1. Create handleSubstraction
   1. Create arrow function

const handleSubtractCount = (*index*) => {

const newTodos = [...todos];

newTodos[index].count -= 1;

setTodos(newTodos);

}

* 1. Add onClick on the button

<button

*lassName*="todoActionButton"

*onClick*={() => handleSubtractCount(*index*)}

                    >

                      <img *src*={minusIcon} *alt*="minus icon" />

                    </button>

* 1. To manage if < 0 make it disappear

Catatan tabahan:

Untuk Form 🡪 onSubmit (tambahkan e.preventDefault() saat di dalam func)

Untuk Input 🡪 onChange

Untuk Button 🡪 onClick

const handleSubtractCount = (*index*) => {

const newTodos = [...todos];

if( newTodos.count > 0 ) {

newTodos[index].count -= 1;

setTodos(newTodos);

} else {

newTodos.splice(index, 1)

setTodos(newTodos);

}

};

1. Create function for add todos bar.

* Create func handleSubmit
* const handleSubmit = (*e*) => {
* *e*.preventDefault();
* const addedTodos = [
* ...todos,
* {
* title: value,
* count: 1,
* },
* ];
* setTodos(addedTodos);
* };
* Add onSubmit in Form tag

<form *className*="form" *onSubmit*={handleSubmit}>

* Input can not be empty
* const handleSubmit = (*e*) => {
* *e*.preventDefault();
* if (!value) {
* alert("Input cannot be empty");
* return;
* }
* const addedTodos = [
* ...todos,
* {
* title: value,
* count: 1,
* },
* ];
* setTodos(addedTodos);
* *//to make input clear again when we after click Add button*
* setValue("");
* };

1. Create Info Section (video 23)

</form>

        <div *className*="info">

          <div *className*="info-total">Total List: {………}</div>

          <div *className*="info-total">Total Count: {……….}</div>

          <div *className*="deleteAllButton" *onClick*={() => setTodos([])}>

            Delete All

          </div>

        </div>

1. Make it works in info section

const getTotalCount = () => {

    const total = todos.reduce((*a*, *b*) => {

      return *a* + *b*.count;

    }, 0);

    return total;

  };

<div *className*="info">

          <div *className*="infoTotal">Total List: {todos.length}</div>

          <div *className*="infoTotal">{`Total Count: ${getTotalCount()}`}</div>

          <div *className*="deleteAllButton">Delete All</div>

        </div>

1. Create components Navbar

* Create folder Components and Navbar
* Add file index.js for Navbar.
* import shoppingLogo from "../../assets/shopping-icon.svg";
* const Navbar = () => {
* return (
* <nav *className*="nav">
* <img *src*={shoppingLogo} *className*="logo" *alt*="Shopping logo" />
* <h1 *className*="nav-item">Daftar BelanjaKu</h1>
* </nav>
* );
* };
* export default Navbar;

in App.jsx

import Navbar from "./components/Navbar";

….

return (

    <>

      <Navbar />

….

1. Create component container
   1. Create folder Container
   2. Add file index.js for Container.

import PropTypes from "prop-types";

function Container(*props*) {

  return <section *className*="container">{*props*.children}</section>;

}

Container.propTypes = {

children: PropTypes.node,

};

export default Container;

in App.jsx

import Container from "./components/Container";

<Container>

        <form *className*="form" *onSubmit*={handleSubmit}>

          <input

…………..

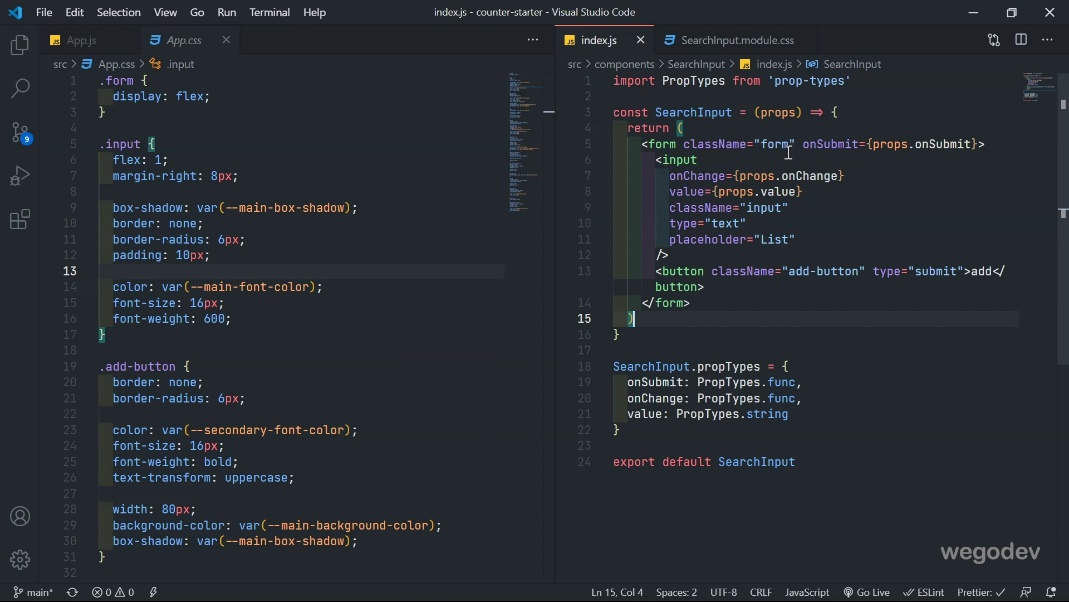
1. Membuat CSS Module

* Create file for Navbar 🡪 Navbar.module.css
  + Di index.js Navbar
* import styles from "./Navbar.module.css";
  + Disesuaikan className nya seperti ini
* <nav *className*={styles.nav}>

*Begitu juga untuk component Container*

1. Membuat components untuk Info, SearchIput, Todos, Empty

Parsing function in App.jsx and SearchInput.jsx make props and prop-types.



1. Props destructing. (we are not use props again but we can parsing the name of the parameter)

function Info({ *todosLength*, *getTotalCount*, *onClick* }) {

  return (

    <div *className*={styles.info}>

      <div *className*={styles.infoTotal}>Total List: {*todosLength*}</div>

      <div *className*={styles.infoTotal}>{`Total Pcs: ${*getTotalCount*}`}</div>

      <div *className*={styles.deleteAllButton} *onClick*={*onClick*}>

        Hapus Semua

      </div>

    </div>

  );

}