

# Rancang Bangun Aplikasi Implementasi Ukiran Bali pada Objek Bangun Datar

I Kadek Erik Priyanto, I Ketut Gede Darma Putra, Ni Made Ika Marini Mandenni

Program Studi Teknologi Informasi Universitas Udayana

Bukit Jimbaran, Bali, Indonesia, telp. (0361) 701806

e-mail: [erik.3121x@gmail.com](mailto:erik.3121x@gmail.com), [ikgdarmaputra@unud.ac.id](mailto:ikgdarmaputra@unud.ac.id), [ika\\_made@yahoo.com](mailto:ika_made@yahoo.com)

## Abstrak

Seni ragam ukiran memiliki peran yang penting digunakan sebagai hiasan bangunan, tempat suci di Bali dan benda fungsional, namun upaya untuk pelestarian kebudayaan masih cenderung kurang. Tradisi, ragam hias, dan teknologi merupakan perpaduan yang sangat tepat untuk membantu melestarikan ragam seni ukiran, sehingga dibangun Aplikasi Implementasi Ukiran Bali pada Objek Bangun Datar. Aplikasi ini menggunakan canvas yang merupakan elemen dari HTML yang memungkinkan dilakukan manipulasi daerah bitmap secara langsung dengan menggunakan JavaScript. Library Fabric.js digunakan untuk mempermudah manipulasi daerah bitmap pada elemen kanvas HTML5. Fabric.js memungkinkan dilakukannya create objek, dan filter citra. Remove white merupakan salah satu filter yang dapat dilakukan. Remove white digunakan untuk menghilangkan warna putih, dimana nilai thresholding digunakan sebagai nilai batas pada aplikasi untuk menghilangkan warna putih pada gambar sehingga memaksimalkan proses registrasi ukiran pada aplikasi. Aplikasi yang dihasilkan dapat dijalankan pada browser dan memberikan hasil ilustrasi implementasi Ukiran yang dapat disimpan dalam format file PNG.

**Kata kunci:** Ukiran Bali, Thresholding, Fabric.js, Aplikasi

## Abstract

Carving art has an important role which is used as decoration of buildings, holy places in Bali and functional objects, but the efforts for cultural preservation still tend to be less. Tradition, decoration, and technology are the perfect combination to help preserve the variety of carving art, so that the application of Bali Carving Implementation in Two Dimensional Object is built. This application uses canvas which is an element of HTML that allows to manipulate bitmap area directly by using JavaScript. Library Fabric.js is used to simplify the manipulation of bitmaps on HTML5 canvas elements. Fabric.js allows the creation of objects, and image filters. Remove white is one filter that can be done. Remove white is used to remove white color, where the threshold value is used as the limit value of the application to remove the white color in the image so as to maximize the carving registration process on the application. The resulting application can be run on the browser and gives an illustration result of the carving implementation that can be saved in PNG file format.

**Keywords:** Carving Bali, thresholding, Fabric.js, Application

## 1. Pendahuluan

Seni adalah karya manusia yang mengkomunikasikan pengalaman-pengalaman batin, pengalaman batin tersebut disajikan secara indah atau menarik sehingga merangsang timbulnya pengalaman batin bagi manusia yang menghayati [1]. Seni ragam hias atau ukiran memiliki peranan yang penting digunakan untuk hiasan bangunan tempat suci di Bali dan pada rumah serta benda fungsional atau benda hias [2]. Ragam hias Bali mengambil motif flora dan fauna sehingga sangat mudah diaplikasikan, namun upaya untuk pelestarian kebudayaan masih cenderung kurang [3] sehingga diperlukan tindakan lebih lanjut untuk melestarikannya. Tradisi, ragam hias, dan

teknologi merupakan perpaduan yang sangat tepat untuk meningkatkan minat masyarakat ikut serta dalam pengembangan pendidikan dan pelestarian ragam hias Ukiran Bali.

Pelestarian ragam hias ukiran dapat diimplementasikan berbasis website dengan memanfaatkan kanvas yang merupakan elemen dari HTML5 [4] sehingga dapat melakukan manipulasi *bitmap* dengan menggunakan Javascript dan Fabric.JS. Fabric.JS mempermudah melakukan berbagai tugas yang berhubungan dengan elemen *canvas* [3], Fabric.JS memiliki fitur *remove white* berfungsi untuk menghilangkan warna putih yang terdapat pada objek, batasan warna putih yang dihilangkan tergantung dari nilai *thresholding* yang diinputkan. *Thresholding* merupakan proses untuk mengubah citra keabuan menjadi citra biner atau hitam putih sehingga dapat terlihat dengan jelas antara *background* dan *foreground* [5] untuk mempermudah untuk memisahkan satu objek dengan yang lainnya.

Aplikasi Implementasi Ukiran Bali pada Objek Bangun Datar dibuat menjadi salah satu solusi pelestarian ukiran. Aplikasi bertujuan untuk memberikan ilustrasi implementasi ukiran ketika sudah diterapkan pada objek datar yang diinginkan sehingga meningkatkan minat masyarakat ikut serta dalam pengembangan pendidikan dan pelestarian ragam hias Ukiran Bali sebagai motif hiasan yang menghiasi dinding-dinding atau benda fungsional.

## 2. Metodologi Penelitian

### 2.1. Gambaran Umum Aplikasi

Rancang Bangun Aplikasi Implementasi Ukiran Bali pada Objek Bangun Datar dengan Metode *Thresholding* terdiri dari dua entitas yaitu entitas pengguna dan entitas aplikasi. Pengguna terlebih dahulu meminta akses aplikasi yang direspon dengan memberikan tampilan halaman aplikasi. Gambaran umum aplikasi ditunjukkan pada Gambar 1.



Gambar 2. Gambaran Umum Aplikasi

Pengguna diminta untuk memilih ingin melakukan registrasi atau melakukan implementasi ukiran saat akses aplikasi, setelah pengguna aplikasi memilih registrasi atau implementasi ukiran, pengguna diminta untuk mengunggah citra yang ingin digunakan. Citra yang diunggah kemudian diproses dengan menggunakan fitur-fitur yang tersedia pada aplikasi. Fitur yang tersedia pada Halaman Registrasi yaitu fitur *grayscale*, *Invert*, dan *remove white*. Fitur yang tersedia pada Halaman Implementasi yaitu *opacity*, *Fill*, *Rotate*, *scale*, dan *tint*. Hasil dari pemrosesan ditampilkan pada halaman aplikasi.

## 3. Kajian Pustaka

### 3.1. State of The Art

*Thresholding* berhasil digunakan untuk menyelesaikan uji coba analisis pembuluh darah retina. Analisis pembuluh darah bertujuan untuk diagnosis penyakit yang dilakukan oleh M. Riza Kurnia, pada analisa segmentasi pembuluh darah retina ini terdiri dari dua tahap, yaitu *preprocessing* dan segmentasi. Tahap *preprocessing* citra menggunakan beberapa langkah yaitu mencari *regional minimum*, *enhancement* citra, dan *filtering* citra. Tahap segmentasi citra terdiri dari ekstraksi citra, pencarian tekstur citra, proses binerisasi gambar, operasi pembersihan,

penghilangan *Field of View* (FOV), *skeletoning* citra, *pruning* citra dan dilasi. Metode ini telah diujikan pada *database* yang terdiri dari 40 foto retina [5].

Penelitian lainnya dilakukan oleh Slamet Imam Syafi'i dimana *Thresholding* digunakan untuk pemisahan cabang-cabang pembuluh darah dengan *background* sehingga meningkatkan akurasi analisa penyakit yang diderita. *Thresholding* di sini telah berhasil memisahkan antara objek yang diinginkan dengan *background* atau citra yang tidak diinginkan atau yang dihilangkan. Penelitian lain terkait penerapan *thresholding* dilakukan oleh Imam Syafi'i, Slamet, Tri Wahyuningrum, Rima, Muntasa yang berjudul segmentasi obyek pada citra digital menggunakan Metode Otsu. Penelitian bertujuan untuk memisahkan objek antara *foreground* dan *background* pada citra digital dengan menggunakan Metode *Otsu Thresholding*. Proses pemisahan menggunakan lima proses, yaitu *input* citra RGB. Tahap kedua dilakukan konversi dari citra RGB menjadi citra *grayscale*. Tahap ketiga proses konversi ke citra biner dengan nilai ambang yang ditentukan secara otomatis menggunakan Metode Otsu *Thresholding*. Tahap keempat dilakukan proses *Invert*, *Noise Removal*, dan *Morphology*. Tahap terakhir adalah perhitungan [6].

### 3.2. Citra

Citra merupakan istilah lain dari gambar, suatu citra diperoleh dari penangkapan sinar yang dipantulkan oleh objek. Penangkapan sinar dilakukan dengan menggunakan alat bantu seperti kamera dan menghasilkan *output* bersifat optik atau foto, analog berupa video atau digital yang dapat langsung disimpan pada pita magnetik [7]. Citra merupakan istilah lain untuk gambar sebagai salah satu komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi. Citra (*image*) adalah gambar pada bidang 2D (2 dimensi). Sumber cahaya menerangi objek, objek memantulkan kembali sebagai dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*) sehingga bayangan objek yang disebut citra tersebut terekam. Citra merupakan suatu larik (*array*) yang memiliki nilai-nilai nyata dan rumit yang dipresentasikan dalam bit tertentu [8]. Pengolahan citra dapat dilakukan dengan teknik *image processing*, sehingga memungkinkan dilakukan proses *grayscale*, *thresholding*, dan *filter* [9].

### 3.3. Thresholding

*Thresholding* merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama. *Thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*. *Thresholding* dapat diklarifikasikan menjadi dua yaitu *thresholding local* dan *thresholding global*. *Thresholding local* menggunakan lebih dari 1 nilai *Thresholding* dimana suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan *thresholding local* pada setiap blok dengan nilai ambang T yang berbeda-beda. *Thresholding local* digunakan untuk mengatasi variasi kontras dan pencahayaan *foreground* dan *background* yang sulit untuk diklarifikasi karena proses *thresholding* berlangsung pada setiap blok piksel dari citra dengan menggunakan nilai ambang T yang berbeda untuk setiap bloknnya. *Thresholding Global* digunakan untuk melakukan konversi dalam citra menggunakan satu nilai ambang. *Thresholding global* lebih cocok dipergunakan untuk citra yang memiliki warna tajam (kontras) antara *background* dan *foreground* sehingga piksel lebih mudah untuk dipisahkan [6] [7] [10].

### 3.4. Fabric.js

*Fabric.js* adalah kerangka kerja yang mempermudah melakukan berbagai tugas yang berhubungan dengan elemen *canvas* pada HTML5 [3]. *Fabric.js* adalah *Framework JavaScript* yang dapat bekerja pada elemen *canvas HTML5*. *Fabric.js* memiliki fungsi yang dapat dipergunakan untuk manipulasi bitmap pada elmen *canvas HTML5*. Manipulasi yang dapat dilakukan oleh *Fabric.js* yaitu *Matrix transformation* dan *Filter Image*. *Matrix transformation* dipergunakan untuk memanipulasi ukuran citra, dan *Filter image* memungkinkan untuk melakukan pengolahan citra pada elemen *canvas HTML5*. *Filter* yang dapat dilakukan yaitu *Grayscale* yang

berfungsi untuk mengubah citra berwarna menjadi citra keabuan, *Invert* atau negasi merupakan proses pemetaan nilai piksel suatu citra, warna hitam dijadikan warna putih dan warna putih diubah menjadi warna hitam untuk citra biner, *Remove White* berfungsi untuk menghilangkan warna putih yang ada pada citra dengan menggunakan nilai *thresholding* untuk nilai batas, dan *brightness* yang digunakan untuk mengatur pencahayaan.

### 3.5. Seni Ukir

Seni merupakan hasil karya yang menggambarkan pengalaman batin yang dituangkan kedalam bentuk ornamen-ornamen atau bentuk-bentuk sehingga dapat mengkomunikasikan pengalaman kepada orang lain yang menghayati atau melihat hasil karya seni. Bali sendiri merupakan salah satu tempat yang banyak menggunakan seni ukiran dalam kehidupan sehari-hari, baik diterapkan dalam bangunan maupun alat-alat yang memiliki objek kosong. Bali sendiri memiliki beragam jenis seni ukiran yang tersebar di berbagai daerah di Bali diantaranya yaitu ukiran *Karang Gajah*, *Karang Goak*, *Karang Boma*, *Karang Sai*, dan berbagai *Keketusan*. Ukiran pada umumnya meniru bentuk-bentuk yang ada di alam misalnya *Karang Gajah* meniru bentuk gajah, dan *Karang Daun* meniru bentuk dedaunan [11].

### 3.6. Canvas

*Canvas* merupakan salah satu elemen yang dimiliki oleh *HTML 5*. *Canvas* adalah cara untuk memanipulasi daerah *bitmap* secara langsung pada layar dengan menggunakan *JavaScript* dan secara langsung mengacu untuk memberikan piksel pada *canvas*. *Programmer* dapat mengatur tampilan pada layar sebelum diterjemahkan sehingga piksel yang sesuai ditampilkan, dengan menggunakan *Canvas API (Application Programming Interface)* yang dapat dipanggil dengan menggunakan *Javascript*. *Canvas API* memiliki dasar meliputi konteks 2 Dimensi yang dapat dipergunakan untuk menggambar berbagai bentuk, membuat teks, dan menampilkan gambar langsung ke area yang ditentukan dari jendela *browser*. *Programmer* juga dapat menerapkan warna; rotasi; mengisi gradasi; transparansi; manipulasi piksel; berbagai jenis garis, kurva, kotak, mengisi untuk menambah bentuk, teks, dan gambar yang ditempatkan ke dalam *kanvas*. *Canvas* dapat digunakan untuk membuat grafik 2 Dimensi dan dengan menambahkan fungsi *JavaScript*, *Browser* yang cocok untuk masukan *mouse*, *keyboard*, interval waktu, *event*, *class*, *object*, suara, atau fungsi matematika dapat membuat animasi yang menakjubkan, dan permainan [4].

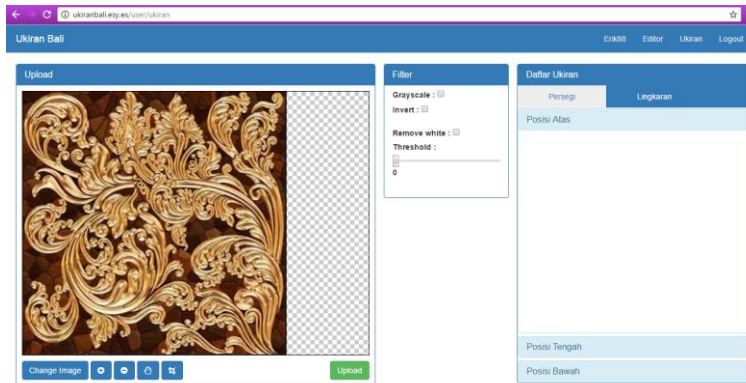
## 4. Hasil dan Pembahasan

Hasil dan Pembahasan membahas tentang hasil dari Halaman Registrasi ukiran dan Halaman Implementasi Ukiran pada aplikasi. Halaman Registrasi Ukiran merupakan halaman yang digunakan untuk mendaftarkan ukiran, dan Halaman Implementasi Ukiran merupakan halaman yang digunakan untuk proses penerapan ukiran pada gambar *background*. Penjelasan lebih detail dari Halaman Registrasi dan Implementasi ukiran dibahas pada subbab selanjutnya.

### 4.1. Halaman Registrasi Ukiran

Registrasi ukiran merupakan proses pendaftaran citra ukiran dalam Aplikasi Implementasi Ukiran Bali pada Objek Bangun Datar. Registrasi ukiran bertujuan untuk menghilangkan *background* dari citra ukiran yang didaftarkan agar mudah untuk dipergunakan dalam proses Implementasi ukiran. Proses untuk mendaftarkan ukiran terdiri dari beberapa langkah.

Langkah awal untuk mendaftarkan ukiran yaitu dengan mengunggah citra ukiran yang didaftarkan pada aplikasi di Halaman Registrasi Ukiran. Citra ukiran yang diunggah ditampilkan pada *Canvas* yang sudah disediakan dalam aplikasi.



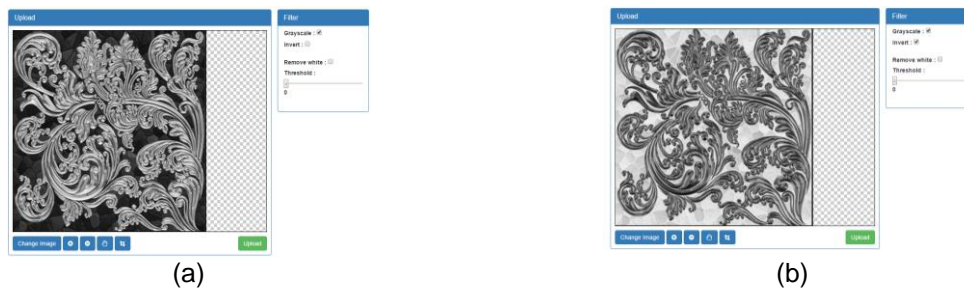
Gambar 3. Tampilan Pemilihan Citra Ukiran

Langkah awal dari proses registrasi yaitu pemilihan citra ukiran yang didaftarkan seperti yang terlihat pada Gambar 3. Citra ukiran yang diunggah selanjutnya ditampilkan pada *Canvas* pada Halaman Registrasi Ukiran. Pengguna aplikasi dapat langsung menyimpan ukiran yang didaftarkan tanpa ditambahkan fitur atau dengan penerapan fitur terlebih dahulu. Fitur yang terdapat pada Halaman Registrasi ukiran yaitu *Grayscale*, *Invert*, dan *Remove White*. Fitur *Grayscale* merupakan Fitur yang dapat digunakan oleh pengguna aplikasi untuk mengubah citra berwarna menjadi citra dengan warna keabuan.



Gambar 4. Proses Filter *Grayscale*

Gambar 4 merupakan penerapan Fitur *Grayscale* pada citra ukiran yang bertujuan untuk mengubah citra berwarna menjadi citra *Grayscale* yang memiliki warna keabuan. Gambar 4(a) merupakan citra awal, dan Gambar 4(b) merupakan citra hasil *Grayscale*. *Grayscale* dapat diterapkan dengan memberikan centang pada *checkbox grayscale* pada kolom filter. Fitur *Invert* (Negasi) merupakan fitur yang dapat digunakan untuk mengubah warna hitam menjadi putih dan warna putih menjadi hitam. Fitur negasi digunakan dengan tujuan untuk mengubah agar citra ukiran diwakili oleh warna hitam.



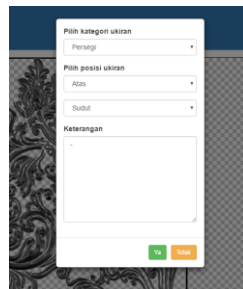
Gambar 5. Tampilan Proses *Invert* (Negasi)

Negasi merupakan proses pemetaan nilai piksel suatu citra, warna hitam dijadikan warna putih dan warna putih diubah menjadi warna hitam untuk citra biner. Nilai maksimum piksel dikurangi dengan nilai piksel yang sedang diproses untuk citra *grayscale* atau citra berwarna. Proses Negasi dapat dilihat pada Gambar 5. Fitur *remove white* merupakan fitur yang dapat digunakan untuk menghilangkan warna putih (*background*) dengan memberikan nilai *thresholding*. Nilai *thresholding* yang dapat diberikan yaitu dari 0 sampai dengan 255.



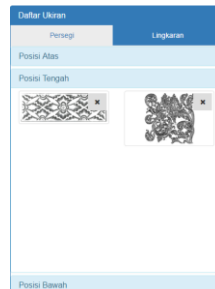
Gambar 6. Tampilan Proses *Remove White*.

Gambar 6 merupakan penerapan fitur *remove white* dan pemberian nilai *thresholding* yang berbeda dimana Gambar 6(a) diberikan nilai *thresholding* yaitu 0 dan citra berwarna putih masih tetap. *Remove White* dengan nilai *threshold* 83 dapat dilihat pada Gambar 6(b) yang menghasilkan citra ukiran tanpa adanya warna putih (*background*). Proses terakhir untuk melakukan registrasi ukiran yaitu *upload* citra agar tersimpan di dalam aplikasi.



Gambar 7. Tampilan Halaman *Upload Ukiran*

*Popup* muncul ketika pengguna aplikasi menekan *Button Upload*. Pengaturan terakhir yang harus dilakukan yaitu mengatur siapa saja yang dapat menggunakan ukiran (*visibilitas*) yang diregistrasikan, terdapat 2 pilihan *visibilitas* yaitu *privat* untuk pengaturan dimana ukiran hanya dapat digunakan sendiri, dan *Public* yaitu pengaturan agar semua pengguna aplikasi dapat menggunakan ukiran yang didaftarkan. Kategori ukiran merupakan pengaturan untuk bentuk ukiran yang didaftarkan, tersedia kategori persegi dan lingkaran. Posisi ukiran merupakan pengaturan pengelompokan ukiran sesuai dengan posisi ukiran pada bangunan seperti yang terlihat pada Gambar 7 Hasil ukiran yang telah didaftarkan dapat dilihat pada daftar ukiran yang ada pada halaman registrasi ukiran.



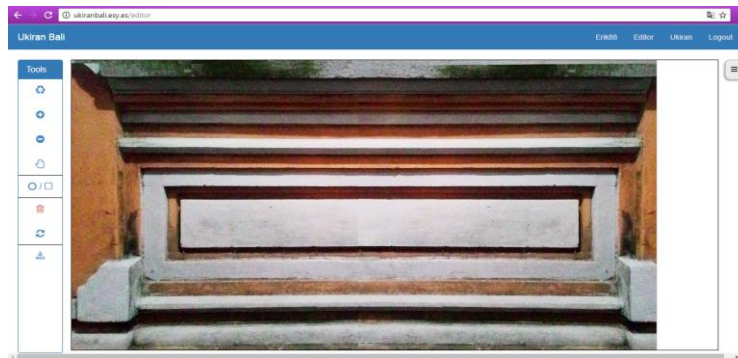


Gambar 8. Tampilan Halaman Daftar Ukiran

Ukiran yang didaftarkan dapat dilihat pada kolom daftar ukiran. Ukiran dikelompokkan berdasarkan bentuk ukiran, dan posisi ukiran sesuai dengan keterangan ukiran pada akhir proses registrasi. Pengguna dapat menghapus ukiran yang telah diregistrasikan dengan menekan tombol silang (X) pada pojok kanan atas ukiran yang ingin dihapus.

#### 4.2. Halaman Implementasi Ukiran

Implementasi ukiran merupakan proses untuk menempelkan ukiran pada *background* menggunakan bentuk objek yang telah disediakan. Bentuk objek yang tersedia yaitu bentuk persegi, dan lingkaran. Ukiran yang direkomendasikan disesuaikan dengan bentuk objek yang dipilih. Halaman Implementasi ukiran merupakan halaman yang dapat digunakan oleh pengguna aplikasi untuk melakukan proses implementasi ukiran.



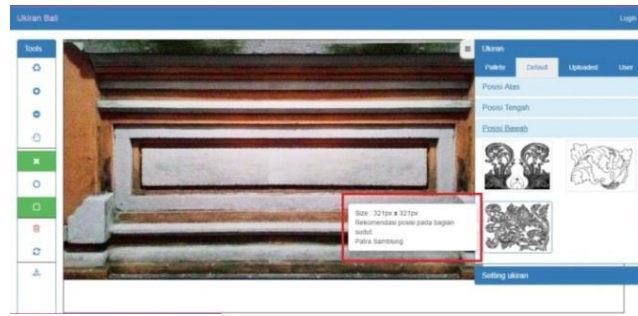
Gambar 9. Tampilan Halaman Implementasi

Gambar 9 merupakan tampilan halaman implementasi ukiran. Pengguna aplikasi diminta untuk memilih gambar yang ingin diterapkan ukiran. Pemilihan bentuk objek dapat dilakukan oleh pengguna aplikasi dengan menekan tombol *selection* yang ada pada kolom *tools*.



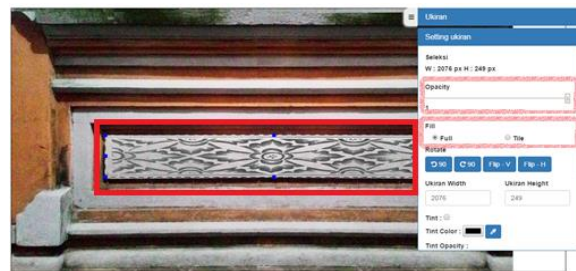
Gambar 10. (a) Proses Pemilihan Bentuk Objek dan (b) Proses Seleksi bentuk persegi panjang

Bentuk objek yang dapat dipilih oleh pengguna aplikasi ada 2 yaitu bentuk persegi dan bentuk lingkaran. Pengguna dapat memilih salah satu bentuk yang diinginkan dan dapat langsung membuat objek diatas gambar yang telah dipilih sebelumnya. Pemilihan bentuk objek dapat dilihat pada Gambar 10(a). Gambar 10(b) merupakan seleksi yang dilakukan diatas gambar yang ingin diterapkan ukiran. Seleksi dilakukan dengan menggunakan bentuk persegi. Proses seleksi dilakukan dengan melakukan *click* dan *drag* sesuai ukuran seleksi yang diinginkan pengguna aplikasi. Rekomendasi ukiran yang diberikan oleh aplikasi yaitu berdasarkan bentuk objek yang dipilih sebelumnya.



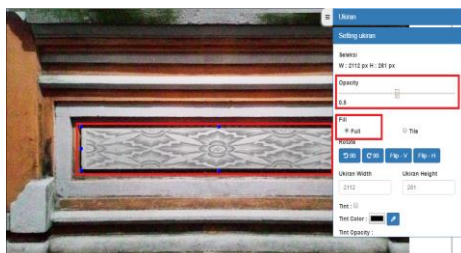
Gambar 11. Tampilan Rekomendasi Ukiran

Gambar 11 merupakan rekomendasi ukiran yang diberikan kepada pengguna aplikasi, dimana bentuk objek yang dipilih oleh pengguna aplikasi berbentuk persegi. Ukiran yang direkomendasikan ditampilkan berdasarkan posisi ukiran. Posisi yang dapat dipilih yaitu posisi atas, tengah, dan bawah. Ukiran yang dipilih diterapkan pada seleksi objek yang dilakukan sebelumnya. Informasi ukiran dapat dilihat dengan menggerakkan *cursor* ke ukiran yang memunculkan informasi dari ukuran ukiran dan rekomendasi letak ukiran. Implementasi ukiran dilakukan pada halaman editor, dimana pengguna aplikasi diberikan rekomendasi ukiran pada kolom ukiran. Ukiran yang dipilih pengguna dapat langsung dilihat pada canvas dan ukuran ukiran disesuaikan dengan ukuran seleksi.

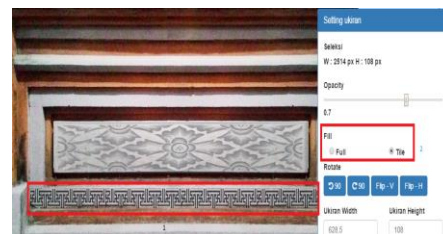


Gambar 12. Proses Penerapan Ukiran

Ukiran diterapkan pada objek dengan ukuran ukiran sesuai dengan ukuran seleksi yang dibuat seperti yang terlihat pada Gambar 12. *Setting* fitur *default* yaitu dengan pengaturan *Opacity* diberikan nilai maksimum yaitu 1, *Fill* diberikan nilai *Full*, dan warna ukiran sesuai dengan warna ukiran ketika diregistrasikan yaitu warna hitam. *Setting* ukiran dapat dilakukan dengan menggunakan fitur yang sudah disediakan, adapun fitur yang disediakan yaitu Fitur *Opacity*, *Fill*, *Rotate*, dan *Tint* pada objek dengan bentuk persegi dan persegi panjang. Fitur *Opacity* merupakan fitur yang dapat dipergunakan untuk mengatur transparansi citra, nilai default *opacity* yaitu 1, Fitur *Fill* merupakan fitur yang dapat dipergunakan untuk mengatur cara ukiran menempati objek. Fitur *Rotate* merupakan fitur yang dapat digunakan untuk memutar ukiran didalam objek, Fitur *Tint* dapat digunakan untuk mengatur warna ukiran. Penerapan Fitur *Opacity* dapat dilihat pada Gambar 13.



(a)



(b)

Gambar 13. Tampilan Fitur *Opacity* (a) *Fill Full* dan (b) *Fill Tile*



Gamabr 13(a) merupakan penerapan fitur *Fill full* dan fitur *opacity*, dimana nilai *opacity* yang diberikan yaitu 0.5 yang menyebabkan citra ukiran menjadi transparan sebagian, dan *fitur fill* dengan pilihan *full*, dimana ukuran ukiran diperbesar dan memenuhi daerah seleksi objek yang dibuat. Gambar 13(b) merupakan penerapan Fitur *Fill* dengan pilihan *Tile* dan *Opacity* diberikan nilai 0.7. Citra ukiran yang diberikan fitur *tile* penerapannya berulang dalam objek yang dibuat. Ukuran seleksi objek yang dibuat yaitu  $W = 2514$ ,  $H = 108$  dan ukuran ukiran yang diterapkan adalah  $W = 628.5$ ,  $H = 108$ , dapat dilihat ukiran yang diterapkan berulang pada objek.



Gambar 14. Tampilan Fitur *Rotate Horizontal* & *Tint*

Penerapan Fitur *Rotate* dan Fitur *Tint* dapat dilihat pada Gambar 14. Gambar 14 memperlihatkan Fitur pada Ukiran *Karang Tapel* Fitur yang diterapkan yaitu *Rotate Horizontal* (Flip-H) untuk melakukan pencermina terhadap sumbu X, dan Fitur *Tint* yang berfungsi untuk memberikan warna pada ukiran dengan pengaturan *Tint Color* yaitu warna yang dipilih untuk diterapkan, dengan *Tint Opacity* yaitu transparansi penerapan warna pada ukiran. Nilai yang diberikan yaitu *Tint Opacity* diberikan nilai 0.7 dan *Tint color* yaitu warna merah, Fitur *Tint* menyebabkan ukiran *Karang Tapel* menjadi berwarna merah.



Gambar 15. Tampilan Hasil Implementasi dengan Fitur *Rotate*, *Opacity*, *Fill*, dan *Tint*.

Gambar 15 merupakan gambar hasil implementasi ukiran dengan menggabungkan fitur *Rotate*, *Opacity*, *Fill*, dan *Tint*. Gambar 15 membuktikan Penggabungan Fitur-fitur yang diterapkan pada ukiran menghasilkan suatu ilustrasi implementasi ukiran pada *background*. Hasil implementasi ukiran dapat disimpan dalam *format file PNG*.

## 5. Kesimpulan

Canvas merupakan elemen HTML5 yang memungkinkan untuk melakukan manipulasi bitmap dengan menggunakan Javascript. Fabric.Js adalah Framework JavaScript yang berkerja pada elemen canvas pada HTML5. Registrasi ukiran pada Aplikasi dapat dioptimalkan dengan menggunakan Fitur *Grayscale*, *Invert*, dan *Remove white*. Implementasi ukiran pada *background* dapat dimaksimalkan dengan menggunakan fitur *opacity*, fitur *Fill*, *size*, *rotate*, dan *tint*. Hasil uji coba yang telah dilakukan menunjukkan Aplikasi Implementasi Ukiran Bali pada Objek Bangun Datar memberikan informasi hasil implementasi ukiran yang dapat diakses dengan menggunakan *browser* pada halaman [www.ukiranbali.esy.es](http://www.ukiranbali.esy.es). Implementasi ukiran dioptimalkan dengan fitur-fitur yang terdapat pada aplikasi. Fitur yang tersedia pada Halaman Implementasi yaitu *opacity*, *Fill*, *Rotate*, *Scale*, dan *Tint* yang dapat dipergunakan untuk memaksimalkan hasil implementasi ukiran. Fitur yang tersedia pada Halaman Registrasi yaitu fitur *Grayscale*, *Invert*, dan *Remove White* yang

dapat dipergunakan untuk memaksimalkan proses registrasi ukiran. Hasil ilustrasi implementasi ukiran pada aplikasi dapat disimpan pada komputer dalam format *file* PNG.

#### Daftar Pustaka

- [1] G. Hartanti and A. Nediari, "Pendokumentasian Aplikasi Ragam Hias Budaya Bali, Sebagai Upaya Konservasi Budaya Bangsa Khususnya Pada Perancangan Interior Hasil Dan Pembahasan Provinsi Bali," *E-Journal BINUS University*, vol. 5, no. 1, pp. 1–20, 2014
- [2] K. N. H. W. Semaya Bakti, Sura Ardana, "Ragam Hias Pura Daging Carik Di Desa Tejakula," *E-Journal Uniknas*, vol. 2, no. 1, pp. 1–10, 2013.
- [3] R. Yusuf and G. W. Darmawan, "APLIKASI BERBASIS WEB DENGAN MENGGUNAKAN PUSTAKA JAVASCRIPT FABRICJS UNTUK PEMBUATAN KOMIK STRIP PUNAKAWAN," *E-Journal Stmik Amikom Yogyakarta*, vol. 13, no. 1, pp. 1–6, 2016.
- [4] D. Geary, *Core HTML5 Canvas*. United States: U.S. Corporate and Government Sales, 2012.
- [5] M. R. Kurnia, H. Tjandrasa, and Y. Wijaya, "Implementasi Segmentasi Pembuluh Darah Retina Pada Citra Fundus Mata Menggunakan Tekstur , Thresholding dan Operasi Morfologi," *E-Journal Publikasi Ilmiah Online Mahasiswa ITS*, vol. 1, no. 1, pp. 1–6, 2012.
- [6] S. I. Syafi'i, R. T. Wahyuningrum, and A. Muntasa, "Segmentasi Obyek Pada Citra Digital Menggunakan Metode Otsu Thresholding," *E-Jurnal Informatika Petra*, vol. 13, no. 1, pp. 1–8, 2016.
- [7] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: C.V. Andi Offset, 2010.
- [8] I. G. Ag *et al.*, "Rancang Bangun Aplikasi Pengenalan Aksara Bali dengan Metode Kurva," *Lontar Komputer*, vol. 5, no. 1, pp. 382–391, 2014.
- [9] P. Novenando, M. Weking, I. P. Agung, and I. N. Piarsa, "Rancang Bangun Aplikasi Pendeteksi Titik Koordinat Frekuensi Lightning Whistler," *Lontar Komputer*, vol. 7, no. 3, pp. 193–202, 2016.
- [10] N. P. Sutramiani, I. K. Gede, D. Putra, and M. Sudarma, "Local Adaptive Thresholding Pada Preprocessing Citra Lontar Aksara Bali," *E-Journal Tekno Elektro*, vol. 14, no. 1, pp. 1–4, 2015.
- [11] M. Radiawan, *Ornamen Tradisi Bali*. BALI: Fakultas Seni Rupa dan Desain Institut Seni Indonesia Denpasar, 2010.