**STEP BY STEP ON USING NODE JS**

Download and install Node JS to computer, setelah itu kita cek apakah sudah terinstall atau belum

1. Cek apakah Node JS dan NPM sudah terinstall

cmd :

*node -v //untuk cek apakah NodeJS sudah terinstall serta versinya*

*npm -v //untuk cek apakah NPM sudah terinstall serta versinya*

Ada juga cara untuk membantu kita dalam melihat command np mini secara keseluruhan, yaitu dengan command :

cmd :

*npm help*

1. Install Package yang akan disimpan pada dependency utama

Dependency adalah daftar library eksternal (beserta versinya) yang digunakan oleh sebuah proyek, dan disimpan serta dikelola oleh npm.

Pertama buka Google dan search NPM, setelah itu click search disana dan cari nama package yang kalian mau. Setelah mendapatkan package yang diperlukan, kemudian copas cmd yang ditunjukan untuk instalisasi package tersebut seperti gambar berikut.

A screenshot of a computer

AI-generated content may be incorrect.

Sebelum kita melakukan install package package yang diperlukan, sebaiknya kita memastikan pada command prompt apakah sudah pada directory yang sudah benar. Maka dari itu kita pastikan bahwa posisi CMD sudah berada pada posisi yang benar, contohnya:

*PS D:\1. Progect\_Program\3. Full Stack Developer\Training\4-NodeJSTraining\Training01>*

Kode diatas merupakan contoh bahwa posisi CMD sudah berada di posisi yang benar. Jika ingin mengeceknya kita bisa menggunakan command:

cmd :

*ls*

Command diatas untuk mengecek apakah directory yang kita pilih pada status terminal sudah benar atau tidak. Karena sudah berada dalam folder progrect yang diingin kan (yaitu pada contohnya (Training01). Baru setelah itu kita jalankan command untuk install packagenya :

cmd :

*npm i moment*

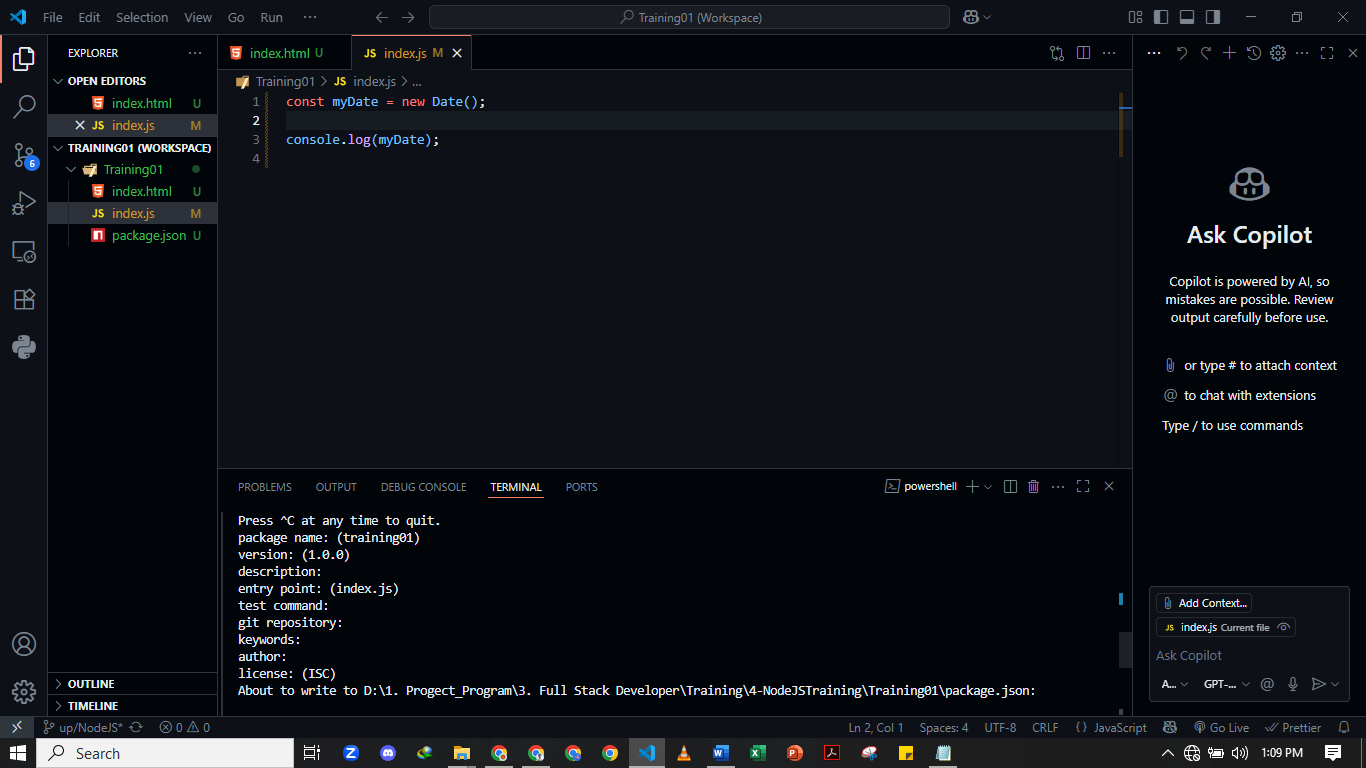
1. Membuat Package.json dan Install package npm pada project kita

Sekarang kita akan melakukan package manager dengan membuat package.json pada project kita, pertama dengan command :

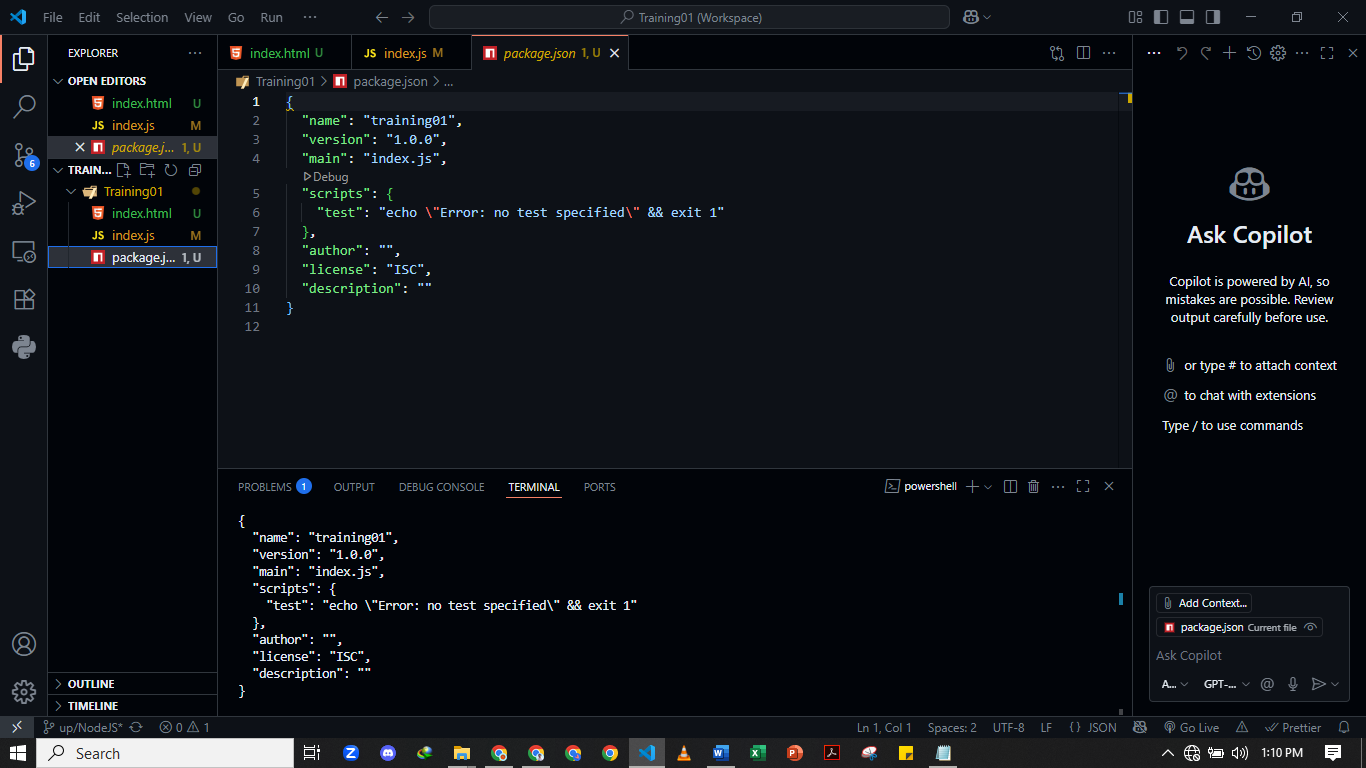
cmd :

*npm init*

Setelah itu kalian akan diminta menkonfigurasi data identitas pada package.json project kita seperti berikut.



Dengan hasilnya seperti berikut



Pada gambar diatas merupakan hanya contoh data default dari npm itu sendiri dalam membuat package.json (artinya tanpa input apa – apa dan hanya pencet enter). Terlihat pada file, key “dependencies” masih belum ada, yang mana artinya belum dilakukan install package dari npm.

Jika kita ingin membuat package.json secara dufault yes karena kita ingin mengeditnya nanti saja, bisa kita gunakan command sebagai berikut.

cmd

*npm init -y* atau *npm init --yes*

Ada juga cara untuk melakukan default author dan license agar setiap kita membuat package.json secara default maka author dan license sudah terisi langsung, berikut caranya

cmd

*npm config set init-author-name “value yang mau ditulis”*

*npm set init-license “value yang mau ditulis”*

Selanjutnya setelah kita membuat package.json, barulah kita instalasi package npm pada project kita, contoh dibawah akan dilakukan instalasi package pada npm bernama moment.

cmd

*npm i moment*

A screenshot of a computer program

AI-generated content may be incorrect.

Terlihat bahwa package moment sudah berhasil diinstall dan kita bisa melihatnya pada folder baru bernama node\_module dengan folder package bernama moment. Terdapat juga file bernama package-lock.json yang mana sangat penting karena berisi package apa saja yang kita install dalam project kita.

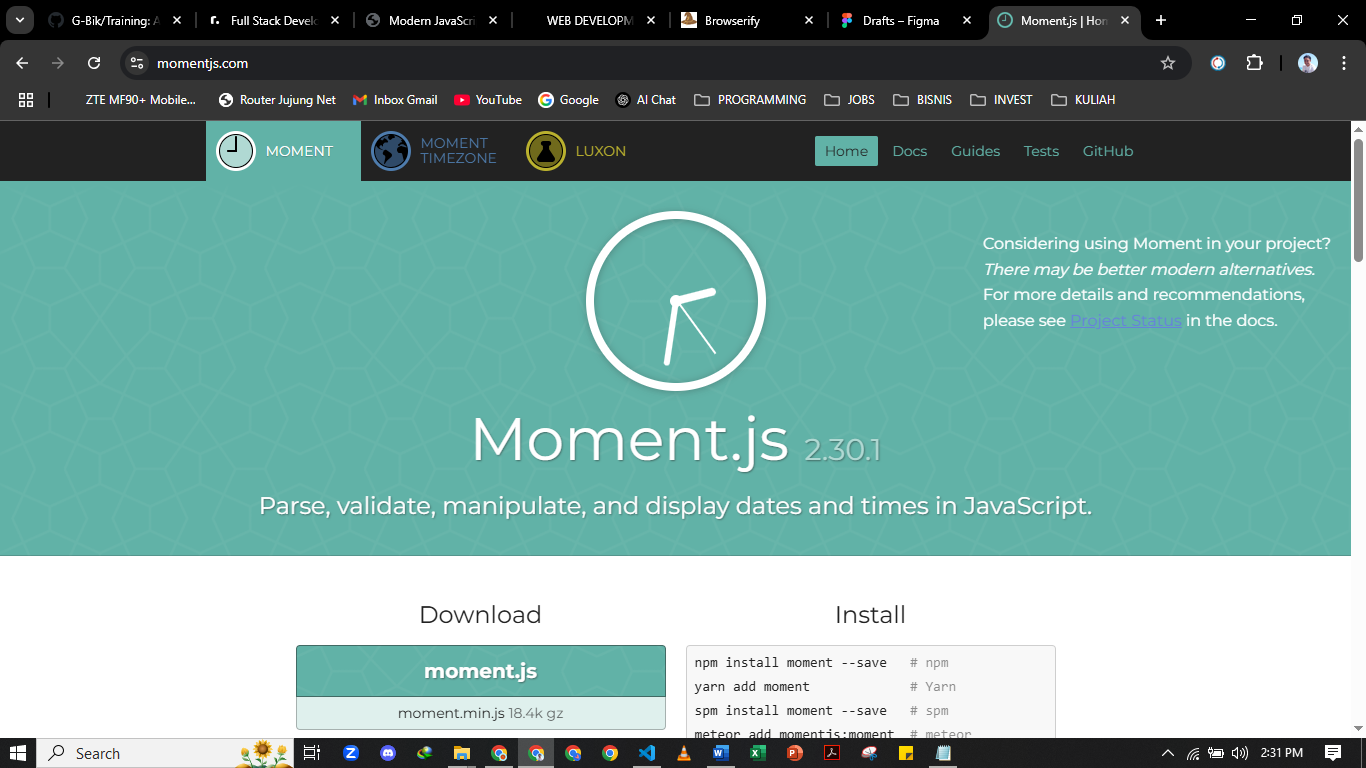
Kita juga bisa melihat list package apa saja yang kita install pada project yang kita buat dengan melihat metadata pada key “dependencies”, serta dibawahnya mengenai detail dari package yang kita install. Dependencies juga akan terlihat pada file package.json seperti berikut.

A screenshot of a computer program

AI-generated content may be incorrect.

1. Gunakan Package teresbut pada Javascript project kita, dan setiap package memiliki cara penggunaannya masing – masing, maka dari itu kita bisa liat langsung link documentasinya pada website npm sebagai berikut.

Contoh Documentation dari package Express pada NPM



Contoh diatas Adalah documentation dari package moment pada npm, yang mana kita bisa lihat cara penggunaanya pada Javascript kita nanti setelah kita menginstall package tersebut

1. Menginstall Bundler

Setelah kita gunakan package tersebut sesuai documentation yang ada pada web resminya seperti berikut.

A screenshot of a computer screen

AI-generated content may be incorrect.

Setelah dijlankan kode tersebut pada browser, yang terjadi malah eror. Alasan utama erornya Adalah browser tidak memahami require pada index.js, yang artinya package kita tidak dimengerti oleh browser itu sendiri seperti berikut.

A screenshot of a computer

AI-generated content may be incorrect.

Maka dari itu kita install bundler browserify, dengan code berikut.

cmd

*npm install -g browserify*

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

Setelah itu kita membuat file bundle seperti command yang diberikan oleh browserify

cmd :

*browserify namafile.js -o bundle.js*

A screenshot of a computer

AI-generated content may be incorrect.

Hasilnya setelah dilakukan instalasi

A screenshot of a computer screen

AI-generated content may be incorrect.

Setelah kita lakukan instalasi, nantinya akan terbentuk file baru yang disebut bundle.js seperti diatas. Terlihat bahwa pada bundler.js menyertakan code index.js yang kita buat dari variable hingga require yang tidak terdeteksi oleh browser tersebut.

**INGAT JANGAN MELAKUKAN SAVE PADA FILE BUNDLER TERSEBUT KARENA BISA MENGUBAH ISINYA.**

Perlu juga diingat bahwa ketika kita memasang bundler ini, maka command ini :

cmd :

*browserify namafile.js -o bundle.js*

***// INI KITA NAMAKAN COMMAND BUNDLER!!!***

harus juga dijalankan setiap kali kita menginstall package baru yang memerlukan “require()” baru pada file javascript kita, dan kalau bisa setiap kita ingin mengeksekusi code kita pada browser, maka jalankan saja command diatas terlebih dahulu.

Selanjutnya kita pasang file bundler tersebut pada file html dan sama seperti memanggil file javascript sebagai berikut.

A screenshot of a computer screen

AI-generated content may be incorrect.

Setelah itu kita bisa mencoba kembali codingan yang menjalankan package moment barusan.

A screenshot of a computer

AI-generated content may be incorrect.

Dan terlihat hasilnya sudah kelihatan bahwa format date dari package moment sudah berhasil dijalankan.

Sekarang kita contohkan mengapa kita perlu melakukan command bundler seperti dibawah disetiap setelah kita install package baru dan memerlukan require() baru juga

cmd :

*browserify namafile.js -o bundle.js*

Ketika kita ingin menginstall package uniq pada project kita (entah bua tapa nantinya guwe juga belum tahu), kita jalankan dulu command :

cmd

*npm install uniq*

A screenshot of a computer screen

AI-generated content may be incorrect.

Setelah kita menginstall package tersebut dan memasang kode yang diperlukan untuk mengetes package uniq ini pada file javascript kita, jika kita tidak menjalankan command bundler maka console.log() package unique tidak akan terlihat. Maka dari itu kita menjalankan command bundler seperti berikut.

A screenshot of a computer screen

AI-generated content may be incorrect.

Jika kita sudah menjalankan command bundler maka hasil console.log() package unique akan terlihat sebagai berikut.

A screenshot of a computer

AI-generated content may be incorrect.

1. asdasdasds
2. asdasdasd
3. asdasda