup>Hendry

Fakultas Teknologi Informasi  
Universitas Kristen Satya Wacana  
Jl. Diponegoro 52-60, Salatiga 50711, Indonesia  
Email: <sup>1)</sup> rogerliem9134@gmail.com, <sup>2)</sup> hendry@staff.uksw.edu

## Abstract

Web 2.0 as we know is currently experiencing an inability to meet the needs of many different users. This inability can be solved by using the Adaptive Web systems where the contents of a web page can be established as the user desires. This technology is then applied to the Extension contained in Google Chrome Extension built by applying the method of Page Synthesis, where pages that are being displayed to the user is first synthesized with Content Script to allow for changes in the orientation and filter of the image. This extension was built to manipulate the orientation of an image and used image processing methods without having to rely on the existing functionality within the web page.

**Keywords** : Adaptive Web System, Page Synthesis, Chrome Extension, Content Script

## 1. Pendahuluan

*Web* yang dikenal saat ini adalah *web* dinamis, *web* 2.0. *Web* yang dulunya hanya menampilkan informasi saja kini berubah menjadi *web* yang isinya dapat diubah dengan mudah sesuai dengan keinginan pengguna. *Web* dinamis ini lama kelamaan menimbulkan suatu masalah baru yaitu, konten *web* yang sama, untuk setiap pengguna yang berbeda. Misalnya, sebuah *website* penjualan buku akan memberikan penawaran buku *best seller* yang sama pada setiap pengguna *website* yang mempunyai kategori buku kesukaan yang berbeda. Sebagai contoh lain, sebuah *website* pembelajaran akan menampilkan penjelasan dan tautan pembelajaran yang sama kepada tiap penggunanya. Masalah seperti ini dapat ditanggulangi menggunakan *Adaptive Web System* [1].

*Adaptive Web System* (AWS) adalah suatu sistem *web* yang dapat menyesuaikan aspek-aspek utama dari *web* itu sendiri berdasarkan perilaku dan keinginan pengguna serta kemampuan perangkat. Aspek-aspek *web* tersebut antara lain, cara menampilkan konten, struktur *web* dan konten *web* tersebut. AWS bisa dibagi menjadi 3 bagian yakni, *domain model*, *user model* dan *adaptation model*. Bagian-bagian tersebut masih belum mencakup seluruh aspek dari AWS, ada aspek lain yang belum termasuk di dalamnya yaitu *presentation model* [2].

*Presentation Model* dalam AWS merupakan bagian yang menjelaskan tata letak elemen dari suatu *website* yang dibangkitkan oleh sistem [3]. Pada *presentation model* dapat diterapkan teknologi *Page Synthesis* untuk

membangkitkan halaman *web* secara dinamis. Ada beberapa jenis sintesa yang digunakan, salah satunya halaman *web* yang ingin dibangkitkan sudah dibuat terlebih dahulu, lalu diadaptasikan dengan keinginan pengguna [2].

*Web* yang semakin kompleks menuntut berkembangnya *web browser*. *Web browser* adalah *software* yang digunakan untuk menerima, menampilkan dan menjelajah informasi di *web*. *Web browser* yang dulu hanya *software* tambahan, kini menjadi suatu *software* yang perlu dimiliki setiap orang untuk menjelajah *internet*. *Browser* yang kini banyak digunakan adalah Google Chrome.

**Tabel 1** Perbandingan Pengguna Internet Explorer, Firefox, Chrome, Safari, Opera dan lainnya di seluruh dunia dari tahun 2008 hingga 2013 [4]

Year	IE	Firefox	Chrome	Safari	Opera	lainnya
2008	67.92	25.49	0.91	2.7	2.7	0.27
2009	59.71	30.48	3.27	3.1	2.62	0.83
2010	51.45	31.27	10.25	4.31	1.96	0.76
2011	42.93	28.2	21.08	5.33	1.84	0.62
2012	33.23	23.79	32.78	7.25	1.69	1.26
2013	28.98	20.58	38.99	8.27	1.11	2.08

Menurut Tabel 1, pada tahun 2013, Google Chrome mempunyai persentase pengguna paling tinggi jika dibandingkan dengan Internet Explorer, Mozilla Firefox, Safari, Opera dan *browser* lain. Kenaikan jumlah pengguna paling tinggi pada tahun 2012, dimana Google Chrome naik ke tingkat 32.78 % [4]. Jumlah pengguna yang tinggi ini disebabkan oleh para *developer* yang terus menerus memperbaharui Google Chrome. Google Chrome memiliki *extension*, yakni aplikasi kecil yang dapat ditambahkan untuk menambah fungsionalitasnya. *Extension* ini memiliki fitur Content Script yang dapat menginjeksikan elemen HTML baru ke dalam halaman *web* yang sedang diakses.

Berdasarkan data dan masalah yang ada, maka dalam penelitian ini akan dibangun suatu Google Chrome *Extension* yang menerapkan *Page Synthesis*. *Extension* ini diharapkan akan mempermudah pengguna dalam memodifikasi tampilan dari halaman *web* yang sedang dikunjungi, khususnya pada manipulasi gambar.

## 2. Kajian Pustaka

Penelitian tentang *adaptive web system* dengan menerapkan metode *Page Synthesis* sudah pernah dilakukan sebelumnya. Pada sebuah penelitian yang berjudul *Adaptive Web Sites: Automatically Synthesizing Web Pages*, teknologi *Page Synthesis* digunakan untuk menyelesaikan masalah *Index Page Synthesis*. *Index Page* adalah suatu halaman yang terdiri dari beberapa tautan yang akan mengarahkan pengguna pada topik – topik tertentu pada sebuah *website*.

Teknologi *Page Synthesis* pada penelitian ini lebih mengkhususkan diri pada sintesa konten halaman dimana *Page Synthesis* digunakan untuk

menambahkan tautan di *Index Page*. Tautan-tautan ini ditambahkan sesuai dengan hasil dari algoritma PageGather yang didasarkan pada *Cluster Mining*. PageGather akan memroses data *Access Log* dari pengguna lalu dikelompokkan berdasarkan data yang mirip. Data yang mirip tersebut kemudian ditautkan pada sebuah *Index Page* yang baru [5].

Dalam penelitian *Multi-objective Optimization for Adaptive Web Site Generation, Index Page Synthesis* dilakukan dengan kombinasi dari 2 algoritma Clustering. Algoritma tersebut adalah DBSCAN dan NSGA-II. Kesimpulan dari penelitian ini adalah kombinasi dari DBSCAN dan NSGA-II dapat menghasilkan sintesa halaman indeks yang lebih baik apabila dibandingkan dengan beberapa algoritma lain, seperti *PageGather*, *K – Means* dan *Hierarchical Agglomerative Clustering* [6].

Perbedaan penelitian ini dengan penelitian sebelumnya adalah, *extension* yang dibuat akan menerapkan *Page Synthesis* pada *Google Chrome Extension*. *Page Synthesis* digunakan untuk memberikan keleluasaan bagi pengguna dalam merubah orientasi gambar yang ada di dalam sebuah halaman *web*. Selain mengubah orientasi gambar, *extension* ini juga dapat mengolah gambar sesuai dengan keinginan pengguna.

*Page Synthesis* merupakan salah satu teknik dalam mencapai *Adaptive Web System*. *Adaptive Web System* adalah suatu sistem *web* yang membangun suatu model dari aktivitas pengguna lalu menggunakan model tersebut sebagai acuan dalam penyampaian informasi [7]. Aktivitas pengguna dapat direkam dari daftar aktivitas pada *web server*. Model yang dibuat dapat dibangun dengan menggunakan kecerdasan buatan atau metode statistika. Ada beberapa cara yang dapat digunakan dalam menerapkan *Adaptive Web System* salah satunya adalah dengan menggunakan *Page Synthesis*.

*Page Synthesis* merupakan sebuah teknik untuk membuat halaman *web* secara dinamis. *Adaptive Web System* biasanya mempunyai cara sintesa yang berbeda. Misalnya, ada beberapa halaman yang sudah dibuat terlebih dahulu, lalu diganti saat diperlukan, atau halaman yang dibuat dihasilkan dari isi *database*. *Page Synthesis* merupakan aspek yang penting, karena dapat melakukan adaptasi yang beragam pada suatu halaman *web*. Ada beberapa jenis sintesa dalam *Page Synthesis*, yaitu [2] : 1) Sintesa bahasa asli; 2) Sintesa berbasis *template*, dan 3) Sintesa halaman baru sepenuhnya.

Jenis sintesa yang digunakan pada penelitian ini adalah sintesa berbasis *template*. *Template* yang digunakan berupa serangkaian tombol yang berguna untuk merubah orientasi dan menerapkan *image filtering* pada gambar yang disorot. *Template* tersebut akan disisipkan pada halaman *web* yang sedang diakses. Hal ini bisa dicapai dengan menggunakan salah satu fitur dari *extension* yang dimiliki oleh Google Chrome, yakni Content Script.

*Extension* dalam Google Chrome adalah sebuah perangkat lunak berukuran kecil yang bisa mengubah dan meningkatkan fungsionalitas dari Chrome. *Extension* hampir tidak memiliki antar muka pengguna. *Extension* menggabungkan semua *file-file* yang ada menjadi 1 paket yang dapat diunduh dan diinstalasikan oleh pengguna. Penggabungan ini berarti *Extension* tidak memerlukan konten dari *web* seperti aplikasi *web* lainnya [8]. *Extension* dapat

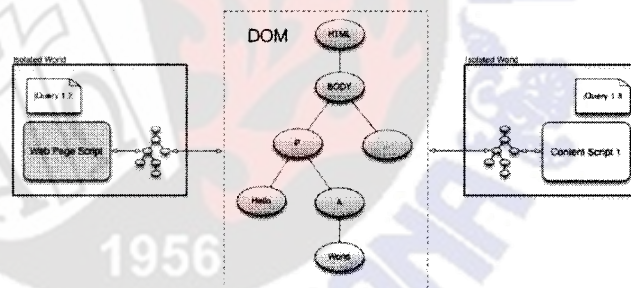
dibangun dengan menggunakan JavaScript, HTML, CSS dan JSON. Salah satu cara pengimplementasian *extension* adalah dengan menggunakan fitur Content Script.

Content Script merupakan salah satu aspek penting dalam membangun sebuah Google Chrome *Extension*. Content Scripts adalah *file* JavaScript yang berjalan di dalam konteks atau isi halaman *web*. Content Script menggunakan standar *Document Object Model* (DOM), sehingga Content Script dapat membaca detail halaman *web* yang sudah dikunjungi atau membuat suatu perubahan pada halaman *web* tersebut.

Beberapa contoh penggunaan Content Script antara lain :

1. Membesarkan ukuran huruf agar teks lebih mudah dibaca.
2. Menemukan URL yang belum mempunyai tautan dan mengubahnya menjadi tautan yang valid.
3. Mengunduh semua gambar dari suatu halaman *web*.

Content Script dieksekusi pada suatu wadah khusus yang disebut *isolated world*. Content Script memiliki akses pada DOM halaman *web* yang dimana Content Script diinjeksikan, tetapi tidak bisa mengakses variabel ataupun fungsi JavaScript pada halaman *web* tersebut. Dengan adanya *isolated world*, setiap content script yang ada diberikan akses untuk mengubah DOM dari *suatu* halaman *web* tanpa mengganggu halaman *web* maupun content script lainnya [8]. Contoh dari *isolated world* dapat dilihat pada gambar 1.

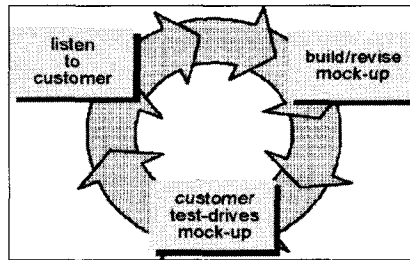


**Gambar 1** Penjelasan *Isolated World*

Dari gambar 2, content script dan *web page script* memiliki *isolated world* masing-masing sehingga tidak terjadi konflik antara *web page script* yang menggunakan jQuery 1.2 dengan content script yang menggunakan jQuery 1.3 [11]. Pada penelitian ini, Content Script digunakan untuk melakukan injeksi fungsi-fungsi yang diperlukan untuk mengubah orientasi gambar dan menerapkan *image filtering*.

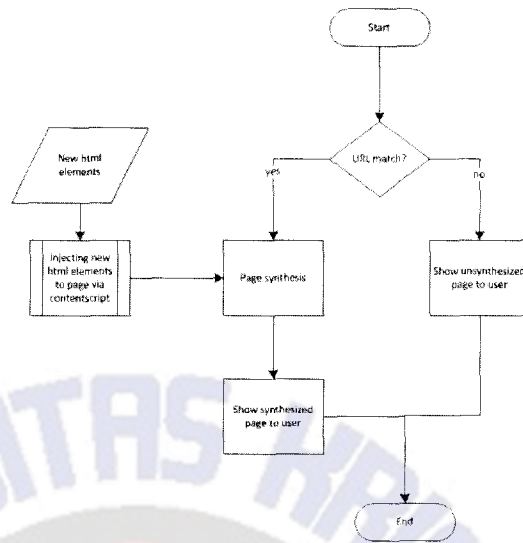
### 3. Metode dan Perancangan

Metode pengembangan yang akan digunakan dalam penelitian ini adalah *prototyping model*. *Prototyping model* merupakan sebuah proses untuk membangun sebuah model dari sebuah sistem berdasarkan dari kebutuhan *user*, dengan kondisi *user* tidak memberikan detail input, proses dan detail output [12]. Bagan *prototyping model* dapat dilihat pada Gambar 2.



**Gambar 2** Bagan *Prototyping Model* [12]

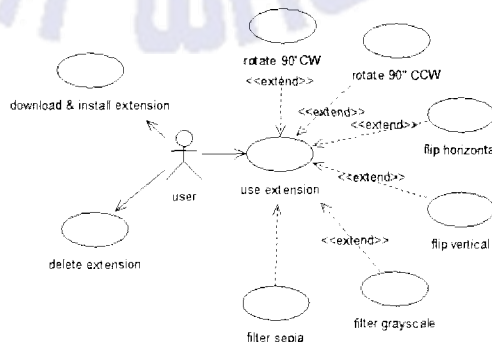
Tahapan pengembangan yang dilakukan dalam penelitian ini adalah: *Listen to customer* atau tahap pengumpulan kebutuhan merupakan tahap awal yang dilakukan dalam pemodelan *prototype*. Pada tahap ini meliputi pengumpulan bahan-bahan yang akan dijadikan referensi untuk merancang dan membangun Google Chrome *Extension* dengan *Page Synthesis*. Pengumpulan kebutuhan dilakukan dengan metode kepustakaan untuk mencari referensi terkait dan difokuskan pada teknik *Page Synthesis*. Sebuah survei dilakukan terhadap 30 mahasiswa FTI UKSW untuk melihat seberapa besar masalah kesalahan orientasi gambar. Hasil dari survey tersebut adalah, 73,3% responden menemukan gambar yang orientasinya salah, 80% responden mengatakan halaman *web* yang ditemukan gambar dengan orientasi yang salah tersebut tidak memiliki fitur untuk mengubah orientasi gambar dan 66,7% responden mengatakan perlu dibuat sebuah aplikasi untuk mengubah orientasi gambar-gambar tersebut. Pengamatan dilakukan pada situs-situs yang menyediakan fitur untuk mengubah orientasi gambar atau menerapkan *image filtering*. Pada tahap ini juga, dilakukan pengumpulan kebutuhan terhadap *software* dan *hardware* yang dibutuhkan. *Build/revise mock-up* atau perancangan perangkat lunak merupakan tahap dimana akan melakukan perancangan. Diagram alur sebagai perancangan awal sistem dapat dilihat pada gambar 3.



Gambar 3 Diagram alur sistem

Sistem akan melakukan pengecekan pada URL halaman *web* yang sedang diakses, jika URL tersebut cocok dengan pola `http://*/*` atau `https://*/*` maka halaman *web* tersebut akan mulai disintesa, jika tidak cocok maka halaman *web* tersebut tidak akan disintesa. Pengecekan ini sebagai pemicu untuk menjalankan `contentscript.js`, sebagai script yang memungkinkan proses sintesa. Proses sintesa akan dimulai dengan menyisipkan fungsi-fungsi yang diperlukan untuk melakukan perubahan pada gambar. Proses penyisipan berjalan secara *parallel* dengan proses *load* halaman *web* tersebut. Sewaktu proses penyisipan dan proses *load* halaman *web* telah selesai, maka *user* dapat dengan mudah melakukan perubahan orientasi gambar.

Perancangan sistem dibuat dengan menggunakan *Unified Modelling Language (UML)*. *Use case diagram* menggambarkan tindakan-tindakan yang dapat dilakukan oleh *actor*. *Use case diagram user* dapat dilihat pada gambar 4.

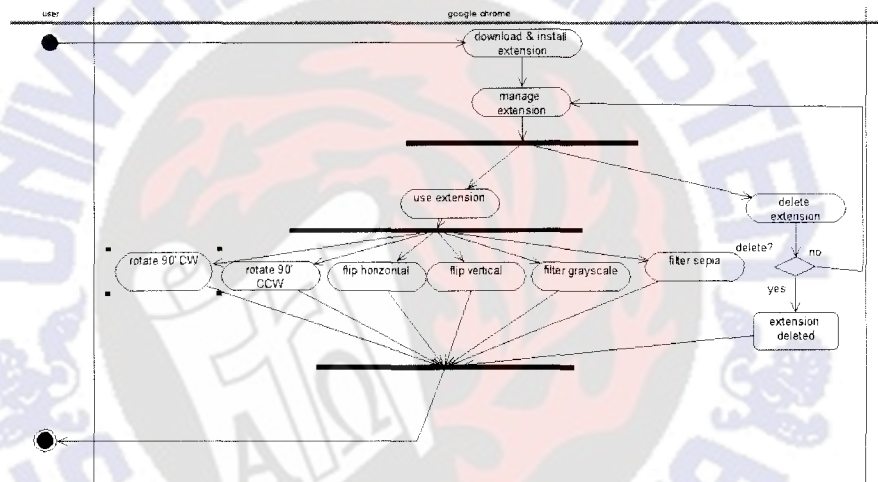


Gambar 4 Use Case Diagram

Di dalam *Use case diagram extension* ini hanya terdapat 1 aktor, *user*. *User*

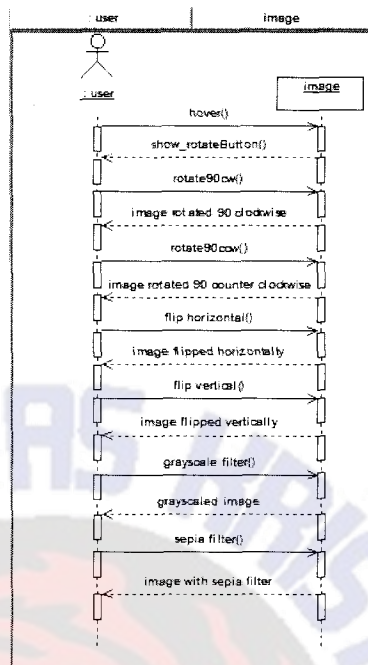
dapat mengunduh dan melakukan instalasi *extension*, menghapus *extension* maupun menggunakan *extension* yang sudah diinstalasikan. *User* dapat melakukan rotasi 90° searah jarum jam atau berlawanan dengan arah jarum jam pada gambar yang diinginkan. *User* dapat memutar gambar secara horizontal maupun vertikal. Selain mengubah orientasi gambar, *User* juga dapat menerapkan beberapa metode pengolahan gambar antara lain, *grayscale* dan *sepia*

*Activity diagram*, digunakan untuk menggambarkan sebuah aktifitas yang dapat dilakukan oleh seorang aktor dari awal aktifitas tersebut hingga berakhir. *Activity diagram user* dapat dilihat pada Gambar 5. *User* mengunduh dan melakukan proses instalasi *extension* terlebih dahulu di dalam Google Chrome. *User* dapat menggunakan *extension* untuk mengubah orientasi dari suatu gambar atau menerapkan metode pengolahan gambar pada gambar tersebut. *User* juga dapat menghapus *extension* jika sudah tidak diperlukan lagi.



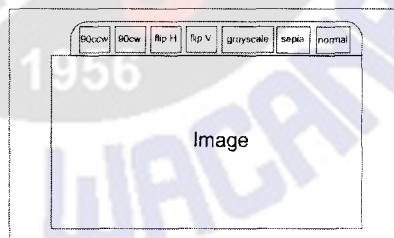
Gambar 5 User Activity Diagram

*Sequence diagram*, digunakan untuk menggambarkan interaksi yang terjadi antar komponen di dalam dan di sekitar sistem (pengguna, *interface*, dan sebagainya) dalam bentuk pesan dan digambarkan terhadap waktu. *Sequence diagram* dapat menggambarkan skenario dan langkah-langkah yang dilakukan sehingga mendapatkan sebuah hasil. Gambar 6 menunjukkan *sequence diagram* saat *user* menggunakan *extension*. *User* menyorot gambar yang ada pada halaman *web* yang sedang diakses, lalu mengubah orientasi gambar dengan menggunakan sekelompok tombol yang dimunculkan oleh fungsi *show\_rotateButton*.



**Gambar 6** User Sequence Diagram

Perancangan antar muka pengguna, dilakukan agar pembuatan aplikasi lebih mudah dan terarah. Antar muka pengguna yang dirancang hanya ada 1 yaitu sekelompok tombol untuk mengubah orientasi gambar dan mengolah gambar ketika pengguna menyorot sebuah gambar. Gambar 7 menunjukkan rancangan antar muka pengguna tersebut.



**Gambar 7** Rancangan Antar Muka Pengguna

*Customer test-drives mock-up* atau tahap evaluasi *prototype* merupakan tahap pengevaluasian atau pengujian terhadap sistem *software* yang telah dibangun apakah sudah memenuhi kebutuhan. Terdapat dua pembuatan purwarupa untuk menghasilkan aplikasi yang sesuai dan tidak dievaluasi lagi. purwarupa pertama bisa dilihat pada gambar sebelah kiri di Gambar 8.



**Gambar 8** Hasil Purwarupa Pertama dan Purwarupa Kedua



Setelah menghasilkan purwarupa pertama, dilakukan evaluasi dari hasil purwarupa tersebut. Hasil evaluasi purwarupa pertama antara lain, purwarupa pertama hanya bisa mengubah orientasi gambar saja dan belum bisa menerapkan image filtering. Purwarupa kedua dilakukan karena purwarupa pertama masih memiliki kekurangan. Hasil dari purwarupa kedua dapat dilihat pada gambar sebelah kanan pada Gambar 8.

Pada hasil purwarupa kedua terlihat beberapa penambahan fungsi yang dilakukan yaitu penambahan fungsi untuk menerapkan *filter grayscale*, *filter sepia* dan fungsi *reset*, untuk mengembalikan gambar ke kondisi semula. Hasil evaluasi pada *prototype* kedua antara lain, yaitu penggunaan judul halaman yang sederhana, penggunaan objek 3d yang bukan objek 3d mata, menu manipulasi objek 3d yang belum lengkap.

#### 4. Hasil dan Pembahasan



Gambar 9 Icon *Extension Hoveraction*



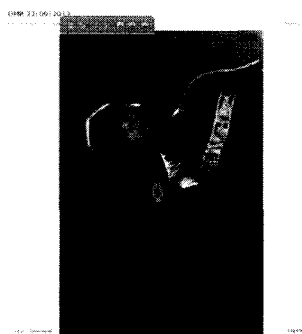
Gambar 10 Antar muka pengguna di dalam menu *Extension*

Gambar 9 merupakan tampilan dari *toolbar* Google Chrome setelah proses instalasi *extension* sedangkan Gambar 10 adalah tampilan yang akan dilihat pengguna ketika membuka menu pengaturan *extension* pada Google Chrome setelah melakukan proses instalasi *extension*. Menu ini dapat diakses dengan buka URL `chrome://extensions/.Extension`.



Gambar 11 Antar muka *Extension*

Antar muka *extension* adalah serangkaian tombol yang muncul saat pengguna menyorot sebuah gambar di dalam halaman yang sudah disintesa. Antar muka ini memiliki 7 fungsi yaitu, memutar gambar 90° berlawanan dengan jarum jam, memutar gambar 90° searah dengan jarum jam, membalikkan gambar secara horizontal, membalikkan gambar secara vertikal, *grayscale filter*, *sepia filter* dan mengembalikan gambar ke kondisi semula. Antar muka ini dapat dilihat pada gambar 11.



**Gambar 12** Halaman yang sudah disisipi *Extension*

Halaman sintesa adalah halaman yang sudah disisipi kode dari *extension*. Pada halaman ini, setiap gambar yang disorot dengan *cursor* akan menampilkan sekelompok tombol di sebelah kiri atas, seperti gambar 12.



**Gambar 13** Gambar yang diubah orientasinya dengan *Extension*

Setiap gambar di dalam halaman yang sudah disintesa, kini dapat dirubah orientasinya dengan mudah. Perubahan orientasi pada gambar dapat dilakukan dengan menekan salah satu dari 6 tombol yang ada. Hasil dari perubahan orientasi dan penerapan *filter sepia* dapat dilihat pada Gambar 13. Untuk membuat suatu Google Chrome *Extension* dengan *Page Synthesis* yang pertama harus dilakukan adalah menulis *manifest.json*. *Manifest.json* digunakan untuk menyimpan informasi penting dari Google Chrome *Extension*. Kode untuk *manifest.json* ini dapat dilihat pada Kode Program 2.

#### **Kode Program 2** Kode *manifest.json*

```
1. {
2.   "manifest_version": 2,
3.   "name": "Hoveraction",
4.   "description": "a simple extension for rotating Image on the fly",
5.   "version": "1.0",
6.   "icons": {
7.     "128": "icon/hovertate-128.png",
8.     "48": "icon/hovertate-48.png",
9.     "32": "icon/hovertate-32.png",
10.    "16": "icon/hovertate-16.png"
11.   },

```

```

12.     "content_scripts": {
13.       {
14.         "matches": [ "http://**/*", "https://**/*"],
15.         "js" : ["contentscript.js"]
16.       }
17.     },
18.     "browser_action": {
19.       "default_icon": "images/icon.png"
20.     },
21.     "permissions": [
22.       "https://**/*",
23.       "http://**/*"
24.     ]
25.   }

```

Kode Program 2 pada baris kedua sampai 11 merupakan kode untuk informasi singkat yang mengenalkan *extension* pada pengguna. Baris 12-17 adalah bagian untuk mendefinisikan *content script*. Baris ke 14 untuk membandingkan *URL* dari suatu halaman *web*, jika *URL* tersebut memenuhi kondisi, maka halaman *web* tersebut akan disintesa dengan *content script*. Lokasi *content script* didefinisikan pada baris ke-15. *Default\_icon* digunakan untuk menampilkan *icon* dari *Extension*, seperti pada gambar 8.

### Kode Program 3 Pseudocode Penyisipan CSS, jQuery dan div globalImageID

1. Create new css for user interface; Append to body tag;
2. Create new script to call jQuery; Append script to head tag;
3. Create new div; Set ID to overlayDiv; Hide div;
4. Set new div innerHTML with buttons and onClick functions; Append to body tag;
5. Create new div; Set ID to globalImageID;
6. Initialize new div innerHTML value; Append to body tag;

Teknik *Page Synthesis* pada *extension* ini diterapkan melalui *contentscript.js*. Script ini akan menambahkan berbagai elemen HTML baru yang digunakan untuk mengubah orientasi gambar maupun menerapkan *image filter* pada gambar itu. Kode Program 3 merupakan pseudocode untuk menginjeksikan css baru, inisialisasi jQuery, menambahkan div dengan ID *globalImageID* pada halaman *web* yang sedang diakses.

### Kode Program 4 Pseudocode Fungsi getDegree

1. Create new script; Set ID to getDegree;
2. Set new script innerHTML with getDegree function;
3. Function getDegree() {
4. idImg = get globalImageID innerHTML value;
5. var objRotate = get element with the ID of idImg;
6. var currentDegree = get objRotate webkit transform value;
7. Initialize variable r, x, y and angle;
8. If currentDegree is not empty then
9. Var temp = split currentDegree value with 'rotate'
10. For every value of temp
11. Do get the r value which is the value of webkit transform rotate; get the x value which is the value of webkit transform rotateX; get the y value which is the value of webkit transform rotateY.
12. Set angle = 'r"+"#"+x+"#"+y';
13. Else Set angle = '0"+"#"+0+"#"+0';
14. Return angle; End function getDegree; Append script to head tag
15. Create new script; Set ID to cw90;
16. idImg = get globalImageID innerHTML value;
17. var angle = split the return value of getDegree function with '#';

```

18. var objRotate = get element with the ID of idImg;
19. set the objRotate's webkit transform rotate value to angle[1] + 90;
20. set the objRotate's webkit transform rotateX value to angle[2];
21. set the objRotate's webkit transform rotateY value to angle[3];
22. Append script to head tag;
    
```

Fungsi *getDegree* berfungsi untuk memberikan dan mengambil nilai *webkit transform* dari gambar yang disorot. Pada awalnya fungsi ini akan mengambil nilai *webkit transform* dari gambar yang disorot, jika gambar tersebut memiliki nilai *webkit transform* maka nilai-nilai tersebut akan dipecah berdasarkan string 'rotate' dan dimasukkan pada variable r, x, dan y. Fungsi *getDegree* akan mengembalikan nilai string berupa '#x#y', jika gambar tersebut tidak memiliki nilai *webkit transform*, maka kembalian yang diberikan adalah '0#0#'.  
 Nilai kembalian dari fungsi *getDegree* akan digunakan pada fungsi-fungsi untuk merubah orientasi gambar. Fungsi-fungsi ini akan memanggil fungsi *getDegree*, lalu memecah nilai kembalian berdasarkan string '#'. Masing-masing dari nilai kembalian tersebut lalu dimasukkan ke dalam *webkit transform*. Nilai kembalian pertama akan dimasukkan ke dalam *rotate*, nilai kedua dimasukkan ke dalam *rotateX* dan yang terakhir dimasukkan ke dalam *rotateY*. Ketiga nilai *webkit transform* itu lalu diproses sesuai dengan fungsinya. Fungsi *cw90* akan menambahkan 90 pada nilai *webkit transform rotate*, fungsi *ccw90* akan mengurangi 90 pada nilai *webkit transform rotate*. Fungsi *flipH* akan menambahkan 180 pada nilai *webkit transform rotateX*. Fungsi *flip V* akan menambahkan 180 pada nilai *webkit transform rotate*.

Dalam perancangan *extension* ini, pengujian *extension* dilakukan dengan *Alpha* dan *Beta Test*. Pada *Alpha Testing*, pengujian dilakukan pada setiap operasi perubahan orientasi gambar dan penerapan metode pengolahan gambar. Skenario yang berhasil diberikan tanda (✓) sedangkan jika skenario gagal, diberikan tanda (✗). Hasil dari *Alpha Testing* dapat dilihat pada Tabel 2.

Tabel 2 Hasil dari *Alpha Testing*

Fungsi	Skenario	Input	Hasil	Validitas
<i>Hover Image</i>	Memuncul kan tombol	Menyorot gambar	Tombol muncul	✓
<i>Rotate and Flip Image</i>	Memutar dan membalikkan gambar	Menekan tombol <i>rotate</i> dan <i>flip</i>	Orientasi gambar berubah sesuai tombol yang ditekan	✓
<i>Image Filter</i>	Menerapkan <i>Grayscale</i> dan <i>Sepia</i> pada gambar	Menekan tombol <i>grayscale</i> atau <i>sepia</i>	Gambar berubah warna sesuai tombol yang ditekan	✓
<b>Kombinasi <i>rotate, flip dan image filter</i></b>	Melakukan kombinasi <i>rotate, flip, dan image filter</i>	Menekan kombinasi tombol <i>rotate, flip, sepia, grayscale</i>	Gambar berubah warna sesuai kombinasi tombol yang ditekan	✓
<i>Reset Image</i>	Mengembalikan gambar pada kondisi awal	Menekan tombol Normal	Gambar kembali pada kondisi awal	✓

Hasil pengujian *Alpha* yang terlihat pada tabel 2 memperlihatkan bahwa operasi-operasi yang ada di dalam *extension* bekerja dengan baik. Pada *Beta Testing*, *extension* diujicobakan pada 30 Mahasiswa FTI UKSW. Mahasiswa menggunakan *extension* untuk mengubah orientasi gambar dan menerapkan pengolahan gambar pada sebuah gambar di dalam halaman *web* yang diakses, kemudian mengisi sebuah kuisioner. Daftar pertanyaan pada beserta tabulasi jawaban dapat disimak pada Tabel 3 berikut ini.

**Tabel 3** Daftar Pertanyaan Kuisioner dan Tabulasi Jawaban Responden

No	Soal	Skor Jawaban				
		1	2	3	4	5
1	Apakah <i>extension</i> ini mudah untuk digunakan?	5	18	7	0	0
2	Apakah <i>extension</i> ini membantu anda dalam melakukan proses perubahan orientasi gambar dan <i>image filtering</i> ?	2	19	9	0	0
3	Apakah fungsi-fungsi di dalam <i>extension</i> ini sudah lengkap?	0	17	13	0	0
4	Apakah <i>user interface extension</i> ini mudah dimengerti?	9	15	6	0	0
5	Apakah anda tertarik untuk menggunakan <i>extension</i> ini?	11	13	6	0	0
<b>Jumlah</b>		27	82	29	0	0

Berdasarkan Tabel 3, terlihat bahwa rata-rata semua siswa memberikan jawaban a, b dan c. Tidak ada mahasiswa yang memberikan nilai d atau e. Melalui hasil kuisioner ini, dapat disimpulkan bahwa sebagian besar mahasiswa menganggap bahwa *extension* ini mudah digunakan, membantu dalam proses proses perubahan orientasi gambar dan *image filtering*, memiliki fungsi-fungsi yang lengkap, mempunyai *user interface* yang mudah dimengerti dan tertarik untuk menggunakan *extension* ini.

## 5. Simpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat ditarik kesimpulan bahwa: 1) Google Chrome *Extension* yang dibuat mudah digunakan; 2) *Extension* ini membantu dalam proses proses perubahan orientasi gambar dan *image filtering*; 3) *Extension* ini memiliki fungsi-fungsi yang lengkap; 4) *User* dapat menyesuaikan orientasi gambar sesuai dengan keinginannya.

Saran untuk pengembangan aplikasi adalah: 1) pengaplikasian *page synthesis* tidak hanya pada *presentation mode*, tetapi bisa diperluas hingga *user*

model; 2) Fungsi-fungsi *image filtering* bisa ditambahkan lagi.

## 6. Daftar Pustaka

- [1] Brusilovsky, Peter, 2002, From Adaptive Hypermedia to the Adaptive Web, pdf.aminer.org/000/490/548/from\_adaptive\_hypermedia\_to\_the\_adaptive\_web.pdf. Diakses tanggal 30 Mei 2013.
- [2] Hossein, Sadat, Ali, A. Ghorbani, 2005, On The Evaluation of Adaptive Web Systems, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.1361.pdf>. Diakses tanggal 30 Mei 2013.
- [3] Hossein, Sadat, Ali, A. Ghorbani, 2005, A presentation specification language for adaptive web systems, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.131.751.pdf>. Diakses tanggal 30 Mei 2013.
- [4] StatCounter, 2013, <http://gs.statcounter.com/#browser-ww-yearly-2008-2013>, Diakses tanggal 1 Juni 2013.
- [5] Perkowitz, Mike, Etzioni, Oren, 1998, Adaptive Web Sites: Automatically Synthesizing Web Pages, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.1361.pdf>. Diakses tanggal 30 Mei 2013.
- [6] Jain, Prateek, Mitra, Pabitra, 2005, Multi-objective Optimization for Adaptive Web Site Generation, [http://link.springer.com/content/pdf/10.1007%2F11590316\\_105.pdf](http://link.springer.com/content/pdf/10.1007%2F11590316_105.pdf). Diakses tanggal 31 Mei 2013.
- [7] Peter, Brusilovsky, Kobsa Alfred, Nejd, Wolfgang, 2007, The Adaptive Web Methods and Strategies of Web Personalization, <http://link.springer.com/book/10.1007/978-3-540-72079-9/page/1>. Diakses tanggal 12 Juni 2013.
- [8] Google, 2012, <http://developer.chrome.com/extensions/index.html>, Diakses tanggal 12 Juni 2013.
- [9] S. T., Hendro, 2007, Penggunaan AJAX pada pengembangan Aplikasi Web. Volume 4 No.1 Februari 2007 : 1-100.
- [10] JSON, 2011, <http://json.org>, Diakses tanggal 1 Juni 2013.
- [11] Google, 2012, [http://developer.chrome.com/extensions/content\\_scripts.html](http://developer.chrome.com/extensions/content_scripts.html), Diakses tanggal 12 Juni 2013.
- [12] Pressman, Roger. 2001. Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu). Yogyakarta: Andi.
- [13] Ayuliana, 2009, Testing dan Implementasi, [http://rifiana.staff.gunadarma.ac.id/Downloads/files/26083/Teknik Pengujian perangkat Lunak - Black Box.pdf](http://rifiana.staff.gunadarma.ac.id/Downloads/files/26083/Teknik_Pengujian_perangkat_Lunak_-_Black_Box.pdf). Diakses pada tanggal 4 Juni 2013.