

**Laporan Praktikum**  
**Mata Kuliah Pemrograman Berbasis Objek**



**Pertemuan 3**

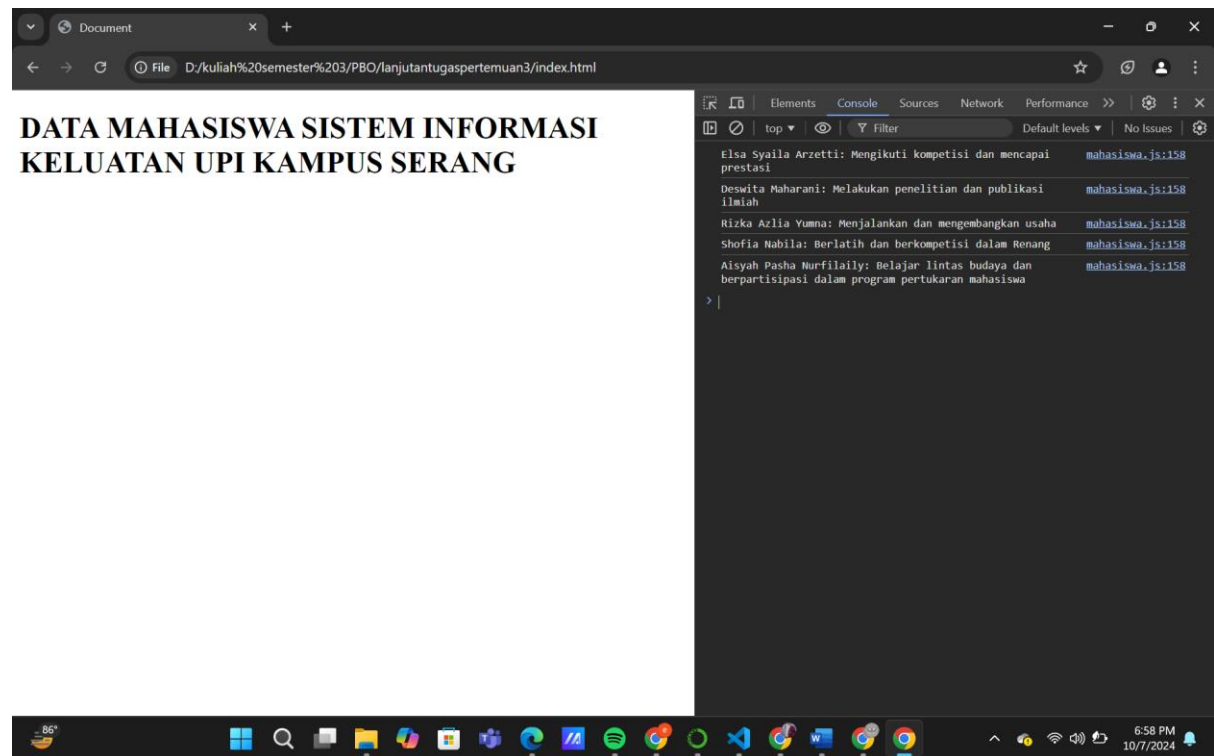
**“Tambahkan 4-5 kelas lagi dan implementasikan polymorphism pada kelas-kelas tersebut.”**

Dosen Pengampu :  
Willdan Aprizal Arifin, S.Pd., M.Kom

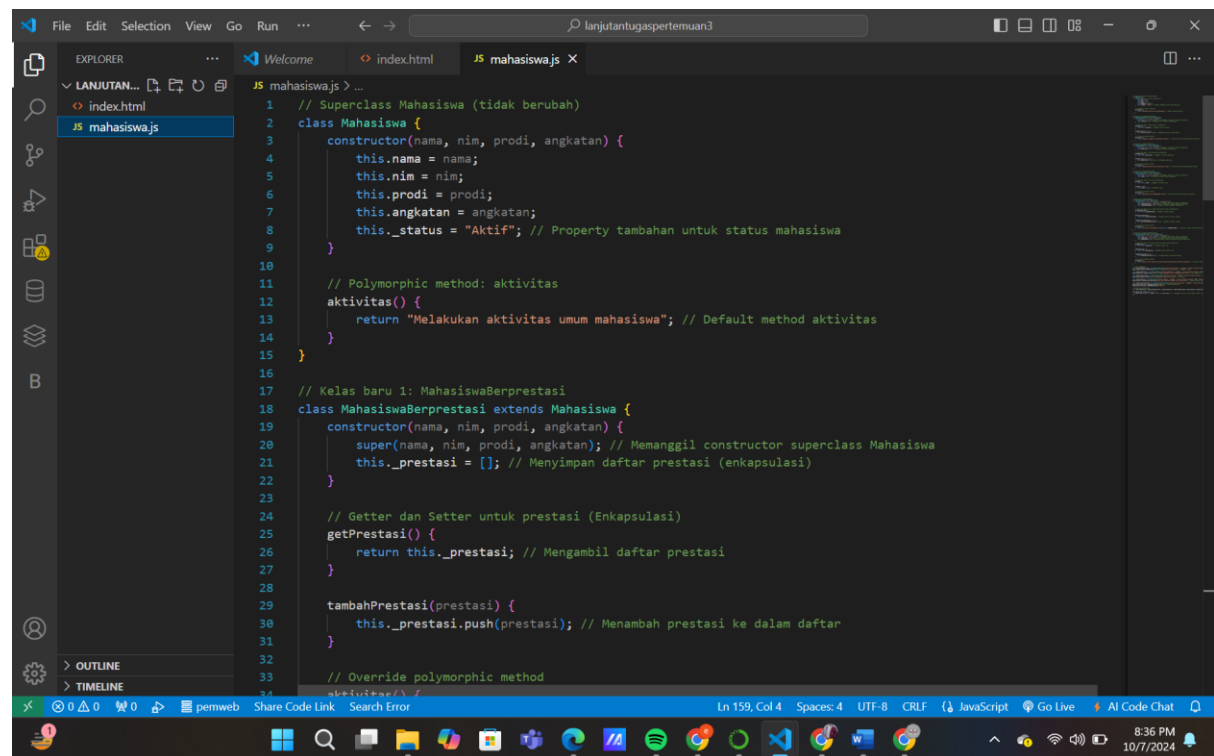
Disusun Oleh :  
Elsa Syaila Arzetti  
2301436

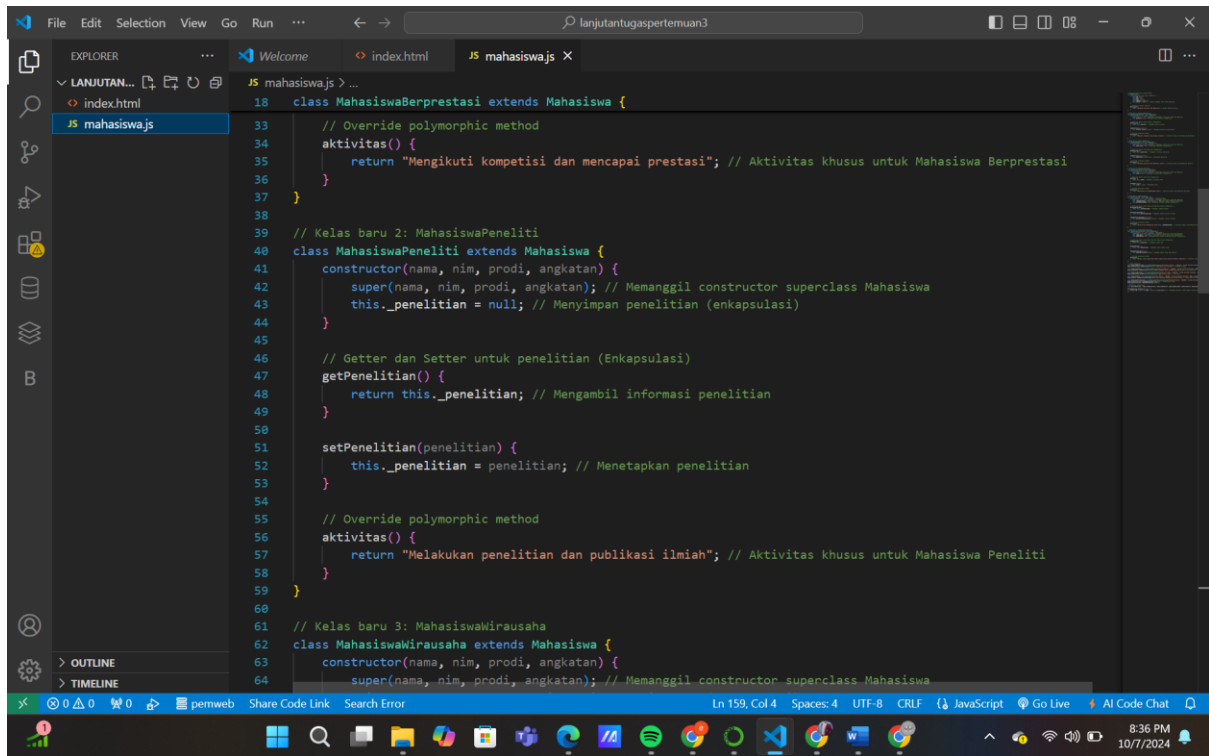
**PROGRAM STUDI SISTEM INFORMASI KELAUTAN**  
**UNIVERSITAS PENDIDIKAN INDONESIA**  
**2024**

## Penjelasan Mengenai Tugas :



Untuk Penjelasan kodingan sendiri sudah diberikan penjelasan di kodingannya langsung.





The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'LANJUTAN...' containing 'index.html' and 'JS mahasiswajs'. The main editor window shows the 'JS mahasiswajs' file with the following code:

```
61 // Kelas baru 3: Mahasiswawirausaha
62 class Mahasiswawirausaha extends Mahasiswa {
63     constructor(nama, nim, prodi, angkatan) {
64         super(nama, nim, prodi, angkatan); // Memanggil constructor superclass Mahasiswa
65         this._usaha = null; // Menyimpan informasi usaha (enkapsulasi)
66     }
67
68     // Getter dan Setter untuk usaha (Enkapsulasi)
69     getUsaha() {
70         return this._usaha; // Mengambil informasi usaha
71     }
72
73     setUsaha(usaha) {
74         this._usaha = usaha; // Menetapkan usaha
75     }
76
77     // Override polymorphic method
78     aktivitas() {
79         return "Menjalankan dan mengembangkan usaha"; // Aktivitas khusus untuk Mahasiswa Wirausaha
80     }
81 }
82
83 // Kelas baru 4: MahasiswaAtlet
84 class MahasiswaAtlet extends Mahasiswa {
85     constructor(nama, nim, prodi, angkatan, cabangOlahraga) {
86         super(nama, nim, prodi, angkatan); // Memanggil constructor superclass Mahasiswa
87         this._cabangOlahraga = cabangOlahraga; // Menyimpan cabang olahraga (enkapsulasi)
88         this._prestasiOlahraga = []; // Menyimpan prestasi olahraga (enkapsulasi)
89     }
90
91     // Getter dan Setter untuk cabang olahraga dan prestasi (Enkapsulasi)
92     getCabangOlahraga() {
93         return this._cabangOlahraga; // Mengambil cabang olahraga
```

The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'LANJUTAN...' containing 'index.html' and 'JS mahasiswajs'. The main editor window shows the 'JS mahasiswajs' file with the following code:

```
84 class MahasiswaAtlet extends Mahasiswa {
85
86     // Getter dan Setter untuk cabang olahraga dan prestasi (Enkapsulasi)
87     getCabangOlahraga() {
88         return this._cabangOlahraga; // Mengambil cabang olahraga
89     }
90
91     getPrestasiOlahraga() {
92         return this._prestasiOlahraga; // Mengambil daftar prestasi olahraga
93     }
94
95     tambahPrestasiOlahraga(prestasi) {
96         this._prestasiOlahraga.push(prestasi); // Menambah prestasi olahraga
97     }
98
99     // Override polymorphic method
100     aktivitas() {
101         return 'Berlatih dan berkompetisi dalam ${this._cabangOlahraga}'; // Aktivitas khusus untuk Mahasiswa Atlet
102     }
103 }
104
105 // Kelas baru 5: MahasiswaInternasional
106 class MahasiswaInternasional extends Mahasiswa {
107     constructor(nama, nim, prodi, angkatan, negaraAsal) {
108         super(nama, nim, prodi, angkatan); // Memanggil constructor superclass Mahasiswa
109         this._negaraAsal = negaraAsal; // Menyimpan informasi negara asal (enkapsulasi)
110         this._bahasaAsing = []; // Menyimpan daftar bahasa asing yang dikuasai (enkapsulasi)
111     }
112
113     // Getter dan Setter untuk negara asal dan bahasa asing (Enkapsulasi)
114     getNegaraAsal() {
115         return this._negaraAsal; // Mengambil negara asal
116     }
117
118     // Override polymorphic method
119     aktivitas() {
120         return 'Berlatih dan berkompetisi dalam ${this._negaraAsal}'; // Aktivitas khusus untuk Mahasiswa Internasional
121     }
122 }
```

```
111 MahasiswaInternasional extends Mahasiswa {
112   tambahBahasaAsing() {
113     return this._bahasaAsing; // Mengambil daftar bahasa asing
114   }
115 }
116
117 tambahBahasaAsing(bahasa) {
118   this._bahasaAsing.push(bahasa); // Menambah bahasa asing yang dikuasai
119 }
120
121 // Override polymorphic method
122 aktivitas() {
123   return "Belajar lintas budaya dan berpartisipasi dalam program pertukaran mahasiswa"; // Aktivitas khusus untuk
124 }
125
126 // Contoh penggunaan
127 mahasiswaBerprestasi = new MahasiswaBerprestasi("Elsa Syaila Arzetti", "2301436", "Sistem Informasi Kelautan", 202
128 mahasiswaBerprestasi.tambahPrestasi("Juara 1 Lomba Karya Tulis Ilmiah"); // Menambah prestasi
129
130 mahasiswaPeneliti = new MahasiswaPeneliti("Deswita Maharani", "2301002", "Sistem Informasi Kelautan", 2023);
131 mahasiswaPeneliti.setPenelitian("Analisis Dampak Perubahan Iklim terhadap Ekosistem Laut"); // Set penelitian
132
133 mahasiswaWirausaha = new MahasiswaWirausaha("Rizka Azlia Yumna", "2301003", "Sistem Informasi Kelautan", 2023);
134 mahasiswaWirausaha.setUsaha("Startup Pengolahan Limbah Plastik Laut"); // Set usaha
135
136 mahasiswaAtlet = new MahasiswaAtlet("Shofia Nabila", "2301004", "Sistem Informasi Kelautan", 2023, "Renang");
137 mahasiswaAtlet.tambahPrestasiOlahraga("Medali Emas Kejuaraan Renang Antar Universitas"); // Tambah prestasi olahraga
138
139 mahasiswaInternasional = new MahasiswaInternasional("Aisyah Pasha Nurfilailly", "2301005", "Sistem Informasi Kelaut
140 mahasiswaInternasional.tambahBahasaAsing("Inggris"); // Tambah bahasa asing
141 mahasiswaInternasional.tambahBahasaAsing("Jerman");
142
143 // Demonstrasi polimorfisme
144 daftarMahasiswa = [mahasiswaBerprestasi, mahasiswaPeneliti, mahasiswaWirausaha, mahasiswaAtlet, mahasiswaInternasi
145
146 console.log(daftarMahasiswa);
```

```
111 class MahasiswaInternasional extends Mahasiswa {
112   tambahBahasaAsing() {
113     return this._bahasaAsing; // Mengambil daftar bahasa asing
114   }
115 }
116
117 // Override polymorphic method
118 aktivitas() {
119   return "Belajar lintas budaya dan berpartisipasi dalam program pertukaran mahasiswa"; // Aktivitas khusus untuk
120 }
121
122 // Contoh penggunaan
123 const mahasiswaBerprestasi = new MahasiswaBerprestasi("Elsa Syaila Arzetti", "2301436", "Sistem Informasi Kelautan", 202
124 mahasiswaBerprestasi.tambahPrestasi("Juara 1 Lomba Karya Tulis Ilmiah"); // Menambah prestasi
125
126 const mahasiswaPeneliti = new MahasiswaPeneliti("Deswita Maharani", "2301002", "Sistem Informasi Kelautan", 2023);
127 mahasiswaPeneliti.setPenelitian("Analisis Dampak Perubahan Iklim terhadap Ekosistem Laut"); // Set penelitian
128
129 const mahasiswaWirausaha = new MahasiswaWirausaha("Rizka Azlia Yumna", "2301003", "Sistem Informasi Kelautan", 2023);
130 mahasiswaWirausaha.setUsaha("Startup Pengolahan Limbah Plastik Laut"); // Set usaha
131
132 const mahasiswaAtlet = new MahasiswaAtlet("Shofia Nabila", "2301004", "Sistem Informasi Kelautan", 2023, "Renang");
133 mahasiswaAtlet.tambahPrestasiOlahraga("Medali Emas Kejuaraan Renang Antar Universitas"); // Tambah prestasi olahraga
134
135 const mahasiswaInternasional = new MahasiswaInternasional("Aisyah Pasha Nurfilailly", "2301005", "Sistem Informasi
136 mahasiswaInternasional.tambahBahasaAsing("Inggris"); // Tambah bahasa asing
137 mahasiswaInternasional.tambahBahasaAsing("Jerman");
138
139 // Demonstrasi polimorfisme
140 const daftarMahasiswa = [mahasiswaBerprestasi, mahasiswaPeneliti, mahasiswaWirausaha, mahasiswaAtlet, mahasiswaIntern
141
142 daftarMahasiswa.forEach(mahasiswa => {
143   console.log(`${mahasiswa.nama}: ${mahasiswa.aktivitas()}`); // Memanggil aktivitas sesuai dengan subclass masing
144 });
```

## Untuk Perintah Utama ada :

1. Pembuatan objek: new MahasiswaBerprestasi(), new MahasiswaPeneliti(), dll.
2. Menambah data: tambahPrestasi(), setPenelitian(), setUsaha(), tambahPrestasiOlahraga(), tambahBahasaAsing()
3. Mengakses data: getPrestasi(), getPenelitian(), getUsaha(), getCabangOlahraga(), getPrestasiOlahraga(), getNegaraAsal(), getBahasaAsing()

4. Demonstrasi polimorfisme: Penggunaan `forEach` untuk memanggil `aktivitas()` pada setiap objek di dalam `daftarMahasiswa`

### **Enkapsulasi: Menyimpan Data dengan Aman**

- **Properti Private:** Dalam kodingan ini itu setiap mahasiswa memiliki data pribadi seperti prestasi, penelitian, atau cabang olahraga. Data-data ini kita simpan di dalam kelas masing-masing mahasiswa, dan kita tandai sebagai "private" atau "rahasia". Artinya, data ini hanya bisa diakses dari dalam kelas itu sendiri.
- **Getter dan Setter:** Untuk membaca atau mengubah data yang bersifat private. Untuk mengetahui prestasi seorang mahasiswa, kita memanggil metode `getPrestasi()`. Untuk mengubah prestasi, kita memanggil metode `tambahPrestasi()`.

### **Implementasi Polimorfisme :**

- **Metode Aktivitas():** Setiap jenis mahasiswa (mahasiswa berprestasi, peneliti, atlet, dll.) memiliki aktivitas yang berbeda. Kita membuat metode `aktivitas()` untuk mewakili aktivitas tersebut.
- **Override:** Setiap kelas turunan (misalnya, `MahasiswaBerprestasi`, `MahasiswaPeneliti`) akan memiliki implementasi metode `aktivitas()` yang berbeda. Artinya, ketika kita memanggil `aktivitas()` pada seorang mahasiswa berprestasi, yang akan dilakukan adalah kegiatan yang terkait dengan prestasi, sedangkan jika kita memanggil `aktivitas()` pada seorang mahasiswa peneliti, yang akan dilakukan adalah kegiatan yang terkait dengan penelitian.