

Nama : Salman Alfarisi

Kelas : S1SE-07-01

NIM : 2311104036

```
// MatematikaLibraries.js

function FPB(a, b) {
  while (b !== 0) {
    let temp = b;
    b = a % b;
    a = temp;
  }
  return Math.abs(a);
}

function KPK(a, b) {
  return Math.abs(a * b) / FPB(a, b);
}

function Turunan(persamaan) {
  let hasil = [];
  let pangkat = persamaan.length - 1;

  for (let i = 0; i < persamaan.length - 1; i++) {
    let koef = persamaan[i] * pangkat;
    if (koef === 0) {
      pangkat--;
      continue;
    }
    let xPart = pangkat - 1 > 0 ? `x${pangkat - 1}` : pangkat - 1 === 0 ? "x" : "";
    hasil.push(`${koef >= 0 && i !== 0 ? "+" : ""}${koef}${xPart}`);
    pangkat--;
  }

  return hasil.join("");
}

function Integral(persamaan) {
  let hasil = [];
  let pangkat = persamaan.length;

  for (let i = 0; i < persamaan.length; i++) {
    let newPangkat = pangkat - i;
    let koef = persamaan[i] / newPangkat;
    let formatKoef = koef === 1 ? "" : koef === -1 ? "-" : koef;
    hasil.push(`${koef >= 0 && i !== 0 ? "+" : ""}${formatKoef}x${newPangkat}`);
  }

  hasil.push("+C");
  return hasil.join("");
}

// Export sebagai object
module.exports = {
  FPB,
  KPK,
  Turunan,
  Integral
};
```

```

// main.js
const MatematikaLibraries = require('./MatematikaLibraries');

function main() {
  const fpb = MatematikaLibraries.FPB(60, 45);
  console.log(`FPB(60, 45) = ${fpb}`); // 15

  const kpk = MatematikaLibraries.KPK(12, 8);
  console.log(`KPK(12, 8) = ${kpk}`); // 24

  const turunan = MatematikaLibraries.Turunan([1, 4, -12, 9]);
  console.log(`Turunan([1,4,-12,9]) = ${turunan}`); // "3x2+8x-12"

  const integral = MatematikaLibraries.Integral([4, 6, -12, 9]);
  console.log(`Integral([4,6,-12,9]) = ${integral}`); // "x4+2x3-6x2+9x+C"
}

main();

```

Kode-kode yang telah disusun dan dijalankan dalam proyek ini merupakan contoh penerapan dari konsep Library Construction dalam pemrograman, khususnya di lingkungan Node.js. Dalam materi Library Construction, inti pembahasannya adalah bagaimana kita membangun sekumpulan fungsi atau modul yang dapat digunakan kembali di berbagai bagian program tanpa perlu menulis ulang logika yang sama. File MatematikaLibraries.js berfungsi sebagai pustaka (library) buatan sendiri yang menyimpan fungsi-fungsi matematika seperti FPB, KPK, Turunan, dan Integral. Setiap fungsi ditulis secara modular dan kemudian diekspor menggunakan module.exports, menjadikannya dapat digunakan di file lain.

Sementara itu, file main.js berperan sebagai konsumen dari library tersebut. Ia mengimpor pustaka MatematikaLibraries menggunakan require() dan menjalankan fungsi-fungsi di dalamnya sesuai kebutuhan. Dengan pemisahan ini, kita menunjukkan praktik library construction yang baik, yaitu membedakan antara logika bisnis (di dalam library) dan eksekusi program utama (di main file). Pendekatan ini meningkatkan keterbacaan, perawatan, dan memungkinkan reusability dari kode — inti dari tujuan membuat library dalam pengembangan perangkat lunak yang modular dan skalabel.