



LOG IN

Home

My Courses

My Tests

Achievements

Certificates

About



# Belajar Yuk

Tempat Mudah Untuk Belajar.

READ MORE





# Belajar Yuk

Tempat Mudah Untuk Belajar.

< Log in to your account

Log in with Google

Log in with Facebook

or

Email

Password

**LOG IN**

[Forgot password?](#)

New Account? [Sign Up](#)



Habib  
Premium

Home

## My Courses

## My Tests

## Achievements

## Certificates

## About

## Statistics with Phyton

```

    Returns a QuerySet of connections for user.
    """
    set1 = self.filter(from_user=user).select_related(depth=1)
    set2 = self.filter(to_user=user).select_related(depth=1)
    return set1 | set2

def are_connected(self, user1, user2):
    """
    If self.filter(from_user=user1, to_user=user2).count() > 0;
    and self.filter(from_user=user2, to_user=user1).count() > 0;
    return True
    else return False
    """
    return False

def remove(self, user1, user2):
    """
    Deletes proper object regardless of the order of users.
    If not connected - do nothing.
    """
    connection = self.filter(from_user=user1, to_user=user2)
    if not connection:
        return
    connection.delete()

```

## Introduction to JavaScript

```

32 self._file = None
33 self._fingerprints = {}
34 self._logistics = True
35 self._path = None
36 self._parent_fingerprint = None
37 self._paths = {}
38 self._parent = None
39 self._root_file = None
40 self._root_fingerprints = {}
41
42 @classmethod
43 def from_setting(cls, settings):
44     debug = settings.get("debug")
45     return cls(settings["fingerprint"], debug)
46
47 def request(self, request):
48     fp = self._parent_fingerprint(request)
49     if fp not in self._fingerprints:
50         return True
51     self._parent_file = os.path.join(fp,
52                                    self._root_file)
53
54     def request_fingerprint(self, request):
55         request._fingerprint = self._parent_fingerprint(request)
56

```

## Introduction to jQuery

```

    user(self, user);
    Returns a QuerySet of connections for user.

    set1 = self.filter(from_user=user).select_related(depth=1)
    set2 = self.filter(to_user=user).select_related(depth=1)
    return set1 | set2

def are_connected(self, user, user2):
    if self.filter(from_user=user1, to_user=user2).count() > 0:
        return True
    if self.filter(from_user=user2, to_user=user1).count() > 0:
        return True
    return False

def remove(self, user1, user2):
    Deletes proper object regardless of the order of ...
    connection = self.filter(...)

    if not connection:
        ...

```

## Introduction to Java



## IT & Software

Design

Marketing

## Writing

Business

## Photo & Film