

DAFTAR PUSTAKA

- Aswad, Iqra dkk. 2017. “Pengembangan Media Proxy untuk Mendukung Komunikasi Real Time Berbasis Web (WebRTC)”. Universitas Hasanuddin, Indonesia.
- Batubara, Febrin Aulia. 2015. “Perancangan Website Pada PT.Ratu Enim Palembang”. Politeknik Negeri Medan, Indonesia. 15-27.
- Butkiewicz, Michael. 2011. “Understanding Website Complexity: Measurements, Metrics, and Implications Categories and Subject Descriptors”. Internet Measurement Conference (IMC).
- Dzacko, Haidar. 2007. “1. Basis Data (Database)”.
- Gelens, dkk. 2014. “gevent-socketio Documentation (Release 0.3.1). Diakses pada 10 Maret 2020 dari <https://media.readthedocs.org/pdf/gevent-socketio/latest/gevent-socketio.pdf> .
- Hannonen, Daria. 2017. “Development of Website Solution for Association to Assist Young Professionals”.
- Julisman, Agung. 2014. “Bikin Aplikasi Android Dengan Angular Mobile MongoDB : Studi Kasus Membuat Chat ala BBM dan Whatsapp”. Bandung. Lokomedia.
- Kurento, 2019. “One to Many Video Call”. Diakses pada 08 Agustus 2020 dari <https://doc-kurento.readthedocs.io/en/6.13.0/tutorials/node/tutorial-one2many.html>

- Levent, Tsahi. "How Many Sessions Can A Kurento Server Hold?". Diakses pada 01 Agustus 2020 dari <https://testrtc.com/sessions-kurento-server/>
- Lopez, Javier dkk. 2013. "Kurento: a Media Server Technology for Convergent WWW/Mobile Real-time Multimedia Communications Supporting WebRTC". Institute of Electrical and Electronics Engineers Conference (IEEE).
- Maesya, Aries. 2010. "*Learning Management System Berbasis Synchronous E-Learning*". Universitas Pakuan. Indonesia.
- Muhammad Agung Rizkyana, R.Sandhika Galih Amalga. 2014. "Rancangan Arsitektur Aplikasi Pengumpulan Tugas Dengan Menggunakan Push Notification". (SemnasIF):70-75
- No, Vol dan Apri Junaidi. 2016. "Studi Perbandingan Antara MongoDB dan MySQL dalam Lingkup Big Data". 2(1):460-65
- Ngako, Kgothatso. 2018. "Secure IP Camera Video Streaming Through Kurento Media Server". Council for Scientific and Industrial Research, Pretoria, South Africa.
- Permana, Endang Cahya. 2016. "Penulisan Fungsi Pada Javascript". 1-27.
- Pimentel, Victoria dan Bradford G. Nickerson. 2012. "Communication and Displaying Real-time Data with WebSocket". IEEE Internet Computing 16(4): 45-53.

Rahmanda, Rahadiyan Yuniar dkk. 2018. “Perancangan dan Implementasi Kelas Virtual FILKOM Universitas Brawijaya dengan Memanfaatkan Teknologi WebRTC (Web Real-time Communication)”. Jurnal Pengembangan Teknologi dan Ilmu Komputer. e-ISSN : 2548-964X

Tunardi, Yovita dan Rita Layona. 2019. “NoSQL Technology in Android Based Mobile Chat Application Using MongoDB”. Departemen Ilmu Komputer, Universitas Binus, Indonesia.

Virag, Ioan. 2015. “A Survey of Web Based Medical Imaging Applications”. Politechnica University of Timisora, Romania.

Wang, V., Salim, Moskovits. 2013. “The Definitive Guide to HTML5 WebSocket”. p.208.

LAMPIRAN

LAMPIRAN 1

Source code app.js (web server)

```
/**
 * Module dependencies.
 */
const express = require('express');
const compression = require('compression');
const session = require('express-session');
const bodyParser = require('body-parser');
const logger = require('morgan');
const chalk = require('chalk');
const errorHandler = require('errorhandler');
const lusca = require('lusca');
const dotenv = require('dotenv');
const MongoStore = require('connect-mongo')(session);
const flash = require('express-flash');
const path = require('path');
const mongoose = require('mongoose');
const passport = require('passport');
const expressValidator = require('express-validator');
const sass = require('node-sass-middleware');
const multer = require('multer');
var minimist = require('minimist');
var kurento = require('kurento-client');
var fs = require('fs');
var url = require('url');
var https = require('https');

var argv = minimist(process.argv.slice(2), {
  default: {
    as_uri: 'https://spellide.unhas.ac.id/',
```

```

        ws_uri: 'ws://127.0.0.1:8888/kurento'
    }
});
dotenv.load({ path: '.env.example' });
/**
 * Controllers (route handlers).
 */
const homeController = require('./controllers/home');
const userController = require('./controllers/user');
const latihanController = require('./controllers/latihan');
// const apiController = require('./controllers/api');
const contactController = require('./controllers/contact');
const classController = require('./controllers/class');
const exerciseController = require('./controllers/exercise');
const courseController = require('./controllers/course');
const workController = require('./controllers/work');
const runController = require('./controllers/run');
const runlogController = require('./controllers/runlog');
const testlogController = require('./controllers/testlog');
const socketioController = require('./controllers/socketio');
/**
 * API keys and Passport configuration.
 */
const passportConfig = require('./config/passport');
/**
 * Create Express server.
 */
const app = express();

//SET URL PORT
var asUrl = url.parse(argv.as_uri);
var port = 5000;
/**
 * Connect to MongoDB.

```

```

*/
mongoose.Promise = global.Promise;
mongoose.connect(process.env.MONGODB_URI);
mongoose.connection.on('error', (err) => {
  console.error(err);
  console.log('%s MongoDB connection error. Please make sure MongoDB is running.',
chalk.red('X'));
  process.exit();
});

/**
 * Express configuration.
 */
app.set('host', process.env.OPENSIFT_NODEJS_IP || '0.0.0.0');
// app.set('port', process.env.PORT || process.env.OPENSIFT_NODEJS_PORT || 8080);
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'pug');
//app.use(expressStatusMonitor());
app.use(compression());
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(expressValidator());
app.use(session({
  resave: true,
  saveUninitialized: true,
  secret: process.env.SESSION_SECRET,
  store: new MongoStore({
    url: process.env.MONGODB_URI,
    autoReconnect: true,
    clear_interval: 3600
  })
}));
app.use(passport.initialize());
app.use(passport.session());

```

```

app.use(flash());
app.use((req, res, next) => {
  if (req.path === '/api/upload') {
    next();
  } else {
    // lusca.csrf()(req, res, next);
    next();
  }
});
app.use((req, res, next) => {
  res.locals.user = req.user;
  next();
});
app.use((req, res, next) => {
  // After successful login, redirect back to the intended page
  if (!req.user &&
    req.path !== '/login' &&
    req.path !== '/signup' &&
    !req.path.match(/^\/auth/) &&
    !req.path.match(/\.\/)) {
    req.session.returnTo = req.originalUrl;
  } else if (req.user &&
    req.path === '/account') {
    req.session.returnTo = req.originalUrl;
  }
  next();
});
app.use(express.static(path.join(__dirname, 'public'), { maxAge: 31557600000 }));
app.set('trust proxy', ['loopback', 'linklocal', 'uniquelocal']);
/**
 * Primary app routes.
 */
app.get('/', homeController.index);
app.get('/login', userController.getLogin);

```

```

app.get('/latihan', latihanController.latihan);
app.post('/login', userController.postLogin);
app.get('/logout', userController.logout);
app.get('/forgot', userController.getForgot);
app.post('/forgot', userController.postForgot);
app.get('/reset/:token', userController.getReset);
app.post('/reset/:token', userController.postReset);
app.get('/signup', userController.getSignup);
app.get('/signup-roles', userController.getSignupRoles);
app.post('/signup', userController.postSignup);
app.get('/contact', contactController.getContact);
app.post('/contact', contactController.postContact);
app.get('/account', passportConfig.isAuthenticated, userController.getAccount);
app.post('/account/profile', passportConfig.isAuthenticated,
userController.postUpdateProfile);
app.post('/account/password', passportConfig.isAuthenticated,
userController.postUpdatePassword);
app.post('/account/password-by-staff', passportConfig.isAuthenticated,
userController.postUpdatePasswordByStaff);
app.post('/account/delete', passportConfig.isAuthenticated,
userController.postDeleteAccount);
app.get('/account/unlink/:provider', passportConfig.isAuthenticated,
userController.getOauthUnlink);
/**
 * SPELL routes
 */
app.get('/class', passportConfig.isAuthenticated, classController.index);
app.post('/class', passportConfig.isAuthenticated, classController.create);
app.get('/class/:classId', passportConfig.isAuthenticated, classController.getById);
app.put('/class/:classId', passportConfig.isAuthenticated, classController.updateById);
app.get('/class/delete/:classId', passportConfig.isAuthenticated,
classController.deleteById);
app.get('/classedit/:classId', passportConfig.isAuthenticated, classController.editById);

app.get('/classlog', passportConfig.isAuthenticated, classController.logIndex);
app.get('/classlog/:classId', passportConfig.isAuthenticated,
classController.getLogsById);

```



```

app.get('/classlog-stats/:classId/:timestamp', /*passportConfig.isAuthenticated,*/
classController.getLogsStatsById);

app.get('/class-staff/:classId', passportConfig.isAuthenticated,
classController.getStaffworkById);

app.get('/student', passportConfig.isAuthenticated, exerciseController.indexByStudent);
app.get('/exercise', passportConfig.isAuthenticated, exerciseController.index);
app.get('/exercise/:exerciseId', passportConfig.isAuthenticated,
exerciseController.getById);
app.put('/exercise/:exerciseId', passportConfig.isAuthenticated,
exerciseController.updateById);
// app.get('/exercise/:exerciseId', passportConfig.isAuthenticated,
exerciseController.getExercise);
app.post('/exercise', passportConfig.isAuthenticated, exerciseController.create);
app.get('/course', passportConfig.isAuthenticated, courseController.index);
app.get('/course/livecourses/:courseId', passportConfig.isAuthenticated,
courseController.livecourses);
app.get('/course/tesUpload', passportConfig.isAuthenticated, courseController.tesUpload);
app.post('/course/upload/:courseId', passportConfig.isAuthenticated,
courseController.doUpload);
app.get('/course/download/:namaFile', passportConfig.isAuthenticated,
courseController.downloadFile);
app.post('/course/saveToDb', passportConfig.isAuthenticated,
courseController.saveFileToDb);

app.get('/course/:courseId', passportConfig.isAuthenticated, courseController.getById);
app.get('/studentcourses', passportConfig.isAuthenticated,
courseController.indexByStudent);
app.get('/course/student/:courseId', passportConfig.isAuthenticated,
courseController.getByStudent);
app.put('/course/:courseId', passportConfig.isAuthenticated, courseController.updateById);
app.get('/course/delete/:courseId', passportConfig.isAuthenticated,
courseController.deleteById);
app.get('/course/preview/:courseId', passportConfig.isAuthenticated,
courseController.previewModuleById);
app.get('/course/preview/:courseId/:moduleId', passportConfig.isAuthenticated,
courseController.previewModuleById2);
app.post('/course', passportConfig.isAuthenticated, courseController.create);
app.post('/course/:courseId', passportConfig.isAuthenticated,
courseController.createModule);

```

```

app.get('/course/exercise/:courseId', passportConfig.isAuthenticated,
courseController.getExerciseByCourse);

app.post('/course/exercise/:courseId', passportConfig.isAuthenticated,
courseController.createExercise);

//classes

app.get('/course/class/:courseId', passportConfig.isAuthenticated,
classController.index2);

//module

app.get('/course/:courseId/:moduleId', passportConfig.isAuthenticated,
courseController.getModuleById);

app.get('/course/delete/:courseId/:moduleId', passportConfig.isAuthenticated,
courseController.deleteModuleById);

app.put('/course/:courseId/:moduleId', passportConfig.isAuthenticated,
courseController.updateModuleById);

//excercises

app.get('/course/exercise/:courseId/:exerciseId', passportConfig.isAuthenticated,
courseController.getExerciseById);

app.put('/course/exercise/:courseId/:exerciseId', passportConfig.isAuthenticated,
courseController.updateExerciseById);

app.get('/work/class/:classId', passportConfig.isAuthenticated,
workController.indexByClass);

app.get('/work/student/:studentId', passportConfig.isAuthenticated,
workController.indexByStudent);

app.get('/work/exercise/:exerciseId', passportConfig.isAuthenticated,
exerciseController.indexByStudentMiddleware, workController.getByExercise);

app.get('/work/student-exercise/:studentId/:exerciseId', passportConfig.isAuthenticated,
workController.getByStudentExercise);

app.post('/work', passportConfig.isAuthenticated, workController.create);

app.put('/work/:workId', passportConfig.isAuthenticated, workController.updateById);

app.post('/worklog', passportConfig.isAuthenticated, workController.addWorkLog)

app.get('/run/work/:workId', passportConfig.isAuthenticated, runController.getByWork);

// app.get('/run/test/:runId', passportConfig.isAuthenticated,
runController.getTestByRun);

app.get('/run/test/:workId', passportConfig.isAuthenticated, runController.getTestByWork);

```

```

app.post('/runlog', passportConfig.isAuthenticated, runlogController.create);
app.post('/testlog', passportConfig.isAuthenticated, testlogController.create);

app.use(errorHandler());
/**
 * Start Express server.
 */
appServer = require('http').createServer(app);
appServer.listen(port, () => {
  console.log('Speel IDE With Kurento');
  console.log(`Open ${url.format(asUrl)} with a WebRTC Capable Browser`);
});
var io = require('socket.io').listen(appServer);
var redis = require('socket.io-redis');
io.adapter(redis({host:'127.0.0.1', port:6379}));
socketioController.set(io);
socketioController.start();
//Var
var candidatesQueue = {};
var kurentoClient = null;
var presenter = null;
var noPresenterMessage = 'No active presenter. Try again later...';
var anotherPresenterIsActive = "Another user is currently acting as presenter. Try again later ...";
var rooms = [];
//Fungsi Untuk Room
getRoom = (socket) => {
  if (rooms[socket.room] == undefined) {
    createRoom(socket.room);
  }
  return rooms[socket.room];
};
createRoom = (room) => {

```

```

rooms[room] = {
  presenter: null,
  pipeline: null,
  viewers: [],
  chat: []
};
};

joinRoom = (socket, data) => {
  // Melepas semua socket id room lainnya
  while(socket.rooms.length) {
    socket.leave(socket.rooms[0]);
  }

  // join socket.io room baru
  socket.join(data.room);
  socket.room = data.room;
  socket.username = data.username;

  // socket.emit('joinedRoom');
  console.log(`Joined on Room : ${data.room} with id : ${data.username}`);
};

//Method Chat Message
newChatMessage = (socket, message) => {
  var message = {message:message.message, username: message.username};
  var data = message.username + ' : ' + message.message;
  io.in(socket.room).emit('newChatMessage', data);

  console.log(`message from username ${message.username} with message :
  ${message.message}`);
};

//Method File Message
newFileMessage = (socket, message) => {
  var namaFile = message.namaFile;

```

```

var sender = message.sender;

io.in(socket.room).emit('newFileMessage', message);

console.log(`${sender} sending file with name ${namaFile} to others`);
};

typingMessage = (socket, message) => {
  io.in(socket.room).emit('typing', message);
};

//End Fungsi Untuk Room

//Fungsi Respon dari WebSocket

acceptPeerResponse = (peerType, sdpAnswer) => {
  return {
    id : peerType + 'Response',
    response : 'accepted',
    sdpAnswer : sdpAnswer
  };
};

rejectPeerResponse = (peerType, reason) => {
  return {
    id : peerType + 'Response',
    response : 'rejected',
    message : reason
  };
};

//End Fungsi Respon

//Manage Socket IO

io.on('connection', (socket) => {
  console.log(`-----Connection received with sessionID : ${socket.id}-----`);

```

```

//listening error
socket.on('error', (error) => {
  console.error('Connection ' + socket.id + ' error', error);
  stop(socket);
});

//listening disconnect
socket.on('disconnect', () => {
  console.log('Connection ' + socket.id + ' closed');
  stop(socket);
});

//listening presenter request
socket.on('presenter', (data) => {
  startPresenter(socket, data.sdpOffer, (error, sdpAnswer) => {
    var response = (error) ? rejectPeerResponse('presenter', error) :
acceptPeerResponse('presenter', sdpAnswer);
    socket.emit(response.id, response);
    if (!error) {
      console.log(socket.username + ' starting publishing to ' + socket.room + ' room');
      socket.broadcast.emit('streamStarted');
    }
  });
});

//listening viewer
socket.on('viewer', (data) => {
  startViewer(socket, data.sdpOffer, (error, sdpAnswer) => {
    response = (error) ? rejectPeerResponse('viewer', error) :
acceptPeerResponse('viewer', sdpAnswer);
    socket.emit(response.id, response);
  });
});

//listening stop

```

```

socket.on('stop', () => {
    stop(socket);
});

//listening onIceCandidate
socket.on('onIceCandidate', (data) => {
    onIceCandidate(socket, data.candidate);
});

//listening masuk stream room
socket.on('subscribeToStream', (data) => {
    msg = {
        room:data.room,
        username:socket.id
    };
    joinRoom(socket, msg);
    var room = getRoom(socket);
    if (room.presenter) {
        socket.emit('streamStarted');
    }
});

//Listening Chat
socket.on('chatMessage', (message) => {
    newChatMessage(socket, message);
});

//listening FileChat
socket.on('fileMessage', (message) => {
    newFileMessage(socket, message);
});

//Listening OnTyping or Not
socket.on('typing', (message) => {

```

```

        typingMessage(socket, message);
    });

    //Listening JoinRoom
    socket.on('joinRoom', (data) => {
        joinRoom(socket, data);
    });
});

//END Manage Socket Io

//Method yang di lakukan listening socket
//Recover kurentoClient pertama kali
getKurentoClient = (callback) => {
    if (kurentoClient !== null) {
        return callback(null, kurentoClient);
    }
    kurento(argv.ws_uri, (error, _kurentoClient) => {
        if (error) {
            console.log("Could not find media server at address " + argv.ws_uri);
            return callback("Could not find media server at address" + argv.ws_uri
                + ". Exiting with error " + error);
        }
        kurentoClient = _kurentoClient;
        callback(null, kurentoClient);
    });
};

//Method Start Presenter
startPresenter = (socket, sdpOffer, callback) => {
    clearCandidatesQueue(socket);
    var room = getRoom(socket);
    if (room.presenter !== null) {

```



```

    stop(socket);
    return callback(anotherPresenterIsActive);
}

room.presenter = {
  webRtcEndpoint : null,
  id: socket.id
};

getKurentoClient((error, kurentoClient) => {
  if (error) {
    stop(socket);
    return callback(error);
  }
  if (room.presenter === null) {
    stop(socket);
    return callback(noPresenterMessage);
  }
  kurentoClient.create('MediaPipeline', (error, pipeline) => {
    if (error) {
      stop(socket);
      return callback(error);
    }

    if (room.presenter === null) {
      stop(socket);
      return callback(noPresenterMessage);
    }
    room.pipeline = pipeline;
    pipeline.create('WebRtcEndpoint', (error, webRtcEndpoint) => {
      if (error) {
        stop(socket);
        return callback(error);
      }
    }
  }
}

```

```

    if (room.presenter === null) {
        stop(socket);
        return callback(noPresenterMessage);
    }
    room.presenter.webRtcEndpoint = webRtcEndpoint;
    if (candidatesQueue[socket.id]) {
        while(candidatesQueue[socket.id].length) {
            var candidate = candidatesQueue[socket.id].shift();
            webRtcEndpoint.addIceCandidate(candidate);
        }
    }

    webRtcEndpoint.on('OnIceCandidate', (event) => {
        var candidate = kurento.register.complexTypes.IceCandidate(event.candidate);
        socket.emit('iceCandidate', { candidate : candidate });
    });
    webRtcEndpoint.processOffer(sdpOffer, (error, sdpAnswer) => {
        if (error) {
            stop(socket);
            return callback(error);
        }
        if (room.presenter === null) {
            stop(socket);
            return callback(noPresenterMessage);
        }
        callback(null, sdpAnswer);
    });
    webRtcEndpoint.gatherCandidates(function(error) {
        if (error) {
            stop(socket);
            return callback(error);
        }
    });
});

```

```

    });
  });
};
//Method Start Viewer
startViewer = (socket, sdpOffer, callback) => {
  clearCandidatesQueue(socket);
  var room = getRoom(socket);
  if (room.presenter === null) {
    stop(socket);
    return callback(noPresenterMessage);
  }
  room.pipeline.create('WebRtcEndpoint', (error, webRtcEndpoint) => {
    if (error) {
      stop(socket);
      console.log('GAGAL KI !!');
      return callback(error);
    }
    room.viewers[socket.id] = {
      "webRtcEndpoint" : webRtcEndpoint,
      "socket" : socket
    };
    if (room.presenter === null) {
      stop(socket);
      return callback(noPresenterMessage);
    }
    if (candidatesQueue[socket.id]) {
      while(candidatesQueue[socket.id].length) {
        var candidate = candidatesQueue[socket.id].shift();
        webRtcEndpoint.addIceCandidate(candidate);
      }
    }

    webRtcEndpoint.on('OnIceCandidate', (event) => {
      var candidate = kurento.register.complexTypes.IceCandidate(event.candidate);

```

```

        socket.emit('iceCandidate', { candidate : candidate });
    });

    webRtcEndpoint.processOffer(sdpOffer, (error, sdpAnswer) => {
        if (error) {
            stop(socket.id);
            return callback(error);
        }
        if (room.presenter === null) {
            stop(socket.id);
            return callback(noPresenterMessage);
        }

        room.presenter.webRtcEndpoint.connect(webRtcEndpoint, (error) => {
            if (error) {
                stop(socket.id);
                return callback(error);
            }
            if (room.presenter === null) {
                stop(socket.id);
                return callback(noPresenterMessage);
            }
            callback(null, sdpAnswer);
            webRtcEndpoint.gatherCandidates((error) => {
                if (error) {
                    stop(socket.id);
                    return callback(error);
                }
            });
        });
    });
});
});
};

```

```

//Method Clear Antrian Candidate
clearCandidatesQueue = (socket) => {
  if (candidatesQueue[socket.id]) {
    delete candidatesQueue[socket.id];
  }
};

//Method Stop
stop = (socket) => {
  var room = getRoom(socket);
  if (room.presenter !== null && room.presenter.id == socket.id) {
    stopPresenter(socket);
  } else if (room.viewers[socket.id]) {
    stopViewing(socket);
  }
};

//Stop Presenter
stopPresenter = (socket) => {
  var room = getRoom(socket);
  var viewers = room.viewers;

  for (var i in viewers) {
    var viewer = viewers[i];
    if (viewer.socket) {
      clearCandidatesQueue(socket);
      viewer.webRtcEndpoint.release();
      viewer.socket.emit('stopCommunication');
    }
  }

  room.presenter.webRtcEndpoint.release();
  room.presenter = null;
  room.pipeline.release();
};

```

```

    room.viewers = [];
};

//Method Stop Viewer
stopViewing = (socket) => {
    var room = getRoom(socket);
    clearCandidatesQueue(socket.id);
    room.viewers[socket.id].webRtcEndpoint.release();
    delete room.viewers[socket.id];
};

//Method onIceCandidate
onIceCandidate = (socket, _candidate) => {
    var room = getRoom(socket);
    var candidate = kurento.register.complexTypes.IceCandidate(_candidate);
    if (room.presenter && room.presenter.id === socket.id && room.presenter.webRtcEndpoint)
    {
        console.info('Sending presenter candidate');
        room.presenter.webRtcEndpoint.addIceCandidate(candidate);
    }
    else if (room.viewers[socket.id] && room.viewers[socket.id].webRtcEndpoint) {
        console.info('Sending viewer candidate');
        room.viewers[socket.id].webRtcEndpoint.addIceCandidate(candidate);
    }
    else {
        console.info('Queueing candidate');
        if (!candidatesQueue[socket.id]) {
            candidatesQueue[socket.id] = [];
        }
        candidatesQueue[socket.id].push(candidate);
    }
};
module.exports = app;

```

LAMPIRAN 2

Source Code layout.pug

```
doctype html
html(lang='en')

  head
    meta(charset='utf-8')
    meta(http-equiv='X-UA-Compatible', content='IE=edge')
    meta(name='viewport', content='width=device-width, initial-scale=1.0')
    meta(name='theme-color' content='#4DA5F4')
    meta(name='csrf-token', content=_csrf)
    title #{title} - UbiconLabs
    link(rel='shortcut icon', href='/favicon.png')
    link(rel='stylesheet', href='/css/main.css')
    link(rel='stylesheet', href='/jq-ui/custom/jquery-ui.css')
    link(rel='stylesheet', href='/summernote/dist/summernote-lite.css')
    link(rel='stylesheet', href='/css/modifsummernote.css')
    //- link(rel='stylesheet', href='/summernote/dist/summernote.css')
    link(rel='stylesheet', href='/summernote-prettyprint-
plugin/bower_components/summernote/dist/summernote.css')
    link(rel='stylesheet', href='/summernote-prettyprint-plugin/src/css/prism.css')
    link(rel='stylesheet', href='/summernote-prettyprint-
plugin/bower_components/prism/themes/prism-coy.css')
    link(rel='stylesheet', href='/css/cardbootstrap.css')
    link(rel='stylesheet', href='/css/btnoutlinebootstrap.css')

  style(type='text/css').
    a {
      color : #18BC9C;
    }
    a:hover {
      text-decoration : none;
      color : #057a63;
```

```

    }
    a:click {
        text-decoration : none;
    }
    .badge {
        background-color : #18bc9c;
    }
    .badge-red {
        background-color : #e74c3c;
    }
    .box-shadow { box-shadow: 0 .25rem .75rem rgba(0, 0, 0, .05); }

block page-style

body
  if user
    include partials/header

  include partials/flash
  block content

  include partials/footer

  script(src='/js/lib/jquery-3.1.1.min.js')
  //- script(src='https://code.jquery.com/ui/1.12.1/jquery-ui.js')
  script(src='/jq-ui/custom/jquery-ui.min.js')
  script(src='/js/lib/bootstrap.min.js')

script(src='https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js')
  script(src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js",
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM",
crossorigin="anonymous")
  script(src='/summernote/dist/summernote.js')
  script(src='https://cdnjs.cloudflare.com/ajax/libs/summernote/0.8.11/summernote-
lite.js')

```



```

script(src='/summernote-prettyprint-plugin/src/js/prism.js')
script(src='/summernote-prettyprint-plugin/src/js/summernote-code-beautify-plugin.js')

block end-script

// Google Analytics: change UA-XXXXX-X to be your site's ID
script.

```

LAMPIRAN 3

Source Code livecourse.pug

```

extends ../layout
block content
  .container
    h2 Live Class
    h4 #{title}
    input#name(style='display:none;' value=user.profile.name)
    input#level(style='display:none;' value=user.role)
    input#idCourse(style='display:none;' value=idCourse)
    input#idUser(style='display:none;' value=user.id)
    input#author(style='display:none;' value=author)
    if(user.role == 'teacher')
      a#terminate.badge.badge-red End Class
  .container(style='')
    video#video-webcam(controls='' autoplay='' playsinline='' style='width:
675px;height:400px;border-radius:5px;border:1px solid black;margin-top:20px;')
    .container#messages(style="overflow:scroll; height:400px;width:40%;float:right;margin-
top:20px;")
    br
    if(user.role == 'teacher')
      button#btnRTC.col-sm-12.col-md-2.btn.btn-primary(style='margin-right:1px')
        i.fa.fa-video-camera &nbsp; Broadcast Your Video
      button#btnScreen.col-sm-12.col-md-2.btn.btn-primary(style='margin-right:1px')
        i.fa.fa-share &nbsp; Share Your Screen
    if (user.role == 'student')

```

```

        button#btnConnect.col-sm-12.col-md-2.btn.btn-primary(style='margin-right:10px')
            i.fa.fa-cloud &nbsp; Connect Now

        button#btnSubmit.col-xs-4.col-md-1.btn.btn-danger(type='submit'           style="margin-
right:1px;")
            i.fa.fa-send &nbsp; Send

        input.col-md-2#myFile(type='file'           style='margin-right:1px;height:45px;margin-
top:12px;margin-left:-12px;')

        .form-group
            input.form-control#txt(type='text'           required           placeholder='Type           Here... '
style="width:40%;float:right")

    append end-script

    script(src='https://unpkg.com/axios/dist/axios.min.js')
    script(src='/socket/socket.io.js')
    //- script(src=`/jq-ui/jquerymain.js?v=${randomNumber}`)
    script(src='https://www.webrtc-experiment.com/common.js')
    script(src=`/bower_components/webrtc-adapter/release/adapter.js?v=${randomNumber}`)
    script(src=`/bower_components/kurento-utils/js/kurento-utils.js?v=${randomNumber}`)
    script(src=`/client_side/client_socket.js?v=${randomNumber}`)

```

LAMPIRAN 4

Source code client-side.js

```

var video;
var webRtcPeer;
var socket = io();
var autoView = true;
var room;
var pesan = '';
var name = document.getElementById('name').value;
var room_class = document.getElementById('idCourse').value;
{
    video = document.getElementById('video-webcam');

    if(document.getElementById('btnRTC') != null){

```

```

        document.getElementById('btnRTC').onclick = () => {
            stop();
            presenter();
        };
    }

    if(document.getElementById('btnScreen') != null){
        document.getElementById('btnScreen').onclick = () => {
            stop();
            screenShare();
        };
    }

    if(document.getElementById('terminate') != null){
        document.getElementById('terminate').onclick = () => {
            stop();
        };
    }

    if(document.getElementById('btnConnect') != null){
        document.getElementById('btnConnect').onclick = () => {
            stop();
            viewer();
        };
    }

    document.getElementById('btnSubmit').onclick = () => {
        // if(document.getElementById('txt').value == ''){
        //     return false;
        // }else{
        //     var message = {
        //         username : name,
        //         message : document.getElementById('txt').value
        //     };
    }

```

```

        //      socket.emit('chatMessage', message);
        //      document.getElementById('txt').value = '';
        // }
        // alert('Hallooo !!');
        sendFile();
};

document.getElementById('txt').onkeyup = (e) => {
    // alert('anda mengetik !');
    if(e.keyCode === 13){
        if(document.getElementById('txt').value == ''){
            return false;
        }else{
            var message = {
                username : name,
                message : document.getElementById('txt').value
            };
            socket.emit('chatMessage', message);
            document.getElementById('txt').value = '';
        }
    }else{
    }
};

document.getElementById('txt').addEventListener('keypress', (e) => {
    socket.emit('typing', 'Someone Typing...');
});

document.getElementById('txt').addEventListener('blur', (e) => {
    socket.emit('typing', 'Type Here...');
});

```

```

//Listening Connect
socket.on('connect', () => {
    console.log('Connected to socket');
    var data = {
        room:currentRoom(),
    };
    socket.emit('subscribeToStream', data);
});

//Listening Disconnect
socket.on('disconnect', () => {
    console.log('Disconnected from socket');
    dispose();
});

//Listening Chat
socket.on('chat message', (data) => {
    if(pesan != data){
        console.log(data);
        $('#messages').append($('

###### ').text(data)); pesan = data; } }); //Listening Typing socket.on('typing', (msg) => { $('#txt').attr('placeholder', msg); }); //Listening Presenter Response socket.on('presenterResponse', (data) => { presenterResponse(data); });


```

```

//Listening Viewer Response
socket.on('viewerResponse', (data) => {
    viewerResponse(data);
});

//Listening StopCommunication
socket.on('stopCommunication', (data) => {
    console.log('stopCommunication');
    dispose();
});

//Listening Ice Candidate
socket.on('iceCandidate', (data) => {
    webRtcPeer.addIceCandidate(data.candidate)
});

//listening stream started
socket.on('streamStarted', (data) => {
    if (autoView) {
        viewer();
    }
});

//Listening Chat Masuk
socket.on('newChatMessage', (data) => {
    if(pesan != data){
        // console.log(data);
        $('#messages').append($('

###### ').text(data)); pesan = data; } }); //Listening File Masuk socket.on('newFileMessage', (data) => {


```

```

        var location =
`https://spellide.unhas.ac.id/course/download/${data.namaFile}`;

        var link = `Upload Some Files</a>`;

        var sender = data.sender;

        var datax = '<h6>' + sender + ' : ' + link + '</h6>';

        if\(pesan != datax\){
            // console.log\(data\);

            document.getElementById\('messages'\).innerHTML += '<h6>' + sender +
' : ' + link + '</h6>';

            pesan = datax;
        }
    }\);

    //WEBCAM SHARING

    presenterResponse = \(message\) => {
        // console.log\('Presenter Response !'\);

        // console.log\(`sdp answer : \${message.sdpAnswer}`\);

        // console.log\(`Response : \${message.response}`\);

        if\(message.response != 'accepted'\){
            var errorMsg = message.message ? message.message : 'Unkown error';

            console.warn\('Call not accepted for the following reason : ' +
errorMsg\);

            dispose\(\);
        }else{
            console.log\(`From Presenter " \${webRtcPeer}`\);

            webRtcPeer.processAnswer\(message.sdpAnswer\);
        }
    }

};

    viewerResponse = \(message\) => {
        if\(message.response != 'accepted'\){
            var errorMsg = message.message ? message.message : 'Unkown error';

            console.warn\('Call not accepted for the following reason : ' +
errorMsg\);

            dispose\(\);
        }
    }
};

```

```

        }else{
            console.log(`From Viewer ${message.sdpAnswer}`);
            webRtcPeer.processAnswer(message.sdpAnswer);
        }
    };
    presenter = () => {
        if(!webRtcPeer){
            var options = {
                localVideo:video,
                onIceCandidate
            };
            webRtcPeer = kurentoUtils.WebRtcPeer.WebRtcPeerSendonly(options,
(error) => {
                if(error){
                    console.log(error);
                }
                console.log(this);
                // this.generateOffer(onOfferPresenter);
                webRtcPeer.generateOffer(onOfferPresenter);
            }, 'webcam');
        }
    };

    onOfferPresenter = (error, offerSdp) => {
        if(error){
            return onError(error);
        }
        var message = {
            sdpOffer:offerSdp,
            room: currentRoom(),
        };
        console.log(`from presenter : ${offerSdp}`)
        socket.emit('presenter', message);
    };

```



```

viewer = () => {
    autoView = true;
    if(!webRtcPeer){
        var options = {
            remoteVideo: video,
            onIceCandidate: onIceCandidate,
            sendSource: 'webcam',
        };

        webRtcPeer = kurentoUtils.WebRtcPeer.WebRtcPeerRecvonly(options,
(error) => {
            if(error){
                return onError(error);
            }
            webRtcPeer.generateOffer(onOfferViewer);
        });
        console.log(`webRtcPeer : ${webRtcPeer}`);
        // webRtcPeer.generateOffer(onOfferPresenter);
    }
};

onOfferViewer = (error, offerSdp) => {
    if(error){
        return onError(error);
    }

    var message = {
        sdpOffer: offerSdp,
        room: currentRoom()
    };

    console.log(`From Viewer ${offerSdp}`);

    socket.emit('viewer', message);
}

```

```

};

onIceCandidate = (candidate) => {
    var message = {
        candidate:candidate
    };
    socket.emit('onIceCandidate', {candidate:candidate});
};

stop = () => {
    autoView = false;
    if(webRtcPeer){
        socket.emit('stop');
        dispose();
    }
};

dispose = () => {
    if(webRtcPeer){
        webRtcPeer.dispose();
        webRtcPeer = null;
    }
};

//END WEBCAM SHARING
//SCREEN SHARE
screenShare = () => {
    if(!webRtcPeer){
        var options = {
            localVideo:video,
            onIceCandidate: onIceCandidate,
            sendSource:'screen',
        };

        webRtcPeer = kurentoUtils.WebRtcPeer.WebRtcPeerSendonly(options,
(error) => {

```

```

        if(error){
            console.log(error);
        }
        console.log(this);
        // this.generateOffer(onOfferPresenter);
        webRtcPeer.generateOffer(onOfferPresenter);
    }, 'screen');
    }
};
//END SCREEN SHARE
//Get Room Course
currentRoom = () => {
    return room_class;
};
//Send File
sendFile = () => {
    var idCourse = document.getElementById('idCourse').value;
    var idUser = document.getElementById('idUser').value;
    var myFile = document.getElementById('myFile').files;

    var formData = new FormData();

    formData.append('course', idCourse);
    formData.append('author', idUser);
    formData.append('myFile', myFile[0]);

    var namaFile = idCourse + '-' + myFile[0].name;
    namaFile = namaFile.replace(/ /g, "_");
    // console.log(`nama : ${namaFile}`);

    var contentType = {
        headers:{
            "content-type" : "multipart/form-data"
        }
    }
};

```

```

    });
    axios.post('/course/upload/'+idCourse, formData, contentType)
    .then((response) => {
        if(response.data.stat == 'true'){
            console.log('File Success Uploaded');
            console.log(`name : ${namaFile}`);
            myFile.value = '';

            //Send Data into MongoDB
            var newFile = {
                title: idCourse+'-'+namaFile,
                course:idCourse,
                contributors:idUser
            };
            //socket io for broadcasting all files
            message = {
                namaFile : namaFile,
                sender : document.getElementById('name').value,
            };
            socket.emit('fileMessage', message);
        }
    })
    .catch((error) => {
        console.log('Error : ' + error);
    })
    };
};

```

LAMPIRAN 5

Source Code course.js (Controller)

```

const User = require('../models/User');
const Class = require('../models/Class');
const Exercise = require('../models/Exercise');

```

```

const Work = require('../models/Work');
const Course = require('../models/Course');
const Module = require('../models/Module');
const File = require('../models/File');
const multer = require('multer');
exports.indexByStudent = async (req, res) => {
  const [workData, classData] = await Promise.all([
    Work.find({student: req.user._id}).exec(),
    Class.find({students: {$all: [req.user._id]}})
      .populate('courseId').exec()
  ]);
  const coursesData = classData.map((e) => { return e.courseId; });
  // console.log(coursesData);
  var courseList = await Course.find({}, {}, {sort: {order : 1}}).exec();
  return res.render('course/studentlist', {courses : coursesData, title : 'My Courses'
});
}
exports.getByStudent = async (req, res) => {
}
exports.livecourses = async (req, res) => {
  var thisCourse = await Course.findById(req.params.courseId).exec();
  var randomNumber = Math.floor((Math.random() * 999999999999) + 1);
  console.log(`randomNumber : ${randomNumber}`);
  // randomNumber = '/client_side/client_socket?v${}';
  return res.render('course/livecourses', {
    title: thisCourse.title,
    idCourse: thisCourse.id,
    author: thisCourse.author,
    randomNumber: randomNumber
  });
};
exports.tesUpload = async (req, res) => {
  return res.render('course/tesUpload');
};

```

```

exports.doUpload = async (req, res) => {
  idMataKuliah = req.params.courseId;
  var storage = multer.diskStorage({
    destination: (req, file, cb) => {
      cb(null, __dirname + '/../uploads')
    },
    filename: (req, file, cb) => {
      var namaFile = file.originalname.replace(/ /g, "_");
      cb(null, idMataKuliah+'-'+namaFile)
    }
  })
  var upload = multer({ storage: storage }).single('myFile')
  upload(req, res, (err) => {
    if (err instanceof multer.MulterError) {
      // A Multer error occurred when uploading.
      console.log(`error ${err}`)
    } else if (err) {
      console.log(`Error : ${err}`)
      // An unknown error occurred when uploading.
    }
  });
  // Everything went fine.
  res.send({'stat':'true'});
  console.log('Sucess !!');
};

exports.downloadFile = async (req, res) => {
  file = __dirname + '/../uploads/' + req.params.namaFile;
  res.download(file);
};

exports.previewModuleById = async (req, res) => {
  var thisCourse = await Course.findById(req.params.courseId).exec();
  var Modules = await Module.find({ courseId : req.params.courseId }).exec();
  console.log
  console.log(Modules);
};

```

```

    console.log(thisCourse);

    return res.render('course/studentmodulelist', {title: thisCourse.title, modules :
Modules});
}
exports.previewModuleById2 = async (req, res) => {
    // var thisCourse = await Course.findById(req.params.courseId).exec();
    // var Modules = await Module.find({ courseId : req.params.courseId }).exec();
    // console.log(Modules);
    // console.log(thisCourse);
    // return res.render('course/studentmodulelist', {title: thisCourse.title, modules :
Modules});
    try {
        var thisModule = await Module.findById(req.params.moduleId).exec();
        return res.render('course/preview', {title: thisModule.titleModule, ex: thisModule});
    } catch(err) {
        console.log(thisCourse);
        res.status(500);
    }
}
exports.index = async (req, res) => {
    if (req.user.role == 'student')
        return res.redirect('/studentcourses');
    var courseList = await Course.find({}, {}, {sort: {order : 1}}).exec();
    // console.log(exerciseList);
    // function(err, exerciseList) {
        return res.render('course/list', {courses : courseList, title: 'Course List' });
    // });
    // MyModel.find({ name: /john/i }, 'name friends', function (err, docs) { })
};
exports.create = (req, res, next) => {
    if (req.user.role !== 'teacher')
        return res.status(401).json({code:401, error: 'Wrong privilege'});

    var newCourse = new Course(req.body);
    newCourse.save(function (err) {

```

```

    if (err)
        return res.status(500).json({code:500, error: err});
    return res.json({code:200});
});
}
exports.saveFileToDb = (req, res, next) => {
    var newFile = new File(req.body);
    newFile.save((err) => {
        if(err){
            return res.status(500).json({code:500, error:err});
        }
        return res.json({code:200});
    });
};
exports.createModule = (req, res, next) => {
    if (req.user.role !== 'teacher')
        return res.status(401).json({code:401, error: 'Wrong privilege'});
    // return res.json({code:200});
    // return res.json( req.body );
    var newModule = new Module(req.body);
    newModule.save(function (err) {
        if (err)
            return res.status(500).json({code:500, error: err});
        return res.json({code:200});
    });
}
exports.createExercise = (req, res, next) => {
    if (req.user.role !== 'teacher')
        return res.status(401).json({code:401, error: 'Wrong privilege'});
    // return res.json({code:200});
    // return res.json( req.body );
    var newExercise = new Exercise(req.body);
    newExercise.save(function (err) {
        if (err)

```



```

        return res.status(500).json({code:500, error: err});
    }
    return res.json({code:200});
});
}
exports.getById = async (req, res) => {
    // console.log(req);
    if (req.user.role == 'student')
        return res.redirect('/studentcourses');

    const [classData, exerciseData, userData, moduleData] = await Promise.all([
        Class.findById(req.params.classId).exec(),
        Exercise.find({}).sort('title').exec(),
        User.find({}).sort('email').exec(),
        Module.find({ courseId : req.params.courseId}).exec()
    ]);
    // console.log(classData);
    //
    console.log(req.params.courseId);
    var teachers = userData.filter((s) => s.role === 'teacher')
        .map((s) => {
            s=Object.assign({},s._doc);
            if(classData!=null)
                s.inClass = classData.staff.indexOf(s._id)>=0;
            a = JSON.stringify(s._id);
            b = JSON.stringify(req.user._id);
            if(a==b) s.disabled = true;
            return s;
        });
    var modules = moduleData;
    console.log(modules);
    console.log('kucing gila');
    try {
        var thisCourse = await Course.findById(req.params.courseId).exec();
    }

```

```

    var author = await User.findById(thisCourse.author).exec();

    return res.render('course/edit', {title: thisCourse.title, ex: thisCourse, teachers,
modules, idCourse: req.params.courseId, author: author.profile.name});

  } catch(err) {

    console.log(thisCourse);

    res.status(500);

  }
};

exports.getModuleById = async (req, res) => {

  // console.log(req.params.moduleId);

  try {

    var thisModule = await Module.findById(req.params.moduleId).exec();

    var thisCourse = await Course.findById(req.params.courseId).exec();

    console.log(thisModule);

    return res.render('course/editmodule', {title: thisModule.titleModule, ex:
thisModule, courseTitle : thisCourse.title, courseId: req.params.courseId, moduleId :
req.params.moduleId});

  } catch (err) {

    console.log(err);

  }

}

exports.getExerciseByCourse = async (req, res) => {

  try {

    // console.log(req.params);

    var thisCourse = await Course.findById(req.params.courseId).exec();

    var thisExercises = await Exercise.find({courseId :
req.params.courseId}).populate('parentId').exec();

    var Modules = await Module.find({courseId : req.params.courseId }).exec();

    // console.log(Modules);

    console.log(thisExercises);

    return res.render('course/exerciseslist', { title: thisCourse.title, ex : thisCourse,
exercises : thisExercises });

  } catch (err) {

    console.log(err);

  }

}
}

```

```

exports.getExerciseById = async (req, res) => {
  try {
    var thisCourse = await Course.findById(req.params.courseId).exec();
    var thisExercise = await Exercise.findById(req.params.exerciseId).exec();
    var thisModules = await Module.find({$or : [{ courseId : req.params.courseId },
{titleModule : 'None'}]}).exec();
    // var thisCourse = await Course.find(req.params.courseId).exec();
    console.log(req.params);
    return res.render('course/editexercise', { title: '', ex: thisExercise, thisCourse,
thisModules, courseId : req.params.courseId, exerciseId : req.params.exerciseId
    });
  } catch (err) {
    console.log(err);
  }
}

exports.previewById = async (req, res) => {
  console.log(req);
  // if (req.user.role == 'student')
  //   return res.redirect('/studentcourses');
  try {
    var thisCourse = await Course.findById(req.params.courseId).exec();
    return res.render('course/preview', {title: thisCourse.title, ex: thisCourse});
  } catch(err) {
    console.log(thisCourse);
    res.status(500);
  }
};

exports.deleteById = (req, res) => {
  if (req.user.role !== 'teacher')
    return res.status(401).json({code:401, error: 'Wrong privilege'});
  Course.findByIdAndRemove(req.params.courseId, function(err,doc) {
    if (err);
    return res.redirect('/course');
  });
}

```

```

exports.deleteModuleById = (req, res) => {
  if (req.user.role !== 'teacher')
    return res.status(401).json({code:401, error: 'Wrong privilege'});

  Module.findByIdAndRemove(req.params.moduleId, function(err,doc) {
    if (err);
    return res.redirect('/course/' + req.params.courseId);
  });
}

exports.updateExerciseById = (req, res, next) => {
  if (req.user.role !== 'teacher')
    return res.status(401).json({code:401, error: 'Wrong privilege'});
  Exercise.findByIdAndUpdate(req.params.exerciseId, req.body, function(err, doc) {
    console.log(req.body);
    if (err)
      return res.status(500).json({code:500, error: err});
    return res.json({code:req.body});
  });
}

exports.updateModuleById = (req, res, next) => {
  if (req.user.role !== 'teacher')
    return res.status(401).json({code:401, error: 'Wrong privilege'});

  Module.findByIdAndUpdate(req.params.moduleId, req.body, function(err, doc) {
    console.log(req.body);
    if (err)
      return res.status(500).json({code:500, error: err});
    return res.json({code:req.params.moduleId});
  });
}

exports.updateById = (req, res, next) => {
  if (req.user.role !== 'teacher')
    return res.status(401).json({code:401, error: 'Wrong privilege'});

```

```
Course.findByIdAndUpdate(req.params.courseId, req.body, function(err, doc) {
    console.log(req.body);
    if (err)
        return res.status(500).json({code:500, error: err});
    return res.json({code:req.params.courseId});
});
}
```



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Poros Malino Km.6Bontomarannu(92172) Gowa, Sulawesi Selatan 92172, Sulawesi Selatan
Telp. (0411) 586015, 586262 Fax (0411) 586015
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
No. 18940/UN4.7.1/DA:08.04/2018

Dari : Dekan Fakultas Teknik Universitas Hasanuddin

Kepada : 1. Dr.Eng. Muhammad Niswar, ST., M.IT Pemb. I
2. Dr. Amil Ahmad Ilham, ST., M.IT Pemb. II

Isi : 1. Berdasarkan Surat Ketua Departemen Teknik Informatika Fakultas Teknik Nomor. 303/UN4.7.7./DA.08.04/2018 tanggal 4 Desember 2018 tentang usul DOSEN PEMBIMBING MAHASISWA, maka dengan ini kami menugaskan Saudara untuk membimbing penulisan Skripsi/Tugas Akhir mahasiswa Teknik Informatika Fakultas Teknik Universitas Hasanuddin di bawah ini :

Nama :
Hermawan Safrin

No. Stambuk :
D421 14 314

Judul Skripsi/Tugas Akhir :

“ Implementasi Teknologi Web Real-Time Communication (WebRTC) dalam Perancangan Kelas Virtual sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento “

2. Surat penugasan pembimbing ini mulai berlaku sejak tanggal ditetapkannya dan berakhir sampai selesainya penulisan Skripsi/Tugas Akhir mahasiswa tersebut.
3. Agar penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.

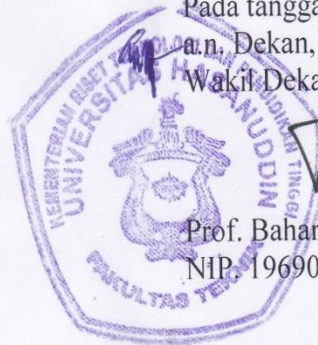
Ditetapkan di Gowa
Pada tanggal 4 Desember 2018

a.n. Dekan,
Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 196903081995121001

Tembusan :

1. Dekan FT-UH,
2. Ketua Departemen Teknik Informatika FT-UH,
3. Mahasiswa yang bersangkutan





KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id>, Email : teknik@unhas.ac.id

SURAT PENUGASAN
19256/UN4.7.1/DA.08.04/2018

Dari : Dekan Fakultas Teknik Universitas Hasanuddin.

Kepada : Mereka yang tercantum namanya di bawah ini.

Isi : 1. Bahwa berdasarkan peraturan Akademik Universitas Hasanuddin Tahun 2003 Pasal 36 butir 3 point a, b (SK. Rektor Unhas Nomor : 1067 /J04/PP.08/2008), dengan ini menugaskan Saudara sebagai PANITIA SEMINAR PROPOSAL Strata Satu (S1) Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin dengan susunan sebagai berikut :

Pembimbing I / Ketua : 1. Dr.Eng. Muhammad Niswar, ST., M.IT
Pembimbing II / Sekretaris : 2. Dr. Amil Ahmad Ilham, ST., M.IT
Anggota : 3. Dr.Eng. Zulkifli Tahir, ST., M.Sc
4. Ir. Christoforus Yohannes, M.T

untuk menguji bagi mahasiswa tersebut di bawah ini :

Nama/NIM : Hermawan Safrin D421 14 314

Departemen : Teknik Informatika

Judul Thesis/Skripsi : **" Implementasi Teknologi Web Real-Time Communication (WebRTC) dalam Perancangan Kelas Virtual sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento "**

2. Waktu seminar ditetapkan oleh Panitia Seminar Proposal Strata Satu (S1).
3. Agar Surat penugasan ini dilaksanakan sebaik-baiknya dengan penuh rasa tanggung jawab.
4. Surat penugasan ini berlaku sejak tanggal ditetapkan sampai dengan berakhirnya Seminar tersebut dengan ketentuan bahwa segala sesuatunya akan ditinjau dan diperbaiki sebagaimana mestinya apabila dikemudian hari ternyata terdapat kekeliruan dalam keputusan ini.

Ditetapkan di Gowa,
Pada tanggal 10 Desember 2018

a.n. Dekan

Wakil Dekan Bidang Akademik

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
NIP. 19690308 199512 1 001

Tembusan :

1. Dekan Fak. Teknik Unhas
2. Ketua Departemen Teknik Informatika FT-UH
3. Mahasiswa yang bersangkutan





KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS HASANUDDIN

FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI UJIAN SEMINAR PROPOSAL

Nomor : 345.../UN4.7.7.TI/PP.15/2018

Pada hari ini Rabu, tanggal 12 Desember 2018 Pukul 09.00 WITA-Selesai bertempat di Ruang Lab.CCIE Departemen Teknik Informatika, telah dilaksanakan Ujian Seminar Proposal bagi Saudara :

Nama : Hermawan Safrin

No. Stambuk : D421 14316

Fakultas/Departemen : Teknik/Teknik Informatika

Judul Skripsi : “ Implementasi Teknologi Web Real-Time Communication (WebRTC) dalam Perancangan Kelas Virtual sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento “

Setelah pembawa ujian seminar proposal menguraikan tugas akhirnya dan menjawab pertanyaan dengan dinyatakan lulus/tidak lulus, Baik/Cukup/Sedang.

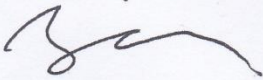
Maka berdasarkan hasil penilaian dinyatakan lulus / tidak lulus

Dengan nilai :

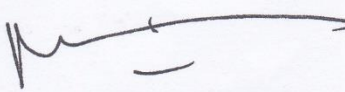
A- B+ B B- C+ C D E

Mengetahui:

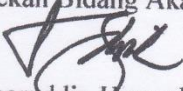
A.n.Ketua Departemen Tek.Informatika,
Sekretaris Departemen


Dr.Indrabayu.,ST.M.T.M.Bus.Sys
Nip. 19750716 200212 1 00

Dosen Penguji,


Dr.Eng. Muhammad Niswar,ST.,M.I.T
Nip.19730922 199903 1 001

Diketahui oleh,
Dekan,
Wakil Dekan Bidang Akademik,


Prof. Baharuddin Hamzah,ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SEMINAR PROPOSAL

Pada hari ini **Rabu**, tanggal **12 Desember 2018** Pukul **09.00 WITA-Selesai** bertempat di **Ruang Lab.CCIE Departemen Teknik Informatika**, telah dilaksanakan Ujian Seminar Proposal bagi Saudara :

Nama : .Hermawan Safrin
No. Stambuk : D421 14314
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : “ **Implementasi Teknologi Web Real-Time Communication (WebRTC) dalam Perancangan Kelas Virtual sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento** “

Yang dihadiri oleh panitia Ujian Skripsi sebagai berikut :

No.	N a m a	Jabatan	Tanda tangan
1.	Dr.Eng. Muhammad Niswar,ST.,M.I.T	Pemb I/Ketua	1...
2.	Dr.Amil Ahmad Ilham,ST.M.I.T	Pemb II/Sekretaris	2...
3.	Dr. Eng. Zukifli Tahir,ST.,M.Sc	Anggota	3...
4.	Ir.Christoforus Yohannes,M.T	Anggota	4...

Hasil keputusan panitia penilai Ujian Seminar Tugas Akhir : **Lulus / Tidak lulus** dengan nilai angka dan huruf

Gowa, 12 Desember, 2018

Ketua/Sekretaris Panitia Ujian

Dr.Eng. Muhammad Niswar,ST.,M.I.T



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SEMINAR PROPOSAL

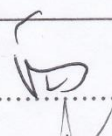
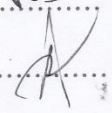

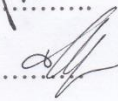
Nama/Stambuk : 1.Hermawan Safrin D421 14314

Judul Skripsi/T.A : “ Implementasi Teknologi Web Real-Time Communication (WebRTC) dalam Perancangan Kelas Virtual sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento “

Hari/Tanggal : Rabu, 12 Desember, 2018

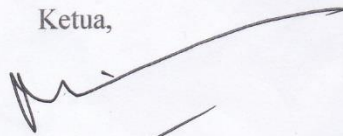
Jam : 09.00 Wita – Selesai

Tempat : Ruang Lab.CCIE Departemen Teknik Informatika

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr.Eng. Muhammad Niswar,ST.,M.I.T	1..... 
	Pembimbing II	2. Dr.Amil Ahmad Ilham,ST.M.I.T	2..... 
II.	Anggota Penguji	3. Dr. Eng. Zukifli Tahir,ST.,M.Sc	3..... 
		4. Ir.Christoforus Yohannes,M.T	4..... 

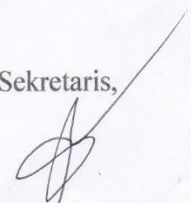
PANITIA UJIAN

Ketua,



Dr.Eng. Muhammad Niswar,ST.,M.I.T

Sekretaris,



Dr.Amil Ahmad Ilham,ST.M.I.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

SURAT KETERANGAN NILAI UJIAN SKRIPSI

Nomor : / UN4.7.7.TI/PK.03.06/2020

Pada hari ini **Jum'at**, tanggal **21 Agustus 2020** Pukul **16.00 WITA** - **Selesai** bertempat di **Ruang Lab. UBICON Departemen Teknik Informatika**, telah dilaksanakan Ujian Skripsi bagi Saudara :

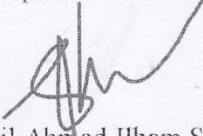
Nama : Hermawan Safrin
No. Stambuk : D42114 314
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : **"Implementasi Teknologi Web Real-Time Communication(WebRTC) dalam Perancangan Kelas Virtual Sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento"**

Setelah pembawa ujian Skripsi menguraikan tugas akhirnya dan menjawab pertanyaan dari Tim Penguji dinyatakan Lulus / Tidak Lulus dengan nilai :

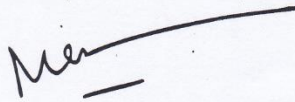
A 87 A- B+ B B- C+ C D E

Mengetahui:

Ketua Departemen Tek. Informatika,


Dr. Amil Ahmad Ilham, ST, M.I.T
Nip. 19731010 199802 1 001

Dosen Penguji,


Dr. Eng. Muhammad Niswar, ST., M.I.T
Nip. 19730922 199903 1 001

Diketahui oleh,
a.n. Dekan,
Wakil Dekan Bidang Akademik, Riset dan Inovasi

Prof. Baharuddin Hamzah, ST., M.Arch., Ph.D
Nip. 19690308 199512 1 001



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK
DEPARTEMEN TEKNIK INFORMATIKA
Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

BERITA ACARA UJIAN SKRIPSI

Pada hari ini **Jum'at**, tanggal **21 Agustus 2020** Pukul **16.00 WITA** - Selesai bertempat di **Ruang Lab. UBICON Departemen Teknik Informatika**, telah dilaksanakan Ujian Skripsi bagi Saudara :

Nama : Hermawan Safrin
No. Stambuk : D42114 314
Fakultas/Departemen : Teknik/Teknik Informatika
Judul Skripsi : **"Implementasi Teknologi Web Real-Time Communication(WebRTC) dalam Perancangan Kelas Virtual Sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento"**

Yang dihadiri oleh Tim Penguji Ujian Skripsi sebagai berikut :

No.	Nama	Jabatan	Tanda tangan
1.	Dr.Eng. Muhammad Niswar,ST.,M.I.T	Pemb I/Ketua	1.
2.	Dr. Amil Ahmad Ilham,ST.,M.I.T	Pemb II/Sekretaris	2.
3.	Ir. Christoforus Yohannes,M.T	Anggota	3.
4.	Dr.Eng.Zulkifli Tahir,ST.,M.Sc	Anggota	4.

Hasil keputusan Tim Penguji Ujian Skripsi/Tugas Akhir : **Lulus / Tidak lulus** dengan nilai angka **0.7** dan huruf **A**

Gowa, 21 Agustus 2020

Ketua/Sekretaris Panitia Ujian,

Dr.Eng. Muhammad Niswar,ST.,M.I.T



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS HASANUDDIN
FAKULTAS TEKNIK

DEPARTEMEN TEKNIK INFORMATIKA

Kampus Fakultas Teknik Unhas, Jl. Poros Malino, Gowa
<http://eng.unhas.ac.id/informatika>, Email : informatika@unhas.ac.id

DAFTAR HADIR UJIAN SKRIPSI MAHASISWA
FAKULTAS TEKNIK UNHAS

Nama/Stambuk : I. Hermawan Safrin D42114 314

Judul Skripsi/T.A : "Implementasi Teknologi Web Real-Time Communication(WebRTC) dalam Perancangan Kelas Virtual Sebagai Sarana Pembelajaran Interaktif dengan Media Server Kurento"

Hari/Tanggal : Jum'at, 21 Agustus 2020

Jam : 16.00 Wita – Selesai

Tempat : Ruang Lab. UBICON Departemen Teknik Informatika Gowa

No.	Jabatan	Nama Dosen	Tanda Tangan
L.	Pembimbing I	1. Dr.Eng. Muhammad Niswar,ST.,M.I.T	1.
	Pembimbing II	2. Dr. Amil Ahmad Ilham,ST.,M.I.T	2.
II.	Anggota Penguji	3. Ir. Christoforus Yohannes,M.T	3.
		4. Dr.Eng.Zulkifli Tahir,ST.,M.Sc	4.

PANITIA UJIAN

Ketua,


Dr.Eng. Muhammad Niswar,ST.,M.I.T

Sekretaris,


Dr. Amil Ahmad Ilham,ST.,M.I.T

KARTU BIMBINGAN SKRIPSI
Departemen S1 Teknik Informatika Universitas Hasanuddin

Stambuk	Nama Mahasiswa
D421 14 314	Hermawan Safrin

Pembimbing	Nama Pembimbing	Paraf & Tgl. Persetujuan Ujian Akhir
I	Dr.Eng Muhammad Niswar, ST., M.IT.	
II	Dr. Amil Ahmad Ilham, ST., M.IT.	
No. SK Pembimbing : 18940 / UNH.7.1/DA.08.04/2018		

Judul Skripsi :	Implementasi Teknologi Web Real-Time Communication (WebRTC) Dalam Perancangan kelas virtual sebagai sarana pembelajaran Interaktif Dengan Media Server Kurento
-----------------	--

No.	Tanggal Bimbingan	Uraian Kegiatan Bimbingan	Paraf
1	10/12/19	konsultasi mengenai rancangan komunikasi webRTC (kelas virtual)	
2	27/12/19	konsultasi rancangan layout kelas virtual	
3	06/01/20	konsultasi Arsitektur (gambaran umum)	
4	12/02/20	konsultasi dan Perbaiki Padi sistem yang telah dibangun	
5	27/02/20	Demonstrasi sistem (local version)	
6	05/03/20	konsultasi untuk membuat sistem agar bisa diakses Online	
7	21/04/20	konsultasi Penulisan Skripsi	
8	13/06/20	konsultasi Skenario uji coba sistem untuk ambil data	
9	23/07/20	konsultasi mengenai hasil dari data yang diambil	
10	10/08/20	konsultasi keseluruhan sistem dan skripsi	
11			
12			
13			

LEMBAR PERBAIKAN SKRIPSI

“Implementasi Teknologi *Web Real-Time Coomunication (WebRTC)* Dalam Perancangan Kelas Virtual Sebagai Sarana Pembelajaran Interaktif Dengan Media Server Kurento”

Oleh :





HERMAWAN SAFRIN

D421 14 314


Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 21 Agustus 2020.

Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi

Persetujuan perbaikan oleh Tim penguji :

	Nama	Tanda Tangan
Ketua	Dr.Eng Muhammad Niswar, ST., M.IT.	
Sekretaris	Dr. Amil Ahmad Ilham, ST., M.IT.	
Anggota	Dr.Eng. Zulkilfi Tahir, ST., M.Sc.	
	Ir. Christoforus Yohannes, M.T.	

Persetujuan perbaikan oleh Pembimbing :

Pembimbing	Nama	Tanda Tangan
I	Dr.Eng Muhammad Niswar, ST., M.IT.	
II	Dr. Amil Ahmad Ilham, ST., M.IT.	