

LAMPIRAN

Lampiran 1 Source Code Aplikasi

a. Source Code Validasi Token

```
/// Fungsi Untuk Menampilkan Dialog Input Token
func showInputToken() {
    let alert = UIAlertController(title:
NSBundleLocalizedString("common.alert.inputtoken.title", comment: ""), message:
NSBundleLocalizedString("common.alert.inputtoken.message", comment: ""),
preferredStyle: .alert)

    alert.addTextField { (textField) in
        textField.placeholder = "Token"
    }

    let cancelAction = UIAlertAction(title:
NSBundleLocalizedString("common.button.cancel.title", comment: ""), style:
UIAlertActionStyle.cancel) {
        UIAlertAction in
            alert.dismiss(animated: true, completion: nil)
    }
    alert.addAction(cancelAction)
    alert.addAction(UIAlertAction(title: "Submit", style: .default, handler: { [weak
alert] (_) in
        let textField = alert?.textFields![0]
        self.checkTokenAvailability(Token: (textField?.text!))
        alert?.dismiss(animated: true, completion: nil)
    })))
    self.present(alert, animated: true, completion: nil)
}

/// Step.1 : Fungsi untuk Memeriksa Ketersediaan Token
///
/// - Parameter token: token
func checkTokenAvailability(Token token : String) {
    guard token != "" else{
        self.showErrorInput(Title: "Token Kosong", Message: "Token tidak boleh
kosong. Silahkan masukan token yang didapat dari dosen")
        return
    }
    if isReachability! {
        Alamofire.request(URL(target:
PresenceAPI.checkMk(token))).responseJSON() {
            response in
                let json = JSON(response.result.value!)
                if let value = json.dictionary{
                    if let result = value["Detail"]?.dictionary{
                        if result["isSuccess"]?.rawString() != false.description{
```

```

        if let detailPresence : DetailPresence =
DetailPresence(JSONString: (value["Detail"]?.rawString()!)) {
            guard detailPresence.namaMk != nil else {
                self.showErrorTokenInput()
                return
            }
            self.currentToken = token;
            self.showPresenceConfirmation(Matakuliah:
detailPresence.namaMk!)
        }
        }else{
            self.showErrorTokenInput()
        }
    }

}

}else{
    self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
nil)
}
}

///Step.1.Message : Fungsi Untuk Menampilkan Dialog Konfirmasi Presensi
///
/// - Parameter mk: Nama Matakuliah
func showPresenceConfirmation(Matakuliah mk : String){
    let alert = UIAlertController(title: "Konfirmasi Presensi", message:"Apakah
Anda Yakin Akan Melakukan Presensi Pada Mata Kuliah \"(mk)\", preferredStyle:
UIAlertControllerStyle.alert)
    let okAction = UIAlertAction(title:
NSLocalizedString("common.button.ok.title", comment: ""), style:
UIAlertActionStyle.default) {
        UIAlertAction in
            self.checkValidasiPresensi(Nim: (self.mhs?.nim!)!, Token:
self.currentToken!)
    }
    let cancelAction = UIAlertAction(title:
NSLocalizedString("common.button.cancel.title", comment: ""), style:
UIAlertActionStyle.cancel) {
        UIAlertAction in
            alert.dismiss(animated: true, completion: nil)
    }
    alert.addAction(okAction)
    alert.addAction(cancelAction)
    self.present(alert, animated: true, completion: nil)
}

/// Step.1.Error.Message : Fungsi untuk Menampilkan Pesan bahwa Token
Salah
func showErrorTokenInput(){
    let alert = UIAlertController(title: "Token Salah", message:"Silahkan periksa
kembali token yang anda masukan.", preferredStyle: UIAlertControllerStyle.alert)

```

```

    let okAction = UIAlertAction(title:
NSLocalizedString("common.button.ok.title", comment: ""), style:
UIAlertActionStyle.default) {
    UIAlertAction in
        alert.dismiss(animated: true, completion: nil)
        self.showInputToken()
    }
    alert.addAction(okAction)
    self.present(alert, animated: true, completion: nil)
}

/// Step.2 : Fungsi untuk Memeriksa status absensi mahasiswa menggunakan
token
///
/// - Parameters:
/// - nim: nim
/// - token: token
func checkValidasiPresensi(Nim nim : String,Token token : String) {
    if isReachability! {
        Alamofire.request(URL(target: PresenceAPI.validasiPresence(nim,
token))).responseJSON() {
            response in
                if let value = response.result.value as? [String: Any] {
                    if let isSuccess = value["isSuccess"] as? Bool {
                        if isSuccess {
                            self.checkKRS(Nim: nim, Token: token)
                        }else{
                            self.showAlreadyPresence()
                        }
                    }
                }
            }
        }
    }else{
        self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
nil)
    }
}

/// Step.2.Error.Message : Fungsi untuk Menampilkan Pesan bahwa Sudah
melakukan absensi
func showAlreadyPresence(){
    let alert = UIAlertController(title: "Presensi Gagal", message:"Anda sudah
melakukan presensi pada mata kuliah ini sebelumnya.", preferredStyle:
UIAlertControllerStyle.alert)
    let okAction = UIAlertAction(title:
NSLocalizedString("common.button.ok.title", comment: ""), style:
UIAlertActionStyle.default) {
        UIAlertAction in
            alert.dismiss(animated: true, completion: nil)
    }
    alert.addAction(okAction)
}

```

```

        self.present(alert, animated: true, completion: nil)
    }

    /// Step.3 : Fungsi untuk Memeriksa status Mahasiswa terdaftar pada
    Matakuliah berdasarkan KRS
    ///
    /// - Parameters:
    /// - nim: nim
    /// - token: token
    func checkKRS(Nim nim : String,Token token : String) {
        if isReachability! {
            Alamofire.request(URL(target: PresenceAPI.checkKRS(nim,
            token))).responseJSON() {
                response in
                if let value = response.result.value as? [String: Any] {
                    if let isSuccess = value["isSuccess"] as? Bool {
                        if isSuccess {
                            self.checkKelas(Nim: nim, Token: token)
                        }else{
                            self.doPresenceFailure(Token: token)
                        }
                    }
                }
            }
        }
        }else{
            self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
            nil)
        }
    }

    /// Step.3.Func.Message : Fungsi untuk Menampilkan Pesan bahwa presensi
    Gagal
    ///
    /// - Parameters:
    /// - token: token
    func doPresenceFailure(Token token : String) {
        if isReachability! {
            Alamofire.request(URL(target:
            PresenceAPI.checkMk(token))).responseJSON() {
                response in
                let json = JSON(response.result.value!)
                if let value = json.dictionary{
                    if let result = value["Detail"]?.dictionary{
                        if result["isSuccess"]?.rawString() != false.description{
                            if let detailPresence : DetailPresence =
                            DetailPresence(JSONString: (value["Detail"]?.rawString()!)) {
                                guard detailPresence.namaMk != nil else{
                                    return
                                }
                            }
                            self.showPresenceFailure(Matakuliah:
                            detailPresence.namaMk!)
                        }
                    }
                }
            }
        }
    }
}

```



```

    }
    }else{
        self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
nil)
    }
}

/// Step.4.Message : Fungsi untuk Menampilkan Pesan bahwa presensi
berhasil
func showWrongClass(){
    let alert = UIAlertController(title: "Presensi Gagal", message:"Anda Tidak
Terdaftar dalam kelas ini.", preferredStyle: UIAlertControllerStyle.alert)
    let okAction = UIAlertAction(title:
NSBundleLocalizedString("common.button.ok.title", comment: ""), style:
UIAlertActionStyle.default) {
        UIAlertAction in
            alert.dismiss(animated: true, completion: nil)
        }
    alert.addAction(okAction)
    self.present(alert, animated: true, completion: nil)
}

/// Step.5A : Fungsi untuk Memasukan absensi Mahasiswa kategori Tepat
Waktu
///
/// - Parameters:
/// - nim: nim
/// - token: token
func insertMhsTimely(Nim nim : String,Token token : String) {
    if isReachability! {
        Alamofire.request(URL(target: PresenceAPI.insertMhsTimely(nim,
token))).responseJSON() {
            response in
                if let value = response.result.value as? [String: Any] {
                    if let isSuccess = value["isSuccess"] as? Bool {
                        if isSuccess {
                            self.doPresenceSuccess(Nim: nim)
                        }else{
                            self.insertMhsLate(Nim: nim, Token: token)
                        }
                    }
                }
            }
        }
    }else{
        self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
nil)
    }
}

/// Step.5B : Fungsi untuk Memasukan absensi Mahasiswa kategori Terlambat
///
/// - Parameters:

```



```

        self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
nil)
    }
}
/// Step.5B.Error.Message : Fungsi untuk Menampilkan Pesan bahwa presensi
Gagal
///
/// - Parameter mk: Nama Matakuliah
func showPresenceClosed(Matakuliah mk : String) {
    let alert = UIAlertController(title: "Sesi Untuk Mata Kuliah \ \(mk) Telah
Ditutup", message:" Anda Harus Datang Tepat Waktu Untuk Dapat Melakukan
Presensi Untuk Mata Kuliah \ \(mk)", preferredStyle: UIAlertControllerStyle.alert)
    let okAction = UIAlertAction(title:
NSBundleLocalizedString("common.button.ok.title", comment: ""), style:
UIAlertActionStyle.default) {
        UIAlertAction in
            alert.dismiss(animated: true, completion: nil)
        }
    alert.addAction(okAction)
    self.present(alert, animated: true, completion: nil)
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "segueDetailPresensi" ,
        let controller = segue.destination as? DetailsPresenceController {
            controller.isReachability = self.isReachability
        }
}

override func shouldPerformSegue(withIdentifier identifier: String, sender:
Any?) -> Bool {
    if identifier == "segueDetailPresensi" {
        if isReachability! {
            return true
        }else{
            self.present(AlertUtils.alertErrorConnection(), animated: true,
completion: nil)
            return false
        }
    }
    return true
}

func showErrorInput(Title title : String,Message message : String){
    let alert = UIAlertController(title: title, message: message, preferredStyle:
UIAlertControllerStyle.alert)
    let okAction = UIAlertAction(title:
NSBundleLocalizedString("common.button.ok.title", comment: ""), style:
UIAlertActionStyle.default) {
        UIAlertAction in
            alert.dismiss(animated: true, completion: nil)
            self.showInputToken()
        }
}

```



```

        if let presenceMahasiswa : PresenceMahasiwa =
PresenceMahasiwa(JSONString: item.rawQuery(!) {
            self.listOfPresence.append(presenceMahasiswa)
        }
    }
    completion(self.listOfPresence)
} else {
    self.showPresenceEmpty()
}
} else {
    self.showPresenceEmpty()
}
}
} else {
    self.present(AlertUtils.alertErrorConnection(), animated: true, completion:
nil)
}
}
}

```

c. Source Code Web service

```

/** ValidasiLogin return isSuccess */
$app->get('/presence/login/{nim}/{password}/success', function (Request $request, Response
$response) {
    $this->logger->addInfo("login query:ValidasiLogin");
    $nim = $request->getAttribute('nim');
    $password = $request->getAttribute('password');
    $mapper = new PresenceMapper();

    $status = $mapper->login($nim,$password);
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array('isSuccess' => $status));

    return $result;
});

/** ValidasiLogin return Mahasiswa */
$app->get('/presence/login/{nim}/{password}', function (Request $request, Response
$response) {
    $this->logger->addInfo("getMahasiswa query:ValidasiLogin");
    $nim = $request->getAttribute('nim');
    $password = $request->getAttribute('password');
    $mapper = new PresenceMapper();
    $mhs = $mapper->getMahasiswa($nim,$password);

    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array('Mahasiswa' => $mhs));

    return $result;
});

/** KetersediaanToken return isSuccess */
$app->get('/presence/{token}/check/success', function (Request $request, Response
$response) {
    $this->logger->addInfo("checkPresence.isSuccess query:KetersediaanToken");
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $status = $mapper->statusMk($token);
}
}
}

```

```

    $result = $response->withHeader('Content-type', 'application/json')-
    >withJson(array('isSuccess' => $status));

    return $result;
});
/** KetersediaanToken return DetailPresense */
$app->get('/presence/{token}/check', function (Request $request, Response $response) {
    $this->logger->addInfo("checkPresence query:KetersediaanToken");
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $detail = $mapper->getDetailPresence($token);
    $result = $response->withHeader('Content-type', 'application/json')->withJson(array('Detail'
=> $detail));
    return $result;
});
/** BerhasilPresensi return isSuccess */
$app->get('/presence/{nim}/success', function (Request $request, Response $response) {
    $this->logger->addInfo("isSuccessPresenceByNim query:BerhasilPresensi");
    $nim = $request->getAttribute('nim');
    $mapper = new PresenceMapper();
    $presence = $mapper->checkStatusPresence($nim);
    $result = $response->withHeader('Content-type', 'application/json')-
    >withJson(array('Presence' => $presence));

    return $result;
});
/** spValidasipresensi return isSuccess */
$app->get('/presence/{nim}/{token}/success', function (Request $request, Response
$response) {
    $this->logger->addInfo("validation.presence.isSuccess query:spValidasipresensi");
    $nim = $request->getAttribute('nim');
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $status = $mapper->validasiPresence($nim,$token);
    $result = $response->withHeader('Content-type', 'application/json')-
    >withJson(array('isSuccess' => $status));

    return $result;
});
/** kuliahToken return namaMk */
$app->get('/presence/{token}', function (Request $request, Response $response) {
    $this->logger->addInfo("inputPresenceToken query:kuliahToken");

    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $namaMk = $mapper->inputPresenceToken($token);
    $result = $response->withHeader('Content-type', 'application/json')-
    >withJson(array('namaMk' => $namaMk));
    return $result;
});
/** CekKRS return isSuccess */
$app->get('/presence/{nim}/{token}/checkkrs/success', function (Request $request, Response
$response) {
    $this->logger->addInfo("check.krs.isSuccess query:CekKRS");
    $nim = $request->getAttribute('nim');
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $status = $mapper->checkKRS($nim,$token);
    $result = $response->withHeader('Content-type', 'application/json')-
    >withJson(array('isSuccess' => $status));

    return $result;
});

```

```

/** cekkelas return isSuccess */
$app->get('/presence/{nim}/{token}/checkkelas/success', function (Request $request,
Response $response) {
    $this->logger->addInfo("check.kelas.isSuccess query:cekkelas");
    $nim = $request->getAttribute('nim');
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $status = $mapper->checkKelas($nim,$token);
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array('isSuccess' => $status));

    return $result;
});
/** InsertMhsTepatWaktu return isSuccess */
$app->get('/presence/{nim}/{token}/timely/insert', function (Request $request, Response
$response) {
    $this->logger->addInfo("insert.mhs.timely.isSuccess query:InsertMhsTepatWaktu");
    $nim = $request->getAttribute('nim');
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $status = $mapper->insertMhsTimely($nim,$token);
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array('isSuccess' => $status));

    return $result;
});
/** InsertMhsTerlambat return isSuccess */
$app->get('/presence/{nim}/{token}/late/insert', function (Request $request, Response
$response) {
    $this->logger->addInfo("insert.mhs.late.isSuccess query:InsertMhsTerlambat");
    $nim = $request->getAttribute('nim');
    $token = $request->getAttribute('token');
    $mapper = new PresenceMapper();
    $status = $mapper->insertMhsLate($nim,$token);
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array('isSuccess' => $status));

    return $result;
});
/** getCourse return */
$app->get('/course/{nim}', function (Request $request, Response $response) {
    $this->logger->addInfo("getCourse query:getCourse");
    $nim = $request->getAttribute('nim');
    $mapper = new PresenceMapper();
    $course = $mapper->getCourse($nim);
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array("Course" =>$course));

    return $result;
});
/** getCourse return */
$app->get('/presence/mahasiswa/{nim}/{courseid}', function (Request $request, Response
$response) {
    $this->logger->addInfo("getCourse query:getCourse");
    $nim = $request->getAttribute('nim');
    $courseid = $request->getAttribute('courseid');
    $mapper = new PresenceMapper();
    $presence = $mapper->getPresenceByNimAndCourseId($nim,$courseid);
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array("PresenceMahasiswa" =>$presence));

    return $result;
});

```

```

$app->get('/presence/app/version', function (Request $request, Response $response) {
    $this->logger->addInfo("getAppVersion query:getAppVersion");
    $mapper = new PresenceMapper();
    $app = $mapper->getAppVersion("PREMI");
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array("Application" =>$app));

    return $result;
});
$app->get('/presence/dosen/app/version', function (Request $request, Response $response) {
    $this->logger->addInfo("getAppVersion query:getAppVersion");
    $mapper = new PresenceMapper();
    $app = $mapper->getAppVersion("PREDI");
    $result = $response->withHeader('Content-type', 'application/json')-
>withJson(array("Application" =>$app));

    return $result;
});

$container["logger"] = function($c) {
    $logger = new \Monolog\Logger("my_logger");
    $file_handler = new \Monolog\Handler\StreamHandler("../logs/app.log");
    $logger->pushHandler($file_handler);
    return $logger;
};
$app->run();

```

Lampiran 2 Proses Pengajuan Aplikasi ke *iOS App Store*

iTunes Connect My Apps ▼ UMY Hadir - Mahasiswa ▼ Eko Prasetyo ▼ Eko Prasetyo ?

App Store Features TestFlight **Activity** App Analytics Sales and Trends

IOS HISTORY

All Builds

App Store Versions

RATINGS AND REVIEWS

iOS App

iOS App Store Versions

Versions shown below are your App Store Versions.

Version 1.0.1

Activity	User	Date
● Ready for Sale	Apple	Dec 7, 2017 at 6:05 AM
● In Review	Apple	Dec 6, 2017 at 11:12 PM
● Waiting for Review	com	Dec 4, 2017 at 6:02 PM
● Metadata Rejected	Apple	Dec 3, 2017 at 4:30 AM
● In Review	Apple	Dec 3, 2017 at 4:02 AM
● Waiting for Review	:om	Nov 30, 2017 at 9:00 PM
● Metadata Rejected	Apple	Nov 29, 2017 at 9:00 AM

● In Review	Apple	Nov 29, 2017 at 8:17 AM
● Waiting for Review	com	Nov 25, 2017 at 7:15 PM
● Prepare for Submission	com	Nov 24, 2017 at 1:34 PM

Copyright © 2017 Apple Inc. All rights reserved. | [Terms of Service](#) | [Privacy Policy](#) | [Contact Us](#)



App Store

Features

TestFlight

Activity

App Analytics

Sales and Trends

iOS HISTORY

All Builds

App Store Versions

RATINGS AND REVIEWS

iOS App

iOS App Store Versions

Versions shown below are your App Store Versions.

▾ Version 2.0.1

Activity	User	Date
● Ready for Sale	Apple	Dec 15, 2017 at 11:18 PM
● In Review	Apple	Dec 15, 2017 at 10:46 PM
● Waiting for Review	com	Dec 13, 2017 at 5:46 PM
● Prepare for Submission	l.com	Dec 13, 2017 at 3:33 PM

> Version 1.0.1