

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 : **IX / D / Genap**
Alokasi Waktu : **20 JP (10 kali pertemuan)**
Tahun Pelajaran : **20.. / 20..**

B. IDENTIFIKASI KESIAPAN PESERTA DIDIK

- **Pengetahuan Awal** : Peserta didik telah mengenal dasar-dasar pemrograman visual menggunakan Scratch dan Blockly, memahami konsep variabel, instruksi sekuensial, percabangan (if-then-else), dan perulangan (loops).
- **Minat** : Sebagian besar peserta didik memiliki minat dalam membuat karya digital kreatif seperti animasi, komik, dan permainan interaktif sederhana.
- **Latar Belakang** : Peserta didik berasal dari latar belakang yang beragam, dengan tingkat paparan yang berbeda terhadap perangkat komputasi dan pemrograman.
- **Kebutuhan Belajar** :
 - **Visual**: Peserta didik belajar melalui demonstrasi langsung pembuatan program, diagram alur, dan presentasi visual.
 - **Auditori**: Peserta didik belajar melalui penjelasan konsep oleh guru, diskusi kelompok, dan presentasi hasil kerja.
 - **Kinestetik**: Peserta didik belajar dengan langsung mempraktikkan pembuatan program (plugged) dan melalui aktivitas simulasi tanpa komputer (unplugged).

C. KARAKTERISTIK MATERI PELAJARAN

- **Jenis Pengetahuan yang Akan Dicapai**
 - **Konseptual**: Memahami konsep modularisasi program (fungsi dan prosedur), parameter, model komputasi, sistem bilangan (biner, oktal, desimal), dan parity bit untuk deteksi kesalahan.
 - **Prosedural**: Mampu merancang dan mengembangkan program visual sederhana, menerapkan aturan translasi konsep dari Scratch ke Blockly, mengimplementasikan fungsi dan prosedur, serta membuat program konverter sistem bilangan.
- **Relevansi dengan Kehidupan Nyata Peserta Didik**: Kemampuan pemrograman dapat digunakan untuk menciptakan solusi digital (aplikasi, game) untuk masalah sehari-hari, memodelkan fenomena sains (seperti pertumbuhan organisme), dan memahami cara kerja teknologi digital di sekitar mereka.
- **Tingkat Kesulitan**: Sedang hingga Tinggi. Materi ini melibatkan pemikiran abstrak (modularisasi, model komputasi) dan logika yang terstruktur.
- **Struktur Materi**:
 1. **Perbandingan Scratch dan Blockly**

2. **Penerapan Algoritma untuk Literasi Numerik**
 3. **Konsep Modularisasi Program (Fungsi dan Prosedur)**
 4. **Penerapan Algoritma untuk Literasi Sains**
 5. **Model Komputasi: Konverter Sistem Bilangan**
 6. **Model Komputasi: Penambahan Parity Bit**
 7. **Proyek Akhir: Menggabungkan Modul Konverter Bilangan**
- **Integrasi Nilai dan Karakter:**
 - **Keimanan dan Ketakwaan terhadap Tuhan Yang Maha Esa, dan Berakhlak Mulia:** Menghargai kemampuan akal yang diberikan Tuhan untuk menciptakan solusi yang bermanfaat bagi sesama.
 - **Bernalar Kritis:** Menganalisis masalah, memecahnya menjadi bagian-bagian kecil (dekomposisi), dan merancang langkah-langkah solusi (algoritma) yang logis dan efisien.
 - **Kreativitas:** Mengembangkan program yang unik, menciptakan solusi alternatif, dan merancang artefak komputasional (poster, program) yang menarik.
 - **Kolaborasi/Bergotong Royong:** Bekerja dalam kelompok untuk menyelesaikan proyek, berdiskusi, dan mempresentasikan hasil karya bersama.
 - **Kemandirian:** Mengerjakan latihan-latihan pemrograman secara individu dan mencari solusi atas tantangan yang dihadapi.
 - **Kepedulian:** Memahami pentingnya keamanan data (melalui konsep parity bit) dan menciptakan program yang andal.

D. DIMENSI PROFIL LULUSAN

- **Keimanan dan Ketakwaan terhadap Tuhan Yang Maha Esa, dan Berakhlak Mulia:** Menggunakan keterampilan informatika untuk tujuan yang baik dan bermanfaat, serta menghargai karya orang lain.
- **Kewargaan:** Memahami dampak teknologi dan berkontribusi secara positif dalam masyarakat digital.
- **Penalaran Kritis:** Menganalisis persoalan komputasi, merancang algoritma yang efisien, dan melakukan debugging (mencari kesalahan) pada program.
- **Kreativitas:** Menghasilkan artefak komputasional yang orisinal dan fungsional, serta menemukan cara-cara baru dalam menyelesaikan masalah.
- **Kolaborasi:** Bekerja sama secara efektif dalam tim untuk merancang, membangun, dan mempresentasikan proyek pemrograman.
- **Kemandirian:** Mampu belajar secara mandiri, mengeksplorasi fitur-fitur baru dalam lingkungan pemrograman, dan menyelesaikan tugas-tugas individu.
- **Kesehatan:** mempraktikkan kebiasaan baik saat menggunakan komputer, seperti posisi duduk yang benar dan istirahat teratur.
- **Komunikasi:** Mampu menjelaskan cara kerja program dan ide-ide di baliknya secara lisan (presentasi) dan tulisan (dokumentasi, poster).

DESAIN PEMBELAJARAN

A. CAPAIAN PEMBELAJARAN (CP)

Pada akhir Fase D, murid memiliki kemampuan sebagai berikut:

- **Berpikir Komputasional**

Menerapkan berpikir komputasional untuk problem dalam kehidupan sehari-hari maupun dalam menghadapi masalah komputasi; memahami konsep himpunan data terstruktur dalam kehidupan sehari-hari; memahami konsep lembar kerja pengolah data; menerapkan berpikir komputasional dalam menyelesaikan persoalan yang mengandung himpunan data berstruktur sederhana dengan volume kecil; serta menuliskan sekumpulan instruksi dengan menggunakan sekumpulan kosakata terbatas atau simbol dalam format pseudocode.

- **Literasi Digital**

Memahami cara kerja dan penggunaan mesin pencari di internet; mengetahui kualitas informasi dan kredibilitas sumber informasi digital; mengenal ekosistem media pers digital; membedakan fakta, opini, dan hoaks; memahami pemanfaatan perangkat teknologi pengolah dokumen, lembar kerja, dan presentasi; mampu mendeskripsikan komponen, fungsi, dan cara kerja komputer; memahami konsep dan penerapan konektivitas jaringan lokal dan internet baik kabel maupun nirkabel; mengetahui jenis ruang publik virtual; memahami pemanfaatan perangkat teknologi digital untuk produksi dan diseminasi konten; memahami pentingnya menjaga rekam jejak digital, mengamalkan toleransi dan empati di dunia digital, memahami dampak perundungan digital, membuat kata sandi yang aman; memahami pengamanan perangkat dari berbagai jenis malware, memilah informasi yang bersifat privat dan publik, melindungi data pribadi dan identitas digital serta memiliki kesadaran penuh (*mindfulness*) dalam dunia digital.

B. LINTAS DISIPLIN ILMU

- **Matematika:** Fungsi kuadrat, persamaan linear, sistem bilangan, logika matematika, perbandingan senilai.
- **Sains (IPA):** Pemodelan dan simulasi pertumbuhan organisme.
- **Bahasa Indonesia:** Menyusun penjelasan dan dokumentasi program yang sistematis.
- **Seni Budaya dan Prakarya (SBdP):** Merancang poster yang informatif dan menarik, membuat kerajinan tangan (aktivitas unplugged).

C. TUJUAN PEMBELAJARAN

- **Pertemuan 1:** Peserta didik mampu membandingkan objek dan perintah dalam lingkungan pemrograman Scratch dan Blockly untuk menyelesaikan masalah sederhana (2 JP).
- **Pertemuan 2:** Peserta didik mampu menerapkan konsep perulangan dan percabangan dalam program visual untuk menyelesaikan masalah literasi numerik (2 JP).
- **Pertemuan 3:** Peserta didik mampu memahami dan mengimplementasikan konsep modularisasi program menggunakan *function* (modul yang mengembalikan nilai) (2 JP).
- **Pertemuan 4:** Peserta didik mampu memahami dan mengimplementasikan konsep modularisasi program menggunakan *procedure* (modul yang tidak mengembalikan nilai)

(2 JP).

- **Pertemuan 5:** Peserta didik mampu menerapkan modularisasi program dan perulangan untuk memodelkan masalah literasi sains (2 JP).
- **Pertemuan 6:** Peserta didik mampu merancang program untuk mengonversi bilangan dari sistem biner dan oktal ke sistem desimal (2 JP).
- **Pertemuan 7:** Peserta didik mampu merancang program untuk mengonversi bilangan dari sistem desimal ke sistem biner dan oktal (2 JP).
- **Pertemuan 8:** Peserta didik mampu memahami konsep *parity bit* dan mengimplementasikannya dalam program konversi bilangan untuk deteksi kesalahan sederhana (2 JP).
- **Pertemuan 9:** Peserta didik mampu mengintegrasikan berbagai modul *function* yang telah dibuat untuk membangun sebuah program konverter sistem bilangan yang utuh (2 JP).
- **Pertemuan 10:** Peserta didik mampu mempresentasikan hasil karya program dan proses pengembangannya dalam bentuk poster digital atau fisik secara kolaboratif (2 JP).

D. TOPIK PEMBELAJARAN KONTEKSTUAL

Pembuatan program untuk menyelesaikan masalah logika, memodelkan fenomena di dunia nyata (pertumbuhan bakteri), memahami cara komputer merepresentasikan data (sistem bilangan), dan memastikan integritas data saat dikirim (parity bit).

E. KERANGKA PEMBELAJARAN

PRAKTIK PEDAGOGIK

- **Model Pembelajaran:** *Project-Based Learning* (Pembelajaran Berbasis Proyek), *Problem-Based Learning* (Pembelajaran Berbasis Masalah).
- **Pendekatan:** Deep Learning (*Mindful, Meaningful, Joyful Learning*)
 - **Mindful Learning:** Peserta didik diajak untuk sadar dan memahami logika di setiap blok program yang mereka susun, bukan hanya sekadar menyalin. Mereka merefleksikan mengapa suatu algoritma bekerja dan bagaimana cara memperbaikinya.
 - **Meaningful Learning:** Proyek yang dikerjakan relevan dengan disiplin ilmu lain (Matematika, Sains) dan kehidupan sehari-hari, sehingga peserta didik memahami manfaat nyata dari pemrograman.
 - **Joyful Learning:** Pembelajaran dilakukan melalui pembuatan karya kreatif (animasi, plotting grafik, poster) dan permainan (aktivitas unplugged), sehingga menciptakan pengalaman belajar yang menyenangkan.
- **Metode Pembelajaran:** Demonstrasi, Praktik, Diskusi Kelompok, Proyek, Presentasi.
- **Strategi Pembelajaran Berdiferensiasi**
 - **Diferensiasi Konten:** Menyediakan tantangan pemrograman dengan tingkat kesulitan yang berbeda (misalnya, konverter biner-desimal sebagai dasar, konverter heksadesimal sebagai pengayaan).
 - **Diferensiasi Proses:** Peserta didik dapat memilih untuk bekerja secara individu, berpasangan, atau dalam kelompok. Menyediakan pilihan aktivitas *plugged* (dengan komputer) dan *unplugged* (tanpa komputer).

- **Diferensiasi Produk:** Hasil akhir dapat berupa program yang berfungsi, poster penjelasan, atau laporan tertulis tentang algoritma yang digunakan.

KEMITRAAN PEMBELAJARAN

- **Lingkungan Sekolah:** Guru mata pelajaran lain (Matematika, IPA) dapat diajak berkolaborasi untuk proyek lintas disiplin.
- **Lingkungan Luar Sekolah/Masyarakat:** (Opsional) Mengundang praktisi (programmer, pengembang game) sebagai guru tamu untuk berbagi pengalaman.
- **Mitra Digital:** Memanfaatkan platform pemrograman online (Blockly, Scratch) dan sumber belajar dari internet.

LINGKUNGAN BELAJAR

Lingkungan pembelajaran yang mengintegrasikan antara ruang fisik, ruang virtual, dan budaya belajar:

- **Ruang Fisik:**
 - Laboratorium komputer dengan penataan tempat duduk yang fleksibel (bisa untuk kerja individu maupun kelompok).
 - Papan tulis atau proyektor untuk demonstrasi dan diskusi.
 - Area untuk pameran hasil karya (poster).
- **Ruang Virtual:**
 - Platform e-learning sekolah untuk pengumpulan tugas dan materi.
 - Lingkungan pemrograman online (Scratch, Blockly).
 - Aplikasi kolaborasi digital (Google Docs/Slides) untuk kerja kelompok.
- **Budaya Belajar:**
 - Mendorong budaya bertanya dan tidak takut salah (*trial and error*).
 - Menghargai proses belajar dan upaya, bukan hanya hasil akhir.
 - Membangun suasana kolaboratif di mana peserta didik saling membantu.

PEMANFAATAN DIGITAL

- **Perpustakaan Digital/Sumber Daring:** Mengakses tutorial dan dokumentasi dari situs seperti developers.google.com/blockly atau scratch.mit.edu.
- **Forum Diskusi Daring:** Menggunakan forum di e-learning untuk bertanya di luar jam pelajaran.
- **Penilaian Daring:** Menggunakan kuis online untuk asesmen diagnostik atau formatif.
- **Media Presentasi Digital:** Peserta didik membuat presentasi hasil proyek menggunakan aplikasi seperti Google Slides atau Canva.
- **Media Publikasi Digital:** Mempublikasikan hasil karya di blog atau media sosial sekolah.

F. LANGKAH-LANGKAH PEMBELAJARAN BERDIFERENSIASI

PERTEMUAN 1 (2 JP : 80 MENIT)

Topik : Perbandingan Scratch dan Blockly

KEGIATAN PENDAHULUAN (15 MENIT)

- **Orientasi:** Guru membuka pelajaran dengan salam dan doa, serta memeriksa kehadiran.
- **Apersepsi:** Guru mengingatkan kembali pengalaman peserta didik menggunakan Scratch di kelas VII dan Blockly di kelas VIII. Guru bertanya, "Apa persamaan dan perbedaan

yang kalian ingat dari kedua aplikasi tersebut?"

- **Motivasi (Joyful):** Guru menunjukkan dua program sederhana yang identik (misal: animasi kucing bergerak) yang dibuat di Scratch dan Blockly, lalu memantik diskusi tentang bagaimana cara membuatnya.
- **Pemberian Acuan:** Guru menyampaikan tujuan pembelajaran, yaitu membandingkan blok-blok perintah di kedua lingkungan untuk memecahkan masalah yang sama.

KEGIATAN INTI (55 MENIT)

- **Eksplorasi (Mindful):** Guru mendemonstrasikan beberapa blok perintah yang serupa tapi tak sama di Scratch dan Blockly (misal: blok repeat vs repeat n times, blok if-then di keduanya).
- **Praktik Terbimbing:** Peserta didik diminta membuat program sederhana untuk menentukan apakah sebuah bilangan adalah bilangan prima. Guru memberikan kerangka algoritma.
- **Praktik Mandiri (Meaningful):** Peserta didik mengimplementasikan algoritma tersebut di lingkungan Blockly. Bagi yang lebih cepat, dapat mencoba mengimplementasikannya juga di Scratch.
- **Diskusi:** Peserta didik mendiskusikan perbedaan yang mereka temukan saat membuat program tersebut (misal: Scratch berbasis *event*, Blockly lebih prosedural).
- **Pembelajaran Berdiferensiasi:**
 - **Proses:** Peserta didik dapat bekerja mandiri atau berpasangan. Bagi yang kesulitan, guru memberikan panduan langkah-demi-langkah. Bagi yang sudah mahir, diberi tantangan tambahan (misal: menampilkan semua bilangan prima dari 1 sampai N).
 - **Produk:** Hasil berupa file program Blockly atau Scratch.

KEGIATAN PENUTUP (10 MENIT)

- **Refleksi:** Guru mengajak peserta didik merefleksikan: "Bagian mana dari perbandingan kedua bahasa ini yang paling menarik?" dan "Kapan kita sebaiknya menggunakan Scratch, dan kapan Blockly?"
- **Rangkuman:** Guru bersama peserta didik menyimpulkan bahwa meskipun bloknya mirip, Scratch lebih cocok untuk karya interaktif berbasis *event*, sementara Blockly (dasar) lebih fokus pada logika prosedural.
- **Tindak Lanjut:** Guru memberikan tugas untuk mengeksplorasi satu blok perintah lain yang berbeda di kedua platform.
- **Penutup:** Salam dan doa.

PERTEMUAN 2 (2 JP : 80 MENIT)

Topik : Penerapan Algoritma untuk Literasi Numerik

KEGIATAN PENDAHULUAN (10 MENIT)

- **Orientasi:** Salam, doa, dan presensi.
- **Apersepsi:** Guru mengaitkan materi sebelumnya dengan bertanya, "Blok apa yang kita gunakan jika ingin melakukan sesuatu berulang kali?" (Loop/Repeat).
- **Motivasi (Meaningful):** Guru menyajikan masalah kontekstual: "Bagaimana cara guru menghitung nilai rata-rata dari seluruh siswa di kelas ini menggunakan program?"

KEGIATAN INTI (60 MENIT)

- **Analisis Masalah:** Secara klasikal, peserta didik dibimbing untuk menganalisis masalah

"Menghitung Rata-Rata N Nilai". Inputnya apa? Outputnya apa? Bagaimana rumusnya?

- **Desain Algoritma (Mindful):** Peserta didik merancang algoritma menggunakan *pseudocode* atau *flowchart* sederhana di buku catatan. Guru menekankan pentingnya variabel untuk jumlah_total dan banyak_data.
- **Implementasi:** Peserta didik mengimplementasikan algoritma tersebut ke dalam program Blockly. Mereka akan menggunakan perulangan untuk meminta masukan nilai sebanyak N kali dan menjumlahkannya.
- **Pengembangan (Joyful):** Setelah berhasil, peserta didik diberi tantangan kedua: "Proyek Hitung Kriteria Ketuntasan Minimum (KKM)". Mereka harus memodifikasi program sebelumnya dengan menambahkan percabangan (if) di dalam perulangan untuk menghitung berapa banyak siswa yang nilainya di bawah KKM.
- **Pembelajaran Berdiferensiasi:**
 - **Proses:** Peserta didik yang kesulitan dengan konsep rata-rata dibimbing secara khusus. Peserta didik yang cepat selesai dapat mencoba menggabungkan kedua program (menampilkan rata-rata dan jumlah siswa tidak tuntas sekaligus).
 - **Produk:** File program Blockly yang dapat menghitung rata-rata dan/atau jumlah siswa di bawah KKM.

KEGIATAN PENUTUP (10 MENIT)

- **Refleksi:** Peserta didik diminta merenungkan, "Selain untuk nilai, masalah sehari-hari apa lagi yang bisa diselesaikan dengan konsep perulangan dan percabangan?"
- **Rangkuman:** Guru menyimpulkan bahwa perulangan sangat efisien untuk memproses banyak data, dan percabangan berguna untuk membuat keputusan berdasarkan kondisi tertentu.
- **Tindak Lanjut:** Guru meminta peserta didik memikirkan modifikasi lain untuk program KKM (misal: menampilkan nama siswa yang tidak tuntas).
- **Penutup:** Salam dan doa.

PERTEMUAN 3 (2 JP : 80 MENIT)

Topik : Konsep Modularisasi Program (Function)

KEGIATAN PENDAHULUAN (10 MENIT)

- **Orientasi:** Salam, doa, dan presensi.
- **Apersepsi:** Guru bertanya, "Jika kalian membuat kue dan sering membutuhkan saus cokelat, apakah kalian akan membuat saus dari awal setiap kali, atau membuat satu resep saus yang bisa dipakai kapan saja?"
- **Motivasi (Meaningful):** Guru menjelaskan bahwa dalam pemrograman, "resep" yang bisa dipakai berulang kali itu disebut modul atau fungsi. Ini membuat kode lebih rapi dan efisien.

KEGIATAN INTI (60 MENIT)

- **Eksplorasi (Mindful):** Guru menjelaskan konsep *function* sebagai blok program yang menerima input (parameter), melakukan proses, dan mengembalikan sebuah nilai (return value). Guru mendemonstrasikan blok *to do something return* di Blockly.
- **Praktik Terbimbing:** Peserta didik dibimbing mengerjakan Aktivitas AP-K9-04 (Fungsi Kuadrat). Mereka akan membuat sebuah *function* bernama HitungFX yang menerima parameter a, b, c, x dan mengembalikan hasil perhitungan $ax^2 + bx + c$.

- Praktik Mandiri: Peserta didik memodifikasi program utama untuk memanggil *function* HitungFX dengan nilai-nilai yang berbeda dan menampilkan hasilnya.
- Pembelajaran Berdiferensiasi:
 - Proses: Bagi yang kesulitan, fokus pada pemahaman parameter dan return. Bagi yang cepat, diberi tantangan membuat *function* lain (misal: persamaan linear dari Aktivitas AP-K9-05).
 - Produk: Program Blockly yang terstruktur dengan minimal satu *function* yang mengembalikan nilai.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi: Guru bertanya, "Apa keuntungan utama membuat perhitungan fungsi kuadrat dalam sebuah *function* terpisah?"
- Rangkuman: Guru menyimpulkan bahwa *function* berguna untuk membungkus perhitungan atau logika yang menghasilkan satu nilai output dan bisa digunakan berulang kali.
- Tindak Lanjut: Peserta didik diminta mencari contoh lain di kehidupan sehari-hari yang bisa dimodelkan sebagai *function*.
- Penutup: Salam dan doa.

PERTEMUAN 4 (2 JP : 80 MENIT)

Topik : Konsep Modularisasi Program (Procedure)

KEGIATAN PENDAHULUAN (10 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Apersepsi: Guru mereview konsep *function* dari pertemuan sebelumnya. Lalu memberikan analogi: "Kemarin kita membuat 'resep' yang menghasilkan sesuatu (saus coklat). Bagaimana jika kita hanya ingin melakukan sebuah perintah, seperti 'bersihkan meja', yang tidak menghasilkan apa-apa?"
- Motivasi (Joyful): Guru menunjukkan program plotting grafik dari Aktivitas AP-K9-07 dan bertanya, "Bagaimana cara kita memerintahkan program untuk 'menggambar sebuah titik' di layar secara berulang?"

KEGIATAN INTI (60 MENIT)

- Eksplorasi (Mindful): Guru menjelaskan konsep *procedure* sebagai blok program yang melakukan serangkaian perintah tapi tidak mengembalikan nilai. Guru mendemonstrasikan blok *to do something* di Blockly.
- Praktik Terbimbing: Peserta didik dibimbing mengerjakan Aktivitas AP-K9-07 (Fungsi Kuadrat dan Plotting). Mereka akan membuat *procedure* bernama *gambarTitik* yang menerima parameter *x* dan *y*, lalu menggambar lingkaran kecil di koordinat tersebut.
- Integrasi: Peserta didik menggabungkan *function* HitungFX dari pertemuan sebelumnya dengan *procedure* *gambarTitik* dalam sebuah perulangan untuk membuat grafik fungsi kuadrat.
- Pembelajaran Berdiferensiasi:
 - Proses: Peserta didik dapat bereksperimen dengan parameter *gambarTitik* (misal: mengubah warna atau ukuran titik).
 - Produk: Program Blockly yang dapat menggambar grafik fungsi kuadrat menggunakan *function* dan *procedure*.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi: Peserta didik mendiskusikan, "Kapan kita menggunakan *function* dan kapan kita menggunakan *procedure*?"
- Rangkuman: Guru menyimpulkan: gunakan *function* jika butuh hasil perhitungan, gunakan *procedure* jika hanya ingin menjalankan serangkaian perintah.
- Tindak Lanjut: Peserta didik diminta memikirkan contoh *procedure* lain (misal: mainkanSuara atau tampilkanPesan).
- Penutup: Salam dan doa.

PERTEMUAN 5 (2 JP : 80 MENIT)

Topik : Penerapan Algoritma untuk Literasi Sains

KEGIATAN PENDAHULUAN (10 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Apersepsi: Guru mengaitkan dengan pelajaran IPA tentang pertumbuhan bakteri. "Bisakah kita membuat simulasi pertumbuhan bakteri menggunakan program komputer?"
- Motivasi (Meaningful): Guru menampilkan data pertumbuhan kasus COVID-19 (seperti di buku) dan menjelaskan bahwa model pertumbuhan serupa bisa diterapkan pada organisme lain.

KEGIATAN INTI (60 MENIT)

- Analisis Masalah: Peserta didik menganalisis masalah dari Aktivitas AP-K9-09: memprediksi jumlah bakteri hari ke-3 berdasarkan data hari ke-1 dan ke-2.
- Praktik Terbimbing: Peserta didik membuat *function* PrediksiHariBerikutnya yang menghitung laju pertumbuhan dan mengembalikan jumlah bakteri di hari selanjutnya.
- Pengembangan (Joyful): Peserta didik memodifikasi program utama untuk memprediksi pertumbuhan selama N hari (Aktivitas AP-K9-10) menggunakan perulangan yang memanggil *function* tersebut berulang kali.
- Praktik Mandiri: Tantangan terakhir adalah mencari tahu berapa hari yang dibutuhkan untuk mencapai jumlah bakteri tertentu (Aktivitas AP-K9-11), yang memerlukan perulangan dengan kondisi (repeat while).
- Pembelajaran Berdiferensiasi:
 - Proses: Peserta didik yang kesulitan dibantu untuk fokus pada logika laju pertumbuhan. Peserta didik yang cepat selesai dapat menambahkan visualisasi sederhana (misal: mencetak jumlah bakteri setiap hari).
 - Produk: Program simulasi pertumbuhan bakteri yang fungsional.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi: "Apa manfaat dari membuat simulasi seperti ini dibandingkan hanya menghitung manual?"
- Rangkuman: Guru menyimpulkan bahwa pemrograman adalah alat yang ampuh untuk memodelkan dan memahami fenomena sains.
- Tindak Lanjut: Peserta didik diminta mencari contoh fenomena sains lain yang bisa disimulasikan.
- Penutup: Salam dan doa.

PERTEMUAN 6 (2 JP : 80 MENIT)

Topik : Konverter Sistem Bilangan (Biner/Oktal ke Desimal)

KEGIATAN PENDAHULUAN (10 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Apersepsi: Guru bertanya, "Komputer hanya mengerti angka 0 dan 1. Bagaimana caranya angka 1011 bisa dimengerti oleh kita sebagai angka 11?"
- Motivasi (Mindful): Guru menjelaskan pentingnya konversi bilangan agar manusia dan komputer bisa "berbicara" dalam bahasa yang sama.

KEGIATAN INTI (60 MENIT)

- Review Konsep: Guru mereviu cara konversi manual dari biner ke desimal (perkalian dengan pangkat 2).
- Dekomposisi Algoritma: Guru membimbing peserta didik memecah algoritma konversi (Aktivitas AP-K9-12):
 1. Bagaimana cara mendapatkan digit terakhir dari sebuah bilangan? (Operator modulo 10).
 2. Bagaimana cara "menghapus" digit terakhir? (Pembagian integer dengan 10).
 3. Bagaimana cara menghitung pangkat 2 yang terus bertambah? (Variabel pangkat yang diinkrementasi).
- Praktik Terbimbing: Peserta didik membangun *function* BinerKeDesimal langkah demi langkah menggunakan logika di atas.
- Praktik Mandiri: Peserta didik memodifikasi *function* tersebut menjadi OktalKeDesimal (Aktivitas AP-K9-13). Mereka akan sadar bahwa yang berubah hanyalah basisnya (dari 2 menjadi 8).
- Pembelajaran Berdiferensiasi:
 - Konten: Bagi yang cepat, diberi tantangan untuk membuat konverter Heksadesimal ke Desimal.
 - Produk: Program dengan *function* konverter yang berfungsi dengan benar.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi: "Bagian mana dari algoritma konversi ini yang paling menantang?"
- Rangkuman: Guru menyimpulkan algoritma umum untuk konversi dari basis-N ke desimal.
- Tindak Lanjut: Guru memberikan soal latihan konversi manual untuk dikerjakan di rumah.
- Penutup: Salam dan doa.

PERTEMUAN 7 (2 JP : 80 MENIT)

Topik : Konverter Sistem Bilangan (Desimal ke Biner/Oktal)

KEGIATAN PENDAHULUAN (10 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Apersepsi: Guru mereviu pertemuan sebelumnya. "Sekarang, bagaimana jika sebaliknya? Bagaimana mengubah angka 13 menjadi 1101?"
- Motivasi (Mindful): Menunjukkan bahwa algoritma sebaliknya juga diperlukan agar komputer bisa menyimpan input dari manusia.

KEGIATAN INTI (60 MENIT)

- Review Konsep: Guru mereview cara konversi manual dari desimal ke biner (pembagian berulang dengan 2 dan mencatat sisa).
- Dekomposisi Algoritma: Guru membimbing peserta didik memecah algoritma (Aktivitas AP-K9-14):
 1. Bagaimana mendapatkan sisa pembagian? (Operator modulo 2).
 2. Bagaimana memperbarui bilangannya? (Pembagian integer dengan 2).
 3. Bagaimana menyusun bilangan biner dari sisa-sisa pembagian? (Ini bagian tersulit: $\text{hasil} = \text{hasil} + \text{sisa} * (10^{\text{pangkat}})$).
- Praktik Terbimbing: Peserta didik membangun *function* DesimalKeBiner.
- Praktik Mandiri: Peserta didik memodifikasi *function* tersebut menjadi DesimalKeOktal (Aktivitas AP-K9-15).
- Pembelajaran Berdiferensiasi:
 - Proses: Guru memberikan bantuan lebih pada logika penyusunan bilangan hasil konversi.
 - Produk: Program dengan *function* konverter desimal ke biner/oktal.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi: "Menurut kalian, mana yang lebih sulit, mengubah ke desimal atau dari desimal? Mengapa?"
- Rangkuman: Guru menyimpulkan algoritma umum untuk konversi dari desimal ke basis-N.
- Tindak Lanjut: Peserta didik diminta mencoba mengonversi tanggal lahir mereka ke biner.
- Penutup: Salam dan doa.

PERTEMUAN 8 (2 JP : 80 MENIT)

Topik : Model Komputasi: Penambahan Parity Bit

KEGIATAN PENDAHULUAN (15 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Apersepsi: "Pernahkah saat mengunduh file, file-nya rusak atau *corrupt*? Bagaimana komputer tahu kalau ada yang salah?"
- Motivasi (Joyful): Guru memulai dengan permainan "Pengantar Pesan" (Aktivitas AP-K9-16-U). Satu siswa mengirim pesan biner, satu siswa (gangguan) boleh mengubah satu bit, dan penerima harus menebak apakah pesannya benar berdasarkan aturan paritas genap/ganjil.

KEGIATAN INTI (55 MENIT)

- Eksplorasi Konsep: Setelah permainan, guru menjelaskan konsep *parity bit* (genap dan ganjil) sebagai cara sederhana untuk mendeteksi kesalahan satu bit.
- Praktik Terbimbing: Peserta didik memodifikasi *function* DesimalKeBiner menjadi DesimalKeBinerGenap (Aktivitas AP-K9-17). Ini memerlukan langkah tambahan: menghitung jumlah bit '1', lalu menambahkan '0' atau '1' di akhir.
- Praktik Mandiri: Peserta didik membuat *function* kebalikannya, BinerGenapKeDesimal (Aktivitas AP-K9-18), yang pertama-tama memeriksa paritas. Jika paritas salah, program mengembalikan nilai error (misal: -1). Jika benar, program mengonversi data bit-nya ke desimal.

- Pembelajaran Berdiferensiasi:
 - Proses: Peserta didik yang kesulitan fokus pada logika menghitung bit '1'. Peserta didik yang cepat, diminta membuat versi paritas ganjil.
 - Produk: Program dengan *function* yang dapat menambah dan memeriksa *parity bit*.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi: "Dari permainan tadi, kapan *parity bit* gagal mendeteksi kesalahan?" (Ketika dua bit berubah).
- Rangkuman: Guru menyimpulkan bahwa *parity bit* adalah mekanisme deteksi error sederhana dengan keterbatasan.
- Tindak Lanjut: Peserta didik diminta mencari tahu mekanisme deteksi error lain yang lebih canggih (misal: CRC).
- Penutup: Salam dan doa.

PERTEMUAN 9 (2 JP : 80 MENIT)

Topik : Proyek Akhir: Menggabungkan Modul Konverter Bilangan

KEGIATAN PENDAHULUAN (10 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Apersepsi: Guru mereviu semua *function* konverter yang telah dibuat. "Kita sudah punya banyak 'alat' atau 'modul'. Sekarang, mari kita gabungkan menjadi satu aplikasi yang lengkap!"
- Motivasi (Meaningful): Guru menunjukkan target akhir: sebuah program di mana pengguna bisa memilih mau konversi dari basis apa ke basis apa.

KEGIATAN INTI (60 MENIT)

- Kerja Kelompok: Peserta didik dibagi menjadi kelompok-kelompok kecil untuk mengerjakan Aktivitas AP-K9-19.
- Perancangan Logika: Kelompok mendiskusikan alur program utama. Mereka perlu menggunakan banyak blok if-then-else untuk menentukan *function* mana yang harus dipanggil berdasarkan input pengguna.
- Problem Solving (Mindful): Guru memberikan petunjuk untuk kasus khusus, misal: konversi dari biner ke oktal. "Bagaimana caranya? Apakah kita perlu membuat fungsi baru?" (Tidak, kita bisa menggunakan fungsi yang ada: Biner -> Desimal -> Oktal).
- Implementasi dan Pengujian: Kelompok mengimplementasikan logika tersebut, menggabungkan semua *function* yang sudah ada, dan menguji program mereka secara menyeluruh.
- Pembelajaran Berdiferensiasi:
 - Proses: Kelompok yang lebih cepat dapat menambahkan lebih banyak pilihan konversi (misal: heksadesimal) atau membuat antarmuka pengguna yang lebih baik.
 - Produk: Satu program Blockly yang berfungsi sebagai konverter multi-basis.

KEGIATAN PENUTUP (10 MENIT)

- Presentasi Singkat: Beberapa kelompok mempresentasikan alur logika yang mereka gunakan.
- Refleksi: "Apa tantangan terbesar saat menggabungkan banyak modul menjadi satu program?"
- Tindak Lanjut: Kelompok diminta mulai merancang poster untuk mempresentasikan

hasil kerja mereka (Aktivitas AP-K9-20).

- Penutup: Salam dan doa.

PERTEMUAN 10 (2 JP : 80 MENIT)

Topik : Presentasi Proyek Poster

KEGIATAN PENDAHULUAN (5 MENIT)

- Orientasi: Salam, doa, dan presensi.
- Motivasi (Joyful): Guru menyiapkan suasana kelas seperti pameran karya dan menjelaskan bahwa hari ini adalah perayaan hasil kerja keras mereka.

KEGIATAN INTI (65 MENIT)

- Finalisasi Poster: Kelompok diberi waktu 15 menit untuk menyelesaikan poster mereka (Aktivitas AP-K9-20), yang menjelaskan tujuan program, cara kerja (menyoroti modularisasi), dan contoh penggunaan.
- Gallery Walk: Sesi pameran karya. Setiap kelompok menempelkan posternya. Setengah anggota kelompok tinggal di "stan" untuk menjelaskan kepada pengunjung, sementara setengahnya lagi berkeliling untuk melihat dan bertanya pada kelompok lain. Sesi ini dilakukan dalam dua putaran agar semua bisa berperan sebagai penjaga stan dan pengunjung.
- Presentasi Kelas: Setiap kelompok memberikan presentasi singkat (2-3 menit) di depan kelas, mendemonstrasikan program mereka dan menjelaskan poin utama dari poster mereka.
- Pembelajaran Berdiferensiasi:
 - Produk: Poster bisa dibuat secara digital (menggunakan Canva/Slides) atau fisik (di kertas karton).
 - Proses: Penilaian presentasi fokus pada pemahaman dan kolaborasi, bukan pada kemampuan berbicara di depan umum.

KEGIATAN PENUTUP (10 MENIT)

- Refleksi dan Apresiasi: Guru memberikan apresiasi kepada semua kelompok. Peserta didik diajak memberikan umpan balik positif kepada kelompok lain.
- Rangkuman Bab: Guru merangkum perjalanan belajar di Bab 7, dari membandingkan bahasa, membuat modul, hingga membangun proyek yang utuh.
- Penutup: Salam dan doa.

G. ASESMEN PEMBELAJARAN

ASESMEN DIAGNOSTIK

- **Tanya Jawab:** Di awal bab, guru bertanya: "Apa itu variabel?", "Apa fungsi dari blok repeat?", "Bagaimana cara kerja blok if-then-else?".
- **Kuis Singkat:** Kuis singkat dengan 3-5 pertanyaan dasar tentang antarmuka Blockly dan fungsi blok-blok dasar.

ASESMEN FORMATIF

- **Tanya Jawab:** Seputar materi yang sedang dibahas, seperti "Mengapa modul ini kita buat sebagai *function*, bukan *procedure*?"
- **Diskusi Kelompok:** Guru mengamati keaktifan, kemampuan berargumentasi, dan kolaborasi peserta didik saat mengerjakan proyek kelompok.

- **Latihan Soal/LKPD:** Pengecekan hasil program yang dibuat pada setiap aktivitas (misal: program bilangan prima, program hitung rata-rata). Guru memberikan umpan balik langsung.
- **Observasi:** Guru mengamati proses *debugging* yang dilakukan peserta didik untuk melihat sejauh mana pemahaman logika mereka.
- **Produk (Proses):**
 - Sketsa algoritma (pseudocode/flowchart) yang dibuat sebelum coding.
 - Versi awal program yang menunjukkan proses pengembangan.
 - Partisipasi dalam diskusi kelompok.

ASESMEN SUMATIF

- **Produk (Proyek):**
 - **Program Konverter Bilangan:** Penilaian pada fungsionalitas, efisiensi, dan struktur program (penggunaan modularisasi yang tepat).
 - **Poster:** Penilaian pada kejelasan informasi, desain visual, dan kemampuan menjelaskan konsep teknis dengan bahasa yang mudah dipahami.
- **Praktik (Kinerja):**
 - **Presentasi Kelompok:** Menilai kemampuan komunikasi, kolaborasi, dan pemahaman materi saat mempresentasikan poster dan mendemonstrasikan program.
 - **Debugging Challenge:** Diberikan sebuah program yang salah, peserta didik diminta menemukan dan memperbaiki kesalahan tersebut dalam batas waktu tertentu.
- **Tes Tertulis:** Tes akhir bab untuk mengukur pemahaman konseptual dan prosedural. (Soal Pilihan Ganda dan Esai)

Contoh Tes Tertulis :

Pilihan Ganda

1. Sebuah modul program yang dirancang untuk melakukan tugas tertentu dan mengembalikan sebuah nilai kepada pemanggilnya disebut...
 - a. Prosedur
 - b. Variabel
 - c. Perulangan
 - d. Fungsi
 - e. Parameter
2. Bilangan biner 1101 jika dikonversi ke dalam sistem bilangan desimal akan menjadi...
 - a. 10
 - b. 11
 - c. 12
 - d. 13
 - e. 14
3. Tujuan utama dari modularisasi program adalah...
 - a. Membuat program berjalan lebih cepat.
 - b. Mengurangi jumlah variabel yang digunakan.
 - c. Memecah program besar menjadi bagian-bagian kecil yang lebih mudah dikelola dan dapat digunakan kembali.
 - d. Menambah jumlah baris kode agar terlihat kompleks.
 - e. Hanya bisa digunakan dalam bahasa pemrograman Blockly.

4. Jika sebuah sistem menggunakan parity bit genap, dan data yang akan dikirim adalah 10110, maka bit paritas yang akan ditambahkan di akhir adalah...
 - a. 1, sehingga data menjadi 101101
 - b. 0, sehingga data menjadi 101100
 - c. 1, karena jumlah bit 1 sudah ganjil
 - d. 0, karena jumlah bit 0 ada dua
 - e. Tidak perlu ditambahkan apa-apa
5. Dalam lingkungan pemrograman Blockly, untuk menggambar sebuah lingkaran di layar, kita dapat menggunakan blok perintah dari kategori...
 - a. Loops
 - b. Logic
 - c. Math
 - d. Text
 - e. Draw

Esai

1. Jelaskan perbedaan mendasar antara *function* dan *procedure* dalam konsep modularisasi program! Berikan contoh kapan sebaiknya kita menggunakan masing-masing.
2. Anda diminta membuat program untuk mengonversi bilangan desimal menjadi bilangan oktal (berbasis 8). Jelaskan langkah-langkah (algoritma) yang akan Anda gunakan untuk membuat program tersebut!

Mengetahui,
Kepala Sekolah

....., 20..
Guru Mata Pelajaran

.....
NIP.

.....
NIP.