

LEMBAR KERJA PESERTA DIDIK (LKPD) INFORMATIKA

KELAS XI

Strategi Algoritmik dan Pemrograman

- Kurikulum Merdeka -

IDENTITAS PESERTA DIDIK

Nama :
Kelas :

Nomor Absen :
Tanggal :

PETUNJUK Pengerjaan



1. Bacalah setiap instruksi dengan teliti.



2. Kerjakan kegiatan secara berurutan dan teliti.



3. Diskusikan dengan teman jika ada yang belum dipahami.



4. Kumpulkan hasil kerja tepat waktu dan jaga kerapian.

RINGKASAN MATERI

PROSES PEMROGRAMAN



Tahapan pembuatan program dimulai dari analisis masalah hingga pemeliharaan agar program berjalan dengan baik.

PSEUDOCODE (KODE SEMU)

```
Mulai  
Input n  
Jika n > 0 maka  
Tulis "Positif"  
Selain itu jika n < 0 maka  
Tulis "Negatif"  
Selain itu  
Tulis "Noi"  
Selesai
```

Deskripsi algoritma dalam bahasa manusia yang terstruktur, mudah dipahami sebelum diturunkan menjadi kode program.

REKURSIF (RECURSION)

```
def faktorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * faktorial(n-1)
```

Teknik pemecahan masalah dengan cara fungsi memanggil dirinya sendiri hingga mencapai kondisi dasar.

ARRAY (LARIK)

Index :	0	1	2	3	4
A	10	20	30	40	50

Struktur data berisi kumpulan elemen dengan tipe data sama yang diakses menggunakan indeks.

MASALAH SEGITIGA



Diketahui tiga sisi (a, b, c) atau dua sisi dan satu sudut, cari sisi atau sudut lainnya menggunakan aturan trigonometri.

FLOWCHART (DIAGRAM ALIR)



Representasi grafis algoritma menggunakan simbol standar untuk mempermudah pemahaman alur penyelesaian masalah.

BAHASA PEMROGRAMAN



Bahasa formal untuk memberi instruksi kepada komputer. Contoh: Python, C++, Java, JavaScript, dan lainnya.

ALGORITMA GREEDY



Strategi memilih solusi terbaik pada setiap langkah saat ini dengan harapan mendapatkan solusi global yang optimal.

STRING (UNTUKAN)

```
S = "Informatika"  
I n f o r m a t i k a  
0 1 2 3 4 5 6 7 8 9 10
```

Array karakter yang digunakan untuk merepresentasikan teks dalam program.

METODE KNAPSACK



Masalah optimasi memilih item dengan nilai maksimum tanpa melebihi kapasitas batasan (knapsack). Contoh: 0/1 Knapsack.

KEGIATAN

Kerjakan kegiatan berikut secara berurutan!

1

Analisis Masalah

Pilih satu masalah sederhana dalam kehidupan sehari-hari. Kerjakan dengan input dan output.

2

Buat Flowchart

Rancang alirannya menggunakan simbol standar.

3

Tulis Pseudocode

Tulis pseudocode dari algoritma yang telah kamu rancang pada kegiatan sebelumnya.

4

Implementasi Program

Implementasikan algoritma tersebut menggunakan bahasa pemrograman Python/Java/C++ atau sesuai pilihan guru.

5

Uji dan Evaluasi

Uji program yang kamu buat dengan beberapa data uji lalu evaluasi hasilnya. Tuliskan kesimpulan kegiatanmu.

PENILAIAN MANDIRI

Kerjakan soal berikut dengan jujur dan teliti!

A. PILIHAN GANDA

Pilihlah jawaban yang paling tepat!

- Tahap pertama dalam proses pemrograman adalah ...
a. Pemilihan kode
b. Analisis masalah
c. Pengujian program
d. Pemeliharaan program

- Simbol berikut ini digunakan untuk ...
a. Proses
b. Keputusan
c. Input/Output
d. Terminator

- Struktur data yang menyimpan banyak nilai dengan indeks disebut ...
a. String
b. Stack
c. Array
d. Queue

- Algoritma Greedy memilih solusi terbaik pada ...
a. Akhir proses
b. Setiap langkah saat ini
c. Sortiran awal
d. Proses input

- Metode Knapsack termasuk jenis masalah ...
a. Pemrograman
b. Optimasi kombinatorial
c. Simulasi

B. URAIAN (ESSAY)

Jawablah pertanyaan proses pemrograman berikut!

- Uraikan tahapan proses pemrograman secara berurutan dan berikan penjelasan singkat untuk tiap tahap!

- Apa perbedaan antara rekursif dan iterasi?

- Jelaskan konsep rekursif dengan contoh program sederhana (misalnya faktorial)!

- Berikan contoh penggunaan array dan string dalam pemrograman!

- Jelaskan cara kerja metode Knapsack 0/1 dengan contoh kasus sederhana!

C. MENYUSUN

Posisikan pernyataan di kolom kiri dengan jawaban yang tepat di kolom kanan!

- | | |
|---------------------|---|
| 1. Flowchart | a. Struktur data berindeks |
| 2. Pseudocode | b. Representasi grafis algoritma |
| 3. Rekursif | c. Teknik fungsi memanggil dirinya sendiri |
| 4. Array | d. Untuk merepresentasikan teks |
| 5. String | e. Strategi memilih terbaik di setiap langkah |
| 6. Algoritma Greedy | f. Problem optimasi dengan batasan kapasitas |
| 7. Masalah Segitiga | g. Perhitungan sisi/sudut segitiga |
| 8. Knapsack | h. Deskripsi algoritma dalam bahasa manusia |

Informatika itu Keren!
Rancang Algoritma,
Wujudkan Solusi!

TIPS SUKSES

- Pahami masalah dengan baik
- Rancang algoritma yang tepat
- Uji dan perbaiki programmu
- Terus belajar dan berlatih!



“Algoritma adalah jantung dari pemrograman, dan kreativitas adalah kunci untuk menciptakan solusi terbaik.”



LEMBAR KERJA PESERTA DIDIK (LKPD)

INFORMATIKA KELAS XI

Strategi Algoritmik dan Pemrograman

Kurikulum Merdeka



IDENTITAS PESERTA DIDIK

Nama :
Kelas :

Nomor Absen :
Tanggal :

Algoritma yang baik adalah langkah kecil hari ini untuk solusi besar esok hari.



PETUNJUK BELAJAR



Bacalah setiap kegiatan dengan cermat.



Pahami materi ringkasan sebelum mengerjakan.



Kerjakan kegiatan secara mandiri atau berkelompok.



Gunakan sumber belajar lain jika diperlukan.



Periksa kembali hasil pekerjaan sebelum dikumpulkan.



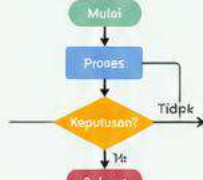
RINGKASAN MATERI

1 Proses Pemrograman



Tahapan sistematis untuk membuat program yang efektif dan efisien.

2 Flowchart



Representasi grafis algoritma menggunakan simbol-simbol standar.

3 Pseudocode

```
ALGORITMA LuasPersegi
INPUT sisi
OUTPUT Luas
BEGIN
    Luas ← sisi * sisi
    CETAK Luas
END
```

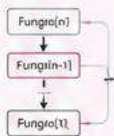
Deskripsi algoritma dengan bahasa sederhana, tidak terikat sintaks bahasa pemrograman tertentu.

4 Bahasa Pemrograman



Alat untuk menerjemahkan algoritma menjadi program yang dapat dieksekusi komputer.

5 Rekursi



Teknik pemrograman di mana fungsi memanggil dirinya sendiri untuk menyelesaikan masalah.

6 Algoritma Greedy



Struktur pilihan terbaik lokal pada setiap langkah dengan harapan mendapat solusi optimal.

7 Array (Lariki)



Struktur data untuk menyimpan kumpulan elemen bertipe sama yang diakses melalui indeks.

8 String (Teks)



Kumpulan karakter yang disimpan berurutan dan diakhiri karakter null ('\0').

9 Masalah Segitiga



Diketahui masalah: Cari jumlah maksimum dari atas ke bawah, hanya boleh bergerak diagonal kiri atau kanan ke bawah.

Dapat diselesaikan dengan pemrograman dinamis.

10 Metode Knapsack



	Berat	Nilai
Item A	2	12
Item B	1	20
Item C	3	15
Item D	2	15

Tentukan kombinasi barang yang memberikan nilai maksimum.

Memilih item dengan berat terbatas untuk memaksimalkan nilai total yang diperoleh.



KEGIATAN (KERJAKAN 5 KEGIATAN BERIKUT!)

1 Membuat Flowchart



Buat flowchart untuk menggambarkan proses kegiatan belajar mengajar di kelas.

2 Pseudocode



Tuliskan pseudocode untuk menggambarkan proses kegiatan belajar mengajar di kelas.

3 Array



Buat array untuk menyimpan nama-nama siswa di kelas.

4 String



Tuliskan algoritma untuk menghitung jumlah huruf vokal dalam sebuah kalimat.

8 Masalah Knapsack (Sederhana)



Tentukan kombinasi barang yang memberikan nilai maksimum dengan kapasitas terbatas.



PENILAIAN MANDIRI

A. PILIHAN GANDA

Pilihlah jawaban yang paling tepat!

- Tahapan pertama dalam proses pemrograman adalah
a. Penulisan kode
b. Analisis masalah
c. Uji coba
d. Pemeliharaan
- Simbol berikut digunakan untuk

a. Proses
b. Input/Output
c. Koneksi
d. Terminator

B. ESSAI

Jawablah pertanyaan berikut dengan jelas!

- Jelaskan perbedaan antara flowchart dan pseudocode!
- Apa yang dimaksud dengan rekursi? Berikan contoh dalam kehidupan sehari-hari!
- Jelaskan cara kerja algoritma greedy dengan contoh sederhana!

C. MENJODOKKAN

Jodohkan pernyataan di kolom kiri dengan jawaban di kolom kanan!

- | | |
|--|--------------|
| 1. Struktur data berdimensi satu | a. Knapsack |
| 2. Kumpulan karakter | b. String |
| 3. Memilih solusi terbaik lokal | c. Flowchart |
| 4. Masalah memilih item dengan batasan kapasitas | d. Array |
| 5. Representasi grafis algoritma | e. Greedy |



KODE ADALAH BAHASA MASA DEPAN.
TERUS BELAJAR, BERLATIH, DAN BERKARYA!



Scan untuk sumber belajar tambahan & latihan online!

Selamat Belajar dan Semangat Berlatih!

LEMBAR KERJA PESERTA DIDIK (LKPD)

INFORMATIKA KELAS 11



Strategi Algoritmik dan Pemrograman



IDENTITAS PESERTA DIDIK

Nama : _____
Kelas : _____



Model Pembelajaran

Problem Based Learning (PBL)



1 PETUNJUK BELAJAR

- Bacalah LKPD ini dengan cermat.
- Kerjakan setiap tahap kegiatan secara berkelompok (3–4 orang).
- Gunakan sumber belajar yang tersedia (buku, internet, video pembelajaran) untuk mendukung pemahaman.
- Diskusikan, tuliskan ide, algoritma, dan kode program dengan rapi.
- Presentasikan hasil kerja kelompok dan refleksi pembelajaran.



2 RINGKASAN MATERI



Berpikir Komputasional

Cara berpikir untuk memecahkan masalah secara sistematis dengan bantuan konsep komputasi.



Strategi Algoritmik

Merancang langkah-langkah penyelesaian masalah secara logis, efisien, dan tepat.



Proses Pemrograman

Tahapan membuat program: memahami masalah, merancang algoritma, menulis kode, menguji, dan mengevaluasi.



Dekomposisi
Memecah masalah menjadi bagian kecil.



Pengenalan Pola
Mencari kesamaan atau pola dari masalah.



Abstraksi
Fokus pada hal penting, abaikan yang tidak perlu.



Perancangan Algoritma
Menyusun langkah-langkah solusi yang logis.

- Berpikir komputasional terdiri dari 4 komponen utama:

- Struktur data yang akan digunakan pada kegiatan ini:



Array (Larik)

Menyimpan sekumpulan data bertipe sama dengan indeks tertentu.



String (Larik Karakter)

Menyimpan data teks berupa kumpulan karakter.



3 TUGAS DISKUSI – STUDI KASUS (MASALAH DUNIA NYATA)

Studi Kasus: Masalah Knapsack 0/1

Seorang pendaki akan melakukan pendakian gunung. Ia memiliki tas ransel dengan kapasitas maksimum W kg. Ia memiliki n barang yang masing-masing memiliki berat (w_i) dan nilai manfaat (v_i). Pendaki harus memilih barang-barang yang akan dibawa agar total berat tidak melebihi kapasitas ransel dan total nilai manfaat menjadi maksimum.



No	Barang	Berat (kg)	Nilai Manfaat
1	Tenda	4	10
2	Sleeping Bag	3	8
3	Kompas	2	6
4	Makanan	5	12
5	Jaket	2	4

Kapasitas ransel (W) = 7 kg

Pertanyaan Diskusi

- Identifikasi masalah utama pada kasus tersebut.
- Tentukan input, output, dan batasan masalah.
- Gunakan berpikir komputasional (dekomposisi, pengenalan pola, abstraksi, perancangan algoritma) untuk merancang solusi.
- Barang apa saja yang sebaiknya dibawa agar nilai manfaat maksimum? Tunjukkan total berat dan total nilai manfaatnya.
- Bagaimana jika kapasitas ransel berubah menjadi 10 kg? Apakah solusi berubah?



4 TUGAS PRAKTIK – IMPLEMENTASI ARRAY DAN STRING

A. Implementasi Array (Solusi Knapsack 0/1 dengan DP)

Gunakan pemrograman untuk menyelesaikan masalah knapsack 0/1 menggunakan Dynamic Programming.

Langkah Kerja

- Deklarasikan array untuk menyimpan berat (w), nilai (v), dan tabel DP (dp).
- Buat tabel DP berukuran $(n+1) \times (W+1)$.
- Isi tabel DP menggunakan rumus:
Jika $w[i-1] > W$ → $dp[i][j] = dp[i-1][j]$
Jika $w[i-1] \leq W$ → $dp[i][j] = \max(dp[i-1][j], dp[i-1][j - w[i-1]] + v[i-1])$
- Tampilkan nilai maksimum ($dp[n][W]$).
- Rekonstruksi solusi untuk mengetahui barang yang dipilih.
- Uji program dengan data pada tabel kasus dan $W = 7$ serta $W = 10$.

B. Implementasi String (Nama Barang)

Gunakan string/larik karakter untuk menyimpan dan menampilkan nama barang.

Langkah Kerja

- Buat array of string untuk menyimpan nama barang.
- Input atau deklarasikan nama-nama barang.
- Tampilkan nama barang yang terpilih berdasarkan hasil knapsack.
- Hitung dan tampilkan panjang nama barang terpilih menggunakan fungsi/operasi string.
- Tampilkan nama barang terpilih dalam format daftar rapi.



Tips

Gunakan bahasa pemrograman yang dikuasai (misal: C++, Python, atau Java). Pastikan program dikomentari dengan baik.



5 REFLEKSI

- Apa yang saya pelajari hari ini? _____
- Bagian mana yang paling menantang? Mengapa? _____
- Bagaimana berpikir komputasional membantu saya menyelesaikan masalah? _____
- Apa manfaat mempelajari algoritma dan pemrograman dalam kehidupan sehari-hari? _____



EVALUASI

Setelah tanda centang (✓) sesuai capaian kelompok Anda.

No	Aspek yang Dinilai	Sangat Baik (4)	Baik (3)	Cukup (2)	Perlu Perbaikan (1)
1	Memahami masalah dan mendefinisikan input/output				
2	Merancang algoritma dengan benar				
3	Implementasi array dan DP berjalan dengan benar				
4	Implementasi string berjalan dengan benar				
5	Kerja sama, komunikasi, dan presentasi kelompok				

Skor Akhir = (Total Skor / 20) x 100 → _____

Komentar Guru: _____



LEMBAR KERJA PESERTA DIDIK (LKPD)

INFORMATIKA KELAS 11

Strategi Algoritmik dan Pemrograman



IDENTITAS PESERTA DIDIK

Nama : _____
Kelas : _____



MODEL PEMBELAJARAN

PROBLEM BASED LEARNING (PBL)

"Belajar dimulai dari masalah nyata, diselesaikan dengan berpikir komputasional."



1. PETUNJUK BELAJAR

- Bacalah LKPD ini dengan cermat dan pahami setiap bagian.
- Kerjakan setiap aktivitas secara berkelompok (3-4 orang).
- Diskusikan dan catat hasil pemikiran kelompok pada kolom yang tersedia.
- Gunakan sumber belajar lain bila diperlukan.
- Presentasikan hasil kerja kelompok di depan kelas.
- Bersikaplah aktif, bertanggung jawab, dan saling menghargai pendapat.



2. RINGKASAN MATERI

A. Berpikir Komputasional

Berpikir komputasional adalah proses berpikir logis untuk memecahkan masalah secara sistematis sehingga solusi dapat diimplementasikan oleh komputer. Komponen utamanya:

- Dekomposisi**
Memecah masalah kompleks menjadi bagian-bagian kecil yang lebih sederhana.
- Pengenalan Pola**
Mengidentifikasi kesamaan atau pola dari bagian-bagian masalah.
- Abstraksi**
Menentukan informasi penting dan mengabaikan hal yang tidak relevan.
- Perancangan Algoritma**
Menyusun langkah-langkah terstruktur untuk menyelesaikan masalah.

B. Proses Pemrograman

Proses pemrograman meliputi tahapan berikut:

- Analisis Masalah**
Memahami masalah dan menentukan kebutuhan solusi.
- Perancangan Algoritma**
Membuat algoritma atau pseudocode.
- Implementasi (Coding)**
Menuliskan kode program sesuai algoritma.
- Pengujian (Testing)**
Menguji program dan memperbaiki kesalahan.
- Dokumentasi & Evaluasi**
Mencatat hasil dan mengevaluasi solusi.



3. TUGAS DISKUSI - STUDI KASUS (Knapsack Problem)

Masalah knapsack (tas ransel) adalah memilih sejumlah barang dengan nilai total maksimum tanpa melebihi kapasitas tas.



Kasus Nyata

Seorang pendaki akan melakukan pendakian dan memiliki tas dengan kapasitas maksimal 15 kg. Ia memiliki beberapa barang dengan berat dan nilai manfaat seperti pada tabel berikut.

No	Barang	Berat (kg)	Nilai Manfaat
1	Tenda	4	10
2	Sleeping Bag	3	6
3	Kompas	2	4
4	Jaket Tebal	5	12
5	Makanan	4	8
6	P3K	1	3

Tugas Diskusi:

- Tentukan kombinasi barang yang dapat dibawa agar nilai manfaat maksimum tanpa melebihi kapasitas 15 kg.
- Jelaskan algoritma/strategi yang digunakan (misalnya brute force atau dynamic programming sederhana).
- Gambarkan alur solusi dalam bentuk pseudocode atau flowchart.
- Presentasikan hasil dan alasan pilihan kelompok kalian.

Jawaban/ Hasil Diskusi Kelompok:



4. KEGIATAN PRAKTIK

Implementasi Array dan String

A. Array - Data Barang

- Buat program untuk menuliskan data barang (nama, berat, nilai) menggunakan array.
- Tampilkan seluruh data barang.
- Hitung total berat dan total nilai dari semua barang.

B. String - Input dan Validasi

- Buat program untuk menerima input nama barang dari pengguna (menggunakan string).
- Validasi bahwa nama barang tidak kosong.
- Cari dan tampilkan data barang berdasarkan nama yang diinput.



Langkah Kerja:



1. Analisis masalah dan tentukan kebutuhan program.



2. Rancang algoritma (pseudocode/flowchart).



3. Implementasikan kode program (Python/C++/Bebasa pemrograman lain).



4. Uji program dengan beberapa data.



5. Simpan dan dokumentasikan kode program serta hasilnya.



5. REFLEKSI

Jawaban diberikan secara jujur:

- Apa yang saya pelajari hari ini?
- Bagian mana yang paling menantang?
- Bagaimana cara kelompok saya mengatasi tantangan tersebut?
- Apa manfaat materi ini dalam kehidupan sehari-hari?



6. EVALUASI

Benar tanda centang (✓) pada kolom yang sesuai.

No	Aspek Penilaian	Sangat Baik (4)	Baik (3)	Cukup (2)	Perlu Perbaikan (1)
1	Memahami konsep berpikir komputasional				
2	Menganalisis masalah dengan benar				
3	Merancang algoritma dengan tepat				
4	Mengimplementasikan program dengan benar				
5	Kerja sama dan partisipasi kelompok				

Catatan Guru:



"Algoritma adalah langkah cerdas hari ini, menjadi solusi hebat di masa depan."





INFORMATIKA KELAS 11

Strategi Algoritmik dan Pemrograman



Algoritma



Array

Berpikir
Komputasional

Pemrograman

IDENTITAS PESERTA DIDIK

Nama :

Kelas :



TUJUAN PEMBELAJARAN

- Peserta didik mampu menerapkan strategi algoritmik untuk menyelesaikan masalah nyata.
- Peserta didik mampu mengimplementasikan solusi menggunakan struktur data array dan string.
- Peserta didik menunjukkan sikap kolaboratif, kritis, dan kreatif dalam memecahkan masalah.



1. PETUNJUK BELAJAR

- Bacalah LKPD ini dengan teliti.
- Kerjakan secara berkelompok (3-4 orang).
- Diskusikan setiap permasalahan menggunakan pendekatan Problem Based Learning (PBL).
- Catat ide, proses, dan hasil diskusi pada tempat yang disediakan.
- Lakukan praktikum pemrograman sesuai langkah kerja.
- Presentasikan hasil kerja kelompok di depan kelas.



3. TUGAS DISKUSI (STUDI KASUS)

Kasus: Persiapan Ekspedisi Pendakian

Tim pendaki akan melakukan pendakian. Mereka memiliki tas ransel dengan kapasitas maksimum 15 kg. Tersedia beberapa barang dengan berat dan nilai manfaat (kepentingan) berbeda-beda. Tim harus memilih barang dengan total berat tidak melebihi kapasitas tas agar manfaat yang diperoleh maksimal.

Tabel Data Barang

No	Barang	Berat (kg)	Nilai Manfaat
1	Tenda	4	8
2	Sleeping Bag	3	7
3	Kompas	2	4
4	Jaket	2	3
5	Makanan	4	9
6	P3K	1	2
7	Senter	1	1

Pertanyaan Diskusi:

- Identifikasi masalah pada kasus di atas.
- Tentukan tujuan dan batasan masalah.
- Rancang strategi/algoritma untuk memilih kombinasi barang terbaik (masalah Knapsack 0/1).
- Berikan kombinasi barang yang dipilih beserta total berat dan total nilai manfaatnya.
- Jelaskan alasan pemilihan kombinasi tersebut.



Petunjuk: Gunakan pendekatan berpikir komputasional dalam menyelesaikan masalah di atas.



2. RINGKASAN MATERI

Berpikir Komputasional (Computational Thinking)

Berpikir komputasional adalah cara berpikir untuk menyelesaikan masalah secara logis dan sistematis yang terdiri dari empat komponen utama:



Dekomposisi

Memecah masalah kompleks menjadi bagian-bagian kecil yang lebih mudah dikelola.



Pengenalan Pola

Mencari kesamaan atau pola dari informasi yang ada.



Abstraksi

Menentukan informasi penting dan mengabaikan detail yang tidak relevan.



Perancangan Algoritma

Menyusun langkah-langkah logis dan terstruktur untuk menyelesaikan masalah.

Proses Pemrograman

- Analisis masalah
- Perancangan algoritma
- Implementasi (coding)
- Pengujian & debugging
- Evaluasi & pemeliharaan



4. KEGIATAN PRAKTIKUM

Tujuan: Mengimplementasikan penyelesaian masalah Knapsack 0/1 menggunakan struktur data array dan string.

Langkah Kerja:

- Analisis**
Pahami kembali masalah dan data barang.
- Representasi Data dengan Array**
 - Buat array untuk menyimpan nama barang (string).
 - Buat array untuk menyimpan berat (integer).
 - Buat array untuk menyimpan nilai manfaat (integer).
- Rancang Algoritma**
 - Gunakan pendekatan brute force (mencoba semua kombinasi) atau dynamic programming.
 - Tentukan kombinasi dengan nilai manfaat maksimum dengan batasan kapasitas.
- Implementasi (Coding)**
Implementasikan algoritma pada bahasa pemrograman pilihan Anda (C++, Python, atau Java).
- Pengujian**
Uji program dengan data pada tabel dan beberapa data tambahan.
- Output Program**
Tampilkan kombinasi barang terpilih, total berat, dan total nilai manfaat.
- Dokumentasi**
Simpan kode program dan hasil eksekusi. Sertakan penjelasan singkat.



Catatan: Pastikan kode program rapi, menggunakan nama variabel yang bermakna, dan diberi komentar.



5. REFLEKSI

- Apa hal paling menantang yang kalian temui saat menyelesaikan masalah?
.....
- Bagaimana cara kelompokmu mengatasi tantangan tersebut?
.....
- Apa yang kalian pelajari tentang berpikir komputasional hari ini?
.....
- Bagaimana penerapan masalah ini dalam kehidupan sehari-hari?
.....



6. EVALUASI

Penilaian Kelompok (Rubrik Singkat)

Aspek Penilaian	Sangat Baik (4)	Baik (3)	Cukup (2)	Perlu Perbaikan (1)
Pemahaman Masalah	Memahami masalah secara menyeluruh	Memahami sebagian besar masalah	Memahami sebagian kecil masalah	Belum memahami masalah
Strategi/Algoritma	Strategi tepat dan sistematis	Strategi cukup tepat	Strategi kurang tepat	Tidak ada strategi
Implementasi Program	Program berjalan sempurna	Program berjalan dengan sedikit error	Program berjalan tetapi banyak error	Program tidak berjalan
Kerja Sama	Sangat aktif & kolaboratif	Aktif & bekerja sama	Kurang aktif & kurang bekerja sama	Tidak aktif & tidak bekerja sama
Presentasi & Dokumentasi	Sangat jelas & terstruktur	Cukup jelas	Kurang jelas	Tidak jelas

Skor Akhir: / 20



LKPD

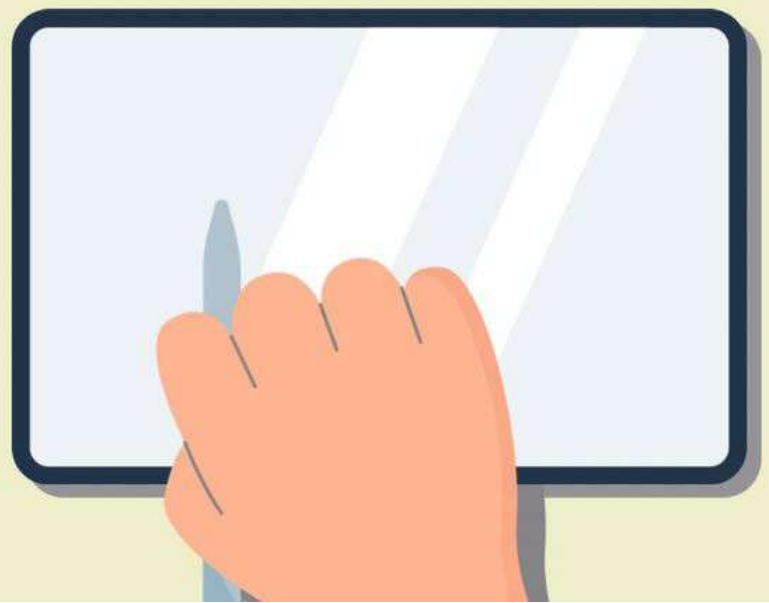
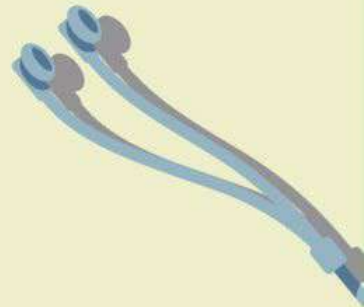
Lembar Kerja Peserta Didik

Strategi Algoritmik & Pemrograman

Informatika Kelas XI

Nama : _____

Kelas : _____



Aktivitas 1

Pilihan Ganda

Jawablah dengan memilih jawaban yang paling tepat!

1. Strategi algoritmik adalah cara berpikir untuk ...
 - A. Menghafal kode program
 - B. Menyelesaikan masalah secara terstruktur dan logis
 - C. Menggunakan komputer dengan cepat
 - D. Menjalankan aplikasi
2. Algoritma disusun dengan tujuan utama untuk ...
 - A. Memperindah tampilan program
 - B. Menyulitkan pemrogram
 - C. Menjelaskan langkah penyelesaian masalah
 - D. Mengurangi penggunaan perangkat keras
3. Salah satu ciri algoritma yang baik adalah ...
 - A. Langkah-langkahnya tidak jelas
 - B. Bersifat ambigu
 - C. Memiliki urutan yang logis
 - D. Sulit dipahami
4. Pemrograman merupakan proses ...
 - A. Menyusun perangkat keras
 - B. Mengubah algoritma menjadi instruksi yang dapat dijalankan komputer
 - C. Menggambar diagram manual
 - D. Menyimpan data
5. Strategi algoritmik sangat penting dalam pemrograman karena ...
 - A. Memperpanjang kode
 - B. Membuat program lebih rumit
 - C. Membantu penyelesaian masalah secara sistematis
 - D. Menghindari penggunaan logika

Aktivitas 2

Soal Melengkapi

Lengkapilah pernyataan berikut dengan jawaban yang tepat!

1. Algoritma adalah urutan langkah yang dan logis.
.....
2. Strategi algoritmik digunakan untuk suatu masalah.
.....
3. Algoritma disusun sebelum proses dilakukan.
.....
4. Setiap algoritma harus memiliki awal dan
.....
5. Langkah dalam algoritma harus bersifat
.....
6. Pemrograman mengubah algoritma menjadi bentuk komputer.
.....
7. Algoritma yang baik mudah dan dipahami.
.....
8. Strategi algoritmik membantu menghindari kesalahan
.....
9. Pemrograman memerlukan logika dan yang tepat.
.....
10. Algoritma dapat diterapkan dalam kehidupan
.....

Aktivitas 3

Uraian

Jawablah pertanyaan berikut dengan lengkap dan jelas!

1. Jelaskan pengertian strategi algoritmik!

2. Mengapa algoritma perlu disusun secara runtut dan logis?

3. Jelaskan hubungan antara algoritma dan pemrograman!

4. Sebutkan manfaat penggunaan strategi algoritmik dalam menyelesaikan masalah!

5. Berikan contoh sederhana penerapan algoritma dalam kehidupan sehari-hari (tanpa menuliskan kode)!

Nama :

Kelas :

ALGORITMA PEMROGRAMAN

Jodohkanlah pernyataan jika maka berikut ini sesuai dengan teori algoritma pemrograman yang tepat!

Jika



Cuci tangan



Mengantuk



Kuku Panjang



Lapar



Makan



Baju Kotor



Maka



Tidur



Tangan Bersih



Kenyang



Potong Kuku



Makan



Cuci Baju



Nama :

Kelas :

ALGORITMA PEMROGRAMAN

Jodohkanlah pernyataan jika maka berikut ini sesuai dengan teori algoritma pemrograman yang tepat!

Jika

Maka



Cuci tangan



Mengantuk



Kuku Panjang



Lapar



Makan



Baju Kotor



Tidur



Tangan Bersih



Kenyang



Potong Kuku



Makan



Cuci Baju

Nama :

Kelas :

ALGORITMA MEMBUAT TEH MENGUNAKAN FLOWCHART



Petunjuk!

- Bacalah kembali materi algoritma dan flowchart yang sudah dijelaskan guru.
- Lihat dan perhatikan contoh flowchart tentang membuat teh.
- Buat flowchart sederhana di bawah ini.
- Hias flowchart agar rapi, jelas, dan menarik.



Nama :

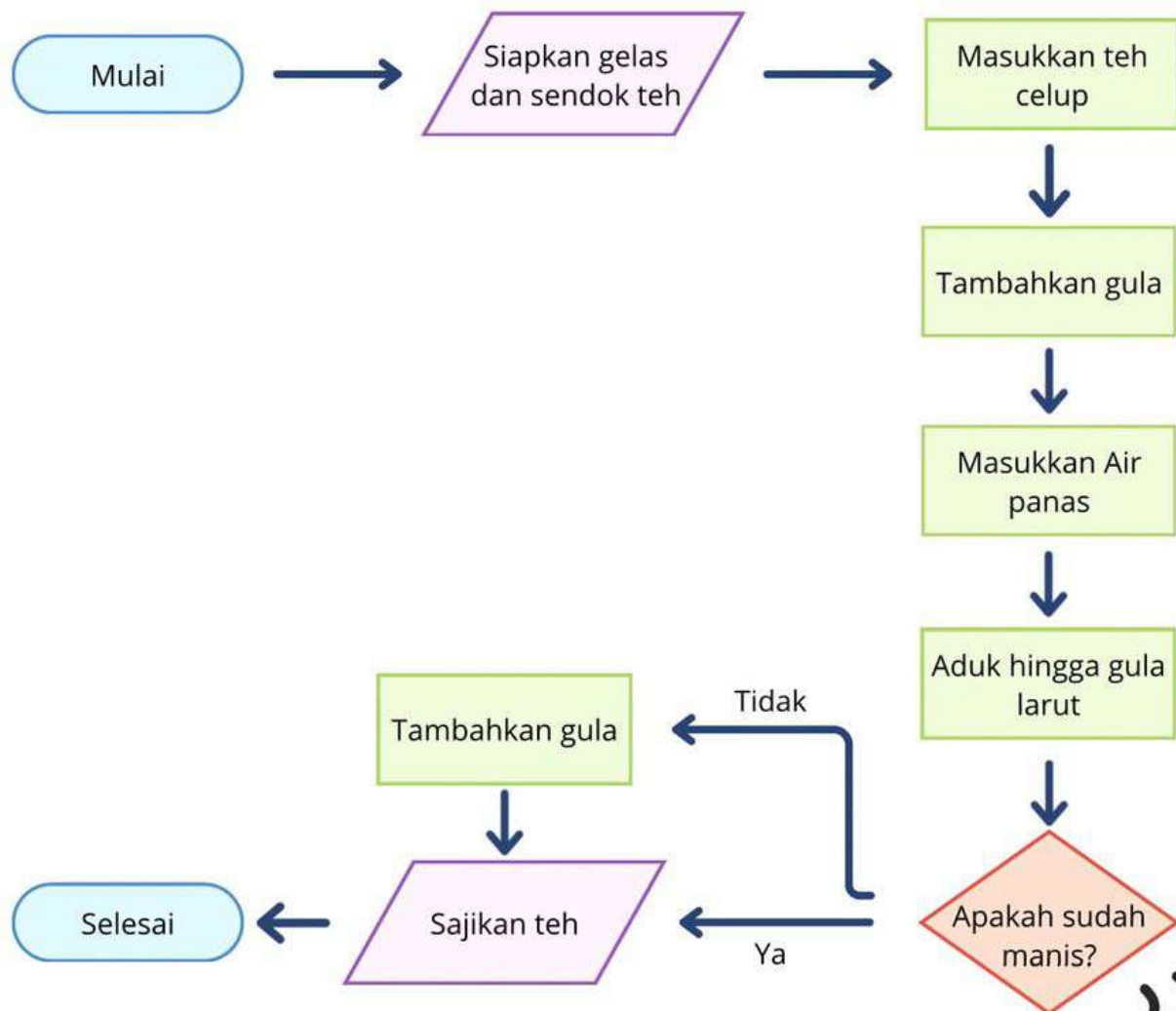
Kelas :

ALGORITMA MEMBUAT TEH MENGUNAKAN FLOWCHART



Petunjuk!

- Bacalah kembali materi algoritma dan flowchart yang sudah dijelaskan guru.
- Lihat dan perhatikan contoh flowchart tentang membuat teh.
- Buat flowchart sederhana di bawah ini.
- Hias flowchart agar rapi, jelas, dan menarik.



MENYUSUN ALGORITMA

Susun langkah-langkah berikut menjadi urutan algoritma yang tepat.

Nomori langkah secara urut.



Algoritma adalah urutan langkah-langkah yang sistematis untuk menyelesaikan masalah.

Susunlah langkah-langkah di bawah ini menjadi urutan yang tepat.

Kasus: Membuat Teh Manis Hangat

Susun langkah-langkah berikut agar algoritma membuat teh manis hangat menjadi tepat dan sistematis.

- A**  Rebus air hingga mendidih.
- B**  Masukkan teh coup ke dalam gelas.
- C**  Tambahkan gula secukupnya.
- D**  Aduk hingga gula larut.
- E**  Tuangkan air panas ka dalam gelas berisi teh celup.
- F**  3-5 menit Diamkan selama 3-5 menit ageh teh meresap.
- G**  Aduk perlahan.
- H**  Angkat teh celup.
- I**  Teh manis hangat siap dinikmati.
- J**  Siapkan bakan dan alat: teh celup, gula, air, gelas, sendok, ketel.

TULISKAN NOMOR URUT
DI KOTAK SEBELAH KANAN

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	



TAHUKAH KAMU?

Algoritma membantu kita berpikir logis dan menyelesaikan masalah dengan langkah yang teratur.

Mulailah dari langkah pertama sampai tujuan tercapai!

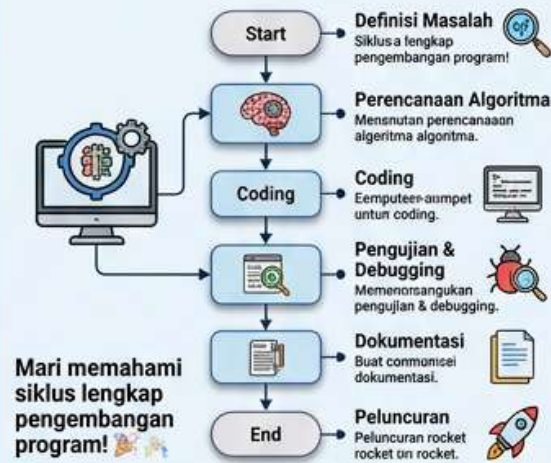


Berpikir sistematis adalah kunci solusi yang tepat!



1 PROSES PEMROGRAMAN

Bagian 1: Proses Pemrograman (Programming Process)



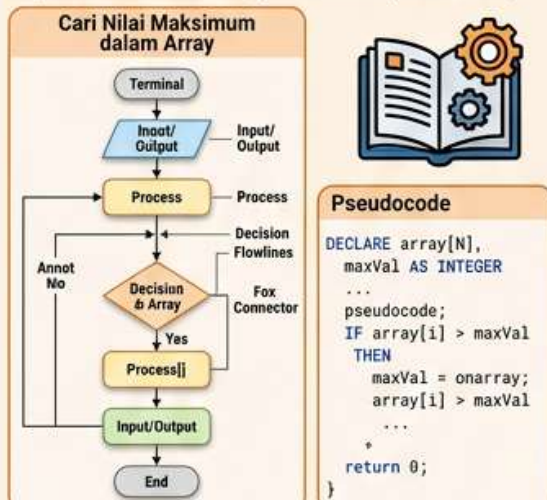
2 BERPIKIR KOMPUTASIONAL

Bagian 2: 4 Pilar Berpikir Komputasional (Computational Thinking)



3 ALGORITMA

Bagian 3: Memahami Algoritma & Representasinya



4 PEMROGRAMAN

Bagian 4: Dari Algoritma ke Kode Program

Code Snippet

```

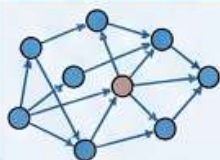
DECLARE array[N],
maxVal AS INTEGER
maxVal = onarray;
IF array[i] > maxVal THEN
    array[i] = maxVal
return
else:
    return 1;
    
```

Contoh Struktur Data

Data	Array	Pengunaan
List/Array	...	Dirseil anangan basas suana dengan mengernanun array
Stack	LIFO	Merontapkan wazuan platemahan stack dari certain sentoi data
Queue	→	Pengenaapan yang bantu msiingan cotah dibanngan dalam boda di adalah

5 LATIHAN & EVALUASI

Studi Kasus: Optimalisasi Rute



1. Buat mengengimilih Greedy/DP sorien justiffican nemsyak breslif antoamanan bafIP?
2. Bagarma yang justifficang?

Coding Challenge 1

Implementanis stadel, implement rekursive function template dialakan snim openams necerna ini-di kutam:

```

function function(rrry){
    maxVal = 2;
    maxVal = maxVal, funeta();
    if array[i] > maxVal:
        array[i] > maxVal, maxVal()
    return 1;
}
    
```

Reflective Section

1. Strategi mana yang paling menantang?

2. Bagaimana penerapan DP berbeda dari Greedy?

3. Bagaimana penerapan DP belat dari Greedy?

Assessment

No.	Ponta Asseson	Butuh (%)	Skor	Nilas (%)
1	Pemahaman Rubrsis			
2	Analysis Data			
3	Presentasi Rustuk			
TOTAL				



Tips Cerdas: Jangan takut berbuat salah saat coding! Kesalahan adalah bagian dari proses belajar.



Motivasi: "The only way to write good code is to write bad code and then make it better." - Martin Fowler