Lampiran 1

Ringkasan Materi

Bagian 1. Dasar-dasar Algoritma

1.1 Konsep Dasar Algoritma

Definisi:
Algoritma adalah prosedur sistematis untuk menyelesaikan masalah melalui serangkaian instruksi logis dan terstruktur. Karakteristik algoritma yang baik meliputi:

- Input: Menerima data awal (misal: bahan membuat teh).
- Output: Menghasilkan hasil yang diharapkan (teh siap minum).
- Definiteness: Langkah-langkah harus jelas dan tidak ambigu.
- Finiteness: Harus berhenti setelah sejumlah langkah terbatas.
- Effectiveness: Setiap langkah harus dapat dieksekusi secara efisien.

Contoh Kehidupan Nyata:

Algoritma Login:

1. Input username dan password.
2. Validasi kecocokan dengan database.
3. Jika valid, akses diberikan; jika tidak, tampilkan pesan error.

Algoritma Pencarian Google:

1. Terima kata kunci.
2. Cari indeks web yang relevan.
3. Urutkan berdasarkan relevansi.
4. Tampilkan hasil.

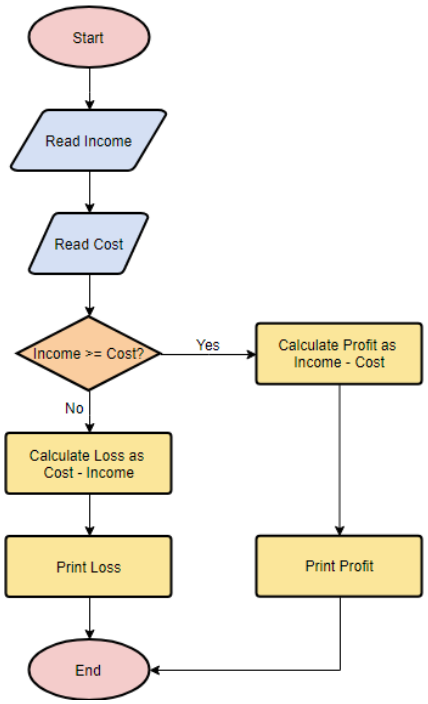
1.2 Flowchart dan Pseudocode

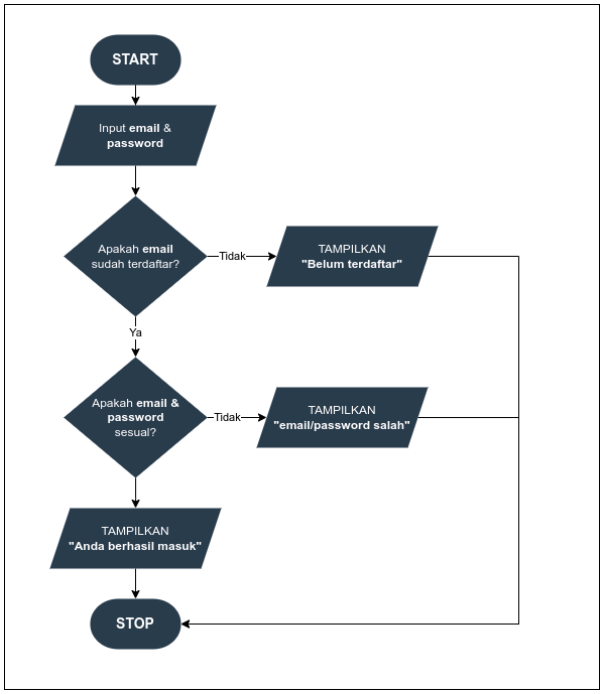
- **Flowchart:**
  Diagram alur yang menggunakan simbol standar:

	Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.
	On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.
	Off-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.
	Terminator Simbol yang menyatakan awal atau akhir suatu program.
	Process Simbol yang menyatakan suatu proses yang dilakukan komputer.
	Decision Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.

	Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedure.
	Display Simbol yang menyatakan peralatan output yang digunakan.
	Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Contoh flowchart:





- **Pseudocode:**
  Representasi logika algoritma dengan bahasa teks yang menyerupai bahasa alami, tanpa aturan sintaksis yang ketat.

- *Contoh pseudocode menentukan bilangan prima:*

```
1. MULAI
2.   BACA angka
3.   JIKA angka <= 1
4.     CETAK "Bukan prima"
5.   ELSE
6.     UNTUK i = 2 SAMPAI sqrt(angka)
7.       JIKA angka MOD i == 0
8.         CETAK "Bukan prima"
9.         BERHENTI
10.    CETAK "Prima"
11. SELESAI
```

Perbandingan Flowchart dan Pseudocode:

- **Flowchart: Visual, cocok untuk dokumentasi proses.**
- **Pseudocode: Lebih fleksibel untuk brainstorming logika.**

Aspek	Flowchart	Pseudocode
Definisi	Representasi grafis dari suatu algoritma yang menggunakan simbol-simbol standar (misalnya: oval, kotak, diamond) untuk menggambarkan alur proses.	Deskripsi langkah demi langkah dari algoritma yang ditulis dalam format teks menyerupai bahasa pemrograman namun tidak terikat oleh sintaks tertentu.
Representasi	Visual, berupa diagram alir.	Teks, ditulis dalam format yang mirip kode.
Keterbacaan	Mudah dipahami secara visual, terutama bagi yang terbiasa dengan diagram atau bagi orang awam.	Lebih cocok bagi programmer atau orang yang sudah memahami logika pemrograman, karena mendekati struktur kode sebenarnya.
Standarisasi	Memiliki standar simbol dan notasi, meskipun dapat bervariasi tergantung metode atau kebutuhan spesifik.	Tidak ada standar baku, sehingga penulis dapat menyesuaikan gaya penulisan sesuai kebutuhan.
Fleksibilitas	Lebih efektif untuk proses yang sederhana; diagram bisa menjadi rumit jika algoritma terlalu kompleks.	Fleksibel dan mudah diperbarui, sehingga lebih cocok untuk algoritma dengan logika kompleks dan banyak cabang kondisi.
Penggunaan	Sering digunakan dalam tahap perancangan awal, dokumentasi, dan	Digunakan untuk menyusun logika algoritma secara rinci sebelum

	presentasi untuk memvisualisasikan alur kerja suatu sistem.	implementasi ke dalam bahasa pemrograman yang sesungguhnya.
Alat/Software	Memerlukan software diagram atau bisa dibuat secara manual (misalnya dengan kertas dan pena).	Dapat ditulis menggunakan text editor sederhana tanpa memerlukan alat khusus.

Bagian 2. Struktur Dasar Pemrograman

2.1 Input dan Output

- **Definisi:**
  Input adalah data yang diterima oleh program dari pengguna atau sumber lain; output adalah hasil pemrosesan data yang ditampilkan.
- **Contoh Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main() {
    int angka;
    printf("Masukkan sebuah angka: ");
    scanf("%d", &angka);
    printf("Angka yang dimasukkan: %d\n", angka);
    return 0;
}

```

  - **Python:**

```python
angka = int(input("Masukkan sebuah angka: "))
print("Angka yang dimasukkan:", angka)

```

  - **JavaScript:**

```javascript
let angka = parseInt(prompt("Masukkan sebuah angka: "));
console.log("Angka yang dimasukkan:", angka);
```

Referensi: *Programiz Input/Output*

- **Validasi Input**:

  Contoh di Python:

```python
while True:
    try:
        angka = int(input("Masukkan angka: "))
        break
    except ValueError:
        print("Input harus bilangan bulat!")
```

  Contoh penggunaan validasi input di kehidupan sehari-hari:

  - Sistem ATM: Input PIN ⟶ Validasi ⟶ Tampilkan menu.

2.2 Percabangan (If-Else)

- **Konsep:**
  Memungkinkan program memilih jalur eksekusi berdasarkan kondisi.
- **Contoh Implementasi:**
  - **C**

```c
#include <stdio.h>
int main() {
    float suhu;
    printf("Masukkan suhu: ");
    scanf("%f", &suhu);
    if (suhu > 30) {
        printf("Panas!\n");
    } else {
        printf("Sejuk!\n");
    }
    return 0;
}

```

  - **Python:**

```python
suhu = float(input("Masukkan suhu: "))
if suhu > 30:
    print("Panas!")
else:
print("Sejuk!")

```

  - **JavaScript:**

```javascript
let suhu = parseFloat(prompt("Masukkan suhu: "));
if (suhu > 30) {
    console.log("Panas!");
} else {
    console.log("Sejuk!");
}

```

2.3 Teknik Loop (Perulangan)

- **Jenis Loop:**
  - **For Loop:** Digunakan ketika jumlah iterasi diketahui.
  - **While Loop:** Digunakan selama kondisi terpenuhi.
  - **Do-While Loop (C):** Menjamin eksekusi minimal satu kali.

- **Contoh Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main() {
    // For Loop
    for (int i = 1; i <= 5; i++) {
        printf("For Loop: %d\n", i);
    }
    // While Loop
    int j = 1;
    while (j <= 5) {
        printf("While Loop: %d\n", j);
        j++;
    }
    return 0;
}

```

  - **Python:**

```python
for i in range(1, 6):
    print("For Loop:", i)
i = 1
while i <= 5:
    print("While Loop:", i)
i += 1

```

  - **JavaScript:**

```javascript
for (let i = 1; i <= 5; i++) {
    console.log("For Loop:", i);
}
let i = 1;
while (i <= 5) {
    console.log("While Loop:", i);
    i++;
}

```

Referensi: GeeksforGeeks Loops

2.4 Deklarasi Variabel, Fungsi, dan Array

- **Konsep:**
  Variabel menyimpan data, fungsi mengenkapsulasi logika, dan array menyimpan koleksi data.
- **Contoh Implementasi:**
  - **C:**

```c
int a = 10;
int arr[5] = {1, 2, 3, 4, 5};
int tambah(int x, int y) {
    return x + y;
}

```

- **Python:**

```python
a = 10
arr = [1, 2, 3, 4, 5]
def tambah(x, y):
return x + y
```

- **JavaScript:**

```javascript
let a = 10;
let arr = [1, 2, 3, 4, 5];
function tambah(x, y) {
    return x + y;
}
```

Bagian 3. Materi Lanjutan dan Penggunaan Library Eksternal

3.1 Function (Fungsi)

- **Definisi:**
  Fungsi adalah blok kode yang dapat dipanggil berkali-kali untuk melakukan tugas tertentu.
- **Contoh Implementasi:**
  - **C, Python, JavaScript:** (lihat contoh pada Bagian 2.4 di atas)

Parameter Default (Python):

python

```python
def sapa(nama="Pengguna"):
    print(f"Halo, {nama}!")
sapa()  # Output: Halo, Pengguna!

```

Fungsi vs Prosedur:

Aspek	Fungsi	Prosedur
Definisi	Subprogram yang mengembalikan nilai sebagai hasil perhitungan.	Subprogram yang menjalankan serangkaian perintah atau aksi tanpa mengembalikan nilai.
Return Value	Mengembalikan nilai (hasil) kepada pemanggilnya.	Tidak mengembalikan nilai.
Penggunaan	Digunakan untuk operasi perhitungan atau proses yang menghasilkan output.	Digunakan untuk menjalankan perintah atau proses yang berfokus pada aksi.

Sintaks	Umumnya menggunakan kata kunci **function** dalam banyak bahasa pemrograman.	Umumnya menggunakan kata kunci **procedure** atau **void** (pada bahasa tertentu).
Contoh	function hitungLuasLingkaran(jari: float): float { ... }	procedure tampilkanPesan() { ... }
Fokus	Fokus pada pengembalian nilai sebagai hasil akhir dari perhitungan.	Fokus pada eksekusi urutan perintah untuk menghasilkan efek samping (side-effect).

3.2 Operator

Kategori Operator	Operator	Deskripsi	Contoh Penggunaan
Aritmatika	+	Penjumlahan dua nilai	a + b
	-	Pengurangan dua nilai	a - b
	*	Perkalian dua nilai	a * b
	/	Pembagian dua nilai	a / b
	%	Modulus (sisa pembagian)	a % b
	++	Increment (menambahkan 1)	a++ atau ++a
	--	Decrement (mengurangi 1)	a-- atau --a
Penugasan	=	Penugasan nilai	a = 10
	+=	Penugasan penjumlahan (a = a + nilai)	a += 5
	-=	Penugasan pengurangan (a = a - nilai)	a -= 3
	*=	Penugasan perkalian (a = a * nilai)	a *= 2
	/=	Penugasan pembagian (a = a / nilai)	a /= 2
	%=	Penugasan modulus (a = a % nilai)	a %= 4
	&=, |=, ^=, <<=, >>=	Penugasan bitwise (gabungan operasi bitwise dan assignment)	`a
Perbandingan	==	Membandingkan kesamaan nilai	a == b
	!=	Membandingkan ketidaksamaan nilai	a != b

	>	Memeriksa apakah nilai lebih besar	a > b
	<	Memeriksa apakah nilai lebih kecil	a < b
	>=	Memeriksa apakah nilai lebih besar atau sama	a >= b
	<=	Memeriksa apakah nilai lebih kecil atau sama	a <= b
	===	(Di JavaScript) Membandingkan nilai dan tipe secara ketat	a === b
	!==	(Di JavaScript) Membandingkan ketidaksamaan nilai dan tipe secara ketat	a !== b
Logika	&&	Operator AND logika	(a > b) && (c > d)
	||	Operator OR logika	(a > b) || (c > d)
	!	Operator NOT (negasi)	!(a > b)
Bitwise	&	Bitwise AND	a & b
	|	Bitwise OR	a | b
	^	Bitwise XOR	a ^ b
	~	Bitwise NOT	~a
	<<	Left shift (menggeser bit ke kiri)	a << 2
	>>	Right shift (menggeser bit ke kanan)	a >> 2
	>>>	Unsigned right shift (terutama di JavaScript)	a >>> 2
Ternary	? :	Operator kondisional (jika kondisi benar, pilih nilai pertama, sebaliknya nilai kedua)	(a > b) ? a : b
Lainnya	,	Operator koma (memisahkan beberapa ekspresi)	a, b, c
	sizeof	Mendapatkan ukuran tipe data (C/C++)	sizeof(int)
	new	Mengalokasikan objek baru (OOP)	new ClassName()

	delete	Menghapus objek atau mengembalikan memori (C++)	delete ptr
	instanceof	Memeriksa apakah objek merupakan instansi dari suatu kelas (Java, JavaScript)	obj instanceof Class
	typeof	Mengetahui tipe data dari variabel (JavaScript)	typeof variable
	.	Akses anggota objek atau properti	object.property
	->	Akses anggota objek melalui pointer (C/C++)	pointer->member
	::	Operator resolusi scope (mengakses anggota statis atau namespace, di C++)	ClassName::staticMethod()

- **Contoh Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main() {
    int a = 10, b = 3;
    printf("Penjumlahan: %d\n", a + b);
    printf("Modulus: %d\n", a % b);
    printf("Lebih besar: %d\n", a > b);
    return 0;
}
```

  - **Python:**

```python
a = 10
b = 3
print("Penjumlahan:", a + b)
print("Modulus:", a % b)
print("Lebih besar:", a > b)

```

  - **JavaScript:**

```javascript
let a = 10, b = 3;
console.log("Penjumlahan:", a + b);
console.log("Modulus:", a % b);
console.log("Lebih besar:", a > b);

```

- **Operator Ternary:**

Python:

```
1. status = "Dewasa" if umur >= 18 else "Anak-anak"
```

JavaScript:

```
1. let status = (umur >= 18) ? "Dewasa" : "Anak-anak";
```

- **Operator Bitwise:**

  C:

```
1. int a = 5; // 0101
2. int b = 3; // 0011
3. printf("%d", a & b);  // 0001 → 1
```

- **Scope Variabel:**
  - **Global vs Lokal:**

Global Scope	Variabel yang dideklarasikan di luar fungsi atau blok kode.	Seluruh program	Sepanjang runtime program	Dapat diakses dari bagian manapun	Memudahkan berbagi data antar bagian program. Namun, penggunaan berlebihan dapat menyebabkan konflik nama (namespace collision) dan sulit untuk dilacak perubahannya, sehingga berpotensi menimbulkan bug.
Function Scope	Variabel yang dideklarasikan dalam sebuah fungsi.	Terbatas pada fungsi tempat variabel dideklarasikan	Hanya selama fungsi berjalan (kecuali penggunaan closure)	Hanya dapat diakses di dalam fungsi	Mengisolasi variabel dari lingkungan eksternal, sehingga menghindari tabrakan nama dan meningkatkan enkapsulasi. Namun, variabel ini tidak tersedia di luar fungsi, sehingga perlu pengelolaan data secara eksplisit jika diperlukan.

Block Scope	Variabel yang dideklarasikan dalam blok kode (misalnya: if, loop).	Terbatas pada blok tempat variabel dideklarasikan	Hanya selama eksekusi blok	Hanya dapat diakses di dalam blok	Membantu menjaga keteraturan kode dengan membatasi ruang lingkup variabel. Implementasi block scope bervariasi antar bahasa (contoh: let dan const di JavaScript vs. var yang tidak mendukung block scope secara penuh).
Module Scope	Variabel yang dideklarasikan pada level modul atau file.	Seluruh modul (file)	Sepanjang modul di-load	Terbatas pada modul, kecuali diekspos secara eksplisit	Mendukung modularitas dan pemisahan kode, sehingga mengurangi konflik namespace antar modul. Variabel ini harus diimpor atau diekspor secara eksplisit agar dapat digunakan lintas modul, sehingga meningkatkan keamanan dan pemeliharaan kode.

Contoh penggunaan scope variable dalam C

```c
#include <stdio.h>

int globalVar = 10; // Variabel global

void contohFungsi() {
    int localVar = 20; // Variabel lokal (scope hanya di dalam fungsi)
    printf("Di dalam fungsi: globalVar = %d, localVar = %d\n", globalVar, localVar);

    // Contoh block scope di C
    {
        int blockVar = 30; // Variabel hanya di dalam blok ini
        printf("Di dalam blok: blockVar = %d\n", blockVar);
    }
    // printf("%d", blockVar); // Akan error, karena blockVar tidak dapat diakses di sini
}

int main() {
    contohFungsi();
    // printf("%d", localVar); // Akan error, karena localVar hanya ada di dalam contohFungsi()
    printf("Di main: globalVar = %d\n", globalVar);
    return 0;
```

```
}
```

- Global Scope: globalVar dideklarasikan di luar fungsi dan dapat diakses oleh seluruh fungsi dalam program.
- Function Scope: localVar hanya dapat diakses di dalam contohFungsi().
- Block Scope: blockVar hanya valid di dalam blok tempat ia dideklarasikan.

Contoh penggunaan scope variable dalam JavaScript

```
// Variabel global
var globalVar = "Global";

function contohFungsi() {
    // Variabel lokal
    var localVar = "Lokal";
    console.log("Di dalam fungsi:", globalVar, localVar);

    // Contoh block scope dengan let/const
    if (true) {
        let blockVar = "Block-Level";
        console.log("Di dalam blok:", blockVar);
    }
    // console.log(blockVar); // Akan error, karena blockVar hanya ada di dalam blok if

    // Contoh penggunaan var tidak mendukung block scope:
    if (true) {
        var tidakBlockVar = "Tidak Block-Level";
    }
    console.log("Variabel yang dideklarasikan dengan var dalam blok if, tapi dapat diakses di fungsi:", tidakBlockVar);
}

contohFungsi();
console.log("Di luar fungsi:", globalVar);
// console.log(localVar); // Akan error, karena localVar hanya dideklarasikan di dalam fungsi

```

- Global Scope: globalVar dideklarasikan di luar fungsi dan tersedia secara global.
- Function Scope: Variabel yang dideklarasikan dengan var di dalam fungsi (misalnya localVar dan tidakBlockVar) hanya dapat diakses di dalam fungsi tersebut.
- Block Scope: Variabel yang dideklarasikan dengan let atau const (misalnya blockVar) hanya tersedia dalam blok tempat mereka dideklarasikan.

Contoh penggunaan scope variable dalam Python

```
global_var = "Global"  # Variabel global

def contoh_fungsi():
    local_var = "Lokal"  # Variabel lokal
    print("Di dalam fungsi:", global_var, local_var)

    # Python tidak memiliki block scope seperti bahasa lain,
```

```
    # sehingga variabel yang dideklarasikan di dalam blok if tetap lokal untuk fungsi ini.
    if True:
        block_var = "Block-Level"  # Masih merupakan variabel lokal fungsi
        print("Di dalam blok:", block_var)

    # block_var masih dapat diakses di sini
    print("Di luar blok namun dalam fungsi:", block_var)

contoh_fungsi()
print("Di luar fungsi:", global_var)
# print(local_var)  # Akan error: NameError, karena local_var tidak didefinisikan secara global

```

- Global Scope: global_var dapat diakses di mana saja, termasuk di dalam fungsi.
- Function Scope: local_var (dan block_var) hanya dapat diakses di dalam contoh_fungsi().
- Block Scope: Python tidak menerapkan block scope pada struktur seperti if/else; variabel di dalam blok if tetap merupakan variabel fungsi.

- **Array Multidimensi:**
    - **Matriks di JavaScript:**

javascript

```
let matriks = [[1, 2], [3, 4]];
console.log(matriks[0][1]);  // 2

```

- **Fungsi Rekursif:**
    - **Deret Fibonacci (Python):**

python

```
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n-1) + fibonacci(n-2)

```

3.3 Output Formatting

- **Definisi:**
  Teknik untuk menampilkan output dengan format yang mudah dibaca.
- **Contoh Implementasi:**
    - **C:**

```
#include <stdio.h>
```

```c
int main() {
    float angka = 3.14159;
    printf("Angka dengan 2 desimal: %.2f\n", angka);
    return 0;
}

```

- **Python:**

```python
angka = 3.14159
print(f"Angka dengan 2 desimal: {angka:.2f}")
print("Angka dengan 2 desimal: {:.2f}".format(angka))

```

- **JavaScript:**

```javascript
let angka = 3.14159;
console.log("Angka dengan 2 desimal:", angka.toFixed(2));

```

- **Alignment & Padding:**
  - **Python:**

```python
print(f"{'Nama':<10} | {'Usia':>5}")
print(f"{'Alice':<10} | {25:>5}")
```

Output:

```
Nama       |  Usia
Alice      |    25
```

- **Notasi Ilmiah (JavaScript):**

```javascript
console.log(0.0015.toExponential(2));  // "1.50e-3"
```

3.4 Penggunaan Library Eksternal

- **Definisi:**
  Library eksternal menyediakan fungsi tambahan yang membantu menyederhanakan tugas-tugas kompleks.
- **Contoh Implementasi:**
  - **Python (Library Pillow untuk pengolahan gambar):**

```python
from PIL import Image

def convert_to_grayscale(image_path, output_path):
    image = Image.open(image_path)
```

```
    grayscale = image.convert("L")
    grayscale.save(output_path)
    print("Gambar berhasil diubah menjadi hitam putih.")

# Contoh penggunaan:
# convert_to_grayscale("input.jpg", "output.jpg")

```

- **JavaScript (Canvas API untuk manipulasi gambar):**

```
function convertToGrayscale() {
    const canvas = document.getElementById("myCanvas");
    const ctx = canvas.getContext("2d");
    const imgData = ctx.getImageData(0, 0, canvas.width, canvas.height);
    const data = imgData.data;
    for (let i = 0; i < data.length; i += 4) {
        const grayscale = 0.3 * data[i] + 0.59 * data[i+1] + 0.11 * data[i+2];
        data[i] = data[i+1] = data[i+2] = grayscale;
    }
    ctx.putImageData(imgData, 0, 0);
    alert("Gambar berhasil diubah menjadi hitam putih.");
}
// Pastikan ada <canvas id="myCanvas"></canvas> di HTML.

```

- **C:**

  Biasanya menggunakan header standar (misalnya, math.h). Contoh penggunaan telah diberikan pada bagian perhitungan akar kuadrat.

LAMPIRAN 2:

Contoh Diversifikasi Persoalan Untuk Berbagai Kelompok Kejuruan

Kelompok 1: Teknologi dan Rekayasa

Studi Kasus 1: Monitoring Efisiensi Mesin Produksi

- **Latar Belakang:**
  Pemantauan kinerja mesin produksi sangat penting untuk mencegah kerusakan dan mengoptimalkan output. Siswa harus memahami cara membaca data sensor dan menghitung rata-rata untuk menilai kinerja.
- **Konsep yang Diterapkan:**
  - Penggunaan perulangan untuk membaca data sensor.
  - Operasi aritmatika (rata-rata).
  - Struktur kondisi untuk evaluasi.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int i;
    float suhu, getaran;
    float total = 0;
    int n = 5;
    float threshold = 50.0;
    srand(time(NULL));
    for(i = 0; i < n; i++) {
        suhu = rand() % 100;
        getaran = rand() % 100;
        float avg = (suhu + getaran) / 2;
        total += avg;
        printf("Reading %d: Suhu=%.2f, Getaran=%.2f, Rata-rata=%.2f\n", i+1,
suhu, getaran, avg);
    }
    float overall = total / n;
    printf("Overall average: %.2f\n", overall);
    if(overall > threshold)
        printf("Mesin tidak optimal\n");
    else
        printf("Mesin optimal\n");
    return 0;
}

```

- **Python:**

```python
import random

n = 5
threshold = 50.0
total = 0
for i in range(n):
    suhu = random.randint(0, 99)
    getaran = random.randint(0, 99)
    avg = (suhu + getaran) / 2
    total += avg
    print(f"Reading {i+1}: Suhu={suhu}, Getaran={getaran}, Rata-rata={avg:.2f}")
overall = total / n
print(f"Overall average: {overall:.2f}")
if overall > threshold:
    print("Mesin tidak optimal")
else:
    print("Mesin optimal")

```

- **JavaScript:**

```javascript
function simulateSensor() {
    let n = 5;
    let threshold = 50.0;
    let total = 0;
    for (let i = 0; i < n; i++) {
        let suhu = Math.floor(Math.random() * 100);
        let getaran = Math.floor(Math.random() * 100);
        let avg = (suhu + getaran) / 2;
        total += avg;
        console.log(`Reading ${i+1}: Suhu=${suhu}, Getaran=${getaran}, Rata-rata=${avg.toFixed(2)}`);
    }
    let overall = total / n;
    console.log(`Overall average: ${overall.toFixed(2)}`);
    if (overall > threshold)
        console.log("Mesin tidak optimal");
    else
        console.log("Mesin optimal");
}
simulateSensor();

```

Studi Kasus 2: Simulasi Pengendalian Robot Lengan untuk Pengelasan

- **Latar Belakang:**
  Robot lengan dalam pengelasan harus mengikuti pergerakan yang presisi. Siswa akan belajar tentang pengendalian posisi dan penggunaan perulangan untuk mencapai target.
- **Konsep yang Diterapkan:**
  - Struktur perulangan untuk menggerakkan robot.
  - Penggunaan kondisi untuk mengevaluasi posisi.
- **Implementasi:**

- **C:**

```c
#include <stdio.h>
int main() {
    int targetX = 10, targetY = 5;
    int currentX = 0, currentY = 0;
    printf("Posisi awal: (%d, %d)\n", currentX, currentY);
    while(currentX != targetX || currentY != targetY) {
        if(currentX < targetX) currentX++;
        if(currentY < targetY) currentY++;
        printf("Bergerak ke (%d, %d)\n", currentX, currentY);
    }
    printf("Target tercapai: (%d, %d)\n", currentX, currentY);
    return 0;
}

```

- **Python:**

```python
targetX, targetY = 10, 5
currentX, currentY = 0, 0
print(f"Posisi awal: ({currentX}, {currentY})")
while currentX != targetX or currentY != targetY:
    if currentX < targetX:
        currentX += 1
    if currentY < targetY:
        currentY += 1
    print(f"Bergerak ke ({currentX}, {currentY})")
print(f"Target tercapai: ({currentX}, {currentY})")

```

- **JavaScript:**

```javascript
function simulateRobot() {
    let targetX = 10, targetY = 5;
    let currentX = 0, currentY = 0;
    console.log(`Posisi awal: (${currentX}, ${currentY})`);
    while (currentX !== targetX || currentY !== targetY) {
        if (currentX < targetX) currentX++;
        if (currentY < targetY) currentY++;
        console.log(`Bergerak ke (${currentX}, ${currentY})`);
    }
    console.log(`Target tercapai: (${currentX}, ${currentY})`);
}
simulateRobot();

```

Kelompok 2: Energi dan Pertambangan

Studi Kasus 1: Prediksi Konsumsi Energi di Pabrik

- **Latar Belakang:**
  Dengan memprediksi konsumsi energi, pabrik dapat mengoptimalkan penggunaan listrik dan mencegah pemborosan.
- **Konsep yang Diterapkan:**
  - Pengumpulan data, perhitungan rata-rata, dan evaluasi dengan kondisi.

- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main() {
    int n = 5;
    float consumption[] = {120.5, 130.0, 125.5, 128.0, 135.0};
    float total = 0;
    for (int i = 0; i < n; i++) {
        total += consumption[i];
    }
    float average = total / n;
    printf("Prediksi konsumsi energi: %.2f kWh\n", average);
    if (average > 130.0)
        printf("Optimasi diperlukan: Kurangi beban mesin.\n");
    else
        printf("Konsumsi optimal.\n");
    return 0;
}

```

  - **Python:**

```python
consumption = [120.5, 130.0, 125.5, 128.0, 135.0]
average = sum(consumption) / len(consumption)
print(f"Prediksi konsumsi energi: {average:.2f} kWh")
if average > 130.0:
    print("Optimasi diperlukan: Kurangi beban mesin.")
else:
    print("Konsumsi optimal.")

```

  - **JavaScript:**

```javascript
function predictEnergy() {
    let consumption = [120.5, 130.0, 125.5, 128.0, 135.0];
    let total = consumption.reduce((acc, val) => acc + val, 0);
    let average = total / consumption.length;
    console.log(`Prediksi konsumsi energi: ${average.toFixed(2)} kWh`);
    if (average > 130.0)
        console.log("Optimasi diperlukan: Kurangi beban mesin.");
    else
        console.log("Konsumsi optimal.");
}
predictEnergy();

```

Studi Kasus 2: Penghitungan Efisiensi Penggunaan Bahan Bakar

- **Latar Belakang:**
  Menghitung efisiensi bahan bakar sangat penting untuk mengontrol biaya operasional alat berat di pertambangan.
- **Konsep yang Diterapkan:**
  - Perhitungan efisiensi (rasio bahan bakar terhadap jarak).
- **Implementasi:**

- **C:**

```c
#include <stdio.h>
int main() {
    float total_bahan_bakar = 500; // liter
    float total_km = 250; // km
    float efficiency = total_bahan_bakar / total_km;
    printf("Efisiensi: %.2f L/km\n", efficiency);
    float standard = 2.5;
    if (efficiency > standard)
        printf("Efisiensi rendah\n");
    else
        printf("Efisiensi optimal\n");
    return 0;
}

```

- **Python:**

```python
total_bahan_bakar = 500  # liter
total_km = 250  # km
efficiency = total_bahan_bakar / total_km
print(f"Efisiensi: {efficiency:.2f} L/km")
standard = 2.5
if efficiency > standard:
    print("Efisiensi rendah")
else:
    print("Efisiensi optimal")

```

- **JavaScript:**

```javascript
function calculateEfficiency() {
    let total_bahan_bakar = 500; // liter
    let total_km = 250; // km
    let efficiency = total_bahan_bakar / total_km;
    console.log(`Efisiensi: ${efficiency.toFixed(2)} L/km`);
    let standard = 2.5;
    if (efficiency > standard)
        console.log("Efisiensi rendah");
    else
        console.log("Efisiensi optimal");
}
calculateEfficiency();

```

Kelompok 3: Teknologi Informasi dan Komunikasi

Studi Kasus 1: Monitoring Jaringan Sederhana

- **Latar Belakang:**
  Monitoring jaringan membantu mendeteksi gangguan secara cepat agar dapat dilakukan tindakan perbaikan.
- **Konsep yang Diterapkan:**
  - Penggunaan fungsi acak untuk simulasi (ping test) dan kondisi untuk evaluasi.
- **Implementasi:**

- **C:**

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main() {
    srand(time(NULL));
    int ping = rand() % 2; // 0 = gagal, 1 = sukses
    if(ping == 0)
        printf("Ping gagal. Mengirim notifikasi...\n");
    else
        printf("Jaringan stabil.\n");
    return 0;
}

```

- **Python:**

```python
import random
ping = random.choice([True, False])
if not ping:
    print("Ping gagal. Mengirim notifikasi...")
else:
    print("Jaringan stabil.")

```

- **JavaScript:**

```javascript
function monitorNetwork() {
    let ping = Math.random() < 0.5;
    if (!ping)
        console.log("Ping gagal. Mengirim notifikasi...");
    else
        console.log("Jaringan stabil.");
}
monitorNetwork();

```

Studi Kasus 2: Chatbot Sederhana untuk Dukungan Pelanggan

- **Latar Belakang:**
  Chatbot dapat membantu mengurangi beban layanan pelanggan dengan merespons pertanyaan umum.
- **Konsep yang Diterapkan:**
  - Pencocokan string sederhana untuk memberikan jawaban berdasarkan input.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
#include <string.h>
int main() {
    char pertanyaan[100];
    printf("Masukkan pertanyaan: ");
    fgets(pertanyaan, 100, stdin);
    pertanyaan[strcspn(pertanyaan, "\n")] = 0;
```

```c
    if(strcmp(pertanyaan, "Apa jam buka toko?") == 0)
        printf("Toko buka dari jam 08:00 hingga 17:00.\n");
    else
        printf("Maaf, pertanyaan tidak dikenal.\n");
    return 0;
}

```

- **Python:**

```python
pertanyaan = input("Masukkan pertanyaan: ")
if pertanyaan.lower() == "apa jam buka toko?":
    print("Toko buka dari jam 08:00 hingga 17:00.")
else:
    print("Maaf, pertanyaan tidak dikenal.")

```

- **JavaScript:**

```javascript
function chatbot() {
    let pertanyaan = prompt("Masukkan pertanyaan:");
    if(pertanyaan.toLowerCase() === "apa jam buka toko?")
        alert("Toko buka dari jam 08:00 hingga 17:00.");
    else
        alert("Maaf, pertanyaan tidak dikenal.");
}
chatbot();

```

Kelompok 4: Kesehatan dan Pekerjaan Sosial

Studi Kasus 1: Aplikasi Pemantauan Kesehatan

- **Latar Belakang:**
  Aplikasi pemantauan kesehatan dapat membantu mendeteksi kondisi abnormal secara dini.
- **Konsep yang Diterapkan:**
  - Pengumpulan data dan evaluasi kondisi menggunakan pernyataan if.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main() {
    int detak_jantung, tekanan_darah;
    printf("Masukkan detak jantung: ");
    scanf("%d", &detak_jantung);
    printf("Masukkan tekanan darah: ");
    scanf("%d", &tekanan_darah);
    int batas_detak = 100;
    int batas_tekanan = 140;
    if(detak_jantung > batas_detak || tekanan_darah > batas_tekanan)
        printf("Periksa kesehatan segera.\n");
    else
        printf("Kondisi normal.\n");
    return 0;
}
```

```
16.
```

- **Python:**

```python
detak_jantung = int(input("Masukkan detak jantung: "))
tekanan_darah = int(input("Masukkan tekanan darah: "))
batas_detak = 100
batas_tekanan = 140
if detak_jantung > batas_detak or tekanan_darah > batas_tekanan:
    print("Periksa kesehatan segera.")
else:
    print("Kondisi normal.")

```

- **JavaScript:**

```javascript
function monitorKesehatan() {
    let detak = parseInt(prompt("Masukkan detak jantung: "));
    let tekanan = parseInt(prompt("Masukkan tekanan darah: "));
    let batasDetak = 100, batasTekanan = 140;
    if (detak > batasDetak || tekanan > batasTekanan)
        alert("Periksa kesehatan segera.");
    else
        alert("Kondisi normal.");
}
monitorKesehatan();

```

Studi Kasus 2: Sistem Rekomendasi Nutrisi

- **Latar Belakang:**
  Sistem rekomendasi nutrisi membantu memberikan saran menu sehat berdasarkan kondisi kesehatan dan preferensi.
- **Konsep yang Diterapkan:**
  - Pencocokan string dan keputusan berdasarkan input.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
#include <string.h>
int main() {
    char preferensi[50];
    printf("Masukkan preferensi makanan (misal: sayur, daging): ");
    fgets(preferensi, 50, stdin);
    preferensi[strcspn(preferensi, "\n")] = 0;
    if(strcmp(preferensi, "sayur") == 0)
        printf("Rekomendasi: Salad hijau dan sup sayur.\n");
    else if(strcmp(preferensi, "daging") == 0)
        printf("Rekomendasi: Steak dan sayur panggang.\n");
    else
        printf("Rekomendasi: Menu seimbang (karbohidrat, protein, sayur).\n");
    return 0;
}

```

  - **Python:**

```python
preferensi = input("Masukkan preferensi makanan (misal: sayur, daging): ").lower()
if preferensi == "sayur":
    print("Rekomendasi: Salad hijau dan sup sayur.")
elif preferensi == "daging":
    print("Rekomendasi: Steak dan sayur panggang.")
else:
    print("Rekomendasi: Menu seimbang (karbohidrat, protein, sayur).")

```

- **JavaScript:**

```javascript
function rekomendasiNutrisi() {
    let preferensi = prompt("Masukkan preferensi makanan (misal: sayur, daging): ").toLowerCase();
    if (preferensi === "sayur")
        alert("Rekomendasi: Salad hijau dan sup sayur.");
    else if (preferensi === "daging")
        alert("Rekomendasi: Steak dan sayur panggang.");
    else
        alert("Rekomendasi: Menu seimbang (karbohidrat, protein, sayur).");
}
rekomendasiNutrisi();

```

Kelompok 5: Agribisnis dan Agroteknologi

Studi Kasus 1: Prediksi Hasil Panen

- **Latar Belakang:**
  Prediksi hasil panen membantu petani dalam perencanaan dan optimasi sumber daya.
- **Konsep yang Diterapkan:**
  - Pengumpulan data historis dan perhitungan rata-rata.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main(){
    int n = 5;
    float produksi[] = {10.0, 12.0, 11.5, 13.0, 12.5};
    float total = 0;
    for (int i = 0; i < n; i++){
        total += produksi[i];
    }
    float rata = total / n;
    printf("Prediksi hasil panen: %.2f ton\n", rata);
    if(rata < 11.0)
        printf("Rekomendasi: Perbaiki sistem irigasi.\n");
    else
        printf("Hasil panen optimal.\n");
    return 0;
}

```

  - **Python:**

```
produksi = [10.0, 12.0, 11.5, 13.0, 12.5]
rata = sum(produksi) / len(produksi)
print(f"Prediksi hasil panen: {rata:.2f} ton")
if rata < 11.0:
    print("Rekomendasi: Perbaiki sistem irigasi.")
else:
    print("Hasil panen optimal.")

```

- **JavaScript:**

```
function prediksiPanen() {
    let produksi = [10.0, 12.0, 11.5, 13.0, 12.5];
    let total = produksi.reduce((acc, val) => acc + val, 0);
    let rata = total / produksi.length;
    console.log(`Prediksi hasil panen: ${rata.toFixed(2)} ton`);
    if (rata < 11.0)
        console.log("Rekomendasi: Perbaiki sistem irigasi.");
    else
        console.log("Hasil panen optimal.");
}
prediksiPanen();

```

Studi Kasus 2: Aplikasi Manajemen Irigasi

- **Latar Belakang:**
  Manajemen irigasi yang tepat dapat menghemat air dan meningkatkan hasil panen.
- **Konsep yang Diterapkan:**
  - Evaluasi kelembapan tanah dan penggunaan kondisi untuk menentukan jadwal irigasi.
- **Implementasi:**
  - **C:**

```
#include <stdio.h>
int main(){
    float kelembapan = 30.0; // persen
    int cuaca = 1; // 1 = mendukung
    float threshold = 40.0;
    if(kelembapan < threshold && cuaca == 1)
        printf("Lakukan irigasi\n");
    else
        printf("Irigasi tidak diperlukan\n");
    return 0;
}

```

  - **Python:**

```
kelembapan = 30.0  # persen
cuaca = True       # True = mendukung
threshold = 40.0
if kelembapan < threshold and cuaca:
    print("Lakukan irigasi")
else:
    print("Irigasi tidak diperlukan")
```

```
8.
```

- **JavaScript:**

```
1. function manajemenIrigasi() {
2.     let kelembapan = 30.0;
3.     let cuaca = true;
4.     let threshold = 40.0;
5.     if (kelembapan < threshold && cuaca)
6.         console.log("Lakukan irigasi");
7.     else
8.         console.log("Irigasi tidak diperlukan");
9. }
10. manajemenIrigasi();
11.
```

Kelompok 6: Kemaritiman

Studi Kasus 1: Aplikasi Navigasi Kapal

- **Latar Belakang:**
  Navigasi kapal yang optimal penting untuk keselamatan dan efisiensi operasional.
- **Konsep yang Diterapkan:**
  - Algoritma pencarian jalur sederhana dan evaluasi kondisi.
- **Implementasi:**
  - **C:**

```
1. #include <stdio.h>
2. int main(){
3.     int rute1 = 120, rute2 = 100;
4.     if(rute1 < rute2)
5.         printf("Rute terbaik: Rute 1 dengan jarak %d km\n", rute1);
6.     else
7.         printf("Rute terbaik: Rute 2 dengan jarak %d km\n", rute2);
8.     return 0;
9. }
10.
```

  - **Python:**

```
1. rute1 = 120
2. rute2 = 100
3. if rute1 < rute2:
4.     print(f"Rute terbaik: Rute 1 dengan jarak {rute1} km")
5. else:
6.     print(f"Rute terbaik: Rute 2 dengan jarak {rute2} km")
7.
```

- **JavaScript:**

```
1. function navigasiKapal() {
2.     let rute1 = 120, rute2 = 100;
3.     if (rute1 < rute2)
4.         console.log(`Rute terbaik: Rute 1 dengan jarak ${rute1} km`);
5.     else
6.         console.log(`Rute terbaik: Rute 2 dengan jarak ${rute2} km`);
```

```
}
navigasiKapal();

```

Studi Kasus 2: Monitoring Cuaca Laut untuk Peringatan Badai

- **Latar Belakang:**
  Sistem monitoring cuaca laut sangat penting untuk peringatan dini dalam navigasi kapal.
- **Konsep yang Diterapkan:**
  - Evaluasi kondisi cuaca dengan menggunakan ambang batas.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main(){
    int cuaca = 80; // skala 0-100
    int ambang = 70;
    if(cuaca > ambang)
        printf("Peringatan: Badai\n");
    else
        printf("Kondisi laut stabil\n");
    return 0;
}

```

  - **Python:**

```python
cuaca = 80  # skala 0-100
ambang = 70
if cuaca > ambang:
    print("Peringatan: Badai")
else:
    print("Kondisi laut stabil")

```

  - **JavaScript:**

```javascript
function monitoringCuacaLaut() {
    let cuaca = 80;
    let ambang = 70;
    if (cuaca > ambang)
        console.log("Peringatan: Badai");
    else
        console.log("Kondisi laut stabil");
}
monitoringCuacaLaut();

```

Kelompok 7: Bisnis dan Manajemen

Studi Kasus 1: Analisis Penjualan untuk Identifikasi Tren

- **Latar Belakang:**
  Analisis penjualan dapat membantu perusahaan menentukan strategi pemasaran dan stok.
- **Konsep yang Diterapkan:**
  - Perhitungan rata-rata dan analisis tren.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main(){
    int n = 5;
    float sales[] = {150, 200, 180, 220, 210};
    float total = 0;
    for (int i = 0; i < n; i++){
        total += sales[i];
    }
    float avg = total / n;
    printf("Rata-rata penjualan: %.2f\n", avg);
    if(avg < 180)
        printf("Rekomendasi: Tingkatkan promosi.\n");
    else
        printf("Rekomendasi: Penjualan stabil.\n");
    return 0;
}

```

  - **Python:**

```python
sales = [150, 200, 180, 220, 210]
avg = sum(sales) / len(sales)
print(f"Rata-rata penjualan: {avg:.2f}")
if avg < 180:
    print("Rekomendasi: Tingkatkan promosi.")
else:
    print("Rekomendasi: Penjualan stabil.")

```

  - **JavaScript:**

```javascript
function analisisPenjualan() {
    let sales = [150, 200, 180, 220, 210];
    let total = sales.reduce((acc, val) => acc + val, 0);
    let avg = total / sales.length;
    console.log(`Rata-rata penjualan: ${avg.toFixed(2)}`);
    if (avg < 180)
        console.log("Rekomendasi: Tingkatkan promosi.");
    else
        console.log("Rekomendasi: Penjualan stabil.");
}
analisisPenjualan();

```

Studi Kasus 2: Sistem Manajemen Inventori Otomatis

- **Latar Belakang:**
  Sistem inventori otomatis membantu perusahaan dalam mengelola stok barang dan menghindari kekurangan atau kelebihan stok.
- **Konsep yang Diterapkan:**
  - Pengurangan stok dan kondisi peringatan stok minimal.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main(){
    int stok_barang = 100;
    int penjualan_harian = 20;
    int sisa = stok_barang - penjualan_harian;
    printf("Sisa stok: %d\n", sisa);
    int batas_minimal = 30;
    if(sisa < batas_minimal)
        printf("Stok menipis\n");
    else
        printf("Stok cukup\n");
    return 0;
}

```

  - **Python:**

```python
stok_barang = 100
penjualan_harian = 20
sisa = stok_barang - penjualan_harian
print(f"Sisa stok: {sisa}")
batas_minimal = 30
if sisa < batas_minimal:
    print("Stok menipis")
else:
    print("Stok cukup")

```

  - **JavaScript:**

```javascript
function manajemenInventori() {
    let stok_barang = 100;
    let penjualan_harian = 20;
    let sisa = stok_barang - penjualan_harian;
    console.log(`Sisa stok: ${sisa}`);
    let batas_minimal = 30;
    if (sisa < batas_minimal)
        console.log("Stok menipis");
    else
        console.log("Stok cukup");
}
manajemenInventori();

```

Kelompok 8: Pariwisata

Studi Kasus 1: Aplikasi Rekomendasi Tempat Wisata

- **Latar Belakang:**
  Aplikasi rekomendasi tempat wisata dapat membantu wisatawan menemukan destinasi yang sesuai dengan preferensi mereka.
- **Konsep yang Diterapkan:**
  - Pencocokan preferensi dengan kategori destinasi.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
#include <string.h>
int main(){
    char preferensi[50];
    printf("Masukkan preferensi wisata (alam/sejarah/hiburan): ");
    fgets(preferensi, 50, stdin);
    preferensi[strcspn(preferensi, "\n")] = 0;
    if(strcmp(preferensi, "alam") == 0)
        printf("Rekomendasi: Gunung Bromo\n");
    else if(strcmp(preferensi, "sejarah") == 0)
        printf("Rekomendasi: Candi Borobudur\n");
    else if(strcmp(preferensi, "hiburan") == 0)
        printf("Rekomendasi: Taman Mini Indonesia Indah\n");
    else
        printf("Rekomendasi: Pilih kategori yang tersedia\n");
    return 0;
}

```

  - **Python:**

```python
preferensi = input("Masukkan preferensi wisata (alam/sejarah/hiburan):
").lower()
if preferensi == "alam":
    print("Rekomendasi: Gunung Bromo")
elif preferensi == "sejarah":
    print("Rekomendasi: Candi Borobudur")
elif preferensi == "hiburan":
    print("Rekomendasi: Taman Mini Indonesia Indah")
else:
    print("Rekomendasi: Pilih kategori yang tersedia")

```

  - **JavaScript:**

```javascript
function rekomendasiWisata() {
    let preferensi = prompt("Masukkan preferensi wisata (alam/sejarah/hiburan):
").toLowerCase();
    if(preferensi === "alam")
        alert("Rekomendasi: Gunung Bromo");
    else if(preferensi === "sejarah")
        alert("Rekomendasi: Candi Borobudur");
    else if(preferensi === "hiburan")
        alert("Rekomendasi: Taman Mini Indonesia Indah");
    else
```

```
        alert("Rekomendasi: Pilih kategori yang tersedia");
}
rekomendasiWisata();

```

Studi Kasus 2: Sistem Pemesanan Tiket Otomatis untuk Atraksi Wisata

- **Latar Belakang:**
  Sistem pemesanan tiket otomatis membantu mengelola kapasitas atraksi dan meningkatkan efisiensi layanan.
- **Konsep yang Diterapkan:**
  - Pengurangan kuota tiket secara otomatis dan validasi stok.
- **Implementasi:**
  - **C:**

```c
#include <stdio.h>
int main(){
    int kuota = 50;
    int jumlah;
    printf("Masukkan jumlah tiket yang ingin dipesan: ");
    scanf("%d", &jumlah);
    if(jumlah <= kuota) {
        kuota -= jumlah;
        printf("Pemesanan berhasil. Sisa kuota: %d\n", kuota);
    } else {
        printf("Tiket habis.\n");
    }
    return 0;
}

```

  - **Python:**

```python
kuota = 50
jumlah = int(input("Masukkan jumlah tiket yang ingin dipesan: "))
if jumlah <= kuota:
    kuota -= jumlah
    print(f"Pemesanan berhasil. Sisa kuota: {kuota}")
else:
    print("Tiket habis.")

```

  - **JavaScript:**

```javascript
function pesanTiket() {
    let kuota = 50;
    let jumlah = parseInt(prompt("Masukkan jumlah tiket yang ingin dipesan: "));
    if (jumlah <= kuota) {
        kuota -= jumlah;
        alert(`Pemesanan berhasil. Sisa kuota: ${kuota}`);
    } else {
        alert("Tiket habis.");
    }
}
pesanTiket();
```

Kelompok 9: Seni dan Industri Kreatif

Studi Kasus 1: Aplikasi Pengolahan Gambar Digital (Filter Hitam Putih)

- **Latar Belakang:**
  Pengolahan citra digital dasar dapat digunakan untuk mengaplikasikan efek visual seperti filter hitam putih.
- **Konsep yang Diterapkan:**
  - Pengolahan setiap pixel menggunakan rumus grayscale:
    Gray = 0.3R + 0.59G + 0.11B.
- **Implementasi:**
  - **Python (menggunakan library Pillow):**

```python
from PIL import Image

def convert_to_grayscale(image_path, output_path):
    image = Image.open(image_path)
    grayscale = image.convert("L")
    grayscale.save(output_path)
    print("Gambar berhasil diubah menjadi hitam putih.")

# Contoh penggunaan:
# convert_to_grayscale("input.jpg", "output.jpg")

```

  - **JavaScript (menggunakan Canvas API):**

```javascript
function convertToGrayscale() {
    const canvas = document.getElementById("myCanvas");
    const ctx = canvas.getContext("2d");
    const imgData = ctx.getImageData(0, 0, canvas.width, canvas.height);
    const data = imgData.data;
    for (let i = 0; i < data.length; i += 4) {
        const grayscale = 0.3 * data[i] + 0.59 * data[i+1] + 0.11 * data[i+2];
        data[i] = data[i+1] = data[i+2] = grayscale;
    }
    ctx.putImageData(imgData, 0, 0);
    alert("Gambar berhasil diubah menjadi hitam putih.");
}
// Pastikan ada elemen <canvas id="myCanvas"></canvas> dalam HTML.

```

Studi Kasus 2: Sistem Rekomendasi Musik Berdasarkan Preferensi Pengguna

- **Latar Belakang:**
  Sistem rekomendasi musik membantu pengguna menemukan lagu atau playlist yang sesuai dengan preferensi mereka.

- **Konsep yang Diterapkan:**
    - Pencocokan input pengguna dengan kategori musik dalam database sederhana.
- **Implementasi:**
    - **C:**

```c
#include <stdio.h>
#include <string.h>
int main(){
    char preferensi[50];
    printf("Masukkan preferensi musik (pop, rock, jazz): ");
    fgets(preferensi, 50, stdin);
    preferensi[strcspn(preferensi, "\n")] = 0;
    if(strcmp(preferensi, "pop") == 0)
        printf("Playlist: Lagu pop terbaik.\n");
    else if(strcmp(preferensi, "rock") == 0)
        printf("Playlist: Lagu rock terbaik.\n");
    else if(strcmp(preferensi, "jazz") == 0)
        printf("Playlist: Lagu jazz terbaik.\n");
    else
        printf("Playlist: Mix lagu berbagai genre.\n");
    return 0;
}

```

    - **Python:**

```python
preferensi = input("Masukkan preferensi musik (pop, rock, jazz): ").lower()
if preferensi == "pop":
    print("Playlist: Lagu pop terbaik.")
elif preferensi == "rock":
    print("Playlist: Lagu rock terbaik.")
elif preferensi == "jazz":
    print("Playlist: Lagu jazz terbaik.")
else:
    print("Playlist: Mix lagu berbagai genre.")

```

    - **JavaScript:**

```javascript
function rekomendasiMusik() {
    let preferensi = prompt("Masukkan preferensi musik (pop, rock, jazz):
").toLowerCase();
    if(preferensi === "pop")
        alert("Playlist: Lagu pop terbaik.");
    else if(preferensi === "rock")
        alert("Playlist: Lagu rock terbaik.");
    else if(preferensi === "jazz")
        alert("Playlist: Lagu jazz terbaik.");
    else
        alert("Playlist: Mix lagu berbagai genre.");
}
rekomendasiMusik();

```