

# LKPD 6.1

## “APLIKASI PEMROGRAMAN: AKAR KUADRAT & NILAI TRIGONOMETRI”

### Tujuan

1. Siswa dapat mengimplementasikan perhitungan akar kuadrat dan trigonometri dalam program tanpa dan dengan library eksternal.
2. Siswa memahami kelebihan penggunaan library/modul eksternal untuk operasi matematika kompleks.

### Alat dan Bahan

1. Laptop/PC dengan IDE untuk bahasa C, Python, atau JavaScript.
2. Modul matematika: `math.h` (C), `math` (Python), `Math` (JavaScript).
3. Kalkulator ilmiah untuk memverifikasi hasil perhitungan manual.

### Studi Kasus

#### Kasus 1: Menghitung Akar Kuadrat

- Tanpa Library: dalam waktu 5 menit, buatlah program untuk menghitung akar kuadrat dari suatu bilangan. Setelah 5 menit, catatlah kesulitan yang Anda alami dalam mengimplementasikan program tersebut.
- Dengan Library: perhatikan dan terapkan contoh yang diberikan guru dalam memanfaatkan library `math.h/math/Math` untuk menghitung akar kuadrat

## Kasus 2: Menghitung Nilai Trigonometri

- Tanpa Library: Tanpa Library: dalam waktu 5 menit, buatlah program untuk menghitung nilai trigonometri (sinus suatu besaran sudut dalam radian). Setelah 5 menit, catatlah kesulitan yang Anda alami dalam mengimplementasikan program tersebut.
- Dengan Library: perhatikan dan terapkan contoh yang diberikan guru dalam memanfaatkan library `math.h/math/Math` untuk menghitung sinus suatu besaran sudut dalam radian

Program dalam C, Python, dan JavaScript untuk menghitung akar kuadrat dan nilai trigonometri dengan menggunakan library eksternal.



Simpulkan perbandingan program berdasarkan pengalaman Anda dalam mencoba membuat program akar kuadrat dan nilai trigonometri tanpa library eksternal dan setelah mencoba program dengan tujuan yang sama, namun memanfaatkan library eksternal menggunakan panduan perbandingan di bawah ini:

Perbandingan Program dengan dan Tanpa Library Eksternal			
Aspek	Tanpa Library Eksternal (Manual)	Dengan Library Eksternal	Perbandingan & Kesimpulan
Jumlah Baris Kode	Lebih panjang karena harus menulis sendiri algoritma perhitungan.	Lebih pendek karena cukup memanggil fungsi dari library eksternal.	Bagaimana modularitas kode dibandingkan dengan pendekatan manual?
Kemudahan Implementasi	Lebih sulit dipahami dan diterapkan karena membutuhkan pemahaman mendalam tentang algoritma numerik.	Lebih mudah digunakan karena cukup memanggil fungsi bawaan tanpa memahami algoritma dasarnya.	Seberapa sulit atau mudah penggunaan masing-masing pendekatan?
Fleksibilitas & Reusability	Sulit digunakan kembali karena kode harus ditulis ulang untuk setiap kasus penggunaan baru.	Mudah digunakan kembali, cukup memanggil fungsi yang sama dengan parameter berbeda.	Mana yang lebih mudah digunakan ulang dalam proyek lain?

Kesimpulan Siswa	(Siswa menuliskan refleksi mereka mengenai kapan harus menggunakan metode manual dan kapan lebih baik menggunakan library eksternal.)		
------------------	---	--	--