MODUL AJAR DEEP LEARNING
MATA PELAJARAN : INFORMATIKA
BAB 7: ALGORITMA DAN PEMROGRAMAN

A. IDENTITAS MODUL

Nama Sekolah	**:**	.....
Nama Penyusun	**:**	.....
Mata Pelajaran	**:**	**Informatika**
Kelas / Fase /Semester	**:**	**VII / D / II (Genap)**
Alokasi Waktu	**:**	**8 JP (4 kali pertemuan)**
Tahun Pelajaran	**:**	**20.. / 20..**

B. IDENTIFIKASI KESIAPAN PESERTA DIDIK

- **Pengetahuan Awal**: Peserta didik adalah pengguna aktif aplikasi dan game, namun sebagian besar belum memiliki pengalaman dalam membuat program atau aplikasi. Mereka memahami instruksi langkah-demi-langkah dalam konteks sehari-hari.
- **Minat**: Peserta didik memiliki minat yang tinggi terhadap game, animasi, dan cerita interaktif. Minat ini menjadi motivasi utama untuk belajar membuat karya digital sendiri.
- **Latar Belakang**: Sebagian peserta didik mungkin pernah mendengar tentang *coding* atau pemrograman, namun belum pernah mencobanya secara langsung. Akses terhadap komputer di luar sekolah bervariasi.
- **Kebutuhan Belajar**:
  - **Visual**: Peserta didik akan sangat terbantu dengan antarmuka pemrograman blok visual (Scratch) yang berwarna dan intuitif.
  - **Auditori**: Penjelasan dari guru tentang logika di balik setiap blok program dan diskusi tentang ide proyek akan sangat membantu.
  - **Kinestetik**: Pengalaman langsung menyusun blok-blok kode, menjalankan program, dan melihat hasilnya secara instan (*trial and error*) adalah kunci utama pembelajaran di bab ini.

C. KARAKTERISTIK MATERI PELAJARAN

- **Jenis Pengetahuan yang Akan Dicapai**:
  - **Konseptual**: Memahami konsep dasar algoritma, program, *event* (kejadian), *sequence* (urutan), dan *loop* (perulangan) dalam konteks pemrograman visual.
  - **Prosedural**: Mampu menggunakan lingkungan pemrograman Scratch untuk membuat akun, memilih *sprite* dan latar, menyusun blok-blok kode untuk membuat animasi atau cerita sederhana, dan menyimpan hasil karya.
- **Relevansi dengan Kehidupan Nyata Peserta Didik**: Materi ini memberdayakan peserta didik dari sekadar konsumen teknologi menjadi pencipta. Mereka belajar bahwa aplikasi dan game yang mereka gunakan dibuat melalui proses logika dan kreativitas yang bisa mereka pelajari.

- **Tingkat Kesulitan**: Sedang. Antarmuka Scratch mudah digunakan, namun menerapkan logika untuk membuat program yang berjalan sesuai keinginan memerlukan pemikiran terstruktur dan kemampuan memecahkan masalah (*debugging*).
- **Struktur Materi**: Materi disusun dari pengenalan konsep, pengenalan alat (Scratch), eksplorasi fitur-fitur dasar, hingga penerapan dalam sebuah proyek kreatif sederhana dan aktivitas *unplugged* untuk memperkuat konsep algoritma.
- **Integrasi Nilai dan Karakter**:
  - **Keimanan dan Ketakwaan terhadap Tuhan Yang Maha Esa, dan Berakhlak Mulia**: Membuat karya yang positif dan tidak mengandung unsur SARA atau perundungan.
  - **Bernalar Kritis**: Mendesain alur program dan mencari solusi (proses *debugging*) ketika program tidak berjalan sesuai harapan.
  - **Kreativitas**: Merancang cerita, animasi, atau game sederhana sesuai imajinasi masing-masing.
  - **Kolaborasi/Bergotong Royong**: Berbagi ide dan saling membantu memecahkan masalah kode dengan teman.
  - **Kemandirian**: Mencoba dan bereksperimen dengan berbagai blok kode untuk menemukan fungsi-fungsi baru secara mandiri.
  - **Kepedulian**: Menghargai karya orang lain dan memberikan masukan yang membangun.

D. DIMENSI PROFIL LULUSAN

- **Kreativitas**: Menghasilkan gagasan solusi dalam bentuk rancangan atau implementasi karya digital kreatif.
- **Penalaran Kritis**: Menerapkan berpikir komputasional untuk menciptakan solusi secara logis dan sistematis.
- **Kemandirian**: Terampil berkarya dan belajar secara mandiri untuk menghasilkan sebuah program sederhana.
- **Komunikasi**: Mampu menjelaskan cara kerja program yang dibuatnya kepada teman atau guru.

DESAIN PEMBELAJARAN

A. CAPAIAN PEMBELAJARAN (CP)

Pada akhir Fase D, peserta didik mampu mengembangkan solusi dari berbagai persoalan dengan ... menulis teks algoritmik terstruktur ... menjadi kumpulan instruksi yang dapat dikerjakan ... komputer, berdasarkan paradigma pemrograman prosedural dengan ukuran dan kompleksitas program yang menaik secara bertahap dan berjenjang, dapat dikerjakan secara mandiri atau berkolaborasi dengan yang lain.

B. LINTAS DISIPLIN ILMU

- **Bahasa Indonesia**: Merancang narasi atau dialog untuk cerita interaktif di Scratch.
- **Seni Budaya (Seni Rupa)**: Mendesain karakter (*sprite*) dan latar belakang (*backdrop*) sendiri.
- **Matematika**: Menggunakan konsep koordinat (X dan Y) untuk mengatur posisi dan pergerakan *sprite*.

C. TUJUAN PEMBELAJARAN

- **Pertemuan 1**: Peserta didik mampu menjelaskan konsep dasar pemrograman dan mengenali antarmuka serta fungsi-fungsi utama dalam lingkungan Scratch. (2 JP)
- **Pertemuan 2**: Peserta didik mampu menggunakan blok-blok dasar (Gerak, Tampilan, Suara) untuk membuat animasi sederhana. (2 JP)
- **Pertemuan 3**: Peserta didik mampu menerapkan blok *event* (kejadian) dan kontrol (perulangan) untuk membuat program yang lebih interaktif. (2 JP)
- **Pertemuan 4**: Peserta didik mampu menerapkan konsep algoritma dalam aktivitas *unplugged* (Robot Manual) dan menyelesaikan proyek mini kreatif. (2 JP)

D. TOPIK PEMBELAJARAN KONTEKSTUAL

"Hidupkan Imajinasimu!". Peserta didik akan belajar bagaimana mengubah ide cerita atau animasi dari pikiran mereka menjadi sebuah karya digital nyata menggunakan kode.

E. KERANGKA PEMBELAJARAN

PRAKTIK PEDAGOGIK

- **Model Pembelajaran**: Pembelajaran Berbasis Proyek (*Project-Based Learning*), *Discovery Learning*.
- **Pendekatan**: *Deep Learning (Mindful, Meaningful, Joyful Learning)*
    - **Mindful Learning**: Peserta didik belajar fokus dan teliti dalam menyusun blok-blok kode, serta sabar dalam proses *debugging*.
    - **Meaningful Learning**: Peserta didik merasakan kepuasan dan makna saat ide mereka terwujud menjadi sebuah program yang bisa dijalankan dan

ditunjukkan kepada orang lain.
    - **Joyful Learning**: Proses belajar menjadi sangat menyenangkan karena sifat Scratch yang visual, interaktif, dan memungkinkan eksperimen tanpa batas.
- **Metode Pembelajaran**: Demonstrasi, praktik langsung, eksplorasi, kerja individu/berpasangan, permainan peran (*unplugged*).
- **Strategi Pembelajaran Berdiferensiasi**:
    - **Diferensiasi Konten**: Menyediakan "kartu tantangan" (*challenge cards*) dengan tingkat kesulitan berbeda. Contoh: "Buat sprite-mu menari" (dasar), "Buat dua sprite berbicara" (menengah), "Buat game kejar-kejaran sederhana" (lanjutan).
    - **Diferensiasi Proses**: Peserta didik dapat bekerja secara individu atau berpasangan. Guru memberikan bimbingan lebih bagi yang kesulitan dan memberikan kebebasan eksplorasi bagi yang sudah mahir.
    - **Diferensiasi Produk**: Proyek akhir bisa berupa animasi sederhana, cerita interaktif, atau game simpel, sesuai dengan kemampuan dan minat peserta didik.

KEMITRAAN PEMBELAJARAN

- **Lingkungan Sekolah**: Memamerkan hasil karya Scratch terbaik di media sosial atau mading sekolah untuk mengapresiasi dan memotivasi peserta didik.
- **Mitra Digital**: Menggunakan studio kelas di situs web Scratch untuk berbagi proyek dan saling memberikan umpan balik.

LINGKUNGAN BELAJAR

- **Ruang Fisik**: Laboratorium komputer dengan koneksi internet untuk mengakses Scratch online. Pengaturan tempat duduk memungkinkan guru untuk berkeliling dan melihat layar peserta didik.
- **Ruang Virtual**: Studio kelas Scratch untuk kolaborasi. Google Classroom untuk instruksi dan pengumpulan link proyek.
- **Budaya Belajar**: Mendorong budaya berbagi, mencoba, gagal, dan mencoba lagi. Menekankan bahwa *bug* (kesalahan) dalam program adalah hal yang wajar dan merupakan bagian dari proses belajar.

F. LANGKAH-LANGKAH PEMBELAJARAN BERDIFERENSIASI

PERTEMUAN 1 (2 JP : 80 MENIT)

Topik : Kenalan dengan Pemrograman dan Scratch

- **KEGIATAN PENDAHULUAN (10 Menit)**: Guru bertanya, "Bagaimana sebuah game tahu apa yang harus dilakukan saat kita menekan tombol?". Guru menjelaskan bahwa itu semua diatur oleh program, dan hari ini kita akan belajar menjadi seorang *programmer*.
- **KEGIATAN INTI (60 Menit)**:
    - **Eksplorasi (Meaningful Learning)**: Guru menjelaskan mengapa belajar pemrograman itu penting.
    - **Demonstrasi (Joyful Learning)**: Guru membuka situs Scratch, menunjukkan contoh proyek yang menarik, dan menjelaskan bagian-bagian utama: *Stage*, *Sprite List*, *Block Palette*, dan *Script Area*.

    - **Praktik Terbimbing**: Peserta didik dibimbing untuk membuat akun Scratch, lalu mencoba menarik beberapa blok dari palet ke area skrip dan mengkliknya untuk melihat apa yang terjadi pada *sprite* kucing.
- **KEGIATAN PENUTUP (10 Menit)**: Refleksi ("Bagian mana dari Scratch yang paling membuatmu penasaran?"), rangkuman, dan penutup.

PERTEMUAN 2 (2 JP : 80 MENIT)
Topik : Membuat Animasiku yang Pertama

- **KEGIATAN PENDAHULUAN (10 Menit)**: Guru menunjukkan animasi singkat dan bertanya, "Menurut kalian, instruksi apa saja yang dibutuhkan untuk membuat animasi ini?".
- **KEGIATAN INTI (60 Menit)**:
    - **Demonstrasi**: Guru mendemonstrasikan cara menggunakan blok dari kategori **Gerak** (maju, putar), **Tampilan** (ganti kostum, katakan sesuatu), dan **Suara** (mainkan suara).
    - **Praktik Terbimbing (Joyful Learning)**: Peserta didik mengikuti tutorial untuk membuat *sprite* kucing berjalan bolak-balik di layar, berganti kostum seolah-olah sedang berjalan, dan mengeluarkan suara "Meow".
    - **Eksplorasi Mandiri**: Peserta didik diberi waktu untuk mengganti *sprite* dan latar belakang, serta mencoba blok-blok lain dari 3 kategori tersebut.
- **KEGIATAN PENUTUP (10 Menit)**: Beberapa peserta didik menunjukkan hasil animasinya. Refleksi ("Blok apa yang paling seru untuk dicoba?").

PERTEMUAN 3 (2 JP : 80 MENIT)
Topik : Membuat Program Interaktif

- **KEGIATAN PENDAHULUAN (10 Menit)**: Guru bertanya, "Mengapa saat kita klik bendera hijau di Scratch, programnya baru berjalan? Apa yang terjadi di sana?".
- **KEGIATAN INTI (60 Menit)**:
    - **Demonstrasi (Mindful Learning)**: Guru memperkenalkan blok dari kategori **Kejadian** (ketika bendera hijau diklik, ketika sprite ini diklik) dan **Kontrol** (ulangi, selamanya). Guru menjelaskan pentingnya blok kejadian sebagai pemicu program.
    - **Praktik Terbimbing**: Peserta didik memodifikasi program dari pertemuan sebelumnya dengan menambahkan blok "ketika bendera hijau diklik" di atasnya. Kemudian mencoba menggunakan blok "ulangi selamanya" agar animasi berjalan terus-menerus.
    - **Tantangan**: Peserta didik ditantang untuk membuat program di mana *sprite* akan melakukan sesuatu (misal: melompat dan bersuara) ketika di-klik.
- **KEGIATAN PENUTUP (10 Menit)**: Diskusi tentang perbedaan antara program yang berjalan sekali dan program yang berjalan terus-menerus (menggunakan *loop*).

PERTEMUAN 4 (2 JP : 80 MENIT)
Topik : Algoritma di Dunia Nyata dan Proyek Mini

- **KEGIATAN PENDAHULUAN (5 Menit)**: Guru mengingatkan bahwa semua

program yang dibuat adalah kumpulan instruksi atau algoritma.

- **KEGIATAN INTI (65 Menit)**:
  - **Aktivitas Unplugged (Joyful & Kinestetik Learning)**: Peserta didik bermain "Robot Manual". Satu siswa menjadi "robot" dan siswa lain menjadi "programmer" yang memberikan instruksi verbal yang sangat spesifik (maju 1 langkah, putar kanan 90 derajat, angkat tangan) untuk memandu robot mengambil sebuah objek di dalam kelas.
  - **Proyek Mini Kreatif**: Peserta didik diberi waktu untuk membuat sebuah proyek sederhana di Scratch yang menggabungkan semua yang telah dipelajari. Tema bebas: bisa animasi, kartu ucapan digital, atau cerita pendek.
  - **Diferensiasi Produk**: Peserta didik bebas menentukan kompleksitas proyeknya sesuai kemampuan.
- **KEGIATAN PENUTUP (10 Menit)**: "Gallery Walk". Peserta didik berkeliling melihat karya teman-temannya. Guru memberikan apresiasi dan menutup pelajaran.

G. ASESMEN PEMBELAJARAN

ASESMEN DIAGNOSTIK

- **Tanya Jawab**: "Menurut kalian, apa itu 'kode'? Pernahkah kalian melihatnya?".

ASESMEN FORMATIF

- **Observasi**: Mengamati cara peserta didik menyusun blok, memecahkan masalah, dan bereksperimen selama sesi praktik.
- **Produk (Proses)**:
  - Hasil latihan membuat animasi pada pertemuan 2.
  - Hasil tantangan interaktif pada pertemuan 3.
  - Keaktifan dalam permainan "Robot Manual".

ASESMEN SUMATIF

- **Produk (Proyek)**:
  - **Proyek Mini Kreatif**: Menilai proyek Scratch akhir berdasarkan kriteria: kreativitas ide, penggunaan minimal 3 kategori blok yang berbeda, dan program berjalan tanpa eror fatal.
- **Tes Tertulis**: Tes singkat untuk mengukur pemahaman konsep.

  Contoh Tes Tertulis :

  Pilihan Ganda

  1. Dalam Scratch, karakter atau objek yang bisa kita program disebut...
     a. Backdrop
     b. Script
     c. Sprite
     d. Stage
  2. Blok "ketika bendera hijau diklik" termasuk dalam kategori blok...
     a. Gerak
     b. Kontrol
     c. Tampilan

d. Kejadian

Essay

1. Apa yang dimaksud dengan algoritma dalam konteks pemrograman? Berikan contoh algoritma sederhana dalam kehidupan sehari-hari!
2. Jelaskan fungsi dari blok "ulangi selamanya" (*forever loop*) dalam Scratch! Mengapa blok ini sangat berguna untuk membuat animasi atau game?

Mengetahui,	....., ... 20..
Kepala Sekolah	Guru Mata Pelajaran
.....	**.....**
NIP.	NIP. .....