

# Lembar Kerja Peserta Didik 1.2

## Mencoba Kode Program

Setelah menyelesaikan LKPD ini, siswa mampu:

1. Mengetahui lingkungan pengembangan (IDE/Compiler)
2. Menulis kode program dari contoh yang tersedia
3. Menerapkan proses eksekusi kode.
4. Menerapkan uji coba kode program dengan skenario sederhana

### Persiapan

1. Kenali IDE/Compiler:
  - o C: Gunakan Code::Blocks, Dev-C++, atau compiler online seperti Programiz.
  - o Python: Gunakan PyCharm, Jupyter Notebook, atau Google Colab.
  - o JavaScript: Gunakan browser (Chrome/Firefox) dengan konsol developer atau JavaScript Playground/
2. Akses, baca, dan gunakan tautan implementasi kode yang disediakan sesuai pembagian masing-masing kelompok.

### Aktivitas 1.1: Sistem ATM (Kelompok 1)

Tujuan: Mencoba program simulasi penarikan uang dengan validasi saldo.

#### Langkah-Langkah:

3. Ketikkan kode program pada compiler
4. Jalankan kode program

*Pseudocode* dan kode program (pilih salah satu):

```
1. Mulai
2. Masukkan saldo pengguna
3. Masukkan jumlah penarikan
4. Jika jumlah penarikan <= saldo
5.     Tampilkan "Transaksi berhasil"
6.     Kurangi saldo dengan jumlah penarikan
7. Jika tidak
8.     Tampilkan "Saldo tidak cukup"
9. Selesai
```

- Implementasi dalam bahasa C: <https://s.id/atm01-c>
- Implementasi dalam bahasa Python: <https://s.id/atm01-py>
- Implementasi dalam bahasa JavaScript: <https://s.id/atm01-js>

#### Output

```
Masukkan saldo: 51000
Masukkan jumlah penarikan: 450000
Saldo tidak cukup.
```

```
=== Code Execution Successful ===
```

#### 5. Input Data:

- Pengguna memasukkan saldo (contoh: 1500000).
- Pengguna memasukkan nominal transaksi (contoh: 750000).

#### 6. Validasi Input:

- Pastikan input berupa bilangan positif.
- Jika input tidak valid (huruf/negatif), tampilkan pesan: "Input tidak valid!".

#### 7. Logika Pengecekan:

- Bandingkan saldo dengan nominal transaksi.
- Jika saldo  $\geq$  transaksi:
  - i. Kurangi saldo.
  - ii. Tampilkan: "Transaksi berhasil. Saldo tersisa: [saldo]".
- Jika saldo  $<$  transaksi:
  - i. Tampilkan: "Saldo tidak mencukupi. Silakan coba nominal lain.".

8. Lakukan modifikasi pada kode program, dan catat hasilnya! Misal: ganti operator < menjadi >, ganti teks yang berada dalam fungsi “print”, atau modifikasi lainnya.

## Aktivitas 1.2: Validasi CAPTCHA (Kelompok 2)

Tujuan: Mencoba program verifikasi CAPTCHA sederhana.

### Langkah-Langkah:

1. Ketikkan kode program pada compiler
2. Jalankan kode program

*Pseudocode* dan kode program (pilih salah satu):

```
1. Mulai
2. Hasilkan kode CAPTCHA acak
3. Tampilkan kode CAPTCHA ke pengguna
4. Masukkan kode CAPTCHA oleh pengguna
5. Jika input pengguna == kode CAPTCHA
6.     Tampilkan "Verifikasi berhasil"
7. Jika tidak
8.     Tampilkan "Verifikasi gagal"
9. Selesai
```

- Implementasi dalam Bahasa C: <https://s.id/EAKFD>
- Implementasi dalam Bahasa Python: <https://s.id/kZABL>
- Implementasi dalam Bahasa JavaScript: <https://s.id/U735t>

### Output

```
Kode CAPTCHA: 3503
Masukkan kode CAPTCHA: 3503
Verifikasi berhasil.
```

```
=== Code Execution Successful ===
```

3. Buat Kode CAPTCHA Acak:
  - Generate 4 digit angka acak (contoh: 3503).
4. Validasi Input Pengguna:
  - Tampilkan kode CAPTCHA.
  - Minta pengguna memasukkan kode yang ditampilkan.

- Jika input sesuai: "Verifikasi berhasil".
  - Jika tidak: "Verifikasi gagal".
5. Lakukan modifikasi pada kode program, dan catat hasilnya! Misal: ganti nilai argumen pada fungsi random, ganti teks yang berada dalam fungsi "print", atau modifikasi lainnya.

## Aktivitas 1.3: Pemesanan Tiket Digital (Kelompok 3)

Tujuan: Mencoba program pengecekan ketersediaan kursi.

### Langkah-Langkah:

1. Ketikkan kode program pada compiler
2. Jalankan kode program

Pseudocode:

```
1. Mulai
2. Masukkan jumlah kursi tersedia
3. Masukkan jumlah kursi yang ingin dipesan
4. Jika jumlah kursi yang dipesan <= kursi tersedia
5.     Tampilkan "Pemesanan berhasil"
6.     Kurangi jumlah kursi tersedia
7. Jika tidak
8.     Tampilkan "Kursi tidak cukup"
9. Selesai
```

- Implementasi dalam Bahasa C: <https://s.id/gmOm3>
- Implementasi dalam Bahasa Python: <https://s.id/endkS>
- Implementasi dalam Bahasa JavaScript: <https://s.id/P5bKU>

**Output**

```
Jumlah kursi tersedia: 50
Masukkan jumlah kursi yang ingin dipesan: 45
Pemesanan berhasil! Sisa kursi: 5

=== Code Execution Successful ===
```

3. Input Data:

- Masukkan jumlah kursi tersedia (contoh: 50).
  - Masukkan jumlah kursi yang dipesan (contoh: 45).
4. Validasi Pesanan:
- Jika kursi tersedia  $\geq$  pesanan:
    - Kurangi kursi tersedia.
    - Tampilkan: "Pemesanan berhasil! Sisa kursi: [sisa]".
  - Jika tidak: "Kursi tidak cukup".
5. Lakukan modifikasi pada kode program, dan amati hasilnya! Misal: ganti operator  $<$  menjadi  $>$ , ganti jumlah kursi yang tersedia, atau modifikasi lainnya.

## Aktivitas 2: Diskusi Kelompok

### Panduan Diskusi bagi Siswa

Item yang Perlu Didiskusikan dalam Kelompok

1. Validasi Input:
  - Apa yang terjadi jika pengguna memasukkan nilai negatif atau tidak valid?
  - Bagaimana usulan cara mencegah kesalahan input?
2. Penerapan di Dunia Nyata:
  - Bagaimana sistem serupa digunakan dalam aplikasi yang kalian gunakan sehari-hari?
3. Fitur apa yang bisa ditambahkan untuk meningkatkan keamanan atau kenyamanan pengguna?

Catat hasil diskusi kelompok Anda!