**Menginstall Depedency**

1. Setelah membuat project Composer, kita perlu menginstall depedency
2. Untuk menginstall depedency dari file composer.json, maka jalankan perintah berikut:

* Composer update

Perintah terserbut akan menginstall semua dependency dari composer.json dan menyimpannya di file composer.lock

Hasil semua instalasi akan disimpan di folder vendor

**Autoload**

1. Autoload digunakan agar kita bisa memasukkan source code di projek ke autoload PHP sehingga proses include PHP class tidak dilakukan satu per satu. Ada standar yang harus diikuti.
2. Cara menggunakan di file PHP:

* use NamaPackageComposerSaatInit\ClassNamePHPyangIngindiLoad

Example

* use ComposerHelloWorld\Car

**Composer dump autoload**

1. Setelah menambahkan autoload pada file composer.json, maka perlu digenerate ulang file autoload.php:

* composer dump-autoload

**Aturan Pembuatan Source Code pada src**

1. Namespace ***Satriabimantara***pada composer.json itu merujuk ke folder src
2. Artinya, jika kita import class di namespace ***Satriabimantara***, maka akan diinclude semua file di direktori src
3. Jika namespacenya ***Satriabimantara\ComposerHelloWorld***, maka folder acuannya adalah src/ComposerHelloWorld, sehingga perlu dibuat sub package di folder src
4. Nama file PHP pada folder src harus sama dengan nama kelasnya pada folder src. Semisal ada kelas **People**, maka filenya **People.php**

**Repository**

1. Repository merupakan tempat semua depedency disimpan. Secara default composer menggunakan repository packagist

<https://packagist.org/>

1. Kita juga bisa menambahkan repository selain packagist jika mau:

<https://getcomposer.org/doc/05-repositories.md>

**Menambah dependency**

1. Terdapat dua jenis library di composer, yaitu library ketika digunakan untuk development dan library ketika aplikasi berjalan

* Contoh library yang digunakan ketika development, yaitu unit test misalnya

1. Untuk menambah library, kita bisa tambahkan di composer dengan attribute **require** untuk library aplikasi berjalan dan **require-dev** untuk library development
2. Struktur pada composer.json dependency:

“require”:{

“library1”: “version”,

“library2”: “version”

},

“require-dev” : {

“libraryDev1” : “version”,

“libraryDev2” : “version”

}

**Versi Library**

1. Penulisan versi library bisa mengacu ke semantic versioning

https://getcomposer.org/doc/articles/versions.md

1. Contoh menambah library Monolog

“require”:{

“monolog/monolog”: “3.3.1”

},

1. Selalu jalankan kode berikut ketika menambahkan depedency baru pada file composer.json

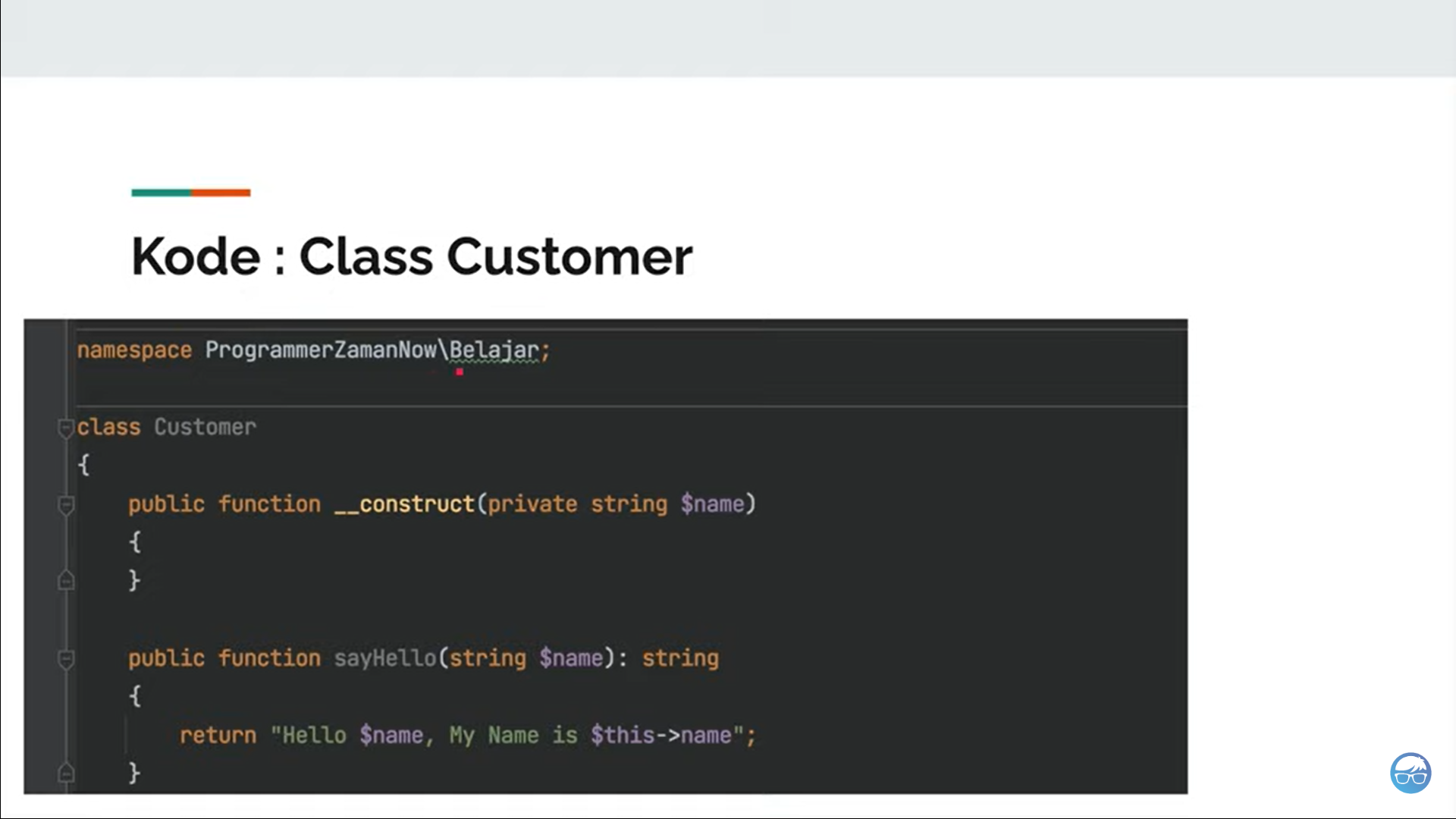
* Composer update
* Composer dump-autoload

**Membuat Library**

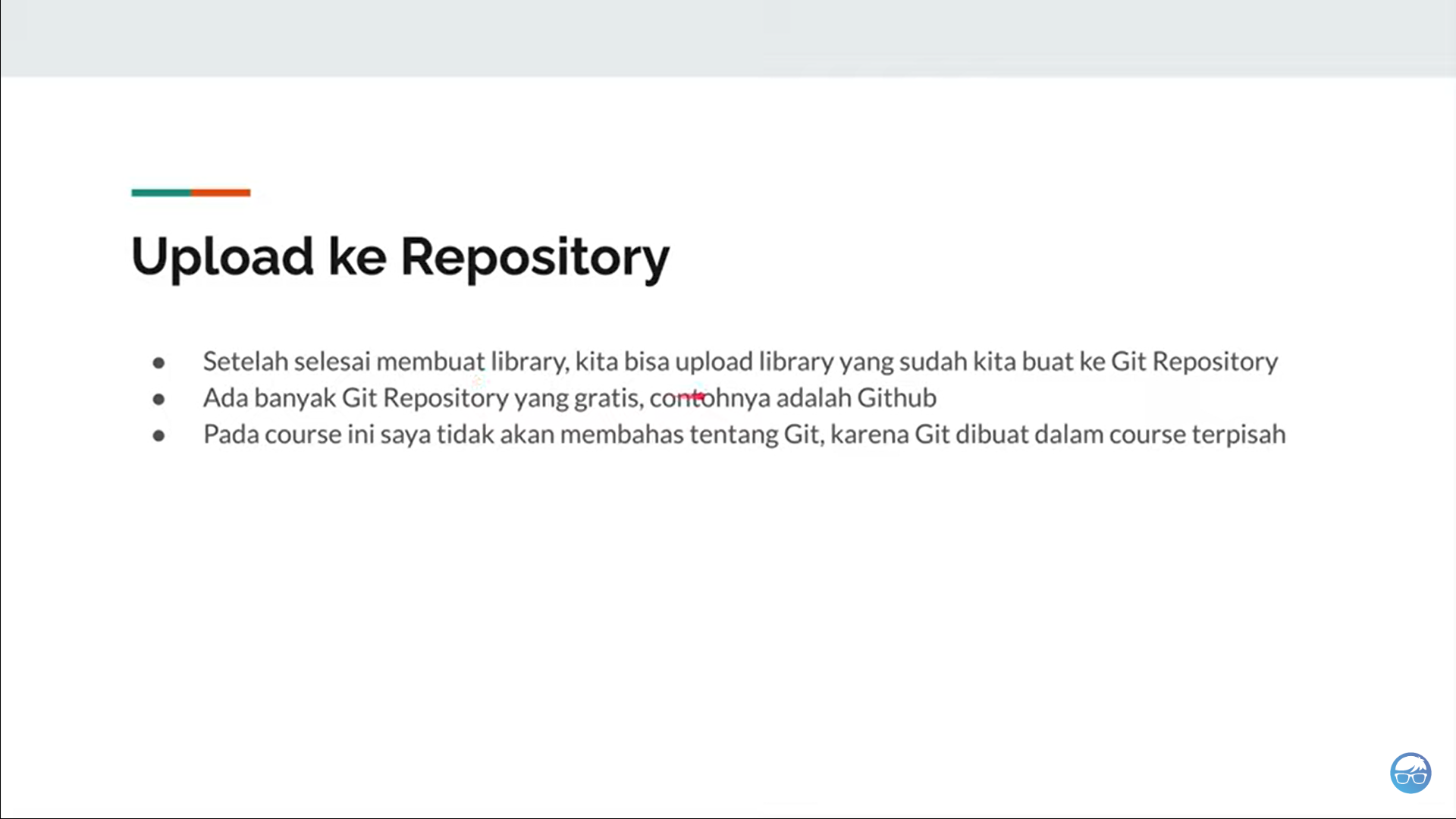
****

A screenshot of a computer

Description automatically generated with medium confidence



**Upload ke Repository**



1. Project composer diupload ke git repository dalam hal ini github. Buat suatu repository khusus di github lalu commit project library tadi ke Github
2. Buat **Release** atau **Tag** di-*github*. Tag nya digunakan sebagai versi dari library composer yang dibuat nanti

A screenshot of a computer

Description automatically generated

**Download dari Repository**

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidence

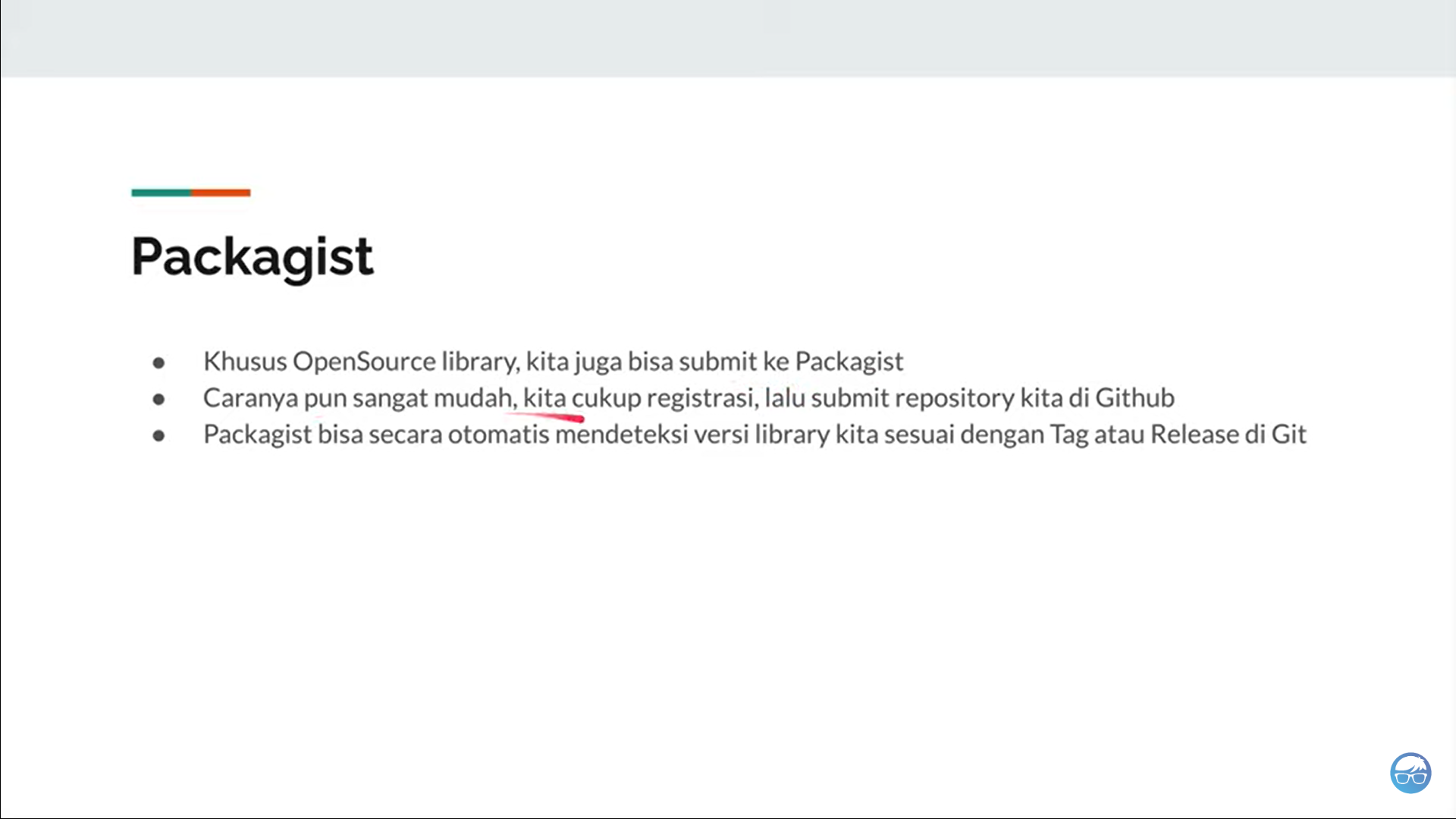
A screen shot of a computer

Description automatically generated with low confidence

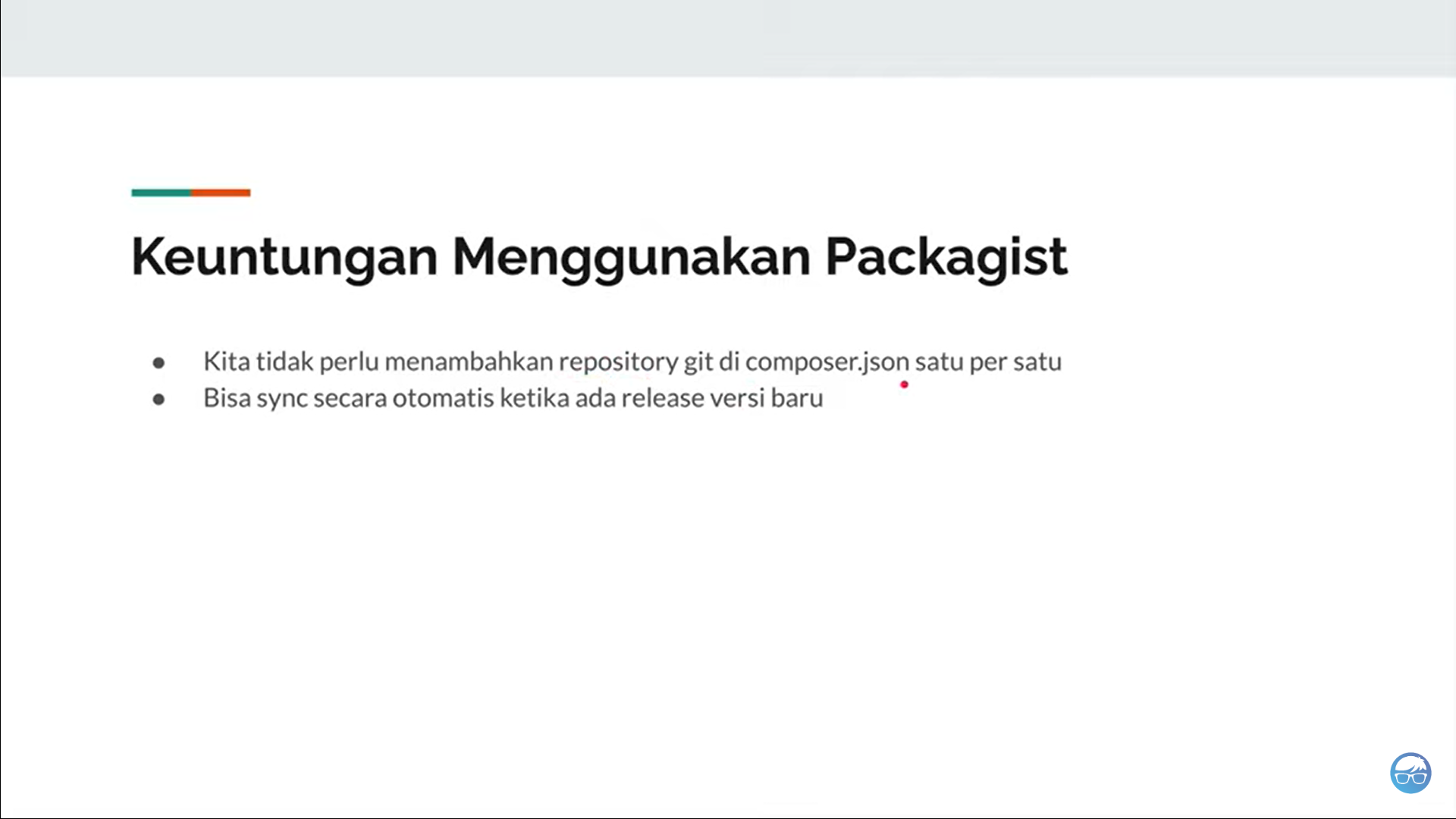
**Update Library**

1. Lakukan pembaharuan kode program composer library seperti memperbarui kode program biasanya
2. Git add, commit dan push
3. Buat release tag baru untuk version update tadi
4. Ganti versi terbaru dari folder project yang menggunakan composer library tersebut ke release tag terbaru

**Submit ke Pakcagist**



1. Registrasi di website Packagist dengan akun Github kita karena kita upload library kita melalui github
2. Submit library yang sudah kita buat di Github ke website Packagistnya dengan memasukkan URL repository sebelumnya



A screenshot of a computer program

Description automatically generated with low confidence

**Fitur lain Composer**

A screenshot of a computer

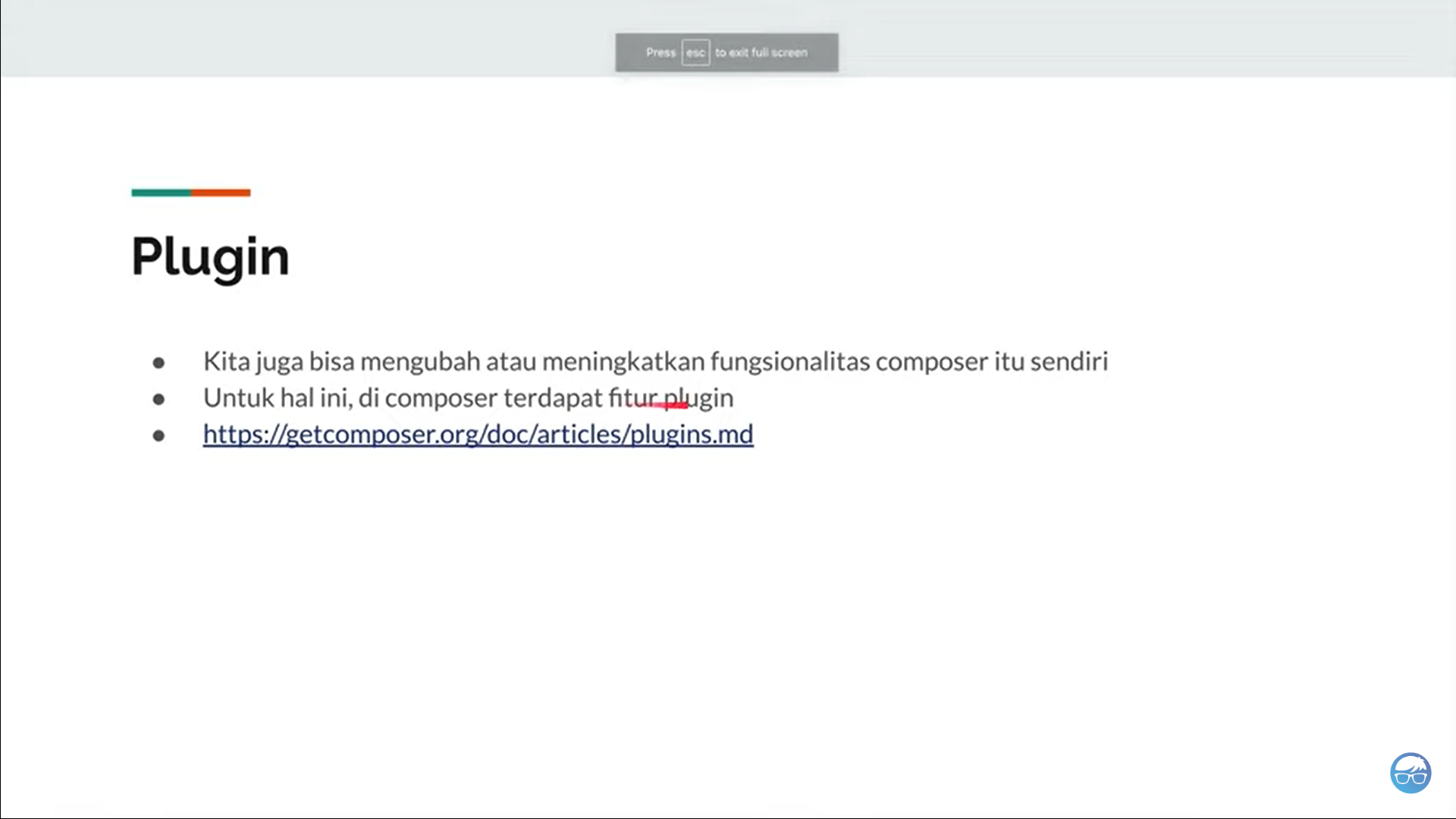
Description automatically generated with medium confidence

A screen shot of a computer script

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence



A screenshot of a computer

Description automatically generated with medium confidence