Node Package Manager

Dalam pengembangan aplikasi saat ini, industri gencar memanfaatkan module atau package luar agar pengembangan dapat lebih cepat. Semakin kompleks aplikasi tersebut, semakin banyak pula module/package yang digunakan. Di sinilah kita memerlukan sebuah package Manager.

Node Package Manager (NPM) merupakan pengelola package untuk JavaScript yang dapat memudahkan kita dalam mengelola package yang tersedia pada https://www.npmjs.com/. NPM merupakan standard package manager yang disediakan oleh Node.js dan sudah otomatis terpasang ketika memasang Node.js pada komputer kita. NPM dapat dioperasikan melalui CMD atau Terminal, Anda pun sudah mencobanya ketika membuat proyek JavaScript.

Selain untuk membuat proyek JavaScript, NPM dapat digunakan untuk memasang atau menghapus *third party module* (modul pihak ketiga). Modul yang dipasang melalui NPM akan disimpan pada folder **node_modules**.

Terdapat dua tipe pemasangan modul melalui NPM: yakni global dan lokal. Bila modul dipasang secara global, maka modul tersebut akan bersifat layaknya core module dan dapat digunakan di mana pun. Sedangkan modul yang dipasang secara lokal hanya dapat digunakan pada cakupan project Node.js yang memasangnya saja.

Namun saat ini, kami sangat menyarankan Anda untuk memasang modul

pihak ketiga secara lokal saja. Hindari pemasangan modul secara global karena akan menyebabkan banyak masalah. Sebaiknya gunakan npx bila Anda ingin menjalankan Node.js package di mana pun Anda inginkan.

MomentJS merupakan salah satu modul pihak ketiga yang populer untuk mengelola waktu di Node.js. Untuk memasangnya secara lokal, jalankan perintah berikut pada Terminal di project Node.js Anda.

```
npm install moment
```

Setelah pemasangan selesai, Anda bisa menggunakan module moment pada proyek Node.js Anda.

```
const moment = require('moment');

const date = moment().format("MMM Do YY");
console.log(date);

/**
 * output:
 * Jan 11th 21
 */
```

Package yang dipasang secara lokal melalui NPM akan tercatat di dalam berkas **package.json**, lebih tepatnya pada objek `dependencies`.

```
{
  "name": "nodejs-basic",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "moment": "^2.29.1"
  }
}
```

Ini menunjukkan bahwa proyek Node.js Anda memiliki ketergantungan atau dependencies terhadap module `moment`. Informasi ini berguna bila Anda hendak membagikan proyek Node.js ke orang lain. Mereka akan mengetahui modul pihak ketiga apa yang akan diinstal ketika memasang proyek Anda melalui perintah `npm install`.

Untuk menghapus modul pihak ketiga, Anda bisa gunakan perintah `npm uninstall <package-name>`. Bila Anda ingin menghapus modul moment, maka tuliskan:

```
npm uninstall moment
```

Terakhir, NPM juga bisa berfungsi sebagai runner script. Ia dapat

menjalankan script yang dituliskan pada objek `scripts` yang ada di berkas `package.json`. Dengan menetapkan script pada package.json, Anda dapat membuat jalan pintas untuk menjalankan Node.js process. Selain itu, Anda bisa membuat lebih dari satu script sesuai dengan environment yang Anda inginkan.

```
1.  "scripts": {
2.      "start-dev": "NODE_ENV=development node app.js",
3.      "start": "NODE_ENV=production node app.js"
4.    }
```

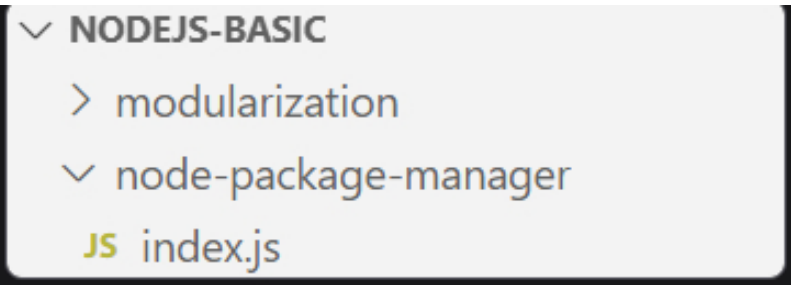
Untuk menjalankan script, Anda cukup menggunakan perintah `npm run <nama-script>`. Berdasarkan contoh di atas, jika Anda ingin menjalankan di environment development maka tuliskan perintah:

```
1.  npm run start-dev
```

Latihan: Node Package Manager

Kini Anda sudah mengenal bagaimana cara memasang module pihak ketiga melalui NPM. Saatnya latihan!

Pada latihan kali ini kita akan mencoba memasang module lodash melalui NPM. Namun sebelum itu, buat dulu folder baru bernama **node-package-manager** dan berkas index.js di dalamnya.



Pada berkas **index.js**, tuliskan starter code berikut:

```
const _ = // TODO

const myOddEvenArray = _.partition([1, 2, 3, 4, 5, 6], (n) => n % 2);

console.log(myOddEvenArray);
```

Tugas Anda ialah:

- Pasang package lodash pada proyek **nodejs-basic**.
- Gunakan package lodash pada TODO sehingga **index.js** dapat dieksekusi dengan baik.

Bila Anda telah mengerjakan semuanya dengan benar, eksekusi berkas **index.js** dengan perintah:

```
node ./node-package-manager/index.js
```

Maka console akan menghasilkan output berikut:

```
C:\javascript-projects\nodejs-basic>node ./node-package-manager/index.js
[ [ 1, 3, 5 ], [ 2, 4, 6 ] ]

C:\javascript-projects\nodejs-basic>
```

> **Mengalami kesulitan dalam menyelesaikan latihan?**
> Cobalah untuk ulas kembali materi yang diberikan atau tanyakan kesulitan yang Anda alami pada forum diskusi. Hindari melihat atau membandingkan kode solusi pada latihan Node Package Manager. sebelum Anda mencobanya sendiri.
