

[Home](#)[My Courses](#)[My Tests](#)[Achievements](#)[Certificates](#)[About](#)[LOG IN](#)

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](#)

[Home](#)[My Courses](#)[My Tests](#)[Achievements](#)[Certificates](#)[About](#)Habib
Premium ✓

```
31     def __init__(self, path, settings):
32         self.path = path
33         self.fingerprints = set()
34         self.logdupes = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(self.path, "r", encoding="utf-8")
39             self.file.seek(0)
40             self.fingerprints.update(self._read_file())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool("logger.debug")
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

Learn Phyton

```
31     def __init__(self, user):
32         Returns a QuerySet of connections for user.
33         set1 = self.filter(from_user=user).select_related(depth=1)
34         set2 = self.filter(to_user=user).select_related(depth=1)
35         return set1 | set2
36
37     def are_connected(self, user1, user2):
38         if self.filter(from_user=user1, to_user=user2).count() > 0:
39             return True
40         if self.filter(from_user=user2, to_user=user1).count() > 0:
41             return True
42         return False
43
44     def remove(self, user1, user2):
45         Deletes proper object regardless of the order of users in connection.
46         connection = self.filter(from_user=user1, to_user=user2)
47         if not connection:
48             connection = self.filter(from_user=user2, to_user=user1)
```

Introduction to JavaScript

```
31     def __init__(self, path, settings):
32         self.path = path
33         self.fingerprints = set()
34         self.logdupes = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(self.path, "r", encoding="utf-8")
39             self.file.seek(0)
40             self.fingerprints.update(self._read_file())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool("logger.debug")
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

Introduction to HTML

```
31     def __init__(self, user):
32         Returns a QuerySet of connections for user.
33         set1 = self.filter(from_user=user).select_related(depth=1)
34         set2 = self.filter(to_user=user).select_related(depth=1)
35         return set1 | set2
36
37     def are_connected(self, user1, user2):
38         if self.filter(from_user=user1, to_user=user2).count() > 0:
39             return True
40         if self.filter(from_user=user2, to_user=user1).count() > 0:
41             return True
42         return False
43
44     def remove(self, user1, user2):
45         Deletes proper object regardless of the order of users in connection.
46         connection = self.filter(from_user=user1, to_user=user2)
47         if not connection:
```

Statistics with Phyton

```
31     def __init__(self, path, settings):
32         self.path = path
33         self.fingerprints = set()
34         self.logdupes = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(self.path, "r", encoding="utf-8")
39             self.file.seek(0)
40             self.fingerprints.update(self._read_file())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool("logger.debug")
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

Introduction to jQuery

```
31     def __init__(self, user):
32         Returns a QuerySet of connections for user.
33         set1 = self.filter(from_user=user).select_related(depth=1)
34         set2 = self.filter(to_user=user).select_related(depth=1)
35         return set1 | set2
36
37     def are_connected(self, user1, user2):
38         if self.filter(from_user=user1, to_user=user2).count() > 0:
39             return True
40         if self.filter(from_user=user2, to_user=user1).count() > 0:
41             return True
42         return False
43
44     def remove(self, user1, user2):
45         Deletes proper object regardless of the order of users in connection.
46         connection = self.filter(from_user=user1, to_user=user2)
47         if not connection:
```

Introduction to Java

IT & Software

Design

Marketing

Writing

Business

Photo & Film