

Lampiran Kode Program

File: app.js

```
var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var fileUpload = require('express-fileupload');
var bodyParser = require('body-parser');
var flash = require('express-flash');
var session = require('express-session');
var index = require('./routes/index');
var compression = require('compression');
var lecturer = require('./routes/lecturer');
var student = require('./routes/student');
var expressValidator = require('express-validator');
var methodOverride = require('method-override');
var connection = require('express-myconnection');
var mysql = require('mysql');

var app = express();
// view engine setup

app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

// uncomment after placing your favicon in /public
app.use(compression())
app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({
  extended: false
}));
app.use(session({
  secret: "12345",
  resave: false,
  saveUninitialized: false
}));

var maxSize = 100000000;
app.use(flash());
app.use(fileUpload({
```

```

    limits: {
      fileSize: maxSize
    },
  }));
app.use(expressValidator());
app.use(methodOverride(function (req, res) {
  if (req.body && typeof req.body == 'object' && '_method' in
req.body) {
    var method = req.body._method;
    delete req.body._method;
    return method;
  }
}));

```

```

app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));
app.use('/jotted', express.static(__dirname +
'/node_modules/jotted/'));
app.use('/codemirror', express.static(__dirname +
'/node_modules/codemirror/'));
app.use('/sweetalert2', express.static(__dirname +
'/node_modules/sweetalert2/'));
app.use('/jquery-comments', express.static(__dirname +
'/node_modules/jquery-comments/'));

```

```

/*-----
   connection peer, register as middleware
   type koneksi : single,pool and request
   -----*/

```

```

app.use(
  connection(mysql, {
    host: 'localhost',
    user: 'root', // your mysql user
    password: '', // your mysql password
    port: 3306, //port mysql
    database: 'subhanrn_vlab' // your database name
  }, 'pool') //or single
);

```

```

app.use('/', index);
// app.use('/customers', customers);
app.use('/student', student);
app.use('/lecturer', lecturer);
// app.use('/users', users);

```

```

// catch 404 and forward to error handler
app.use(function (req, res, next) {
  var err = new Error('Not Found');
  err.status = 404;
  next(err);
});

// error handler
app.use(function (err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err :
  {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});

process.on('uncaughtException', function (err) {
  console.error(err);
  console.log("Node NOT Exiting...");
});

module.exports = app;

```

File: authentication.js

```

var Auth = {
  is_login: function (req, res, next) {
    if (!req.session.is_login) {
      req.flash('msg_error', "Please log in first before
access this area!");
      return res.redirect('/login');
    }
    next();
  },
  is_student: function (req, res, next) {
    if (!req.session.is_student) {
      req.flash('msg_error', "Your are not allowed to access
this student area!");
      return res.redirect('/login');
    }
    next();
  }
};

```

```

    },
    is_lecturer: function (req, res, next) {
        if (!req.session.is_lecturer) {
            req.flash('msg_error', "Your are not allowed to access
this lecturer area!");
            return res.redirect('/login');
        }
        next();
    }
};
module.exports = Auth;

```

File: index.js

```

var express = require('express');
var router = express.Router();
// var modul = require('../modul/modul');
var request = require('request');
var md5 = require('md5');
var authentication_md1 = require('../middlewares/authentication');
var svgCaptcha = require('svg-captcha-smooth');
var randomstring = require('randomstring');
var dateTime = require('node-datetime');

var session_store;

/* GET home page. */
router.get('/', function (req, res, next) {
    res.redirect('/login');
});
router.get('/captcha', function (req, res, next) {
    var captcha = svgCaptcha.create({
        noise: 5
    });
    req.session.captcha = captcha.text;
    res.set('Content-Type', 'image/svg+xml');
    res.status(200).send(captcha.data);
    // console.log(captcha.text);
});
router.get('/login', function (req, res, next) {
    if (req.session.is_student) {
        res.redirect('/student/compiler/frontend');
    }
});

```

```

    } else if (req.session.is_lecturer) {
        res.redirect('/lecturer/class');
    } else {
        res.render('main/login', {
            title: "Login Page"
        });
        // var random = randomstring.generate({
        //   length: 6,
        //   charset: '!@#$$%&01234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ'
        // });
        // console.log(random);

        // var dt = dateTime.create();
        // var formatted = dt.format('Y-m-d H:M:S');
        // console.log(formatted);
    }
});
router.post('/login', function (req, res, next) {
});

router.get('/logout', function (req, res) {
    req.session.destroy(function (err) {
        if (err) {
            console.log(err);
        } else {
            res.redirect('/login');
        }
    });
});
});

module.exports = router;

```

File: student.js

```

var express = require('express');
var router = express.Router();
var dateTime = require('node-datetime');
var authentication_md1 = require('../middlewares/authentication');
// var session_store;

router.get('/compiler/frontend', authentication_md1.is_student,
function (req, res, next) {
    res.render('student/frontend', {
        username: req.session.username,

```

```

        name: req.session.name,
        year: req.session.year,
    });
});

router.get('/compiler/backend', authentication_mdl.is_student,
function (req, res, next) {
    res.render('student/backend', {
        hostname: req.hostname,
        username: req.session.username,
        name: req.session.name,
        year: req.session.year
    });
});

router.get('/profile', authentication_mdl.is_student, function (req,
res) {
    res.render('student/profile', {
        title: 'Profile Page',
        level: req.session.level,
        address: req.session.address,
        phone: req.session.phone,
        username: req.session.username,
        name: req.session.name,
        majors: req.session.majors,
        year: req.session.year,
    });
});
});

```

File: lecturer.js

```

var express = require('express');
var md5 = require('md5');
var fs = require('fs');
var dateTime = require('node-datetime');
var randomstring = require('randomstring');
var router = express.Router();
var authentication_mdl = require('../middlewares/authentication');

router.get('/compiler/frontend', authentication_mdl.is_lecturer,
function (req, res, next) {
    res.render('lecturer/frontend', {
        username: req.session.username,
        name: req.session.name,
        session_store: req.session
    });
});

```

```

    });
});

router.get('/compiler/backend', authentication_mdl.is_lecturer,
function (req, res, next) {
    res.render('lecturer/backend', {
        hostname: req.hostname,
        username: req.session.username,
        name: req.session.name,
        session_store: req.session
    });
});

router.get('/class/tasks/(:class_id)/add',
authentication_mdl.is_lecturer, function (req, res, next) {
    req.getConnection(function (err, connection) {
        var query = connection.query('SELECT * FROM class where
class_id="' + req.params.class_id + "'", function (err, rows) {
            res.render('lecturer/add-task', {
                param_class_id: req.params.class_id,
                username: req.session.username,
                name: req.session.name,
                session_store: req.session
            });
        });
    });
});
});
});

```