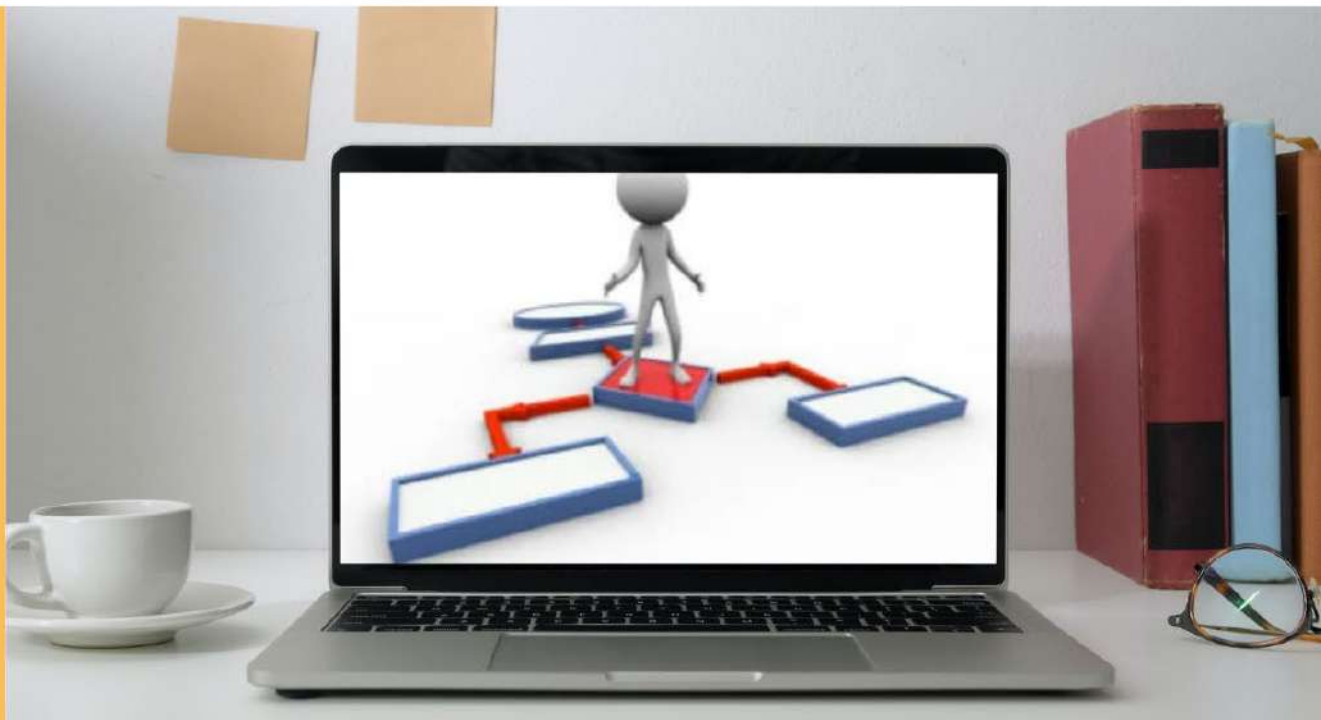


MODUL AJAR INFORMATIKA

ALGORITMA DAN PEMROGRAMAN UNPLUGGED



Bambang Subeno, S.T., M.Kom.

Fase E

**KELAS
X**

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	1 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Problem Based Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ <i>Unplugged</i>
Tujuan Pembelajaran	Siswa mampu membaca dan menulis algoritma dengan benar		
Kata kunci	Algoritma, Diagram Alir		
Deskripsi umum kegiatan	Siswa mempelajari pengertian algoritma dan pemrograman dengan diagram alir, guru akan memberikan contoh penggunaan algoritma dalam diagram alir dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	<p>Materi Ajar: Algoritma, diagram alir</p> <p>Alat: Gunting</p> <p>Bahan: Alat peraga, lembar <i>screenshot</i> / kartu <i>symbol</i> diagram alir.</p>		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-1: Latihan Menelusuri Diagram Alir (1 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Mengenal Algoritma dan Pemrograman	Siswa mampu membaca dan menulis algoritma dengan benar	1 JP	Kuis Cepat Tepat Lembar Kerja

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi perkakas algoritma dan pemrograman sesuai panduan.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik melakukan kolaborasi dan berdiskusi mengembangkan kasus algoritma dan pemrograman	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Apa yang kalian ketahui tentang algoritma dan pemrograman ?
- 2. Apakah pernah mendengar atau mengetahui tentang diagram alir ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru mengenalkan notasi diagram alir.

2. Guru memberikan pertanyaan cepat tepat kepada siswa dengan bentuk alat peraga penamaan notasi diagram alir.
3. Guru memberikan contoh membuat diagram alir dan cara penelusuran diagram alir.
 - Memahami Soal, menentukan *problem* (Permasalahan) dan *solution* (Solusi pemecahan masalah)
 - Membuat diagram alir.
 - Menelusuri diagram alir dengan pendekatan input-proses-output.
4. Guru mengarahkan siswa untuk mengerjakan **aktivitas Latihan Membuat Diagram Alir dan Menelusuri Diagram Alir**. Berikan waktu kepada siswa untuk bekerja mandiri.
5. Guru memberikan beberapa soal tambahan jika diperlukan untuk dapat diberikan sebagai pekerjaan rumah.
6. Guru menginformasikan hasil pengerjaan diagram alir dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara *unplugged* adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang algoritma dan pemrograman?	
Apa yang kalian rasakan ketika mencoba Algoritma dalam bentuk Diagram Alir	
Bagaimana pengalaman kalian mencoba Latihan Diagram Alir?	
Apa yang kalian rasakan ketika mencoba penelusuran diagram alir?	

G. Lembar Kerja Peserta Didik

Aktivitas, Latihan Membuat Diagram Alir dan Menelusuri Diagram Alir.

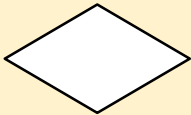




Petunjuk:

Setelah mendengarkan penjelasan dari guru, lakukan sebagai berikut:

- 1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
- 2. Kerjakan soal secara mandiri tidak berkelompok.
- 3. Kerjakan soal berikut ini.

Soal:

- 1. Sebutkan dan jelaskan fungsi simbol Diagram Alir berikut ini ?

	Jawab:
	Jawab:
	Jawab:
	Jawab:
	Jawab:

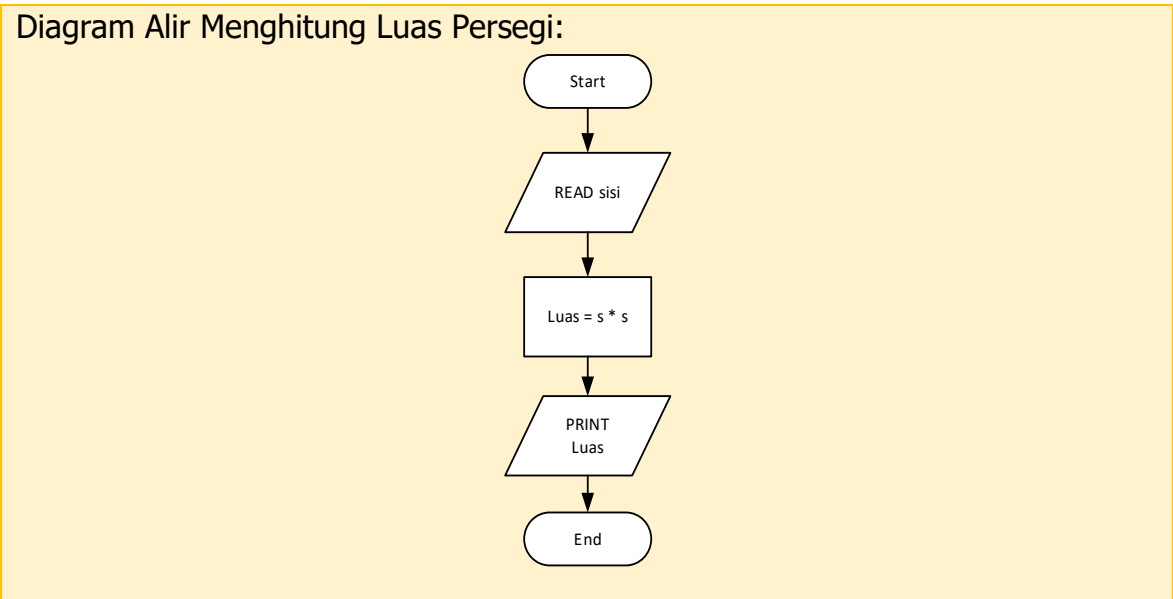
- 2. Buatlah diagram alir untuk menghitung hasil perkalian dua buah bilangan

Jawab:

- 3. Buatlah diagram alir untuk menghitung mundur dari N ke 1

Jawab:

4. Lakukan Penelusuran Diagram alir dari soal berikut:
- a. Jika diketahui Diagram alir Mengitung Luas Persegi, buatlah penelusuran diagram alir dengan scenario pertama nilai Panjang sisi adalah 10 dan skenario kedua Panjang sisi adalah 150



Petunjuk:

- a) Tentukan *problem* dan *solution* dari soal yang ada.

Problem:

Solution:

- b) Buat table Ilustrasi dalam bentuk input-proses-output.
- c) Tuliskan nilai masukan yang menjadi input. Dalam diagram alir dinotasikan dengan READ.
- d) Simpan nilai masukan tersebut kedalam variable yang ditentukan yaitu variable sisi
- e) Lakukan proses menghitung Luas, dengan memasukan nilai variable sisi kedalam rumus $Luas = sisi * sisi$
- f) Tampilkan hasil perhitungan Luas, yang dinotasikan dengan PRINT.

Jawab Skenario Pertama:

Input	Nilai masukan: Nilai Variable:
Proses	
Output	

Jawab Skenario Kedua:

Input	Nilai masukan: Nilai Variable:
Proses	
Output	

b. Siswa melakukan penelusuran dari diagram alir menghitung hasil perkalian dua buah bilangan yang telah dibuat,jika diketahui bilangan A=10, bilangan B=15.

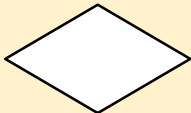
Jawab:





c. Siswa melakukan penelusuran dari diagram alir menghitung mundur dari N ke 1. Jika diketahui nilai N=7.

Jawab:

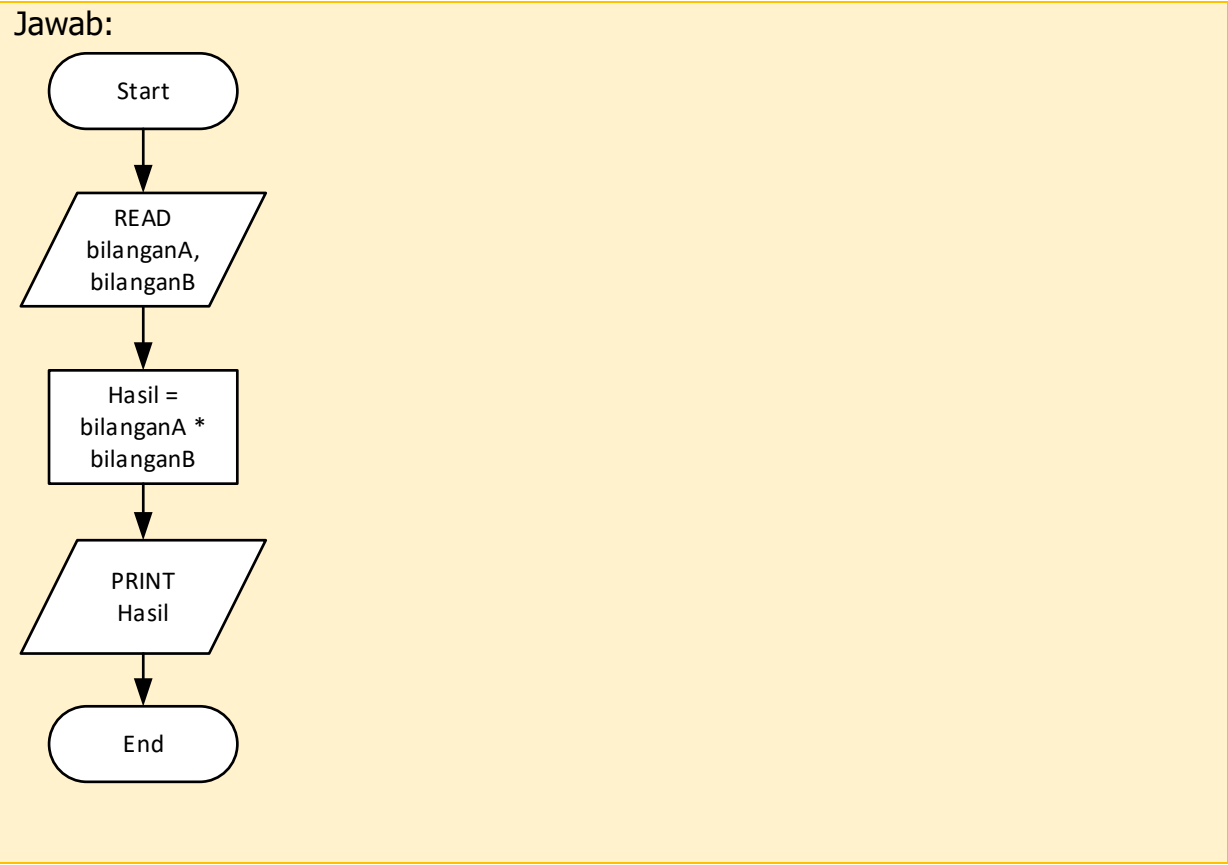
Jawaban:

1. Sebutkan dan jelaskan fungsi simbol Diagram Alir berikut ini ?

	Jawab: <i>Decission</i> / Keputusan, Merupakan titik percabangan untuk menentukan keputusan berdasarkan suatu kondisi
---	--

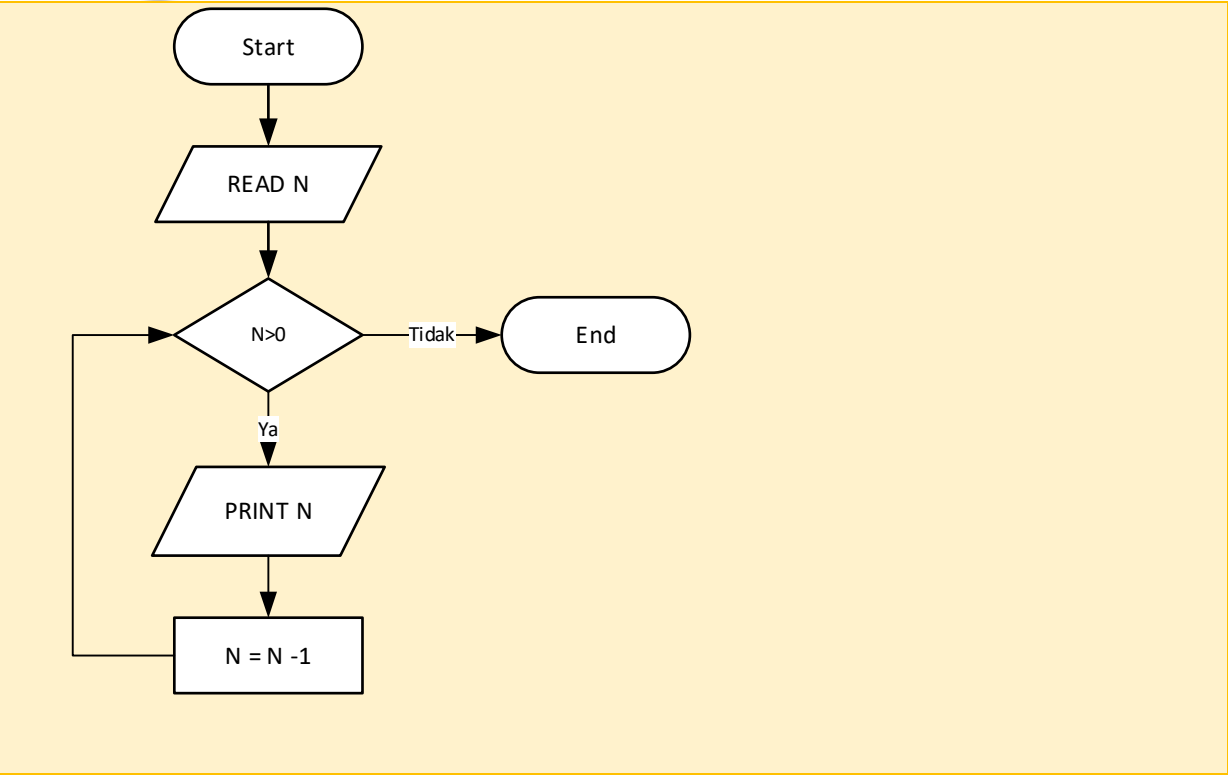
	Jawab: <i>Terminator</i> , yang digunakan untuk memulai (<i>Start</i>) atau mengakhiri (<i>End</i>) program
	Jawab: <i>Flowline</i> / Garis Alir, Digunakan untuk menunjukkan arah aliran program dari awal sampai akhir
	Jawab: <i>Input / Output</i> Data, Digunakan untuk proses <i>input</i> / <i>output</i> Data
	Jawab: <i>On Page Reference</i> , Digunakan untuk menghubungkan titik diagram alir pada halaman yang sama

2. Buatlah diagram alir untuk menghitung hasil perkalian dua buah bilangan



3. Buatlah diagram alir untuk menghitung mundur dari N ke 1

Jawab:



4. Penelusuran Diagram alir.

a. Jawaban:

Problem: Bagaimana Mengitung Luas Persegi

Solution: menggunakan Luas persegi yaitu $\text{Luas} = \text{sisi} * \text{sisi}$

Jawab Skenario Pertama:

Input	Nilai masukan: 10 Nilai Variable: sisi = 10
Proses	$\text{Luas} = \text{sisi} * \text{sisi}$ $\text{Luas} = 10 * 10$ $\text{Luas} = 100$
Output	100

Jawab Skenario Kedua:

Input	Nilai masukan: 150 Nilai Variable: sisi = 150
Proses	$\text{Luas} = \text{sisi} * \text{sisi}$ $\text{Luas} = 150 * 150$
Output	22500

b. Jawaban:

Input	Nilai masukan: bilangan A = 10 dan bilangan B = 15 Nilai Variable: bilanganA=10, bilanganB=15
Proses	Hasil = bilanganA * bilanganB Hasil = 10 * 15 Hasil = 150
Output	150

c. Jawaban:

Input	Nilai masukan: nilai N =3 Nilai Variable: N=3
Proses	Apakah nilai 3 > 0 ? Ya Tampilkan 3 N=N-1=3-1=2 Apakah nilai 2 > 0 ? Ya Tampilkan 2 N=N-1=2-1=1 Apakah nilai 1 > 0 ? Ya Tampilkan 1 N=N-1=1-1=0 Apakah nilai 0 > 0 ? Tidak Proses berhenti
Output	3 2 1

Pengertian Algoritma




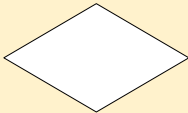

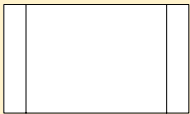
Algoritma adalah suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan dalam bentuk program komputer untuk menyelesaikan suatu permasalahan komputasi tertentu. Algoritma merupakan bentuk dari suatu strategi Langkah-langkah atau ‘resep’ yang kalian gunakan untuk menyelesaikan suatu masalah. Algoritma lahir dari suatu proses berpikir komputasional oleh seseorang untuk menemukan solusi dari suatu permasalahan yang diberikan. Dengan demikian, berpikir komputasional merupakan keahlian yang kalian perlukan untuk dapat membuat algoritma, program, atau suatu karya informatika yang dapat digunakan dengan efektif dan efisien



Cara untuk merepresentasikan algoritma yaitu dengan diagram alir, pseudocode. Untuk itu, siswa harus memahami teknik bagaimana membaca algoritma yang sering disebut dengan tracing atau penelusuran program dan teknik bagaimana menuliskan algoritma baik dalam bentuk diagram alir, pseudocode.

1. Diagram Alir

Diagram alir adalah diagram yang memperlihatkan langkah-langkah dalam suatu proses, yang mudah untuk dibuat dan mudah untuk dimengerti. Simbol Notasi Diagram Alir yang sering digunakan dapat dilihat pada Table 1.1.

Tabel 1.1 Simbol Notasi Diagram Alir

Simbol	Nama	Keterangan
▼	Flowline / Garis Alir	Digunakan untuk menunjukkan arah aliran program dari awal sampai akhir
	Terminator	Simbol notasi terminator ini digunakan untuk memulai (Start) atau mengakhiri (End) program.
	Manual Input	Digunakan untuk proses input secara manual.
	Input / Output Data	Digunakan untuk proses input / output Data.
	Decission / Keputusan	Merupakan titik percabangan untuk menentukan keputusan berdasarkan suatu kondisi
	Process	Proses yang dilakukan oleh program. Misalnya operasi artimatika
	Sub Process	Digunakan untuk proses lain yang telah didefinisikan sebelumnya.

	On Page Reference	Digunakan untuk menghubungkan titik diagram alir pada halaman yang sama
	Off Page Reference	Digunakan untuk menghubungkan titik diagram alir pada halaman yang berbeda

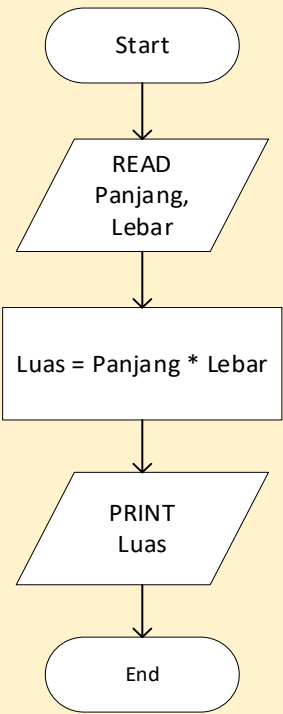
2. Menelusuri Diagram Alir

a. Contoh Menelusuri Diagram Alir: **Menghitung Luas Persegi Panjang**

Diagram alir dimulai dari simbol Start, kemudian mengikuti arah panah sampai dengan selesai yang dismbolkan dengan End. Sebagai contoh kita akan menghitung Luas persegi Panjang.

Problem: Mencari Luas Persegi Panjang
Solution: Rumus: Luas = Panjang x Lebar

Yang dibutuhkan untuk menghitung Luas adalah kita harus mengetahui Panjang dan Lebar - nya, setelah mengetahui Panjang dan Lebar kemudian melakukan proses penghitungan Luas dengan Rumus aritmatika. Untuk lebih jelasnya perhatikan diagram alir sebagai berikut:

Diagram Alir	Keterangan
	<p>Soal: Mencari Luas Persegi Panjang</p> <p>Start: Mulai Program</p> <p>Input: dengan menggunakan notasi READ, Panjang dan Lebar merupakan variable untuk menampung nilai.</p> <p>Proses: Masukan Rumus Luas Persegi Panjang = Panjang * Lebar</p> <p>Output: menampilkan hasil perhitungan Luas</p> <p>End: Mengakhiri program</p>

Untuk dapat melakukan penelusuran (*tracing*) secara terstruktur, kita dapat membuat ilustrasi dalam bentuk tabel sederhana yang terbagi tiga bagian, yaitu masukan, nilai variabel, Proses dan keluaran (*Output*). Bagian **Input** terdiri dari masukan dan nilai variable, masukan akan diisi dengan data yang akan diproses, bagian nilai variabel menjadi tempat mencatat nilai yang disimpan di dalam variable yang ditentukan, bagian **proses** dan bagian **keluaran(Output)** akan diisi dengan data yang dihasilkan oleh diagram alir. Penelusuran ini menjadi penting saat kalian ingin memahami perilaku dari

suatu algoritma, atau saat kalian ingin mengecek ketepatan keluaran dari suatu algoritma.

Contoh dari diagram alir yang telah dibuat, kita akan melakukan penelusuran diagram alir menghitung Luas persegi Panjang, dengan sekenario pertama diketahui Panjang 10 dan lebar 5. Dan sekenario kedua pajang 5 dan lebar 3.

Penjelasan

- g) Buat table Ilustrasi dalam bentuk *input-proses-output*.
- h) Tuliskan nilai masukan yang menjadi input. Dalam diagram alir dinotasikan dengan READ.
- i) Simpan nilai masukan tersebut kedalam variable yang ditentukan yaitu variable Panjang dan variable Lebar
- j) Lakukan proses menghitung Luas, dengan memasukan nilai variable Panjang dan Variable Lebar kedalam rumus Luas = Panjang * Lebar
- k) Tampilkan hasil perhitungan Luas, yang dinotasikan dengan PRINT.

Jawab Skenario Pertama:

Input	Nilai masukan: Panjang = 10, Lebar = 5 Nilai Variable: Panjang = 10, Lebar = 5
Proses	Luas = Panjang * Lebar Luas = 10 * 5 Luas = 50
Output	50

Jawab Skenario Kedua:

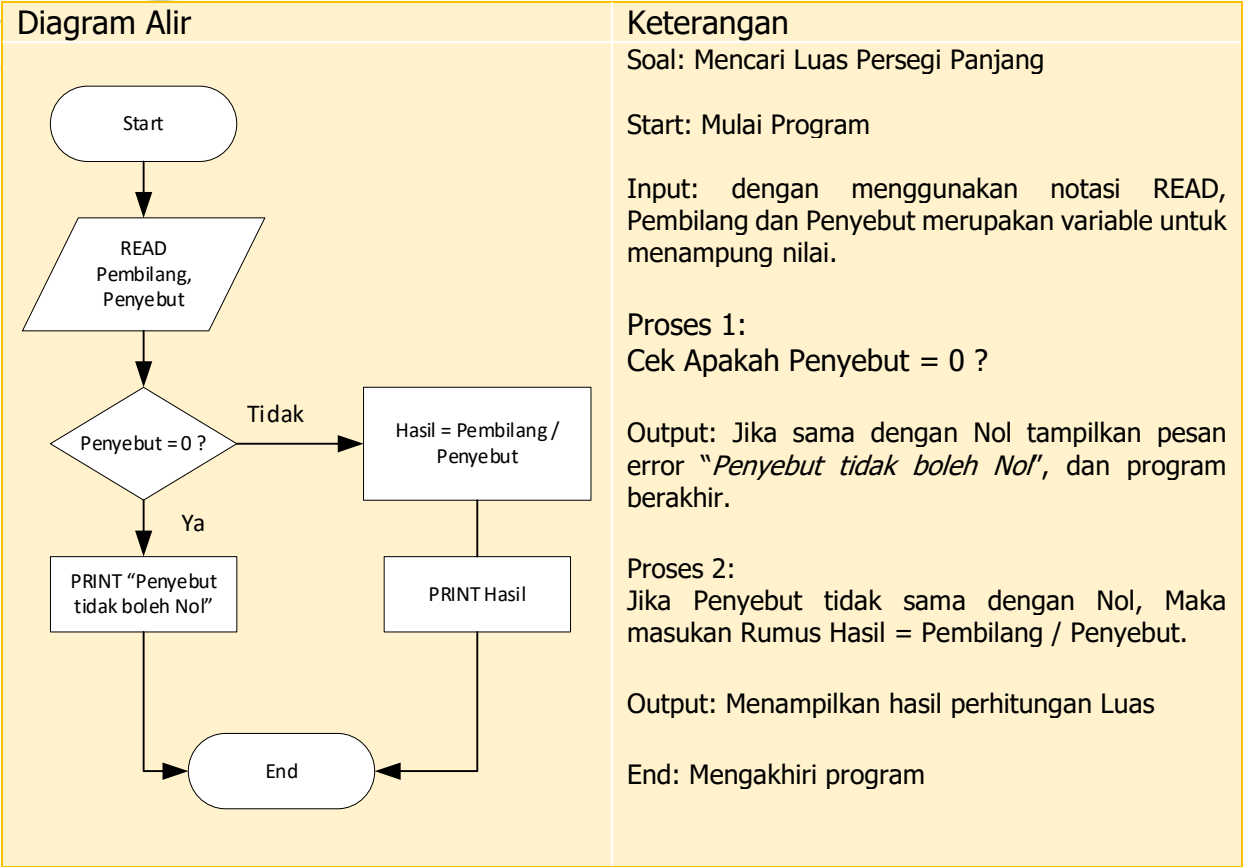
Input	Nilai masukan: Panjang = 5, Lebar = 3 Nilai Variable: Panjang = 5, Lebar = 3
Proses	Luas = Panjang * Lebar Luas = 5 * 3 Luas = 15
Output	15

b. Contoh Menelusuri Diagram Alir: **Membagi Bilangan**

Diagram alir dimulai dari simbol Start, kemudian mengikuti arah panah sampai dengan selesai yang dismbolkan dengan End. Sebagai contoh kita akan menghitung Hasil bagi.

Problem: Mencari Hasil bagi dari bilangan pembilang dan penyebut
Solution: Rumus: Hasil = Pembilang / Penyebut

Yang dibutuhkan untuk menghitung Hasil Bagi adalah kita harus mengetahui Pembilang dan Penyebut-nya, setelah mengetahui Pembilang dan Penyebut kemudian melakukan proses penghitungan hasil bagi dengan Rumus aritmatika. Untuk lebih jelasnya perhatikan diagram alir sebagai berikut:



Lakukan penelusuran diagram alir Membagi dengan skenario pertama diketahui Pembilang 35 penyebut 5 dan skenario kedua pembilang 35 penyebut 0.

Jawab Skenario Pertama:

Input	Nilai masukan: Pembilang = 35, Penyebut = 5 Nilai Variable: Pembilang = 35, Penyebut = 5
Proses	Apakah Penyebut sama dengan 0 ? tidak, maka lakukan proses perhitungan dengan rumus Hasil = Pembilang / Penyebut Hasil = 35 / 5 Hasil = 7
Output	7

Jawab Skenario Kedua:

Input	Nilai masukan: Pembilang = 35, Penyebut = 0 Nilai Variable: Pembilang = 35, Penyebut = 0
Proses	Apakah Penyebut sama dengan 0 ? Ya, maka lakukan proses tidak dilakukan perhitungan Hasil bagi. Tapi langsung ke proses output.
Output	"Penyebut tidak boleh nol"

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	2 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ <i>Unplugged</i>
Tujuan Pembelajaran	Siswa mampu membaca dan menulis algoritma dengan benar		
Kata kunci	Algoritma, <i>Diagram Alir</i> , Pemrograman		
Deskripsi umum kegiatan	Siswa mempelajari pengertian algoritma dan pemrograman dengan diagram alir, guru akan memberikan contoh penggunaan algoritma dalam diagram alir, menuliskan deskripsi algoritma dan <i>pseudocode</i> dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	<p>Materi Ajar: Membaca dan menulis algoritma, diagram alir</p> <p>Alat: Gunting</p> <p>Bahan: Papan tulis, alat peraga, lembar <i>screenshot</i> / kartu symbol diagram alir, lem.</p>		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-2: Latihan Menulis Algoritma (2 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Mengenal Algoritma dan Pemrograman	Siswa mampu membaca dan menulis algoritma dengan benar	2 JP	Lembar Kerja

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi perkakas algoritma dan pemrograman sesuai panduan.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik melakukan kolaborasi dan berdiskusi mengembangkan kasus algoritma dan pemrograman	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat algoritma secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Deskripsi urutan proses aktivitas sehari-hari dari rumah sampai dengan sekolah

E. Kegiatan Pembelajaran

Pendahuluan

- 1) Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2) Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3) Guru menjelaskan teknik asesmen/ penilaian;
- 4) Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5) Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

1. Guru menjelaskan deskripsi algoritma dan *pseudocode*
2. Guru memberikan contoh algoritma dalam bentuk diagram alir, deskripsi algoritma, dan *pseudocode* dari sebuah operasi matematika yang telah dikenal oleh siswa (misal: menghitung luas persegi Panjang atau sejenisnya)
3. Guru mengarahkan siswa untuk mengerjakan aktivitas Latihan Menulis Algoritma. Berikan waktu kepada siswa untuk bekerja mandiri (atau berkelompok) untuk menulis algoritma, baik dalam bentuk diagram alir atau *pseudocode*.
4. Guru meminta siswa yang telah selesai mengerjakan Latihan menulis program untuk menuliskan hasilnya dipapan tulis. Pancing siswa untuk memberikan umpan balik konstruktif satu sama lain.
5. Guru menginformasikan hasil pengerjaan latihan menulis algoritma dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara *unplugged* adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang membuat algoritma?	
Apa yang kalian rasakan ketika mencoba Algoritma dalam bentuk Diagram Alir/Deskripsi Algoritma/Pseudocode?	
Setelah menyelesaikan LKPD, bagaimana perasaan kalian membuat algoritma ?	

G. Lembar Kerja Peserta Didik

Aktivitas, Latihan Menulis Algoritma

Petunjuk:

Setelah mendengarkan penjelasan dari guru, lakukan sebagai berikut:

- 1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
- 2. Kerjakan soal secara berkelompok yang terdiri dari 2 orang.

Soal:

- 1. Buatlah deskripsi urutan proses aktivitas sehari-hari dari rumah sampai dengan sekolah.

Jawab:

- 2. Tuliskan program dalam bentuk deskripsi algoritma dan dan *pseudocode*.
 - a. Menghitung Luas segitiga

Jawab:

Deskripsi Algoritma	<i>Pseudocode</i>

- b. Menghitung Mundur dari N hingga 1

Jawab:

Deskripsi Algoritma	<i>Pseudocode</i>

- c. Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan

Jawab:

Deskripsi Algoritma	<i>Pseudocode</i>

3. Kelompok yang ditunjuk, menuliskan jawabanya dipapan tulis.

Jawaban:

1. Jawabannya akan beragam sesuai dengan deskripsi masing-masing siswa, sebagai contoh:
Deskripsi:
a. Mandi
b. Memakai seragam
c. Sarapan Pagi
d. naik sepeda
e. Lewat jalan A
f. Sampai sekolah
2. Jawab:
a. Menghitung Luas segitiga

Deskripsi Algoritma	Pseudocode
1. Mulai program 2. Baca Nilai Alas dan Tinggi 3. Hitung Luas segitiga 4. Cetak Hasil Luas	Algoritma LuasSegitiga Read Alas, Tinggi Luas = (Alas * Tinggi)/2 Print Luas

b. Menghitung Mundur dari N hingga 1

Deskripsi Algoritma	Pseudocode
1. Mulai program 2. Baca Nilai N 3. Cek kondisi apakah N > 0, jika ya maka, prosesnya: a. Cetak N b. Jalankan rumus N=N-1 c. Ulangi proses sampai kondisi terpenuhi yaitu nilai N lebih kecil sama dengan 0 4. Jika nilai N lebih kecil sama dengan 0 maka, prosesnya a. Berhenti.	Algoritma Mundur Read N While (N > 0) do Print N N=N-1 End While

c. Mencari Bilangan Terbesar dari Suatu Himpunan Bilangan

Deskripsi Algoritma	Pseudocode
<ol style="list-style-type: none">1. Mulai program2. Baca Nilai i, Terbesar3. Cek kondisi apakah $i \leq 10$, jika ya maka, prosesnya:<ol style="list-style-type: none">d. Cek apakah nilai $i > \text{Terbesar}$, jika yae. Update nilai $\text{Terbesar} = i$f. Jalankan $i=i+1$g. Ulangi proses sampai kondisi memenuhi nilai $i \leq 10$4. Jika nilai i terpenuhi $i \leq 10$ maka5. Cetak nilai Terbesar6. Program berakhir	<p>Algoritma BilanganTerbesar</p> <p>Read i, Terbesar</p> <p>While ($i \leq 10$) do if ($i > \text{Terbesar}$) then Terbesar=i end if i=i+1 End While Print Terbesar</p>

H. Bahan Bacaan

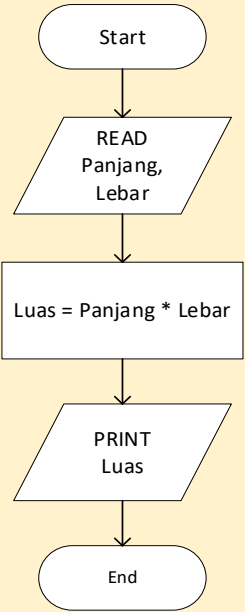
3. Pseudocode

Pseudocode (kode semu atau kode pseudo) adalah suatu bahasa buatan manusia yang sifatnya informal untuk merepresentasikan algoritma. Pseudocode dibuat untuk menutupi kekurangan diagram alir dalam merepresentasikan konsepkonsep pemrograman terstruktur. Pseudocode memungkinkan representasi langkah-langkah yang lebih detail dan dekat dengan bahasa pemrograman. Karena sifatnya yang informal, tidak ada aturan khusus dalam standar notasi yang dapat digunakan. Akan tetapi, ada beberapa prinsip dasar yang perlu diperhatikan, yaitu satu baris untuk satu pernyataan (*statement*) dan pentingnya menggunakan indentitas dalam menuliskan pernyataan. Misalnya, untuk menunjukkan bahwa suatu pernyataan merupakan bagian dari sebuah struktur kontrol keputusan atau struktur kontrol perulangan (lihat konsep blok pada pemrograman visual yang telah kalian pelajari di tingkat SMP)

4. Deskripsi Algoritma

Deskripsi algoritma adalah penjelasan urutan Langkah dari tahap awal sampai akhir, yang dijelaskan dalam bentuk deskripsi. Tidak ada aturan baku yang digunakan, akan tetapi urutan Langkah digambarkan dengan penomoran untuk menunjukan Langkah pertama, kedua, dan seterusnya.

Contoh Algoritma; Dalam bentuk diagram alir, deskripsi algoritma dan pseudocode sebagai berikut:

Diagram Alir	Deskripsi Algoritma	Pseudocode
	<div>5. Mulai program</div> <div>6. Baca Nilai Panjang dan Lebar</div> <div>7. Hitung Luas Persegi Panjang</div> <div>8. Cetak Hasil Luas</div>	<div>Algoritma LuasPersergiPanjang</div> <div>Read Panjang, Lebar</div> <div>Luas = Panjang * Lebar</div> <div>Print Luas</div>

AP-03

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	1 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ <i>Unplugged</i>
Tujuan Pembelajaran	Siswa memahami cara menginstal <i>tools</i> dan lingkungan pemrograman (IDE) yang dipakai		
Kata kunci	<i>IDE</i> , Pemrograman, Bahasa C		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana melakukan simulasi instalasi <i>tools</i> IDE Bahasa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	Materi Ajar: Cara menginstal <i>tools</i> lingkungan pemrograman (IDE) Bahasa C Alat: Gunting Bahan: Alat peraga, lembar screenshot instalasi IDE Bahasa C, karton, lem.		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-3: Simulasi Instalasi IDE Bahasa C (1 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Mengenal Algoritma dan Pemrograman	Siswa memahami cara menginstal tools dan lingkungan pemrograman (IDE) yang dipakai	1 JP	Kuis Cepat Tepat Keaktifan Lembar Kerja

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi perkakas instalasi IDE Bahasa C sesuai panduan.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik melakukan kolaborasi dan berdiskusi melakukan simulasi instalasi IDE Bahasa C	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Apa yang dimaksud dengan IDE (*Integrated Development Environment*) ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru mengenalkan IDE Bahasa C.
 - Guru menyiapkan alat peraga berupa screenshot/kartu Logo IDE

- Menjelaskan IDE Bahasa C menggunakan alat peraga.
- 2. Guru membentuk kelompok dan memberikan pertanyaan cepat tepat, terkait tentang IDE Bahasa C. Dengan contoh soal berikut (guru dapat mengembangkan pertanyaan sejenis).
 - a. Apa yang dimaksud dengan IDE
 - b. Dengan menggunakan alat peraga jenis-jenis IDE Bahasa C, Guru menunjukan lembar *screenshot*/kartu Logo IDE Bahasa C kepada siswa. Jenis IDE apakah ini ?
 - c. Gunakan juga soal simbol notasi diagram alir untuk mengingat Kembali pelajaran sebelumnya.
- 3. Guru menjelaskan prosedur cara instalasi IDE Bahasa C.
 - Guru menyiapkan lembar *Screenshot*/Kartu petunjuk instalasi IDE Bahasa C.
 - Guru melakukan simulasi dan menjelaskan Langkah pertama dengan menempelkan Lembar *Screenshot* instalasi IDE Bahasa C di papan tulis sampai membentuk rangkaian prosedur instalasi IDE bahasa C yang lengkap.
 - Guru menjelaskan Langkah untuk melakukan *setting compiler*, gunakan alat peraga berupa lembar *screenshot*/kartu prosedur setting compiler.
- 4. Guru meminta siswa menjelaskan ulang prosedur instalasi dengan dengan menggunakan lembar *screenshot*/kartu instalasi yang disediakan. Pilih Siswa secara acak.
 - Guru Menyusun Lembar *screenshot*/kartu instalasi secara acak.
 - Siswa yang terpilih maju kedepan diminta untuk menyusun dengan cara menempelkan urutan instalasi di papan tulis.
 - Siswa yang lain bersama-sama melakukan koreksi
- 5. Lakukan hal yang sama untuk proses *setting compiler*
- 6. Guru mengarahkan siswa untuk mengerjakan aktivitas Simulasi Instalasi IDE Bahasa C
- 7. Guru menginformasikan hasil pengerjaan aktivitas Simulasi instalasi IDE Bahasa C dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang IDE Bahasa C?	
Apa yang kalian rasakan ketika mencoba simulasi instalasi IDE Bahasa C secara <i>unplugged</i> ?	
Apa yang anda rasakan Ketika mempelajari IDE Bahasa C secara unplugged pertama kali?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian simulasi secara unplugged?	

G. Lembar Kerja Peserta Didik

Aktivitas, Simulasi Instalasi IDE Bahasa C

Petunjuk:

Setelah mendengarkan penjelasan dari guru, lakukan sebagai berikut:

- 1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
- 2. Siswa menyiapkan karton untuk yang dijadikan media untuk menempelkan lembar Langkah instalasi IDE Bahasa C
- 3. Siswa menyiapkan gunting, lem.
- 4. Kerjakan soal secara berkelompok yang terdiri dari 2-5 orang.

Soal:

- 1. Jelaskan apa yang dimaksud dengan IDE (*Integrated Development Environment*).

Jawab:

- 2. Apa yang dimaksud dengan

a. *Compiler*

Jawab:

b. *Executable*

Jawab:

c. *Linker*

Jawab:

3. Buat susunan Langkah-langkah instalasi IDE Bahasa C dengan menggunakan code::Blocks pada media karton, gunakan lembar *screenshot*/kartu yang telah disediakan.
4. Buat penjelasan disetiap Langkah instalasi IDE Bahasa C.

Jawaban:

1. IDE (*Integrated Development Environment*) adalah aplikasi yang menggabungkan *teks editor*, *compiler* dan *linker* dalam 1 aplikasi.
2. Yang dimaksud dengan:
 - a. Compiler yang akan membaca kode bahasa C yang telah ditulis dan mengubahnya menjadi bahasa mesin, atau bahasa assembly
 - b. *Executable*, berkas yang dapat dieksekusi.
 - c. *Linker* yang akan menyatukan beberapa berkas yang dihasilkan dalam proses-proses sebelumnya
3. Jawaban akan berupa Susunan Langkah Instalasi IDE Bahasa C yang di tempel di media karton
4. Jawaban akan tentative bagaimana menyimpulkan proses instalasi dari setiap kelompok

H. Bahan Bacaan

Mengenalkan IDE Bahasa C

Bahasa Pemrograman C, yang selanjutnya dapat disebut dengan bahasa C saja dikembangkan oleh Dennis M. Ritchie dan Brian W. Kernighan pada awal tahun 1970. Perkembangan bahasa C tidak bisa dipisahkan dari perkembangan sistem operasi UNIX, yang 90% lebih di antaranya ditulis dalam bahasa C. Karena sejarah yang panjang tersebut, kemudian muncul beberapa standar bahasa C yang merupakan spesifikasi dari bahasa C. Spesifikasi ini akan berpengaruh pada perilaku program yang dibuat dengan menggunakan bahasa tersebut. Beberapa standar yang ada meliputi: definisi Kernighan & Ritchie (K&R), ANSI-C (X-3.159 -1989-), Definisi AT&T (untuk superset C, C++), dan GNU Coding Standards. Versi pada PC misalnya: Lattice C, Microsoft C/Microsoft QuickC, dan Turbo C/Borland C++.

Bahasa C banyak dipakai untuk membuat sistem operasi dan program-program sistem, pemrograman tingkat rendah, atau yang "dekat" ke perangkat keras (misalnya untuk kontrol peralatan), membuat toolkit pemrograman, dan menulis aplikasi. Kelebihan bahasa C sehingga banyak digunakan ada pada kemampuannya untuk menghasilkan kode yang singkat, efisien, tetapi tetap mudah dibaca. Berbeda halnya dengan bahasa mesin yang efisien, tetapi membutuhkan latihan khusus untuk membacanya, atau bahasa tingkat tinggi lain yang enak dibaca, tetapi tidak efisien. Walaupun demikian, perlu diakui bahwa kesulitan untuk membaca program bahasa C lebih tinggi daripada bahasa tingkat tinggi lain.

Untuk dapat membuat program dalam bahasa C, diperlukan beberapa perangkat lunak. Perangkat lunak yang pertama ialah sebuah editor teks tempat kita mengetikkan kode program. Selanjutnya, ada sebuah kompilator (*compiler*) yang akan membaca kode bahasa C yang telah ditulis dan mengubahnya menjadi bahasa mesin, atau bahasa assembly. Setelah itu, terdapat sebuah assembler yang akan mengubah bahasa mesin tersebut ke dalam kode biner yang dapat dipahami dan dieksekusi oleh komputer. Terakhir, terdapat sebuah penghubung (*linker*) yang akan menyatukan beberapa berkas yang dihasilkan dalam proses-proses sebelumnya ke dalam sebuah bentuk berkas yang dapat dieksekusi (*executable*)

Pada awalnya, perangkat lunak tersebut terpisah, tetapi untuk memudahkan, akhirnya, dibuatlah sebuah perangkat lunak terintegrasi yang mencakup semua perangkat lunak di atas. Perangkat lunak tersebut disebut lingkungan pengembangan terpadu (*integrated development environment*). IDE (*Integrated Development Environment*) adalah aplikasi yang menggabungkan *teks editor*, *compiler* dan *linker* dalam 1 aplikasi.

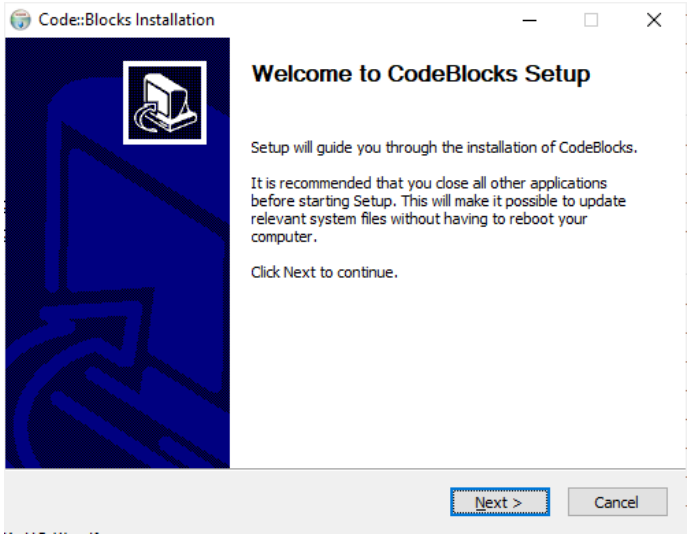
1. Jenis IDE Bahasa C:

Lattice C	
Turbo C / Borland C ++	
Dev C++	
Microsoft Visual C++	

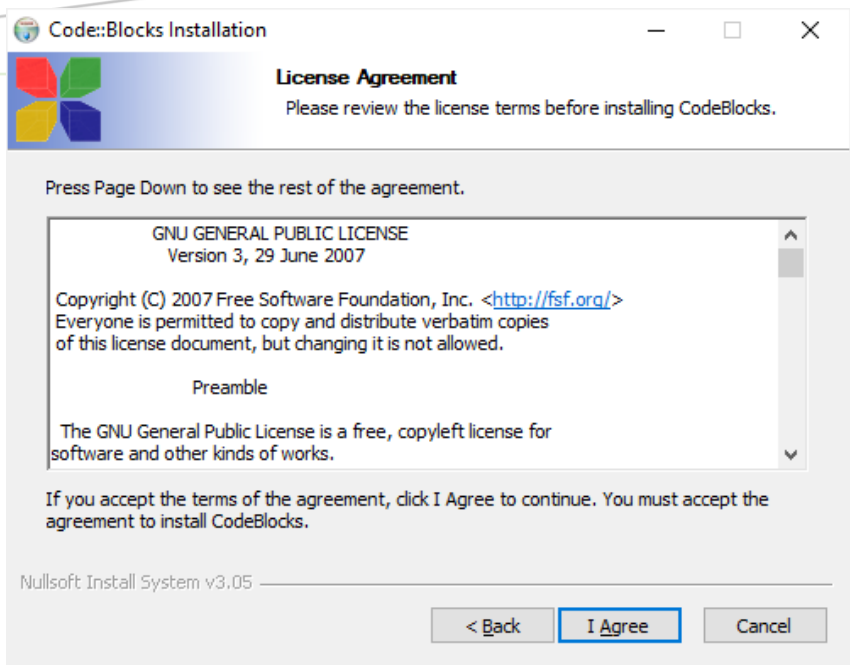


2. Prosedur Instalasi IDE Code::Blocks

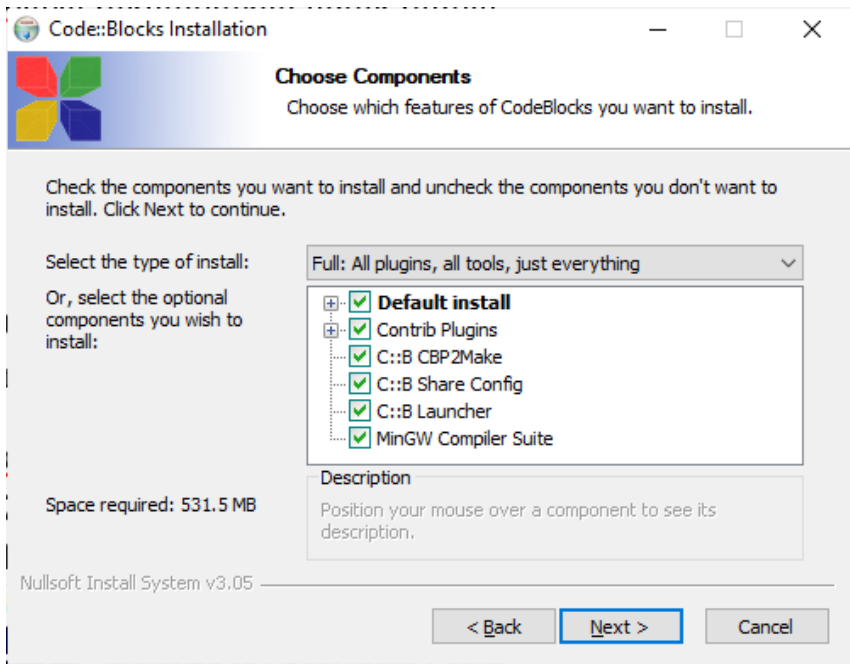
- 1) File codeblock harus ada terlebih dahulu, misalkan nama file installer codeblock adalah **codeblocks-20.03mingw-setup.exe**
- 2) **Double klik file codeblocks-20.03mingw-setup.exe, akan muncul dialog setup.**



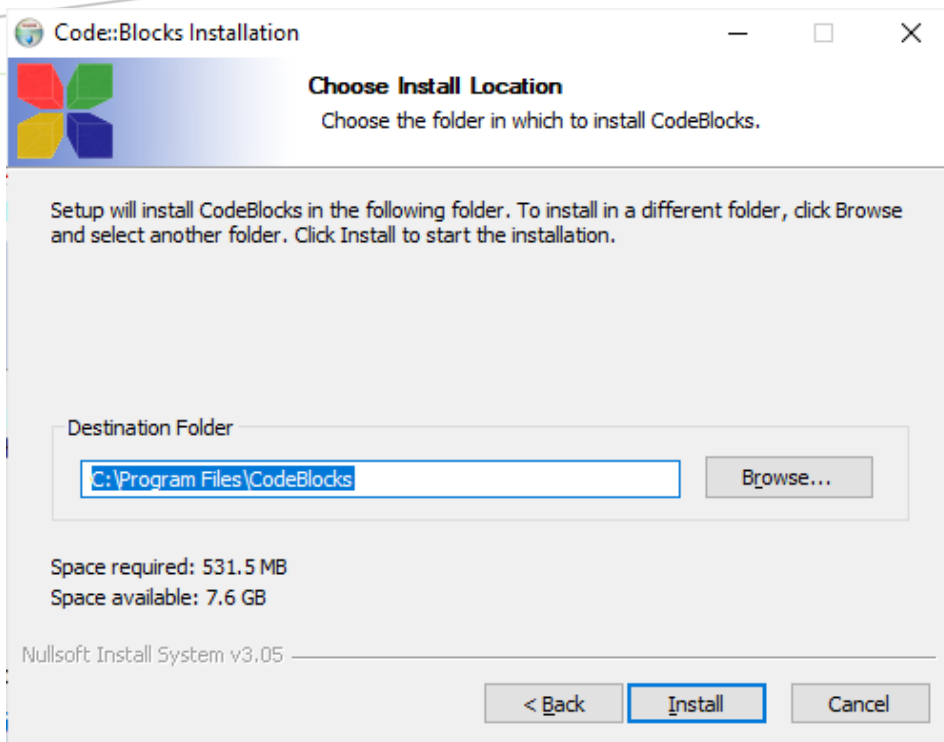
- 3) Klik tombol **“Next”**, akan muncul dialog Licence Agreement.



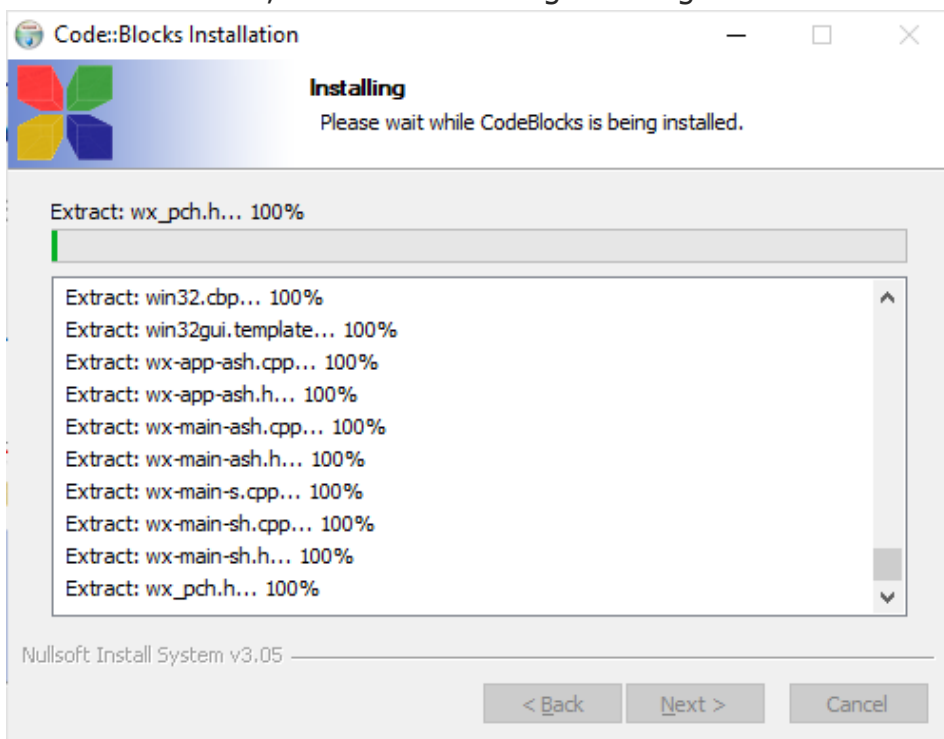
4) Klik tombol “ **I Agree** ”, akan muncul dialog choose components.



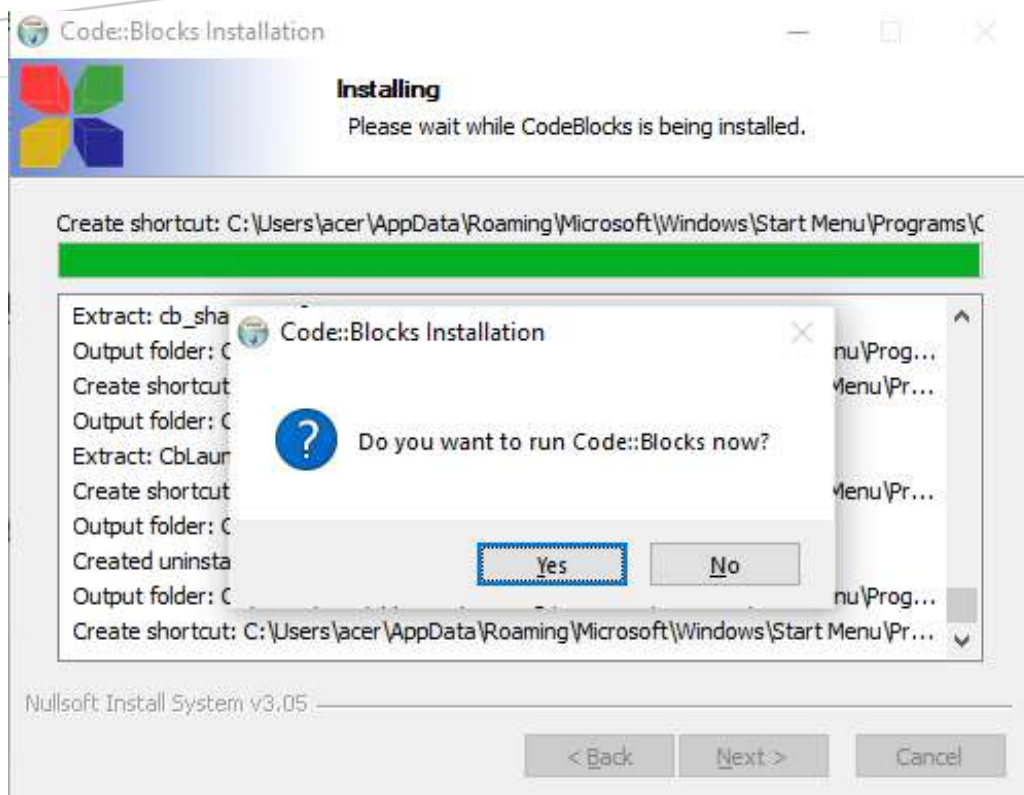
5) Klik Tombol Next, akan muncul dialog Choose Install Location.



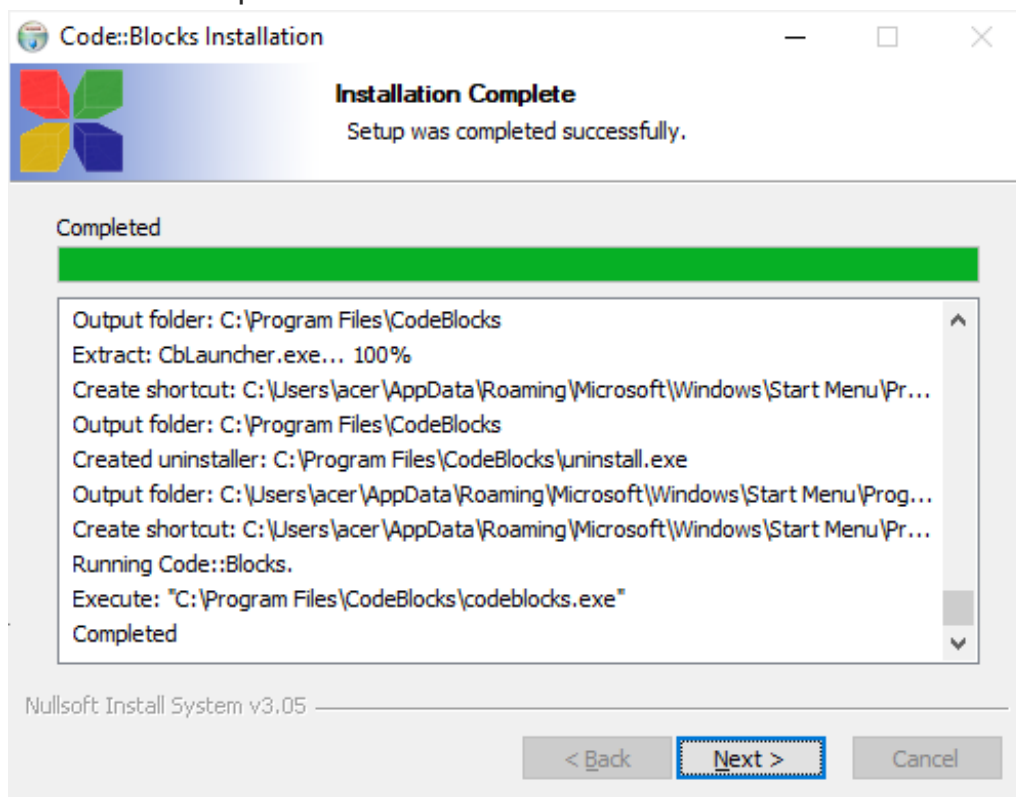
6) Klik Tombol Install, akan muncul dialog installing.



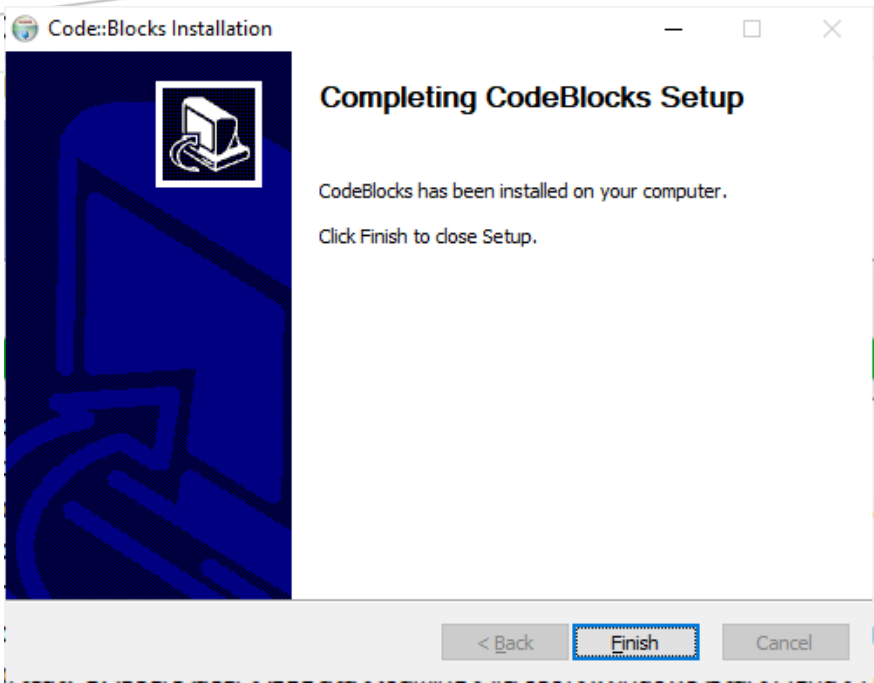
7) Tunggu Sampai proses selesai, setelah selesai akan muncul dialog pilihan run code::blocks atau tidak. Klik **Yes** untuk menjalankan IDE Code::Blocks.



8) Installation Complete



9) Klik Tombol Next, akan muncul dialog completing setup.

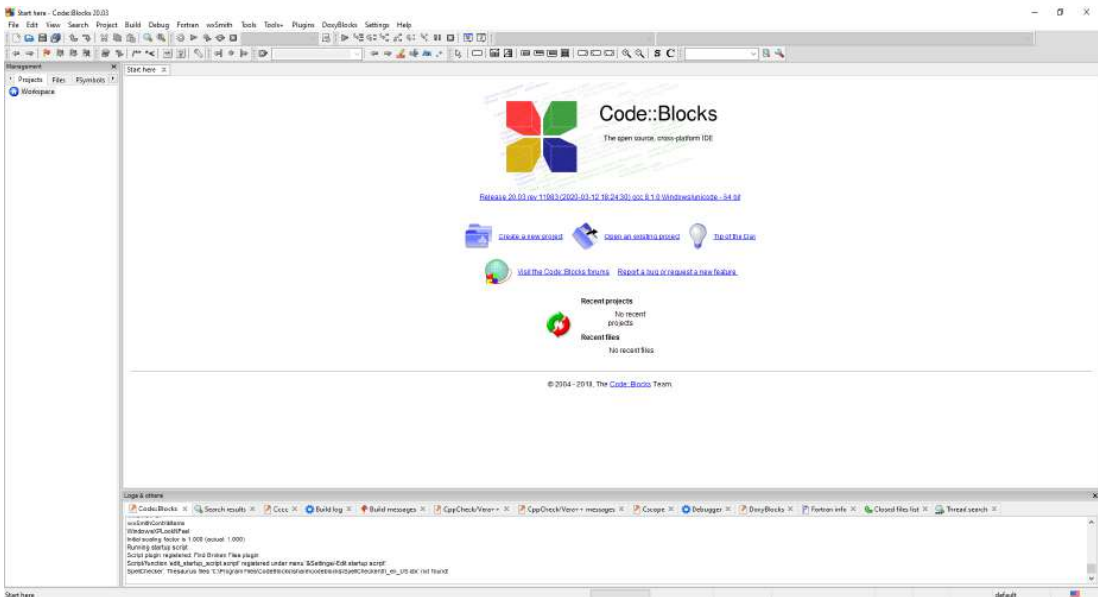


10) Klik Tombol Finish

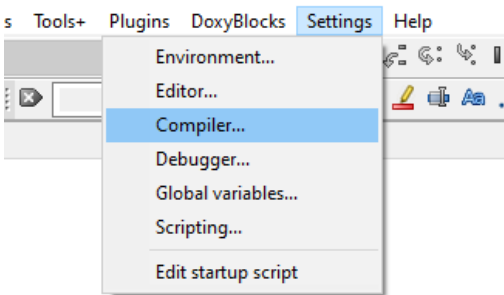
3. Melakukan setting Compiler

Jika telah selesai melakukan install IDE Code::Blocks, Langkah selanjutnya memastikan pengaturan Lokasi Compiler, dengan Langkah sebagai berikut:

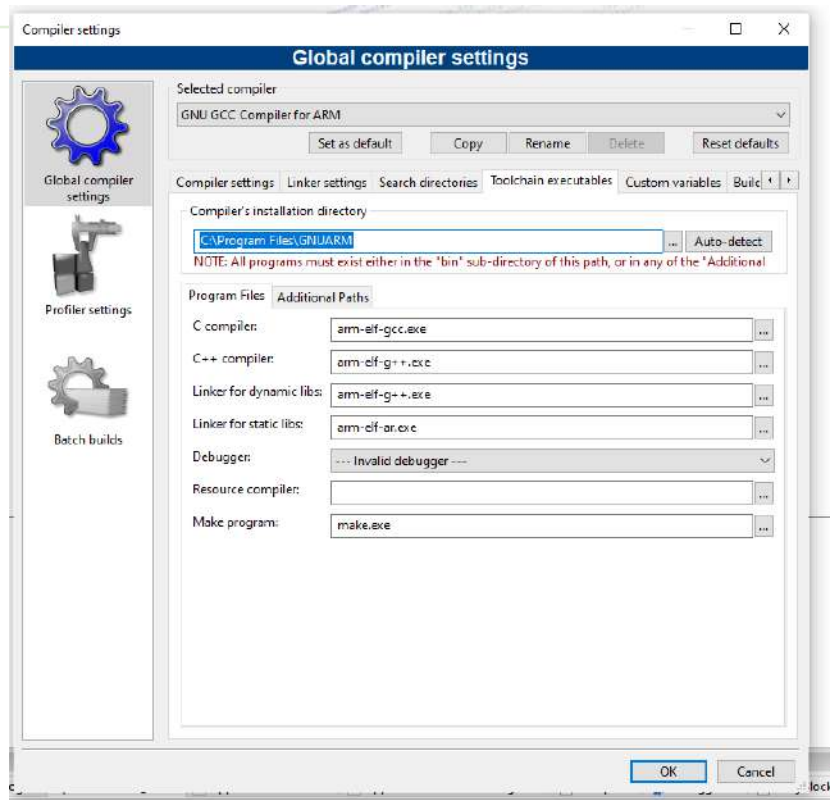
1) Buka IDE Code::Blocks



2) Klik menu Setting, kemudian pilih Compilers



3) Global Compiler settings



Selected compiler: GNU GCC Compiler for ARM

4) Klik tombol OK untuk selesai.

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	2 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa mampu menuliskan ulang kode program dan memahami cara menjalakkannya dalam IDE yang digunakan		
Kata kunci	<i>Algoritma, Diagram Alir, Pemrograman, Bahasa C</i>		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana menulis kode program dengan menggunakan IDE Bahasa C, guru akan memberikan contoh penggunaan IDE Bahasa C dan menulis kode pogram dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	<p>Materi Ajar: Menulis ulang kode program Cara menjalankan kode program pada IDE Bahasa C</p> <p>Alat: Gunting</p> <p>Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas, lem, karton</p>		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-4: Membuat Program Pertama dengan Bahasa C (2 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Mengenal Algoritma dan Pemrograman	Siswa mampu menuliskan ulang kode program dan memahami cara menjalakkannya dalam IDE yang digunakan	2 JP	Lembar Kerja Keaktifan Kelompok

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi perkakas IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik melakukan kolaborasi dan berdiskusi membuat program pertama dalam Bahasa C	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Sebutkan macam-macam bahasa pemrograman yang anda ketahui ?
- 2. Sebutkan Struktur Program Bahasa C ?

E. Kegiatan Pembelajaran

Pendahuluan

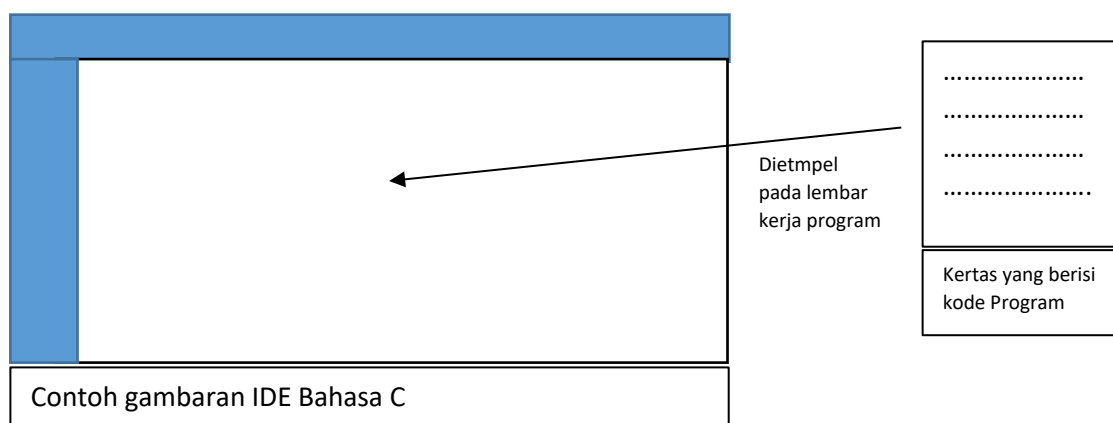
- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru menjelaskan IDE Bahasa C dengan **Code::Blocks**:

- Guru dapat menggunakan alat bantu berupa screenshot lembar IDE dan ditempatkan di papan tulis.
 - Guru menjelaskan fungsi *menubar* dan *toolbar* dengan menggunakan media lembar screenshot yang telah disediakan.
2. Guru menjelaskan struktur program Bahasa C (struktur utama Bahasa C dan komponen program Bahasa C)
 3. Guru menjelaskan cara membuat program pertama dengan Bahasa C:
 - Guru menyiapkan potongan-potongan screenshot IDE Bahasa C **Code::Blocks**.
 - Guru membuat media simulasi dari karton / papan tulis sehingga terbentuk seperti tampilan IDE Bahasa C Code::Blocks sebenarnya.



- Guru melakukan simulasi dengan menggunakan alat bantu IDE yang telah disiapkan terlebih dahulu. Dari membuat lembar kerja program, menyimpan, dan membuat program pertama dengan Bahasa C.
4. Guru mengarahkan siswa untuk mengerjakan aktivitas Latihan Membuat Program Pertama dengan Bahasa C.
 5. Guru meminta siswa menuliskan kode program pada lembar kertas dan meminta siswa untuk menempelkan pada Lembar IDE yang telah disediakan dan melakukan simulasi run untuk mengecek apakah kode program ditulis sudah benar atau salah.
 6. Guru membentuk kelompok minimal 2 orang siswa untuk menuliskan kode program dan siswa saling bertukar jawaban untuk melakukan koreksi (berfungsi sebagai *debug dan run*).
 7. Guru mengarahkan setiap Kelompok membuat kerangka ilustrasi penelusuran kode program (buat dalam bentuk *diagram alir, pseudocode*, dan penelusuran program *input-proses-output*)
 8. Guru mengulas kode program yang masih menjadi kesalahan penulisan kode program.
 9. Guru menginformasikan hasil pengerjaan Latihan membuat program pertama dengan Bahasa C dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang Bahasa C?	
Apa yang kalian rasakan ketika mencoba membuat program pertama Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara <i>unplugged</i> ?	

G. Lembar Kerja Peserta Didik

Aktivitas, Latihan Membuat Program Pertama dengan Bahasa C

Petunjuk:

Setelah mendengarkan penjelasan dari guru, lakukan sebagai berikut:

- 1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
- 2. Siswa mengerjakan soal secara berkelompok yang terdiri dari 2 orang.

Soal:

- 1. Siswa menjelaskan apa yang dimaksud dengan fungsi dari tombol berikut:

	Jawab:
	Jawab:
	Jawab:
	Jawab:

- 2. Siswa Membuat Program Pertama dalam Bahasa C

Deskripsi Soal

Buatlah program untuk menampilkan / mencetak kata "Halo Dunia". Langkah-langkah yang harus dilakukan sebagai berikut:

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Gunakan satu lembar kertas sebagai lembar kerja program
- Tuliskan Langkah-langkah pada lembar kerja program, Langkah-langkah ini harus dituliskan pada Blok Komentar.

a. Siswa Menuliskan program dengan Bahasa C sebagai berikut:





```
/* Program Pertamaku: */
/* Membuat program untuk mencetak "Halo Dunia!" */
/*
1. Buka IDE Bahasa C menggunakan code::blocks
2. Menyiapkan Lembar Kerja program
   a. Klik Menubar File
   b. Klik Submenubar File, New, Empty File
3. Lembar Kerja Program akan terbentuk dengan nama *Untitled1
4. Buat Folder/Directory untuk menyimpan program yang dibuat.
5. Simpan terlebih dahulu *Untitled1 dengan nama latihan1 pada folder/directory yang telah dibuat.
6. Menulis program,
*/
#include <stdio.h>
int main() {
    printf("Halo Dunia!\n");
    return 0;
}
```

- b. Jika sudah selesai menulis program, maka selanjutnya jalankan Program dengan klik tombol “**Run**” atau “**Debug dan Run**”. Untuk menjalankan kode program, Siswa melakukan simulasi Klik tombol Run pada Program, siswa bertugas melakukan *Run* atau *debug dan Run* melakukan pengecekan kode program yang dituliskan, buat dalam bentuk tabel ilustrasi
- c. Minta Kelompok lain untuk melakukan pengecekan dan menuliskan hasil penelusuran dari program yang dibuat dalam bentuk tabel ilustrasi.

3. Tentukan hasil dari Kode Program berikut ini:

<pre>#include <stdio.h> int main() { int panjang=10; int lebar = 5; float luas; luas = panjang * lebar; printf("luas persegi panjang adalah %.f", luas); return 0; }</pre>	Jawab:
--	---

Jawaban:
1. Yang dimaksud dengan fungsi dari tombol berikut:

	Jawab: Tambah lembar kerja program
	Jawab: Run Program
	Jawab: Save / Simpan
	Jawab: Build dan Run Program

- 2. Menulis program di lembar kerja program
 - a. Siswa menulis di kertas sebagai media lembar kerja program
 - b. Penelusuran dalam bentuk ilustrasi:

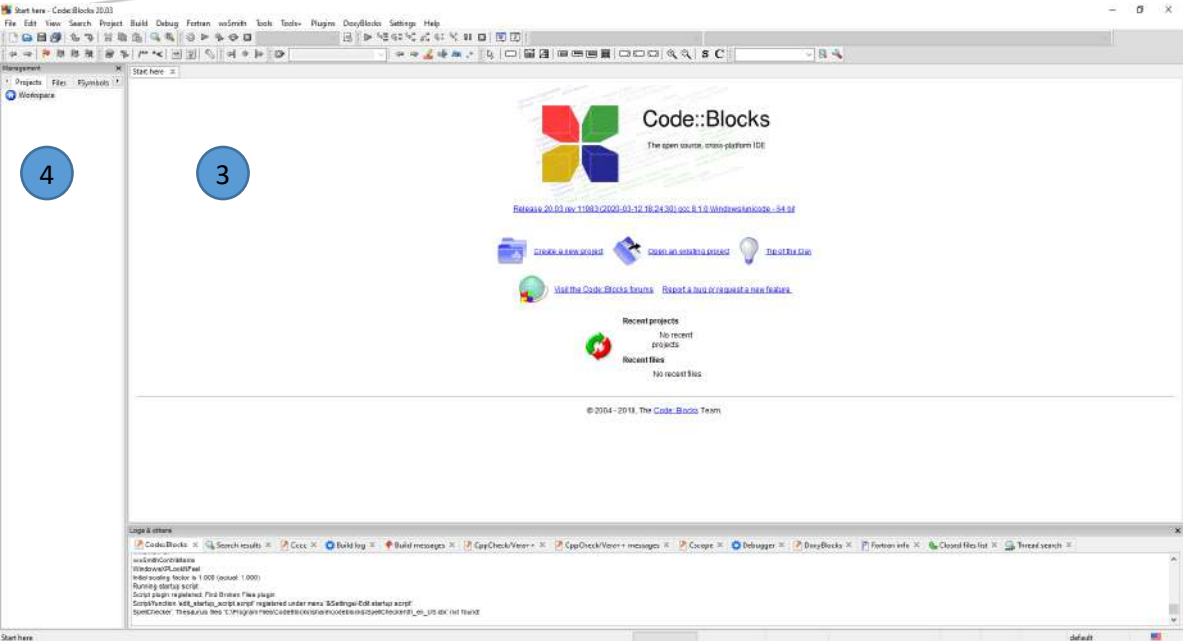
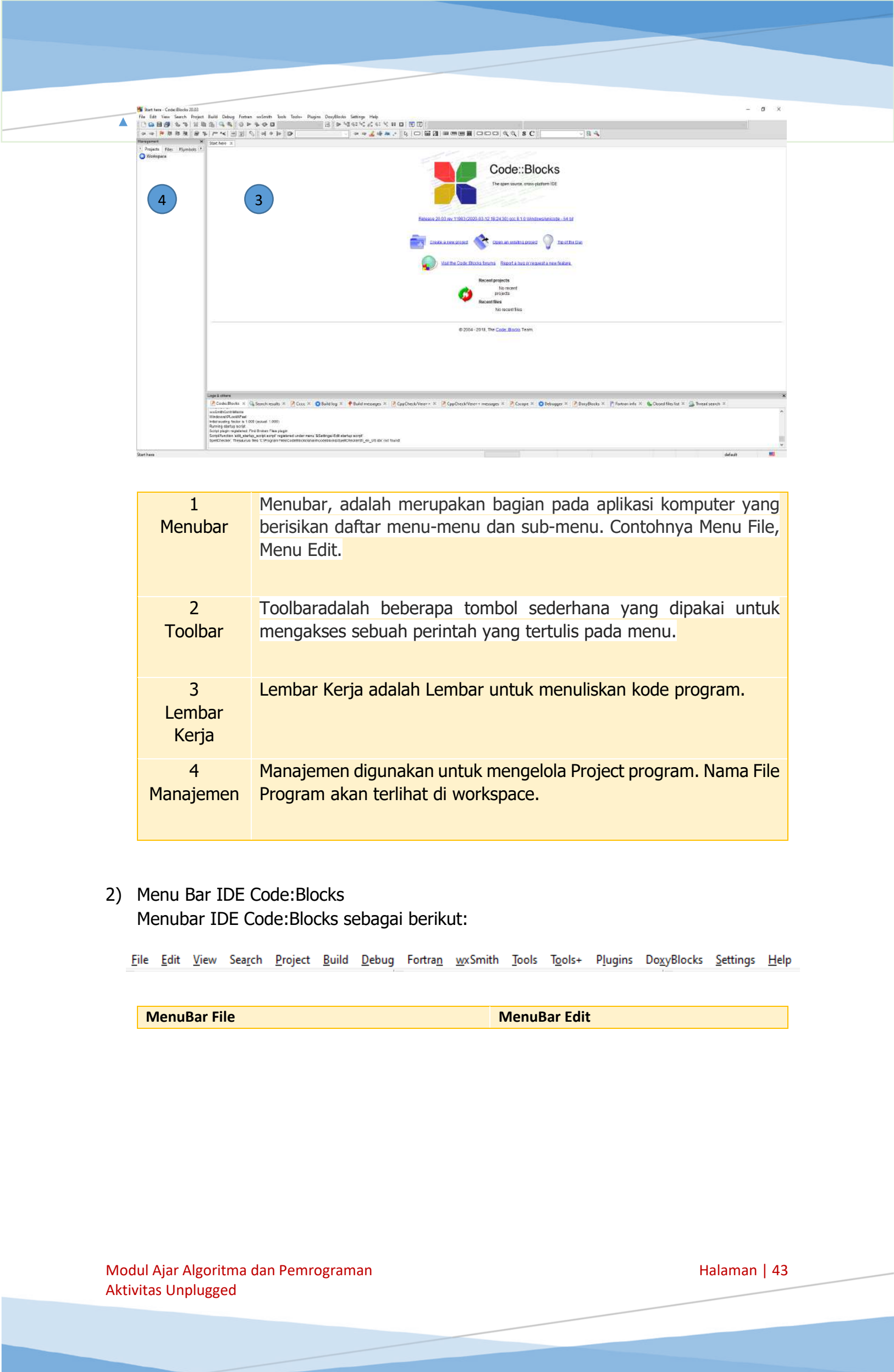
Input	Nilai masukan: <tidak ada nilai masukan> Nilai Variable: <tidak ada nilai variabel>
Proses	Menampilkan kata "Halo Dunia" menggunakan perintah printf
Output	Halo Dunia

- c. Jawaban sama dengan yang b.
- 3. Jawab: luas persegi panjang adalah 50

H. Bahan Bacaan

Menjelaskan IDE Bahasa C dengan Code::Blocks

- 1) IDE Code::Block
 - 2
- 1



1 MenuBar	Menubar, adalah merupakan bagian pada aplikasi komputer yang berisikan daftar menu-menu dan sub-menu. Contohnya Menu File, Menu Edit.
2 Toolbar	Toolbar adalah beberapa tombol sederhana yang dipakai untuk mengakses sebuah perintah yang tertulis pada menu.
3 Lembar Kerja	Lembar Kerja adalah Lembar untuk menuliskan kode program.
4 Manajemen	Manajemen digunakan untuk mengelola Project program. Nama File Program akan terlihat di workspace.

2) Menu Bar IDE Code:Blocks
Menubar IDE Code:Blocks sebagai berikut:

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

MenuBar File	MenuBar Edit
--------------	--------------

Start here - Code::Blocks 20.03

File

Edit

View

Search

Project

Build

Debug

Fortran

wxSmith

Tools

Tools+

Plugins

New

Open...

Open With Hex Editor

Open default workspace

Recent projects

Recent files

Import project

Save file

Save file as...

Save project

Save project as...

Save project as template...

Save workspace

Save workspace as...

Save everything

Close file

Close all files

Close project

Close workspace

Print...

Export

Properties...

Quit

Empty file

Class...

Project...

Build target...

File...

Custom...

From template...

Nassi Shneiderman diagram

Undo

Redo

Clear changes history

Cut

Copy

Paste

Swap header/source

Highlight mode

Bookmarks

Folding

End-of-line mode

File encoding

Special commands

Editor tweaks

Select all

Select next occurrence

Select skip to next occurrence

Comment

Uncomment

Toggle comment

Stream-comment

Box-comment

Goto matching brace

Show call tip

Complete code

Auto-complete

Rename symbols

Show tooltip

Next call tip

Previous call tip

Spelling...

Thesaurus...

Add Todo item...

MenuBar View

Perspectives

Toolbars

Manager

Logs

Start Page

Script console

Hide editor tabs

Status bar

Symbols browser

Code snippets

Fortran Call/Called-By tree

Fortran symbols browser

Man pages viewer

Highlighted Occurrences

Open files list

Closed file list

Thread search

Todo list

Preserve open editors

Full screen

Focus editor

Reopen last closed editor

Focus management panel

Focus logs and others

Switch tabs

Jump

Browse Tracker

Nassi-Shneiderman diagram

Focus Thread Search

MenuBar Build

File

Edit

View

Search

Project

Build

Build

Compile current file

Run

Build and run

Rebuild

Clean

Build workspace

Rebuild workspace

Clean workspace

Abort

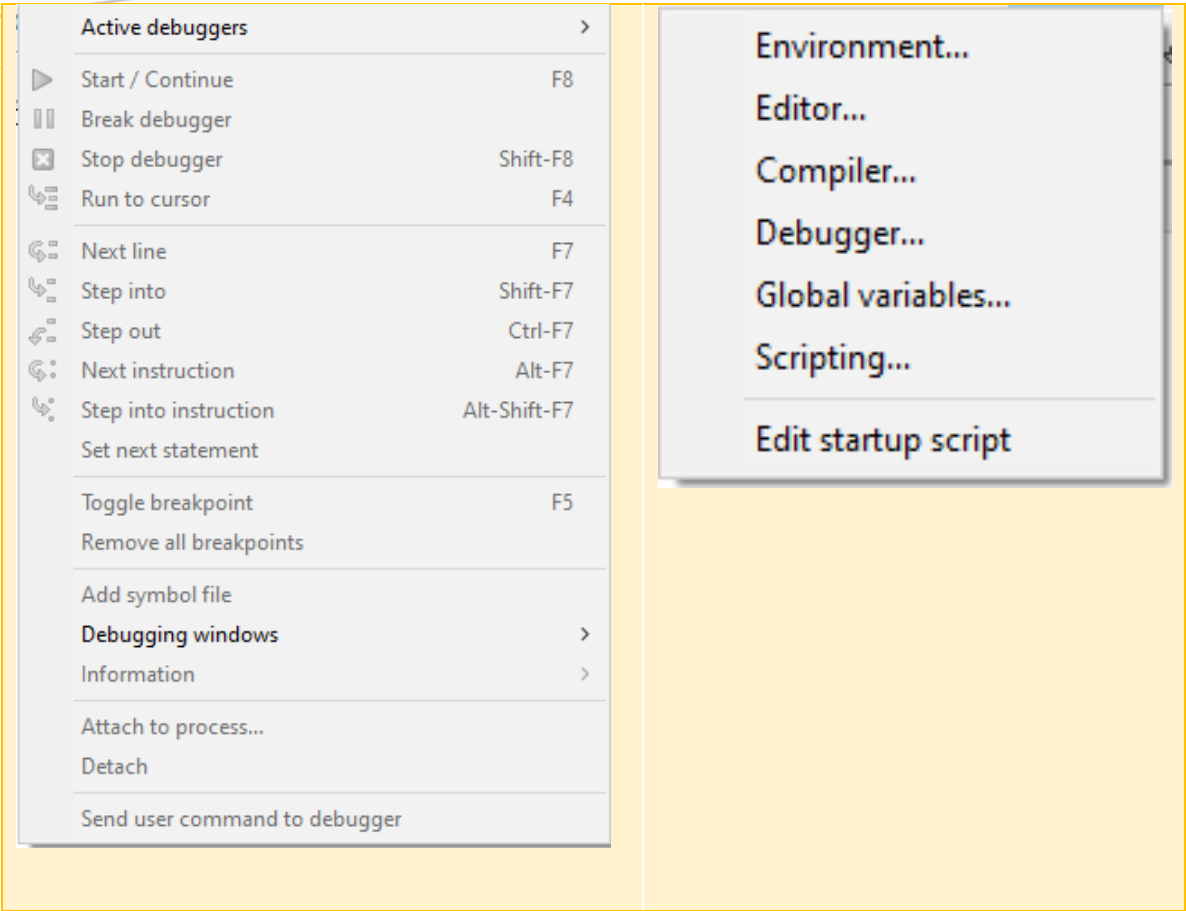
Errors

Select target

Export Makefile




Menubar Debug

Menubar Settings

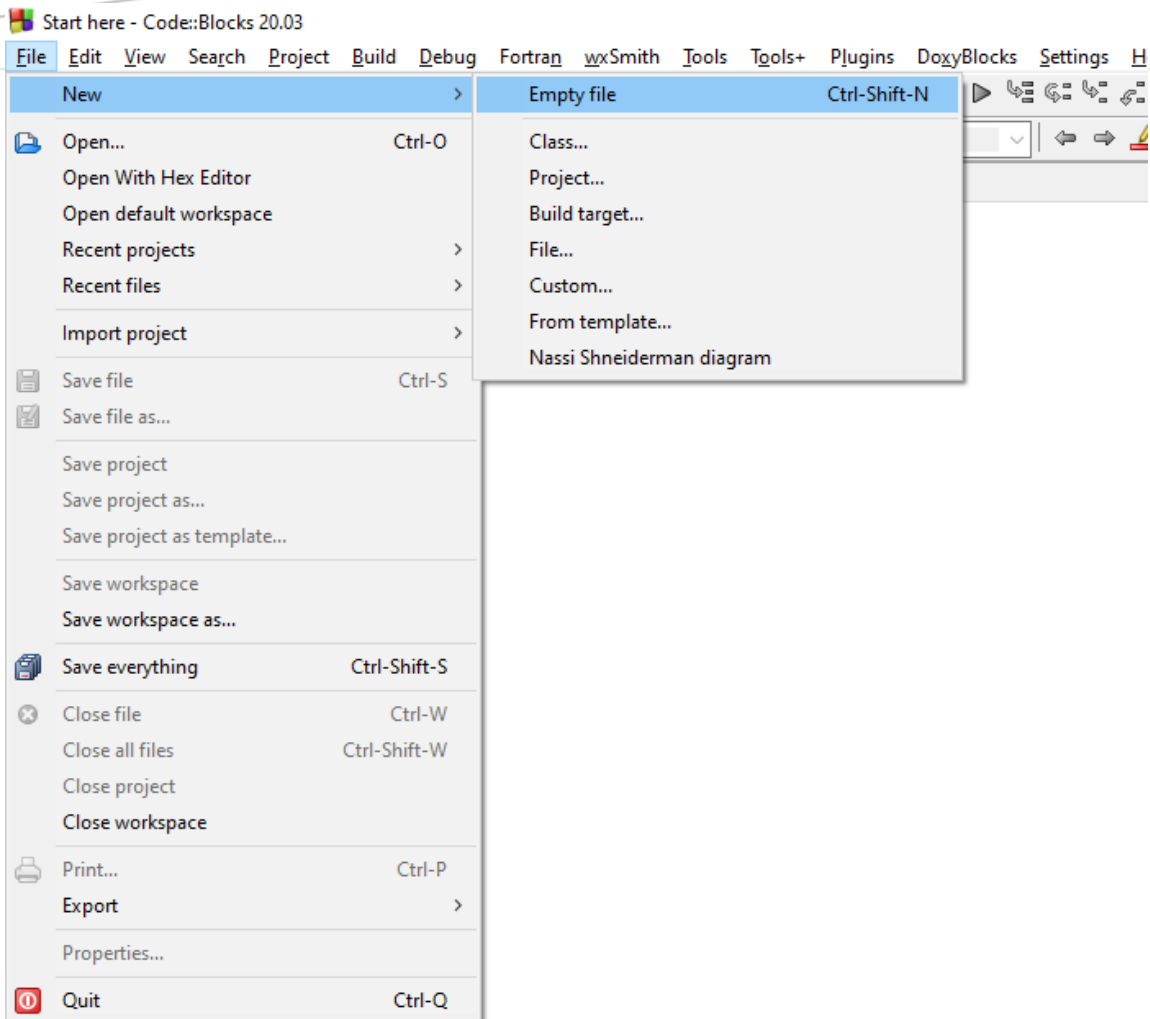


3) Toolbar IDE Code::Blocks



	Tambah lembar kerja program
	Save / Simpan
	Run Program
	Build Program
	Build dan Run Program

- 4) Membuat Lembar Kerja Program
- Terdapat dua cara untuk membuat lembar kerja program, yaitu:
- a. Lembar kerja program dibuat melalui menubar

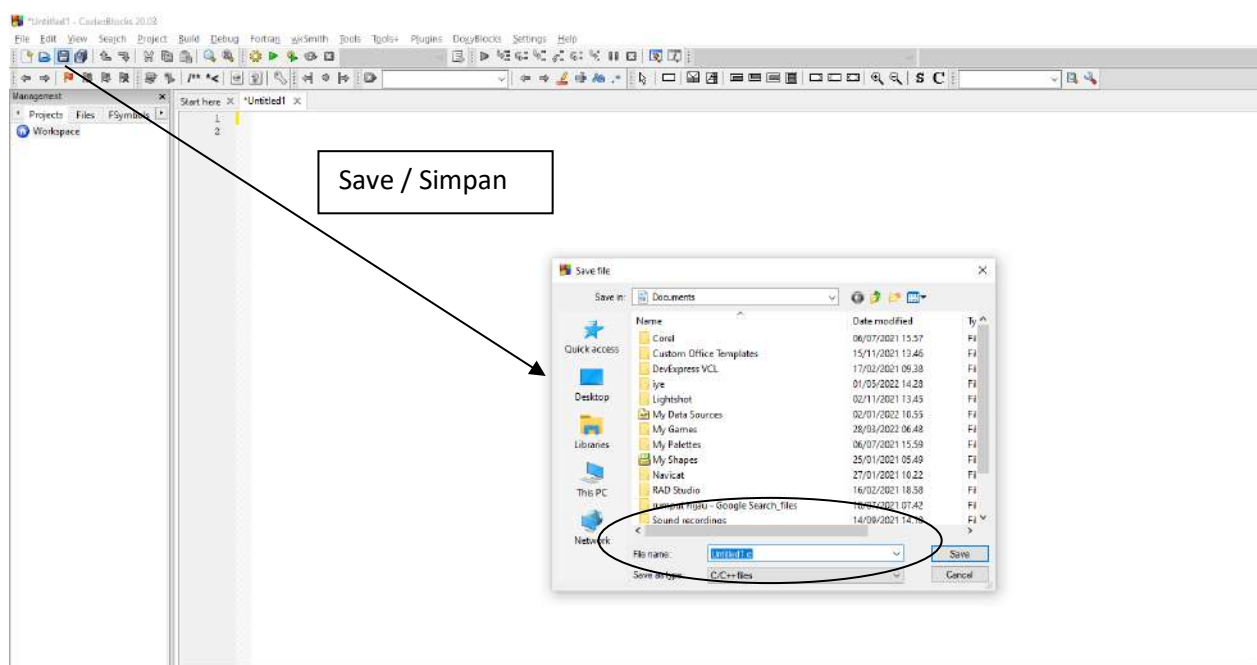


- Klik Menubar **File**, kemudian **New**, kemudian **Empty File**

b. Lembar kerja program dibuat melalui toolbar

Setelah klik empty file, maka akan terbentuk file dengan nama ***Untitled1**, **tanda bintang (*)** artinya file lembar kerja belum tersimpan.

5) Menyimpan Lembar Kerja



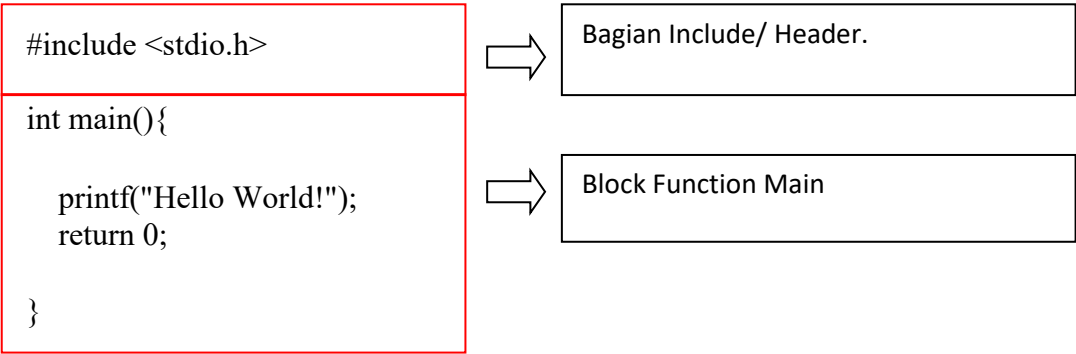
- Klik Tombol Save/Simpan

- Akan Muncul Dialog Save File
- Ubah File Name dengan nama yang sesuai dengan program yang dibuat. Misalnya LuasPersegiPanjang
- Klik Tombol Save

1. Membuat Program Pertama dengan Bahasa C

Yang perlu diketahui dalam menulis kode program menggunakan Bahasa C adalah sebagai berikut:

1) Struktur program Bahasa C



- 2) **#include** berfungsi untuk mengimpor fungsi-fungsi yang sudah didefinisikan pada *header file*.
- 3) Fungsi **main()** adalah fungsi utama dalam program. Fungsi ini akan dieksekusi pertamakali saat program dijalankan.
- 4) *Statement* adalah perintah-perintah untuk melakukan sesuatu. Misalnya perintah menampilkan kata "Hello Word". Atau perintah menghitung luas. Statement ini dituliskan didalam Block Function Main.
- 5) Penulisan kode program dalam Bahasa C bersifat case sensitive, artinya huruf besar kecil berbeda.
- 6) Setiap penulisan statement diakhiri dengan titik koma (;)
- 7) Blok Komentar digunakan untuk menjelaskan sesuatu, penulisan komentar diawali dengan /* dan diakhir dengan */ atau diawali dengan // jika itu komentar satu baris.
Contoh
/*
 Ini adalah program pertama menggunakan Bahasa C
 Program ini untuk menampilkan kata Hello Word
*/

// Ini adalah program pertama menggunakan Bahasa C
// Program ini untuk menampilkan kata Hello Word
- 8) Setiap perintah komentar tidak akan di jalankan oleh program

2. Komponen Program dalam Bahasa C

Sebelum membuat program yang lebih kompleks, yang melibatkan ekspresi, struktur kontrol keputusan, atau struktur kontrol perulangan, kita perlu memahami makna dari berbagai komponen penyusun bahasa C, di antaranya meliputi kata kunci, identifier, variabel, tipe data, dan konstanta.

1) Kata Kunci (*Keyword*)

Kata kunci (*Keyword*) merupakan kata yang telah memiliki makna khusus yang tidak dapat diubah oleh pemrograman. Dengan kata lain, kalian tidak dapat menggunakan sebagai suatu identifier. Terdapat 32 kata kunci standar pada bahasa C, yaitu:

auto, double, int, struct, break, else, long, switch, case, enum, register, typedef, char, extern, return, union, continue, for, signed, void, do, if, static, while, default, goto, sizeof, volatile, const, float, short, dan unsigned.

Dengan hanya 32 kata kunci, kita dapat menulis berbagai program apa saja dalam bahasa C.

2) Identifier

Identifier adalah nama unik yang dapat kalian ingat yang diberikan ke dalam *entitas* program C, seperti variabel dan fungsi. *Identifier* terdiri atas serangkaian karakter dengan aturan berikut:

- Tidak boleh sama dengan kata kunci (keyword) dalam bahasa C.
- Disusun dari kombinasi huruf (besar dan kecil), angka, dan underscore "_".
- Harus dimulai dengan huruf atau underscore.
- Bersifat case-sensitive, atau sensitif terhadap huruf besar atau kecil (kapitalisasi karakter). Dengan kata lain, sisi dan Sisi akan dianggap sebagai dua identifier yang berbeda

Meskipun dalam penamaan penulisan identifier sangat bebas, ada beberapa yang perlu diketahui untuk menghindari kebingungan dalam membaca dan menulis kode program seperti berikut:

- Nama yang diawali oleh underscore digunakan untuk keperluan tertentu dan tidak seharusnya digunakan dalam membuat program di unit ini.
- Nama variabel dan fungsi harus ditulis dengan huruf non-kapital. Akan tetapi, apabila terdiri atas dua atau lebih kata, kalian dapat menggunakan teknik menulis dengan standar camel case. Huruf awal kata, selain kata pertama, ditulis menggunakan huruf kapital. Contoh: totalHargaBarang, namaMahasiswa.
- Hindari menggunakan identifier yang sangat mirip dalam satu kode program. Misalnya, totalHarga dengan total_harga. Hindari juga dua atau lebih variabel yang hanya berbeda di kapitalisasi seperti totalharga dan totalHarga.
- Hindari identifier yang terlihat mirip, misalnya karakter 'I', '1', dan 'l' terlihat sangat mirip satu sama lain. Catatan: karakter yang disebutkan ialah huruf i kapital, angka 1, dan huruf L non-kapital.
- Identifier haruslah bersifat mnemonic. Untuk variabel, artinya identifier variabel tersebut harus menunjukkan isi dari variabel tersebut. Untuk fungsi, identifier menunjukkan apa yang dilakukan oleh fungsi tersebut.
- Dalam dunia pemrograman, ada beberapa nama variabel yang menjadi kebiasaan untuk dipakai, misalnya nama variabel untuk mengunjungi elemen tabel, dipakai indeks i, j, dan k.
- Konstanta dinamai dengan huruf kapital. Misalnya, PI

3) Tipe Data

Komputer dapat mengolah data yang beragam. Pada dasarnya, data yang diolah oleh komputer, baik berupa numerik ataupun karakter, akan disimpan dalam bentuk biner. Oleh karena itu, nilai yang kalian masukkan dalam komputer pastilah akan disimpan dalam bentuk biner. Program perlu mengetahui bagaimana bilangan biner dibaca sehingga diperlukan suatu mekanisme untuk memberi tahu program tentang data yang kita simpan pada variabel tersebut. Hal ini diakomodir melalui tipe data. Suatu tipe data akan memiliki nama tipe, jenis data yang disimpan, dan rentang yang berbeda. Pada bahasa C, terdapat beberapa tipe data dasar yang dapat digunakan

Tabel Tipe Data Bahasa C Berikut dengan Ukuran Memori dan Rentang Nilainya

Nama Tipe	Jenis Data	Ukuran Memori	Rentang
int	Bilangan bulat	4 byte	-2.1x10 ⁹ hingga 2.1x10 ⁹
short	Bilangan bulat	2 byte	-32768 hingga 32767
long	Bilangan bulat	8 byte	-9.2x10 ⁸ hingga 9.2x10 ⁸
float	Bilangan riil	4 byte	1.2x10 ³⁸ hingga 3.4x10 ³⁸
double	Bilangan riil	8 byte	2.3x10 ³⁰⁸ hingga 1.7x10 ³⁰⁸
char	Karakter*	1 byte	-127 hingga 128

* Seperti yang termuat pada kode ASCII: ascii-code.com

Perhatikan bahwa rentang yang diberikan memungkinkan nilai negatif hingga positif, atau disebut tipe data *signed*. Apabila kita menambahkan kata kunci *unsigned* di depan tipe data, tipe data tersebut hanya akan menampung bilangan positif dengan rentang dari 0 hingga $2^{\text{jumlah bit}} - 1$. Gunakan tipe data yang sesuai dengan kebutuhan. Sebagai contoh, saat mengolah data usia manusia dalam satuan tahun, cukup menggunakan tipe data *short* yang memerlukan memori lebih kecil. Jika membutuhkan pengolahan nilai yang mengandung koma gunakan *float*.

4) Variabel

Pada matematika, kita mengenal variabel sebagai sebuah wadah untuk menyimpan suatu nilai. Variabel pada program memiliki fungsi yang sama. Nilai yang diberikan pada sebuah variabel akan disimpan di memori komputer. Komputer memberikan alamat pada lokasi memori tersebut yang sulit diingat oleh manusia. Oleh karena itu, variabel diberikan nama simbolik yang mudah untuk diingat oleh kalian dengan menggunakan *identifier*.

Dalam bahasa C, variabel perlu dideklarasikan dengan memberikan tipe data dan identifiers sebelum dapat digunakan. Deklarasi dapat dilakukan dengan menggunakan pernyataan berikut:

```
<tipe_data> <nama_variabel>;
```

Contoh:

```
int panjang;
```

```
int lebar;
```

Pada saat deklarasi, variabel juga dapat diberikan nilai awal, misalnya dalam bentuk:

```
<tipe_data> <nama_variabel> = <nilai_awal>;
```

Contoh

```
int panjang = 25;
int lebar = 4;
```

Variabel dengan tipe yang sama dapat dideklarasikan secara ringkas seperti:

```
<tipe_data> <nama_variabel1>,<nama_variabel2> ;
```

Contoh: **int** panjang, lebar;

Untuk lebih jelasnya, perhatikan table contoh untuk mendklarasikan sebuah variable sebagai berikut:

Tipe Data	Identifier	Deklarasi	Deklarasi dengan Nilai Awal
int	totalHarga	int totalHarga;	int totalHarga = 15000;
short	usia	short usia;	short usia = 18;
long	jumlahAtom	long jumlahAtom;	long jumlahAtom = 9123151252214;
float	jarak	float jarak;	float jarak = 2.28;
double	galat	double galat;	double galat= 0.0000000001234;
char	huruf	char huruf;	char huruf = 'a';

Tempat deklarasi variabel akan berpengaruh pada penggunaan variabel tersebut. Apabila deklarasi variabel dilakukan di dalam sebuah fungsi, variabel tersebut hanya dapat digunakan di dalam fungsi tersebut. Variabel seperti ini disebut variabel lokal. Apabila deklarasi dilakukan di luar fungsi, variabel tersebut akan dapat diakses di bagian program mana pun. Variabel ini disebut variabel global.

5) Konstanta

Berbeda dengan variabel yang nilainya dapat berubah, konstanta tidak dapat diubah. Saat dideklarasikan, nilai dari konstanta diberikan dan tidak dapat diubah kembali. Apabila kalian memaksa mengubah konstanta, kompilator akan memberikan pesan kepada kalian. Penggunaan konstanta yang lazim ialah untuk menyimpan nilai konstan seperti pi (π), rho (ρ), dan konstanta lainnya yang lazim digunakan.

Konstanta dapat dideklarasikan seperti variabel, dengan menambah kata kunci const di depan tipe data. Nilai awal harus langsung diberikan pada saat deklarasi. Misalnya, deklarasi konstanta pi dapat dilakukan sebagai berikut:

```
const float PI = 3.14;
```

6) *Operand* dan *Operator*

Operand adalah nilai asal yang dipakai dalam sebuah proses operasi. Sedangkan *Operator* adalah instruksi yang diberikan untuk mendapatkan hasil dari proses tersebut. Biasanya *operator* berupa karakter matematis atau perintah singkat sederhana. Sebagai contoh, pada operasi: 7 + 2. Angka 7 dan 2 disebut sebagai *operand*, sedangkan tanda tambah (karakter +) adalah *operator*. Jenis-jenis Operator dalam Bahasa Pemrograman C sebagai berikut:

a. *Operator Aritmatika*

Operator	Keterangan	Contoh
+	Penambahan	a = 7 + 3
-	Pengurangan	a = 7 - 3
*	Perkalian	a = 7 * 3
/	Pembagian	a = 9 / 3
%	Sisa hasil bagi (Modulus)	a = 7 % 3

b. *Operator Perbandingan/ Relational*

Operator	Keterangan	Contoh	Hasil
==	Sama dengan	7 == 7	True
!=	Tidak sama dengan	7 != 7	False
*	Perkalian	a = 7 * 3	21
/	Pembagian	a = 9 / 3	3
%	Sisa hasil bagi (Modulus)	a = 7 % 3	1

Contoh Latihan 1: Membuat Program Bahasa C

Deskripsi Soal:

Membuat program untuk menampilkan / mencetak kata "Halo Dunia". Langkah-langkah yang harus dilakukan sebagai berikut:

Petunjuk:

- 1. Siapkan kertas dan alat tulis sebagai media simulasi.
- 2. Gunakan satu lembar kertas sebagai lembar kerja program
- 3. Tuliskan Langkah-langkah pada lembar kerja program, Langkah-langkah ini harus dituliskan pada Blok Komentar.

Langkah-langkah membuat program:

- 1. Buka IDE Bahasa C menggunakan code::blocks (gunakan simulasi screenshoot IDE Bahasa C yang sudah disusun)
- 2. Menyiapkan Lembar Kerja program
 - a. Klik Menubar File

- b. Klik Submenubar File, New, Empty File
3. Lembar Kerja Program akan terbentuk dengan nama ***Untitled1** (siulasi menyiapkan kertas sebagai lembar kerja program)
4. Buat Folder/Directory untuk menyimpan program yang dibuat.
5. Simpan terlebih dahulu *Untitled1 dengan nama **latihan1** pada folder/directory yang telah dibuat.
(pemberian nama file di lembar kerja program dalam hal ini di kertas yang sudah disiapkan)
6. Menulis program, berikut:

```
/* Program Pertamaku: */  
/* Membuat program untuk mencetak "Halo Dunia!" */  
/*  
1. Buka IDE Bahasa C menggunakan code::blocks  
2. Menyiapkan Lembar Kerja program  
   a. Klik Menubar File  
   b. Klik Submenubar File, New, Empty File  
3. Lembar Kerja Program akan terbentuk dengan nama *Untitled1  
4. Buat Folder/Directory untuk menyimpan program yang dibuat.  
5. Simpan terlebih dahulu *Untitled1 dengan nama latihan1 pada folder/directory yang  
   telah dibuat.  
6. Menulis program,  
*/  
#include <stdio.h>  
int main() {  
    printf("Halo Dunia!\n");  
    return 0;  
}
```

7. Jika sudah selesai menulis program, maka selanjutnya jalankan Program dengan klik tombol **"Run"** atau **"Debug dan Run"**.
8. Minta teman sebangku/kelompok untuk melakukan *Run* atau *debug dan Run*, untuk mengecek program yang dituliskan.
9. Teman/Kelompok yang berfungsi sebagai **"Run"** atau **"Debug dan Run"** menuliskan hasil penelusuran dari program yang dibuat dalam bentuk table ilustrasi.

Contoh Latihan 2: Membuat Program Bahasa C

Deskripsi Soal:

Membuat program untuk menampilkan Luas Persegi Panjang. Langkah-langkah yang harus dilakukan sebagai berikut:

Petunjuk:

1. Siapkan kertas dan alat tulis sebagai media simulasi.
2. Gunakan satu lembar kertas sebagai lembar kerja program
3. Tuliskan Langkah-langkah membuat program pada lembar terpisah dari lembar kerja program.

Langkah-langkah membuat program:

1. Buka IDE Bahasa C menggunakan code::blocks
2. Menyiapkan Lembar Kerja program
 - c. Klik Menubar File
 - d. Klik Submenubar File, New, Empty File
3. Lembar Kerja Program akan terbentuk dengan nama ***Untitled1**
4. Buat Folder/Directory untuk menyimpan program yang dibuat.
5. Simpan terlebih dahulu *Untitled1 dengan nama **latihan2** pada folder/directory yang telah dibuat.
6. Menulis program, berikut:

```
// Membuat Program menghitung Luas persegi Panjang

#include <stdio.h>

int main()
{
    int panjang=10;
    int lebar = 5;
    float luas;

    luas = panjang * lebar;

    printf("luas persegi panjang adalah %.d", luas);

    return 0;
}
```

7. Jika sudah selesai menulis program, maka selanjutnya jalankan Program dengan klik tombol **"Run"** atau **"Debug dan Run"**.
8. Minta teman sebangku/kelompok untuk melakukan *Run* atau *debug dan Run*, untuk mengecek program yang dituliskan.
9. Teman/Kelompok yang berfungsi sebagai **"Run"** atau **"Debug dan Run"** menuliskan hasil penelusuran dari program yang dibuat dalam bentuk table ilustrasi.

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	2 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa memahami konsep <i>input output</i> dan mampu menuliskan program sederhana yang membaca dan menulis		
Kata kunci	<i>input, ouput, Algoritma, Diagram Alir, Pemrograman, Bahasa C</i>		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana penggunaan input dan ouput dalam Bahasa C dan menulis kode program dengan menggunakan IDE Bahasa C, guru akan memberikan contoh penggunaan input dan ouput dalam bahawa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	Materi Ajar: Konsep <i>input</i> dan <i>output</i> Alat: Gunting Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas. lem, karton		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-5: Input dan Output Program (3 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durai (JP)	Asesmen
Mengenal Algoritma dan Pemrograman	Siswa memahami konsep input output dan mampu menuliskan program sederhana yang membaca dan menulis	3 JP	Lembar Kerja Keaktifan

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas .

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik mengeksplorasi perintah input dan output dalam Bahasa C, secara mandiri maupun kolaborasi.	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat program Bahasa C secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Perintah output dalam Bahasa C menggunakan apa ?
- 2. Apa yang perlu diperhatikan dalam membuat variable dalam bahasa C?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;

5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru menyampaikan materi dan dapat menjelaskan langsung mengenai program yang akan dibuat pada latihan ini.
- 2. Guru dapat menggunakan media lembar screenshot IDE Kembali atau dapat mengangap bahwa papan tulis itu sebagai media lembar kerja program.
- 3. Guru membentuk kelompok yang terdiri dari 2-5 orang:

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan Ilustrasi Penelusuran program <i>input-proses-output</i>

- Guru membagikan kode program ke setiap kelompok yang berisi kode program yang benar dan yang salah.

Contoh Kode Program yang benar

<pre>// Membuat Program Penjumlahan #include <stdio.h> int main() { int nilaiA, nilaiB; int hasiljumlah; scanf("%d", &nilaiA); scanf("%d", &nilaiB); hasiljumlah= nilaiA + nilaiB; printf("hasil jumlah: %.d", hasiljumlah); return 0; }</pre>	<p>Soal 1: Buat Skenario pertama nilaiA diinputkan dengan angka 7 dan nilaiB diinputkan dengan angka 15.</p> <p>Soal 2: Buat Skenario kedua nilaiA diinputkan dengan huruf karakter "A" dan nilaiB diinputkan dengan angka 17.</p> <p>Soal 3: Buat Skenario input yang lain</p>
---	---

Contoh Kode Program yang salah

<pre>// Membuat Program perkalian #include <stdio.h> int main() { int nilaiA, nilaiB;</pre>	<p>Soal 1: Buat Skenario pertama nilaiA diinputkan dengan angka 7 dan nilaiB diinputkan dengan angka 15.</p>
---	--

<pre>int hasiljumlah; scanf("%d", &nilaiA); scanf("%d", &nilaiB); hasilkali= nilaiA x nilaiB; printf("hasil kali: %.d", hasilkali); return 0; }</pre>	<p>Soal 2: Buat Skenario kedua nilaiA diinputkan dengan huruf karakter "A" dan nilaiB diinputkan dengan angka 17.</p> <p>Soal 3: Buat Skenario input yang lain</p>
<p>Kesalahan Kode program:</p> <ul style="list-style-type: none"> Variable yang digunakan tidak sama, rumus menggunakan variable hasilkali, tetapi hasilkali tidak dibuat nama variabelnya. <p>Ubah <i>int hasiljumlah;</i> menjadi <i>int hasilkali;</i></p> <ul style="list-style-type: none"> Operator perkalian yang digunakan salah, dalam Bahasa C tidak mengenal operator x tetapi untuk perkalian menggunakan operator * 	

- Setiap siswa menjalankan sesuai dengan tugasnya masing-masing yaitu membuat diagram alir, membuat deskripsi algoritma, pseudocode, menulis kode program, melakukan debug dan run, membuat skenrario input dan ilustrasi penelusuran program *input-proses-output*
 - Siswa membuat rangkuman dari pengerjaan kode program.
 - Siswa secara bergantian untuk melakukan tugas yang berbeda.
 - Setelah itu, pancing siswa/kelompok lain untuk memperbaiki program apabila terdapat kesalahan pada program
4. Guru mengarahkan siswa ke aktivitas Ayo, Kita Berlatih **Menulis dan Memperbaiki Program**. Biarkanlah siswa mengerjakan latihan mandiri. Guru memberikan umpan balik dan *scaffolding* pada siswa yang mengalami kesulitan
 5. Guru dapat membahas aktivitas dengan meminta beberapa siswa menuliskan deskripsi dan menjelaskan hasil pekerjaannya.
 6. Guru dapat mengulas kembali beberapa kesalahan umum yang terjadi pada saat latihan menulis program. Setelah itu, siswa dapat diminta untuk memperbaiki program di sisa waktu, atau dikerjakan kembali ke rumah.
 7. Guru menginformasikan hasil pengerjaan berlatih menulis dan memperbaiki program dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang Bahasa C?	
Apa yang kalian rasakan ketika mencoba membuat program input output Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara <i>unplugged</i> ?	

G. Lembar Kerja Peserta Didik

Aktivitas Ayo, Kita Berlatih Menulis dan Memperbaiki Program

Petunjuk:

- 1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
- 2. Siswa mengerjakan soal secara berkelompok yang terdiri dari 2-5 orang.

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan Ilustrasi Penelusuran program <i>input-proses-output</i>

Soal:

- 1. Siswa membuat program untuk **membaca** tiga jenis bilangan dan **mencetaknya** kembali tiga jenis bilangan tersebut.
Skenario input Bilangan pertama merupakan bilangan bulat positif dengan nilai maksimum 100. Bilangan kedua merupakan bilangan riil positif dengan nilai maksimum 80. Bilangan ketiga merupakan bilangan bulat positif dengan nilai maksimum 40.

Jawab:

- 2. Membuat program untuk mencetak kalimat berikut:

Selamat Datang, di "Bandung dan Jakarta".

Perbaikilah Kode Program berikut, sehingga dapat mencetak/menampilkan hasil dengan benar.

```
#include <stdio.h>
int main()
{
    printf("Selamat Datang, "Bandung dan Jakarta".");
    return 0;
}
```

3. Buat kerangka ilustrasi penelusuran program, buat dalam bentuk *diagram alir*, *pseudocode*, dan penelusuran program *input-proses-output*.

Jawaban:

1. Membuat program membaca tiga jenis bilangan

```
// Membuat Program Penjumlahan
#include <stdio.h>
int main()
{
    int bilangan1, bilangan2, bilangan3;

    scanf("%d", &bilangan1);
    scanf("%d", & bilangan2);
    scanf("%d", & bilangan3);

    printf("bilangan 1 adalah: %.d", bilangan1);
    printf("bilangan 2 adalah: %.d", bilangan2);
    printf("bilangan 3 adalah: %.d", bilangan3);

    return 0;
}
```

2. Kode program setelah diperbaiki:

```
#include <stdio.h>
int main()
{
    printf("Selamat Datang, \"Bandung dan Jakarta\".");
    return 0;
}
```

3. Ilustasi penelusuran program
 - a. Ilustrai penelusuran program membaca tiga jenis bilangan

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama, dan akan berhenti di baris ketujuh (karena ada perintah *scanf* yang membutuhkan inputan), sampai pengguna/user menginputkan nilai dari bilangan. Baris kedelapan , kesembilan juga sama seperti baris ketujuh.

Setelah sampai pada baris kesebelas, maka akan menghasilkan output/keluaran:
Bilangan 1 adalah <bilangan yang sesuai dengan angka yang diinputkan>

Setelah sampai pada baris keduabelas, maka akan menghasilkan output/keluaran:
Bilangan 2 adalah <bilangan yang sesuai dengan angka yang diinputkan>

Setelah sampai pada baris ketigabelas, maka akan menghasilkan output/keluaran:
Bilangan 3 adalah <bilangan yang sesuai dengan angka yang diinputkan>

b. Ilustrai penelusuran program membaca tiga jenis bilangan

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama sampai akhir. Sampai pada baris keempat maka akan menampilkan kata **“Selamat Datang, “”Bandung dan Jakarta”**

3. Membaca dan Menulis (*Input dan Output*)

Manusia membuat program , padadasarnya untuk dapat membantu mempermudah pekerjaan manusia, program harus dilengkapi dengan kemampuan berkomunikasi. Ada banyak cara untuk berkomunikasi lewat antarmuka pengguna (user interface), tetapi bentuk komunikasi dasar yang perlu dikuasai komunikasi lewat *command line interface* (CLI). Dengan lewat CLI, kita dapat berkomunikasi dengan sebuah program menggunakan teks, dan program pun akan merespons dengan menggunakan teks. Dengan kata lain, interaksi menggunakan CLI sangat bergantung pada kemampuan program untuk membaca data yang diberikan oleh pengguna (*user*) dan menuliskan hasil pekerjaan.

Agar dapat membaca dan menulis dalam Bahasa C, program yang dibuat perlu menggunakan *header* **stdio.h** yang memuat fungsi masukan-keluaran standar menggunakan CLI. Dua fungsi utama yang dapat digunakan ialah *scanf* untuk membaca atau perintah *input* dan *printf* untuk menulis atau perintah *output*. Untuk lebih jelasnya, perhatikanlah contoh program berikut.

```
1  /* Program Baca Tulis */
2  #include <stdio.h>
3  int main(){
4      int bilangan;
5      scanf("%d", &bilangan);
6      printf("Bilangan yang dibaca bernilai: ");
7      printf("%d.\n", bilangan);
8      return 0;
9  }
```

Perhatikan pada baris ke lima, program memanggil fungsi *scanf* yang digunakan untuk membaca masukan dari pengguna (*user*). Cara kerja dari fungsi *scanf* adalah Ketika saat baris ini dieksekusi, program akan berhenti hingga pengguna memasukkan suatu bilangan dan menekan tombol *enter*.

scanf("%d", &bilangan);

Fungsi fungsi *scanf* menerima dua buah parameter, yaitu **%d** yang merupakan spesifikasi format (*format specifier*) dan **&bilangan** yang merupakan variabel untuk menampung nilai yang masukan. Maksudnya adalah Ketika kita pengguna program berhenti dan memasukan angka bilangan yang bertipe integer, kemudian ditekan tombol enter maka nilai masukan yang pengguna masukan akan disimpan pada variable yang namanya bilangan, contoh angka yang menjadi masukan adalah 10, maka variable bilangan akan sama dengan 10. Di depan bilangan, terdapat tanda *ampersand* (&) yang wajib digunakan untuk melakukan pembacaan atau disebut pointer untuk menunjukan posisi dari suatu variable yang digunakan.

Setiap tipe data memiliki spesifikasi format yang dapat digunakan untuk menjelaskan jenis data kepada program. Ingat, program membaca dan menyimpan semua data sebagai bilangan biner. Spesifikasi format yang digunakan untuk tipe data pada bahasa C dapat dilihat pada table berikut:

Tabel Spesifikasi Format pada Bahasa C

Tipe Data	Spesifikasi Format (Signed)	Keterangan
int	%d	untuk menampilkan nilai angka atau bilangan decimal.
short	%hd	untuk menampilkan nilai angka atau bilangan decimal dengan nilai yang lebih kecil dari int.
long	%ld	untuk menampilkan nilai angka atau bilangan decimal dengan nilai lebih besar dari int.
float	%f	untuk menampilkan nilai pecahan.
double	%lf	untuk menampilkan nilai pecahan dengan nilai lebih besar dari float.
char	%c	Untuk menampilkan karakter

Perhatikan pada baris keenam dan ketujuh, menggunakan fungsi printf, yang merupakan fungsi untuk menulis. Pada baris keenam adalah perintah program untuk mencetak suatu kalimat yang diapit dengan tanda petik ganda yaitu *"Bilangan yang dibaca bernilai:"*. Pada baris ketujuh adalah perintah program untuk mencetak nilai dari variabel bilangan. Mirip seperti fungsi scanf, menggunakan spesifikasi format dan variabel. Bedanya, di depan variabel, tidak perlu ada tanda *ampersand*.

Tanda \n merupakan *escape sequence* yang digunakan untuk membuat garis baru, seperti Ketika kita menekan tombol *enter* pada aplikasi pengolah kata. Berikut adalah perintah escape sequence yang dapat digunakan:

```
\n (newline)
\t (tab)
\v (vertical tab)
\f (new page)
\b (backspace)
\r (carriage return)
\n (newline)
\a (beep, bell)
\\ (garis miring)
\" (tanda kutip ganda)
\` (tanda kutip)
```

Contoh Latihan 3: Input dan Output Program Bahasa C

Deskripsi Soal:

Membuat program untuk membaca nilai dari variable yang diinputkan, kemudian menampilkan (output) dari nilai bilangan yang diinputkan. Langkah-langkah yang harus dilakukan sebagai berikut:

Petunjuk:

- 1. Siapkan kertas dan alat tulis sebagai media simulasi.
- 2. Gunakan satu lembar kertas sebagai lembar kerja program
- 3. Kerjakan secara mandiri atau kelompok.
- 4. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: *latihan_input_output*

Langkah-langkah:

- 1. Buka Lembar Kerja Program
- 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

<i>Problem</i>	4. Membaca Nilai dari Variabel 5. Menampilkan nilai variable yang telah diinputkan
<i>Solution</i>	1. Menggunakan perintah <i>scanf</i> 2. Menggunakan perintah <i>printf</i>

3. Menulis program

```
1  /* Program Latihan Input Output */
2  #include <stdio.h>
3  int main(){
4      int bilangan;
5      scanf("%d", &bilangan);
6      printf("Bilangan yang dibaca bernilai: ");
7      printf("%d.\n", bilangan);
8      return 0;
9  }
```

- 4. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri.

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama, dan akan berhenti di baris kelima (karena ada perintah *scanf* yang membutuhkan inputan), sampai pengguna/user menginputkan nilai dari bilangan. Misalkan masukan angka

5 (setelah menginputkan angka 5, user harus menekan tombol *enter*)

Maka akan berpindah pada baris keenam, Ketika baris keenam di eksekusi, akan menghasilkan/output

Bilangan yang dibaca bernilai:

Setelah menampilkan hasil pada bari keenam, maka akan berpindah pada baris ketujuh, Ketika dieksekusi akan menghasilkan/output: 5

Jadi Hasil dari kode program tersebut adalah:

Bilangan yang dibaca bernilai: 5

AP-06

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	3 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa memahami konsep <i>variable</i> dan ekspresi dan mengaplikasikannya dalam bentuk program		
Kata kunci	<i>Variable</i> , Ekspresi, <i>Algoritma</i> , Diagram Alir, Pemrograman, Bahasa C		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana penggunaan variabel, eksperesi dalam Bahasa C dan menulis kode program dengan menggunakan IDE Bahasa C, guru akan memberikan contoh penggunaan input dan ouput dalam bahawa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	Materi Ajar: Konsp Variable dan Ekspresi Alat: Gunting Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas, lem, karton		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-6: Ekspresi (3 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Membuat Program dengan Bahasa C	Siswa memahami konsep variable dan eskpresi dan mengaplikasikannya dalam bentuk program	3 JP	Latihan Mandiri Lembar Kerja Keaktifan

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik mengeksplorasi perintah ekspresi dalam Bahasa C, secara mandiri maupun kolaborasi.	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat program Bahasa C secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Apa yang kalian ketahui tentang ekspresi ?
- 2. Apakah kalian pernah melakukan operasi matematika seperti penjumlahan, pengurangan, pembagian, perkalian ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;

- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru menyampaikan materi dan dapat memberikan penjelasan langsung mengenai program yang akan dibuat pada latihan ini.
- 2. Guru dapat menggunakan media lembar screenshot IDE Kembali atau dapat menganggap bahwa papan tulis itu sebagai media lembar kerja program.
- 3. Guru memberi **Contoh Latihan 4, Latihan Ekspresi menghitung luas sebidang tanah** dan menjelaskan ilustrasi kode program.
- 4. Guru memancing siswa untuk menulis ulang kode program di lembar kerja program yang telah disediakan (dapat menggunakan media papan tulis)
- 5. Pancing siswa lain untuk melakukan debug dan run dengan menjelaskan ilustrasi setiap baris program.
- 6. Guru membentuk kelompok yang terdiri dari 2-5 orang:

6. Bertugas Membuat Diagram Alir
7. Bertugas Membuat Deskripsi Algoritma, Pseudocode
8. Bertugas Membuat/Menulis Kode Program
9. Bertugas sebagai Debug dan Run
10. Bertugas Membuat Skenario Input dan ilustrasi setiap baris program

- 7. Guru mengarahkan siswa/Kelompok ke aktivitas Ayo, Kita Berlatih **Latihan Ekspresi**. Biarkanlah siswa mengerjakan latihan mandiri. Guru memberikan umpan balik dan *scaffolding* pada siswa yang mengalami kesulitan.
- 8. Mintalah kelompok lain untuk menguji program rekan mereka dengan menggunakan kasus uji yang diberikan, atau dibuat sendiri oleh siswa. Setelah itu, pancing siswa untuk memperbaiki program apabila terdapat kesalahan pada program.
- 9. Guru dapat membahas aktivitas dengan meminta beberapa siswa menuliskan deskripsi dan menjelaskan hasil pekerjaannya.
- 10. Guru dapat mengulas kembali beberapa kesalahan umum yang terjadi pada saat latihan menulis program. Setelah itu, siswa dapat diminta untuk memperbaiki program di sisa waktu, atau dikerjakan kembali ke rumah.
- 11. Guru menginformasikan hasil pengerjaan Latihan ekspresi dapat dikumpulkan untuk dinilai.

Penutup

- 1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
- 2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
- 3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang ekspresi dalam Bahasa C?	
Apa yang kalian rasakan ketika mencoba membuat program ekspresi Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara <i>unplugged</i> ?	

G. Lembar Kerja Peserta Didik

Aktivitas Ayo, Kita Berlatih Latihan Ekspresi.

Petunjuk:

1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
2. Siswa mengerjakan soal secara berkelompok yang terdiri dari 2-5 orang.

- | |
|---|
| 1. Bertugas Membuat Diagram Alir |
| 2. Bertugas Membuat Deskripsi Algoritma, Pseudocode |
| 3. Bertugas Membuat/Menulis Kode Program |
| 4. Bertugas sebagai Debug dan Run |
| 5. Bertugas Membuat Skenario Input dan Ilustrasi setiap baris program |

3. Siswa Mengerjakan Soal Latihan berikut:

Soal:

1) Deskripsi Soal:

Pak Hamim memiliki sebidang tanah berbentuk segitiga. Ia ingin mengetahui berapa luas tanah yang ia miliki. Bantulah Pak Hamim menghitung dengan membuat program untuk menghitung luas tanahnya.

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buat Diagram Alir, Deskripsi Algoritma, dan Pseudocode.
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

```
// Program Menghitung Luas sebidang tanah berbentuk segitiga
/*
Nama Kelompok:
Anggota Kelompok:
1.
2.
3.
4.
5.
*/
#include <stdio.h>
int main()
{

return 0;
}
```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Skenario Input:
 - Skenario Pertama:** Alas dan tingginya adalah 25 dan 6
 - Skenario Kedua:** Alas dan tingginya adalah 30 dan 5
- Buat Ilustrasi Setiap Baris Program

2) Deskripsi Soal:

Pak Asep memiliki sebidang tanah berbentuk persegi. Ia ingin mengetahui berapa luas tanah yang ia miliki. Bantulah Pak Asep menghitung dengan membuat program untuk menghitung luas tanahnya.

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buat Diagram Alir, Deskripsi Algoritma, dan *Pseudocode*.
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

```
// Program Menghitung Luas persegi
/*
  Nama Kelompok:
  Anggota Kelompok:
  1.
  2.
  3.
  4.
  5.
*/
#include <stdio.h>
int main()
{
    return 0;
}
```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Skenario Input:
Skenario Pertama: Alas dan tingginya adalah 25 dan 25
Skenario Kedua: Alas dan tingginya adalah 30 dan 30
- Buat Ilustrasi Setiap Baris Program

3) Deskripsi Soal:

Roni memiliki kelereng 100 dan ingin membagi ke beberapa orang temannya. Setiap teman mendapat berapa kelereng jika roni memiliki 5 orang teman. Buatlah program untuk menyelesaikan soal tersebut.

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buat Diagram Alir, Deskripsi Algoritma, dan *Pseudocode*.
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.


```
// Program Menghitung pembagian
/*
Nama Kelompok:
Anggota Kelompok:
1.
2.
3.
4.
5.
*/
#include <stdio.h>
int main()
{

return 0;
}
```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Skenario Input:
 - Skenario Pertama:** sudah jelas (100 kelereng dan 5 orang)
 - Skenario Kedua:** jika roni ingin membagi ke 2 orang.
- Buat Ilustrasi Setiap Baris Program

4) Deskripsi Soal:

Diketahui Kode Program berikut ini:

```
// Program Ekspresi Benar (True) atau Salah (False)
#include int main()
{
int bilanganA, bilanganB;

scanf("%d", &bilanganA);
scanf("%d", &bilanganB);

hasil1 = bilanganA==bilanganB;
hasil2 = bilanganA>=bilanganB;
hasil3 = bilanganA!=bilanganB;
hasil4 = bilanganA<bilanganB;

print("%d \n", hasil1);
print("%d \n", hasil2);
print("%d \n", hasil3);
print("%d \n", hasil4);

return 0;
}
```

Petunjuk:

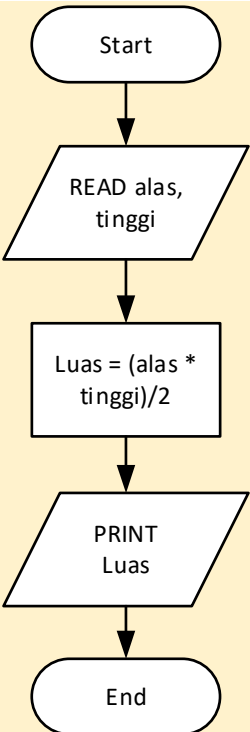
- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?)
- Lakukan Pengecekan Program (Debug dan Run)
- Buat Skenario Input:

Skenario Pertama: bilanganA=7 dan bilanganB = 9
Skenario Kedua: bilanganA=11 dan bilanganB = 11

- Tuliskan Hasil/output dari setiap skenario.

Jawaban:

- Problem: Mengetahui luas sebidang tanah berbentuk persegi
Solution: menggunakan rumus luas segitiga = (panjang * tinggi)/2

Diagram Alir	Deskripsi Algoritma	Pseudocode
	<ol style="list-style-type: none">Mulai programBaca Nilai alas dan tinggiHitung Luas segitigaCetak Hasil Luas	<p>Algoritma LuasSegitiga</p> <p>Read alas, tinggi</p> <p>Luas = (alas * tinggi)/2</p> <p>Print Luas</p>

Kode Program:

```
1 // Program Menghitung Luas segitiga
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14     float Luas, alas, tinggi;
15 }
```

```

16 scanf("%f", &alas);
17 scanf("%f", &tinggi);
18
19 Luas = (alas * tinggi)/2;
20 printf("Luas: %f", Luas);
21
22 return 0;
23 }

```

Jawab Skenario Pertama: Alas dan tingginya adalah 25 dan 6

Input	Nilai masukan: alas=25, tinggi=6 Nilai Variable: alas=25, tinggi=6
Proses	Luas = (alas * tinggi)/2 Luas = (25*6)/2 Luas = 75
Output	75

Jawab Skenario Kedua: Alas dan tingginya adalah 30 dan 5

Input	Nilai masukan: alas=30, tinggi=5 Nilai Variable: alas=30, tinggi=5
Proses	Luas = (alas * tinggi)/2 Luas = (30*5)/2 Luas = 75
Output	75

Ilustrasi Setiap Baris Program:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel luas, alas, tinggi
15	
16	Layer akan menampilkan layer: (pengguna/user harus memasukan nilai alasnya) Misalkan yang dimasukan angka 6
17	Layer akan menampilkan: (pengguna/user harus memasukan nilai tingginya) Misalkan yang dimasukan angka 4

18	
19	Melakukan proses hitng luas sesuai dengan rumus yang dituliskan luas = (alas * tinggi) / 2; luas = (6 * 4) / 2; luas = (24) / 2 ; luas = 12
20	Menampilkan: Luas: 12
21	
22, 23	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

2. *Problem:* Bagaimana Mengitung Luas Persegi
- Solution:* menggunakan Luas persegi yaitu Luas = sisi * sisi

Diagram Alir	Deskripsi Algoritma	Pseudecode
<pre>graph TD; Start([Start]) --> Read[/READ sisi/]; Read --> Process[Luas = s * s]; Process --> Print[/PRINT Luas/]; Print --> End([End]);</pre>	<div>1. Mulai program</div> <div>2. Baca Nilai sisi</div> <div>3. Hitung Luas persegi</div> <div>4. Cetak Hasil Luas</div>	<div>Algoritma LuasSegitiga</div> <div>Read sisi</div> <div>Luas = sisi * sisi</div> <div>Print Luas</div>

Kode Program:

1	// Program Menghitung persegi
2	/*
3	Nama Kelompok:
4	Anggota Kelompok:
5	1.
6	2.
7	3.
8	4.
9	5.

```
10  */
11  #include <stdio.h>
12  int main()
13  {
14      int Luas, sisi;
15
16      scanf("%d", &sisi);
17      Luas = sisi * sisi;
18
19      printf("Luas: %d", Luas);
20
21      return 0;
22  }
```

Jawab Skenario Pertama:

Input	Nilai masukan: 25 Nilai Variable: sisi = 25
Proses	Luas = sisi * sisi Luas = 25 * 25 Luas = 625
Output	625

Jawab Skenario Kedua:

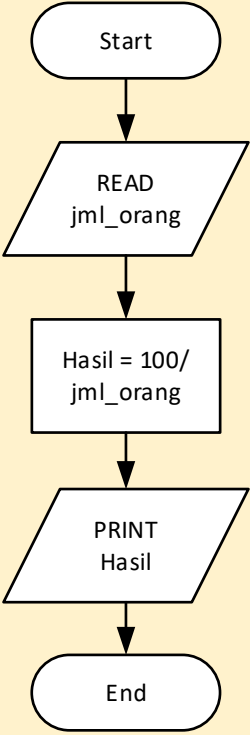
Input	Nilai masukan: 30 Nilai Variable: sisi = 30
Proses	Luas = sisi * sisi Luas = 30 * 30
Output	900

Ilustrasi Setiap Baris Program:

Ilustrasi dalam menjalankan program , Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini	
Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel luas, sisi
15	
16	Layer akan menampilkan layer: (pengguna/user harus memasukan nilai sisinya) Misalkan yang dimasukan angka 5

17	Melakukan proses hitng luas sesuai dengan rumus yang dituliskan luas = sisi * sisi; luas =5 * 5; luas = 25
18 19	Menampilkan: Luas: 25
20 21, 21	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

3. *Problem:* Membagi kelereng 100 ke beberapa orang
Solution: menggunakan rumus pembagian = 100 / jml orang

Diagram Alir	Deskripsi Algoritma	Pseudocode
	<ol style="list-style-type: none">1. Mulai program2. Baca jml_orang3. Hitung Pembagian: 100/jml_orang4. Cetak Hasil Pembagian	<p>Algoritma Pembagian</p> <p>Read jml_orang</p> <p>Hasil = 100 / jml_orang</p> <p>Print Hasil</p>

Kode Program:

```
1 // Program Menghitung Pembagian
2 /*
3 Nama Kelompok:
4 Anggota Kelompok:
5 1.
6 2.
7 3.
8 4.
```

```
9      5.  
10     */  
11     #include <stdio.h>  
12     int main()  
13     {  
14         int Hasil, jml_orang;  
15  
16         scanf("%d", &jml_orang);  
17         Hasil = 100/jml_orang;  
18  
19         printf("Hasil: %d", Hasil);  
20  
21         return 0;  
22     }
```

Jawab Skenario Pertama: dibagi ke 5 orang

Input	Nilai masukan: 5 Nilai Variable: jml_orang = 5
Proses	Hasil = 100/jml_orang Hasil = 100/5 Hasil = 20
Output	20

Jawab Skenario Kedua: dibagi ke 2 orang

Input	Nilai masukan: 2 Nilai Variable: jml_orang = 2
Proses	Hasil = 100/jml_orang Hasil = 100/2 Hasil = 50
Output	50

Ilustrasi Setiap Baris Program:

Ilustrasi dalam menjalankan program , Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini	
Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel jml_orang
15	
16	Layer akan menampilkan layer:

	(pengguna/user harus memasukan nilai jml_orang) Misalkan yang dimasukan angka 5
17	Melakukan proses hitng pembagian dengan rumus yang dituliskan Hasil = 100/jml_orang; Hasil =100/5; Hasil = 20
18	
19	Menampilkan: Hasil: 20
20	
21, 21	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

4. *Problem:* Mengetahui hasil kode program
Solution: melakukan penelusuran setiap baris program

Jawab Skenario Pertama: bilanganA=7 dan bilanganB = 9

Input	Nilai masukan: 7 dan 9 Nilai Variable: bilanganA=7, bilanganB=9
Proses	hasil1 = bilanganA==bilanganB hasil1= 7==9 hasil1= false hasil2 = bilanganA>=bilanganB hasil1= 7>=9 hasil1= false hasil3 = bilanganA!=bilanganB hasil1= 7!=9 hasil1= true hasil4 = bilanganA<bilanganB hasil1= 7<9 hasil1= true
Output	false false true true

Jawab Skenario Kedua: bilanganA=11 dan bilanganB = 11

Input	Nilai masukan: 11 dan 11 Nilai Variable: bilanganA=11, bilanganB=11
Proses	hasil1 = bilanganA==bilanganB hasil1= 11==11 hasil1= true

	<pre>hasil2 = bilanganA>=bilanganB hasil1= 11>=11 hasil1= true hasil3 = bilanganA!=bilanganB hasil1= 11!=11 hasil1= false hasil4 = bilanganA<bilanganB hasil1= 11<11 hasil1= false</pre>
Output	<pre>true true false false</pre>

H. Bahan Bacaan

4. Ekspresi

Ekspresi merupakan bagian yang tidak terpisahkan dari program. Di dalam matematika, ekspresi terdiri atas kombinasi beberapa operand dan operator yang memiliki makna. Kalian telah terbiasa menulis ekspresi pada matematika,

Contoh Latihan 4: Latihan Ekspresi

Deskripsi Soal:

Membuat program untuk menghitung Luas sebidang tanah yang berbentuk segitiga. Langkah-langkah yang harus dilakukan sebagai berikut:

Petunjuk:

- 1. Siapkan kertas dan alat tulis sebagai media simulasi.
- 2. Gunakan satu lembar kertas sebagai lembar kerja program
- 3. Kerjakan secara mandiri.
- 4. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: *ekspresi1*

Langkah-langkah:

- 1. Buka Lembar Kerja Program
- 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

Problem	Mencari luas sebidang tanah yang berbentuk segitiga
Solution	Menggunakan Rumus Luas segitiga Luas = (Alas * Tinggi) / 2 1. Membuat variable Luas, Alas, Tinggi 2. Untuk Alas dan tinggi diinputkan secara bebas berarti menggunakan fungsi scanf

1. Menulis program

1	<pre>/* Program Latihan Penggunaan Ekspresi */</pre>
---	--

```
2  #include <stdio.h>
3  int main(){
4      float luas, alas, tinggi;
5      scanf("masukan nilai alas: %d", &alas);
6      scanf("masukan nilai tinggi: %d", &tinggi);
7      luas = (alas * tinggi) / 2;
8      printf("Jadi Luas sebidang tanah tersebut adalah: ");
9      printf("%d.\n", luas);
10     return 0;
11 }
```

2. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri, pada proses ini menggunakan pendekatan ilustrasi setiap baris, agar siswa lebih memahami.

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. Perhatikan ilustrasi berikut

Baris Kode Program	Keterangan eksekusi
1	Tidak akan dijalankan karena berisi komentar
2	Memanggil fungsi stdio.h
3	Membaca badan program
4	Membentuk variabel luas, alas, tinggi
5	Layer akan menampilkan: Masukan nilai alas: (pengguna/user harus memasukan nilai alasnya) Misalkan yang dimasukan angka 6
6	Layer akan menampilkan: Masukan nilai tinggi: (pengguna/user harus memasukan nilai tingginya) Misalkan yang dimasukan angka 4
7	Melakukan proses hitng luas sesuai dengan rumus yang dituliskan luas = (alas * tinggi) / 2; luas = (6 * 4) / 2; luas = (24) / 2 ; luas = 12
8	Menampilkan kata: Jadi Luas sebidang tanah tersebut adalah:

9	Menampilkan angka luas, yang diletakan setelah “ Jadi Luas sebidang tanah tersebut adalah: ”, dengan tampilan dilayar akan seperti berikut: Jadi Luas sebidang tanah tersebut adalah: 12
10, 11	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

AP-07

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	3 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	Cooperative Learning
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa memahami konsep struktur kontrol keputusan dan mengaplikasikannya dalam bentuk program		
Kata kunci	Keputusan, Algoritma, Diagram Alir, Pemrograman, Bahasa C		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana penggunaan struktur kontrol keputusan dalam Bahasa C dan menulis kode program dengan menggunakan IDE Bahasa C, guru akan memberikan contoh penggunaan struktur kontrol keputusan dalam bahasa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	Materi Ajar: Konsep Struktur kontrol keputusan Alat: Gunting		
	Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas, lem, karton		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-7: Struktur Kontrol Keputusan (3 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Membuat Program dengan Bahasa C	Siswa memahami konsep dan Struktur kontrol keputusan dan mengaplikasikannya dalam bentuk program	3 JP	Latihan Mandiri Lembar Kerja Keaktifan

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik mengeksplorasi perintah kontrol keputusan dalam Bahasa C, secara mandiri maupun kolaborasi.	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat program Bahasa C secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Pernahkan kalian mengambil suatu keputusan ?
- 2. Apa yang kalian ketahui tentang keputusan ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

1. Guru menyampaikan materi dan dapat memberikan penjelasan langsung mengenai program yang akan dibuat pada latihan ini.
2. Guru dapat menggunakan media lembar screenshot IDE Kembali atau dapat mengangap bahwa papan tulis itu sebagai media lembar kerja program.
3. Guru membuat simulasi kontrol keputusan:

Simulasi 1:

- Jika guru menyebutkan "if nomor absen ganjil", maka siswa yang berabsen ganjil akan "mengacungkan tangan".
- Jika guru menyebutkan "if nomor absen genap", maka siswa yang berabsen genap meneriakkan kata "genap"

Simulasi 2:

- Jika guru menyebutkan "if nomor absen ganjil", maka siswa yang berabsen ganjil akan "mengacungkan tangan".
- Jika guru menyebutkan "else ", maka siswa yang berabsen genap meneriakkan kata "genap"

Simulasi 3:

- Lakukan secara bergantian pengucapan ganap dan ganjil.
- Atur kecepatan durasi penyebutan.

4. Guru memberi **Contoh Latihan Struktur Kontrol Keputusan if-else** (contoh latihan 5), **switch-case** (contoh latihan 6), **bersarang** (contoh latihan 7) dan menjelaskan ilustrasi kode program.
5. Guru memancing siswa untuk menulis ulang kode program di lembar kerja program yang telah disediakan (dapat menggunakan media papan tulis)
6. Pancing siswa lain untuk melakukan debug dan run dengan menjelaskan ilustrasi setiap baris program.
7. Guru membentuk kelompok yang terdiri dari 2-5 orang:

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan ilustrasi setiap baris program

8. Guru mengarahkan siswa/Kelompok ke aktivitas Ayo, Kita Berlatih **Latihan Struktur Kontrol Keputusan**. Biarkanlah siswa mengerjakan latihan mandiri. Guru memberikan umpan balik dan *scaffolding* pada siswa yang mengalami kesulitan.

9. Mintalah kelompok lain untuk menguji program rekan mereka dengan menggunakan kasus uji yang diberikan, atau dibuat sendiri oleh siswa. Setelah itu, pancing siswa untuk memperbaiki program apabila terdapat kesalahan pada program.
10. Guru dapat membahas aktivitas dengan meminta beberapa siswa menuliskan deskripsi dan menjelaskan hasil pekerjaannya.
11. Guru dapat mengulas kembali beberapa kesalahan umum yang terjadi pada saat latihan menulis program. Setelah itu, siswa dapat diminta untuk memperbaiki program di sisa waktu, atau dikerjakan kembali ke rumah.
12. Guru menginformasikan hasil pengerjaan Latihan struktur kontrol keputusan dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang strukur kontrol keputusan dalam Bahasa C?	
Apa yang kalian rasakan ketika mencoba membuat program strukur kontrol keputusan Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara unplugged?	

G. Lembar Kerja Peserta Didik

Aktivitas Ayo, Kita Berlatih Latihan Struktur Kontrol Keputusan.

Petunjuk:

- 1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
- 2. Siswa mengerjakan soal secara berkelompok yang terdiri dari 2-5 orang.

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan Ilustrasi setiap baris program

- 3. Siswa Mengerjakan Soal Latihan berikut:

Soal:

1) Deskripsi Soal:

Buatlah program untuk mengecek apakah sebuah bilangan bulat adalah bilangan bulat positif.

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buat Diagram Alir, Deskripsi Algoritma, dan *Pseudocode*.
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

```
// Program Pengecekan bilangan bulat positif
/*
  Nama Kelompok:
  Anggota Kelompok:
  1.
  2.
  3.
  4.
  5.
*/
#include <stdio.h>
int main()
{
    return 0;
}
```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Skenario Input:

Jika diketahui sebuah bilangan bulat adalah n, dimana n berada pada rentang $-25 < n < 50$

Skenario Input	Output
Bilangan yang diinputkan -20	"Bukan bilangan bulat positif"
Bilangan yang diinputkan 35	"Bilangan bulat positif"
Bilangan yang diinputkan 100	

- Buat Ilustrasi Setiap Baris Program

2) Deskripsi Soal:

Susunlah kode program berikut sehingga membentuk kode program yang benar dan tidak terjadi error.

if (a==b) {

1

#include <stdio.h>

2

if (b>c) {

3

d = b + c;

4

}else{

5

}

6

}

7

printf("%d", d);

8

int main() {

9

}

10

printf(" a tidak sama dengan b");

11

return 0;

12

int a=5, b=5, c=4, d;

13

Petunjuk:

- Siapkan kertas, gunting, lem dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Gunakan satu lembar kertas/karton sebagai lembar kerja program untuk Menyusun kode program yang benar.
- Jika sudah tersusun buatlah Diagram Alir, Deskripsi Algoritma, dan *Pseudocode*.
- Lakukan Pengecekan Program (Debug dan Run)
- Buat Ilustrasi Setiap Baris Program

Jawaban:

1. *Problem*: cek bilangan bulat positif
Solution: menggunakan strukur kontrol keputusan

Diagram Alir	Deskripsi Algoritma	Pseudocode
<pre>graph TD; Start([Start]) --> Read[/READ Bilangan/]; Read --> Decision{Bilangan > 0}; Decision -- Ya --> PrintYes[/PRINT "Bilangan bulat positif"/]; PrintYes --> End([End]); Decision -- Tidak --> PrintNo[/PRINT "Bukan bilangan bulat positif"/]; PrintNo --> End;</pre>	<ol style="list-style-type: none">1. Mulai program2. Baca Nilai bilangan3. Cek kondisi nilai bilangan, apakah bilangan > 0, jika ya maka, prosesnya:<ul style="list-style-type: none">- Cetak kata “Bilangan bulat positif”4. Jika nilai bilangan< 0 maka, prosesnya:<ul style="list-style-type: none">- Cetak kata “Bukan bilangan bulat positif”	<p>Algoritma CekBilangan</p> <p>Read bilangan</p> <pre>if (bilangan > 0) then Print "Bilangan bulat positif" else Print "Bukan bilangan bulat positif" end if</pre>

Kode Program

```
1 // Program Pengecekan bilangan bulat positif
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14     int bilangan;
15
16     scanf("%d", &bilangan);
```

```
17  if(bilangan > 0){
18      printf("Bilangan bulat positif");
19  }else{
20      printf("Bukan bilangan bulat positif");
21  }
22
23  return 0;
24  }
```

Ilustrasi Setiap Baris Program:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel jml_orang
15	
16	Layer akan menampilkan layer: (pengguna/user harus memasukan nilai bilangan) Misalkan yang dimasukan angka 4
17	Melakukan pengecekan jika bilangan lebih dari nol, maka
18	Cetak kata "Bilangan bulat positif"
19	Jika bilangan tidak lebih dari nol
20	Cetak kata "Bukan bilangan bulat positif"
21	Menutup/akhir dari kondisi pengecekan
22	
23, 24	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

2. Menyusun kode program:

2	9	13	1	3	4	8	6	5	11	7	12	10
---	---	----	---	---	---	---	---	---	----	---	----	----

Problem: Menyusun kode program
Solution: harus disusun dengan benar dengan memperhatikan struktur penulisan Bahasa C

Diagram Alir	Deskripsi Algoritma	Pseudocode
<pre>graph TD; Start([Start]) --> Read[/READ a=5, b=5, c=4, d/]; Read --> AEqualsB{a=b}; AEqualsB -- Tidak --> Print1[/PRINT "a tidak sama dengan b"/]; AEqualsB -- Ya --> BGTc{b>c}; BGTc -- Ya --> Print2[/d=b+c PRINT d/]; BGTc -- Tidak --> End([End]); Print1 --> End; Print2 --> End;</pre>	<ol style="list-style-type: none">1. Mulai program2. Baca Nilai a, b, c, d3. Cek kondisi nilai apakah a = b, jika ya maka:4. Cek kondisi nilai apakah b>c, jika ya, maka:5. Hitung nilai d dengan rumus d=b+c dan Cetak nilai d6. Jika a tidak sama dengan b maka7. Cetak kata “a tidak sama dengan b”	<p>Algoritma CekBilangan</p> <p>Read bilangan</p> <pre>if (a = b) then if(b>c) then d=b+c cetak end if else PRINT “a tidak sama dengan b” end if</pre>

Ilustrasi Setiap Baris Program , jika sudah tersusun maka akan terlihat kode program seperti berikut:

```
1  #include <stdio.h>
2  int main(){
3      int a=5, b=5, c=4, d;
4      if(a==b){
5          if(b>c){
6              d=b+c;
7              printf("%d",d);
8          }
9      }else{
10         printf("a tidak sama dengan b");
11     }
12     return 0;
13 }
```

Ilustrasi Setiap Baris Program:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1	Memanggil fungsi stdio.h
2	Membaca badan program
3	Membentuk variabel a,b,c,d,
4	Cek kondisi nilai apakah a = b, jika ya
5	Cek kondisi nilai apakah b>c
6	Jika terpenuhi kondisi b>c maka proses penjumlahan d=b+c dijalankan
7	Mencetak nilai d
8	Menutup kondisi b>c
9	Jika nilai a selain b maka jalankan
10	Cetak kata "a tidak sama dengan b"
11	Akhir dari kondisi a=b
12, 13	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

5. Kontrol Keputusan

Komputer merupakan alat yang membantu banyak aktivitas manusia. Pada dasarnya, komputer menjalankan perintah dari manusia. Perintah-perintah tersebut dituangkan secara tertulis dalam sebuah aturan tertentu yang disebut sebagai kode program yang bertujuan untuk mengatur bagaimana komputer harus bertindak untuk menyelesaikan sebuah permasalahan tertentu. Hal ini termasuk juga dalam proses pengambilan keputusan, seperti halnya dalam contoh pemilihan rute (**ke-kiri** atau **ke-kanan**), atau pemilihan mode transportasi (**naik kendaraan bermotor** atau **naik sepeda**).

Pada bagian ini kita akan mempelajari bagaimana pengambilan keputusan dilakukan dalam sebuah program. Istilah yang sering digunakan untuk ini adalah kondisional. Apa itu kondisional? Secara sederhana, kondisional adalah sebuah bentuk pernyataan “jika ..., maka ...”. Pada bahasa pemrograman C, ada beberapa jenis pernyataan kondisional sebagai berikut:

- 1) Struktur Kontrol Keputusan If – Else
- 2) Struktur Kontrol Keputusan Switch-Case
- 3) Struktur Kontrol Keputusan Bersarang

Contoh Latihan 5: Latihan Struktur Kontrol Keputusan If - Else

Deskripsi Soal:

Membuat program untuk membandingkan bilangan a dan bilangan b. Langkah-langkah yang harus dilakukan sebagai berikut:

Petunjuk:

- 1. Siapkan kertas dan alat tulis sebagai media simulasi.
- 2. Gunakan satu lembar kertas sebagai lembar kerja program
- 3. Kerjakan secara mandiri.
- 4. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: kontrol_keputusan1

Langkah-langkah:

- 1. Buka Lembar Kerja Program
- 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

<i>Problem</i>	Bagaimana membandingkan bilangan a dan bilangan b
<i>Solution</i>	Menggunakan operator ==

3. Menulis program

```
1  * Program Membandingkan Bilangan */
2  #include <stdio.h>
3  int main(){
4      int a = 4, b = 4;
```

```
5    if (a == b) {
6        printf("a sama dengan b \n");
7    }
8    return 0;
9 }
```

4. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri, pada proses ini menggunakan pendekatan ilustrasi setiap baris, agar siswa lebih memahami.

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. Perhatikan ilustrasi berikut

Baris Kode Program	Keterangan eksekusi
1	Tidak akan dijalankan karena berisi komentar
2	Memanggil fungsi stdio.h
3	Membaca badan program
4	Membaca variabel a=4 dan b=4
5, 6, 7	Cek kondisi Jika a sama dengan b maka Cetak pada layer: a sama dengan b
8, 9	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

5. Ubahlah nilai variabel a dan b dan gunakan operator "!=", ">", "<", ">=" dan buatlah ilustrasi kode program.

Contoh Latihan 6: Latihan Struktur Kontrol Keputusan Switch-Case

Deskripsi Soal:
Membuat program untuk menampilkan nama bulan. Langkah-langkah yang harus dilakukan sebagai berikut:

- Petunjuk:**
- 1. Gunakan Switch-case untuk menyelesaikan program ini, karena memiliki banyak cabang.
 - 2. Siapkan kertas dan alat tulis sebagai media simulasi.
 - 3. Gunakan satu lembar kertas sebagai lembar kerja program
 - 4. Kerjakan secara mandiri.
 - 5. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: kontrol_keputusan2

- Langkah-langkah:**
- 1. Buka Lembar Kerja Program
 - 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

<i>Problem</i>	Bagaimana menampilkan nama bulan dari angka yang diinputkan
<i>Solution</i>	<ul style="list-style-type: none">- Struktur kontrol keputusan menggunakan switch-case- Untuk menampilkan nama bulan menggunakan perintah printf

3. Menulis program

```
1  /* Program Menampilkan nama bulan */
2  #include <stdio.h>
3  int main(){
4      int bulan;
5      scanf("%d", &bulan);
6      switch(bulan){
7          case 1:
8              printf("Januari");
9              break;
10         case 2:
11             printf("Februari");
12             break;
13         case 3:
14             printf("Maret");
15             break;
16         case 4:
17             printf("April");
18             break;
19         case 5:
20             printf("Mei");
21             break;
22         case 6:
23             printf("Juni");
24             break;
25         case 7:
26             printf("Juli");
27             break;
28         case 8:
29             printf("Agustus");
30             break;
31         case 9:
32             printf("September");
33             break;
34         case 10:
35             printf("Oktober");
36             break;
37         case 11:
38             printf("November");
39             break;
40         case 12:
41             printf("Desember");
42             break;
43         default:
44             printf("Bulan yang dimasukan, tidak sesuai");
45             break;
46     }
47     return 0;
```

4. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri, pada proses ini menggunakan pendekatan ilustrasi setiap baris, agar siswa lebih memahami.

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. Perhatikan ilustrasi berikut

Baris Kode Program	Keterangan eksekusi
1	Tidak akan dijalankan karena berisi komentar
2	Memanggil fungsi stdio.h
3	Membaca badan program
4	Membaca variabel bulan
5	Akan berhenti sampai pengguna memasukan angka bulan
6	Cek kondisi angka bulan yang dimasukan
7 - 45	<div>Merupakan kondisi yang sesuai dengan angka bulan yang dimasukan, misalkan angka bulan yang dimasukan 3, maka program akan masuk ke case 3, kode program yang ada di case 3 akan dijalankan sehingga tampilan program akan menghasilkan:</div> <div>Maret</div>
47, 48	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

Contoh Latihan 7: Latihan Struktur Kontrol Keputusan Bersarang

Deskripsi Soal:
Membuat program untuk mengetahui bilangan yang diinputkan adalah lebih besar dari 1 dan lebih kecil dari 15, jika bilangan yang dimasukan adalah rentang tersebut tampilkan apakah bilangannya ganjil atau genap.

- Petunjuk:**
- 1. Gunakan if-else bersarang untuk menyelesaikan program ini, karena memiliki banyak cabang.
 - 2. Siapkan kertas dan alat tulis sebagai media simulasi.
 - 3. Gunakan satu lembar kertas sebagai lembar kerja program
 - 4. Kerjakan secara mandiri.
 - 5. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: kontrol_keputusan3

- Langkah-langkah:**
- 1. Buka Lembar Kerja Program
 - 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

<i>Problem</i>	<ul style="list-style-type: none">- menampilkan bilangan yang diinputkan antara 1 – 15- menampilkan bilangannya apakah ganjil atau genap
<i>Solution</i>	<ul style="list-style-type: none">- Struktur kontrol keputusan menggunakan if-else bersarang

3. Menulis program

```
1  /* Program Menampilkan bilangan dan mengetahui bilangan ganjil atau genap*/
2  #include <stdio.h>
3  int main(){
4      int bilangan;
5      scanf("%d", &bilangan);
6
7      if (bilangan >= 1 && bilangan <=15){
8          printf("bilangan yang diinputkan sudah sesuai");
9          if(bilangan % 2) {
10             printf("bilangan genap");
11         }else{
12             printf("bilangan ganjil");
13         }
14     }else{
15         printf("bilangan yang diinputkan tidak sesuai");
16     }
17     return 0;
18 }
```

4. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri, pada proses ini menggunakan pendekatan ilustrasi setiap baris, agar siswa lebih memahami.

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. Perhatikan ilustrasi berikut

Baris Kode Program	Keterangan eksekusi
1	Tidak akan dijalankan karena berisi komentar
2	Memanggil fungsi stdio.h
3	Membaca badan program
4	Membaca variabel bilangan
5	Akan berhenti sampai pengguna memasukan angka bilangan
6	
7	Cek kondisi bilangan yang dimasukan Kondisi = Apakah bilangan yang dimasukan adalah rentang 1 s.d 15 ? Jika ya, maka proses selanjutnya pada baris ke-8 Jika tidak, maka proses selanjutnya menuju baris ke-14
8	Menampilkan kata bilangan yang diinputkan sudah sesuai
9	Cek kondisi apakah bilangan genap ? Jika ya, maka proses selanjutnya masuk pada baris ke-10 Jika tidak, maka proses selanjutnya masuk pada baris ke-11
10	Menampilkan kata bilangan genap

11	Kondisi jika bilangan selain genap
12	Menampilkan kata bilangan ganjil
13	Akhir dari kondisi pengecekan bilangan genap atau ganjil
14	Kondisi jika bilangan yang dimasukan diluar rentang 1 s.d 15
15	Menampilkan kata bilangan yang diinputkan tidak sesuai
16	Akhir dari kondisi pengecekan bilangan rentang 1 s.d 15
17, 18	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

AP-08

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	3 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa memahami konsep dan Struktur kontrol perulangan dan mengaplikasikannya dalam bentuk program		
Kata kunci	Perulangan, Algoritma, Diagram Alir, Pemrograman, Bahasa C		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana penggunaan struktur kontrol perulangan dalam Bahasa C dan menulis kode program dengan menggunakan IDE Bahasa C, guru akan memberikan contoh penggunaan struktur kontrol perulangan dalam bahasa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	<p>Materi Ajar: Konsep struktur kontrol perulangan</p> <p>Alat: Gunting</p> <p>Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas, lem, karton</p>		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-8: Struktur Kontrol Perulangan (3 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Membuat Program dengan Bahasa C	Siswa memahami konsep dan Struktur kontrol perulangan dan mengaplikasikannya dalam bentuk program	3 JP	Latihan Mandiri Lembar Kerja Keaktifan

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik mengeksplorasi perintah kontrol perulangan dalam Bahasa C, secara mandiri maupun kolaborasi.	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat program Bahasa C secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Apa yang kalian ketahui tentang perulangan ?
- 2. Berikan contoh perulangan dalam kehidupan sehari-hari ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru menyampaikan materi dan dapat memberikan penjelasan langsung mengenai program yang akan dibuat pada latihan ini.
- 2. Guru dapat menggunakan media lembar screenshot IDE Kembali atau dapat mengangap bahwa papan tulis itu sebagai media lembar kerja program.
- 3. Guru memberi contoh kontrol perulangan dalam kehidupan sehari-hari secara sederhana, misalnya push-up, sit-up, lari keliling sebanyak 5 kali.

Contoh	Kode program Bahasa C
Lakukan Push-up sebanyak 5 kali	<pre>for (int i=1;i<=5;i++){ printf("kerjakan push up \n"); }</pre>
	<pre>int i=1; while (i<=5){ printf("kerjakan push up \n"); i=i+1; }</pre>
	<pre>int i=0; do { i=i+1; printf("kerjakan push up \n"); }while(i<5);</pre>
Perulangan dengan kontrol keputusan, misalkan jika nomor absen genap, maka lakukan Push-up sebanyak 5 kali	<pre>int nomorAbsen; scanf("%d", &nomorAbsen); if(nomorAbsen % 2==0) { for (int i=1;i<=5;i++){ printf("kerjakan push up \n"); } }</pre>
	<pre>int nomorAbsen; scanf("%d", &nomorAbsen); if(nomorAbsen % 2==0) { int i=1; while (i<=5){ printf("kerjakan push up \n"); i=i+1; } }</pre>
	<pre>int nomorAbsen; scanf("%d", &nomorAbsen); if(nomorAbsen % 2==0) { int i=0; do { i=i+1; printf("kerjakan push up \n"); }</pre>

```
}while(i<5);  
}
```

- 4. Guru memberi **Contoh Latihan 8, Latihan Struktur Kontrol Perulangan menggunakan For, while, do while** dan menjelaskan ilustrasi kode program.
- 5. Guru memancing siswa untuk menulis ulang kode program di lembar kerja program yang telah disediakan (dapat menggunakan media papan tulis)
- 6. Pancing siswa lain untuk melakukan debug dan run dengan menjelaskan ilustrasi setiap baris program.
- 7. Guru membentuk kelompok yang terdiri dari 2-5 orang:

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan ilustrasi setiap baris program

- 8. Guru mengarahkan siswa/Kelompok ke aktivitas Ayo, Kita Berlatih **Latihan Struktur Kontrol Perulangan**. Biarkanlah siswa mengerjakan latihan mandiri. Guru memberikan umpan balik dan *scaffolding* pada siswa yang mengalami kesulitan.
- 9. Mintalah kelompok lain untuk menguji program rekan mereka dengan menggunakan kasus uji yang diberikan, atau dibuat sendiri oleh siswa. Setelah itu, pancing siswa untuk memperbaiki program apabila terdapat kesalahan pada program.
- 10. Guru dapat membahas aktivitas dengan meminta beberapa siswa menuliskan deskripsi dan menjelaskan hasil pekerjaannya.
- 11. Guru dapat mengulas kembali beberapa kesalahan umum yang terjadi pada saat latihan menulis program. Setelah itu, siswa dapat diminta untuk memperbaiki program di sisa waktu, atau dikerjakan kembali ke rumah.
- 12. Guru menginformasikan hasil pengerjaan struktur kontrol perulangan dapat dikumpulkan untuk dinilai.

Penutup

- 1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
- 2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
- 3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang strukur kontrol perulangan dalam Bahasa C?	
Apa yang kalian rasakan ketika mencoba membuat program strukur kontrol perulangan Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara <i>unplugged</i> ?	

G. Lembar Kerja Peserta Didik

Aktivitas Ayo, Kita Berlatih Latihan Struktur Kontrol Perulangan.

Petunjuk:

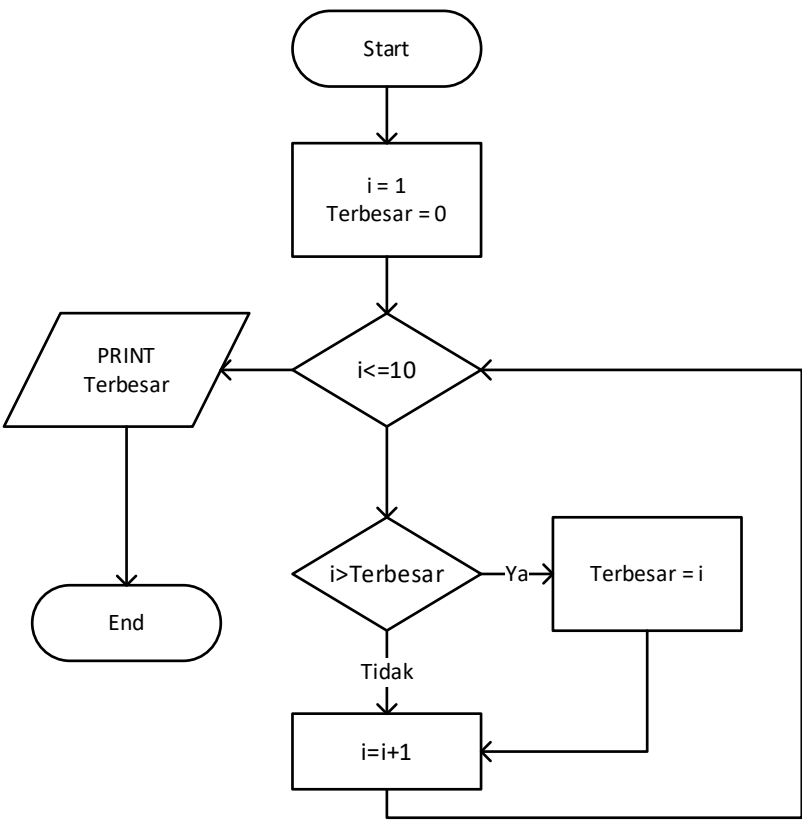
1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
2. Siswa mengerjakan soal secara berkelompok yang terdiri dari 2-5 orang.

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan Ilustrasi setiap baris program

3. Siswa Mengerjakan Soal Latihan berikut:

Soal:

- 1) **Deskripsi Soal:**
- Berikut diagram alir untuk mencetak bilangan terbesar, buatlah program Bahasa C dengan menggunakan perulangan for, while.



Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).

- Buatlah Deskripsi Algoritma dan *Pseudocode*
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

```

1 // Program mencetak nilai terbesar menggunakan for
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14
15
16
17     return 0;
18 }

```

```

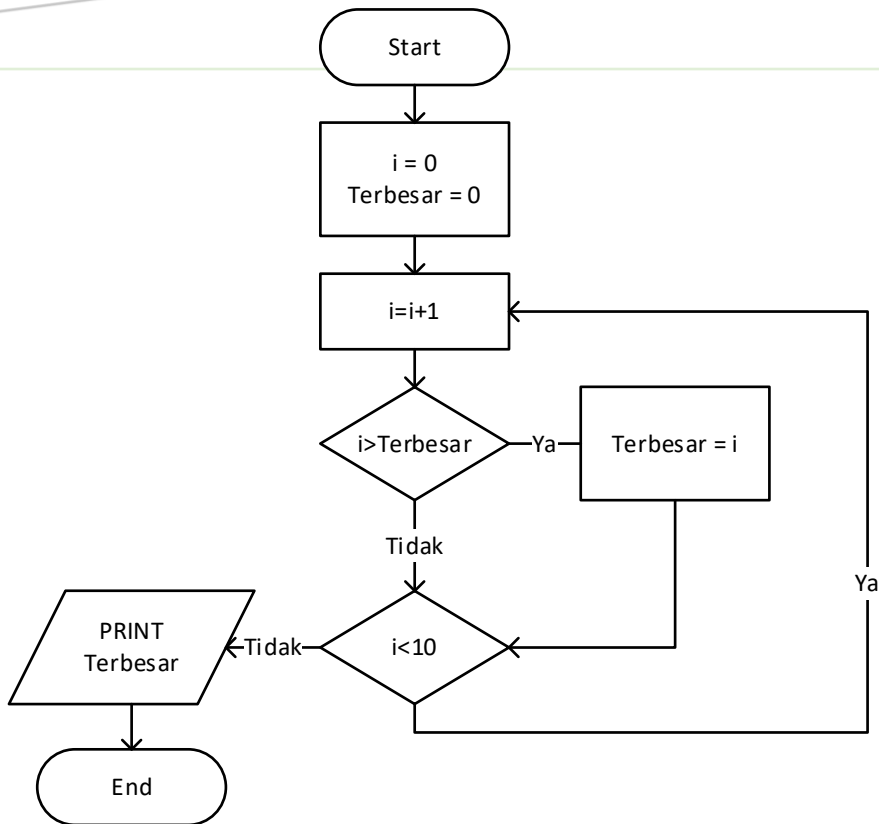
1 // Program mencetak nilai terbesar menggunakan while
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14
15
16
17     return 0;
18 }

```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Ilustrasi Setiap Baris Program.

2) Deskripsi Soal:

Berikut diagram alir untuk mencetak bilangan terbesar, buatlah program Bahasa C dengan menggunakan perulangan do while.



Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buatlah Deskripsi Algoritma dan *Pseudocode*
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

```

1 // Program mencetak nilai terbesar menggunakan do while
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14
15
16
17     return 0;
18 }
  
```

3) Deskripsi Soal:

Dari diagram alir pada soal nomer satu dan dua buatlah kesimpulan singkat, apa yang kalian fahami.

Jawaban:

1. *Problem:* Membuat/menuliskan kode program Bahasa C dengan menggunakan perulangan *for*, *while* dari diagram alir
Solution: memahami diagram alir, membuat deskripsi algoritma, pseudocode kemudian menuliskan kode program dalam Bahasa C.

Deskripsion Algoritma	Pseudocode
1. Mulai program	Algoritma BilanganTerbesar
2. Baca Nilai i, Terbesar	Read i, Terbesar
3. Cek kondisi apakah i <= 10, jika ya maka, prosesnya:	While (i <= 10) do
a. Cek apakah nilai i > Terbesr, jika ya	if (i>Terbesar) then
b. Update nilai Terbesar = i	Terbesar=i
c. Jalankan i=i+1	end if
d. Ulangi proses sampai kondisi memenuhi nilai i <=10	i=i-1
4. Jika nilai i terpenuhi i<=10 maka	End While
5. Cetak nilai Terbesar	Print Terbesar
6. Program berakhir	

Kode Program

```
1 // Program mencetak nilai terbesar menggunakan for
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14     int Terbesar=0;
15     for (int i=0;i <=10;i++ ) {
16         if ( i>Terbesar) {
17             Terbesar=i;
18         }
19     }
20     printf("%d", Terbesar);
21
22     return 0;
23 }
```

```
1 // Program mencetak nilai terbesar menggunakan while
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
13 {
14     int i=0, Terbesar=0;
15     while (i <=10 ) {
16         if ( i>Terbesar) {
17             Terbesar=i;
18         }
19         i=i+1;
20     }
21     printf("%d", Terbesar);
22
23     return 0;
24 }
```

Ilustrasi Setiap Baris Program untuk for:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel Terbesar
15	Perulangan menggunakan for dengan kondisi Nilai i <= 10
16	Cek kondisi nilai i >Terbesar
17	Update nilai Terbesar=i
18	Akhir dari kondisi pengecekan nilai i>Terbesar
19	Akhir dari perulangan
20	Cetak nilai Terbesar
21	
22, 23	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

Ilustrasi Setiap Baris Program untuk while:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel i, Terbesar
15	Perulangan menggunakan while dengan kondisi Nilai i <= 10
16	Cek kondisi nilai i >Terbesar
17	Update nilai Terbesar=i
18	Akhir dari kondisi pengecekan nilai i>Terbesar
19	Update nilai i dengan i=i+1
20	Akhir dari perulangan
21	Cetak nilai Terbesar
22	
23, 24	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

2. *Problem:* Membuat/menuliskan kode program Bahasa C dengan menggunakan perulangan *do-while* dari diagram alir

Solution: memahami diagram alir, membuat deskripsi algoritma, pseudocode kemudian menuliskan kode program dalam Bahasa C.

Deskripsion Algoritma	Pseudocode
1. Mulai program 2. Baca Nilai i, Terbesar 3. Jalankan i=i+1 4. Cek apakah nilai i > Terbesr, jika ya 5. Update nilai Terbesar = i 6. Cek kondisi apakah nilai i<=10, jika ya maka ulangi proses sampai terpenuhi nilai i<=10 7. Cetak nilai Terbesar 8. Program berakhirAlgoritma BilanganTerbesar	<i>Algoritma</i> BilanganTerbesar <i>Read</i> i, Terbesar do i=i+1 if (i>Terbesar) then Terbesar=i end if while (i <10) Print Terbesar

Kode Program

```
1 // Program mencetak nilai terbesar menggunakan do while
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 int main()
```

```
13 {
14     int i=0, Terbesar=0;
15     do {
16         i=i+1;
17         if ( i>Terbesar) {
18             Terbesar=i;
19         }
20     } while (i <10 );
21     printf("%d", Terbesar);
22
23     return 0;
24 }
```

Ilustrasi Setiap Baris Program untuk do-while:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12, 13	Membaca badan program
14	Membentuk variabel i, Terbesar
15	Perulangan menggunakan do, kode program yang berada di dalam block do-while akan dijalankan terlebih dahulu baru dilakukan pengecekan kondisi
16	Update nilai i dengan i=i+1
17	Cek kondisi nilai i >Terbesar
18	Update nilai Terbesar=i
19	Akhir dari kondisi pengecekan nilai i>Terbesar
20	Cek kondisi perulangan apakah nilai i<10, jika ya ulangi sampe kondisi terpenuhi
21	Cetak nilai Terbesar
22	
23, 24	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

3. Kesimpulan:
- Untuk for dan while dapat menggunakan diagram alir yang sama, tetapi untuk kasus soal do while tidak bisa menggunakan diagram alir yang sama tetapi menggunakan diagram alir yang sesuai dengan do-while, karena untuk kasus for dan while akan di cek dulu kondisinya, jika sesuai maka baru di eksekusi programnya, dan uuntuk kasus do while itu kebalikannya program akan di eksekusi dulu baru di cek kondisinya.

6. Kontrol Perulangan

Perulangan atau disebut *looping* adalah perintah kusus dalam Bahasa pemrograman dan algoritma yang digunakan untuk mengulang beberapa perintah sesuai dengan kondisi yang telah ditentukan. Pada bahasa pemrograman ada beberapa jenis perulangan sebagai berikut:

- 1) *For*
- 2) *While*
- 3) *Do while*

Contoh Latihan 8: Latihan Perulangan menggunakan For

Deskripsi Soal:

Membuat program untuk menampilkan kata “halo dunia” sebanyak 5 kali.

Petunjuk:

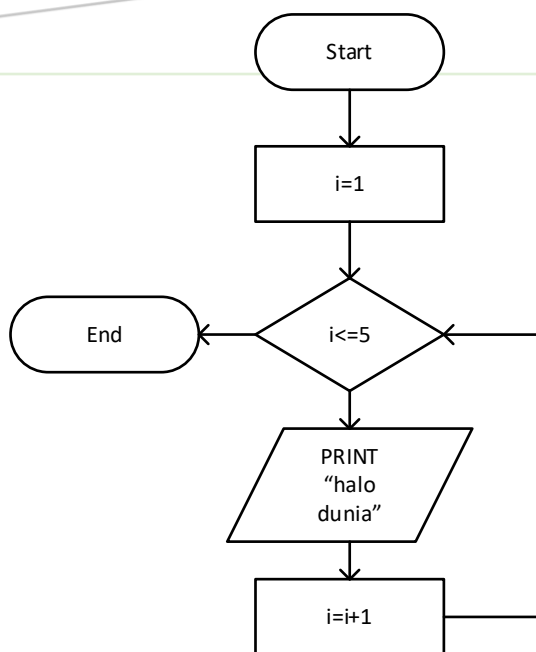
- 1. Siapkan kertas dan alat tulis sebagai media simulasi.
- 2. Gunakan satu lembar kertas sebagai lembar kerja program
- 3. Kerjakan secara mandiri.
- 4. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: kontrol_perulangan1

Langkah-langkah:

- 1. Buka Lembar Kerja Program
- 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

<i>Problem</i>	Menampilkan kata “halo dunia” sebanyak 5 kali
<i>Solution</i>	Menggunakan for

- 3. Membuat Diagram Alir



4. Menulis program

```

1  /* Program menampilkan kata secara berulang */
2  #include <stdio.h>
3  int main(){
4
5      for (int i = 1; i <= 5; i++) {
6          printf("halo dunia");
7      }
8      return 0;
9  }
  
```

5. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri, pada proses ini menggunakan pendekatan ilustrasi setiap baris, agar siswa lebih memahami.

Ilustrasi dalam menjalankan program , Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. Perhatikan ilustrasi berikut		
Baris Kode Program	Keterangan eksekusi	
1	Tidak akan dijalankan karena berisi komentar	
2	Memanggil fungsi stdio.h	
3	Membaca badan program	
4		
5,6,7	Merupakan block perulangan, perintahnya ulang dari kondisi dari i = 1 sampai dengan i<=5. Ilustrainya sebagai berikut:	
	Loop	Deskripsi
	1	i=1 cek apakah 1 <= 5 ? Ya i++ atau i=i+1 jadi nilai i=1+1=2 tampilkan kata "halo dunia"
	2	i=2 cek apakah 2 <= 5 ? Ya i++ atau i=i+1 jadi nilai i=2+1=3 tampilkan kata "halo dunia"

	3	i=3 cek apakah 3 <= 5 ? Ya i++ atau i=i+1 jadi nilai i=3+1=4 tampilkan kata "halo dunia"
	4	i=4 cek apakah 4 <= 5 ? Ya i++ atau i=i+1 jadi nilai i=4+1=5 tampilkan kata "halo dunia"
	5	i=5 cek apakah 5 <= 5 ? Ya i++ atau i=i+1 jadi nilai i=5+1=6 tampilkan kata "halo dunia"
	6	i=6 cek apakah 6 <= 5 ? Tidak proses loo[berhenti
8, 9	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.	

AP-09

Modul Ajar Algoritma dan Pemrograman
Aktivitas Unplugged

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	3 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa mampu mengaplikasikan konsep-konsep pemrograman <i>procedural</i>		
Kata kunci	Procedural, Fungsi, Algoritma, Diagram Alir, Pemrograman, Bahasa C		
Deskripsi umum kegiatan	Siswa mempelajari bagaimana penggunaan fungsi procedural dalam Bahasa C dan menulis kode program dengan menggunakan IDE Bahasa C, guru akan memberikan contoh penggunaan fungsi <i>procedural</i> dalam bahasa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	<p>Materi Ajar: Konsep pemrograman procedural pada Bahasa C</p> <p>Alat: Gunting</p> <p>Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas, lem, karton</p>		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-9: Fungsi (3 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Membuat Program dengan Bahasa C	Siswa mampu mengaplikasikan konsep-konsep pemrograman procedural	3 JP	Latihan Mandiri Lembar Kerja Keaktifan

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik mengeksplorasi perintah fungsi procedural dalam Bahasa C, secara mandiri maupun kolaborasi.	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat program Bahasa C secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Apa yang kalian ketahui tentang fungsi ?
- 2. Pernahkan melakukan operasi sebuah fungsi ? berikan contohnya ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

- 1. Guru menyampaikan materi dan dapat memberikan penjelasan langsung mengenai program yang akan dibuat pada latihan ini.
- 2. Guru memberi **Contoh Latihan 9, Latihan Fungsi** dan menjelaskannya.
- 3. Guru memancing siswa untuk melakukan debug dan run dengan menjelaskan ilustrasi setiap baris program.
- 4. Guru membentuk kelompok yang terdiri dari 2-5 orang:

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan ilustrasi setiap baris program

- 5. Guru mengarahkan siswa ke aktivitas Ayo, Kita Berlatih **Fungsi**. Biarkanlah siswa mengerjakan latihan mandiri. Guru memberikan umpan balik dan *scaffolding* pada siswa yang mengalami kesulitan.
- 6. Minta siswa secara berpasangan mengecek kode dan program yang telah dibuat oleh siswa lainnya. Mintalah siswa menguji program rekan mereka dengan menggunakan kasus uji yang diberikan, atau dibuat sendiri oleh siswa. Setelah itu, pancing siswa untuk memperbaiki program apabila terdapat kesalahan pada program.
- 7. Guru dapat membahas aktivitas dengan meminta beberapa siswa menuliskan deskripsi dan menjelaskan hasil pekerjaannya. Hal ini akan melatih siswa dalam mengkomunikasikan suatu artefak pemrograman dengan rekannya
- 8. Guru dapat mengulas kembali beberapa kesalahan umum yang terjadi pada saat latihan menulis program. Setelah itu, siswa dapat diminta untuk memperbaiki program di sisa waktu, atau dikerjakan kembali ke rumah.
- 9. Guru menginformasikan hasil pengerjaannya dan dapat dikumpulkan untuk dinilai.

Penutup

- 1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
- 2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
- 3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian tentang fungsi procedural dalam Bahasa C?	
Apa yang kalian rasakan ketika mencoba memahami, membuat program dalam fungsi procedural Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara <i>unplugged</i> ?	

G. Lembar Kerja Peserta Didik

Aktivitas Ayo, Kita Berlatih Fungsi

Petunjuk:

1. Siswa menyiapkan alat tulis untuk menjawab pertanyaan.
2. Siswa mengerjakan soal secara berkelompok yang terdiri dari 2-5 orang.

1. Bertugas Membuat Diagram Alir
2. Bertugas Membuat Deskripsi Algoritma, Pseudocode
3. Bertugas Membuat/Menulis Kode Program
4. Bertugas sebagai Debug dan Run
5. Bertugas Membuat Skenario Input dan Ilustrasi setiap baris program

3. Siswa Mengerjakan Soal Latihan berikut:

Soal:

1) Deskripsi Soal:

Buatlah kode program dengan menggunakan fungsi untuk menghitung luas permukaan kubus.

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buatlah Diagram Alir, Deskripsi Algoritma dan *Pseudocode*
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

```
1 // Program luas permukaan kubus
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 float luasPermukaanKubus(int sisi){
13
14     return luas;
15 }
16
17 int main()
18 {
19
```



```
20     return 0;
21 }
```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Ilustrasi Setiap Baris Program.

2) Deskripsi Soal:

Buatlah kode program dengan menggunakan fungsi untuk menampilkan kata "Halo semua" sebanyak 3 kali.

Petunjuk:

- Siapkan kertas dan alat tulis sebagai media simulasi.
- Dari pemahaman soal, tentukan *Problem* (permasalahannya apa?) dan *Solution* (Pemecahannya bagaimana?).
- Buatlah Diagram Alir, Deskripsi Algoritma dan *Pseudocode*
- Gunakan satu lembar kertas sebagai lembar kerja program.
- Tuliskan Kode Program pada lembar kerja program.

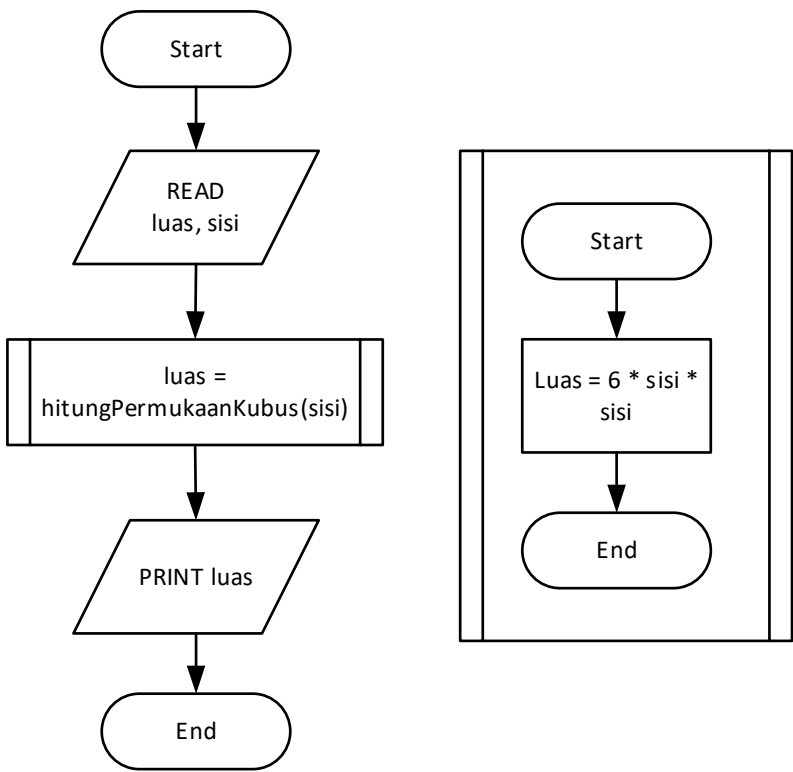
```
1 // Program Menmpilkan Kata
2 /*
3     Nama Kelompok:
4     Anggota Kelompok:
5     1.
6     2.
7     3.
8     4.
9     5.
10 */
11 #include <stdio.h>
12 char* menampilkanKata(){
13     return "Halo semua \n";
14 }
15
16 int main()
17 {
18     char* kata;
19
20
21     return 0;
22 }
```

- Lakukan Pengecekan Program (Debug dan Run)
- Buat Ilustrasi Setiap Baris Program.

Jawaban:

1. *Problem*: Membuat/menuliskan kode program Bahasa C dengan menggunakan fungsi untuk menghitung luas permukaan kubus
Solution: menggunakan rumus luas permukaan kubus = $6 * sisi*sisi$. Membuat deskripsi algoritma, *pseudocode* kemudian menuliskan kode program dalam Bahasa C.

Diagram Alir:



Deskripsion Algoritma	Pseudecode
<div>1. Mulai program</div> <div>2. Baca Nilai luas, sisi</div> <div>3. Menghitung luas dengan memanggil fungsi luasPermukaanKubus</div> <div>4. Menjalankan fungsi luasPermukaanKubus<ul style="list-style-type: none">- Memulai fungsi- Menghitung rumus luas- Mengakhiri fungsi</div> <div>5. Cetak nilai luas</div> <div>6. Program berakhir</div>	<div>Algoritma LuasPermukaanKubus</div> <div>Function luasPermukaanKubus(sisi) luas = 6 * sisi * sisi return luas</div> <div>Read luas,sisi luas=luasPermukaanKubus(sisi) Print luas</div>

Kode Program

```
1 // Program luas permukaan kubus
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
```

```

12 float luasPermukaanKubus(int sisi){
13
14     float luas;
15     luas=6*sisi*sisi;
16     return luas;
17 }
18
19 int main()
20 {
21     float luas;
22     int sisi;
23     scanf("%d", &sisi);
24     luas=luasPermukaanKubus(sisi);
25     printf("%f",luas);
26     return 0;
27 }

```

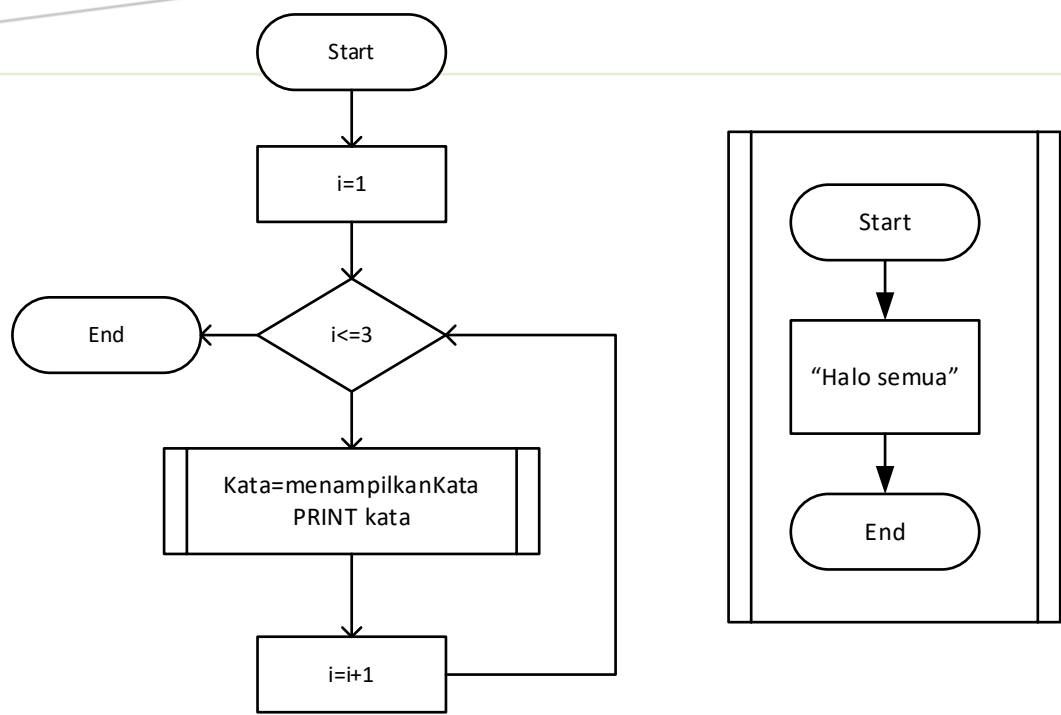
Ilustrasi Setiap Baris Program:

Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12-17	Blok fungsi luasPermukaanKubus
18	
19,20	Membaca badan program
21	Membentuk variabel luas dengan tipe data float
22	Membentuk variabel sisi dengan tipe data int
23	Membaca nilai variabel sisi yang dimasukan
24	Menghitung luas dengan cara memanggil fungsi luasPermukaanKubus
25	Cetak nilai luas
26, 27	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

2. *Problem:* Membuat/menuliskan kode program Bahasa C dengan menggunakan fungsi untuk menampilkan kata “Halo semua” sebanyak 3 kali
- Solution:* karena fungsi sudah disediakan dalam kode programnya, solusi selanjutnya menerapkan fungsi tersebut dalam perulangan menggunakan for.

Diagram Alir:



Deskripsi Algoritma	Pseudecode
<div>1. Mulai program</div> <div>2. Baca i</div> <div>3. Cek kondisi nilai $i \leq 3$, maka memanggil fungsi menampilkanKata dan mencetak kata “Halo semua”</div> <div>4. Menjalankan fungsi menampilkanKata<ul style="list-style-type: none">- Memulai fungsi- Mengembalikan kata “Halo semua”- Mengakhiri fungsi</div> <div>5. Jika nilai i lebih besar dari 3 maka Program berakhir</div>	<div>Algoritma MenampilkanKata</div> <div>Function menampilkanKata() return “Halo semua”</div> <div>Read kata</div> <div>For (i=1;i<=3;i++) kata=menampilkanKata() Print kata End For</div>

Kode Program

```
1 // Program Menmpilkan Kata
2 /*
3  Nama Kelompok:
4  Anggota Kelompok:
5  1.
6  2.
7  3.
8  4.
9  5.
10 */
11 #include <stdio.h>
12 char* menampilkanKata(){
```

```
13     return "halo semua \n";
14 }
15
16 int main()
17 {
18     char* kata;
19     for(int i=1;i<=5;i++){
20         kata=menampilkanKata();
21         printf("%s",kata);
22     }
23     return 0;
24 }
```

Ilustrasi Setiap Baris Program:

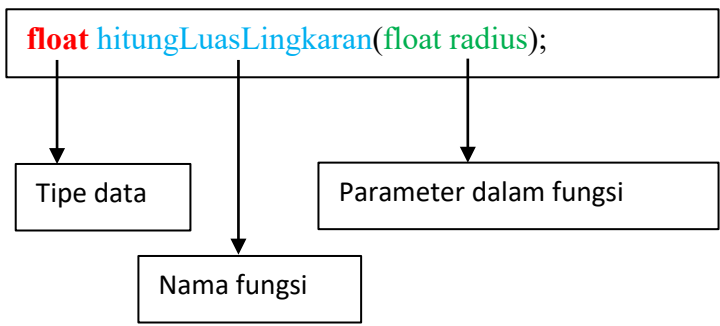
Ilustrasi dalam menjalankan program, Kode Program akan dijalankan mulai baris pertama hingga baris terakhir. ilustrasi berikut ini

Baris Kode Program	Keterangan eksekusi
1 - 10	Tidak akan dijalankan karena berisi komentar
11	Memanggil fungsi stdio.h
12-14	Blok fungsi luasPermukaanKubus
15	
16,17	Membaca badan program
18	Membentuk variabel char*
19	Perulangan dari nilai i=1 sampai dengan i<=3
20	memanggil fungsi menampilkanKata dan menyimpannya dalam variabel kata
21	Cetak kata
22	Akhir dari perulangan
23, 24	Return 0; dimaksud mengembalikan nilai 0 dan proses program berhenti.

Fungsi adalah sub-program yang berisi kode program untuk menyelesaikan tugas tertentu dan dapat digunakan dalam program itu sendiri, maupun di program yang lain. Sebagai contoh dalam Bahasa C, fungsi yang sering digunakan adalah `main()`, fungsi ini merupakan yang wajib ada dalam pemrograman Bahasa C yang merupakan badan utama program dan akan dieksekusi pertama kali.

a. Cara membuat fungsi

yang perlu diperhatikan dalam membuat fungsi adalah **tipe data** untuk mengembalikan nilai fungsi, **nama fungsi** untuk mempersentasikan tujuan dari fungsi, dan **parameter** yang dimasukan dalam fungsi.



Contoh penulisan fungsi secara lengkap sebagai berikut

```
float hitungLuasLingkaran(float radius) {
    // berisi kode program
    float luas; // variable luas
    luas = 3.14 * radius * radius; // rumus luas
    return luas; // mengembalikan nilai luas
}
```

b. Memanggil fungsi

Fungsi yang sudah dibuat dapat dipanggil (*function call*) di dalam kode program. Sebagai contoh memanggil fungsi `hitungLuasLingkaran`.

```
1  /* Program Hitung Luas Lingkaran */
2
3  #include <stdio.h>
4  float hitungLuasLingkaran(float radius) {
5      // berisi kode program
6      float luas; // variable luas
7      luas = 3.14 * radius * radius; // rumus luas
8      return luas; // mengembalikan nilai luas
9  }
10
11 int main() {
12     float radius, luas;
13     scanf("%f", &radius);
14     luas = hitungLuasLingkaran(radius);
15     printf("%f\n", luas);
16     return 0;
}
```

Pada baris ke-14 akan memanggil fungsi hitungLuasLingkaran. Cukup mudah bukan dalam penggunaan fungsi.

Contoh Latihan 9: Latihan Membuat Fungsi

Deskripsi Soal:

Membuat program untuk menghitung luas lingkaran menggunakan fungsi.

Petunjuk:

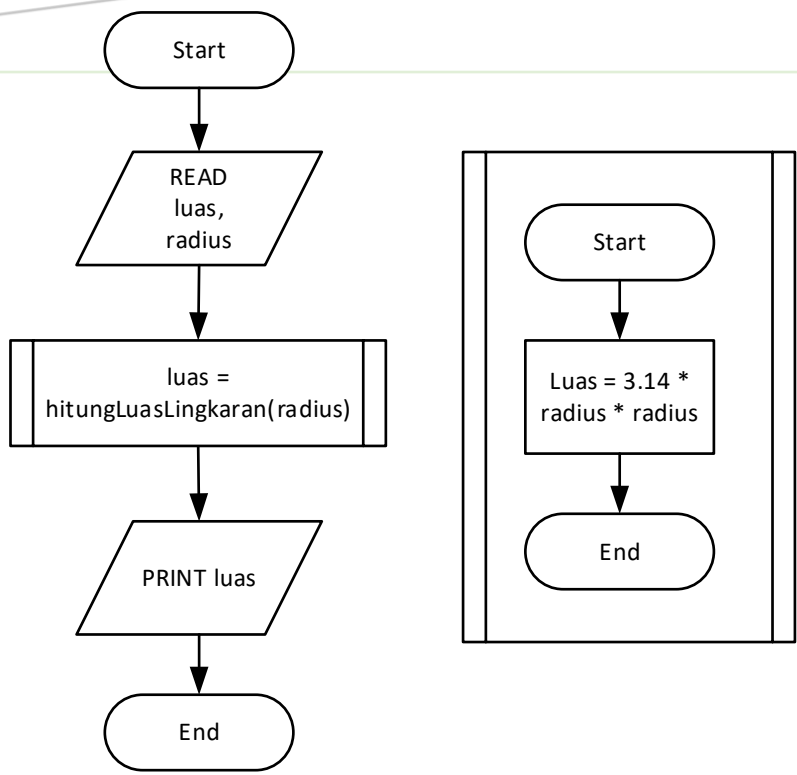
- 1. Siapkan kertas dan alat tulis sebagai media simulasi.
- 2. Gunakan satu lembar kertas sebagai lembar kerja program
- 3. Kerjakan secara mandiri.
- 4. Siswa dianggap sudah membuat lembar kerja program dan menyimpan dengan nama file: *fungsi1*

Langkah-langkah:

- 1. Buka Lembar Kerja Program
- 2. Fahami Soal dan tentukan *problem* dan *solution* dari berdasarkan soal yang diberikan.

<i>Problem</i>	Menghitung luas lingkaran
<i>Solution</i>	Menggunakan rumus luas lingkaran = 3.14 * radius * radius. RUmus tersebut dibuat fungsi

- 3. Membuat Diagram Alir



4. Menulis program

```
1  /* Program Hitung Luas Lingkaran */
2
3  #include <stdio.h>
4  float hitungLuasLingkaran(float radius) {
5      // berisi kode program
6      float luas; // variable luas
7      luas = 3.14 * radius * radius; // rumus luas
8      return luas; // mengembalikan nilai luas
9  }
10
11 int main() {
12     float radius, luas;
13     scanf("%f", &radius);
14     luas = hitungLuasLingkaran(radius);
15     printf("%f\n", luas);
16     return 0;
17 }
```

5. Menjalankan Program (dengan tombol Run atau Debug dan Run), lakukan pengecekan sendiri, pada proses ini menggunakan pendekatan ilustrasi setiap baris, agar siswa lebih memahami.

AP-10

Nama	Bambang Subeno	Jenjang/Kelas	SMA/ X
Asal sekolah	Stmik Jabar	Mapel	INFORMATIKA
Alokasi waktu	6 x 45 menit	Jumlah siswa	
Profil pelajar Pancasila yang berkaitan	<ul style="list-style-type: none">● Mandiri● Bergotong Royong● Bernalar Kritis● Kreatif	Model pembelajaran	<i>Cooperative Learning</i>
Fase	E	Elemen Mapel/ Aktivitas	Algoritma dan Pemrograman/ Unplugged
Tujuan Pembelajaran	Siswa memahami implementasi fungsi dalam program		
Kata kunci	Implementasi, fungsi, Algoritma, Diagram Alir, Pemrograman, Bahasa C		
Deskripsi umum	Siswa akan banyak berlatih penggunaan-penggunaan fungsi dalam Bahasa C yang telah dipelajari dan menulis kode program dengan menggunakan IDE Bahasa C, guru		
kegiatan	akan mengarahkan, memberikan contoh penggunaan penggunaan fungsi dalam latihan penulisan program dalam bahasa C dengan menggunakan media alat peraga yang telah disediakan.		
Materi ajar, alat, dan bahan	<p>Materi Ajar: Implementasi fungsi dalam program Bahasa C</p> <p>Alat: Gunting</p> <p>Bahan: Papan tulis, Alat peraga lembar screenshot IDE Bahasa C, kertas, lem, karton</p>		
Sarana Prasarana	Ruang Kelas		

Pertemuan ke-10: Latihan Pemrograman (6 JP)

A. Tujuan Pembelajaran

Materi	Tujuan Pembelajaran	Durasi (JP)	Asesmen
Membuat Program dengan Bahasa C	Siswa memahami implementasi fungsi dalam program	6 JP	Latihan Mandiri Lembar Kerja Keaktifan

B. Asesemen

Penilaian sesuai yang telah dituliskan dalam tabel tujuan pembelajaran di atas

C. Pemahaman Bermakna

Pengalaman Bermakna	Profil Pelajar Pancasila	Berpikir Komputasional	Praktik Inti
Peserta didik mengeksplorasi IDE Bahasa C.	Mandiri, Bernalar Kritis	Abstraksi, Algoritma, Dekomposisi	Abstraksi
Peserta didik mengeksplorasi perintah fungsi procedural dalam Bahasa C, secara mandiri maupun kolaborasi.	Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional
Peserta didik melakukan membuat program Bahasa C secara mandiri maupun kolaborasi	Mandiri, Bergotong Royong, Bernalar Kritis, Kreatif	Abstraksi, Algoritma, Dekomposisi	Kolaborasi, Pendefinisian persoalan, Abstraksi, Mengembangkan artefak komputasional

D. Pertanyaan Pemantik

- 1. Setelah kalian mempelajari algoritma dan pemrograman, ekspresikan pemahaman kalian (Mudah / Sulit / Biasa-biasa saja) ?
- 2. Maukah kalian berlatih lagi pemrograman Bahasa C ?

E. Kegiatan Pembelajaran

Pendahuluan

- 1. Membuka pembelajaran dan memeriksa kesiapan peserta didik;
- 2. Guru menjelaskan tujuan pembelajaran dan langkah-langkah pembelajaran;
- 3. Guru menjelaskan teknik asesmen/ penilaian;
- 4. Guru memeriksa penugasan kompetensi awal dengan cara tanya jawab secara lisan;
- 5. Guru diharapkan membawa apersepsi sesuai pertemuannya.

Inti

Kegiatan inti yang dilakukan adalah sebagai berikut:

1. Guru menyampaikan materi dan dapat memberikan penjelasan langsung mengenai program yang akan dibuat pada latihan ini.
2. Guru mengarahkan siswa ke aktivitas **Latihan Pemrograman**. Biarkanlah siswa mengerjakan latihan mandiri. Guru memberikan umpan balik dan *scaffolding* pada siswa yang mengalami kesulitan.
3. Mintalah siswa lain menguji program rekan mereka dengan menggunakan kasus uji yang diberikan.
4. Guru menanyakan hasil pengujian yang dilakukan oleh siswa. Setelah itu, pancing siswa untuk memperbaiki program apabila terdapat kesalahan pada program.
5. Guru dapat membahas aktivitas dengan meminta beberapa siswa menuliskan deskripsi dan menjelaskan hasil pekerjaannya. Hal ini akan melatih siswa dalam mengkomunikasikan suatu artefak pemrograman dengan rekannya
6. Guru dapat mengulas kembali beberapa kesalahan umum yang terjadi pada saat latihan menulis program. Setelah itu, siswa dapat diminta untuk memperbaiki program di sisa waktu, atau dikerjakan kembali ke rumah.
7. Guru menginformasikan hasil pengerjaannya dan dapat dikumpulkan untuk dinilai.

Penutup

1. Guru menutup kelas dan mengarahkan siswa pada aktivitas refleksi bahwa algoritma dan pemrograman secara unplugged adalah pengetahuan yang dapat diimplementasikan pada kehidupan sehari-hari.
2. Guru menginformasikan pembelajaran pada pertemuan selanjutnya;
3. Berdoa menutup pembelajaran.

F. Refleksi

Guru

Setelah mengajarkan materi algoritma dan pemrograman secara *unplugged*, guru diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apakah ada sesuatu yang menarik selama pembelajaran?	
Apa yang sudah berjalan baik di dalam kelas?	
Apa yang Anda sukai dari kegiatan pembelajaran kali ini dan Apa yang tidak Anda sukai?	
Apa yang ingin Anda ubah untuk meningkatkan/ memperbaiki pelaksanaan/ hasil pembelajaran?	
Yang harus/akan saya ingat ketika menyusun pembelajaran di masa depan	
Apa masukan atau pengetahuan yang paling dominan dipelajari peserta didik saya selama unit ini berlangsung	

Peserta Didik

Setelah mempelajari materi algoritma dan pemrograman secara *unplugged*, peserta didik diharapkan merefleksi proses pembelajaran yang telah dilakukan. Adapun kegiatan refleksi sebagai berikut:

Pertanyaan	Jawaban
Apa pendapat kalian pemrograman dalam Bahasa C?	
Apa yang kalian rasakan ketika pemrograman dalam Bahasa C secara <i>unplugged</i> ?	
Setelah menyelesaikan semua LKPD, bagaimana perasaan kalian belajar pemrograman Bahasa C secara <i>unplugged</i> ?	

G. Lembar Kerja Peserta Didik

1. *Problem:* Mencetak Faktor Prima. Kalian pasti telah mengenal faktor prima dari suatu bilangan. Misalnya, 20 akan memiliki faktor prima 2 dan 5. Rancang dan buatlah sebuah program untuk mencetak faktor prima dari sebuah bilangan bulat yang diberikan.
2. *Problem:* Mencetak Nama Sendiri
Pada problem ini, tugas kalian ialah merancang dan membuat sebuah program yang dapat mencetak nama sendiri dengan menggunakan fungsi.
3. *Problem:* Mencetak Bilangan Genap/Ganjil
Pada problem ini, tugas kalian ialah merancang dan membuat sebuah program yang dapat mencetak bilangan genap/ganjil dari bilangan yang diinputkan dengan menggunakan fungsi.
4. *Problem:* Mencetak Luas Persegi Panjang Tanpa Fungsi
Pada problem ini, tugas kalian ialah merancang dan membuat sebuah program yang dapat mencetak Luas Persegi Panjang dari bilangan yang diinputkan tanpa menggunakan fungsi.
5. *Problem:* Mencetak Luas Persegi Panjang Dengan Fungsi
Pada problem ini, tugas kalian ialah merancang dan membuat sebuah program yang dapat mencetak Luas Persegi Panjang dari bilangan yang diinputkan dengan menggunakan fungsi.
6. *Problem:* Mengecek Tanggal
Menulis tanggal tidak boleh sembarang. Ada aturan-aturan yang berlaku. Misalnya, ada aturan tahun kabisat serta jumlah hari dalam suatu bulan yang telah ditentukan. Pada problem ini, tugas kalian ialah merancang dan membuat sebuah program yang dapat memeriksa apakah sebuah tanggal dengan format yang kalian rancang, saat diberikan oleh pengguna dinyatakan benar.

H. Bahan Bacaan

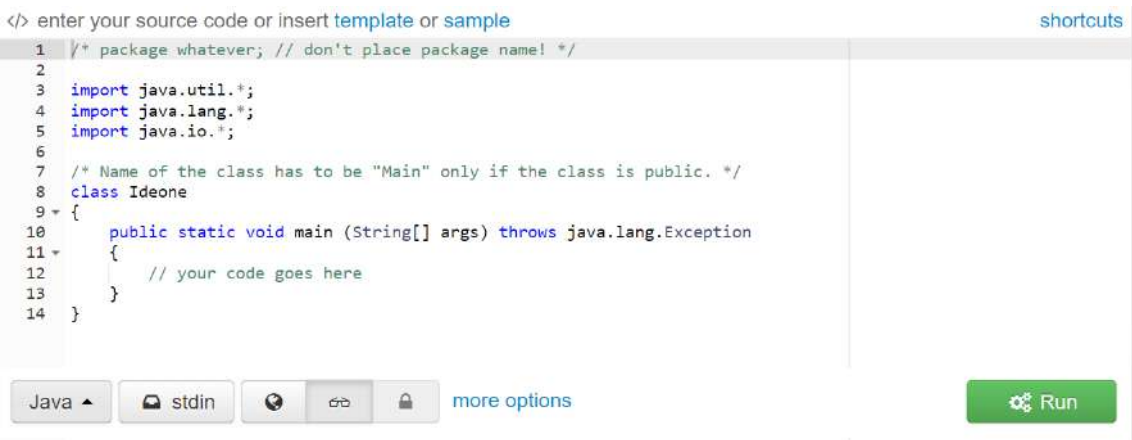
Latihan pemrograman ini diberikan untuk melatih keterampilan siswa. Siswa akan diberikan beberapa problem yang harus kalian selesaikan dengan pemrograman. Gunakan kemampuan berpikir komputasional kalian untuk menganalisis permasalahan yang diberikan. Rancanglah strategi dalam bentuk algoritma untuk menyelesaikan permasalahan tersebut. Kemudian, implementasikan algoritma tersebut ke dalam bentuk menuliskan kode program menggunakan bahasa C

Pengayaan dan Remedial

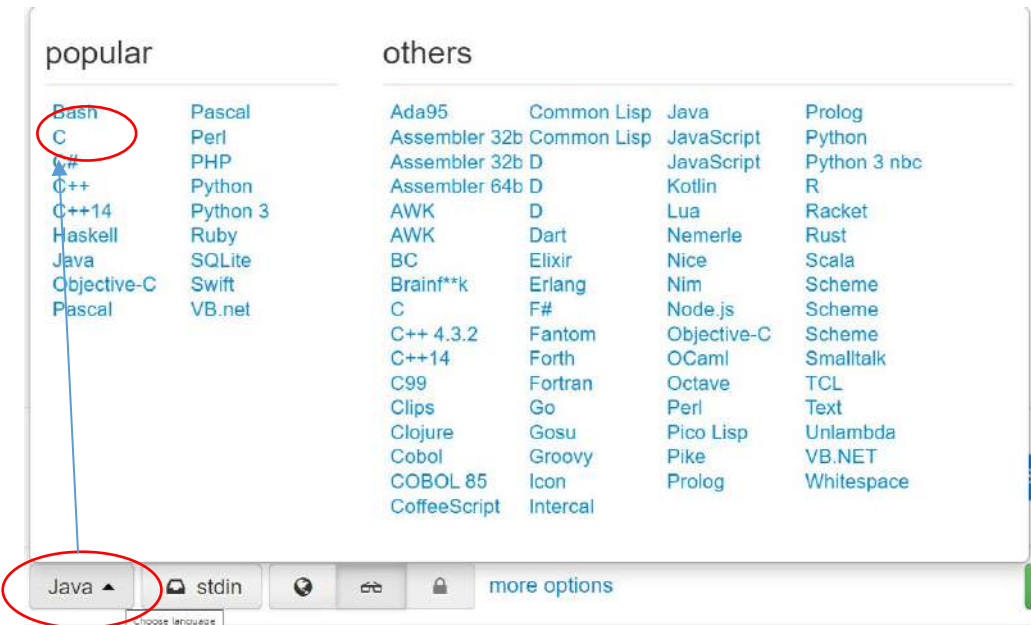
- 1. Menenal IDE Bahasa C secara *online* atau daring, tanpa melakukan instalasi IDE Bahasa C
 - a. IDE ideone

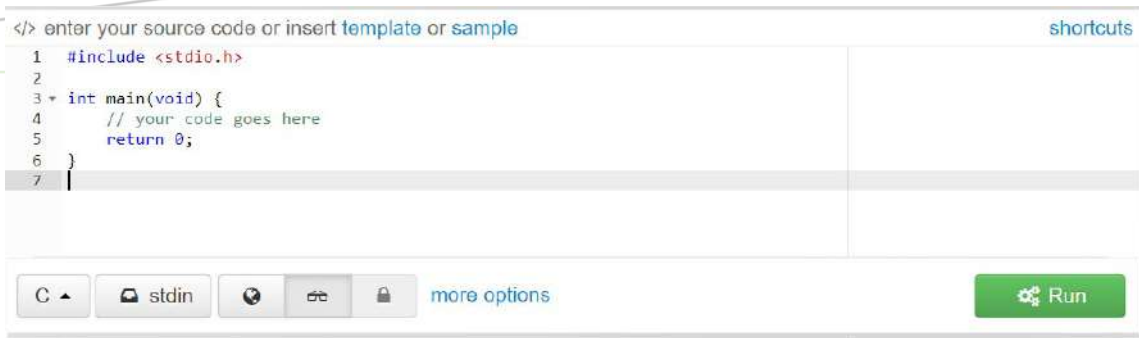
Ideone merupakan compiler secara online tanpa melakukan instalasi terlebih dahulu. Pengguna cukup mengunjungi situs dengan menginputkan alamat <https://ideone.com> melalui browser. Tampilan dari ideone adalah sebagai berikut:

Pertama muncul akan seperti berikut:



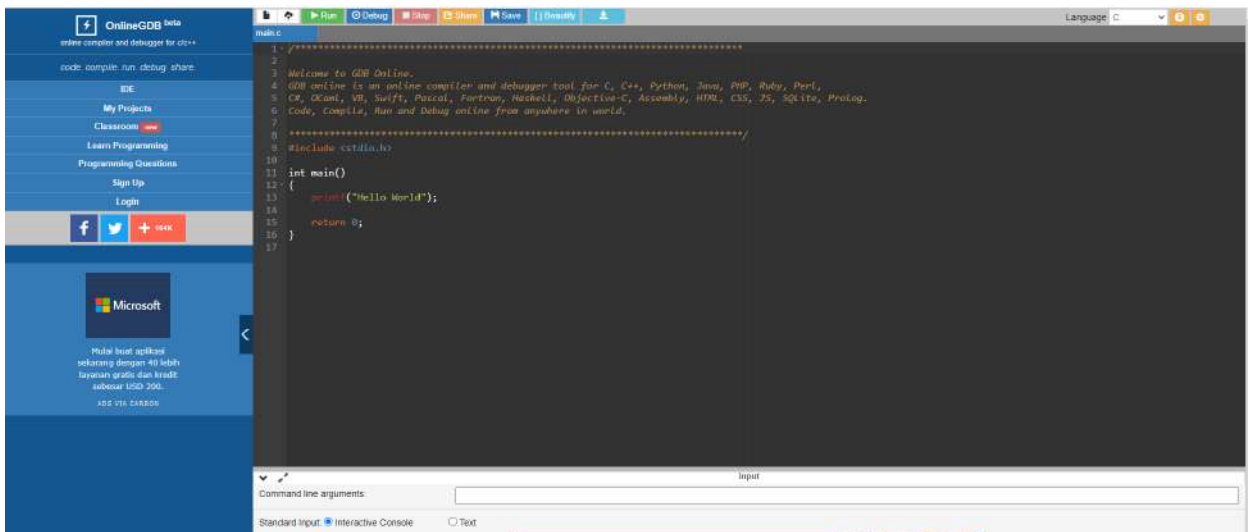
Secara default ideone akan menampilkan Bahasa Java, kita ubah Bahasa pemrograman yang digunakan menjadi Bahasa C.





b. onlineGDB

onlineGDB merupakan compiler secara online tanpa melakukan instalasi terlebih dahulu. Pengguna cukup mengunjungi situs dengan menginputkan alamat <https://onlineGDB.com> melalui browser. Tampilan dari onlineGDB adalah sebagai berikut:



Secara default menggunakan Bahasa C.

2. Contoh Kode Program: Menghitung Jumlah Bilangan dari 1 hingga N

```
/* Program Hitung Jumlah Bilangan 1 s/d N dalam Bahasa C */
#include <stdio.h>

int main() {
    int n, jumlah;
    scanf("%d", &n);
    jumlah = 0;

    for (int i = 0; i < n; i++) jumlah = jumlah + i;
    printf("jumlah bilangan 1 s/d %d = %d\n", n - 1, jumlah);
    return 0;
}
```

3. Contoh Kode Program: Menampilkan Biodata

```
/* Program Menampilkan Biodata*/  
#include <stdio.h>  
  
int main() {  
    printf("Nama: Asep\n");  
    printf("Alamat: Bandung\n");  
    printf("Pendidikan: SMA\n");  
  
    return 0;  
}
```


Glosarium

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

bahasa pemrograman (*programming language*) kumpulan perintah, instruksi, dan sintaks lain yang digunakan untuk membuat suatu program

blok (*block*) sekumpulan kalimat C yang ditulis di antara { dan }

debug Proses identifikasi dan perbaikan suatu kesalahan pada kode program

deklarasi (*declaration*) memperkenalkan suatu entitas sebagai bagian dari suatu lingkup program, yang meliputi memberikan nama dan sifat dari objek contoh: deklarasi variable

pseudokode (*pseudocode*) deskripsi program informal yang tidak mengandung sintaks kode atau pertimbangan teknologi yang mendasari

sintaks (*syntax*) aturan yang mendefinisikan cara menulis elemen bahasa pemrograman yang legal (harus dipatuhi oleh pemrograman) tanpa mepedulikan makna dari penulisan tersebut

variabel (*variable*) nama simbolik yang digunakan untuk melacak nilai yang dapat berubah saat program berjalan

Daftar Pustaka

Wahyono, dkk (2021). Informatika Buku Panduan Guru: Jakarta. Kemendikbudristek.

Musthofa, dkk (2021). Informatika: Jakarta. Kemendikbudristek.

User Manual Code::Blocks, <https://www.codeblocks.org/>

Learn Pemrograman, <https://www.onlinegdb.com/>

Penilaian dalam Tim

Komponen	Baik Sekali	Baik	Cukup	Kurang
Pembagian Peran	Peran terbagi ke semua anggota dengan sangat baik.	Peran terbagi ke semua anggota dengan baik.	Peran terbagi ke semua anggota dengan cukup baik.	Peran tidak terbagi ke semua anggota
Pembagian Tugas	Tugas terbagi ke semua anggota dengan sangat baik.	Tugas terbagi ke semua anggota dengan baik.	Tugas terbagi ke semua anggota dengan cukup baik.	Tugas tidak terbagi ke semua anggota.



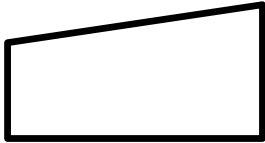

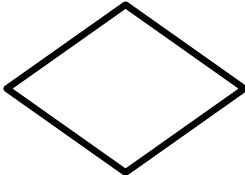


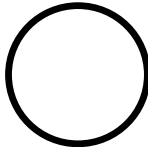
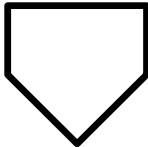
Penilaian Individu

Komponen	Baik Sekali	Baik	Cukup	Kurang
Keaktifan	Peserta Didik sangat aktif ketika bekerja individu maupun tim.	Peserta Didik aktif ketika bekerja individu maupun tim.	Peserta Didik cukup aktif ketika bekerja individu maupun tim.	Peserta Didik kurang aktif ketika bekerja individu maupun tim.

Penilaian Aktivitas

Komponen	Baik Sekali	Baik	Cukup	Kurang
Mampu memahami, membaca dan menulis algoritma dalam bentuk diagram alir	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak < 40 %
Mampu memahami, membaca dan menulis algoritma dalam bentuk deskripsi algoritma dan <i>pseudocode</i>	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak < 40 %
Mampu memahami cara menginstal tools IDE Bahasa C	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak < 40 %
Mampu menuliskan ulang kode program Bahasa C dan membuat deskripsi algoritma dan <i>pseudocode</i> .	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak < 40 %
Mampu memahami konsep <i>input</i> dan	Kategori tepat $\geq 80\%$	Kategori tepat	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak < 40 %

<i>output</i> program Bahasa C		sebanyak 60% - 79%		
Mampu memahami konsep <i>variable</i> dan ekspresi	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak $< 40\%$
Mampu memahami konsep struktur kontrol keputusan	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak $< 40\%$
Mampu memahami konsep dan Struktur kontrol perulangan	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak $< 40\%$
Mmapu memahami konsep pemrograman <i>procedural</i>	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak $< 40\%$
Mampu mengimplementasi fungsi dalam program	Kategori tepat $\geq 80\%$	Kategori tepat sebanyak 60% - 79%	Kategori tepat sebanyak 40% - 59%	Kategori tepat sebanyak $< 40\%$

<div>Terminator</div> <div></div> <div>Digunakan untuk memulai (<i>Start</i>) atau mengakhiri (<i>End</i>) program</div>	<div>Flow Line/Garis Alir</div> <div></div> <div>Digunakan untuk menunjukan arah aliran program dari awal sampai akhir</div>	<div>Manual Input</div> <div></div> <div>Digunakan untuk proses input secara manual</div>
<div>Input/Output</div> <div></div> <div>Digunakan untuk proses input / output Data</div>	<div>Decission / Keputusan</div> <div></div> <div>Merupakan titik percabangan untuk menentukan keputusan berdasarkan suatu kondisi</div>	<div>Process</div> <div></div> <div>Proses yang dilakukan oleh program. Misalnya operasi artimatika</div>
<div>Sub Process</div> <div></div> <div>Digunakan untuk proses lain yang telah didefinisikan sebelumnya</div>	<div>On Page Reference</div> <div></div> <div>Digunakan untuk menghubungkan titik diagram alir pada halaman yang sama</div>	<div>Off Page Refrence</div> <div></div> <div>Digunakan untuk menghubungkan titik diagram alir pada halaman yang berbeda</div>

codeblocks-20.03mingw-setup.exe

