

p5.js AS AN INCLUSIVE MEDIUM FOR GENERATIVE ART IN ART EDUCATION

p5.js SEBAGAI MEDIUM INKLUSIF DALAM PRAKTIK SENI GENERATIF PADA PENDIDIKAN SENI

Devi Ratnasari^{1*}, Anitalia Stefany Welayana², Arya Firmansyah³

^{1*,2,3} Program Studi Seni Murni, Fakultas Seni Rupa dan Desain,
Institut Seni Indonesia Surakarta

^{1*}deviratnasari@isi.ac.id, ²anitaliastefanywelayana@isi.ac.id,

³aryafirmansyah21@gmail.com

[Article History] Submitted: November 2, 2025; Revised: November 20, 2025;
Accepted: December 31, 2025

ABSTRACT

*The rapid development of digital technology has opened new opportunities in both art practice and education, one of which is through code-based generative art. However, for art students, programming is often perceived as a barrier due to its complexity and lack of beginner-friendly accessibility. This study aims to explore **p5.js** as an inclusive medium in generative art practice, focusing on the creation of algorithm-based visual works. The research employs a **practice-based research** approach, where the creation process is carried out through a series of algorithmic experiments involving iteration, branching, transformation, randomness, and mathematical functions. Each experiment produces visual outputs that are documented and reflectively analyzed to understand the relationship between algorithmic structures and the resulting aesthetic outcomes. The findings show that **p5.js**, with its simple, web-based, and easily accessible interface, is effective as an inclusive medium for learning generative art. This process not only produces representative artworks but also demonstrates how algorithms can function as a visual language that bridges art and technology within the context of art education.*

Keywords: *p5.js, generative art, graphic algorithms, digital technology, practice-based research*

ABSTRAK

Kemajuan teknologi digital telah membuka peluang baru dalam praktik dan pembelajaran seni, salah satunya melalui seni generatif berbasis kode. Namun, bagi mahasiswa seni, pemrograman sering dipandang sebagai hambatan karena dianggap kompleks dan tidak ramah bagi pemula. Penelitian ini bertujuan untuk mengeksplorasi p5.js sebagai medium inklusif dalam praktik seni generatif, dengan fokus pada penciptaan karya visual berbasis algoritma. Metode yang digunakan adalah *practice-based research*, di mana penciptaan karya dilakukan melalui serangkaian eksperimen algoritmik yang melibatkan perulangan,

percabangan, transformasi, fungsi acak, dan fungsi matematis. Setiap eksperimen menghasilkan keluaran visual yang didokumentasikan dan dianalisis secara reflektif untuk memahami keterkaitan antara struktur algoritmik dan hasil estetika yang tercipta. Hasil penelitian menunjukkan bahwa p5.js, dengan antarmuka yang sederhana, berbasis web, dan mudah diakses, efektif digunakan sebagai medium inklusif dalam pembelajaran seni generatif. Proses ini tidak hanya menghasilkan karya representatif, tetapi juga memperlihatkan bagaimana algoritma dapat berfungsi sebagai bahasa visual yang menjembatani seni dan teknologi dalam konteks pendidikan seni.

Kata kunci: p5.js, Seni Generatif, Algoritma Grafis, Pendidikan Seni, Practice Based Research

PENDAHULUAN

Dalam beberapa dekade terakhir, kemajuan teknologi digital yang begitu pesat telah mengubah cara manusia dalam berinteraksi, bekerja, hingga mencipta, termasuk berkarya dalam dunia seni. Perkembangan teknologi tidak hanya menciptakan alat baru, tetapi juga menggeser paradigma estetika dan cara berpikir kreatif seniman. Salah satu wujud dari perubahan ini adalah munculnya seni generatif (*generative art*). Seni generatif merupakan sebuah karya seni yang diciptakan dengan bantuan sistem algoritma. Berbeda dengan metode konvensional yang telah ada selama ini, seni generatif memungkinkan seniman untuk menggunakan kode sebagai medium penciptaan layaknya menggambar di atas kanvas. Cara ini mampu menghadirkan visual karya dengan pola-pola yang terstruktur namun juga sering tidak terduga hasilnya (McCormack & D'Inverno, 2012) (Giannini & Bowen, 2023). Di sinilah seni dan teknologi bertemu dalam sebuah proses yang kolaboratif dan eksploratif melalui komputasi.

Di lingkungan pendidikan, integrasi seni dan teknologi telah lama diterapkan di berbagai institusi internasional. MIT (*Massachusetts Institute of Technology*), Amerika Serikat misalnya telah menerapkan pemrograman ke dalam kurikulum seni mereka melalui berbagai program dan inisiatif yang menggabungkan seni dan teknologi seperti ACT (*Art, Culture, and Technology*) dan MAS (*Media Arts and Sciences*). Bahkan mereka menyediakan sebuah wadah khusus bagi peneliti di bidang seni dan teknologi melalui MIT Media Lab dan telah menghasilkan banyak inovasi karya seni yang kreatif dan fokus pada pengembangan teknologi. Praktik semacam ini menjadi inspirasi bagi pendidikan seni di era digital, termasuk di Indonesia.

Salah satu *tools* atau alat yang banyak digunakan dalam praktik seni generatif adalah p5.js. Merupakan sebuah pustaka pemrograman berbasis *JavaScript* dengan antarmuka yang mudah diakses oleh seniman. Hal ini merupakan faktor penting pada media digital untuk menjadi ruang bagi seniman berkesperimen tanpa hambatan teknis yang berat karena memiliki kualitas visual,

interaktivitas, dan pengalaman pengguna yang baik (Aji et al., 2023). Melalui pustaka ini, pengguna dapat menciptakan karya visual yang interaktif melalui kode pemrograman menggunakan elemen-elemen bentuk dan struktur algoritma.

Di Indonesia, beberapa penelitian telah mengeksplorasi seni generatif dari berbagai pendekatan. Pandanwangi, misalnya mengeksplorasi penggunaan seni algoritma sebagai solusi untuk merancang motif geometris melalui teknologi komputer. Penelitiannya menggunakan metode deskriptif kualitatif, dengan fokus pada penggunaan perangkat lunak *processing* dan *algorithmic art* untuk menghasilkan motif-motif geometris secara digital (Pandanwangi et al., 2022). Selanjutnya Pasaribu, yang mengintegrasikan seni tradisional batik dengan teknologi modern melalui penerapan konsep fraktal. Penelitian ini mengembangkan sebuah website interaktif berbasis fraktal yang memungkinkan pengguna mendesain motif batik secara mandiri menggunakan algoritma fungsi matematis (Pasaribu et al., 2024). Sementara Soikun, menggunakan bahasa pemrograman Python untuk memperkenalkan seni generatif sebagai alternatif seni. Melalui pendekatan berbasis praktik, penelitiannya menunjukkan bagaimana Python dan berbagai *library*-nya seperti *matplotlib*, *numpy*, dan *turtle* dapat digunakan untuk membuat karya seni generatif dengan menerapkan algoritma fungsi matematis dengan parameter yang dapat disesuaikan (Soikun, 2023). Meskipun beberapa penelitian telah dilakukan, namun belum ada penelitian yang secara khusus mengeksplorasi potensi *p5.js* sebagai medium penciptaan seni generatif dalam konteks seni rupa Indonesia. Dengan demikian, penelitian ini hadir untuk mengisi kekosongan tersebut, dengan menekankan potensi *p5.js* sebagai medium yang inklusif bagi seniman dan mahasiswa seni..

Dalam konteks pendidikan seni, *p5.js* menghadirkan peluang inklusif bagi mahasiswa untuk mengenal seni generatif tanpa harus memiliki latar belakang pemrograman yang kuat. Karena berbasis web dan bersifat *open-source*, mahasiswa dapat langsung menulis, menjalankan, serta membagikan kode mereka dengan mudah, bahkan hanya melalui *browser* (peramban internet). Hal ini menciptakan proses belajar yang lebih partisipatif, di mana eksperimen kreatif dapat dilakukan secara langsung dan hasilnya dapat segera dilihat. Pendekatan ini menawarkan cara baru bagi mahasiswa dalam berpikir visual berbasis logika komputasi, sekaligus membuka peluang yang lebih luas dalam eksplorasi estetika (McCarthy et al., 2015) (Wiguna et al., 2022).

Tujuan dari penelitian ini adalah mengeksplorasi penerapan algoritma grafis dasar seperti perulangan (*Looping*), percabangan (*Conditional*), fungsi acak (*Random*), transformasi geometris, dan fungsi matematis pada *p5.js* sebagai medium penciptaan seni generatif. Eksperimen visual yang dihasilkan kemudian dianalisis secara visual untuk mengetahui bagaimana algoritma dapat dipahami sebagai bahasa visual. Melalui pendekatan *practice-based-research*, penelitian ini menempatkan penciptaan karya sebagai sumber pengetahuan karena

memungkinkan eksplorasi langsung terhadap proses kreatif, serta mampu memberikan pemahaman mahasiswa terkait hubungan antara struktur logika pemrograman dan hasil visual dalam konteks praktik seni generatif berbasis kode (Chibalashvili et al., 2023) (Levin & Brain, 2021).

PEMBAHASAN

Bagian ini menyajikan hasil eksplorasi p5.js melalui serangkaian eksperimen algoritmik. Setiap eksperimen merepresntasikan satu kategori algoritma grafis. Hasil visual yang diperoleh kemudian dianalisis untuk menafsirkan hubungan antara struktur algoritmik dan karakter estetika yang dihasilkan.

Eksperimen Algoritma Grafis

a. Eksperimen 1: Perulangan (*Looping*)

Algoritma perulangan (*looping*) digunakan untuk membentuk pola repetitif dalam karya seni generatif. Melalui fungsi *for loop*, seniman dapat menciptakan susunan elemen visual yang berulang secara sistematis, baik dalam pola grid maupun komposisi teratur lainnya. Pola repetitif dalam seni generatif menekankan bagaimana algoritma sederhana dapat menghasilkan kompleksitas visual ketika digabung dengan variasi warna, ukuran, atau posisi (Wu, 2025).

Eksperimen pada algoritma perulangan dilakukan dengan membuat pola spiral yang terbuat dari *ellipse* (lingkaran kecil) yang mengikuti bentuk spiral. Setiap ellipse di tempatkan mengikuti sudut dan jari-jari tertentu yang bertambah secara bertahap, sehingga membentuk pola spiral yang berkelok-kelok dari pusat kanvas ke luar kanvas. Berikut merupakan kode (struktur algoritmik) dari eksperimen ini.

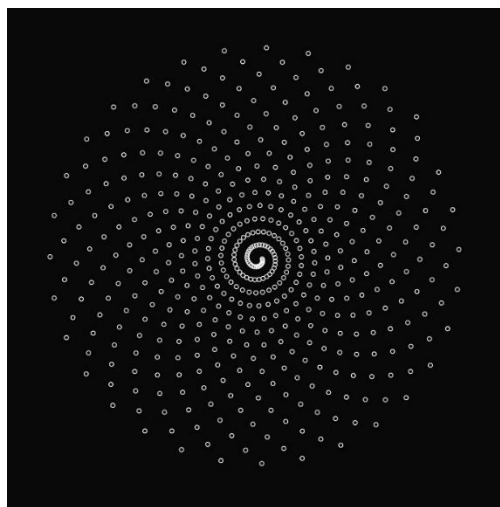
Tabel 1. Kode eksperimen 1: Perulangan (*Loop*)

```
function setup() {  
  createCanvas(600, 600);  
  background(10);  
  stroke(255);  
  noFill();  
  noLoop();  
  
  let angle = 0;  
  let radius = 5;  
  
  for (let i = 0; i < 500; i++) {  
    let x = width/2 + cos(angle) * radius;  
    let y = height/2 + sin(angle) * radius;  
    ellipse(x, y, 5);  
    angle += 0.2;  
    radius += 0.5;  
  }  
}
```

Program diawali dengan inisialisasi variabel *angle* dan *radius* pada nilai 0 dan 5. Kemudian, sebuah perulangan (*for loop*) berjalan sebanyak 500 kali. Pada setiap langkah, posisi ellipse dihitung berdasarkan sudut dan radius yang bertambah setiap iterasi:

- x dan y dihitung dengan rumus koordinat dari pusat (kanvas tengah) yang diputar mengikuti sudut *angle*.
- *ellipse*(x, y, 5) menggambar lingkaran kecil di posisi tersebut dengan ukuran tetap 5 piksel.
- Setelah gambar, *angle* ditambah 0.2 radian dan *radius* ditambah 0.5 sehingga setiap *ellipse* berposisi semakin jauh dari pusat dan membentuk pola spiral.

Dengan konsep, proses dan struktur algoritmik yang diterapkan, eksperimen 1: Perulangan menghasilkan visual seperti terlihat pada Gambar 1.



Gambar 1. Hasil Visual eksperimen *Looping ellipse* berbentuk spiral

b. Eksperimen 2: Percabangan (*Conditional*)

Algoritma percabangan (*conditional*) digunakan untuk menciptakan keputusan visual berdasarkan kondisi tertentu. Dalam konteks seni generatif, percabangan berfungsi sebagai logika penentu variasi bentuk, warna, atau perilaku visual lainnya, sehingga menghasilkan komposisi yang lebih dinamis dan tidak terduga. Hal ini memperlihatkan bagaimana keputusan logis dapat diterjemahkan menjadi pilihan visual yang memiliki nilai estetik.

Eksperimen pada algoritma percabangan dilakukan dengan membuat bentuk komposisi grid dengan variasi bentuk dan warna. Komposisi dibangun dari bentuk-bentuk sederhana seperti persegi, lingkaran, dan segitiga yang dipilih secara acak berdasarkan nilai probabilitas tertentu. Variasi ukuran dan warna

ditambahkan agar pola yang dihasilkan lebih menarik dan tidak monoton. Berikut merupakan kode (struktur algoritmik) dari eksperimen ini.

Tabel 2. Kode eksperimen 2: Percabangan (*Conditional*)

```
function setup() {
  createCanvas(600, 600);
  background(255);
  colorMode(HSB, 360, 100, 100, 100);
  noStroke();

  let grid = 30;
  for (let x = 0; x < width; x += grid) {
    for (let y = 0; y < height; y += grid) {
      let r = random(1);

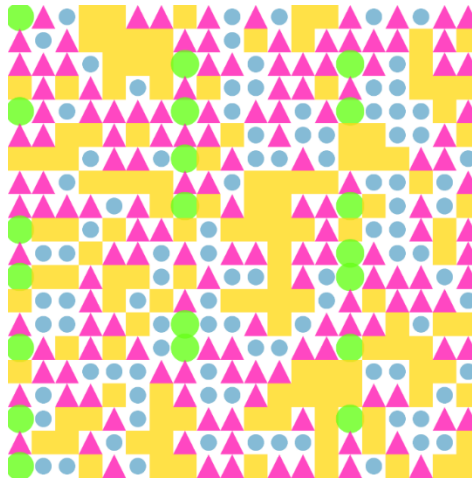
      if (r < 0.3) {
        fill(50, 90, 100, 80); //kuning
        rect(x, y, grid, grid);
      } else if (r < 0.6) {
        if (x % 70 === 0) {
          fill(100, 90, 100, 80); //hijau
          ellipse(x+grid/2, y+grid/2, grid*1.2);
        } else {
          fill(200, 50, 80, 80); //biru
          ellipse(x+grid/2, y+grid/2, grid*0.7);
        }
      } else {
        fill(320, 90, 100, 80); //pink
        triangle(x, y+grid, x+grid/2, y, x+grid, y+grid);
      }
    }
  }
}
```

Program di atas menggunakan dua lapisan perulangan (*nested loops*) untuk membangun struktur grid dari kiri ke kanan dan atas ke bawah dengan jarak antar titik sebesar 30 piksel. Pada setiap titik grid, dihasilkan angka acak (r) antara 0–1 sebagai dasar keputusan bentuk dan warna.

- Jika $r < 0.3$ → digambar persegi berwarna kuning.
- Jika r berada di antara 0.3–0.6 → digambar bentuk *ellipse* berwarna hijau (jika posisi x kelipatan 70) atau *ellipse* kecil berwarna biru (jika tidak).
- Jika $r > 0.6$ → digambar segitiga berwarna pink.

Melalui logika ini, sistem menentukan elemen visual berdasarkan peluang dan posisi, menghasilkan variasi bentuk dan warna pada setiap titik.

Dengan konsep, proses dan struktur algoritmik yang diterapkan, eksperimen 2: Percabangan menghasilkan visual seperti terlihat pada Gambar 2.



Gambar 2. Hasil Visual eksperimen 2: Percabangan

c. Eksperimen 3: Fungsi Acak (*Random*)

Algoritma acak (*randomness*) dalam seni generatif digunakan untuk menciptakan elemen-elemen visual yang tidak dapat diprediksi, menghadirkan variasi dan spontanitas di setiap hasil karya. Fungsi *random()* pada *p5.js* memungkinkan seniman menghasilkan nilai acak untuk berbagai parameter seperti warna, ukuran, posisi, atau bentuk. Pendekatan ini memberikan peluang untuk mengeksplorasi estetika ketidakteraturan, di mana setiap eksekusi kode menghasilkan komposisi yang unik dan tidak berulang.

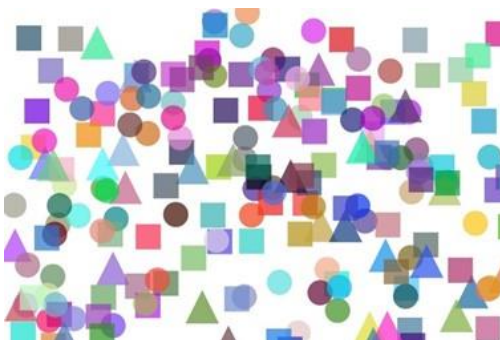
Eksperimen ini dilakukan untuk menunjukkan bagaimana nilai acak dapat dimanfaatkan dalam menciptakan variasi bentuk dan warna secara spontan. Dalam percobaan ini, dibuat gambar yang terdiri dari berbagai bentuk secara acak dan berwarna-warni. Sebelum menggambar, urutan bentuk diacak menggunakan fungsi *shuffle()* supaya setiap kali program dijalankan, tampilannya berbeda. Kemudian, bentuk-bentuk tersebut digambar secara acak di posisi yang berbeda di atas kanvas, dengan warna yang juga acak. Berikut merupakan kode (struktur algoritmik) dari eksperimen ini.

Tabel 3. Kode eksperimen 3: Algoritma Fungsi Acak (*Random*)

```
let saveButton;  
let shapes = ["ellipse", "rect", "triangle", "ellipse", "rect"];  
  
function setup() {  
  createCanvas(600, 400);  
  background(255);  
  noStroke();  
  
  saveButton = createButton("Download");  
  saveButton.position(10, 10);  
  saveButton.mousePressed(() => saveCanvas("Acak5", "jpg"));  
  
  shapes = shuffle(shapes); // acak urutan bentuk  
  
  for (let i = 0; i < 200; i++) {  
    let s = random(shapes);  
    fill(random(255), random(255), random(255), 180);  
    let x = random(width);  
    let y = random(height);  
    if (s === "ellipse") ellipse(x, y, 30, 30);  
    else if (s === "rect") rect(x, y, 30, 30);  
    else if (s === "triangle") triangle(x, y, x+20, y+40, x-20, y+40);  
  }  
}
```

Pada program diatas, bentuk-bentuk yang akan digambar disimpan dalam *array*, lalu urutannya diacak dengan *shuffle()*. Setelah itu, program melakukan pengulangan sebanyak 200 kali; di setiap langkah, bentuk yang akan digambar dipilih secara acak dari *array* yang sudah diacak. Warna setiap bentuk juga diatur secara acak, begitu juga dengan posisi di kanvas. Bentuk yang digambar bisa berupa lingkaran, kotak, atau segitiga, tergantung dari pilihan acak tersebut. Jadi, secara keseluruhan, program ini secara otomatis membuat gambar penuh bentuk yang berbeda-beda setiap kali dijalankan.

Dengan konsep, proses dan struktur algoritmik yang diterapkan, eksperimen 3: Algoritma Fungsi Acak menghasilkan visual seperti terlihat pada Gambar 3.



Gambar 3. Hasil Visual eksperimen 3: Algoritma Fungsi Acak

d. Eksperimen 4: Transformasi Geometris

Transformasi geometris merupakan salah satu algoritma penting dalam seni generatif untuk memanipulasi posisi, bentuk, atau orientasi elemen visual pada kanvas. Melalui fungsi transformasi seperti *translate()*, *rotate()*, dan *scale()*, seniman dapat mengatur ulang koordinat ruang gambar secara efisien tanpa harus menghitung posisi absolut setiap elemen. Prinsip ini memungkinkan penciptaan komposisi yang lebih dinamis dan terstruktur berdasarkan perpindahan sistem koordinat.

Eksperimen 4 dilakukan untuk menunjukkan bagaimana salah satu fungsi transformasi yaitu *translate()* dapat digunakan untuk memindahkan posisi gambar dan menciptakan beberapa bentuk yang tersebar di area kanvas. Dengan teknik translasi, seniman dapat menggambar bentuk pada posisi relatif terhadap titik baru yang telah digeser, sehingga proses kreatif menjadi lebih efisien dan fleksibel. Berikut merupakan kode (struktur algoritmik) dari eksperimen ini.

Tabel 4. Kode eksperimen 4: Transformasi Geometris

```
let saveButton;

function setup() {
  createCanvas(600, 600);
  background(240);
  noStroke();

  // Tombol download
  saveButton = createButton("Download Karya");
  saveButton.position(10, 10);
  saveButton.mousePressed(() => saveCanvas("transform1", "jpg"));

  // Warna dasar
  fill(200, 100, 150, 180);
  rect(100, 100, 150, 80);

  // Geser kanvas dengan translate untuk menaruh bentuk lain
  translate(200, 150);
  fill(100, 180, 200, 180);
  ellipse(0, 0, 120, 120);

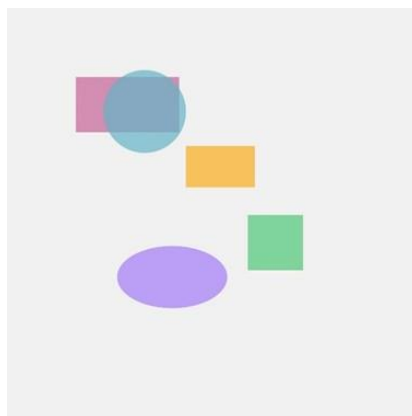
  translate(120, 90);
  fill(250, 180, 50, 200);
  rect(-60, -40, 100, 60);

  translate(-80, 150);
  fill(150, 100, 250, 150);
  ellipse(0, 0, 160, 90);

  translate(150, -50);
  fill(80, 200, 120, 180);
  rect(-40, -40, 80, 80);
}
```

Pada program di atas, Gambar pertama berupa persegi berwarna merah muda dibuat di posisi awal (di koordinat asli). Setelah itu, fungsi *translate*(200, 150) dipanggil, yang menggeser seluruh koordinat ke posisi baru. Di posisi ini, gambar ellipse berwarna biru muda digambar di pusat posisi tersebut. Selanjutnya, fungsi *translate*(120, 90) memindahkan posisi lagi ke kanan dan ke atas, lalu gambar persegi panjang berwarna kuning digambar di posisi tersebut, dengan koordinat relatif terhadap posisi baru. Kemudian, fungsi *translate*(-80, 150) kembali menggeser posisi jauh ke kiri dan ke bawah, dan gambar ellipse besar berwarna ungu digambar di posisi itu. Akhirnya, fungsi *translate*(150, -50) menggeser posisi lagi ke kanan dan ke atas, dan gambar persegi berwarna hijau diambil di posisi itu juga. Setiap kali *translate*() dipanggil, posisi gambar berikutnya bergeser sesuai nilai yang ditentukan, sehingga semua bentuk ditempatkan di posisi berbeda di kanvas, tidak perlu menghitung posisi secara manual lagi.

Berdasarkan konsep, proses dan struktur algoritmik yang diterapkan, eksperimen 4 menghasilkan visual seperti terlihat pada Gambar 4.



Gambar 4. Hasil Visual eksperimen 4: Algoritma Fungsi Transformasi Geometris

e. Eksperimen 5: Fungsi Matematis

Algoritma fungsi matematis memiliki peran penting dalam seni generatif karena mampu menghasilkan pola visual yang teratur namun kompleks melalui penerapan rumus dan persamaan tertentu. Fungsi trigonometri seperti *cos*() dan *sin*() sering digunakan untuk membangun bentuk yang bersifat berulang, harmonis, dan simetris.

Eksperimen ini dilakukan untuk memperlihatkan bagaimana fungsi *cos*() dapat dimanfaatkan untuk menciptakan pola berbentuk bintang dengan variasi radial. Melalui pendekatan ini, konsep gelombang berulang diterjemahkan ke dalam ruang dua dimensi menggunakan sistem koordinat polar, di mana setiap titik ditentukan oleh nilai fungsi trigonometri yang berubah secara kontinu. Berikut merupakan kode (struktur algoritmik) dari eksperimen ini.

Tabel 5. Kode eksperimen 5: Fungsi Matematis

```
let saveButton;  
  
function setup() {  
  createCanvas(600, 400);  
  background(240);  
  noStroke();  
  
  saveButton = createButton("Download");  
  saveButton.position(10, 40);  
  saveButton.mousePressed(() => saveCanvas("math2", "jpg"));  
  
  translate(width/2, height/2);  
  for (let a = 0; a < TWO_PI; a += 0.1) {  
    let r = 150 + cos(a * 5) * 50;  
    let x = cos(a) * r;  
    let y = sin(a) * r;  
    fill(100, 150, 255, 150);  
    ellipse(x, y, 10, 10);  
  }  
}
```

Program dimulai dengan membuat kanvas berukuran 600×400 piksel dan mengatur titik pusat koordinat ke tengah kanvas menggunakan fungsi *translate(width/2, height/2)*. Perulangan dilakukan dari sudut 0 hingga *TWO_PI* (setara dengan 360°) dengan langkah sebesar 0.1 radian. Pada setiap langkah, radius (r) dihitung menggunakan fungsi $\cos(a * 5)$ yang dikalikan dengan 50 dan ditambahkan ke nilai dasar 150. Perubahan nilai ini menghasilkan variasi jarak dari pusat yang berulang secara periodik, menciptakan bentuk menyerupai bintang. Selanjutnya, posisi koordinat x dan y dihitung berdasarkan nilai $\cos(a)$ dan $\sin(a)$ dikalikan dengan radius r . Setiap titik digambar menggunakan fungsi *ellipse()* dengan warna biru muda transparan, sehingga keseluruhan komposisi membentuk pola radial yang simetris.

Dengan konsep, proses, dan struktur algoritmik yang diterapkan, eksperimen 5: Fungsi Matematis menghasilkan visual seperti terlihat pada Gambar 5.



Gambar 5. Hasil Visual eksperimen 5: Algoritma Fungsi Matematis

Analisis Hasil Eksperimen

Analisis dilakukan untuk memahami bagaimana setiap kategori algoritma dalam p5.js yaitu perulangan, percabangan, fungsi acak, transformasi geometris, dan fungsi matematis membentuk struktur visual tertentu dan menghasilkan karakter estetika yang khas. Berikut merupakan hubungan antara aspek algoritmik dan estetika visual dari hasil setiap eksperimen yang dilakukan.

Tabel 6. Hubungan antara aspek algoritmik dan estetika visual

Eksperimen	Struktur Algoritmik	Karakteristik Estetika yang Dihasilkan
Perulangan (<i>Looping</i>)	Menggunakan <i>for loop</i> dengan peningkatan sudut dan radius secara bertahap untuk menggambar elemen berulang (<i>ellipse</i>) membentuk pola spiral.	Pola spiral yang teratur, ritmis, dan harmonis.
Percabangan (<i>Conditional</i>)	Menggunakan <i>if-else</i> untuk menentukan bentuk dan warna berdasarkan nilai probabilitas dan posisi grid.	Komposisi visual yang dinamis dan variatif, dengan kontras warna dan bentuk berdasarkan kondisi tertentu.
Fungsi Acak (<i>Random</i>)	Memanfaatkan <i>random()</i> dan <i>shuffle()</i> untuk menentukan urutan bentuk, warna, dan posisi elemen.	Visual tampak spontan, ekspresif, dan tidak berulang. Setiap hasil memiliki pola unik.
Transformasi Geometris	Menggunakan fungsi <i>translate()</i> untuk memindahkan titik koordinat sehingga setiap bentuk digambar dari posisi relatif baru.	Komposisi terstruktur dengan persepsi ruang yang berlapis dan dinamis.
Fungsi Matematis	Menggunakan fungsi <i>cos()</i> dan <i>sin()</i> untuk menghitung posisi titik dalam sistem koordinat polar, menghasilkan pola radial.	Pola menyerupai bintang yang simetris.

Hasil eksperimen menunjukkan bahwa setiap algoritma memiliki peran berbeda dalam membentuk pengalaman visual. Struktur algoritmik berfungsi sebagai kerangka logis, sedangkan karakter estetika menjadi hasil ekspresif dari

penerapan logika tersebut. Pada eksperimen perulangan, struktur logika yang sederhana justru menghasilkan komposisi yang kompleks. Pola spiral yang muncul menjadi representasi visual dari konsep keteraturan matematis yang bersifat estetik. Sementara itu, percabangan menampilkan bagaimana keputusan logis “jika–maka” dapat diterjemahkan menjadi keputusan visual berupa variasi bentuk dan warna. Eksperimen fungsi acak memperlihatkan bahwa ketidakteraturan dapat menjadi sumber estetika baru. Nilai acak memunculkan pola yang tak dapat diprediksi. Sedangkan pada transformasi geometris, logika koordinat menjadi dasar penciptaan ruang visual yang lebih kompleks. Pergeseran posisi melalui *translate()* menunjukkan bagaimana algoritma memediasi persepsi ruang dan kedalaman dalam komposisi digital. Dan pada fungsi matematis, fungsi cosinus maupun formula matematika tertentu dapat memvisualisasikan gelombang berulang yang harmonis.

KESIMPULAN

Penelitian ini menunjukkan bahwa *p5.js* dapat digunakan sebagai medium inklusif dalam praktik seni generatif berbasis kode, terutama bagi mahasiswa seni yang baru mengenal pemrograman. Eksperimen visual dengan berbagai algoritma grafis juga memperlihatkan bahwa setiap pendekatan logika komputasi memiliki dampak langsung terhadap bentuk dan komposisi karya seni generatif. Perulangan (*looping*) menghasilkan pola repetitif yang memberi kesan keteraturan dan ritme visual, sementara percabangan (*conditional*) menghadirkan kontras yang jelas melalui variasi warna, bentuk, atau posisi. Fungsi acak (*random*) menghasilkan komposisi yang tidak terduga, menekankan spontanitas dan ketidakpastian estetik. Transformasi geometris seperti rotasi, translasi, dan skala memungkinkan terbentuknya komposisi dinamis yang memperlihatkan gerakan, sementara fungsi matematis seperti sinus dan kosinus dapat membangun pola melingkar atau berulang. Dari keseluruhan eksperimen ini dapat disimpulkan bahwa algoritma bukan hanya instruksi teknis, melainkan juga berfungsi sebagai “bahasa visual” bagi seniman untuk menciptakan dan membentuk karya generatif yang estetik sesuai dengan keinginan mereka. Pada akhirnya, logika komputasi yang ada dapat menggantikan gestur tangan seniman, sedangkan keputusan dalam kode nya menjadi tindakan artistik. Hal ini membuka paradigma baru bahwa dalam pembelajaran dan praktik seni berbasis generatif, estetika digital lahir melalui kombinasi antara struktur logika dan intuisi visual seniman. Selain itu, serangkaian eksperimen yang dilakukan dapat diterapkan sebagai model pembelajaran berbasis praktik di kelas seni rupa, untuk mendorong mahasiswa memahami logika komputasi melalui proses kreatif visual.

DAFTAR PUSTAKA

- Aji, A., Budiyono, N., Suhirman, S., Ratnasari, D., & Sejati, R. H. (2023). Metode Design Thinking Untuk Perancangan UI/UX Pada Aplikasi Konsultasi Karir. *INTEK: Jurnal Informatika Dan Teknologi Informasi*, 6(1), 42–48. <https://doi.org/10.37729/intek.v6i1.2997>
- Chibalashvili, A., Savchuk, I., Olianina, S., Shalinskyi, I., & Korenyuk, Y. (2023). Creative Coding as a Modern Art Tool. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 14(2), 115–127. <https://doi.org/10.18662/brain/14.2/447>
- Giannini, T., & Bowen, J. P. (2023). Generative Art and Computational Imagination: Integrating poetry and art. *Proceedings of Eva London 2023*, 211–219. <https://doi.org/10.14236/ewic/eva2023.37>
- Levin, G., & Brain, T. (2021). *Code as Creative Medium: A Handbook for Computational Art and Design*. The MIT Press.
- McCarthy, L., Reas, C., & Fry, B. (2015). *Getting started with p5.js: making interactive graphics in JavaScript and Processing*.
- McCormack, J., & D'Inverno, M. (2012). Computers and Creativity: The road ahead. In *Computers and Creativity*. Berlin, Heidelberg: Springer Berlin Heidelberg., 421–424.
- Pandanwangi, A., Prijono, A., & Ratnadewi, R. (2022). Generative Art sebagai Pembentuk Motif Geometris. *Ideas: Jurnal Pendidikan, Sosial, Dan Budaya*, 8(4), 1199. <https://doi.org/10.32884/ideas.v8i4.959>
- Pasaribu, M. A., Nasution, M. I., Mardiana, A., Cahaya, P., & Kartika, D. (2024). Transformasi Seni Tradisional ke Era Digital: Penerapan Konsep Fraktal dalam Desain Batik Berbasis Website untuk Meningkatkan Kreativitas dan Inovasi. *Journal Of Social Science Research Volume*, 4(6), 8578–8589.
- Soikun, T. M. (2023). Understanding of Generative Art With 'Python' Programming. *Jurnal Gendang Alam (GA)*, 13(2), 58–69. <https://doi.org/10.51200/ga.v13i2.4735>

- Wiguna, I. P., Zen, A. P., & Yuningsih, C. R. (2022). Painting with algorithms: The potential for using the P5.js programming language for new media artist. *Embracing the Future: Creative Industries for Environment and Advanced Society 5.0 in a Post-Pandemic Era*, 271–275. http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciu rbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI
- Wu, J. (2025). *Exploring Bridges Between Creative Coding and Visual Generative AI*. 1–15. <http://arxiv.org/abs/2406.05508>