**Session 2**

**LINKED LIST**

**Part I**

**Intro to Linked List**

1. **Single Linked List = navigation is forward only**

* A node contains two main fields a value and a pointer to the next node.
* Single linked list memiliki hanya 1 arah panah/ pointer yang dimiliki node tersebut.
* Linked list node paling depan bernama head dan node terakhir bernama tail. Definisi node terakhir di linked list adalah node yang pointernya menunjuk ke NULL.
* Linked list tidak ada indexing.harus pake loop dan lewatin semuanya satu-satu.
* Linked list tidak cocok untuk searching.

1. **Double Linked List = navigation can be forward and backward**

* Bisa bolak balik karena nyimpan pointer ke belakang juga

1. **Circular Linked List = last node points to the first element**

* Node

peran head dan tail disini untuk menutupi kelemahan linked list yang ga mendukung indexing

di array kan kita sudah tau, "head" di array adalah index 0

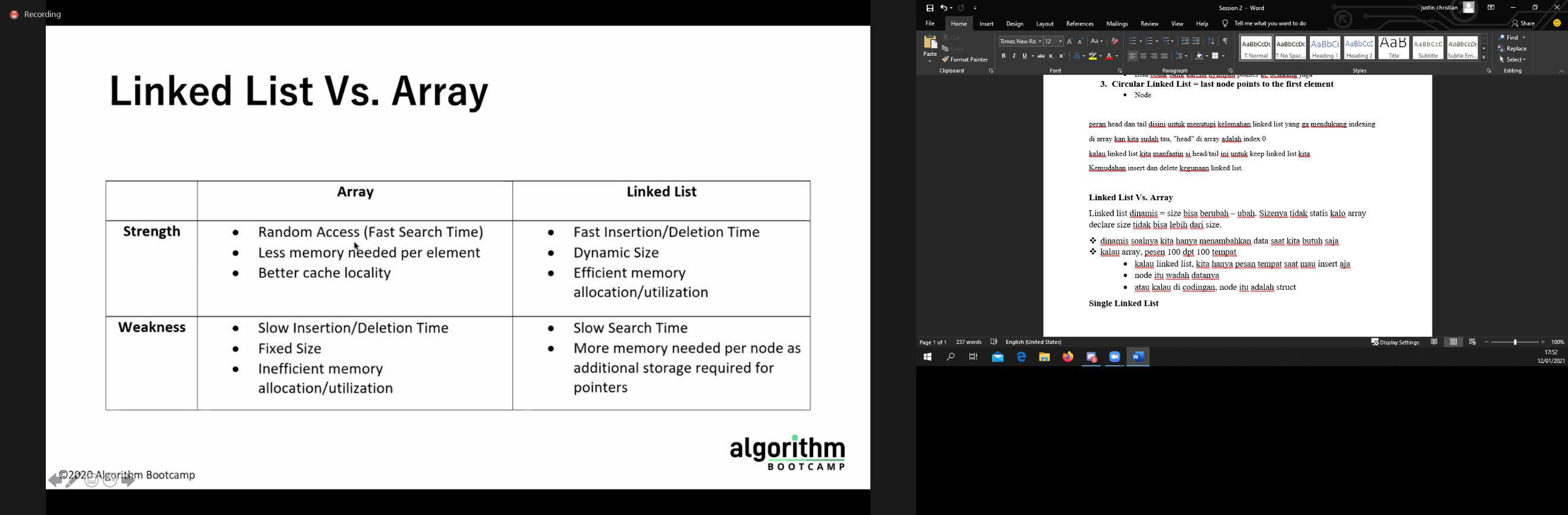
kalau linked list kita manfaatin si head/tail ini untuk keep linked list kita.

Kemudahan insert dan delete kegunaan linked list.

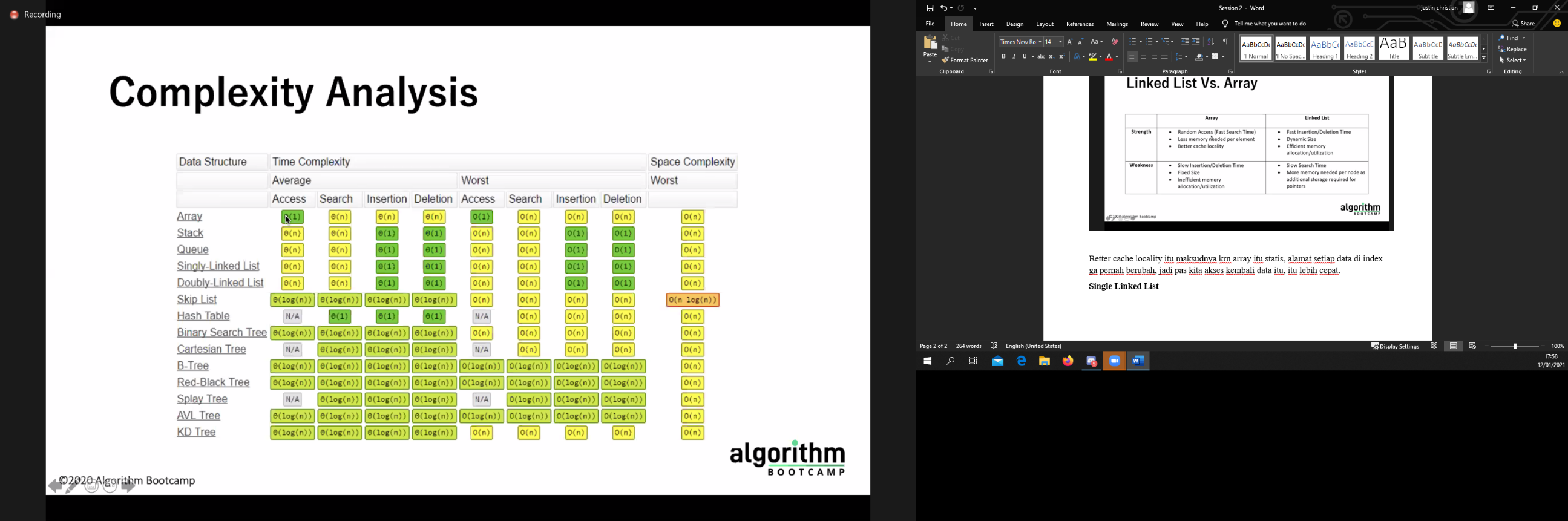
**Linked List Vs. Array**

Linked list dinamis = size bisa berubah – ubah. Sizenya tidak statis kalo array declare size tidak bisa lebih dari size.

* dinamis soalnya kita hanya menambahkan data saat kita butuh saja
* kalau array, pesen 100 dpt 100 tempat
* kalau linked list, kita hanya pesan tempat saat mau insert aja
* node itu wadah datanya
* atau kalau di codingan, node itu adalah struct



Better cache locality itu maksudnya krn array itu statis, alamat setiap data di index ga pernah berubah, jadi pas kita akses kembali data itu, itu lebih cepat.



2 4 2 3 1

mau insert 10 di index 3

2 4 2 10 3 1 (3 dan 1 harus geser ke kanan trus)

ini contoh kasus kenapa insert di array O(N)

dan delete juga sama aja, bedanya geser ke kiri

**Single Linked List**

A node can be made using struct.

A linked list is dynamic by nature and can allocate and deallocate memory at runtime.

Struct tnode{

Int value;

Struct tnode \*next;

};

struct buat linked list itu mending di global atau di local?

local

ini balik lagi ke global vs local

dimana mana ya lebih bagus local, karena scopingnya bisa kita atur

visualgo.net/en/list

kalo insert dari head otomatis ke geser?

iyaa headnya terganti

tapi "geser" ini definisinya gimana dlu

alamat memori node yang lain itu tidak berubah ya

jadi ttp disitu aja

hanya kita mengganti peran head aja.

Kodingan Linked List

jadi template/engga bukan jadi maslaah

data structure itu susahnya di penentuan DS itu sendiri

missal kamu dikasih masalah X, kamu mau pake DS apa? DS a? b? c?

itu yang susah

jdi ga boleh focus ke kodingan doang, tpi focus ke kegunaan linked list dan kenapa linked list ini dibuat