

DAFTAR PUSTAKA

- Batubara, Febrin Aulia. 2015. "PERANCANGAN WEBSITE PADA PT . RATU ENIM PALEMBANG." 15–27.
- Butkiewicz, Michael. 2011. "Understanding *Website* Complexity : Measurements, Metrics, and Implications Categories and Subject Descriptors."
- Chrome DevTools. (n.d.). Chrome Developers. Retrieved October 23, 2022, from <https://developer.chrome.com/docs/DevTools/>
- Chervyakov, N., Lyakhov, P., & Nagornov, N. (2020). Analysis of the Quantization Noise in Discrete Wavelet Transform Filters for 3D Medical Imaging. *Applied Sciences*, 10(4), 1223.
<https://doi.org/10.3390/app10041223>.
- Dicoding Indonesia, 2021. Apa itu Node Package Manager: Tutorial NPM dan JavaScript. Diakses dari <https://www.dicoding.com/blog/node-package-manager/>
- Fajrin, Rachmat. (2017). "PENGEMBANGAN SISTEM INFORMASI GEOGRAFIS BERBASIS NODE.JS UNTUK PEMETAAN MESIN DAN TRACKING ENGINEER DENGAN PEMANFAATAN GEOLOCATION PADA PT IBM INDONESIA". *Jurnal Informatika* Vol. 11, No. 2
- Fathurahman. 2014. "Membuat Website Mudah dan Praktis dengan Weebly" . Jakarta: PT Elex Media Komputindo.
- Haidar Dzacko. 2007. "1 .BASIS DATA (DATABASE)."
- Idcloudhost. 2022. Tips Meningkatkan Optimalisasi Gambar/Image EO Bagi Ranking Website Tahun 2022. Diakses dari <https://idcloudhost.com/tips-meningkatkan-optimalisasi-gambar-image-seo-bagi-ranking-website-tahun-2022/>

- Imavijit, 2020. Python | Peak Signal-to-Noise Ratio (PSNR). Diakses dari <https://www.geeksforgeeks.org/python-peak-signal-to-noise-ratio-psnr/>
- Khuat, Tung. 2018. “Developing a Frontend Application Using ReactJS and Redux.”
- Kurnialensya, Taufik. 2022. Kelebihan Dan Kekurangan Sebuah Database MySQL. Diakses dari <http://teknik-informatika-s1.stekom.ac.id/informasi/baca/Kelebihan-dan-Kekurangan-Sebuah-Database-MySQL/240305a633409c1eae6af77a27c91bcddbcec23d>
- Maratkar, Pratik Sharad, Pratibha Adkar. (2021). *ReactJS – An Emerging Frontend Javascript Library*. IRE Journals, vol 4, issue 12
- Ni. (2020). Peak Signal-to-Noise Ratio as an Image Quality Metric. Diakses dari <https://www.ni.com/en-id/innovations/white-papers/11/peak-signal-tonoise-ratio-as-an-image-quality-metric.html>
- Permana, Endang Cahya. 2016. “Penulisan Fungsi Pada *Javascript*.” 1-27.
- Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi
- Standisyah, R. E., & Restu N.S, I.S. (2017). IMPLEMENTASI PHPMYADMIN PADA RANCANGAN SISTEM PENGADMINISTRASIAN. *Jurnal UJMC*, 38-44.
- Suprpto, Agung, Dimas Sasongko. (2021). Studi Empiris Evaluasi Performa Website IAIN Salatiga Menggunakan Automated Software Testing. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 5, no. 1
- Sommerville, Ian. 2011. *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga
- Tanjua, Muhammad Habil Likai. Apasih PNG dan JPEG itu?. Diakses dari <http://mgt.unida.gontor.ac.id/apasih-png-dan-jpeg-itu/>
- Google Developer. 2022. WebP: a new image format for the web. Diakses dari <https://developers.google.com/speed/webp>

LAMPIRAN

Lampiran 1 : Source Code

a. Index.js

```
const express = require("express")
const dotenv = require("dotenv")
const cors = require("cors")
const path = require("path")
const bodyParser = require("body-parser")
const WisataRoute = require("./routes/WisataRoute.js")
const cookieParser = require("cookie-parser")
const db = require('./models')

dotenv.config()
const app = express();

const allowedOrigins = ['http://localhost:3000', 'http://localhost:3001',
'http://127.0.0.1:5000', 'http://127.0.0.1:5000']
app.use(cors({ credentials:true, origin: allowedOrigins })))
app.use(cookieParser())
app.use(express.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(bodyParser.json());
app.use(express.static(path.join(__dirname, "/public")));
app.use(WisataRoute);

const run = async () => {
};
```

```
db.sequelize.sync({ force: false }).then(() => {  
  console.log("Drop and re-sync db.");  
  run();  
});  
  
app.listen(5000, () => console.log("Server running.."));
```

b. WisataWebp.js

```
const Sequelize = require("sequelize")  
const db = require("../config/Database.js")  
  
const { DataTypes } = Sequelize  
  
const WisataWebp = db.define('wisataswebp', {  
  name: DataTypes.STRING,  
  desc: DataTypes.TEXT,  
  image: DataTypes.STRING,  
}, {  
  freezeTableName: true  
})  
  
module.exports = WisataWebp;  
  
(async()=>{  
  await db.sync();  
})()
```

c. Multer.js

```

const multer = require('multer');
const path = require('path');
const fs = require('fs');
const storage = multer.diskStorage({
  destination: function(req, file, cb){
    var dir = "/public/images"
    if(!fs.existsSync(dir)){
      fs.mkdirSync(dir)
    }
    cb(null,dir)
  },
  filename: function(req, file, cb){
    cb(null,`${file.fieldname}-
${Date.now()}${path.extname(file.originalname)}`)
  }
});
const upload = multer({
  storage: storage,
  limits: { fileSize: "10000" },
  fileFilter: (req, file, cb) => {
    const fileTypes = /webp|jpeg|jpg|JPG|JPEG|png|PNG|gif/;
    const mimeType = fileTypes.test(file.mimetype);
    const extname = fileTypes.test(path.extname(file.originalname));

    if (mimeType && extname) {
      return cb(null, true);
    }
    cb("Give proper files formate to upload");
  },
}).single("image");

module.exports = upload;

```

d. Image.js

```

const upload = require('./multer');
const webp = require('webp-converter');
const fs = require('fs');
const wisataWebp = require('./WisataWebpController')
const WisataWebp = require("../models/WisataWebp.js");
module.exports = {
  saveImage(req, res){
    upload(req, res, err =>{
      if(err){
        return res.json({ status:'fail',msg:err })
      }else if(req.file===undefined){
        return res.json({ status:'fail',msg:"You must select a image." })
      }else{
        var dir = "/public/images/webp"
        if(!fs.existsSync(dir)){
          fs.mkdirSync(dir)
        }
        webp.cwebp(req.file.path,`${dir}/${req.file.filename.substring(0,
req.file.filename.lastIndexOf('.')}).webp`,"-q 90",async function(status){
          let finalImageURL =
            req.protocol + "://" + req.get("host") + "/images/webp/" +
`${req.file.filename.substring(0, req.file.filename.lastIndexOf('.')}).webp`;
          let info = {
            image: finalImageURL,
            name: req.body.name,
            desc: req.body.desc,
          };
          const response = await WisataWebp.create(info);
          res.status(201).send(response);
          return res.json({ status:'success',msg:"Image have been save
successfully." })
        });
      }
    })
  }
}

```

e. Psnr_imgz.py

```
import math
import cv2
import numpy as np

original = cv2.imread("ori.jpg")
multi = cv2.imread("jpg.jpg")

def psnr(img1, img2):
    mse = np.mean((img1 - img2) ** 2)
    if mse == 0:
        return ("inf")
    PIXEL_MAX = 255.0
    return 20 * math.log10(PIXEL_MAX / math.sqrt(mse))

psnr_value = psnr(original, multi)

print("PSNR Multi-frame =", psnr_value)
```