

Pembuatan Aplikasi *Benchmarking* Perambah *Web Mobile*

Anugerah Christian Rompis, Ir. H. F. Wowor, M.Kom, A.S.M. Lumenta, ST., MT., V. Tulenan, S.Kom., MTI.
Jurusan Teknik Elektro Fakultas Teknik Unsrat. starleaf1@gmail.com

Abstract – *With Android developing into the most used operating system in the world, various software developers produce various web browsers for Android. To measure the performance of the browsers, the author develops a web browser benchmarking application. The author develops the application using PHP programming language and supported by MySQL database management system using extreme programming methodology. The application measures web page rendering speed, tests the browser's HTML5 video playback capability, and measures the browser's JavaScript performance.*

Keywords: *benchmark, browser, extreme programming, HTML, JavaScript, mobile device, performance, PHP, web.*

Abstrak – Dengan berkembangnya sistem operasi Android menjadi sistem operasi yang paling banyak digunakan di dunia, berbagai pengembang perangkat lunak mengembangkan berbagai perambah *web* untuk Android. Untuk mengukur kinerja berbagai perambah *web*, maka penulis membangun aplikasi *benchmarking* perambah *web*. Penulis mengembangkan aplikasi dengan menggunakan bahasa pemrograman PHP dan didukung sistem basis data MySQL memakai metode *extreme programming*. Aplikasi tersebut dapat mengukur kecepatan render halaman *web*, menguji kemampuan pemutaran video HTML5, serta mengukur kinerja JavaScript

Kata kunci: *benchmark, extreme programming, HTML, JavaScript, perambah web, perangkat mobile, kinerja, PHP, web.*

I. PENDAHULUAN

Dengan berkembangnya teknologi jaringan nirkabel dan telekomunikasi seluler, pertukaran data belakangan ini meluas hingga mencakup berbagai peralatan elektronik yang bersifat mobile. Akses Internet yang tadinya terbatas hanya pada perangkat-perangkat seperti PC, server, dan perangkat-perangkat yang relatif berat lainnya kini dapat diperoleh melalui perangkat-perangkat yang bersifat portabel dan relatif ringan.

Pada mulanya, akses Internet untuk perangkat portabel hanya terbatas pada halaman-halaman yang ditulis khusus untuk perangkat tersebut. Perambahan juga hanya dapat dilakukan di atas protokol khusus. Teknologi ini disebut WAP. Teknologi ini ditujukan untuk mengadaptasikan perambahan *web* dengan kemampuan perangkat-perangkat portabel yang waktu itu masih sangat terbatas.

Kini, dengan lahirnya teknologi Wi-Fi, 3G dan HSPA, transfer data dengan volume yang relatif lebih besar

melalui perangkat portabel dapat diwujudkan. Hal ini didukung pula oleh lahirnya perangkat-perangkat smartphone, dan tablet PC serta sistem operasi mobile seperti iOS dan Android yang memungkinkan pengalaman merambah *web* yang mendekati pengalaman merambah dengan menggunakan PC desktop atau laptop.

Sekarang, sistem operasi Android telah menguasai mayoritas pasaran sistem operasi mobile. Mengikuti peristiwa tersebut, pengembang perangkat perambah *web* mengembangkan berbagai perambah untuk para pengguna *web* dengan berbagai fitur. Beberapa di antaranya adalah Mozilla yang mengembangkan Firefox for Android, Google yang mengembangkan Chrome for Android, dan Dolphin, Inc. yang mengembangkan Dolphin Browser.

Salah satu faktor yang diperhatikan oleh pengguna maupun pengembang adalah performa. Meskipun demikian, hingga saat ini metode-metode pengukuran performa perambah *web* masih bersifat terbatas pada aspek-aspek tertentu, atau berupa pengukuran yang sifatnya subjektif. Kebanyakan metode tersebut juga ditujukan untuk perambah yang dijalankan di atas perangkat non-mobile.

Berdasarkan hal-hal tersebut, penulis berniat menyusun skripsi dengan judul “Pembuatan Aplikasi *Benchmarking* Perambah *Web Mobile*.”

II. LANDASAN TEORI

A. Perambah Web

Perambah *web* adalah suatu perangkat lunak yang berfungsi utama membuka halaman-halaman *web*, terutama yang tersedia di Internet. Perambah *web* mengambil arsip HTML yang tersedia dalam server, menginterpretasi isi halaman tersebut, kemudian menampilkannya kepada pengguna.

B. Hypertext Markup Language (HTML)

Hypertext Markup Language adalah sebuah bahasa mark-up yang merupakan bahasa standar untuk menampilkan sebuah halaman *web*. Sebuah halaman HTML terdiri atas elemen-elemen yang menyusun isi dari halaman tersebut. Elemen-elemen HTML membentuk sebuah struktur semantik dari informasi yang perlu disampaikan dalam halaman tersebut (misalnya judul, isi, tautan, catatan kaki, dan lain-lain). Elemen-elemen tersebut akan diinterpretasi oleh perangkat perambah *web*, kemudian ditampilkan sebagai teks dengan berbagai format, gambar, video, dan lain-lain.

Elemen-elemen HTML ditandai dengan *tags* yang dibungkus dengan tanda lebih kecil dari (<) dan tanda lebih besar dari (>). *Tag* biasanya berpasangan, mengapit teks yang akan dijadikan elemen HTML. *Tag* depan dapat memuat atribut-atribut tambahan yang dapat mengubah penampilan dan perilaku dari elemen yang bersangkutan.

C. JavaScript

JavaScript adalah sebuah bahasa *scripting* yang biasa diimplementasikan dalam *web* untuk menyediakan interaktifitas bagi pengguna dan menghadirkan halaman *web* yang bersifat dinamis. JavaScript bekerja secara *client-side* dalam *Document Object Model* dalam sebuah halaman HTML. JavaScript berjalan segera sesudah semua elemen HTML selesai dimuat. JavaScript memperlakukan elemen-elemen HTML sebagai kumpulan *objects*, dan dapat mengubah atribut-atribut sebuah elemen HTML, atau isi dari elemen itu sendiri, pada halaman tanpa perlu meminta halaman baru dari *web server*.

D. Cascading Style Sheet (CSS)

Cascading Style Sheet adalah sebuah bahasa *style sheet* yang digunakan untuk menentukan tampilan dan *formatting* dari sebuah dokumen yang ditulis dalam sebuah *mark-up language*, pada umumnya dokumen HTML. Manfaat utama CSS adalah untuk memisahkan definisi dokumen dengan definisi tampilannya. Hal ini memudahkan pembacaan dokumen yang dimaksud serta memungkinkan penyeragaman tampilan beberapa dokumen sekaligus. CSS mengatur berbagai aspek penampilan dari dokumen HTML seperti warna, bentuk teks, tata letak, dan lain-lain.

E. Hypertext Transmission Protocol (HTTP)

HTTP merupakan protokol yang dipakai untuk mengirimkan halaman-halaman *web* beserta *resource* lain yang akan ditanamkan dalam halaman tersebut.

HTTP mengikuti model *client-server* dan beroperasi dengan metode *request-response*. HTTP berada pada *application layer* dalam *TCP/IP protocol suite*. Dalam sebuah sesi HTTP, komputer *client* akan membangun sebuah koneksi dengan server (biasanya pada port 80). Server kemudian menunggu adanya permintaan HTTP dari *client* (biasanya sebuah perambah *web*), dan memberikan jawaban berupa pesan status mengenai permintaan tersebut. Apabila tidak terjadi *error*, pesan status ini akan diikuti dengan pesan dari server itu sendiri berupa *resource* yang tadi diminta oleh *client*. Apabila terjadi *error*, pesan status akan berisi keterangan tentang *error* tersebut.

F. PHP

PHP merupakan sebuah bahasa pemrograman yang khusus dikembangkan untuk pengembangan *web*. PHP berjalan di atas server HTTP dan dieksekusi saat server menerima request HTTP dari client. Ketika request HTTP diterima oleh server atas sebuah berkas yang mengandung script PHP, maka server akan mengeksekusi script tersebut. Keluaran dari eksekusi tersebut kemudian dikirimkan kepada client sebagai jawaban atas request-nya.

Keluaran dari sebuah eksekusi PHP biasanya berupa teks, dan biasanya adalah script HTML yang akan di-render oleh perambah sebagai halaman *web* yang kemudian ditampilkan kepada pengguna akhir.

G. MySQL

MySQL adalah sebuah sistem manajemen basis data yang menggunakan bahasa *Structured Query Language* (SQL). MySQL merupakan sebuah perangkat lunak yang bersifat *open source*. MySQL digunakan pada berbagai aplikasi seperti TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal, dan perangkat lunak lainnya.

H. Google Charts API

Google Charts API adalah sebuah peralatan berupa *application programming interface* untuk bahasa pemrograman JavaScript yang bermanfaat untuk me-render berbagai grafik dan diagram ke dalam suatu halaman *web*.

Google Charts API menyediakan berbagai *classes* di mana pengembang dapat memasukkan data yang akan di-render, dan menetapkan berbagai aspek dari grafik yang akan digambar.

Grafik yang digambar di-render sebagai objek-objek HTML dalam sebuah div yang sudah ditetapkan oleh pengembang aplikasi.

I. Video Container Format

MP4

MP4 adalah sebuah format yang dikembangkan oleh *Moving Picture Experts Group*. Sebuah file dengan format MP4 dapat mengandung video dengan *codec* antara lain MPEG-2 Part 2, MPEG-4 ASP, H.264/MPEG-4 AVC, H.263, VC-1, dan Dirac. Sementara audio yang dapat disimpan dalam file tersebut bisa memiliki *codec* yaitu MPEG-2/4 (HE)-AAC, MPEG-1/2 Layers I, II, III (MP3), AC-3, Apple Lossless, ALS, SLS, Vorbis, dan lain-lain.

WebM

WebM adalah sebuah format yang bersifat *open source* yang disponsori oleh *Google*. Menurut dokumentasi WebM yang tersedia *online*, video dalam sebuah file WebM disimpan dengan *codec* VP8 sementara audio disimpan dengan *codec* Vorbis.

OGG

Ogg adalah sebuah format yang bersifat *open source* yang dikembangkan oleh Xiph. Ogg dapat berisi video dengan *codec* Theora, Dirac, OggUVS, MNG, dan lain-lain. Sementara audio yang dapat disimpan dengan *container* Ogg dapat memiliki *codec* Vorbis, FLAC, Speex, Opus, OggPCM, dan lain-lain.

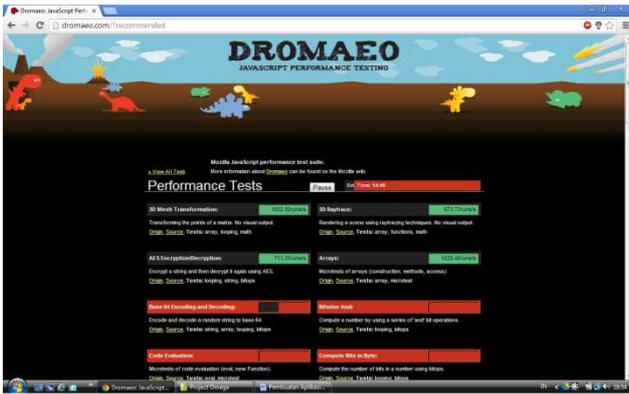
J. Metode dan Perangkat Benchmarking

Metode Lifehacker

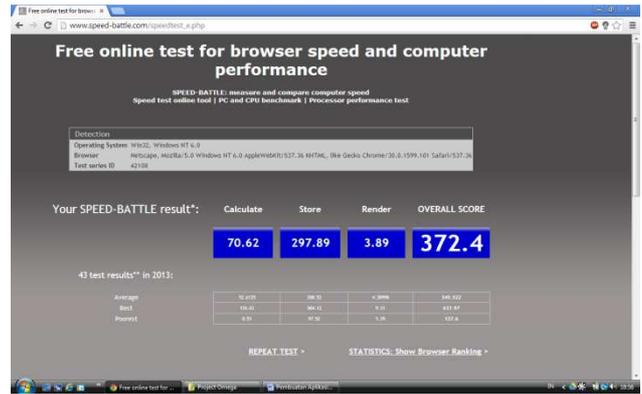
Lifehacker adalah sebuah situs yang dimiliki oleh Gawker Media yang berisi artikel-artikel tentang kiat-kiat dan trik-trik meningkatkan produktifitas dan efisiensi kerja, serta perangkat-perangkat lunak komputer yang dapat membantu mencapai tujuan tersebut.

Pengujian yang dilakukan di Lifehacker dijalankan pada sebuah PC dengan CPU Intel Core i7 3,8 GHz; 12 GB RAM; jaringan *ethernet*; dengan sistem operasi Windows 8.

Metode yang dipakai oleh Lifehacker menguji aspek-aspek yaitu kecepatan *start-up*, kecepatan pemuatan halaman, penggunaan memori, serta kinerja JavaScript dan CSS.



Gambar 1. Dromaeo Testing Suite



Gambar 2. Speed-Battle

Dromaeo Testing Suite

Gambar 1 menunjukkan sebuah *screenshot* dari *Dromaeo Testing Suite*. Dromaeo adalah sebuah aplikasi *benchmarking* berbasis *web* yang disusun oleh Mozilla. Dromaeo menitikberatkan kepada kinerja JavaScript, namun terdapat beberapa pengujian dengan sasaran mengukur kemampuan *CSS engine*. Pengujian-pengujian dalam Dromaeo dikelompokkan menjadi beberapa *sub-test* berdasarkan aspek yang akan diuji. Dromaeo dirancang secara spesifik untuk mengukur kinerja eksekusi *script* dan menghindari masuknya faktor-faktor yang berasal dari proses *rendering*.

Speed-Battle

Gambar 2 menunjukkan sebuah *screenshot* dari perangkat Speed-Battle. Speed-Battle adalah sebuah aplikasi *benchmarking* berbasis *web* yang menguji kemampuan perhitungan, penyimpanan, serta *rendering* dari JavaScript dengan cara menjalankan algoritma-algoritma sederhana pada perambah *web*. Hasil pengujian disimpan ke dalam sebuah basis data. Aplikasi ini dapat menampilkan statistik yang bersumber dari hasil-hasil pengujian yang telah dilakukan oleh *users* sebelumnya.

K. Extreme Programming (XP)

Extreme Programming (XP) adalah suatu metode perancangan aplikasi secara cepat. Tulisan-tulisan yang berkenaan dengan XP banyak dijumpai pada tahun-tahun 1980-an, tetapi salah satu dari karya-karyaterpenting dalam XP ditulis oleh Kent Beck.

XP merekomendasikan kolaborasi dalam bentuk komunikasi yang informal dan bersifat lisan di antara seluruh pemegang andil. *Feedback* yang berlangsung harus berkesinambungan dan tim pengembang harus siap beradaptasi. *Feedback* bisa bersumber dari perangkat lunak itu sendiri, dari pelanggan, maupun dari anggota tim pengembang yang lain. Apabila terdapat perubahan terhadap *requirements*, tim pengembang harus dapat melakukan perubahan dengan segera.

Dokumentasi yang dihasilkan oleh metode *extreme programming* bersifat singkat dan padat. Strategi pengujian pada XP dilaksanakan dengan *testing units*. *Testing units* merupakan perwakilan dari fungsi-fungsi perangkat lunak yang sesungguhnya, dan setiap hasil *testing unit* membawa *feedback* kepada tim pengembang.

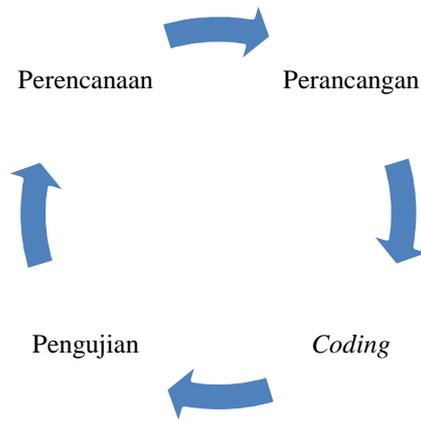
Beck menekankan pentingnya “keberanian” dalam mengembangkan dengan XP. Setiap anggota tim pengembang harus menanamkan nilai-nilai bahwa setiap keputusan yang diambil dalam proses pengembangan harus didasarkan pada kebutuhan yang ada saat itu juga, dan bukan atas kemungkinan di masa depan. Tim pengembang harus mengambil resiko menyusun ulang perangkat lunaknya apabila di masa depan terdapat perubahan yang drastis.

Proses XP memiliki ciri khas dibandingkan dengan metodologi pengembangan aplikasi yang lain. XP bersifat dini, konkrit, dan mendapatkan *feedback* secara berkesinambungan melalui *cycles* yang berdurasi pendek. XP menggunakan pendekatan perencanaan yang bersifat inkremental, yang secara cepat mendatangkan rencana keseluruhan yang diharapkan akan berevolusi selama proyek berlangsung. XP mampu menjadwalkan implementasi fungsionalitas secara fleksibel sesuai dengan perubahan pada kebutuhan bisnis. XP mengandalkan komunikasi lisan, pengujian, serta *source code* dalam rangka menyampaikan struktur dan tujuan dari perangkat lunak. XP mengandalkan kolaborasi para *programmer* yang memiliki keahlian yang biasa saja.

XP mengandalkan praktek-praktek yang sesuai bagi intuisi para *programmer* yang bersifat jangka pendek, tapi juga sesuai bagi tujuan proyek yang berjangka panjang.

Kepada para *programmer*, XP menjanjikan bahwa mereka bisa mengerjakan hal-hal yang sungguh berarti, bahwa mereka tidak akan menghadapi situasi menakutkan sendirian. Mereka akan bisa melakukan apapun yang mereka mampu lakukan agar perangkat lunak yang mereka buat berhasil. Mereka akan bisa mengambil keputusan-keputusan terbaik, dan bukan keputusan yang di luar kualifikasi mereka.

Kepada para pelanggan dan manajer, XP menjanjikan bahwa mereka akan mendapatkan nilai penuh dari setiap minggu pemrograman. Tiap minggu, mereka akan melihat kemajuan yang konkrit pada tujuan-tujuan yang mereka kejar. Mereka dapat mengubah arah proyek di tengah proses pengembangan, tanpa mengakibatkan biaya yang terlalu tinggi.



Gambar 3. Diagram sebuah proses iterasi dalam XP

Extreme programming merupakan sebuah metodologi pengembangan perangkat lunak yang bersifat iteratif. Perangkat lunak dikembangkan berulang-ulang hingga akhirnya diterima oleh calon pengguna. Secara garis besar, setiap iterasi diawali dengan perencanaan, diikuti dengan perancangan, selanjutnya *coding*, dan pengujian. Proses iterasi XP ditunjukkan oleh Gambar 3.

Perencanaan

Perencanaan adalah tahap di mana berbagai aspek dari perangkat lunak ditentukan. Dalam tahap perencanaan, terdapat pengumpulan *user stories*, perencanaan *acceptancetest*, dan *iteration planning*.

User stories adalah gambaran dari fitur-fitur, keluaran program, serta fungsionalitas dari perangkat lunak yang diinginkan oleh pengguna. Sebuah *user story* disusun sedemikian rupa sehingga detail yang ada dalam *user story* tersebut hanya cukup untuk para *programmer* untuk memperkirakan jangka waktu yang mereka butuhkan untuk mengimplementasikan *story* tersebut ke dalam perangkat lunak. Sebuah *user story* juga tidak mengandung rancangan teknis secara terperinci. Hal-hal seperti DBMS, tataletak GUI, algoritma, dan lain-lain tidak ada di dalam *user stories*.

Iteration planning adalah tahap di mana jalannya iterasi tersebut disusun. Dalam *iteration planning*, pengguna diminta memilih menetapkan nilai prioritas pada *user story* yang sudah terkumpul. *User stories* kemudian diterjemahkan menjadi *tasks*. *Tasks* adalah gambaran akan hal-hal yang perlu dilakukan oleh perlu dilaksanakan oleh para *programmer*. *Tasks* bersifat teknis. Setiap *task* harus dilaksanakan oleh *programmer* yang memberikan perkiraan penyelesaian dari *user story* yang terkait.

Perancangan

Tahap perancangan dari sebuah proses *extreme programming* memiliki suatu ciri khas yaitu *spike solutions*. Tujuan *spike solution* adalah membuka wawasan terhadap penyelesaian-penyelesaian dari permasalahan-permasalahan teknis dan desain yang tergolong sulit, yang diperkirakan akan timbul selama proses pengembangan. *Spike solutions* berupa program-program kecil yang dibuat dalam rangka menyelesaikan masalah yang dimaksud, tanpa

mempertimbangkan masalah-masalah yang lain. Manfaat *spike solution* terhadap proyek pengembangan adalah agar para *programmer* dapat menghasilkan perkiraan penyelesaian *user stories* yang lebih handal.

Coding

Menurut Beck, penulisan *source code* adalah bagian paling penting dari pengembangan perangkat lunak dengan metodologi *extreme programming*. *Source code* adalah artifak dan media komunikasi utama dalam pengembangan dengan metodologi XP. *Source code* menggambarkan secara persis bagaimana wujud sebuah perangkat lunak.

Wells berpendapat bahwa ciri khas *extreme programming* dalam aspek *coding* adalah *pair programming*. *Pair programming* dilakukan dengan oleh dua orang *programmer* berbagi satu komputer; keduanya secara bergiliran memakai monitor dan *keyboard*. Pasangan *pair programmer* adalah dua individu yang setara. Hubungan antara pasangan *pair programmer* tidak seperti hubungan guru dan murid.

Pengujian

Terdapat dua pengujian yang wajib dilakukan apabila mengembangkan perangkat lunak dengan metode *extreme programming*. Pengujian-pengujian tersebut adalah *acceptance testing* dan *unit testing*.

Wells memaparkan bahwa *acceptance testing* adalah pengujian yang bertujuan mengetahui sejauh mana perangkat lunak yang sedang dalam tahap pengembangan memenuhi permintaan pengguna. Jalannya *acceptance test* direncanakan pada tahap *iteration planning* bersama dengan pengguna. *Acceptance test* dijalankan untuk tiap-tiap *user story*, dan sebuah *user story* dapat memiliki satu atau lebih skenario pengujian. *Acceptance testing* bersifat *black box*, dan pengguna yang menentukan apakah sebuah *acceptance test* sukses atau gagal. Pengembangan sebuah *user story* tidak akan dianggap selesai sebelum lulus *acceptance test* yang bersangkutan.

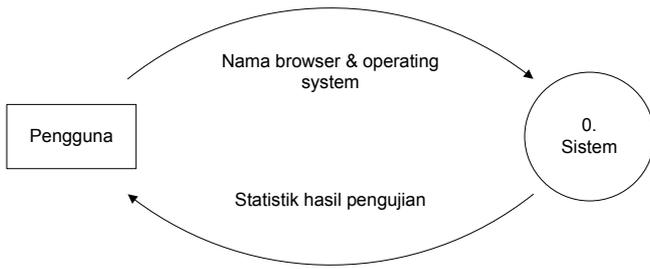
Menurut Wells, *unit testing* adalah pengujian yang bertujuan untuk mendeteksi *bug* dalam pemrograman. *Unit testing* adalah pengujian yang terotomatisasi yang disusun sebelum tahap *coding*. *Unit test* yang telah disusun diharapkan dapat menelusuri tiap-tiap tahap dari jalannya sebuah perangkat lunak, dan dengan demikian setiap tahap tersebut tidak memicu *bug*.

III. METODOLOGI PENELITIAN

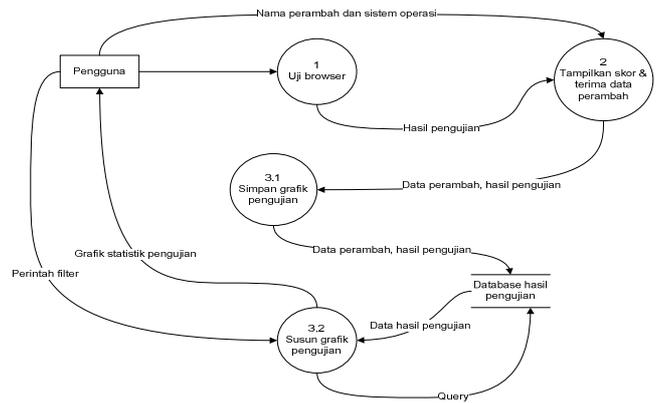
A. Perencanaan

Perencanaan bertujuan untuk menentukan aspek-aspek perangkat lunak dengan cara mengumpulkan *user stories*. Berikut adalah *user stories* yang sudah terkumpul yaitu pengguna dapat melakukan pengukuran kecepatan *rendering*. Pengguna dapat melihat kemampuan perambah dalam melakukan pemutar video HTML5. Pengguna dapat mengukur kinerja JavaScript perambah. Pengguna dapat melihat hasil pengukuran. Pengguna dapat memfilter hasil pengukuran berdasarkan sistem *operasi*.

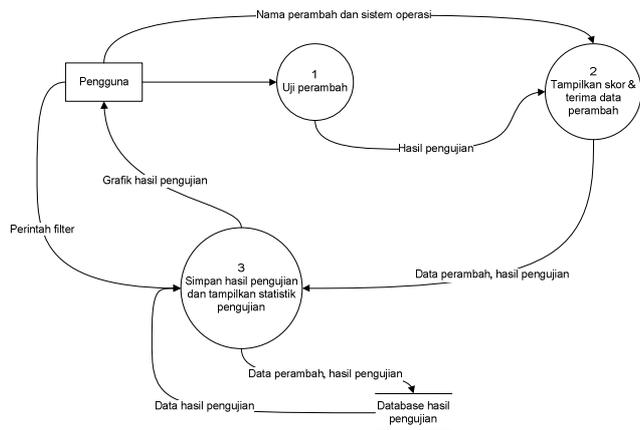
B. Perancangan



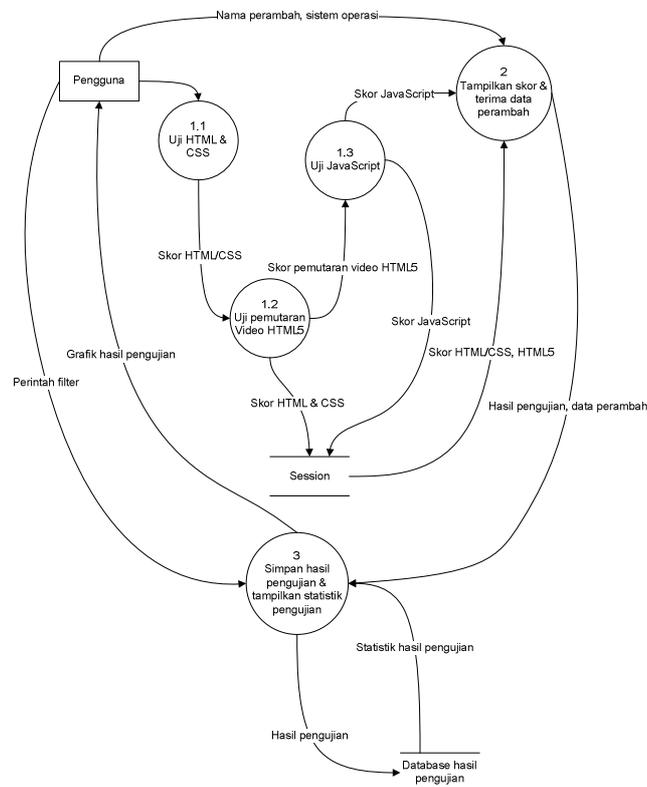
Gambar 4. Dataflow Diagram level 0



Gambar 7. Dataflow diagram level 2. Proses 3



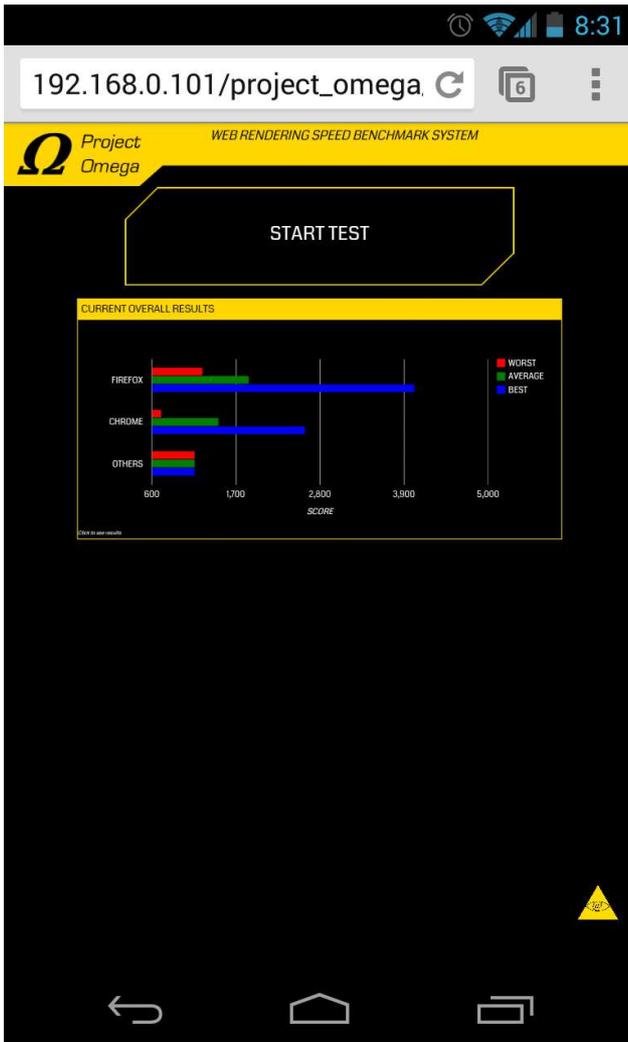
Gambar 5. Dataflow diagram level 1.



Gambar 6. Dataflow diagram level 2 proses 1

TABEL I. BASIS DATA HASIL PENGUJIAN

Nama field	Tipe	Properties	Keterangan
id	int(11)	not null, auto increment, primary key	Pengenal utama dari masing-masing pengujian
date	date	not null	Tanggal pengujian dilakukan
browser	varchar(50)	not null	Nama perambah yang diuji
os	varchar(50)	not null	Platform di mana perambah yang diuji dijalankan
score	float	not null	Hasil pengujian secara keseluruhan.
score_html	float	default null	Hasil pengujian kecepatan render
score_mp4	Float	default null	Hasil pengujian kemampuan pemutaran MP4.
score_webm	Float	default null	Hasil pengujian kemampuan pemutaran WebM
score_ogg	Float	default null	Hasil pengujian kemampuan pemutaran OGG
score_js	Float	default null	Hasil pengujian kinerja JavaScript



Gambar 8. Tampilan awal Project Omega



Gambar 9. Tampilan pengujian kecepatan render HTML

Setelah mengumpulkan *user stories*, pengembang menyusun rancangan aplikasi dengan menggunakan *Dataflow diagram*. Dalam rangka memungkinkan aplikasi menampilkan statistik pengukuran, maka aplikasi perlu menyimpan hasil-hasil pengukuran dalam sebuah basis data (Gambar 6). Selain itu, dalam rangka memungkinkan pemfilteran hasil pengukuran berdasarkan sistem operasi, pengembang merancang aplikasi seperti pada Gambar 7

C. Coding

Coding adalah proses yang paling penting dalam pengembangan perangkat lunak dengan metode *extreme programming*. *Coding* menghasilkan artefak inti dari proses pengembangan yakni *source code*, seperti yang diutarakan oleh Beck.

Proses *coding* dalam pembuatan aplikasi ini dilaksanakan secara manual, tanpa menggunakan perangkat *code generator* yang disediakan oleh berbagai perangkat pengembangan *Agile* yang saat ini tersedia, tidak juga menggunakan perangkat perancangan aplikasi *web*. Tiap-tiap perubahan yang dilakukan ke dalam aplikasi langsung dijalankan dan diperiksa pada saat *runtime*.

IV.

V.

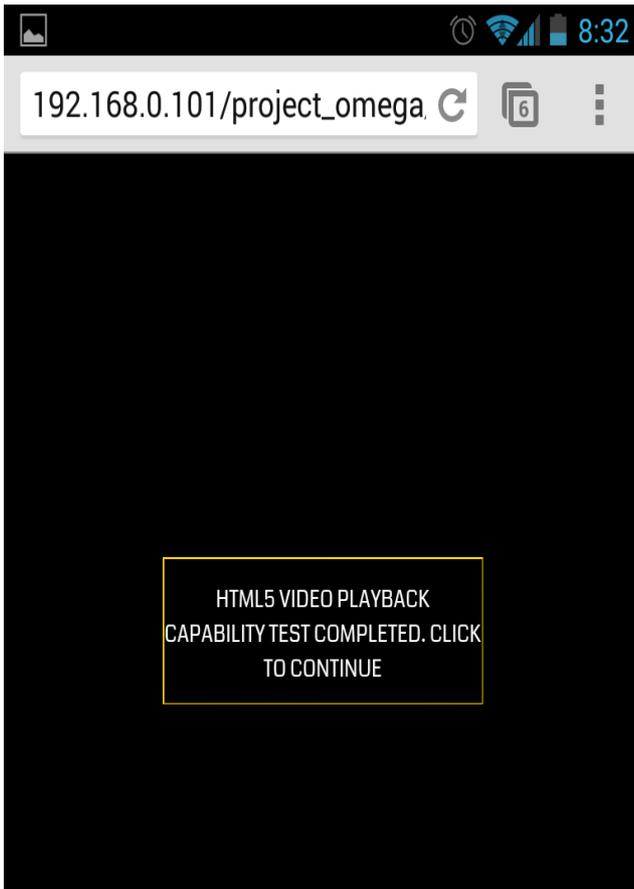
VI. HASIL DAN PEMBAHASAN

A. Alur Program

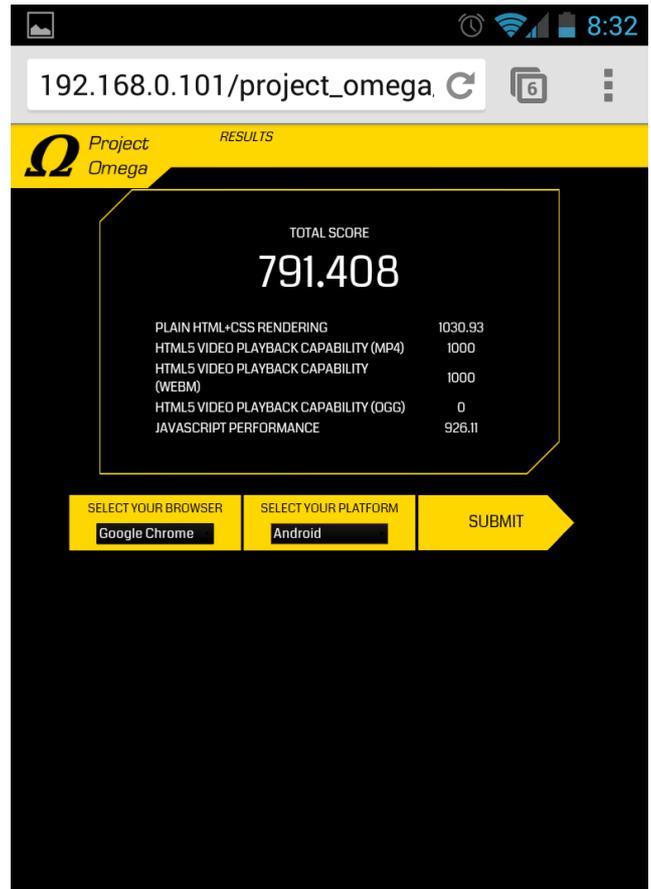
Aplikasi *benchmarking* ini diberi nama *Project Omega*. Nama tersebut diambil dari fakta bahwa aplikasi ini merupakan tugas akhir penulis di Universitas Sam Ratulangi. Pengembangan *Project Omega* dimulai pada tanggal 3 Maret 2013. *Working prototype* selesai dibangun pada tanggal 29 Maret 2013. Versi yang dipakai dalam pembahasan berikut selesai dibangun pada tanggal 13 Oktober 2013.

Project Omega diakses dengan mengunjungi URL di mana aplikasi tersebut di-host, dengan menggunakan perambah *web* yang akan diukur kinerjanya. Tampilan awal *Project Omega* adalah seperti ditunjukkan pada Gambar 8.

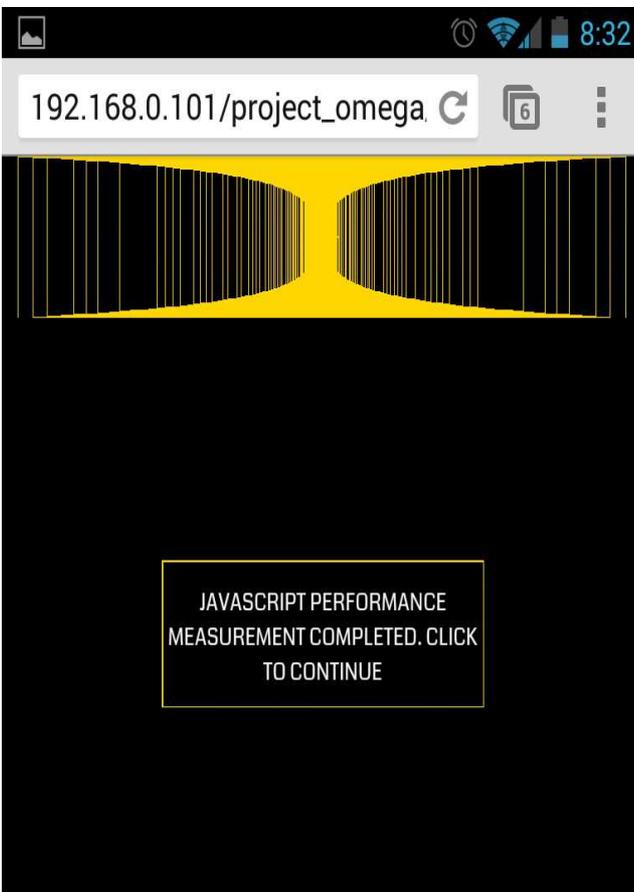
Pengujian dimulai dengan mengklik tombol *Start Test*. Pengguna akan diarahkan langsung ke tahap awal pengujian.



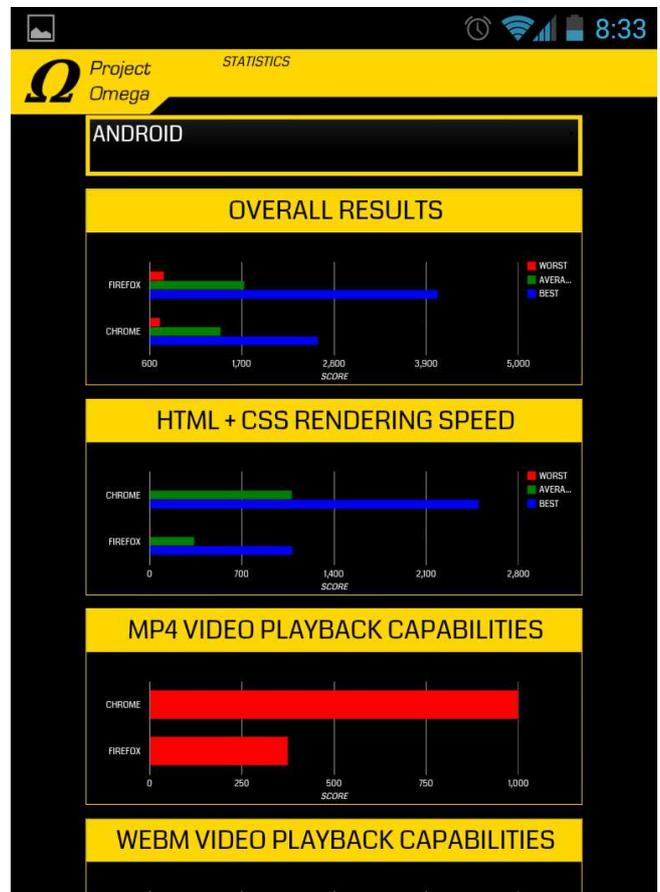
Gambar 10. Tampilan pengujian kemampuan pemutaran video



Gambar 12. Tampilan hasil pengujian



Gambar 11. Tampilan pengujian kinerja JavaScript



Gambar 13. Tampilan statistik pengujian

TABEL II. HASIL BENCHMARKING PADA SISTEM OPERASI ANDROID DENGAN PROJECT OMEGA

Perambah	Skor					
	HTML	Video HTML5			JavaScript	Skor Akhir
		MP4	WEBM	OGG		
Dolphin	6144.780	500.000	500.000	0.000	525.598	1534.077
Firefox	167.106	1000.000	1000.000	1000.000	1335.740	900.568
Chrome	1125.777	1000.000	1000.000	0.000	850.356	795.226
Opera	908.272	1000.000	1000.000	0.000	775.768	736.809

Tahap pengujian yang pertama adalah pengujian kemampuan perambah dalam me-render halaman web. Halaman ini terdiri atas 100 div yang bersifat scrollable. Masing-masing memiliki sebuah nested div dan sebuah paragraf teks lorem ipsum seperti yang ditunjukkan pada Gambar 9.

Sebelum pengujian mulai dilaksanakan, aplikasi akan membersihkan session dari data-data pengujian yang mungkin telah dilakukan sebelumnya. Dengan menggunakan Web Performance API, waktu di mana halaman selesai diambil dari server dan waktu di mana halaman berhasil ditampilkan di layar diukur. Selisih kedua nilai tersebut kemudian diubah menjadi skor dengan rumus, di mana adalah waktu di mana pengambilan halaman dari server selesai dan adalah waktu di mana halaman selesai ditampilkan ke layar. Hal ini dimaksudkan untuk memberikan sebuah skor yang berbanding lurus dengan kinerja (kinerja yang lebih cepat akan menghasilkan skor dengan angka yang lebih besar).

Dengan mengklik dialog box yang muncul di layar, pengguna akan dibawa ke pengujian selanjutnya. Skor yang sudah didapat dari pengujian HTML dikirimkan ke script pengujian selanjutnya dengan menggunakan HTTP POST request.

Pengujian berikutnya adalah pengujian kemampuan pemutaran video HTML5 (Gambar 10). Format-format yang diuji oleh Project Omega yaitu OGG, MP4, dan WebM. Sebelum pengujian dimulai, script pengujian HTML5 akan menerima skor yang dikirim oleh script pengujian sebelumnya dan menyimpannya dalam session. Kemampuan pemutaran tiap-tiap format video dilakukan dengan method canPlayType('formatName') yang ada pada tiap DOM object yang berwujud video. Method canPlayType() dapat menghasilkan satu dari tiga value berikut:

Sebuah String Kosong

Yang berarti video dengan format tersebut kemungkinan besar tidak dapat diputar. Apabila format yang diuji menghasilkan nilai ini, maka skornya ditetapkan sebagai 0.

“Maybe”

Yang berarti video dengan format tersebut mungkin bisa diputar. Apabila format yang diuji menghasilkan nilai ini, maka skornya ditetapkan sebagai 500.

Tabel III. Hasil benchmarking perambah pada sistem operasi windows versi desktop dengan Project Omega

Perambah	Skor					
	HTML	Video HTML5			JavaScript	Skor Akhir
		MP4	WEBM	OGG		
Opera	5509.804	0.000	1000.000	1000.000	4000.800	2302.121
Firefox	217.713	400.000	1000.000	1000.000	8466.133	2212.769
Chrome	625.200	1000.000	1000.000	1000.000	4167.500	1558.540
Internet explorer	3058.188	1000.000	0.000	0.000	2361.583	1283.954

“Probably”

Yang berarti video dengan format tersebut hampir dapat dipastikan bisa diputar. Apabila format yang diuji menghasilkan nilai ini, maka skornya ditetapkan sebagai 1000.

Dengan mengklik dialog box yang tampil, user diantar ke pengujian selanjutnya. Skor yang dihasilkan dikirimkan ke script berikutnya melalui HTTP POST request.

Pengujian selanjutnya adalah pengujian kinerja JavaScript. Sebelum pengujian dimulai, script pengujian JavaScript akan menerima skor yang dikirim oleh script pengujian video HTML5 dan menyimpannya ke dalam session.

Pengujian JavaScript dilakukan dengan cara mengambil waktu pada saat pengujian dijalankan, menjalankan sebuah kode yang dijadikan bobot pengujian, dan mengambil waktu di mana kode selesai dijalankan.

Kode yang dijadikan bobot pengujian dalam Project Omega bekerja dengan cara menciptakan dan menampilkan 100 div secara berurutan, di mana div tersebut terletak secara nested satu dalam yang lain. Tiap div berukuran 95% dari div yang berada satu tingkat di luarnya (Gambar 11).

Selisih waktu eksekusi dimulai dan eksekusi selesai kemudian diubah menjadi skor dengan rumus, di mana adalah waktu di mana eksekusi kode dimulai dan adalah waktu di mana eksekusi kode selesai.

Dengan mengklik dialog box yang tampil, user diantar ke hasil pengujian (Gambar 12). Skor yang dihasilkan dikirimkan ke script berikutnya melalui HTTP POST request.

Hasil akhir pengujian adalah rata-rata dari skor-skor dari pengujian-pengujian yang telah dilakukan. Pengguna diberi kesempatan untuk menyertakan nilai-nilai tersebut ke dalam statistik hasil pengukuran, beserta sistem operasi dan perangkat perambah web yang dipakainya (Gambar 12). Pada tahap ini, skor-skor pengujian diambil dari session dan diletakkan dalam beberapa input fields. Dengan mengklik tombol submit, data-data tersebut dikirimkan ke script berikutnya beserta perambah dan sistem operasi yang dipilih pengguna melalui HTTP POST.

B. Hasil Benchmarking

Project Omega

Halaman pada Gambar 13 menampilkan grafik berdasarkan hasil-hasil pengujian yang sudah dilakukan. Masing-masing grafik merupakan output dari sebuah script yang mengambil rata-rata nilai dari aspek yang bersangkutan dari basis data, dan me-render data tersebut ke dalam objek-objek HTML yang membentuk berbagai chart dengan menggunakan Google Visualization API. Grafik yang ditampilkan dalam halaman ini dapat difilter berdasarkan sistem operasi. Tiap kali pengaturan filter tersebut diubah, halaman ini akan dimuat ulang dan segala script yang bertugas me-render grafik-grafik akan diberikan parameter berdasarkan sistem operasi yang dipilih.

Dromaeo

TABEL VII. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI ANDROID dengan DROMAEO

Perambah	Skor			
	Percobaan I	Percobaan II	Percobaan III	Rata-rata
Chrome	119.20	115.13	120.01	118.113
Firefox	87.77	86.61	86.42	86.933
Dolphin	122.25	122.10	121.00	121.783
Opera	106.04	107.01	106.10	106.383

TABEL VIII. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI BLACKBERRY DENGAN DROMAEO

Perambah	Skor			
	Percobaan I	Percobaan II	Percobaan III	Rata-rata
Default Browser	83.38	83.45	81.07	82.63

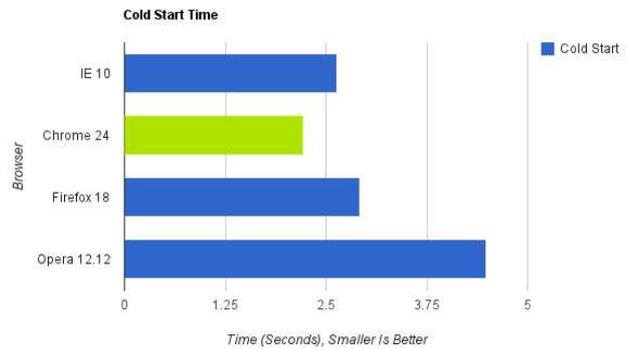
TABEL IX. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI WINDOWS VERSI DESKTOP DENGAN DROMAEO

Perambah	Skor			
	Percobaan I	Percobaan II	Percobaan III	Rata-rata
Opera	536.67	535.11	536.90	536.23
Firefox	380.14	376.22	390.11	382.16
Internet explorer	196.81	195.55	197.11	196.49
Chrome	0.00	0.00	0.00	0.00

TABEL X. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI LINUX DENGAN DROMAEO

Perambah	Skor			
	Percobaan I	Percobaan II	Percobaan III	Rata-rata
Chrome	603.44	610.12	601.23	604.93

Firefox	374.39	362.21	375.24	370.61
Lain-lain	370.35	371.33	312.00	351.23



Gambar 14. Hasil pengujian kecepatan start-up dengan metode Lifehacker

Tabel II sampai dengan Tabel VI merupakan hasil-hasil pengujian dengan menggunakan Project Omega. Tabel VII sampai dengan tabel XI merupakan hasil pengujian dengan menggunakan Dromaeo. Tabel XII sampai dengan tabel XVI merupakan hasil pengujian dengan menggunakan Speed-Battle.

Speed-Battle

TABEL XII. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI ANDROID DENGAN SPEED-BATTLE

Perambah	Skor			
	Calculate	Store	Render	Overall
Chrome	16.25	89.04	4.34	109.63
Firefox	12.05	70.62	4.20	86.87
Dolphin	13.65	95.26	2.84	111.75
Opera	12.72	8.57	3.24	24.93

TABEL XIII. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI BLACKBERRY DENGAN SPEED-BATTLE

Perambah	Skor			
	Calculate	Store	Render	Overall
Default Browser	15.29	58.61	2.13	76.00

TABEL XIV. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI WINDOWS VERSI DESKTOP DENGAN SPEED-BATTLE

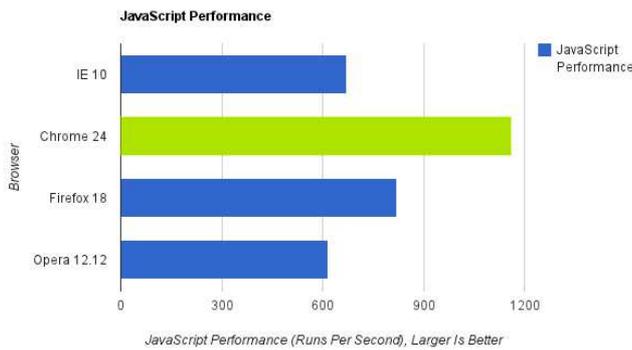
Perambah	Skor			
	Calculate	Store	Render	Overall
Chrome	65.02	224.44	4.92	294.38
Opera	75.16	264.26	4.38	343.80
Firefox	30.12	368.18	56.89	455.19
Internet explorer	40.96	78.77	3.51	123.24

TABEL XV. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI LINUX DENGAN SPEED-BATTLE

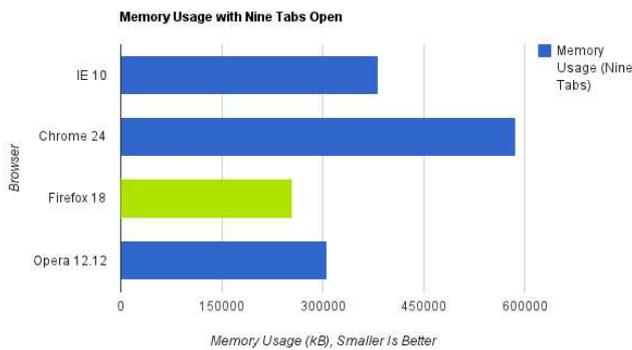
Perambah	Skor			
	Calculate	Store	Render	Overall
Firefox	58.27	381.02	23.01	472.30
Chrome	58.51	234.06	3.98	296.55
Lain-lain	58.02	210.10	3.11	271.23

TABEL XVI. HASIL BENCHMARKING PERAMBAH PADA SISTEM OPERASI MAC OS X DENGAN SPEED-BATTLE

Perambah	Skor			
	Calculate	Store	Render	Overall
Firefox	97.44	570.23	89.88	757.55
Opera	89.82	520.39	33.03	643.24
Chrome	30.57	165.12	11.01	206.70



Gambar 16. Hasil pengujian kinerja JavaScript dengan metode Lifehacker



Gambar 17. Hasil pengujian penggunaan memory dengan metode Lifehacker

Lifehacker

Chrome merupakan perambah dengan kinerja JavaScript tercepat (Gambar 16). Firefox merupakan perambah dengan penggunaan *memory* terbaik (Gambar 17). Chrome merupakan perambah dengan durasi *start-up* paling cepat (Gambar 14).

VII. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pembahasan di atas, penulis mengambil kesimpulan bahwa Dolphin adalah perambah *web* dengan kinerja secara rata-rata paling tinggi di antara perambah-perambah yang dijadikan objek penelitian, berdasarkan hasil pengujian dengan Project Omega. Sementara itu, kinerja Dolphin dalam mengeksekusi JavaScript adalah yang paling rendah di antara perambah-perambah yang dijadikan objek penelitian berdasarkan hasil pengujian dengan Project Omega. Untuk aspek video HTML5, MP4 adalah format

video yang memiliki dukungan perambah paling banyak. Firefox adalah perambah yang paling cocok untuk memutar konten video HTML5.

B. Saran

Penulis menyarankan agar aplikasi Project Omega dapat dikembangkan lebih lanjut agar dapat mencakup lebih banyak aspek dari sebuah perambah *web* serta sistem penilaiannya diharapkan agar dibuat lebih ilmiah. Penulis menyarankan agar pengembang teknologi *web* untuk lebih menyelaraskan teknologi mereka dengan standar yang ada. Untuk pengembang aplikasi *web* disarankan untuk tidak menggunakan OGG apabila hendak menanamkan video HTML5 dalam halaman *web*. 5. Pengguna Android disarankan untuk menggunakan Firefox untuk mengunjungi halaman *web* yang memiliki banyak konten JavaScript atau video HTML5.

DAFTAR PUSTAKA

- [1] B. Kent. *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley, 2000.
- [2] D. Gosselin. *JavaScript, Fifth Edition*. Course Technology, 2011.
- [3] E. Tittel, J. Noble. *HTML, XHTML & CSS For Dummies®, 7th Edition*. Indianapolis: Wiley Publishing, 2011.
- [4] Goodman, Morrison, Novitski, Rayl. *JavaScript Bible, 7th Edition*. Indianapolis: Wiley Publishing. Internet Engineering Task Force. "RFC5334 — Ogg Media Types."
- [5] Internet: <http://tools.ietf.org/pdf/rfc5334.pdf>, [10 Agustus 2013].
- [6] J. D. Wells. "Extreme Programming: A Gentle Introduction." Internet: <http://www.extremeprogramming.org/>, [5 Juli 2013].
- [7] K. Purdy. "Speed Testing the Latest Web Browsers." Internet: <http://lifehacker.com/396048/speed-testing-the-latest-web-browsers>, 29 Juni 2010 [20 Maret 2013].
- [8] Library of Congress. "MPEG-4 File Format, Version 2." Internet: <http://www.digitalpreservation.gov/formats/fdd/fdd000155.shtml>, [10 Agustus 2013].
- [9] MP4 Registration Authority. "Registered types — Codecs, MPEG-4 Registration authority." Internet: <http://www.mp4ra.org/codecs.html>, [10 Agustus 2013].
- [10] W. Gordon. "How We Test Web Browsers." Internet: <http://lifehacker.com/5575535/how-we-test-web-browsers>, 29 Juni 2010 [20 Maret 2013].
- [11] The WebM Project. "About WebM." Internet: <http://www.webmproject.org/about>, [10 Agustus 2013].