

The screenshot shows a code editor with a dark theme. In the top left, there are tabs for various files: perbandingan.js, index.js, perjumlahan.js, pengurangan.js, pembagian.js, perkalian.js, package.json, objectLiteral.js, and objkLiteral.js. The objkLiteral.js tab is active. The code in the editor is:

```

week_04 > objkLiteral.js >-
1 const putri = {
2   "name": "Putri",
3   "birthday": "2005-02-13",
4   "phone": "081234567898",
5   "address": {
6     "primary": "Bandung",
7     "secondary": "Sukabumi"
8   }
9 }
10
11 const bayu = Object.assign({}, putri);
12 bayu.name = "Bayu"
13 bayu.address.primary = "Aceh";
14
15 console.log(putri.name);
16 console.log(putri.address.primary);
17
18
19

```

Below the editor is a terminal window titled "powershell - week\_04". The terminal output is:

```

PS C:\Users\Axioo Hype 7\OneDrive\Documents\js_learning> ls

Directory: C:\Users\Axioo Hype 7\OneDrive\Documents\js_learning

Mode                LastWriteTime       Length Name
--<----->
da--<1>    9/22/2025 10:39 AM          0 week_01
da--<1>    9/22/2025 9:29 PM          0 week_02
da--<1>    10/6/2025 8:59 AM          0 week_03
da--<1>    10/14/2025 9:15 AM          0 week_04
da--<1>    10/13/2025 8:13 PM          0 week_05

PS C:\Users\Axioo Hype 7\OneDrive\Documents\js_learning> cd week_04
PS C:\Users\Axioo Hype 7\OneDrive\Documents\js_learning\week_04> node objkLiteral.js
Putri
Aceh

```

A tooltip from Docker is visible in the center-right of the terminal area.

The reason why Putri's primary address changed to Aceh is because the code uses **Object.assign()**, which performs a **shallow copy**. This means that first-level properties such as *name*, *birthday*, and *phone* are copied by value, but properties that are objects within objects, like *address*, are copied by reference to the same object in memory. As a result, **bayu.address** and **putri.address** point to the same object. Therefore, when **bayu.address.primary** is changed to "Aceh", that change also affects **putri.address.primary**, since both actually share the same address object reference.

The screenshot shows a code editor with a dark theme. In the top left, there are tabs for various files: perbandingan.js, index.js, perjumlahan.js, pengurangan.js, pembagian.js, perkalian.js, package.json, objectLiteral.js, and objkLiteral.js. The objkLiteral.js tab is active. The code in the editor is identical to the one in the previous screenshot:

```

week_04 > objkLiteral.js >-
1 const putri = {
2   "name": "Putri",
3   "birthday": "2005-02-13",
4   "phone": "081234567898",
5   "address": {
6     "primary": "Bandung",
7     "secondary": "Sukabumi"
8   }
9 }
10
11 const bayu = structuredClone(putri);
12 bayu.name = "Bayu";
13 bayu.address.primary = "Aceh";
14
15 console.log(putri.name);
16 console.log(putri.address.primary);
17
18
19

```

Below the editor is a terminal window titled "powershell - week\_04". The terminal output is:

```

PS C:\Users\Axioo Hype 7\OneDrive\Documents\js_learning> node objkLiteral.js
Putri
Banding

```

If we want to fully copy the data so that Bayu and Putri each have their own separate `address` objects (not connected to each other), we need to create a **deep copy**. This can be done using `structuredClone(putri)` or `JSON.parse(JSON.stringify(putri))`. That way, when `bayu.address.primary` is changed to "Aceh", the value of `putri.address.primary` will remain "Bandung" (as shown in the picture above).