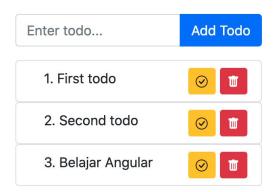
Practice: + Membuat Todo App

Sekarang, kita akan mencoba membuat aplikasi todo dengan Angular. Nantinya dalam aplikasi ini terdapat fitur:

- Menambahkan todo
- Menghapus todo
- Menyelesaikan todo





- Buatlah aplikasi angular baru dengan command ng new nama_aplikasi
- Kemudian install bootstrap dan bootstrap-icon untuk keperluan styling dengan command

npm install bootstrap bootstrap-icons

 Masukkan konfigurasi ke file angular.json dan styles.css agar bisa menggunakan style dan icon dalam aplikasi Angular.

```
"styles": [
   "node_modules/bootstrap/dist/css/bootstrap.min.css",
   "src/styles.css"
],
```

angular.json

```
src > 3 styles.css M ×

src > 3 styles.css

1 /* You can add global styles to this file, and also import "~bootstrap-icons/font/bootstrap-icons.css";
```

styles.css



Selanjutnya, generate beberapa component. Disini akan diletakkan dalam folder components.

```
ng generate component components/navbar
ng generate component components/todo-list
ng generate component components/add-todo-form
```

Ubah template app.component dengan menggunakan class/style dari bootstrap



Isi template navbar komponen dengan menggunakan komponen bootstrap yang disediakan di: https://getbootstrap.com/docs/5.1/components/navbar/



Buatlah 1 interface Todo yang mempunyai 2 properti yaitu content dan completed.

Disini diletakkan dalam folder src/app/models

```
TS Todo.ts U X

src > app > models > TS Todo.ts > •• Todo

1    export interface Todo {
2    content:string;
3    completed:boolean;
4 }
```



Todo App - List

```
import { Component, OnInit } from '@angular/core';
import { Todo } from '../../models/todo';
@Component({
  selector: 'app-todo-list',
 templateUrl: './todo-list.component.html',
 styleUrls: ['./todo-list.component.css']
export class TodoListComponent implements OnInit {
  todos: Todo[]=[];
  ngOnInit(): void {
    this.todos = [
        content: 'First todo',
        completed: false
      },
        content: 'Second todo',
        completed: false
```

Sekarang kita akan menyiapkan todo list. Pada file **todo-list.component.ts**, siapkan properti todos dengan tiper Todo[].

Dan isi todos dengan list todo di hooks ngOnInit.



Todo App - List

```
<div class="ms-3 me-auto">
   <div class="content {{(todo.completed ? 'done' : '')}}">
      {{i + 1}}, {{ todo.content }}
   </div>
  </div>
  <button class="btn btn-warning btn-sm completed">
   <i class="bi bi-check-circle"></i>
  </button>
  <button class="btn btn-danger btn-sm">
   <i class="bi bi-trash-fill"></i>
  </button>
```

1. First todo



2. Second todo





Isi template di todo-list. Disini menggunakan list-group dari bootstrap.

https://getbootstrap.com/docs/5.1/components/lis t-group/.

Gunakan directive ngFor untuk melakukan perulangan todo.

Sampai sini, coba jalankan aplikasi dan lihatlah apakah todo list sudah berhasil muncul atau belum.



Todo App - Delete & Completed

```
toggleDone (id:number) {
   this.todos.map((v, i) => {
      if (i == id) v.completed = !v.completed;
      console.log(v)
      return v;
   })
}

deleteTodo (id:number) {
   this.todos = this.todos.filter((v, i) => i !== id);
}
```

Di file todo-list.component.ts,

Tambahkan 2 method ini, untuk mengubah status completed todo dan menghapus todo.



Todo App - Delete & Completed

Kembali ke file template html todo-list, pasang event bind click ke method **toggleDone** dan **deleteTodo** yang menerima variable i.

Variable i berasal dari ngFor yang berisi index (index ngFor selalu dimulai dari 0).



Todo App - Delete & Completed

Agar action completed terlihat, pasang css line-through, dan masukkan kondisi di bagian todo content pada file template.





```
import { FormsModule } from '@angular/forms';

@NgModule({
   declarations: [...
   ],
   imports: [
     BrowserModule,
     FormsModule
   ],
   providers: [],
   bootstrap: [AppComponent]
})
```

Sekarang fitur add todo. Akan menggunakan two way binding ngModel,

Maka di file **app.module.ts,** import FormsModule dan masukkan ke metadata imports dalam @NgModule



```
addTodo(todo: Todo) {
   this.todos.push(todo);
}

todo-list.component.ts
```

Tambahkan method **addTodo** di todo-list yang mengepush todo ke properti todos

Di file **todo-list.component.html**, pasang komponen app-add-todo-form

Dan siapkan binding event **newTodoEvent** yang mengeksekusi **addTodo(\$event)** yang dibuat sebelumnya.



```
A add-todo-form.component.ts U X
src > app > components > add-todo-form > 🔼 add-todo-form.component.ts > ...
        import { Component, EventEmitter, Output } from '@angular/core';
       import { Todo } from '../../models/Todo'
       @Component({
         selector: 'app-add-todo-form',
         templateUrl: './add-todo-form.component.html',
         styleUrls: ['./add-todo-form.component.css']
       export class AddTodoFormComponent {
         @Output() newTodoEvent = new EventEmitter<Todo>();
         inputTodo:string = "";
         addTodo () {
           const todo: Todo = {
             content: this.inputTodo,
             completed: false
            this.newTodoEvent.emit(todo)
            this.inputTodo = "";
```

Siapkan

- . properti **inputTodo**,
- decorator @Output dengan event bernama newTodoEvent
- method addTodo() yang berisi pembuatan todo baru berdasar inputTodo dan mengemit/mengirim todo baru ke parent.



```
    add-todo-form.component.html ∪ ×

src > app > components > add-todo-form > 🥫 ad
       <form>
          <div class="input-group mb-3">
           <input
              type="text"
             name="inputTodo"
             placeholder="Enter todo..."
              class="form-control"
              [(ngModel)]="inputTodo"
           />
           <input
              type="submit"
              value="Add Todo"
              class="btn btn-primary"
              (click)="addTodo()"
           />
          </div>
       </form>
```

Di template add-todo-form,

Buat form yang berisi input text dan input submit
button.

- Input text mempunyai binding ngModel ke inputTodo,
- Input Button mempunyai binding event click ke method addTodo() di add-todo-form.component.ts

