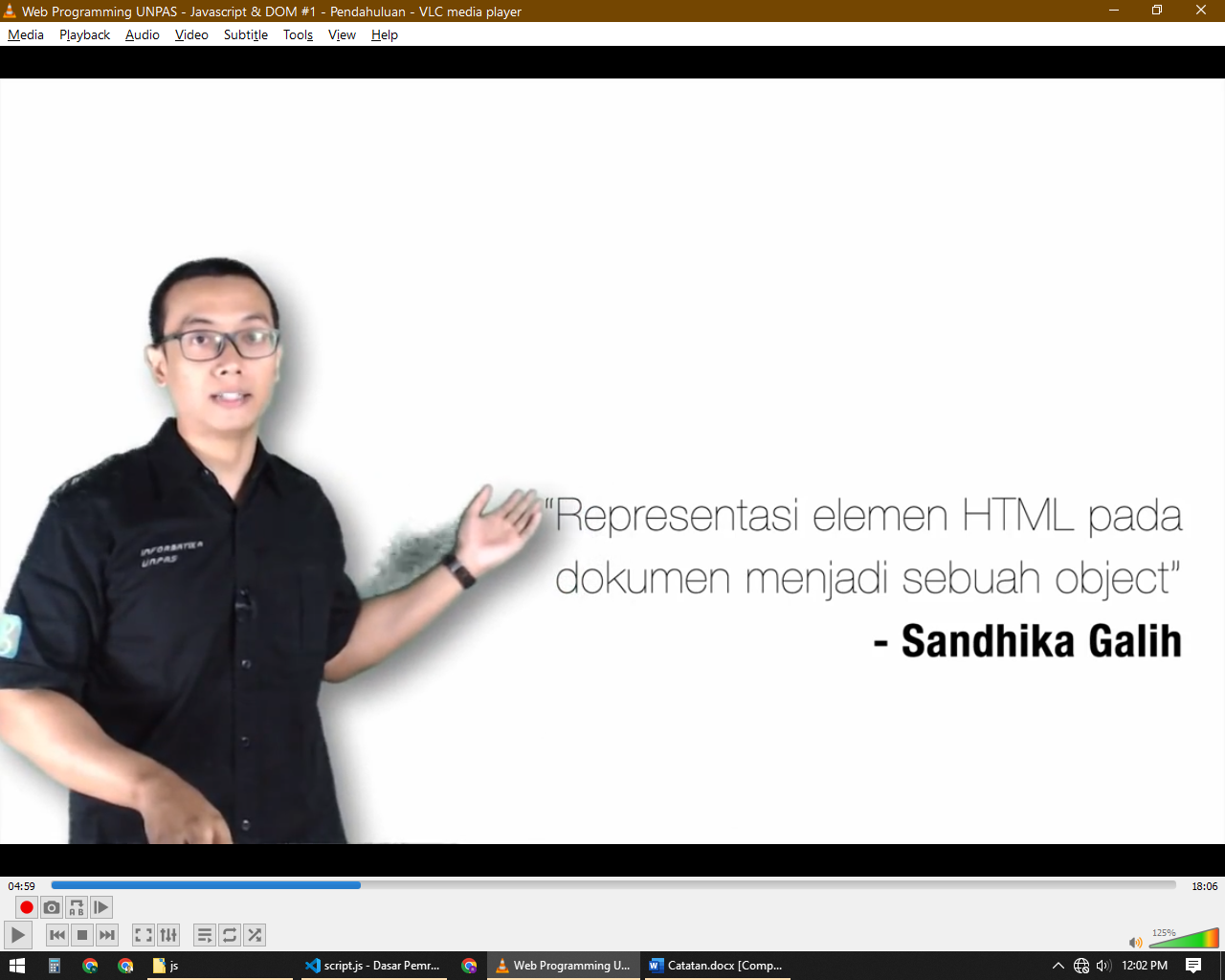
# **Javascript Document Object Model (DOM)**

## Definisi Javascript Document Object Model (DOM)



Menurut ***Mozilla*** DOM ***“merupakan antarmuka pemrograman untuk HTML yang merepresentasikan halaman web, sehingga program dapat mengubah dan memanipulasi strukturnya”***

dan Menurut bapak ***Sandhika Galih*** Javascript DOM memiliki definisi “Representasi elemen HTML pada Dokumen menjadi sebuah object”



Dan Definisi DOM menurut saya pribadi adalah “***Javascript DOM merupakan sebuah cara untuk memanipulasi element HTML pada dokumen(yang direpresentasikan sebagai object) agar bisa dirubah strukturnya secara pemrograman menggunakan javascript***”

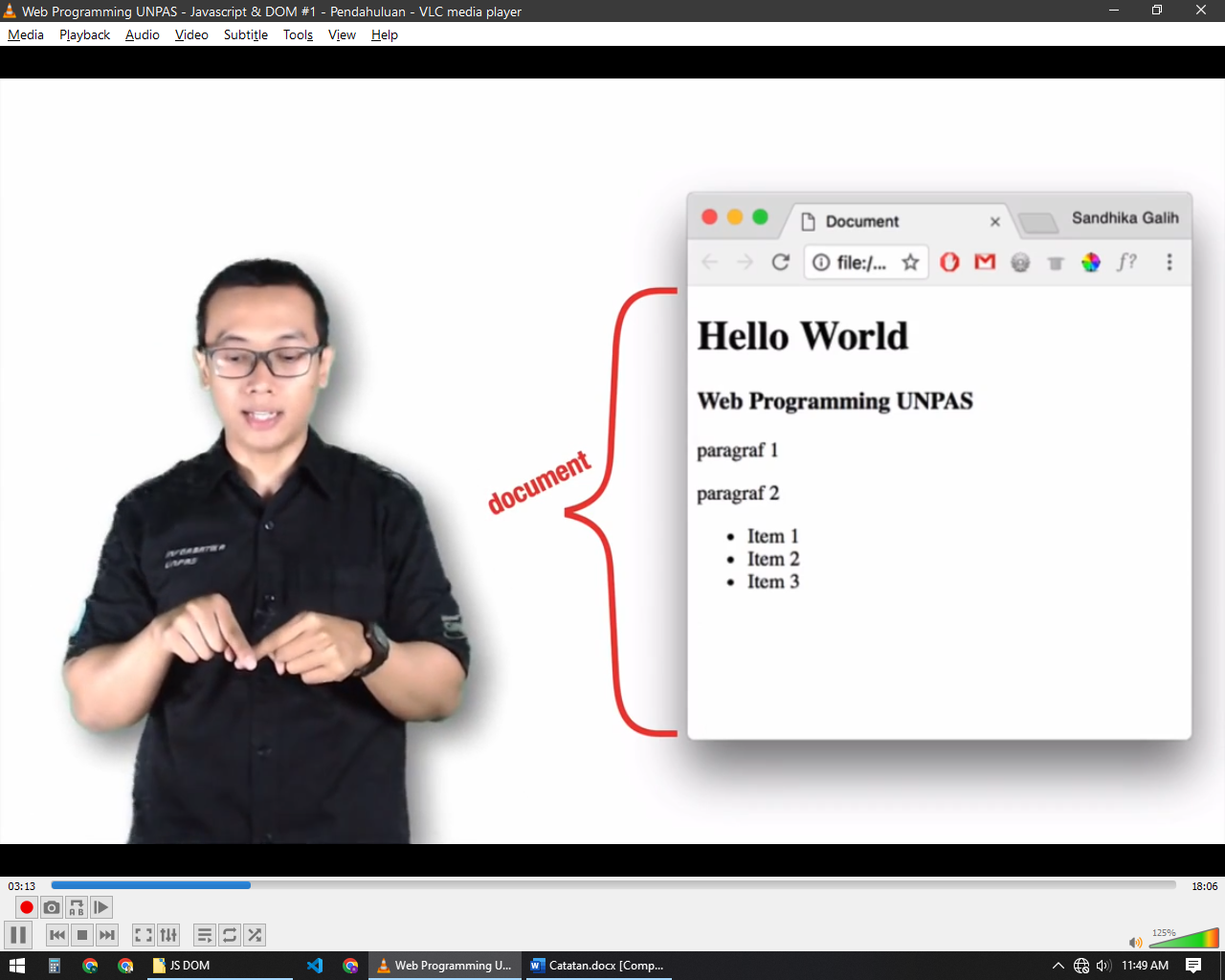
DOM berfungsi untuk membuat halaman website menjadi lebih interaktif menarik dan lebih baik

Untuk mempelajari Javascript DOM diperlukan pemahaman mengenai pemrograman dasar khususnya menggunakan Bahasa pemrograman Javascript dan DOM ini merupakan sebuah cara untuk Bahasa Javascript berinteraksi dengan HTML

***DOM*** Merupakan kependekan dari 3 element yaitu ***Document Object Model***

1. ***Document***

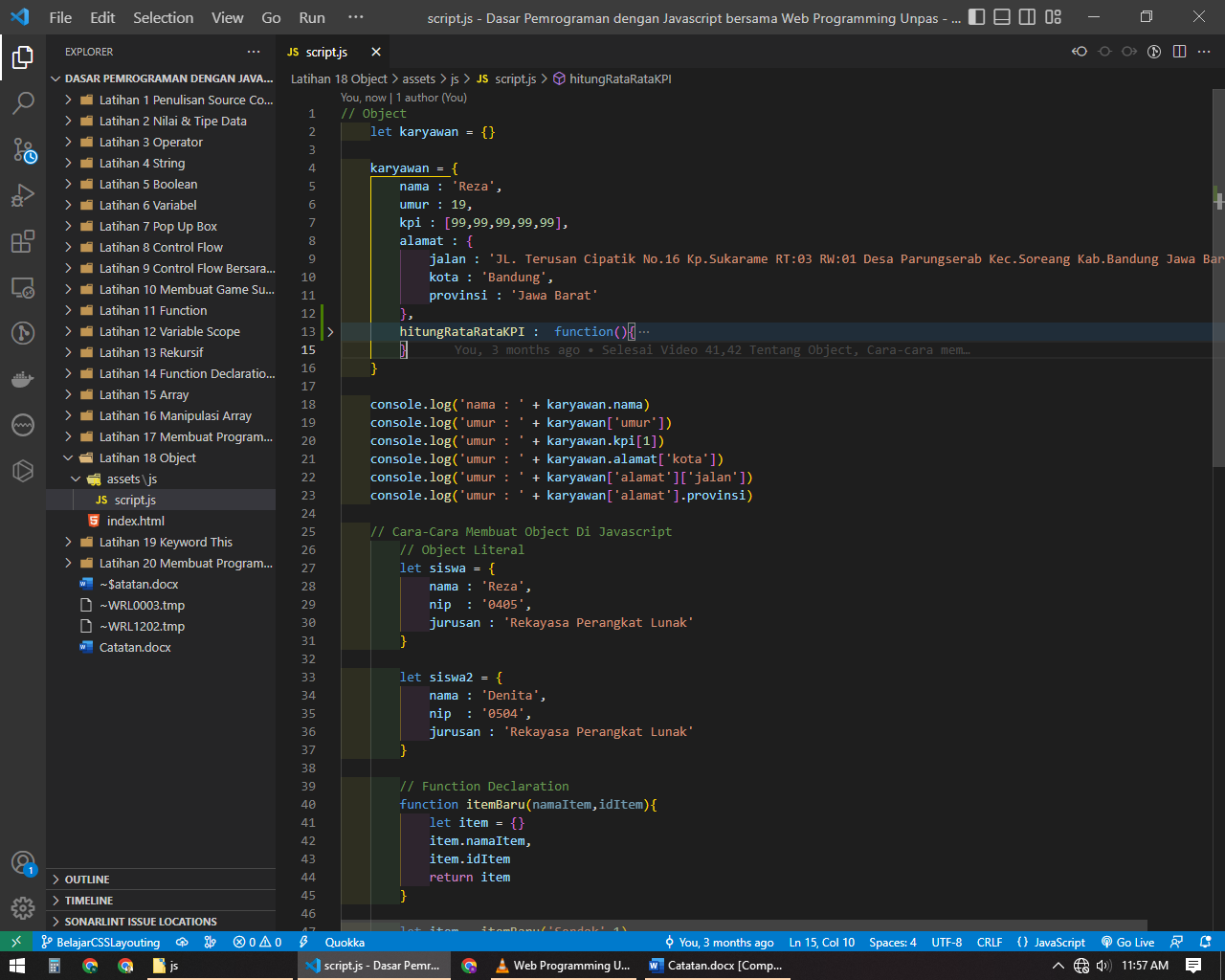
***Document*** pada DOM merujuk pada isi website



1. ***Object***

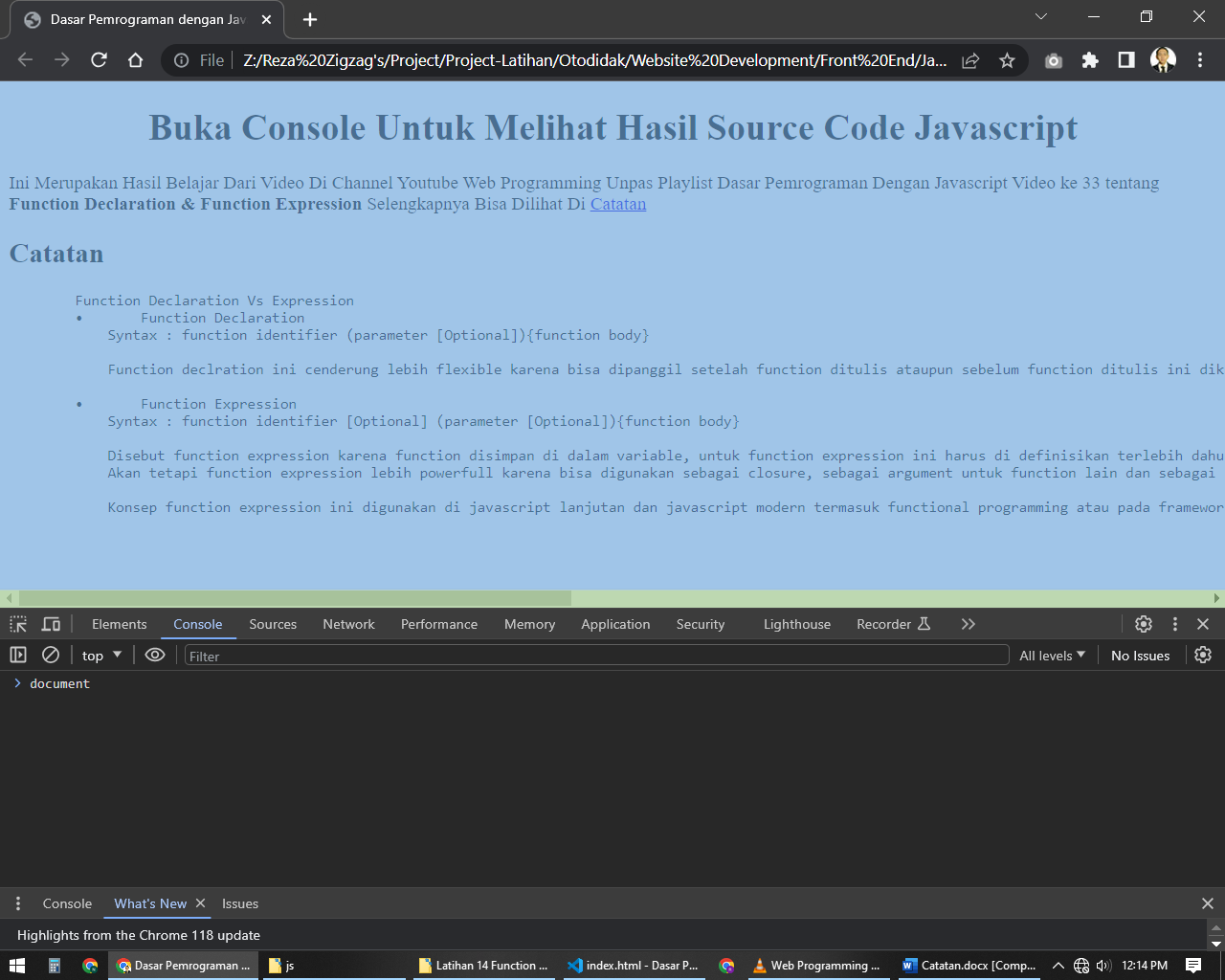
***Object*** pada DOM merupakan Nilai atau Tipe Data yang ada pada Javascript contohnya adalah object sederhana pada gambar dibawah yaitu sebuah variable object yang di dalamnya terdapat banyak nilai berupa property dan method.

***Object*** pada Javascript bisa juga menjadi kompleks misalnya di dalam object terdapat property berisi object dan object tersebut memiliki property berisi array yang isinya object

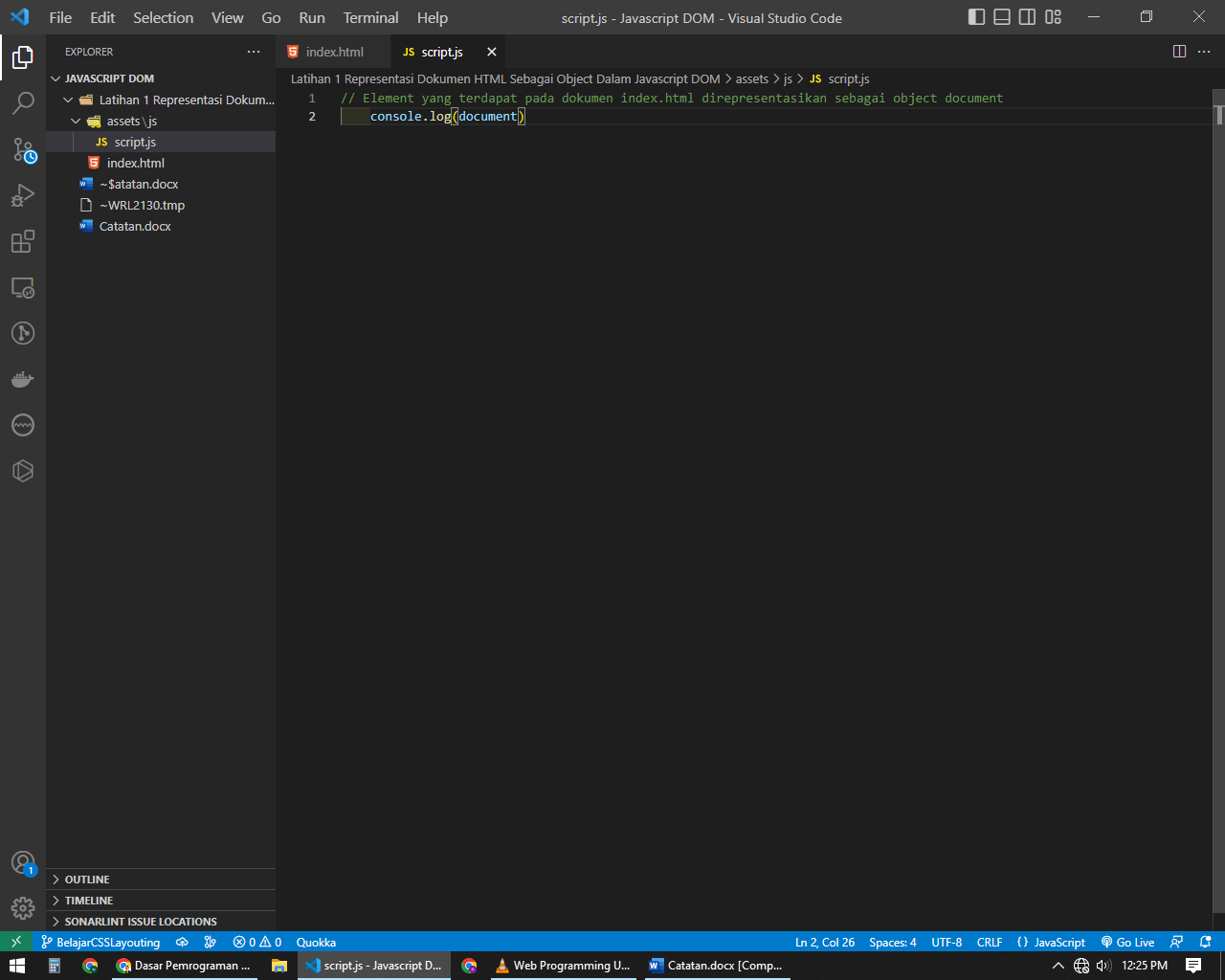


1. ***Model***

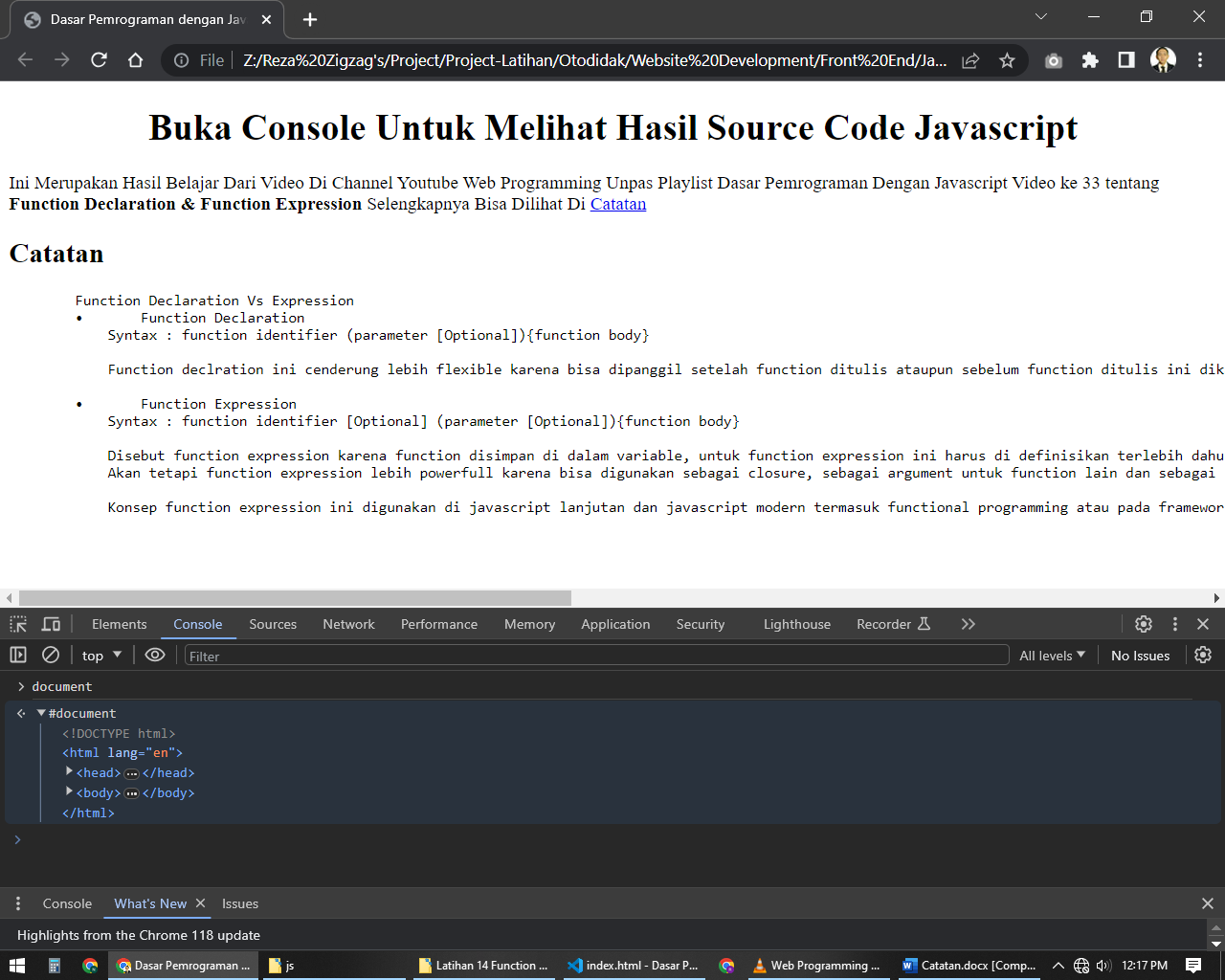
***Model*** dalam Javascript DOM bisa dimaknai sebagai representasi dari sesuatu lebih tepatnya ketika membuka sebuah dokumen html seluruh strukturnya diterjemahkan(direpresentasikan) menjadi sebuah object document



Apabila di dalam console pada browser ditulis document maka browser akan menghiglight dan mengembalikan seluruh halaman sebagai isi dari object document



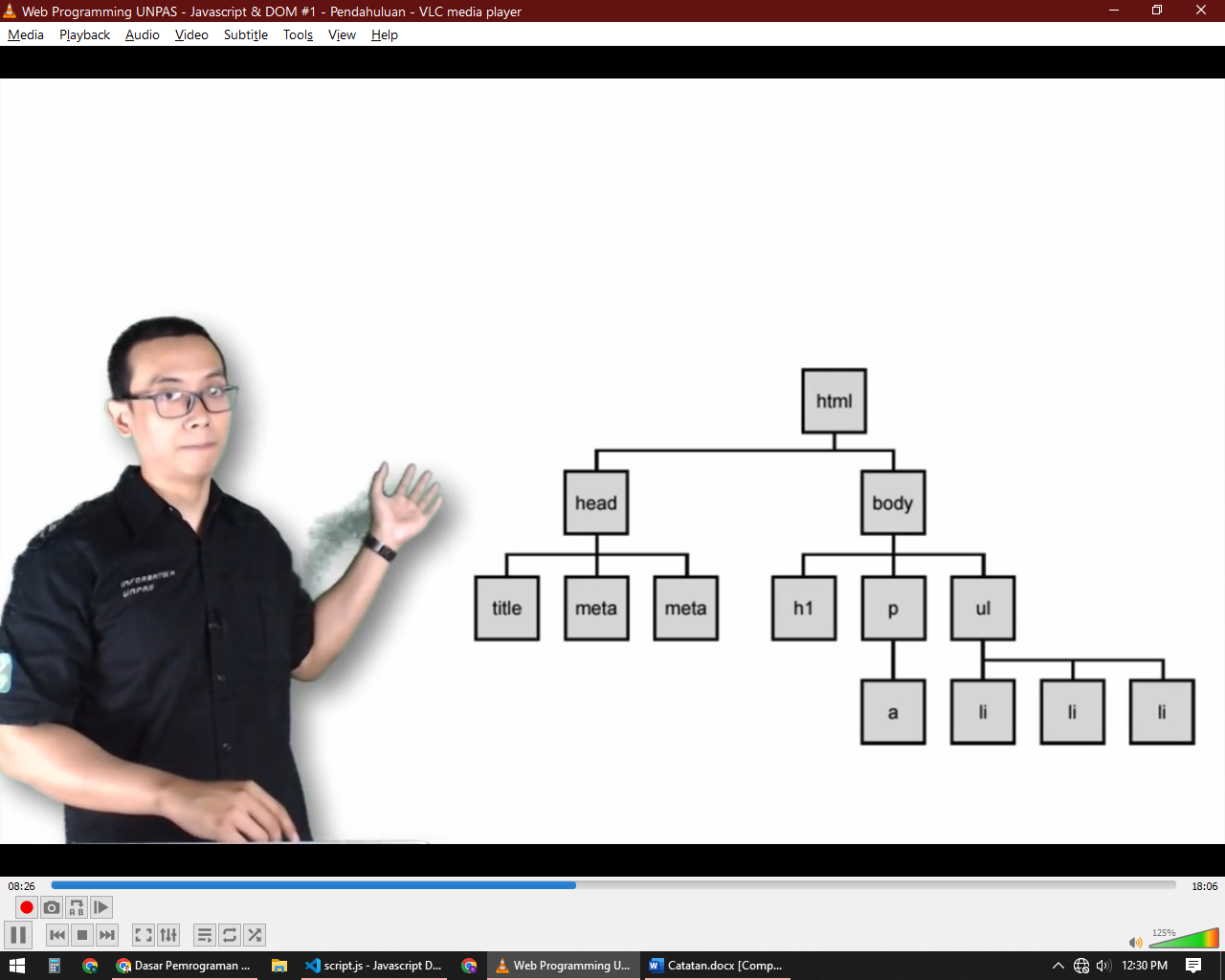
Tampilan pada console bisa dilihat pada gambar berikut



### Visualisasi Model dari Document sebagai DOM Tree

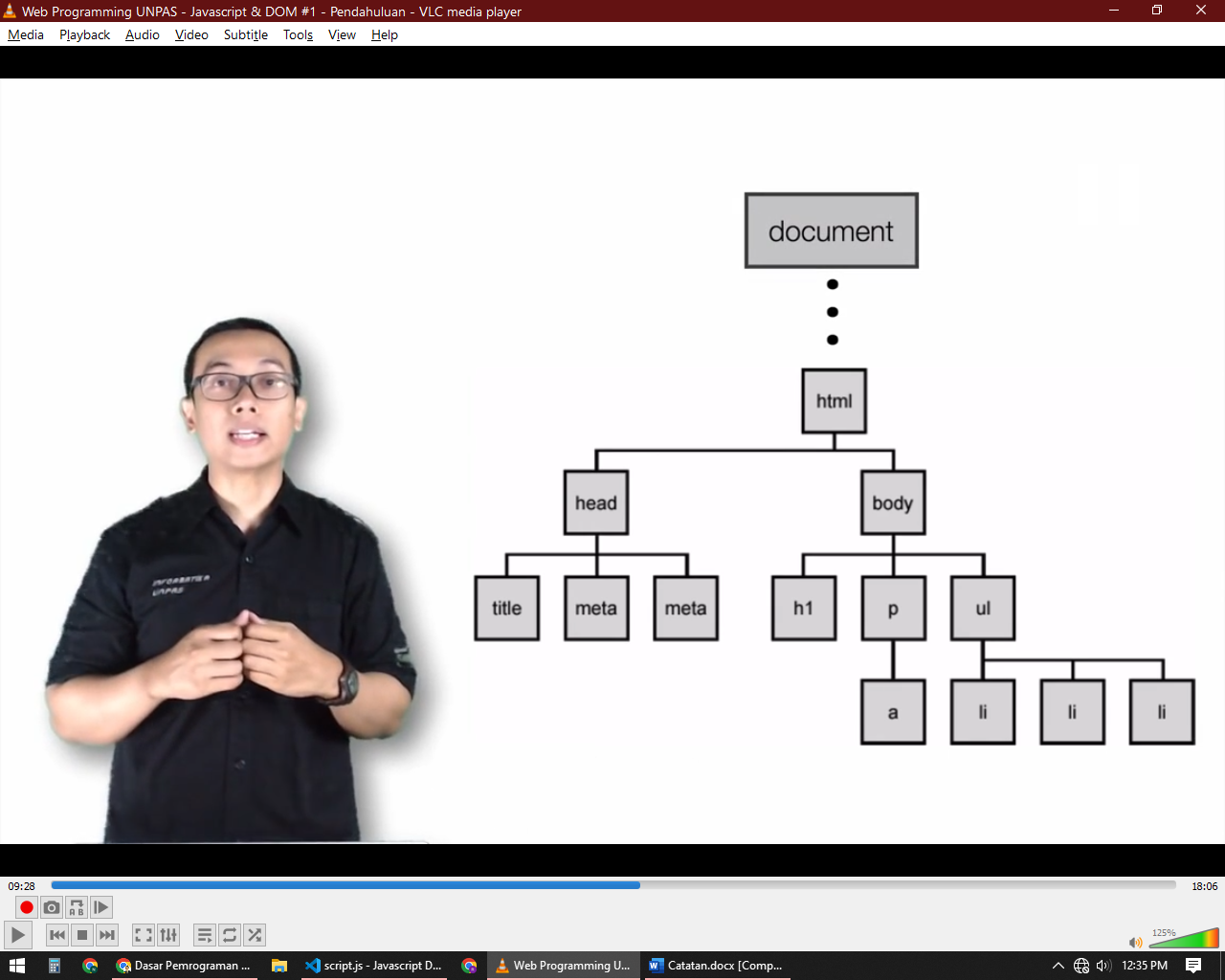
Halaman website di representasikan sebagai model yang tersimpan di dalam memory berbentuk ***Pohon Hirarki DOM*** atau disebut juga ***DOM Tree***

Visualisasi DOM Tree bisa dilihat pada gambar berikut

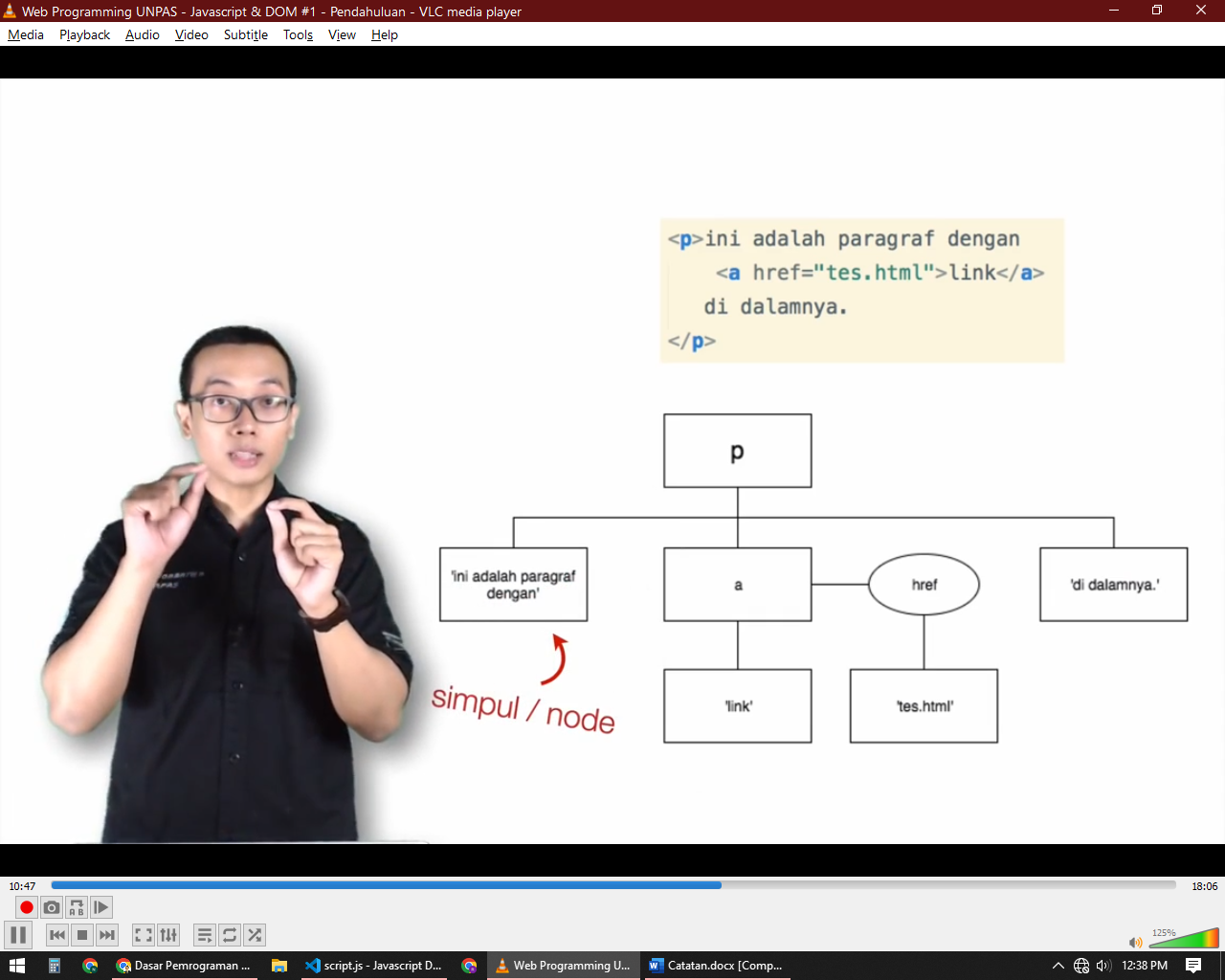


Hirarki ini sesuai dengan struktur HTML yang dibuat pada dokumen HTML nya, pemahaman mengenai DOM Tree akan menjadi penting saat akan melakukan pemilihan atau seleksi dan penelusuran element

Tiap kotak yang ada pada visualisasi DOM Tree diatas disebut dengan ***Simpul*** atau ***Node*** dan bentuknya merupakan object juga dan seluruh simpul yang ada pada halaman HTML terdapat pada satu buah simpul yaitu document



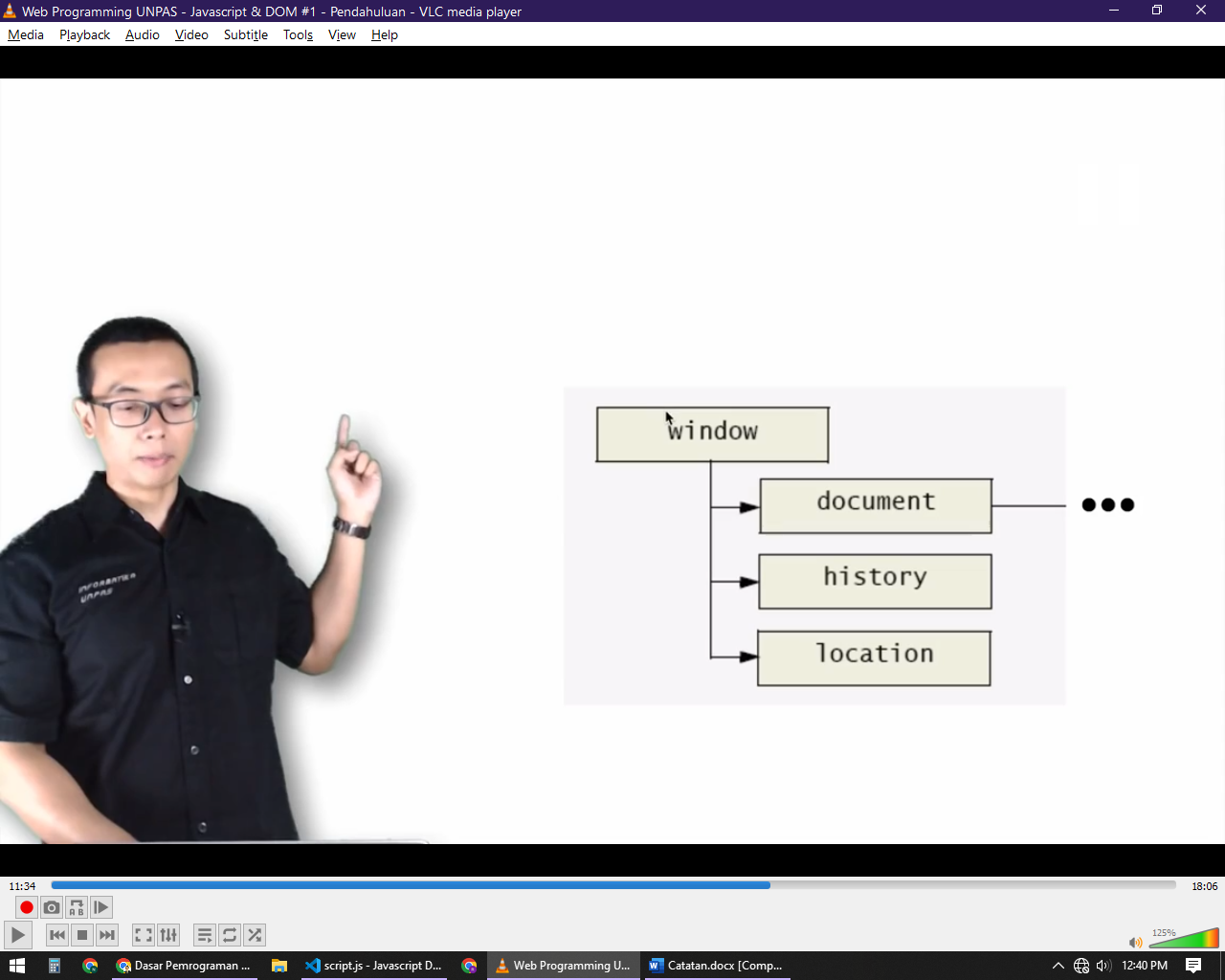
Contoh Lain :

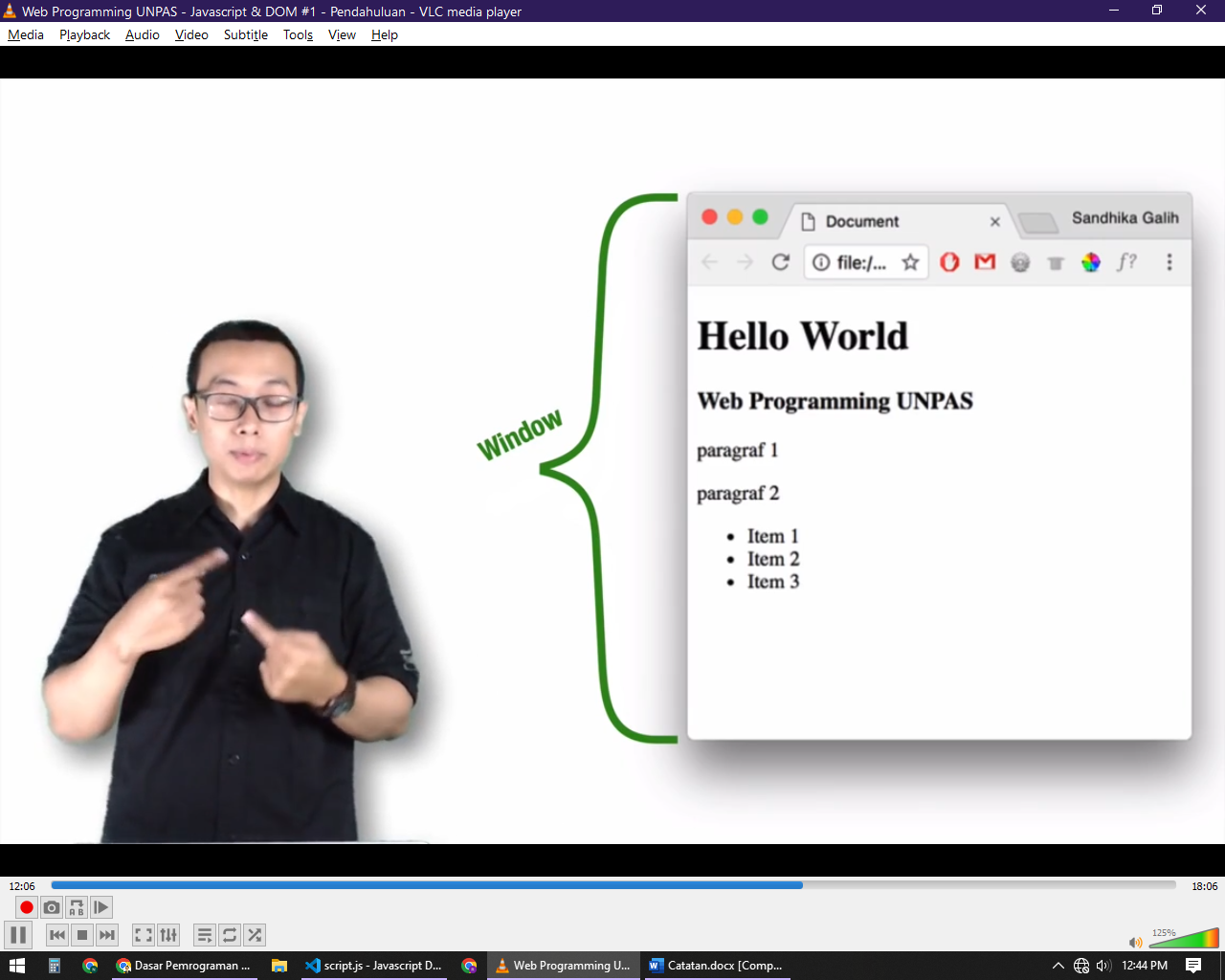


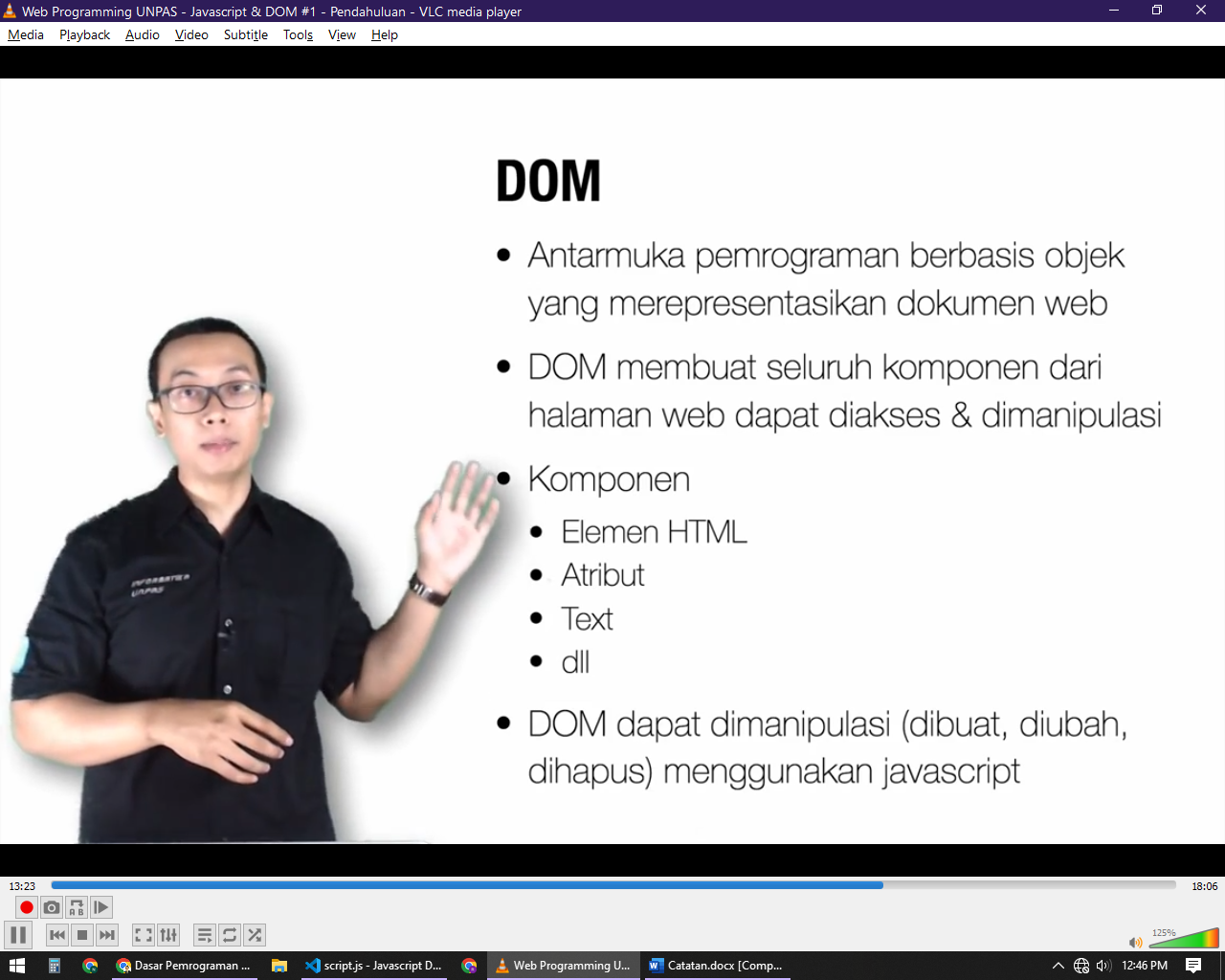
Semakin kompleks struktur HTML maka DOM Tree nya pun akan semakin besar

#### **Window Node**

Simpul Document bukan lah yang paling tinggi ada yang lebih tinggi kedudukanya yaitu object window, object document hanya memiliki akses terhadap website sedangkan object window memiliki akses yang lebih luas termasuk histori, lokasi kursor, besar ukuran window, dll







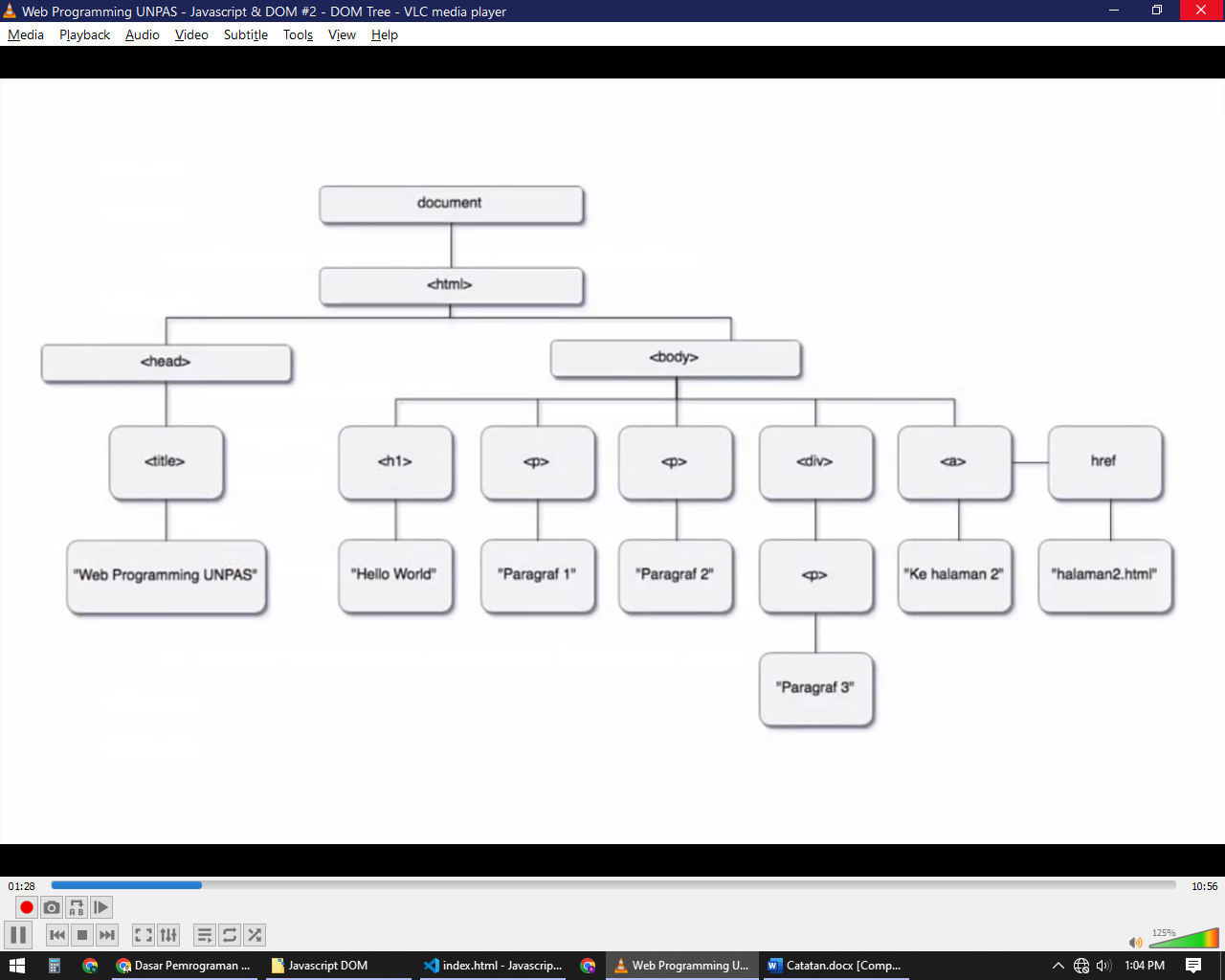
Kesimpulannya adalah DOM merupakan :

* Sebuah Antarmuka Pemrograman Berbasis Object Yang merepresentasikan dokumen web
* DOM Membuat seluruh komponen halaman web dapat diakses dan dimanipulasi
* komponen yang bisa diakses meliputi element HTML, Atribut pada Elemen, text, dll
* DOM dapat dimanipulasi(dibuat, diubah, dihapus ,menggunakan javascript)

## DOM Tree

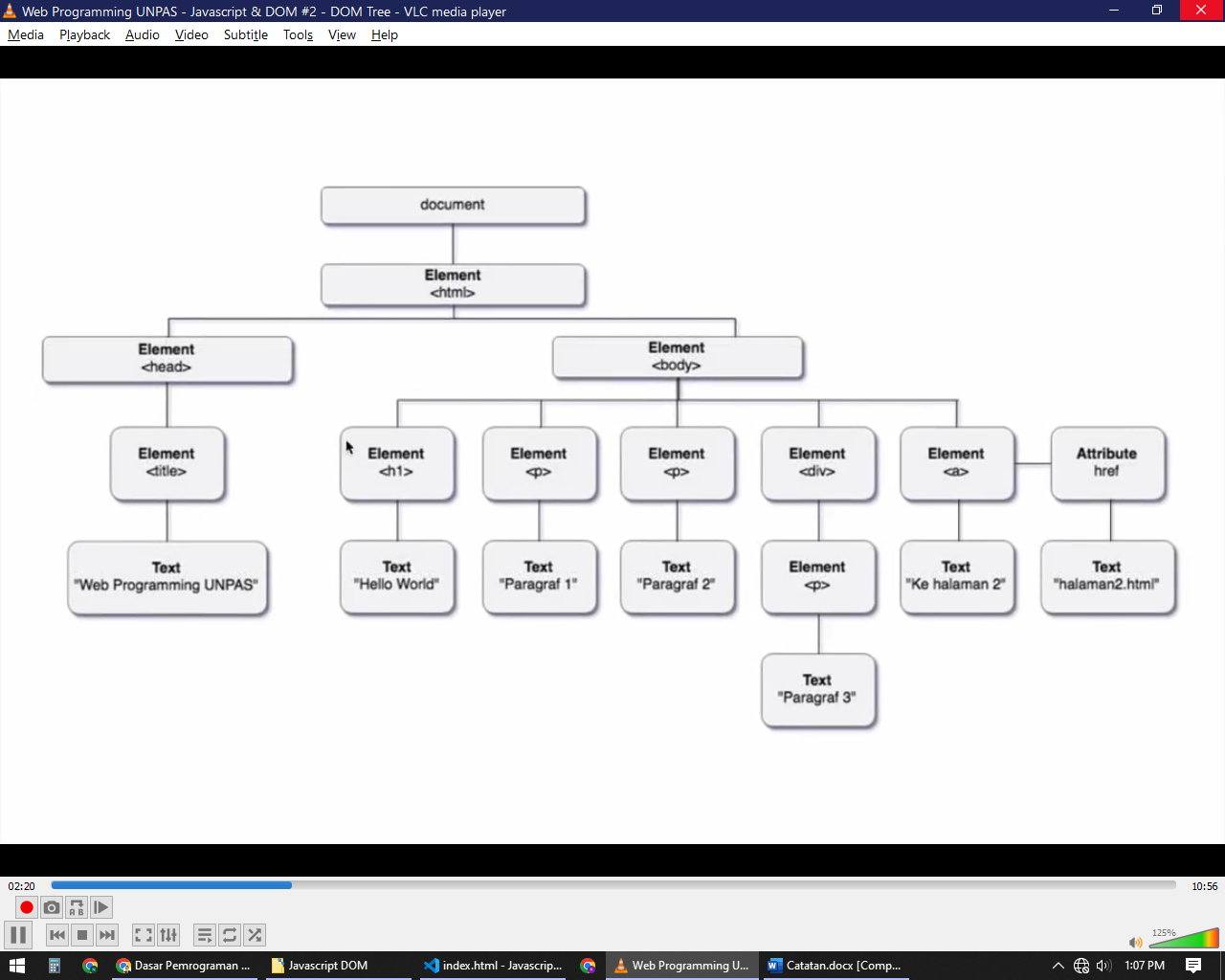
***DOM Tree*** Merupakan representasi element element HTML di dalam browser dan seluruh element pada DOM Tree Disebut sebagai ***Node*** atau ***Simpul***



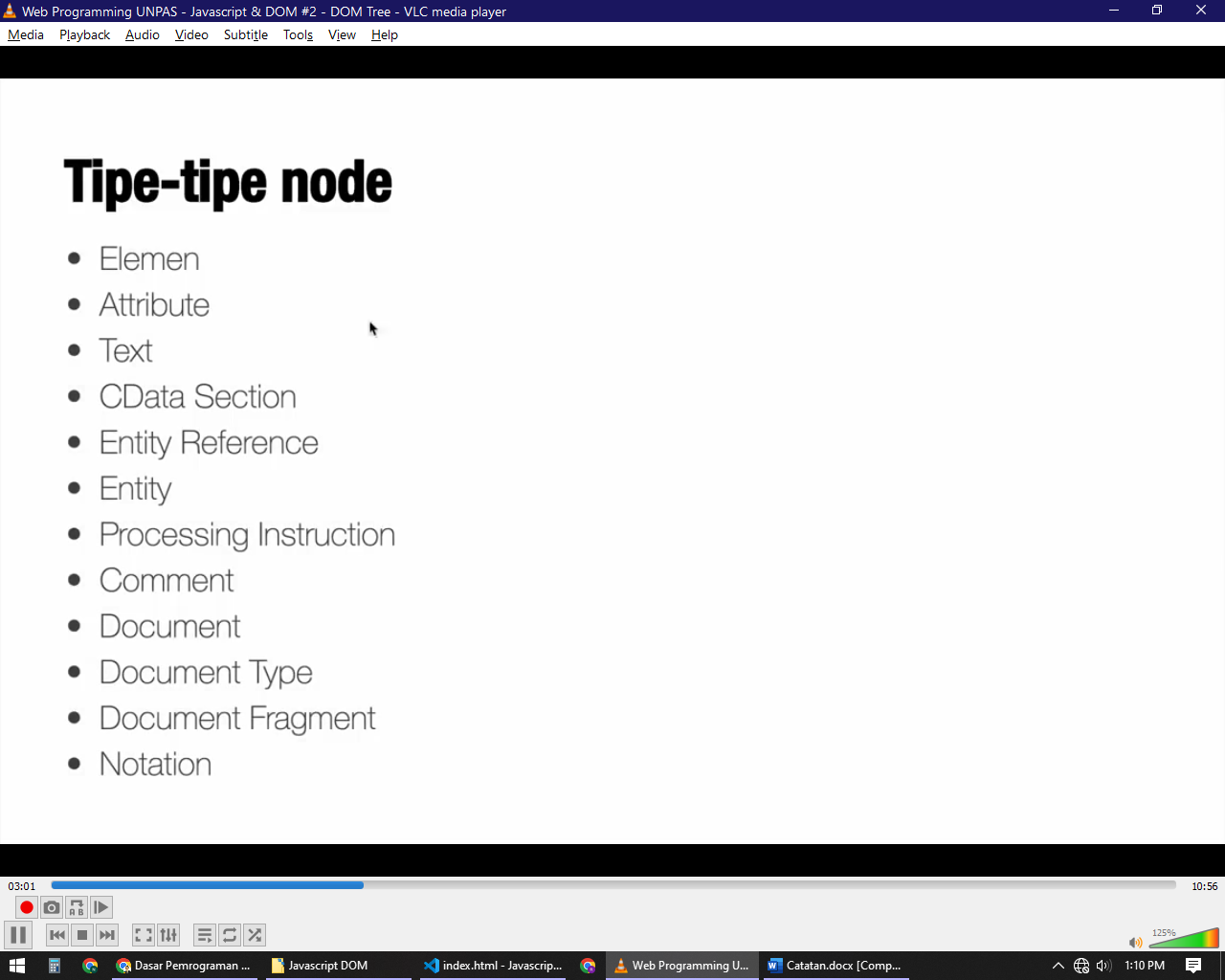


Node paling atas adalah ***Root*** atau ***Document*** head dan body merupakan element terluar dari HTML

### Type Pada Node



Node Pada DOM Tree memiliki tipe yang berbeda beda seluruh element HTML memiliki type ***Element*** dan tulisan yang berada di dalam element memiliki type ***Text*** ada juga simpul dengan type ***Attribute(Deprecated)*** seluruh simpul bisa disebut dengan Node + type nya misal ***Text Node*** atau ***Element Node***

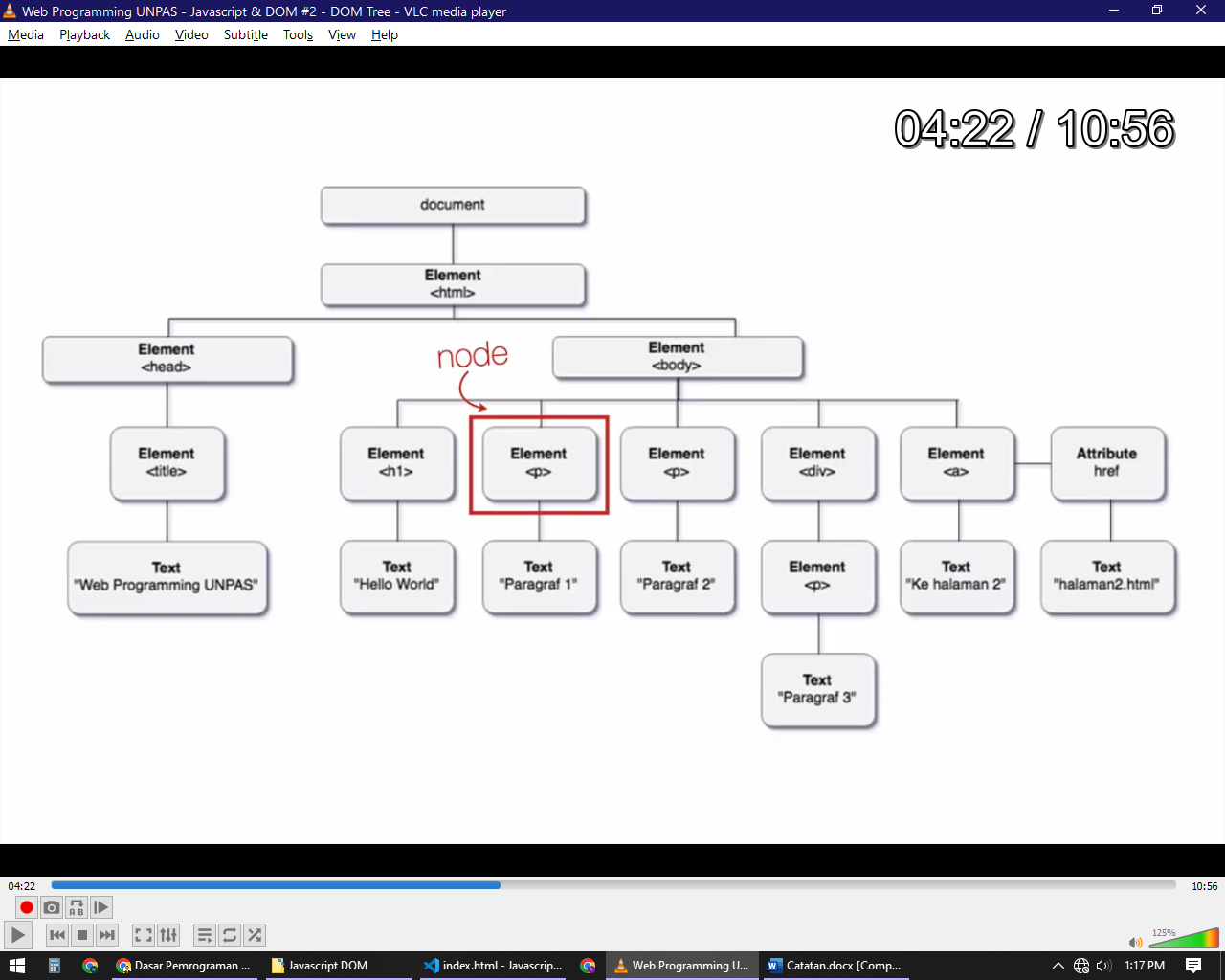


### Node List & HTML Collection

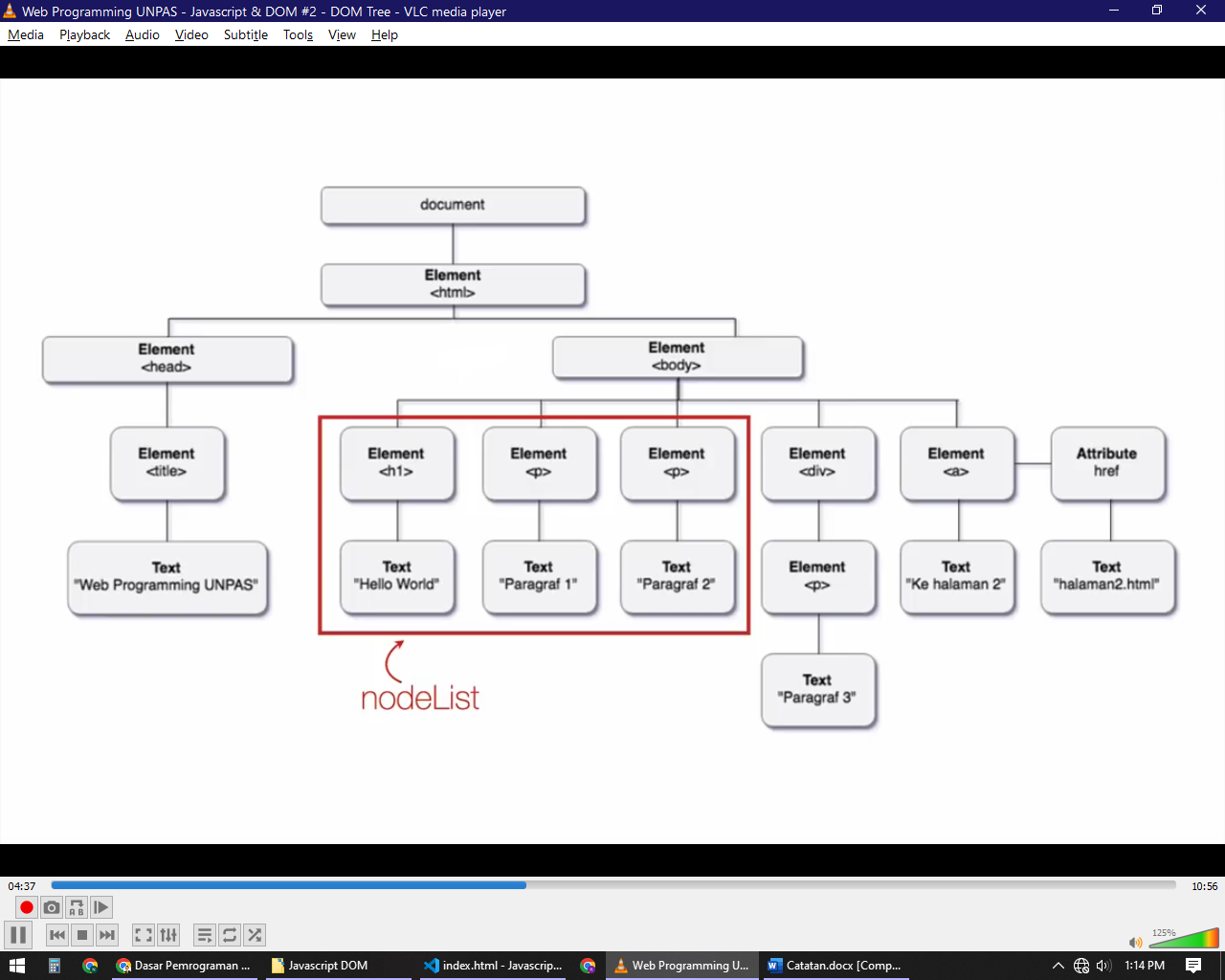
Node List & HTML Collection merupakan kumpulan node atau simpul yang dipilih kumpulan node yang berbeda, Perbedaan ini akan membuat cara pengelolaannya juga berbeda

struktur data kumpulan node mirip dengan struktur data array karena kumpulan node ini berisikan nilai nilai atau object object namun ada beberapa kriteria yang membuat kumpulan node tidak bisa disebut array

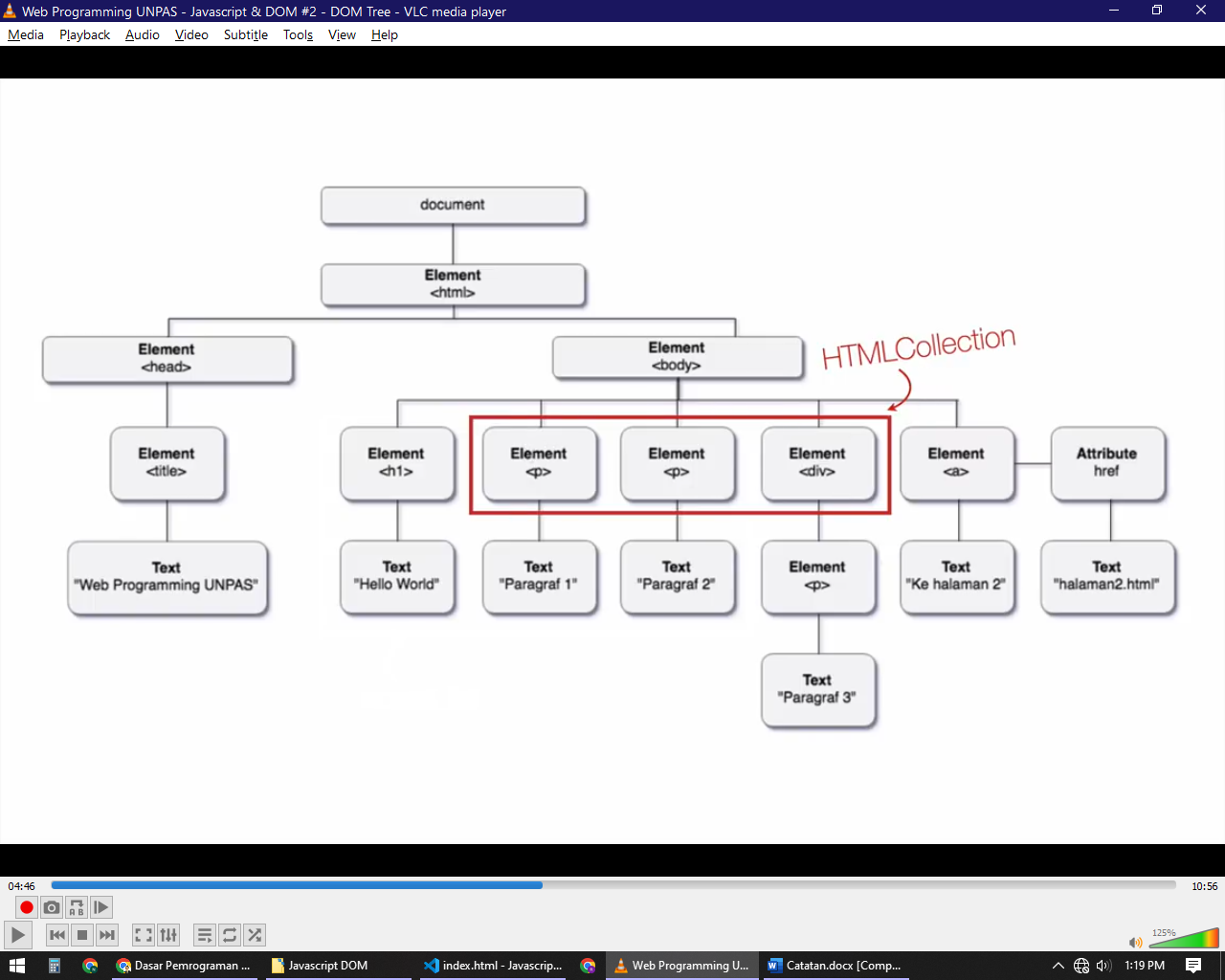
Ketika memilih sebuah node maka node yang dipilih akan sesuai dengan tipe node yang dipilih jika bertipe element maka node yang dipilih adalah ***node element***



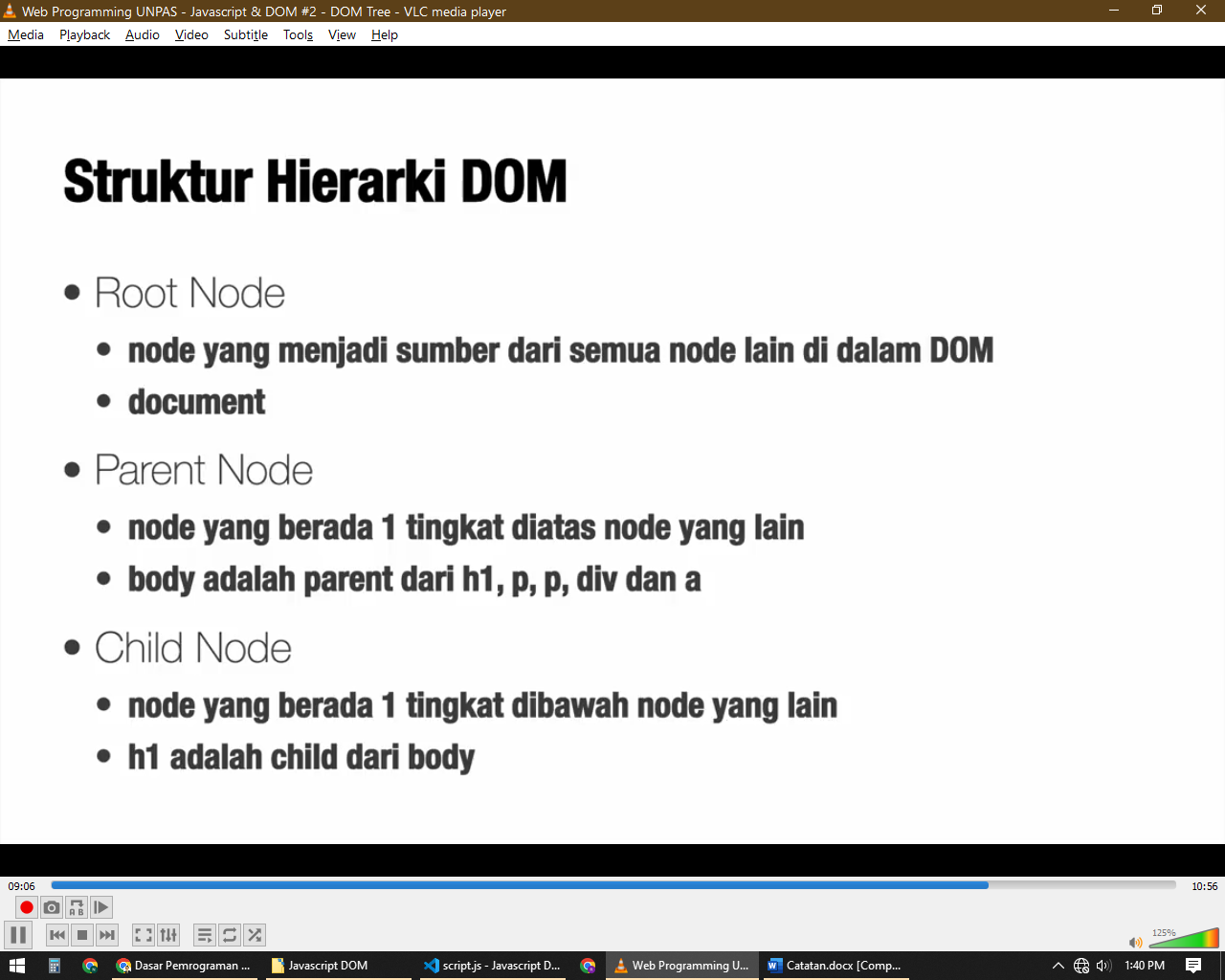
Dan jika memilih lebih dari satu node itu akan disebut sebagai ***Node List*** dan tidak memperdulikan tipe node nya apa



Sedangkan ***HTML Collection*** merupakan kumpulan node yang harus berisi node bertipe element saja HTML Collection bersifat Live sedangkan nodeList tidak



### Struktur Hirarki DOM Tree



Struktur Hirarki DOM Tree bisa dikatakan mirip dengan silsilah keluarga karena terdapat konsep konsep seperti parent dan child

#### **Root Node**

Merupakan node yang menjadi sumber dari semua node lain di dalam DOM secara default ***Root Node*** adalah ***document***

#### **Parent Node**

Merupakan Node yang berada 1 tingkat diatas node yang lain contohnya body adalah parent dari element dibawahnya seperti terdapat element h1, p, img, a

Apabila berada 2 tingkat diatas node lain ada yang menyebutnya sebagai ***Grand Parent Node*** dan jika lebih dari 2 tingkat diatas node lain ada yang menyebutnya sebagai ***Anchestor Node***

#### **Child Node**

Merupakan Node yang berada 1 tingkat dibawah node yang lain contohnya element h1, p, img, dan a merupakan child node dari body

Dan child node tidak memperdulikan tipe nodenya sedangkan child node yang spesifik tipe element nodenya disebut dengan ***Children Node***

Ada juga yang disebut dengan ***Sibling Node*** yaitu child node yang bersumber dari parent node yang sama

Dan apabila node yang berada lebih dari 1 tingkat dibawah node yang lain disebut dengan ***Descendant Node***