

MODUL AJAR
BAB 2 : STRATEGI ALGORITMA DAN PEMROGRAMAN

INFORMASI UMUM

A. IDENTITAS MODUL

Nama Penyusun :
Satuan Pendidikan : SMA
Kelas / Fase : XI (Sebelas) - F
Mata Pelajaran : Informatika
Prediksi Alokasi Waktu :
Tahun Penyusunan : 20

B. KOMPETENSI AWAL

Bab Strategi Algoritmik dan Pemrograman pada Kelas XI ini merupakan gabungan dari Bab Berpikir Komputasional serta Algoritma dan Pemrograman di Kelas X. Pada Kelas X, aktivitas yang diberikan pada Algoritma dan Pemrograman masih dititikberatkan pada kompetensi dasar seperti kemampuan memahami algoritma yang ditulis dalam bentuk diagram alir atau *pseudocode* dan kemampuan menulis kode program (*coding*). Pada Kelas XI, aktivitas yang diberikan lebih dititikberatkan pada kemampuan pemrograman untuk menyelesaikan suatu permasalahan. Oleh karena itu, berpikir komputasional serta algoritma dan pemrograman menjadi suatu kesatuan yang tidak dapat dipisahkan. Hal inilah yang menjadi alasan utama untuk menyatukan kedua bab tersebut menjadi Bab Strategi Algoritmik dan Pemrograman.

C. PROFIL PELAJAR PANCASILA

Beriman, bertakwa kepada Tuhan yang Maha Esa, bergotong royong, bernalar kritis, kreatif, inovatif, mandiri, berkebhinekaan global

D. SARANA DAN PRASARANA

- | | | |
|-----------------------|----------------------------|----------------------------------|
| 1. Gawai | 4. Buku Teks | 7. Handout materi |
| 2. Laptop/Komputer PC | 5. Papan tulis/White Board | 8. Infokus/Proyektor/Pointer |
| 3. Akses Internet | 6. Lembar kerja | 9. Referensi lain yang mendukung |

E. TARGET PESERTA DIDIK

Peserta didik reguler/tipikal: umum, tidak ada kesulitan dalam mencerna dan memahami materi ajar.

F. MODEL PEMBELAJARAN

Blended learning melalui model pembelajaran dengan menggunakan *Project Based Learning* (PBL) terintegrasi pembelajaran berdiferensiasi berbasis *Social Emotional Learning* (SEL).

KOMPONEN INTI

A. TUJUAN PEMBELAJARAN

- Memahami proses pemrograman.
- Menguasai konsep dan penerapan rekursi.
- Menganalisis beberapa strategi algoritmik (algoritma *greedy* dan pemrograman dinamis) secara kritis untuk menyelesaikan suatu permasalahan dengan memberikan justifikasi efisiensi.
- Menguasai konsep dan penerapan larik dalam pemrograman.
- Menguasai konsep dan penerapan karakter dan string dalam pemrograman.
- Menerapkan strategi algoritmik tersebut ke dalam suatu program yang lebih kompleks
- Bergotong royong dalam mengembangkan dan menguji program untuk menyelesaikan *problem*.

B. PEMAHAMAN BERMAKNA

Tabel 2.2 Pengalaman Belajar Bermakna, Profil Pelajar Pancasila, Berpikir Komputasional, dan Praktik Inti Unit SAP

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Berlatih konsep Strategi Algoritma dan Pemrograman	Mandiri, Bernalar kritis.	Abstraksi, Algoritma, Dekomposisi, Pengenalan Pola	Mengenali dan mendefinisikan <i>problem</i> komputasional, mengembangkan dan menggunakan abstraksi.
Merancang program sebagai solusi dari permasalahan yang terinspirasi dari mata pelajaran lain.	Bernalar kritis, bergotong royong, kreatif, berkebinekaan global.	Abstraksi, Algoritma, Dekomposisi, Pengenalan Pola	Mengenali dan mendefinisikan <i>problem</i> komputasional, mengembangkan dan menggunakan abstraksi.
Membuat program berdasarkan hasil rancangan.	Bernalar kritis, bergotong royong, kreatif, berkebinekaan global	Abstraksi, Algoritma, Dekomposisi, Pengenalan Pola	Membuat artifak komputasional, menguji dan meningkatkan kualitas artifak komputasional, mengkomunikasikan informasi terkait Informatika.

C. PERTANYAAN PEMANTIK

Pernahkah kalian berpikir bagaimana program yang kita gunakan dalam kehidupan sehari-hari dikembangkan? Bagaimana para pemrogram menghasilkan program yang sangat kompleks yang kalian gunakan di kehidupan sehari-hari? Bagaimana program tersebut dirancang dan diimplementasikan hingga dapat melayani ribuan hingga jutaan pengguna dengan efisien sehingga program tetap terasa nyaman saat digunakan?

D. KEGIATAN PEMBELAJARAN

PERTEMUAN KE-1

PROSES PEMROGRAMAN (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru bertanya pada peserta didik mengenai program dan aplikasi yang pernah dipakai peserta didik, dan menggali persepsi mereka tentang proses pemrograman yang terjadi di balik program dan aplikasi yang mereka gunakan sehari-hari.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (5 menit) Penjelasan tujuan pertemuan dan kegiatan yang akan dilakukan
- (20 menit) Penjelasan materi tentang proses pemrograman
- (5 menit) Pembagian peserta didik dalam kelompok
- (90 menit) Pelaksanaan kegiatan SAP-K11-01-U Ayo Berlatih: Mengamati Evolusi Program atau aplikasi yang mereka gunakan dari versi awalnya
- (90 menit) Presentasi hasil pencarian oleh peserta didik
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan

- Untuk aktivitas ini, peserta didik diharapkan dapat menyajikan hasil studi sederhana dengan melakukan pencarian dan membaca dari sumber-sumber di web yang menunjukkan sebuah contoh riwayat evolusi sebuah program/ aplikasi, dan kemudian menyampaikan hasil temuan mereka kepada guru/ di depan kelas. Salah satu format yang dapat mereka tampilkan adalah seperti pada contoh pada gambar berikut yang menunjukkan evolusi sistem operasi Microsoft Windows.
- Peserta didik harus dapat menjelaskan secara ringkas versi-versi apa saja yang telah dibuat, serta menjelaskan perubahan/penambahan fitur-fitur penting apa yang membedakan antara satu versi dengan versi-versi sebelumnya. Sebagai sumbernya, peserta didik dapat mencari dari Wikipedia, dari web resmi masing-masing aplikasi, ataupun dari toko-toko digital penjualan aplikasi (seperti App Store) yang menyediakan sejarah versi aplikasi-aplikasi yang tersedia pada toko tersebut.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi proses pemrograman yang telah dipelajari melalui aktivitas SAP-K11-01-U Ayo Berlatih: Mengamati Evolusi Program.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-2

REKURSI (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru dapat menunjukkan gambar-gambar yang menunjukkan konsep rekursi, seperti yang terdapat dalam contoh di Buku Siswa.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan

- (5 menit) Penjelasan tujuan pertemuan dan kegiatan yang akan dilakukan
- (10 menit) Penjelasan materi tentang rekursi
- (5 menit) Pembagian peserta didik ke dalam kelompok
- (45 menit) Pelaksanaan aktivitas SAP-K11-02-U Ayo Berlatih: Memahami Relasi Rekurensi
- (30 menit) Pembahasan aktivitas SAP-K11-02-U Ayo Berlatih: Memahami Relasi Rekurensi
- (85 menit) Pelaksanaan aktivitas SAP-K11-03-U Ayo Berlatih: Menerapkan Konsep Rekursi
- (30 menit) Pembahasan aktivitas SAP-K11-03-U Ayo Berlatih: Menerapkan Konsep Rekursi
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan

(SAP-K11-02-U) Ayo Berlatih!: Memahami Relasi Rekurensi

1. Kita hitung mulai dari suku ketiga dan seterusnya, menggunakan definisi rekursi yang diberikan sehingga kita dapatkan barisan tersebut adalah:

$$\{a_i\} = 1, 1, 3, 5, 11, 21, 43, 85, 171, 341, \dots$$

sehingga suku ke-10 dari barisan itu ialah 341.

2. Kita harus menentukan terlebih dahulu, apakah kita dapat menyatakan nilai sebagai hasil dari perhitungan menggunakan saja, ataukah kita memerlukan nilai dan seterusnya.

Ternyata, dalam hal ini, kita hanya memerlukan nilai saja, karena

$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1 = n \times (n-1)!$$

Kita juga harus menentukan nilai basis/awal dari fungsi faktorial, yaitu bahwa $1! = 1$. Oleh karena itu, definisi lengkap secara rekursif dari fungsi faktorial adalah

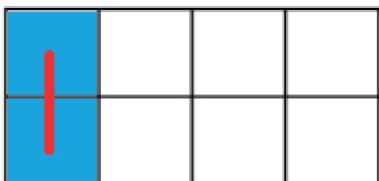
$$n! = \begin{cases} 1, & \text{jika } i = 1 \\ n \times (n-1)! & \text{jika } i > 1 \end{cases}$$

(SAP-K11-03) Ayo Berlatih: Menerapkan Konsep Rekursi

Permasalahan 1: Memasang Keramik

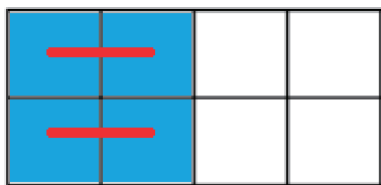
Untuk menghitung banyaknya cara memasang keramik untuk secara langsung, akan sulit dan rentan terjadi kesalahan. Cara yang lebih baik adalah mencari sebuah hubungan rekursif yang dapat membantu kita dalam menghitung banyaknya cara pemasangan keramik berdasarkan nilai, dan dari nilai-nilai yang sudah diketahui sebelumnya. Kita misalkan terlebih dahulu, bahwa banyaknya cara memasang keramik untuk lantai berukuran ialah sebanyak . Kemudian, kita berpikir secara rekursi, sebagai berikut.

1. Pertama-tama, kita dapat memilih untuk memasang keramik pada kolom pertama secara vertikal. Dengan demikian, akan tersisa kolom (atau dengan kata lain, sebuah lantai berukuran). Perhatikan ilustrasi pada gambar di bawah ini, untuk $N = 4$. Sisa lantai ini tentunya dapat diisi dengan keramik selanjutnya. Banyaknya cara mengisi sisa lantai dengan keramik ini tentunya adalah



2. Kedua, apabila kita memilih untuk meletakkan keramik paling kiri secara horizontal, kita harus mengisi dua kolom dan baris pertama, dengan dua buah keramik secara horizontal.

Hal ini berarti tersisa kolom (atau dengan kata lain, sebuah rantai berukuran). Perhatikan ilustrasi pada gambar di bawah ini, untuk $N = 4$.



Dengan demikian, sisa kolom tadi dapat dipasang keramik dengan sebanyak cara.

- Karena kedua cara tersebut di atas dapat dipilih secara bebas, banyaknya cara memasang keramik untuk rantai berukuran adalah hasil penjumlahan banyaknya cara dari kedua kasus di atas. Atau dengan kata lain, $FN = FN-1 + FN-2$. Relasi Rekurensi ini sama dengan relasi rekurensi pada barisan Fibonacci yang dijelaskan sebelumnya.
- Terakhir, kita harus menentukan nilai basis dari rekurensi ini. Karena relasi rekurensi di atas melibatkan dua suku sebelumnya ($FN-1$ dan $FN-2$), kita harus menentukan dua nilai pertama dari barisan FN , yaitu $F1$ dan $F2$. Untuk $N = 1$, jelas bahwa hanya ada satu cara memasang keramik pada rantai berukuran 2×1 , yaitu secara vertikal saja. Untuk $N = 2$, terdapat 2 cara memasang keramik, yaitu keduanya secara horizontal, atau keduanya secara vertikal. Jadi, kita simpulkan bahwa $F1 = 1$ dan $F2 = 2$. Dari hasil perumusan secara rekursif baris FN di atas, kita dapat menghitung $F8$ dengan lebih mudah, yaitu: dimulai dengan nilai $F1 = 1$ dan $F2 = 2$, setiap suku berikutnya didapat dengan cara menjumlahkan dua suku terakhir. Jadi, barisan FN yang didapatkan adalah sebagai berikut:

$$\{FN\} = 1, 2, 3, 5, 8, 13, 21, 34, \dots$$

Sehingga, jawaban yang diinginkan adalah $F8 = 34$.

Permasalahan 2: Menumpuk Panekuk

Kita dapat menyelesaikan permasalahan penumpukan panekuk dengan berpikir secara rekursif sebagai berikut: untuk memindahkan sebanyak n buah panekuk-panekuk dari piring A ke piring C (menggunakan piring B sebagai tempat sementara), kita dapat melakukan 3 tahap berikut:

- Pindahkan $n - 1$ buah panekuk paling atas dari piring A ke piring B (dengan menggunakan piring C sebagai tempat sementara)
- Pindahkan panekuk paling bawah (paling besar) dari piring A ke piring C
- Pindahkan $n - 1$ buah panekuk dari piring B ke piring C

Jika jumlah langkah minimal untuk memindahkan n buah panekuk dinyatakan sebagai barisan HN , maka kita memerlukan $HN-1$ langkah pemindahan untuk melakukan tahap no. 1 dan 3 di atas, sedangkan tahap no. 2 hanya memerlukan 1 langkah. Oleh karena itu, kita dapat menyimpulkan bahwa barisan HN dapat didefinisikan secara rekursif dengan menggunakan relasi rekurensi sebagai berikut:

$$HN = HN-1 + 1 + HN-1 = 2HN-1 + 1$$

Sebagai basis dari rekurensi, jelas bahwa $H1 = 1$. Dari sini, kita dapat menghitung barisan HN sebagai berikut:

$$\{HN\} = 1, 3, 7, 15, 31, 63, \dots$$

Sehingga jawaban yang diinginkan adalah $H6 = 63$.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi rekursi yang telah dipelajari melalui aktivitas SAP-K11-02-U Ayo Berlatih: Memahami Relasi Rekurensi dan aktivitas SAP-K11-03-U Ayo Berlatih: Menerapkan Konsep Rekursi. Kemudian guru memberi

motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-3

ALGORITMA GREEDY (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Berikan ilustrasi sederhana tentang tujuan dari *greedy* terkait kehidupan sehari-hari. Misalnya ketika mencari jalur terdekat pada suatu peta, atau menukarkan sejumlah uang namun dengan meminimalkan banyaknya pecahan uang yang diperlukan, dan sebagainya.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (5 menit) Penjelasan tujuan pertemuan dan kegiatan yang akan dilakukan
- (15 menit) Penjelasan materi tentang algoritma greedy
- (5 menit) Pembagian peserta didik ke dalam kelompok
- (45 menit) Pelaksanaan aktivitas SAP-K11-04-U Ayo Berlatih: Mengerjakan Pekerjaan Rumah (PR)
- (25 menit) Pembahasan aktivitas SAP-K11-04-U Ayo Berlatih: Mengerjakan Pekerjaan Rumah (PR)
- (85 menit) Pelaksanaan aktivitas SAP-K11-05-U Ayo Berlatih: Mengunjungi Kebun Binatang atau SAP-K11-06-U Ayo Berlatih: Menukarkan Uang
- (30 menit) Pembahasan aktivitas SAP-K11-05-U atau SAP-K11-06-U
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan

(SAP-K11-04-U) Ayo Berlatih: Mengerjakan Pekerjaan Rumah (PR) Dengan menerapkan algoritma *greedy*, kita akan memprioritaskan PR-PR dengan waktu pengerjaan yang kecil terlebih dahulu supaya bisa mengerjakan sebanyak-banyaknya tugas/PR. Oleh karena itu, kita harus mengurutkan PR berdasarkan waktu pengerjaan, mulai dari yang terkecil sampai yang terbesar.

Tabel 2.3 Pembahasan Aktivitas (SAP-K11-04-U) Ayo Berlatih: Mengerjakan PR

No	PR	Waktu pengerjaan (jam)	No	PR	Waktu pengerjaan (jam)
1	D	0.5	6	A	1.5
2	I	0.5	7	J	2
3	C	1	8	G	2.5
4	H	1	9	B	3
5	F	1	10	E	4

Kita kemudian tinggal memilih PR-PR mulai dari atas (yang paling kecil), sampai kita mendapatkan total waktu pengerjaan maksimal = 8 jam. Dalam hal ini, dapat dilihat bahwa Budi hanya dapat mengerjakan maksimal 7 buah PR, yaitu D, I, C, H, F, A dan J, dengan total waktu pengerjaan = $0.5 + 0.5 + 1 + 1 + 1 + 1.5 + 2 = 7.5$ jam.

(SAP-K11-05-U) Ayo Berlatih: Mengunjungi Kebun Binatang

Untuk menyelesaikan permasalahan ini, kita dapat menerapkan prinsip *greedy* sebagai berikut: untuk dapat mengunjungi sebanyak-banyaknya atraksi hewan, Dina harus selalu memilih (dari pilihan yang tersisa) atraksi yang akan selesai paling dulu. Oleh karena itu, akan memudahkan kita jika kita urutkan terlebih dahulu daftar semua atraksi berdasarkan waktu selesainya, seperti ditunjukkan pada tabel berikut.

Tabel 2.4 Pembahasan Aktivitas (SAP-K11-05) Ayo Berlatih: Mengunjungi Kebun Binatang

No	Jam Mulai	Jam Selesai	Pertunjukan Hewan
1	8:00	9:30	Penguin
2	9:15	10:30	Orang utan
3	10:00	12:00	Harimau
4	11:00	12:00	Burung pemangsa
5	13:00	14:30	Beruang madu
6	14:00	15:00	Buaya
7	15:00	15:30	Singa
8	15:30	16:00	Anjing Laut
9	15:30	16:30	Panda
10	16:00	17:00	Ular Piton

Pertama-tama, kita pasti akan memilih pertunjukan yang selesai paling awal (Penguin). Selanjutnya, mulai dari atraksi kedua dan seterusnya, kita pilih atraksi berikutnya dari daftar yang sudah terurut di atas yang **tidak bentrok** dengan atraksi terakhir yang sudah dipilih. Berturut-turut, berikut adalah atraksi yang dipilih:

No	Jam Mulai	Jam Selesai	Pertunjukan Hewan
1	8:00	9:30	Penguin
2	10:00	12:00	Harimau
3	13:00	14:30	Beruang madu
4	15:00	15:30	Singa
5	15:30	16:00	Anjing Laut
6	16:00	17:00	Ular Piton

Sehingga total ada 6 pertunjukan yang dapat ditonton oleh Dina. Tidak ada cara pemilihan yang lain yang dapat dilakukan oleh Dina untuk dapat menonton lebih dari 6 pertunjukan.

(SAP-K11-06-U) Ayo Berlatih: Menukarkan Uang

Untuk menyelesaikan permasalahan ini, kita dapat menerapkan algoritma *greedy* sebagai berikut: kita lakukan beberapa langkah untuk memilih pecahan yang diambil, sampai didapatkan jumlah yang diperlukan:

Dapat dipahami bahwa pada setiap langkah, kita menerapkan algoritma *greedy* dengan mencari pecahan terbesar yang masih bisa diambil tanpa melewati besaran nilai yang diinginkan. Dengan memilih pecahan terbesar pada setiap langkah, kita dijamin akan meminimalkan banyaknya pecahan yang diperlukan.

Namun, perlu ditekankan dan dipahami bahwa pendekatan *greedy* tidak selalu dijamin berhasil, apabila nominal pecahan-pecahan uangnya diubah. Misalnya, andaikan bahwa pecahan yang tersedia bernilai seribuan, 13 ribuan dan dua puluh ribuan. Maka, jika kita ingin mencapai nilai 27 ribu rupiah, pendekatan secara *greedy* akan menghasilkan solusi = 1 lembar dua puluh ribuan dan 7 lembar seribuan, dengan total 8 lembar pecahan.

Padahal, kita dapat menggunakan dua lembar 13 ribuan ditambah 1 lembar seribuan, dengan total 3 lembar pecahan. Jadi, untuk permasalahan seperti ini, perlu dipastikan bahwa memang pendekatan *greedy* menghasilkan solusi yang benar-benar optimal. Apabila tidak, maka solusi dengan pendekatan lain, misalnya dengan menggunakan pemrograman dinamis (*dynamic programming*), mungkin lebih cocok untuk diterapkan pada permasalahan tersebut.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi algoritma *greedy* yang telah dipelajari melalui aktivitas SAP-K11-04-U Ayo Berlatih: Mengerjakan Pekerjaan Rumah, aktivitas SAP-K11-05-U Ayo Berlatih: Mengunjungi Kebun Binatang dan atau aktivitas SAP-K11-06-U Ayo Berlatih: Menukarkan Uang.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-4

PEMROGRAMAN DINAMIS (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Pada kegiatan pemanasan, guru bisa menunjukkan proses perhitungan fungsi *Fibonacci* secara rekursif dan mengapa **memoisasi** diperlukan untuk menghindari perhitungan yang diulang-ulang.
- Penjelasan guru dapat mengikuti alur seperti pada video: <https://youtu.be/UxICSjrdlJA>

Kegiatan Inti (90 Menit)

Sebelum masuk ke kegiatan inti, guru membuka kelas dan memberikan apersepsi dan materi pemanasan selama 10-15 menit. Selanjutnya guru bersama peserta didik dapat memulai kegiatan inti:

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (10 menit) Penjelasan tujuan pertemuan dan kegiatan yang akan dilakukan
- (50 menit) Penjelasan materi tentang pemrograman dinamis
- (100 menit) Pelaksanaan aktivitas SAP-K11-07-U Ayo Berlatih: Bermain Angka
- (50 menit) Pembahasan aktivitas SAP-K11-07-U Ayo Berlatih: Bermain Angka
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan

(SAP-K11-07) Ayo Berlatih: Bermain Angka

Asumsikan jumlah langkah yang diperlukan untuk mengubah sebuah bilangan n menjadi 1, sesuai dengan ketentuan pada soal, dinyatakan sebagai barisan $L(n)$. Jawaban yang ingin dihitung adalah $L(25)$.

Pertama-tama, perlu dipahami terlebih dahulu bahwa algoritma *greedy* disini juga tidak selalu menghasilkan nilai yang optimal. Misalnya, jika $n = 10$, maka dengan menerapkan algoritma *greedy*, kita akan cenderung untuk menerapkan langkah pembagian dengan 2 terlebih dahulu (karena 10 genap dan tidak habis dibagi 3). Dengan cara ini kita akan memerlukan 4 langkah, yaitu $10 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1$. Namun ternyata ada cara yang lebih singkat, yaitu $10 \rightarrow 9 \rightarrow 3 \rightarrow 1$ (3 langkah), sehingga $L(10) = 3$. Ini berarti kita harus memperhatikan semua kemungkinan jalur yang ada untuk mencapai 1, dan mencari yang terpendek. Namun, jika kita mencoba semua

kemungkinan, akan banyak sekali kemungkinan nilai yang berulang yang harus kita hindari supaya tidak dihitung lebih dari sekali (seperti pada kasus menghitung bilangan Fibonacci). Misalnya, saat menghitung $L(25)$, kita bisa menggunakan jalur langkah:

$25 \rightarrow 24 \rightarrow 8 \rightarrow 4 \rightarrow \dots$

Atau

$25 \rightarrow 24 \rightarrow 12 \rightarrow 4 \rightarrow \dots$

Sehingga nilai $L(4)$ dapat dihitung beberapa kali. Ini yang harus dihindari pada penyelesaian soal dengan teknik DP, yaitu dengan menggunakan tabel memoisasi. Dengan tabel memoisasi, kita dapat menyimpan nilai L yang sudah dihitung dan menggunakannya untuk menghitung nilai-nilai L yang lebih besar. Berikut ini adalah langkah-langkah yang kita lakukan:

Pertama, simpan nilai $L(1) = 0$ (tidak perlu melakukan apa-apa).

Untuk setiap nilai n selanjutnya, ambil 3 buah nilai:

- $A = L(n - 1)$ dari tabel memoisasi.
- Jika n habis dibagi 2, ambil nilai $B = L(n/2)$ dari tabel memoisasi.
- Jika n habis dibagi 2, catat pula $C = L(n/3)$ dari tabel memoisasi

Kemudian, ambil nilai terkecil dari A , B dan C . Misalkan hasilnya adalah D .

Maka, selanjutnya, isikan pada tabel memoisasi, nilai $L(n) = D + 1$.

Perhatikan bahwa untuk setiap nilai n , berlaku bahwa $n - 1$, $n/2$ (jika n habis dibagi 2) dan $n/3$ (jika n habis dibagi 3) adalah bilangan bulat positif yang lebih kecil dari n . Dengan demikian, kita yakin bahwa jika kita membangun tabel memoisasi ini dari bawah (bottom up), maka nilai A , B dan C pada langkah 2, 3 dan 4 di atas pasti sudah tersedia/terisi sebelumnya, sehingga tidak perlu kita hitung lagi.

Berikut ini adalah hasil tabel memoisasi yang dibangun dengan cara di atas (sampai $n = 25$). Dari tabel tersebut, dapat disimpulkan bahwa jawaban dari soal tersebut adalah $L(25) = 5$.

Tabel 2.5 Hasil Memoisasi dari sampai $n = 25$

N	L(n)		
1	0	14	4
2	1	15	4
3	1	16	4
4	2	17	5
5	3	18	3
6	2	19	4
7	3	20	4
8	3	21	4
9	2	22	5
10	3	23	6
11	4	24	4
12	3	25	5
13	4		

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi pemrograman dinamis yang telah dipelajari melalui aktivitas SAP-K11-07-U Ayo Berlatih: Bermain Angka.

- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-5

LARIK ATAU ARRAY (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru dapat memancing peserta didik tentang contoh-contoh larik di dunia nyata, misalnya seperti loker, kotak obat, atau contoh lain di sekitar mereka.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pemanasan
- (10 menit) Penjelasan tujuan pertemuan dan kegiatan yang akan dilakukan
- (45 menit) Penjelasan materi larik
- (100 menit) Melakukan aktivitas (SAP-K11-08) Ayo Berlatih: Latihan Larik
- (60 menit) Pembahasan aktivitas (SAP-K11-08) Ayo Berlatih: Latihan Larik
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan

(SAP-K11-08) Ayo Berlatih: Latihan Larik

1. Rancang dan buatlah sebuah program yang membaca N buah bilangan dan mencetaknya secara terbalik. Misalnya jika diberikan masukan 1 2 3 4 5 6, program akan mencetak 6 5 4 3 2 1.
2. Rancang dan buatlah sebuah program yang membaca sebuah matriks berukuran N x M dan mencetak hasil transpose matriksnya.
3. Modifikasilah kode program untuk menghitung jarak dua kota yang menerima suatu rute. Misalnya, jika rute yang dimasukkan adalah Kota
4. Tantangan larik: perhatikan bahwa pada Gambar 2.14 di Buku Siswa terdapat data yang berganda. Hal ini dikarenakan matriks tersebut menyimpan informasi jarak antara 2 kota (misal A dan B) sebagai jarak dari A-B dan B-A. Dapatkah kalian menemukan representasi yang lebih baik daripada contoh tersebut sehingga tidak ada duplikasi informasi dalam penyimpanan data jarak kota?

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi larik atau *array* yang telah dipelajari melalui aktivitas SAP-K11-08 Ayo Berlatih: Latihan Larik.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-6

KARAKTER DAN STRING (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru dapat memancing peserta didik untuk menjelaskan artefak informatika yang mereka gunakan yang menerima masukan berupa kata, kalimat, atau sejenisnya yang masuk dalam kategori string.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (10 menit) Penjelasan tujuan pertemuan dan kegiatan yang akan dilakukan
- (45 menit) Penjelasan materi karakter dan *string*
- (100 menit) Pengerjaan Latihan (SAP-K11-09) Ayo Berlatih: Latihan Karakter dan *String*
- (60 menit) Pembahasan latihan karakter dan *string*
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan

(SAP-K11-09) Ayo Berlatih: Latihan Karakter dan String

1. Rancang dan buatlah sebuah program yang dapat membaca sebuah string dan mencetaknya secara terbalik.
2. Rancang dan buatlah sebuah program yang membaca sebuah kata sandi dan mencetak jumlah karakter yang berupa angka, huruf, dan simbol. Bedakan huruf kapital dan non kapital.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi karakter dan *string* yang telah dipelajari melalui aktivitas SAP-K11-09 Ayo Berlatih: Latihan Karakter dan *String*.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-7

SIMULASI GERAK BURUNG (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru dapat memancing jawaban peserta didik mengenai apa saja contoh yang mereka tahu tentang penerapan fisika di program atau aplikasi yang mereka gunakan sehari-hari. Contoh seperti film animasi dan permainan dapat digunakan apabila tidak banyak yang bisa menjawab penerapan fisika di program atau aplikasi yang mereka gunakan.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, dan pemanasan
- (5 menit) Penjelasan tujuan pertemuan, kegiatan yang akan dilakukan, serta *subproblem* yang akan dikerjakan
- (45 menit) Pengerjaan aktivitas SAP-K11-10-U Ayo Merancang Program: Merancang Algoritma Simulasi Burung
- (135 menit) Pengerjaan aktivitas SAP-K11-11 Ayo Buat Program: Membuat Program Simulasi Burung
- (30 menit) Pembahasan kedua aktivitas: presentasi kelompok dan diskusi
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan dan Jawaban Aktivitas

Kunci jawaban akan diberikan pada dokumen terpisah sebagai suplemen yang tersedia pada Strategi Pembelajaran poin ke 6 tentang Solusi *Problem*.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi program simulasi gerak burung yang telah dipelajari melalui aktivitas SAP-K11-10-U Ayo Merancang Program: Merancang Algoritma Simulasi Burung dan aktivitas SAP-K11-11 Ayo Buat Program: Membuat Program Simulasi Burung.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-8

PENGELOLAAN DONOR DARAH (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru dapat memancing pengalaman peserta didik dalam mengambil keputusan berdasarkan data, baik dari pengalaman mereka sendiri maupun pengalaman orang lain. Contoh misalnya seperti merencanakan perjalanan wisata yang melibatkan pemilihan lokasi penginapan, waktu perjalanan, dan sebagainya yang sekarang dapat dilakukan dengan lebih akurat menggunakan berbagai aplikasi dan *web*.

Kegiatan Inti (90 Menit)

Sebelum masuk ke kegiatan inti, guru membuka kelas dan memberikan apersepsi dan materi pemanasan selama 5-10 menit. Selanjutnya guru bersama peserta didik dapat memulai kegiatan inti:

- (5 menit) Penjelasan tujuan pertemuan, kegiatan yang akan dilakukan, serta *subproblem* mana yang akan dikerjakan.
- (5 menit) Pembagian peserta didik dalam kelompok berpasangan.
- (50 menit) Pengerjaan aktivitas SAP-K11-12-U Ayo Merancang Program: Merancang Algoritma Pengelolaan Donor Darah
- (140 menit) Pengerjaan aktivitas SAP-K11-13 Ayo Buat Program: Membuat Program Pengelolaan Donor Darah
- (35 menit) Pembahasan kedua aktivitas (presentasi kelompok dan diskusi)
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan dan Jawaban Aktivitas

Kunci jawaban akan diberikan pada dokumen terpisah sebagai suplemen yang tersedia pada Strategi Pembelajaran poin ke 6 tentang Solusi *Problem*.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi program pengelolaan donor darah yang telah dipelajari melalui aktivitas SAP-K11-12-U Ayo Merancang Program: Merancang Algoritma Pengelolaan Donor Darah dan aktivitas SAP-K11-13 Ayo Buat Program: Membuat Program Pengelolaan Donor Darah.

- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-9

PERSILANGAN TANAMAN (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru mengingatkan peserta didik tentang konsep persilangan tanaman (hukum Mendel) yang telah mereka pelajari di mata pelajaran Biologi.
- Apabila peserta didik belum mendapatkan materi tersebut, guru dapat mengulas sekilas tentang hukum Mendel melalui video.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (10 menit) Penjelasan tujuan pertemuan, kegiatan yang akan dilakukan, serta *subproblem* mana yang akan dikerjakan
- (5 menit) Pembagian peserta didik ke dalam kelompok berpasangan
- (60 menit) Pengerjaan aktivitas SAP-K11-14-U Ayo Merancang Program: Merancang Algoritma Persilangan Tanaman
- (105 menit) Pengerjaan aktivitas SAP-K11-15 Ayo Buat Program: Membuat Program Persilangan Tanaman
- (35 menit) Pembahasan kedua aktivitas (presentasi kelompok dan diskusi)
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan dan Jawaban Aktivitas

Kunci jawaban akan diberikan pada dokumen terpisah sebagai suplemen yang tersedia pada Strategi Pembelajaran poin ke 6 tentang Solusi *Problem*.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi persilangan tanaman yang telah dipelajari melalui aktivitas SAP-K11-14-U Ayo Merancang Program: Merancang Algoritma Persilangan Tanaman dan aktivitas SAP-K11-15 Ayo Buat Program: Membuat Program Persilangan Tanaman.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-10

SIMULASI STOIKIOMETRI (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru mengingatkan peserta didik tentang konsep stoikiometri yang telah mereka pelajari di Mata Pelajaran Kimia.
- Apabila peserta didik belum mendapatkan materi tersebut, guru dapat mengulas sekilas tentang hukum Mendel lewat video.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (10 menit) Penjelasan tujuan pertemuan, kegiatan yang akan dilakukan, serta *subproblem* yang akan dikerjakan
- (5 menit) Pembagian peserta didik dalam kelompok berpasangan
- (45 menit) Pengerjaan aktivitas SAP-K11-16-U Ayo Merancang Program: Merancang Algoritma Simulasi Stoikiometri
- (120 menit) Pengerjaan aktivitas SAP-K11-17 Ayo Buat Program: Membuat Program Simulasi Stoikiometri
- (35 menit) Pembahasan kedua aktivitas (presentasi kelompok dan diskusi)
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan dan Jawaban Aktivitas

Kunci jawaban akan diberikan pada dokumen terpisah sebagai suplemen yang tersedia pada Strategi Pembelajaran poin ke 6 tentang Solusi *Problem*.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi simulasi stoikiometri yang telah dipelajari melalui aktivitas SAP-K11-16-U Ayo Merancang Program: Merancang Algoritma Simulasi Stoikiometri dan aktivitas SAP-K11-17 Ayo Buat Program: Membuat Program Simulasi Stoikiometri.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-11

PRAKTIK LINTAS BIDANG (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru dapat mengajak peserta didik untuk mengingat kembali materi-materi strategi penyelesaian masalah yang telah dipelajari pada bagian Berpikir Komputasional sebelumnya, yaitu *greedy* dan pemrograman dinamis, serta mengajak peserta didik untuk berpikir bagaimana mengimplementasikan algoritma yang telah dirancang menjadi sebuah program sebagaimana dilakukan pada pertemuan-pertemuan sebelumnya.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan
- (60 menit) Penjelasan permasalahan knapsack sebagaimana dijelaskan di Buku Siswa
- (5 menit) Penjelasan tujuan pertemuan, kegiatan yang akan dilakukan serta cara pengerjaan
- (5 menit) Pembagian peserta didik ke dalam kelompok berpasangan
- (55 menit) Pengerjaan aktivitas SAP-K11-18-U Aktivitas PLB: Memahami Permasalahan *Knapsack*
- (55 menit) Pengerjaan aktivitas SAP-K11-19 Aktivitas PLB: Mengkodekan Permasalahan *Knapsack*
- (35 menit) Pembahasan kedua aktivitas (presentasi kelompok dan diskusi)

Aktivitas

(SAP-K11-18-U) Aktivitas PLB: Memahami Permasalahan *Knapsack* Pada aktivitas ini, peserta didik diminta untuk memahami permasalahan *knapsack* sebagai sebuah permasalahan komputasional. Peserta didik diminta untuk menjawab beberapa pertanyaan yang diberikan terkait dengan permasalahan *knapsack* untuk menguji pemahaman mereka, serta kemampuan mereka untuk menyelesaikan beberapa kasus permasalahan *knapsack* yang sederhana (berukuran kecil), meskipun mungkin belum menemukan sebuah algoritma generik yang dapat diterapkan untuk menyelesaikan semua kasus permasalahan *knapsack*.

Pembahasan dan Jawaban Aktivitas

(SAP-K11-18-U) Aktivitas PLB: Memahami Permasalahan *Knapsack*

1. *Knapsack* adalah permasalahan optimasi mencari **maksimum**, yaitu total nilai barang yang terbesar yang dapat dimasukkan ke dalam wadah.
2. Fungsi tujuan dari optimasi pada permasalahan *knapsack* adalah total nilai barang yang dimasukkan ke dalam wadah.
3. Kendala optimasi pada permasalahan *knapsack* adalah kapasitas wadah.
4. Jawaban:
 - a. Mengambil barang-barang B, D, E dan F tidak diperbolehkan sebagai solusi, karena total bobot yang dihasilkan adalah $8 + 4 + 10 + 8 > 24$.
 - b. Mengambil A, D, E saja diperbolehkan karena total bobot yang didapatkan adalah $3 + 4 + 10 = 17 < 24$. Namun, nilai fungsi tujuannya disini (yaitu $6 + 6 + 5 = 17$) masih belum optimal, karena kita dapat memilih misalnya A, D dan F, dengan total bobot $3 + 4 + 8 = 15 < 24$, dan total nilai $= 6 + 6 + 10 = 22 > 17$.
5. Untuk variasi permasalahan *rational knapsack*, sebagaimana yang akan dipelajari nanti, solusi dari permasalahan dapat diperoleh dengan menerapkan strategi *greedy* yaitu dengan memilih barang-barang dengan rasio nilai terhadap bobot yang terbesar terlebih dahulu.

Tabel berikut menunjukkan proses ini:

Barang	Bobot	Nilai	Rasio nilai/ bobot	Proporsi pengambilan	Bobot terambil	Nilai terambil
A	3	6	2	1	3	6
D	4	6	1,5	1	4	6
F	8	10	1,25	1	8	10
C	5	5	1	1	5	5
B	8	4	0,5	0,5	4	2
E	10	5	0,5	0	0	0
Total					24	29

Jika diurutkan berdasarkan rasio nilai/bobot dari yang terbesar menuju ke yang terkecil, urutan barang adalah: A, D, F, C, B dan E. Kita berturut-turut mengambil barang berdasarkan urutan prioritas ini. Sampai dengan barang C, kita telah memperoleh total bobot = 20 kg, dan total nilai = 27. Karena kapasitas wadah hanya 24 kg, kita tidak dapat mengambil seluruh barang B yang berbobot 8 kg. Karena kita tinggal memiliki sisa kapasitas wadah 4 kg lagi, maka berarti kita hanya dapat mengambil 0.5 bagian dari barang B, untuk mendapatkan nilai sebanyak $0.5 * 4 = 2$. Jadi total nilai maksimal yang dapat kita kumpulkan adalah $27 + 2 = 29$.

6. Pada variasi *0-1 knapsack*, pilihan optimal didapatkan dengan memilih barang-barang A, D, F, dan C (dengan total bobot = 20 kg) dan total nilai = 27.

Pada aktivitas ini, peserta didik diminta untuk memahami dan menggunakan sebuah skema pengkodean permasalahan *knapsack*. Tugas mereka adalah pertama menjawab pertanyaan-pertanyaan yang diberikan pada Buku Siswa tentang skema pengkodean permasalahan *knapsack*, dan menuliskan jawabannya pada lembar kerja (analisis mereka).

Kedua, mereka harus membuat sebuah program (dalam bahasa C/C++) yang membaca masukan berupa sebuah skema permasalahan *knapsack* dan kemudian menampilkan kembali deskripsi permasalahan tersebut sebagai keluaran.

(SAP-K11-19) Aktivitas PLB: Mengkodekan Permasalahan *Knapsack*

Membuat Skema Masukan Program

Jawaban dari pertanyaan-pertanyaan adalah:

1. Skema pengkodean untuk permasalahan pada Tabel 2 adalah:

6
3 8 5 4 10 8
6 4 5 6 5 10
25

2. Deskripsi permasalahan untuk pengkodean pada Contoh masukan 2 adalah:

Diberikan 8 buah barang pada tabel berikut:

Barang	A	B	C	D	E	F	G	H
Bobot (kg)	3	10	6	7	9	10	7	5
Nilai	1	10	8	1	7	8	9	18

Tentukan nilai maksimal yang dapat diperoleh dengan kapasitas maksimal 35 kg!

3. Deskripsi pengkodean: “Masukan dimulai dengan satu buah angka N yang menunjukkan banyaknya barang. Kemudian diikuti dengan 2 buah baris masing-masing terdiri dari N buah angka. Baris pertama menunjukkan bobot dari masing-masing barang, sedangkan baris kedua menunjukkan nilai-nilai dari masing-masing barang. Masukan diakhiri dengan satu buah baris berisi satu buah angka yang menunjukkan kapasitas maksimal”.
4. Sebuah representasi data yang diberikan bisa saja tidak valid. Misalnya: sebuah pengkodean masalah dimana terjadi ketidak-konsistenan antara banyaknya nilai N yang menunjukkan banyaknya barang, dan banyaknya angka yang menunjukkan bobot-bobot barang, atau antara banyaknya angka yang menunjukkan bobot barang dan banyaknya angka yang menunjukkan nilai barang.

Membaca Masukan dan Menyimpan Data

Untuk menjawab permasalahan kedua, yaitu membaca dan merepresentasikan masukan kode permasalahan *knapsack*, dapat digunakan program seperti berikut:

```
/* Membaca dan menyimpan data permasalahan knapsack */  
#include <vector>  
#include <algorithm>  
int main()  
{  
    int n;  
    scanf("%d", &n);  
    std::vector<float> bobot;  
    std::vector<float> nilai;  
    float kapasitas;  
    // Membaca daftar bobot barang  
    for(int i=0; i<n; i++)  
    {  
        float bobot_sekarang;
```

```

scanf("%f", &bobot_sekarang);
bobot.push_back(bobot_sekarang);
}
// Membaca daftar nilai barang
for(int i=0; i<n; i++)
{
float nilai_sekarang;
scanf("%f", &nilai_sekarang);
nilai.push_back(nilai_sekarang);
}
scanf("%f", &kapasitas);
// Menampilkan kembali data yang telah tersimpan
printf("Banyaknya barang = %d\n", n);
for(int i=0; i<n; i++)
{
printf("Barang ke-%d: Bobot = %f, Nilai = %f\n", i+1, bobot[i], nilai[i]);
}
printf("Kapasitas maksimal = %f kg", kapasitas);
return 0;
}

```

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi praktik lintas bidang yang telah dipelajari melalui aktivitas SAP-K11-18-U Aktivitas PLB: Memahami Permasalahan *Knapsack* dan aktivitas SAP-K11-19 Aktivitas PLB: Mengkodekan Permasalahan *Knapsack*.
- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

PERTEMUAN KE-12

Praktik Lintas Bidang 2 (5 JP)

Kegiatan Pendahuluan (10 Menit)

- Guru mengajak peserta didik menyampaikan refleksi mereka dari kegiatan PLB sebelumnya, apa saja yang sulit, apa saja yang menurut mereka telah mereka kerjakan dengan baik. Khususnya, guru hendaknya menanyakan kepada peserta didik apakah mereka sudah paham dengan baik permasalahan *knapsack*, serta apakah mereka telah dapat mengimplementasikan program membaca masukan data *knapsack* dengan baik. Selanjutnya, guru dapat menguji peserta didik dengan meminta mereka untuk membuat sebuah pengkodean permasalahan *knapsack*, lalu mengujikannya sebagai masukan pada program yang telah dibuat sebelumnya.

Kegiatan Inti (90 Menit)

- (5 menit) Kegiatan pembukaan, apersepsi, pemanasan

- (10 menit) Penjelasan tujuan pertemuan, kegiatan yang akan dilakukan, serta *subproblem* mana yang akan dikerjakan
- (45 menit) Pengerjaan aktivitas SAP-K11-20-U Aktivitas PLB: Merancang Algoritma Penyelesaian Masalah *Knapsack*
- (120 menit) Pengerjaan aktivitas SAP-K11-21 Aktivitas PLB: Mengimplementasikan dan Menguji Program Solusi *Knapsack*
- (35 menit) Pembahasan kedua aktivitas (presentasi kelompok dan diskusi)
- (10 menit) Kegiatan penutup dan refleksi

Pembahasan dan Jawaban Aktivitas

(SAP-K11-20-U) Aktivitas PLB: Merancang Algoritma Penyelesaian Masalah *Knapsack*

Untuk menyelesaikan permasalahan rational knapsack, dapat digunakan algoritma dengan strategi greedy sebagai berikut:

1. Urutkan barang berdasarkan rasio antara nilai dan bobot barang (secara menurun)
2. Inisialisasi total_bobot = 0
3. Inisialisasi total_nilai = 0
4. Mulai dari barang pertama (sesuai urutan pada langkah 1), kita melakukan hal sebagai berikut:
 - a. Jika total_bobot + bobot sekarang $\leq$ kapasitas
 - i. Total_bobot = total_bobot + bobot sekarang
 - ii. Total_nilai = total_nilai + nilai sekarang
 - b. Selainnya:
 - i. Hitung sisa kapasitas = kapasitas - total_bobot
 - ii. Hitung nilai rasio = sisa/bobot sekarang
 - iii. Total_nilai = total_nilai + rasio * nilai sekarang
5. Tampilkan total_nilai

Untuk 0-1 *knapsack*, strategi *greedy* tidak lagi dapat digunakan untuk menyelesaikannya. Hal ini karena pada 0 - 1 *knapsack*, kita harus mengambil barang secara penuh, dan tidak boleh secara parsial. Sehingga, urutan barang berdasarkan rasio nilai/bobot tidak selalu relevan untuk menentukan barang berikutnya yang harus dipilih. Dalam hal ini, strategi yang lebih sesuai untuk digunakan misalnya adalah menggunakan pemrograman dinamis.

(SAP-K11-21) Aktivitas PLB: Mengimplementasikan dan Menguji Program Solusi *Knapsack*

Perlu diketahui bahwa agar peserta didik dapat menyelesaikan permasalahan ini, mereka perlu mengenal bagaimana melakukan pengurutan data secara sederhana dengan C++. Hal ini dapat dilakukan dengan mudah menggunakan fungsi `sort()` yang telah disediakan oleh C++.

Program contoh yang diberikan pada Buku Siswa dapat dipakai oleh peserta didik sebagai teladan untuk mempelajari bagaimana menggunakan fungsi `sort()` pada vector. peserta didik dapat mengubah-ubah kode program untuk mempelajari lebih lanjut mengenai ini, misalnya bagaimana untuk mengubah urutan dari naik (*ascending*) menjadi menurun (*descending*).

Perlu ditekankan disini, bahwa tujuan dari latihan ini bukan untuk mempelajari algoritma pengurutan (*sorting*) dan bagaimana algoritma *sorting* diimplementasikan di dalam bahasa C/C++, namun cukup sekedar untuk mengetahui bagaimana menggunakannya dalam konteks implementasi solusi permasalahan rational *knapsack*.

Implementasi Program

Program yang menyelesaikan permasalahan rational knapsack dengan teknik greedy dapat diimplementasikan sebagai berikut:

```
/* Menyelesaikan permasalahan rational knapsack dengan pendekatan
```

```
algoritma greedy */
```

```
#include <vector>
```

```
#include <algorithm>
```

```
int main()
```

```
{
```

```
int n;
```

```
// Membaca masukan
```

```
scanf("%d", &n);
```

```
std::vector<float> bobot;
```

```
std::vector<float> nilai;
```

```
float kapasitas;
```

```
for(int i=0; i<n; i++)
```

```
{
```

```
float bobot_sekarang;
```

```
scanf("%f", &bobot_sekarang);
```

```
bobot.push_back(bobot_sekarang);
```

```
}
```

```
for(int i=0; i<n; i++)
```

```
{
```

```
float nilai_sekarang;
```

```
scanf("%f", &nilai_sekarang);
```

```
nilai.push_back(nilai_sekarang);
```

```
}
```

```
scanf("%f", &kapasitas);
```

```
// Menghitung rasio nilai/bobot
```

```
std::vector<float> nilai_per_bobot;
```

```
for(int i=0; i<n; i++)
```

```
{
```

```
nilai_per_bobot.push_back(nilai[i]/bobot[i]);
```

```
}
```

```
// Mengurutkan barang berdasarkan nilai/bobot (menurun)
```

```
std::vector<int> index;
```

```
for(int i=0; i<n; i++)
```

```
index.push_back(i);
```

```
sort(index.begin(), index.end(), [&nilai
```

```
per_bobot](int index1, int index2)
```

```
{
```

```
return nilai_per_bobot[index1] > nilai_per_bobot[index2];
```

```
}
```

```
);
```

```

float total_bobot_terpilih = 0;
float total_nilai_terpilih = 0;
// Memilih barang sampai mendapatkan nilai optimal
for (int i=0; i<n; i++)
{
float bobot_sekarang = bobot[index[i]];
if (total_bobot_terpilih + bobot_sekarang <= kapasitas)
{
total_bobot_terpilih += bobot_sekarang;
total_nilai_terpilih += nilai[index[i]];
}
else
{
float sisa_kapasitas = kapasitas - total_bobot_terpilih;
float rasio_bobot = sisa_kapasitas/bobot_sekarang;
total_bobot_terpilih = kapasitas;
total_nilai_terpilih += nilai[index[i]] * rasio_bobot;
break;
}
}
printf("%f\n", total_nilai_terpilih);
return 0;
}

```

Perlu dipahami dan diingat bahwa kode program jawaban yang benar dari peserta didik bisa jadi berbeda dan tidak sama persis dengan kode program di atas. Yang penting adalah bahwa program tersebut menghasilkan keluaran yang benar ketika diberikan sebuah kasus uji. Hal ini yang harus dipastikan pada bagian pengujian program.

Pengujian Program

Pada bagian ini, peserta didik bekerja berpasangan untuk menghasilkan beberapa (misal: 10) kasus uji untuk permasalahan *rational knapsack*. Masing-masing kasus uji berisi satu buah pasang masukan dan keluaran.

Untuk memeriksa dan menentukan apakah setiap kasus uji benar atau tidak, guru dapat melakukan dua langkah sebagai berikut:

1. Memastikan bahwa masukan yang diberikan adalah **valid dan sesuai dengan skema pengkodean** pada Aktivitas PLB sesuai tabel pengujian 2. 23 pada Buku Siswa.
2. Memastikan bahwa keluaran yang diberikan sesuai dengan yang diharapkan.

Hal ini dapat diperiksa dengan menjalankan program pada kunci jawaban Implementasi Program di atas dengan memberikan masukan dari kasus uji tersebut, dan memeriksa apakah keluaran pada kasus uji **sesuai** dengan keluaran dari program.

Kegiatan Penutup (10 Menit)

- Guru memberikan penguatan pemahaman tentang materi praktik lintas bidang yang telah dipelajari melalui aktivitas SAP-K11-20-U Aktivitas PLB: Merancang Algoritma

Penyelesaian Masalah *Knapsack* dan aktivitas SAP-K11-21 Aktivitas PLB:

Mengimplementasikan dan Menguji Program Solusi *Knapsack*.

- Kemudian guru memberi motivasi kepada peserta didik agar dapat meningkatkan pemahaman materi dengan berlatih dan mempelajari berbagai sumber belajar lainnya yang relevan serta mendorong untuk membaca materi yang hendak dipelajari pada pertemuan berikutnya.

E. ASESMEN / PENILAIAN

Asesmen kemampuan peserta didik dalam Berpikir Komputasional dilakukan terhadap proses dan hasil pelaksanaan aktivitas yang diberikan. Terdapat tiga jenis asesmen, yaitu:

1. Asesmen terhadap Aktivitas **Ayo Merancang Program!** untuk mengukur kompetensi peserta didik dalam menganalisis domain permasalahan, memformulasikan *problem*, dan merancang solusi informatika menggunakan kemampuan berpikir komputasional yang siap untuk diimplementasikan dalam bentuk program. Asesmen dilakukan terhadap model sederhana yang memberikan representasi permasalahan serta rancangan solusi dalam bentuk algoritma.
2. Asesmen terhadap **Ayo Buat Program** untuk mengukur kompetensi peserta didik dalam mengimplementasikan solusi yang telah dirancang dalam bentuk program, serta memelihara dan menyempurnakannya. Asesmen dilakukan terhadap program yang diimplementasikan dalam bahasa pemrograman tertentu, serta dokumentasi pengujian terhadap program tersebut.
3. Asesmen terhadap kemampuan **bekerja sama dan gotong royong** dalam menyelesaikan aktivitas. Rubrik penilaian dari ketiga aktivitas tersebut diberikan sebagai berikut ini: Rubrik Penilaian Aktivitas Ayo Merancang Program!

Tabel 2.6 Rubrik Penilaian Aktivitas Ayo Merancang Program Unit Materi SAP

Indikator	Sangat Baik = 4	Baik = 3	Cukup = 2	Kurang = 1
Kemampuan memahami dan menganalisis permasalahan	Peserta didik mampu menjawab sebagian besar/semua pertanyaan pada Aktivitas PLB 1 (konsep optimasi pada permasalahan <i>knapsack</i>) dan mampu mencari penyelesaian untuk soal kasus <i>knapsack</i> sederhana, baik versi <i>rational</i> maupun <i>knapsack</i>	Peserta didik dapat menjawab sebagian besar pertanyaan pada Aktivitas PLB 1 (konsep optimasi pada permasalahan <i>knapsack</i>) dan mampu mencari penyelesaian untuk minimal 1 soal kasus <i>knapsack</i> sederhana, misalnya <i>rational knapsack</i>	Peserta didik mampu menjawab pertanyaan 1 - 4 dari PLB 1 (konsep optimasi pada permasalahan <i>knapsack</i>) namun ia belum mampu mencari solusi dari sebuah permasalahan <i>knapsack</i> secara mandiri	Peserta didik masih memiliki kesalahan dalam menjawab pertanyaan 1 - 4 PLB 1 serta belum dapat menjawab pertanyaan 5 dan 6 dengan benar
Kemampuan Menyusun Strategi Penyelesaian	Peserta didik dapat menyusun algoritma yang dapat menyelesaikan	Peserta didik dapat menyusun algoritma yang dapat menyelesaikan	Peserta didik tidak dapat menyusun algoritma yang eksak untuk	Peserta didik tidak memiliki pemahaman yang cukup untuk dapat

	permasalahan dengan benar dan dapat menjelaskan mengapa algoritma tersebut benar menjawab permasalahan <i>knapsack</i>	permasalahan, namun tidak dapat memberikan penjelasan yang memadai mengapa algoritma tersebut benar	menyelesaikan permasalahan dengan benar, namun dapat dengan tepat membedakan mana solusi yang benar dan mana yang tidak	mencari solusi permasalahan, atau pun mengenali mana solusi yang benar dari yang salah
Komunikasi Hasil Pekerjaan Secara Non-Verbal	Peserta didik dapat menyajikan rancangan algoritma yang telah mereka buat dalam bentuk yang sesuai dengan contoh kaidah penulisan algoritma dengan menggunakan notasi pseudocode yang telah dipelajari di Kelas X, sehingga mudah untuk dipahami dan dijalankan oleh orang lain	Peserta didik dapat menyajikan hasil analisis dan algoritma yang telah mereka buat dalam bentuk yang tidak formal dan seperti yang dicontohkan di kelas X, namun masih dapat dipahami dan dijalankan oleh orang lain.	Peserta didik dapat tidak memiliki pengetahuan yang cukup mengenai penulisan algoritma dalam bentuk <i>pseudocode</i> sehingga algoritma yang disajikan masih dalam bentuk yang sulit dipahami oleh orang lain.	Peserta didik hanya mampu menjelaskan algoritma yang dirancangnya secara verbal

Rubrik Penilaian Aktivitas Ayo Buat Program!

Tabel 2.7 Rubrik Penilaian Aktivitas Ayo Buat Program Unit Materi SAP

Indikator	Sangat Baik = 4	Baik = 3	Cukup = 2	Kurang = 1
Praktik Baik Pemrograman	Peserta didik menerapkan hampir semua (>80%) praktik baik pemrograman yang telah dipelajari di Kelas X	Peserta didik menerapkan sebagian besar (>50%) praktik baik pemrograman yang telah dipelajari di Kelas X	Peserta didik menerapkan sedikit (<50%) praktik baik pemrograman yang telah dipelajari di Kelas X	Peserta didik tidak menerapkan praktik baik pemrograman.
Ketepatan Program	Program dapat memberikan keluaran yang benar untuk semua kasus uji yang diberikan	Program dapat memberikan keluaran yang benar untuk sebagian besar	Program dapat memberikan keluaran yang benar untuk sebagian (>	Program tidak berhasil memberikan keluaran yang benar untuk sebagian besar

		(>80%) kasus uji yang diberikan	50%) kasus uji yang diberikan	kasus uji yang diberikan
Ketepatan Pengujian	Kasus uji yang diberikan semuanya benar	Kasus uji yang diberikan hampir semuanya (> 80%) benar	Kasus uji yang diberikan sebagian besar (> 50%) benar	Kasus uji yang diberikan sebagian besar salah
Integritas	Peserta didik membuat program dengan jujur dan tidak melakukan plagiasi.	Peserta didik membuat program dengan adanya plagiasi pada sebagian kecil kode program.		Peserta didik membuat program dengan adanya plagiasi pada sebagian besar atau semua kode program.

Rubrik Penilaian Kemampuan Kerja Sama dan Gotong Royong

Tabel 2.8 Rubrik Penilaian Kemampuan Kerja Sama unit materi SAP

Indikator	Sangat Baik = 4	Baik = 3	Cukup = 2	Kurang = 1
Pembagian Tugas	Peserta didik membagi tugas dengan jelas dalam kelompok pasangannya dan masing-masing melakukan tugas sesuai pembagian tersebut	Peserta didik membagi tugas meskipun kurang rinci, dan masing-masing berusaha menjalankan tugasnya masing-masing	Pembagian tugas antar peserta didik di dalam kelompok/ pasangan tidak begitu jelas, namun masing-masing masih memiliki peranan	Peserta didik tidak melakukan pembagian tugas sebelum mengerjakan, atau salah satu tidak berperan aktif dalam pengerjaan tugas
Komunikasi di dalam Kelompok	Peserta didik berkomunikasi dengan baik dan melakukan pencatatan hasil diskusi	Peserta didik mampu berkomunikasi meskipun tidak tercatat	Peserta didik berkomunikasi namun tidak terstruktur dan jarang	Peserta didik tidak dapat berkomunikasi dengan baik dalam kelompoknya

F. PENGAYAAN DAN REMEDIAL

G. REFLEKSI GURU DAN PESERTA DIDIK

Setelah melalui beberapa bab pembelajaran, guru dapat melakukan refleksi bersama tim pengajar (jika ada). Guru juga dapat melibatkan peserta didik untuk menggali minat mereka terhadap topik pembelajaran atau aktivitas yang telah dilakukan. Berikut adalah contoh pertanyaan refleksi yang bisa dilakukan bersama, baik bersama tim guru maupun peserta didik.

- Berdasarkan bab/aktivitas pembelajaran yang sudah berlangsung, bab/aktivitas mana yang memiliki respon paling positif dan respon paling negatif di kelas? Dan mengapa bisa bab/aktivitas tersebut?
- Bab/aktivitas mana yang paling anda kuasai?
- Pada bab/aktivitas apa anda merasa kreatif ketika mengajar?
- Jika anda harus bercerita kepada rekan sesama guru mengenai hal yang telah anda pelajari selama mengajar, apa yang akan anda ceritakan?
- Jika anda dapat bertanya kepada rekan guru Informatika di sekolah/ kelompok kerja guru Informatika/komunitas guru Informatika mengenai kegiatan pembelajaran Informatika, hal apa yang ingin anda tanyakan?
- Jika anda memiliki kesempatan untuk mengulang bab/topik yang sudah berlangsung, topik apa yang menurut anda perlu disampaikan lagi dengan cara yang berbeda?

Apakah anda berdiskusi dan berkolaborasi dengan guru bidang lain untuk menggali kemungkinan praktik Strategi Algoritmik dan Pemrograman sehingga akan melahirkan proyek menarik di kemudian hari?

LAMPIRAN 1

LEMBAR KERJA PESERTA DIDIK (LKPD)

Aktivitas Individu

Aktivitas SAP-K11-01-U: Mengamati Evolusi Program

Deskripsi Tugas

Pada bagian ini telah dijelaskan bahwa suatu program dikembangkan setahap demi setahap dalam suatu siklus. Hal ini juga berlaku pada program, baik berbentuk aplikasi atau web yang kalian gunakan. Sekarang pilihlah satu aplikasi atau web yang sering kalian gunakan, kemudian buatlah infografis linimasa (timeline) sederhana yang menunjukkan proses perubahan program seiring dengan berjalannya waktu. Kalian dapat melakukan pencarian untuk menemukan versi lama dari program. Untuk web, kalian dapat menggunakan aplikasi web seperti web.archive.org yang dapat menyimpan versi lama dari web secara periodik.

Aktivitas Individu

Aktivitas SAP-K11-02-U: Memahami Relasi Rekurensi

Deskripsi Tugas

Relasi rekurensi (*recurrence relation*) adalah sebuah tipe relasi matematis dimana definisi dari sebuah fungsi atau barisan dinyatakan secara rekursif, artinya merujuk pada fungsi atau barisan itu sendiri. Pada bagian ini, kalian akan berlatih untuk memahami definisi relasi rekurensi dan bagaimana menerapkannya, serta membuat definisi rekursif dari sebuah permasalahan.

1. Tentukan suku ke-10 dari barisan yang didefinisikan sebagai berikut:

$$a_i = \begin{cases} 1 & \text{jika } i = 1 \text{ atau } i = 2 \\ 2a_{i-1} + 2a_{i-2} & \text{jika } i > 2 \end{cases}$$

atau dengan kata lain, barisan tersebut dimulai dengan nilai 1, 1, kemudian untuk menghitung suku berikutnya, kita jumlahkan antara suku sebelumnya dengan dua kali dari suku sebelum suku sebelumnya.

2. Faktorial dari sebuah bilangan bulat $n \geq 1$, ditulis sebagai $n!$, didefinisikan sebagai sebuah nilai yang dihitung dengan mengalikan semua bulat dari 1 sampai dengan n . Sebagai contoh, faktorial dari 5 adalah $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$. Buatlah sebuah definisi rekursif untuk menghitung nilai $n!$.

Aktivitas Individu

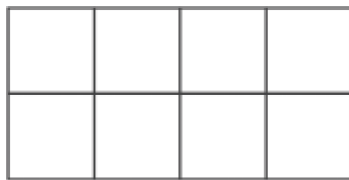
Aktivitas SAP-K11-03-U: Menerapkan Konsep Rekursi

Deskripsi Tugas

Selesaikanlah dua problem berikut dengan menerapkan konsep rekursi yang telah kalian pelajari. Setelah mengerjakan problem tersebut, diskusikanlah solusi kalian dengan teman.

Permasalahan 1: Memasang Keramik

Terdapat sebuah lantai yang berukuran $2 \times N$. Pada lantai tersebut, ingin dipasang N buah keramik, yang masing-masing berukuran 1×2 (perhatikan ilustrasi pada gambar 2.6).



Lantai



Keramik

Setiap keramik dapat dipasang secara mendatar (horizontal) ataupun secara tegak (vertikal). Sebagai contoh, jika $N=4$, maka akan ada 5 buah cara berbeda memasang 4 keramik, yaitu:

- Semua keramik dipasang secara horizontal.
- Semua keramik dipasang secara vertikal.
- Paling kiri ada 2 keramik vertikal, sisanya horizontal.
- Paling kiri ada 2 keramik horizontal, sisanya vertikal.
- Paling kiri dan paling kanan vertikal, sisanya horizontal.
- Tentukan ada berapa cara memasang keramik untuk $N=8$?

Permasalahan 2: Menumpuk Panekuk

Budi memiliki setumpuk panekuk yang memiliki ukuran dari besar sampai kecil. Panekuk-panekuk tersebut ditumpuk di atas sebuah piring, dengan aturan bahwa panekuk yang besar harus berada di bawah panekuk yang lebih kecil.



Budi ingin memindahkan panekuk ini dari satu piring ke piring lainnya, namun dalam prosesnya ia tetap ingin mengikuti aturan bahwa panekuk yang besar harus selalu berada di bawah panekuk yang lebih kecil. Selain itu, Budi juga hanya boleh memindahkan satu buah panekuk saja, pada satu waktu tertentu, dari satu piring ke piring lainnya. Budi menyadari bahwa ia memerlukan sebuah piring tambahan untuk dapat melakukan perpindahan ini. Jika piring asal panekuk diberi nama A, piring tujuan diberi nama C, maka Budi akan menyiapkan sebuah piring bantuan sebagai tempat sementara, yang diberi nama piring B.

Budi ingin mengetahui, berapa banyak langkah pemindahan panekuk yang harus ia lakukan, untuk dapat memindahkan semua panekuk yang dimilikinya, dari piring A ke piring C (dengan mungkin menggunakan piring B sebagai tempat sementara). Misalnya, jika Budi memiliki 2 buah panekuk saja (diberi nama panekuk 1 dan, dimana panekuk 1 berukuran lebih kecil dari panekuk 2), maka ia akan membutuhkan minimal 3 langkah pemindahan:

1. Pindahkan panekuk 1 dari piring A ke piring B
2. Pindahkan panekuk 2 dari piring A ke piring C
3. Pindahkan panekuk 1 dari piring B ke piring C

Berapakah jumlah langkah minimal yang diperlukan apabila Budi memiliki 6 buah panekuk?

Aktivitas Kelompok

Aktivitas SAP-K11-04-U: Mengerjakan Pekerjaan Rumah (PR)

Deskripsi Tugas

Cici menerima 10 buah pekerjaan rumah (PR) yang harus ia kerjakan. Setelah melihat isi dari masing-masing PR, Cici memiliki perkiraan, berapa lama waktu yang diperlukan untuk mengerjakan masing-masing PR tersebut, seperti terlihat pada tabel di bawah.

No.	PR	Waktu pengerjaan (jam)	No.	PR	Waktu pengerjaan (jam)
1.	A	1.5	6.	F	1
2.	B	3	7.	G	2,5
3.	C	1	8.	H	1
4.	D	0,5	9.	I	0,5
5.	E	4	10.	J	2

Sayangnya, ia tidak punya banyak waktu untuk mengerjakan semua PR. Cici menghitung bahwa ia hanya punya waktu total = 8 jam sebelum semua PR tersebut harus dikumpulkan. Cici ingin menentukan, PR mana yang harus ia kerjakan terlebih dahulu, dengan pertimbangan bahwa setiap PR memiliki nilai yang sama besarnya (terhadap nilai akhir Cici). Bantulah Cici menentukan PR yang mana saja yang harus ia kerjakan dalam waktu maksimal 8 jam, untuk mendapatkan total nilai akhir yang sebesar-besarnya.

Aktivitas Individu

Aktivitas SAP-K11-05-U: Mengunjungi Kebun Binatang

Deskripsi Tugas

Dina sedang bertamasya mengunjungi kebun binatang. Setiap hari, kebun binatang mengadakan beberapa pertunjukan atraksi hewan yang dapat ditonton oleh para pengunjung. Berikut adalah jadwal yang telah ditetapkan oleh pengelola kebun binatang.

No.	Waktu Mulai	Waktu Selesai	Pertunjukan/Atraksi Hewan
1.	9:15	10:30	Orang Utan
2.	8:00	9:30	Penguin
3.	10:00	12:00	Harimau
4.	13:00	14:30	Beruang Madu
5.	11:00	12:30	Burung Pemangsa
6.	14:00	15:00	Buaya
7.	15:30	16:30	Panda
8.	16:00	17:00	Ular Piton
9.	15:00	15:30	Singa
10.	15:30	16:00	Anjing Laut

Tentunya dalam satu waktu tertentu, Dina hanya dapat menonton satu pertunjukan atraksi hewan. Dina ingin dapat melihat sebanyak-banyaknya pertunjukan dalam satu hari tersebut, dan ia tidak memiliki preferensi dalam melihat pertunjukan hewan (semuanya ia anggap sama menariknya). Tentukan, ada berapa banyak maksimal pertunjukan yang dapat ditonton oleh Dina?

Aktivitas Individu

Aktivitas SAP-K11-06-U: Menukarkan Uang

Deskripsi Tugas

Dalam kehidupan sehari-hari, kita pasti sudah banyak terbiasa dengan perhitungan yang melibatkan uang. Misalnya, ketika Anda membeli sebuah barang/makanan, atau pun ingin membayar untuk sebuah jasa tertentu, kita seringkali menyiapkan sejumlah uang tertentu, sesuai dengan harga barang atau jasa tersebut. Selanjutnya, bagi penjual atau penyedia jasa, apabila mereka menerima uang pembayaran dengan jumlah total yang lebih besar dari harga yang ditetapkan, mereka pun juga harus menyiapkan uang kembalian sesuai dengan jumlah kelebihan pembayaran. Di Indonesia, mata uang rupiah memiliki beberapa pecahan uang, mulai dari yang terkecil, 100 Rupiah, 200 Rupiah, 500 Rupiah, dst sampai dengan 100 000 Rupiah. Seandainya kita memiliki sejumlah pecahan uang, misalnya beberapa uang seribuan, dua ribuan, lima ribuan, sepuluh ribuan dan dua puluh ribuan. Jika kita ingin mendapatkan uang tepat sejumlah 38 ribu rupiah, maka kita dapat memilih beberapa cara, misalnya:

- 3 lembar sepuluh ribuan, ditambah 1 lembar lima ribuan, ditambah 2 lembar ribuan, ditambah 2 koin lima ratus, dengan total ada 8 buah lembaran uang/koin
- 1 lembar dua puluh ribuan, ditambah 1 lembar sepuluh ribuan ditambah 4 lembar dua ribuan, totalnya menjadi 6 lembaran uang
- 1 lembar dua puluh ribuan, ditambah 1 lembar sepuluh ribuan, ditambah 1 lembar lima ribuan ditambah 1 lembar dua ribuan, ditambah 1 lembar seribuan, dengan total ada 5 lembaran uang.

Jelas bahwa jumlah total lembaran yang dibutuhkan tergantung dari pemilihan pecahan uang yang kita gunakan. Nah, permasalahan yang mungkin kita tanyakan adalah: Bagaimana caranya memilih pecahan-pecahan uang yang akan digunakan sedemikian rupa, sehingga total lembaran yang diperlukan untuk menghasilkan suatu nilai uang tertentu menjadi sekecil mungkin?

Pada contoh di atas, dapat diperiksa bahwa untuk menghasilkan nilai uang sebesar tiga puluh delapan ribu rupiah dari pecahan-pecahan seribuan, dua ribuan, lima ribuan, sepuluh ribuan dan dua puluh ribuan, maka diperlukan minimal 5 buah lembar, yaitu sesuai dengan cara terakhir di atas. Dapatkah Anda mencari strategi yang umum untuk menyelesaikan permasalahan serupa, jika jumlah nilai uang yang dihasilkan berbeda (namun dengan pecahan-pecahan uang yang sama)? Kita bisa menganggap bahwa jumlah nilai yang diinginkan selalu merupakan kelipatan ribuan rupiah (sehingga selalu bisa didapatkan dengan menggabungkan pecahan-pecahan di atas).

Aktivitas Berpasangan

Aktivitas SAP-K11-07-U: Bermain Angka

Deskripsi Tugas

Ani dan Budi sedang bermain dengan sebuah permainan angka: pertama Ani akan memilih sebuah angka bilangan bulat positif n . Selanjutnya, Budi harus mengubah bilangan n ini menjadi angka 1 dengan menerapkan serangkaian langkah sebagai berikut:

1. Budi boleh mengganti bilangan n dengan $n - 1$.
2. Jika bilangan saat ini adalah genap (habis dibagi 2), maka Budi boleh menggantinya dengan $n/2$.
3. Jika bilangan saat ini habis dibagi 3, maka Budi boleh menggantinya dengan $n/3$.

Proses ini harus dilakukan oleh Budi secara terus menerus sampai bilangan yang dimilikinya menjadi 1. Misalnya, jika Ani memilih $n = 5$, maka Budi dapat melakukan proses mengubah 5

menjadi 1 sebagai berikut: $5 \rightarrow 4 \rightarrow 2 \rightarrow 1$ (dalam tiga langkah). Tentukan, berapakah jumlah langkah minimum yang diperlukan, jika Ani memilih $n = 25$?

Aktivitas Individu

Aktivitas SAP-K11-08: Latihan Larik

1. Rancang dan buatlah sebuah program yang membaca N buah bilangan dan mencetaknya secara terbalik. Misalnya jika diberikan masukan 1 2 3 4 5 6, program akan mencetak 6 5 4 3 2 1.
2. Rancang dan buatlah sebuah program yang membaca sebuah matriks berukuran $N \times M$ dan mencetak hasil *transpose* matriksnya.
3. Modifikasilah kode program untuk menghitung jarak dua kota yang menerima suatu rute. Misalnya, jika rute yang dimasukkan adalah Kota A - Kota B - Kota C, maka nilai total jarak yang dikeluarkan adalah $10 + 8 = 18$ km.
4. Tantangan larik: perhatikan bahwa pada Gambar 2.9 terdapat data yang berganda. Hal ini dikarenakan matriks tersebut menyimpan informasi jarak antara 2 kota (misal A dan B) sebagai jarak dari A-B dan B-A. Dapatkan kalian menemukan representasi yang lebih baik daripada contoh tersebut sehingga tidak ada duplikasi informasi dalam penyimpanan data jarak kota?

Aktivitas Individu

Aktivitas SAP-K11-09: Latihan Karakter dan String

1. Rancang dan buatlah sebuah program yang dapat membaca sebuah string dan mencetaknya secara terbalik.
2. Rancang dan buatlah sebuah program yang membaca sebuah kata sandi dan mencetak jumlah karakter yang berupa angka, huruf, dan simbol. Bedakan huruf kapital dan non kapital.

Aktivitas Berpasangan

Aktivitas SAP-K11-10-U: Merancang Algoritma Simulasi Burung

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang simulasi burung yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam Buku Kerja kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Aktivitas Berpasangan

Aktivitas SAP-K11-11: Membuat Program Simulasi Burung

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Aktivitas Berpasangan

Aktivitas SAP-K11-12-U: Merancang Algoritma Pengelolaan Donor Darah

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang pengelolaan donor darah yang menjadi domain permasalahan yang diberikan.

Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam Buku Kerja kalian. Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Aktivitas Berpasangan

Aktivitas SAP-K11-13: Membuat Program Pengelolaan Donor Darah

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Aktivitas Berpasangan

Aktivitas SAP-K11-14-U: Merancang Algoritma Persilangan Tanaman

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang persilangan tanaman yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam Buku Kerja kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Aktivitas Berpasangan

Aktivitas SAP-K11-15: Membuat Program Persilangan Tanaman

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat

memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Aktivitas Berpasangan

Aktivitas SAP-K11-16-U: Merancang Algoritma Simulasi Stoikiometri

Berdasarkan deskripsi permasalahan di atas, secara individu, definisikanlah permasalahan dan rancanglah algoritma solusi dari permasalahan tersebut. Kalian dapat membuka kembali bahan belajar yang terkait tentang stoikiometri yang menjadi domain permasalahan yang diberikan. Dokumentasikanlah setiap langkah yang kalian kerjakan, termasuk apa yang kalian hasilkan dalam **Buku Kerja** kalian.

Setelah kalian selesai merancang algoritma, secara berpasangan, saling tukarkan algoritma kalian. Setelah itu, telusurilah algoritma teman kalian dan cek apakah algoritma tersebut sudah benar atau belum. Apabila belum benar, secara bersama-sama, diskusikanlah apa yang dapat diperbaiki dari rancangan algoritma kalian. Jangan lupa untuk membandingkan solusi yang telah kalian hasilkan. Apabila solusi kalian berbeda, tapi sama-sama menghasilkan jawaban yang benar, bandingkanlah kedua solusi tersebut.

Aktivitas Berpasangan

Aktivitas SAP-K11-17: Membuat Program Simulasi Stoikiometri

Sekarang, secara individu, implementasikanlah algoritma yang telah kalian rancang dalam bentuk program dengan menggunakan bahasa pemrograman yang telah kalian kuasai. Sebelum program kalian kompilasi, secara berpasangan, saling tukarkan kode program kalian dan cek apakah kode program tersebut sudah ditulis dengan benar. Setelah itu, kompilasi kode tersebut menjadi program, dan ujilah program teman kalian dengan kasus uji yang kalian rancang. Apabila program teman kalian belum menghasilkan jawaban yang benar, sampaikanlah kepada teman kalian agar ia dapat memperbaiki kode program tersebut hingga menghasilkan jawaban yang benar. Setelah selesai, presentasikanlah hasil kerja kalian di depan kelas, mengikuti petunjuk dari guru.

Aktivitas Berpasangan

Aktivitas SAP-K11-18-U: Memahami Permasalahan Knapsack

Jawablah pertanyaan-pertanyaan berikut dan tuliskan dalam laporan analisis.

1. Apakah jenis optimasi pada permasalahan *knapsack*? Apakah mencari minimum, ataukah maksimum? Jelaskan!
2. Tentukan apa yang menjadi fungsi tujuan dari permasalahan *knapsack*!
3. Tentukan apa yang menjadi kendala pada optimasi untuk permasalahan *knapsack*!
4. Perhatikan permasalahan knapsack yang ditunjukkan oleh Tabel 2.20 berikut. Diberikan 6 buah barang, A, B, s/d F dengan bobot dan nilai sebagai berikut:

Tabel 2.20 Keterangan Bobot dan Nilai dari 6 Barang (*Knapsack Problem*) Aktivitas PLB

Barang	A	B	C	D	E	F
Bobot	3	8	5	4	10	8
Nilai	6	4	5	6	5	10

Asumsikan bahwa tas memiliki kapasitas maksimal = 24 kg.

- a. Apakah pilihan mengambil barang-barang B, D, E dan F diperbolehkan sebagai solusi sesuai dengan kendala optimasi pada permasalahan tersebut? Mengapa?

- b. Apakah pilihan mengambil barang-barang A, D, E diperbolehkan sebagai solusi sesuai dengan kendala optimasi pada permasalahan tersebut? Apakah fungsi tujuan mencapai nilai optimal dengan memilih A, D dan E saja? Mengapa?
5. Tentukan jawaban permasalahan knapsack tersebut pada soal no. 4, jika menggunakan variasi permasalahan rational *knapsack*!
6. Pada soal no. 4, apakah solusinya, jika digunakan variasi 0-1 knapsack? Apakah sama dengan solusi untuk variasi rational *knapsack*?

Aktivitas Berpasangan

Aktivitas SAP-K11-19: Mengkodekan Permasalahan Knapsack

Membuat Skema Masukan Program

Sebelum kita dapat menyelesaikan permasalahan di atas dengan menggunakan sebuah program, hal pertama yang perlu kita lakukan adalah menentukan bagaimana menyatakan sebuah kasus permasalahan *knapsack* menjadi sebuah bentuk/format yang mudah dibaca dan diolah oleh komputer. Cara yang paling mudah adalah dengan menuliskan besaran-besaran yang ada pada permasalahan tersebut dan menyatakannya dalam *arrayarray* atau deretan bilangan yang sesuai. Kita dapat memulai dengan menuliskan banyaknya barang yang tersedia, kemudian diikuti dengan bobot masing-masing barang, kemudian diikuti dengan nilai masing-masing barang, dan terakhir kita tuliskan kapasitas maksimal dari tas. Sebagai contoh, untuk kasus *knapsack* dimana barang-barang yang akan dimasukkan pada tas ditunjukkan pada Tabel 2.21 berikut:

Tabel 2.21 Keterangan Bobot dan Nilai dari 5 Barang (*Knapsack Problem*) Aktivitas PLB

Barang	A	B	C	D	E
Bobot	5	4	7	8	10
Nilai	10	5	7	12	8

Dan dimana jumlah maksimal bobot yang dapat ditampung adalah 20 kg, maka kita dapat merepresentasikan permasalahan ini menjadi bentuk sebagai berikut:

Contoh masukan 1

5

5 4 7 8 10

10 5 7 12 8

20

Sekarang perhatikan permasalahan pada Tabel 2.22 berikut:

Tabel 2.22 Keterangan Bobot dan Nilai dari 6 Barang (*Knapsack Problem*) Aktivitas PLB

Barang	A	B	C	D	E	F
Bobot	3	8	5	4	10	8
Nilai	6	4	5	6	5	10

Dimana kapasitas maksimal yang dapat ditampung adalah 25 kg. Sekarang perhatikan juga skema masukan pada contoh berikut:

Contoh masukan 2

8

3 10 6 7 9 10 7 5

Jawablah pertanyaan-pertanyaan berikut pada lembar jawaban/laporan PLB!

Pertanyaan:

1. Tentukan pengkodean yang sesuai untuk permasalahan yang ditunjukkan pada Tabel 2.22 di atas, dengan menggunakan skema pengkodean yang telah dijelaskan. Jelaskan!
2. Tuliskan/jelaskan pada laporan analisis kalian, deskripsi permasalahan untuk contoh masukan 2 di atas!
3. Jelaskan dalam satu paragraf, skema pengkodean di atas, agar dapat dipahami oleh orang lain!
4. Mungkinkah sebuah representasi data tidak valid/tidak sesuai? Berikan contohnya, dan tuliskan penjelasannya pada laporan analisis!

Aktivitas Berpasangan

Aktivitas SAP-K11-20-U: Merancang Algoritma Penyelesaian Masalah Knapsack

Pada bagian Berpikir Komputasional, kita sudah mempelajari beberapa jenis teknik penyelesaian masalah. Untuk permasalahan *knapsack* pada PLB ini, kita harus menentukan strategi penyelesaian masalah yang manakah yang sesuai untuk diterapkan.

1. Untuk permasalahan *rational knapsack*, tentukan apakah strategi *greedy* ataukah *dynamic programming* yang sesuai untuk diterapkan? Jelaskan pada laporan analisis kamu, bagaimana strategi *greedy* atau *dynamic programming* dapat diterapkan pada permasalahan *rational knapsack*!
2. Untuk permasalahan 0–1 *knapsack*, tentukan apakah strategi *greedy* ataukah *dynamic programming* yang sesuai untuk diterapkan?
3. Tuliskan dalam notasi *pseudocode* algoritma yang sesuai untuk menyelesaikan permasalahan *rational knapsack* menggunakan strategi yang Anda pilih pada bagian nomor 1!
4. **[Opsional]** Tuliskan dalam notasi *pseudocode* algoritma yang sesuai untuk menyelesaikan permasalahan 0-1 *knapsack* menggunakan strategi yang Anda pilih pada bagian nomor 2!

Aktivitas Berpasangan

Aktivitas SAP-K11-21: Mengimplementasikan dan Menguji Program Solusi Knapsack

Implementasi program

Pada bagian ini, setiap kelompok akan membuat program yang menyelesaikan permasalahan *rational knapsack*. Program yang dibuat memiliki spesifikasi (ketentuan) sebagai berikut:

1. Program membaca masukan berupa representasi pengkodean aksus bseagaimana ijdelaskan i d at.a s
2. Kemudian program tersebut menjalankan strategi penyelesaian program sebagaimana telah dirancang pada *pseudocode* pada bagian sebelumnya.
3. Program kemudian menghasilkan keluaran berupa sebuah angka yang menunjukkan berapa nilai total terbesar yang dapat dicapai (nilai optimal) untuk kasus yang diberikan pada masukan.

LAMPIRAN 2

BAHAN BACAAN GURU DAN PESERTA DIDIK

GLOSARIUM

Algoritma, (*algorithm*) suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan dalam bentuk program komputer untuk menyelesaikan suatu permasalahan komputasi tertentu.

Algoritma Greedy, (*greedy algorithm*) setiap algoritma yang berusaha mencapai solusi suatu permasalahan dengan membuat pilihan lokal yang optimal pada setiap tahap.

Analisis Data (*data analytics*), proses inspeksi, pembersihan dan pemodelan data dengan tujuan menemukan informasi yang berguna, menginformasikan kesimpulan dan mendukung pengambilan keputusan.

Aplikasi Desktop (*desktop application*) , perangkat lunak yang dibuat untuk dapat dijalankan pada komputer bertipe *desktop*.

Aplikasi Mobile (*mobile application*) , perangkat lunak yang dibuat untuk dapat dijalankan pada perangkat bergerak.

Aplikasi Web (*Web Application*) , perangkat lunak yang dapat dijalankan pada suatu server dan dapat dijalankan di menggunakan peramban web.

App Inventor, adalah lingkungan pemrograman visual yang intuitif yang memungkinkan semua orang, bahkan anak-anak, untuk membangun aplikasi yang berfungsi penuh untuk *smartphone* dan tablet Android dan iOS. App Inventor awalnya dikembangkan oleh Google, dan saat ini dikelola oleh Massachusetts Institute of Technology.

Bahasa Pemrograman (*Programming Language*) , kumpulan perintah, instruksi, dan sintaks lain yang digunakan untuk membuat suatu program.

Coding, kegiatan menulis kode sumber program.

Checksum, metode verifikasi yang digunakan untuk memeriksa apakah data yang dikirim ke penerima telah berubah atau rusak; dihitung dari blok data data yang dikirim; nilai checksum dikirim untuk setiap blok data

Data (*Data*) , fakta yang dikumpulkan dan digunakan untuk referensi atau analisis. Data bisa digital atau nondigital dan bisa dalam berbagai bentuk, termasuk angka, teks, uluran tangan, gambar, suara, atau video.

Deforestasi, peristiwa hilangnya hutan alam beserta dengan atributnya yang diakibatkan oleh penebangan hutan

Diagram Alir (*Flowchart*) , sebuah bagan atau diagram dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail serta hubungan antar proses.

Dokumentasi (*Documentation*) , perangkat lunak adalah teks atau ilustrasi tertulis yang menyertai perangkat lunak komputer atau disematkan dalam kode sumber. Dokumentasi menjelaskan bagaimana perangkat lunak beroperasi atau bagaimana menggunakannya, dan mungkin memiliki arti yang berbeda bagi orang-orang dalam peran yang berbeda.

Graf (*Graph*), suatu struktur dari sekumpulan objek di mana beberapa pasangan objek memiliki hubungan atau keterkaitan tertentu.

Infografis (*Infographics*) , adalah representasi visual (grafis) dari suatu informasi, data, atau pengetahuan untuk menyajikan informasi yang dapat disajikan dengan cepat dan jelas; biasanya menggunakan elemen grafis untuk menyajikan informasi dengan cara yang menarik secara visual.

Informatika (*Informatics*), ilmu yang mempelajari penggunaan komputer untuk mengatur dan menganalisis data yang berukuran besar.

Inklusif (*Inclusive*) , dalam konteks Matematika dan Informatika, inklusif berarti ‘termasuk’. keterangan 1 sampai 100 (inklusif) artinya kalian dapat memilih bilangan 1, 100, dan semua bilangan di antara 1 dan 100.

Input/Masukan, data yang diterima oleh program untuk diproses.

Internet (*Internet*) , jaringan komputer global yang saling berhubungan dengan menggunakan paket protokol internet untuk berkomunikasi dengan jaringan dan perangkat-perangkat yang saling terhubung.

Internet Of Things (IoT) , kemampuan terhubungnya benda dan perangkat (misalnya penyiram tanaman, perangkat sensor, dan peralatan sehari-hari lainnya) dengan jaringan yang memungkinkan pengiriman informasi antar-benda menggunakan internet.

Jaringan Komputer (*Computer Network*) , kumpulan dari dua atau lebih komputer yang dihubungkan bersama-sama untuk tujuan berbagi informasi, dan sumber daya, antara satu sama lain.

Kasus Uji (*Test Case*) , suatu kumpulan nilai dengan kondisi tertentu yang dimasukkan ke dalam program oleh penguji untuk menentukan apakah program yang diuji memenuhi spesifikasi atau berjalan dengan benar.

Kepala Paket (*Packet Header*) , bagian dari paket data yang berisi alamat IP pengirim dan penerima, termasuk nomor paket yang memungkinkan penyusunan kembali paket data

Kode Sumber (*Source Code*) , bentuk program yang diberikan kepada kompilator untuk dikonversi menjadi *object code*.

Kompleksitas (*Complexity*) , Jumlah sumber daya minimum, seperti memori, waktu, atau pesan, yang diperlukan untuk menyelesaikan suatu masalah (*problem*) atau menjalankan suatu algoritma.

Larik (*Array*) , larik adalah suatu tipe data terstruktur yang dapat menyimpan banyak data dengan suatu nama yang sama dan menempati tempat di memori yang berurutan serta bertipe data sama pula dan dapat diakses berdasarkan indeksinya.

Memoization, sebuah teknik atau cara untuk menyimpan hasil perhitungan yang telah diperoleh sebelumnya, agar jika diperlukan lagi, tidak perlu dihitung kembali.

Node, pada jaringan komputer adalah tahapan dalam jaringan yang dapat menerima dan mengirimkan paket data; router adalah *node* di jaringan komputer

Packet Switching, metode transmisi data yang efisien dimana pesan dipecah menjadi unit yang relatif kecil yang disebut paket data, yang dikirimkan secara independen dan kemudian disusun kembali.

Paket Data, pecahan kecil dari pesan/data yang dikirimkan melalui jaringan; setelah transmisi semua paket data dipasang kembali untuk membentuk pesan/data asli.

Pemrogram (*Programmer*) , orang yang melakukan kegiatan pemrograman.

Pemrograman (*Programming*) , aktivitas yang dilakukan untuk menghasilkan suatu program, termasuk analisis kebutuhan, perancangan, implementasi, dan pengujian suatu program.

Pemrograman Dinamis, (*dynamic programming*) sebuah strategi penyelesaian masalah optimasi komputasional yang bersifat rekursif, dimana solusi permasalahan awal didapatkan dengan menggabungkan solusi dari sub-sub soal permasalahan awal tersebut, namun dengan menghindari adanya redundansi/pengulangan perhitungan dengan memanfaatkan teknik memoisasi.

Perangkat Keras (*Hardware*) , komponen fisik yang menyusun sistem komputasi, komputer, atau perangkat komputasi.

Perangkat Lunak (*Software*) , program yang berjalan di atas sistem komputasi, komputer, atau perangkat komputasi lainnya.

rekursif (*recursive*) , memiliki sifat atau mengandung rekursi.

Router, perangkat jaringan yang meneruskan paket data antar jaringan komputer.

String , urutan huruf, angka, dan/atau simbol lainnya. Sebuah string dapat mewakili data seperti nama, alamat, atau judul lagu.

Ujung Paket (Packet Trailer) , bagian dari paket data yang menunjukkan akhir paket data dan cara pemeriksaan kesalahan

Visualisasi , representasi grafis dari data, umumnya dipergunakan sebagai cara efisien untuk mengkomunikasikan data dalam jumlah banyak

LAMPIRAN 4

DAFTAR PUSTAKA

- _____. (n.d.), Digital literacy, diakses dari en.wikipedia.org/wiki/Digital_literacy pada tanggal 10 November 2021
- _____. (n.d.), Critical Thinking, diakses dari westernsydney.edu.au/__data/assets/pdf_file/0006/1082382/Critical_Thinking.pdf pada tanggal 26 November 2021.
- _____. (n.d.). What is digital literacy, diakses dari westernsydney.edu.au/studysmart/home/study_skills_guides/digital_literacy/what_is_digital_literacy pada tanggal 10 November 2021
- _____. (n.d.). Decision-making process, diakses dari umassd.edu/media/umassdartmouth/fycm/decision_making_process.pdf pada tanggal 29 November 2021
- _____. (2020). One Tree Planted. What is Deforestation. Diakses dari youtu.be/vJnnrpSDWPI pada 15 November 2021.
- _____. (2017). National Geographic. Climate 101: Deforestation. Diakses dari youtu.be/Ic-J6hcSKa8. Pada 12 November 2021
- _____. (7 Desember 2020). Angka Deforestasi Netto Indonesia Di Dalam Dan Di Luar Kawasan Hutan Tahun 2013-2019 (Ha/Th). Diakses dari bps.go.id/statistictable/2019/11/25/2081/angka-deforestasinetto-indonesia-di-dalam-dan-di-luar-kawasan-hutan-tahun-2013-2019-ha-th.html pada 12 November 2021.
- _____. (n.d.), Digital literacy, diakses dari en.wikipedia.org/wiki/Digital_literacy pada tanggal 10 November 2021
- _____. (n.d.), Critical Thinking, diakses dari westernsydney.edu.au/__data/assets/pdf_file/0006/1082382/Critical_Thinking.pdf pada tanggal 26 November 2021.
- _____. (n.d.). What is digital literacy, diakses dari westernsydney.edu.au/studysmart/home/study_skills_guides/digital_literacy/what_is_digital_literacy pada tanggal 10 November 2021
- _____. (n.d.). Decision-making process, diakses dari umassd.edu/media/umassdartmouth/fycm/decision_making_process.pdf pada tanggal 29 November 2021.
- _____. (n.d.), Robot Medis, diakses dari id.wikipedia.org/wiki/Robot_medis pada tanggal 5 November 2021
- _____. (n.d.). What is a Transmission Control Protocol TCP/IP Model?. Diakses dari fortinet.com/resources/cyberglossary/tcp-ip pada tanggal 5 November 2021.
- _____. (n.d). App Inventor Tutorial, diakses dari <http://appinventor.mit.edu/explore/sites/all/files/hourofcode/TalkToMe-Part1.pdf>, pada tanggal 10 November 2021
- _____. (n.d). Introduction to Machine Learning: Image Classification, diakses dari <https://appinventor.mit.edu/explore/resources/ai/image-classification-look-extension>, pada tanggal 10 November 2021

- _____. (n.d). Voice Calculator Tutorial, diakses dari <https://appinventor.mit.edu/explore/resources/ai/voice-calculator>, padatanggal 10 November 2021
- _____. (n.d). I have a dream tutorial, diakses dari <http://www.appinventor.org/content/ai2apps/simpleApps/dream>, padatanggal 10 November 2021
- Agustini, P. (2021 Mei 3), Kementerian Komunikasi dan Informatika, diakses melalui aptika.kominfo.go.id/2021/05/kominfo-catat-1-733-hoaks-covid-19-dan-vaksin pada tanggal 10 November 2021.
- Alexander, H.B., (2020), Hebatnya China, Rumah Sakit Corona Dilengkapi Robot Medis, diakses melalui properti. kompas.com/read/2020/02/03/234056221/hebatnya-china-rumah-sakit-corona-dilengkapi-robot-medis pada tanggal 5 November 2021.
- Annur, C.M. (2021 Mei 11), Katadata, diakses melalui katadata.co.id/ariayudhistira/analisisdata/609a43a46aa5e/pencurian-data-pribadi-dalam-pusaran-bisnis-fintech-ilegal pada tanggal 10 November 2021.
- Baker, Dennis, et.al.,(2001), Guidebook to Decision Making Methods, Department of Energy, United States of America, diakses melalui researchgate.net/publication/255621095_Guidebook_to_Decision-Making_Methods pada tanggal 10 November 2021.
- Booth, W. (n.d.) Rainforest Deforestation and Its Effects. Dikases dari youtu.be/Nc7f5563azs pada 15 November 2021.
- Carpenter, M., T. Bauer, B. Erdogan, (n.d.), Management Principles v 1.0. ,diakses dari 2012books.lardbucket.org/books/management-principles-v1.0/s15-decision-making.html pada tanggal 20 November 2021.
- Clarke, John (2019). Critical Dialogues: Thinking Together in Turbulent Times. Bristol: Policy Press. p. 6. ISBN 978-1-4473-5097-2.
- Cholle, F.P. (2011 Agustus). What is Intuition and How Do We Use It. psychologytoday.com/us/blog/the-intuitive-compass/201108/what-is-intuition-and-how-do-we-use-it
- CPPReference.com. Standard library header <cstring>. en.cppreference.com/w/cpp/header/cstring. Diakses pada tanggal 10 Januari 2021.
- CPPReference.com. std::basic_string. en.cppreference.com/w/cpp/header/cstring. Diakses pada tanggal 10 Januari 2021.
- CPPReference.com. std::vector. en.cppreference.com/w/cpp/container/vector. Diakses pada tanggal 10 Januari 2021.
- CPPReference.com. Array. en.cppreference.com/w/c/language/array. Diakses pada tanggal 10 Januari 2021.
- Delima, R., H.B. Santoso, dan J. Purwadi, “Kajian Aplikasi Pertanian yang Dikembangkan di Beberapa Negara Asia dan Afrika”, Prosiding Seminar, Seminar Nasional Aplikasi Teknologi Informasi (SNATi), 2016.
- Deitel, P. & Deitel, H. (2016). C: How to Program Edisi ke-8
- Dhany, F.W.W. (2021 November 24), “Panen Data Pribadi lewat Challenge di Media Sosial”, Harian Kompas 24 November 2021.
- Edward M. Glaser. “Defining Critical Thinking”. The International Center for the Assessment of Higher Order Thinking (ICAT, US)/Critical Thinking Community. Retrieved 22 March 2017.
- Forouzan, B. A. (2013). Data Communication and Networking. 5th Ed. New York: McGraw-Hill. ISBN:0073376221

- Herdiana, dan Y. Hermawan, “Analisis Dampak Perubahan Revolusi Industri Pertanian 4.0 terhadap Sosial Ekonomi Petani di Kecamatan Praya Barat Kabupaten Lombok Tengah – NTB”, Jurnal Media Bina Ilmiah, Vol. 15 no. 4, November 2020, diakses melalui ejurnal.binawakya.or.id/index.php/MBI/article/view/774/pdf
- Humas Litbangkes, (2019) Pemanfaatan Teknologi Informasi dalam Kajian Etik, diakses melalui litbang.kemkes.go.id/pemanfaatan-teknologi-informasi-dalam-kajian-etik/ pada tanggal 5 November 2021
- IMD, IMD World Competitiveness Yearbook 2021, diakses melalui imd.org/centers/world-competitiveness-center/rankings/ pada tanggal 10 November 2021.
- Kurnia, T. (2019 Januari 7), Liputan 6, diakses melalui liputan6.com/bisnis/read/3863361/kisah-kerugian-material-akibat-hoaks-di-berbagai-negara pada tanggal 10 November 2021.
- Kurniawan, C., (2016), Masa Depan Bidang Kesehatan! Inilah 7 Robot Medis Super Canggih, diakses melalui today.line.me/id/v2/article/Masa+Depan+Bidang+Kesehatan+Inilah+7+Robot+Medis+Super+Canggih-abd8a10c84b8a0b7c9dc-4d86c6b2b93e57715ad56c609c73f8c58ef51d084838 pada tanggal 5 November 2021.
- Kusnandar, V.B. (2021 Oktober 14), databoks, diakses melalui databoks.katadata.co.id/datapublish/2021/10/14/pengguna-internet-indonesia-peringkat-ke-3-terbanyak-di-asia pada tanggal 10 November 2021
- Miles, B. & Spies-Butcher, B. (2012). Short exercise practice 1: Critical analysis –reading. Sydney: Department of Sociology, Macquarie University.
- Oktari, R., (2021), Indonesiabaik, diakses melalui indonesiabaik.id/videografis/indonesia-makin-melek-literasi-digital pada tanggal 10 November 2021.
- Paul, R. & L. Elder. (2006). The Miniature Guide to Critical Thinking Concepts and Tools. The Foundation for Critical Thinking.
- Subagio, J., (2019), Terinspirasi Transformer, Ahli AS Bikin Robot yang Bisa Lawan Kanker, diakses melalui sains.kompas.com/read/2019/05/27/094638323/terinspirasi-transformer-ahli-as-bikin-robot-yang-bisa-lawan-kanker pada 5 November 2021
- Sumartiningtyas, H.N.K., (2020), Robot Medis ini Mengambil Darah Pasien, Akankah Gantikan Peran Dokter?, diakses melalui sains.kompas.com/read/2020/02/10/180300223/robot-medis-ini-mengambil-darah-pasien-akankah-gantikan-peran-dokter- pada 5 November 2021
- Tim detikcom (2021 Agustus 7), Detiknews, diakses melalui news.detik.com/berita/d-5673218/terulang-lagi-remaja-tewas-ditabrak-truk-demi-konten-dit-angerang pada tanggal 10 November 2021.
- Todd, C. (2015). Deforestation Effects on Climate. Diakses dari youtu.be/AVh-2DEgppsM pada 12 November 2021

Sumber Gambar

Gambar 1.2 The Great Principles of Computing. Sumber :

<https://www.americanscientist.org/article/the-great-principles-of-computing>

Gambar 5.4. Ilustrasi Proses Klasifikasi Gambar, Gambar kucing diambil dari Sumber: By Kari Shea karishea - <https://unsplash.com/photos/eMzblc6JmXMIImageGallery>, CC0, <https://commons.wikimedia.org/w/index.php?curid=62177124>

Gambar 5.5. Gambar kucing dan anjing dan kelasnya, Sumber: Foto Kucing-1, Oleh Kari Shea karishea - <https://unsplash.com/photos/eMzblc6JmXMIImageGallery>, CC0,

<https://commons.wikimedia.org/w/index.php?curid=62177124>, Foto Kucing 2,
<https://www.wallpaperhi.com/thumbnails/detail/20200701/5efc68309ab42.jpg>, Foto
Kucing-3, Oleh Dustin Warrington- Flickr, CC BY-SA 2.0,
<https://commons.wikimedia.org/w/index.php?curid=2645977>. Foto Anjing-1:
https://upload.wikimedia.org/wikipedia/commons/0/0d/A_type_of_dog.jpg, Foto Anjing-2:
Oleh Lokal_Profil - Ownworkshop, CC BY 2.5,
<https://commons.wikimedia.org/w/index.php?curid=811336>, Foto Anjing-3:
https://upload.wikimedia.org/wikipedia/commons/f/fe/American_Eskimo_Dog_1.jpg

Gambar 5.6. Pengujian dengan gambar baru, Foto Caracal Oleh Derek Keatsdari Johannesburg,
South Africa - Caracal on the road, early morningin Kgalagadi, CC BY 2.0,
<https://commons.wikimedia.org/w/index.php?curid=61699151>

Gambar 5.7. Pengujian dengan gambar yang sangat berbeda dari kelas, Fotokuda oleh Larissa Allen
- Contact us/Photo submission, CC BY-SA
3.0,<https://commons.wikimedia.org/w/index.php?curid=6642407>