# Week 1
## Basic Programming
## Web Development

BLEND

# Code Editor

- Visual Studio Code

  **Visual Studio Code** adalah software code editor gratis buatan Microsoft yang bisa dijalankan di berbagai operating system pada perangkat desktop.

  Download : https://code.visualstudio.com/Download

# Code Editor

- Shortcut (File > Preference > Keyboard Shortcut)

1. CTRL + D

   Berfungsi untuk multiple select terhadap character yang sama.

2. CTRL + Tab

   Berfungsi untuk pindah tab file

3. Alt + Capslock + up / down

   Berfungsi untuk memindahkan blok kode keatas atau kebawah.

# Code Editor

4. CTRL + G

   Berfungsi untuk berpindah Line.

5. CTRL + P

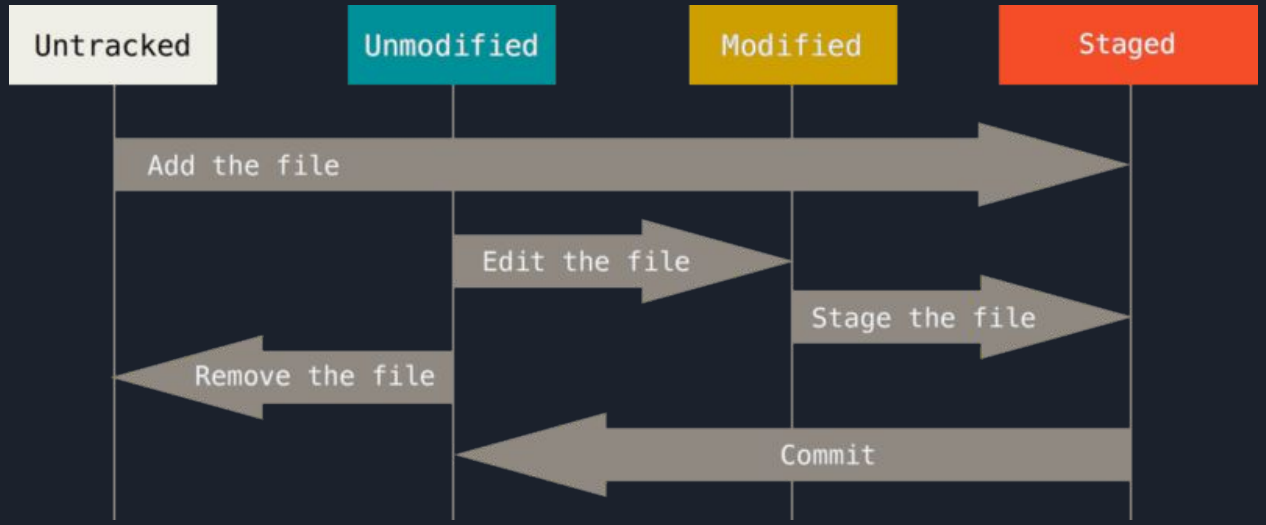
   Berfungsi untuk mencari nama File

# Code Editor

- Extension
  - ESLINT
  - JavaScript (ES6) Code Snippets
  - Live Server
  - Gitlens
  - HTML Boilerplate

# Code Version

- Installation (https://git-scm.com/downloads)



# Code Version

- Command
  - Git Config
    - git config --global user.name "Nama Anda"
    - git config --global user.email "email"
  - Git Setup
    - git init
    - git remote add origin https://github.com/user/repo.git
  - Git Basic
    - git pull origin namabranch
    - git add nama_file.js / git add .
    - git commit -m "Pesan komit Anda disini"
    - git push origin namabranch

BLEND

# Code Version

- Command
  - Git Branching
    - git branch namabranchbaru
    - git checkout namabranch
    - git merge namabranch
    - git fetch
  - Git Track
    - git status
    - git log

BLEND

# Node JS (JavaScript)

- Apa itu Node JS

Node.js adalah lingkungan runtime JavaScript yang dibangun di atas mesin JavaScript V8 dari Google Chrome. Ini memungkinkan menjalankan kode JavaScript di sisi server, bukan hanya di browser.

Node.js memiliki ekosistem yang besar dan aktif yang terdiri dari berbagai pustaka, modul, dan alat yang memungkinkan pengembangan aplikasi web, server, dan lainnya menggunakan JavaScript.

# Node JS (JavaScript)

- Fitur Node JS
  - Asynchronous Programming
  - Node Package Manager (NPM)
  - Server Side Development
- Installation (https://nodejs.org/dist/v16.16.0/node-v16.16.0-x64.msi)
- Running Javascript di Node JS
  - `node namafile.js`

# Node JS (JavaScript)

- Variabel JavaScript
  - Deklarasi Variabel

```
var name = "John";
let age = 25;
const PI = 3.14;
```

# Node JS (JavaScript)

- Conditional
  - If Statement

```
var age = 18;

if (age >= 18) {
  console.log("Anda sudah dewasa");
}
```

# Node JS (JavaScript)

- Conditional
  - If - else Statement

```
var age = 15;

if (age >= 18) {
  console.log("Anda sudah dewasa");
} else {
  console.log("Anda masih anak-anak");
}
```

# Node JS (JavaScript)

- Conditional
  - If - else if - else Statement

```
var score = 85;

if (score >= 90) {
  console.log("Anda mendapatkan nilai A");
} else if (score >= 80) {
  console.log("Anda mendapatkan nilai B");
} else if (score >= 70) {
  console.log("Anda mendapatkan nilai C");
} else {
  console.log("Anda mendapatkan nilai D");
}
```

BLEND

# Node JS (JavaScript)

- Conditional
  - Ternary Operator

```
var age = 20;
var message = age >= 18 ? "Anda sudah dewasa" : "Anda masih anak-anak";
console.log(message);
```

BLEND

# Node JS (JavaScript)

- Conditional
  - Switch Statement

```
var day = "Minggu";

switch (day) {
  case "Senin":
    console.log("Hari kerja");
    break;
  case "Selasa":
  case "Rabu":
  case "Kamis":
  case "Jumat":
    console.log("Hari kerja");
    break;
  case "Sabtu":
  case "Minggu":
    console.log("Hari libur");
    break;
  default:
    console.log("Hari tidak valid");
}
```

BLEND

# Node JS (JavaScript)

- Looping
  - For Loop

```
for (var i = 0; i < 5; i++) {
  console.log(i);
}
```

  - While Loop

```
var i = 0;
while (i < 5) {
  console.log(i);
  i++;
}
```

BLEND

# Node JS (JavaScript)

- Looping
  - Do - While Loop

```
var i = 0;
do {
  console.log(i);
  i++;
} while (i < 5);
```

  - For .. of Loop

```
var colors = ["merah", "hijau", "biru"];
for (var color of colors) {
  console.log(color);
}
```

BLEND

# Node JS (JavaScript)

- Function
  - Mendefinisikan Function

```
function greet(name) {
  return "Halo, " + name + "!";
}
```

# Node JS (JavaScript)

- Logic Operator
  - AND (&&)

```
var isSunny = true;
var isWarm = true;
var isNiceWeather = isSunny && isWarm; // true
```

  - OR (||)

```
var hasMoney = false;
var hasCreditCard = true;
var canBuyItem = hasMoney || hasCreditCard; // true
```

  - Not (!)

```
var isRaining = true;
var isNotRaining = !isRaining; // false
```

BLEND

# Node JS (JavaScript)

- Logic Operator
  - Perbandingan

```
var age = 25;
var isAdult = age >= 18; // true
```

# Node JS (JavaScript)

- Array manipulation
  - Push & Unshift

```
var fruits = ['apel', 'pisang'];
fruits.push('mangga');     // ['apel', 'pisang', 'mangga']
fruits.unshift('jeruk');   // ['jeruk', 'apel', 'pisang', 'mangga']
```

  - Pop & Shift

```
var numbers = [1, 2, 3, 4];
numbers.pop();      // [1, 2, 3]
numbers.shift();    // [2, 3]
```

  - Splice

```
var colors = ['merah', 'hijau', 'biru'];
colors.splice(1, 1);     // Menghapus elemen kedua ("hijau")
colors.splice(0, 1, 'kuning'); // Mengganti elemen pertama dengan "kuning"
```

BLEND

# Node JS (JavaScript)

- Array manipulation
  - Concat & Join

```
var arr1 = [1, 2];
var arr2 = [3, 4];
var combined = arr1.concat(arr2);  // [1, 2, 3, 4]

var fruits = ['apel', 'pisang', 'mangga'];
var fruitString = fruits.join(', '); // "apel, pisang, mangga"
```

  - Filter & Map

```
const numbers = [1, 2, 3, 4, 5];
const evenNumbers = numbers.filter(num => num % 2 === 0); // [2, 4]

const squaredNumbers = numbers.map(num => num * num); // [1, 4, 9, 16, 25]
```

BLEND

# Node JS (JavaScript)

- Object Oriented Programming (OOP)
  - Constructor

```
class Animal {
  constructor(name) {
    this.name = name;
  }

  speak() {
    console.log(`${this.name} mengeluarkan suara.`);
  }
}

const sound = new Animal("Anjing");
sound.speak(); //Anjing mengeluarkan suara
```

BLEND

# Node JS (JavaScript)

- Object Oriented Programming (OOP)
  - Extends

```
class Animal {
  constructor(name) {
    this.name = name;
  }

  speak() {
    console.log(`${this.name} mengeluarkan suara.`);
  }
}

class Dog extends Animal {
  constructor(name, breed) {
    super(name);
    this.breed = breed;
  }

  speak() {
    console.log(`${this.name} (ras ${this.breed}) menggonggong.`);
  }
}

const dog = new Dog("Buddy", "Golden Retriever");
dog.speak(); // Output: Buddy (ras Golden Retriever) menggonggong.
```

BLEND

# Node JS (JavaScript)

- Object Oriented Programming (OOP)
  - Public

```
class Person {
  constructor(name) {
    this.name = name; // Variabel publik
  }

  sayHello() {
    console.log(`Halo, nama saya ${this.name}`);
  }
}

const person = new Person("John");
console.log(person.name); // Output: John
person.sayHello(); // Output: Halo, nama saya John
```

BLEND

# Node JS (JavaScript)

- Object Oriented Programming (OOP)
  - Private

```
class Counter {
  #count = 0; // Variabel privat

  increment() {
    this.#count++;
  }

  getCount() {
    return this.#count;
  }
}

const counter = new Counter();
counter.increment();
console.log(counter.getCount()); // Output: 1
console.log(counter.#count);
```

```
Property '#count' is not accessible outside class 'Counter' because it has a private identifier.
View Problem (⌥F8)    No quick fixes available
```

# Node JS (JavaScript)

- Asynchronous
  - Promise - Chaining

```
function fetchData() {
  return new Promise((resolve, reject) => {
    setTimeout(() => {
      const data = 'Data dari server';
      resolve(data);
    }, 1000);
  });
}

fetchData()
  .then((result) => {
    console.log(result); // Output: Data dari server
  })
  .catch((error) => {
    console.error(error);
  });
```

BLEND

# Node JS (JavaScript)

- Asynchronous
  - Async - Await

```
async function fetchData() {
  return new Promise((resolve) => {
    setTimeout(() => {
      const data = 'Data dari server';
      resolve(data);
    }, 1000);
  });
}

async function main() {
  try {
    const result = await fetchData();
    console.log(result); // Output: Data dari server
  } catch (error) {
    console.error(error);
  }
}

main();
```

BLEND

# Tugas

1. Find Maximum Number
   a. Input : arr = [2, 3, 10, 6, 4, 8, 1]
   b. Output : 10
2. Compress string
   a. input – ‘aaaaabbbbcccccccdaa’
   b. Output – ‘a5b4c7da2’
3. Sort Array
   a. input : arr = [2, 3, 10, 6, 4, 8, 1]
   b. Output : arr = [1, 2, 3, 4, 6, 8, 10]
4. Find odd and even number
   a. input : arr = [1, 2, 3, 4, 5, 6, 7, 8, 9]
   b. Output : arr = [‘ganjil’, 2, ‘ganjil’, 4, ‘ganjil’, 6, ‘ganjil’, 8, ‘ganjil’]
5. Reverse String
   a. Input : ‘abcdefg’
   b. Output : ‘gfedcba’
6. Create file on based on test number (ex: 1.js, 2.js), and push to repository Github.

BLEND