**ANALISIS DATA PEGAWAI UNTUK MEMPREDIKSI GAJI BERDASARKAN FAKTOR-FAKTOR SPESIFIK DENGAN PENDEKATAN MACHINE LEARNING**

**LAPORAN PROYEK II**

Ditujukan sebagai salah satu syarat

Untuk memperoleh nilai pada kegiatan Proyek II

Program Studi DIV Teknik Informatika

**Oleh**

Bachtiar Ramadhan (1204077)

Nur Tri Ramadhanti Adiningrum (1204061)



**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**BANDUNG**

**2021**

# LEMBAR PERSETUJUAN

**ANALISIS DATA PEGAWAI UNTUK MEMPREDIKSI GAJI BERDASARKAN FAKTOR-FAKTOR SPESIFIK DENGAN PENDEKATAN MACHINE LEARNING**

**LAPORAN PROYEK 2**

Program Studi DIV Teknik Informatika

**Oleh**

Bachtiar Ramadhan (1204077)  
Nur Tri Ramadhanti Adiningrum (1204061)

Telah disetujui dan diserahkan

Bandung, ……………………… 2022

**Pembimbing**

**Syafrial Fachri Pane, S.T., M.T.I., EBDP**

NIK : 117.88.233

# SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME

Saudara yang bertanda tangan dibawah ini :

|  |  |  |
| --- | --- | --- |
| Ketua Kelompok | : | Bachtiar Ramadhan (1.20.4.077) |
| Anggota | : | Nur Tri Ramadhanti Adiningrum (1.20.4.061) |
| Judul Laporan | : | ANALISIS DATA PEGAWAI UNTUK MEMPREDIKSI GAJI BERDASARKAN FAKTOR-FAKTOR SPESIFIK DENGAN PENDEKATAN MACHINE LEARNING |

Menyatakan bahwa kami tidak melakukan tindakan peniruan, penjiplakan ataupun penyalinan skripsi karya ilmiah yang telah ada. Apabila kami terbukti melakukan tindakan tersebut, maka kami bersedia menerima sanksi yang diberikan sesuai dengan ketentuan yang ditetapkan dan berlaku di Program Studi Diploma IV Teknik Informatika Politeknik Pos Indonesia.

|  |  |
| --- | --- |
| Yang membuat pernyataan, | Yang membuat pernyataan, |
| Ketua Kelompok | Anggota Kelompok |
|  |  |
| Bachtiar Ramadhan | Nur Tri Ramadhanti Adiningrum |
| 1.20.4.077 | NPM 1.20.4.061 |

# ABSTRAK

# *ABSTRACK*

# KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya lah kami dapat menyelesaikan kegiatan Proyek 2 berupa penyusunan laporan yang berjudul “ANALISIS DATA PEGAWAI UNTUK MEMPREDIKSI GAJI BERDASARKAN FAKTOR-FAKTOR SPESIFIK DENGAN PENDEKATAN REGRESI LINIER” dengan baik.

Dalam kesempatan kali ini tidak lupa kami mengucapkan terimakasih kepada semua pihak yang telah memberikan bantuan moral dan spiritual langsung maupun tidak langsung dalam kami menyelesaikan laporan ini, terutama kepada :

* 1. Orang tua dengan dukungan doanya.
  2. DR. Ir. Agus Purnomo., M.T. selaku Direktur Politeknik Pos Indonesia.
  3. M. Yusril Helmi Setyawan, S. Kom., M. Kom. selaku Ketua Program Studi DIV Teknik Informatika.
  4. Roni Andarsyah S.T., M. Kom. selaku Koordinator Proyek 2 Teknik Informatika.
  5. Syafrial Fachri Pane, S.T., M.T.I.,EBDP. selaku Pembimbing Proyek 2 yang telah memberikan pengarahan dan membimbing kami
  6. ….. selaku Dosen Penguji Proyek 2
  7. Serta semua pihak yang tidak dapat kami sebutkan satu persatu.

Kami menyadari bahwa laporan ini masih jauh dari kategori sempurna. Namun, kami sudah berusaha semampu kami. Oleh karena itu, kami sangat mengharapkan masukan baik saran maupun kritik yang membangun untuk kesempurnaan laporan yang akan datang.

Bandung, ………………. 2022

Penulis

# DAFTAR ISI

# DAFTAR GAMBAR

# DAFTAR TABEL

# BAB I

# PENDAHULUAN

## **Latar Belakang**

Perkembangan ilmu pengetahuan dan teknologi pada Revolusi Industri 4.0 semakin berkembang pesat. Revolusi Industri 4.0 sendiri mulai terjadi melalui rekayasa intelegensia dan *internet of thing* sebagai tulang punggung pergerakan dan konektivitas antara manusia dengan mesin[1]. Sehingga, terdapat penggabungan teknologi digital dan internet dengan industri konvensional, yang bertujuan untuk meningkatkan produktivitas, efisiensi dan layanan konsumen secara signifikan[2]. Era revolusi ini akan mendisrupsi berbagai kegiatan diberbagai bidang seperti pada bidang teknologi, ekonomi, sosial, dan politik[1]. Saat ini, kehidupan berada diawal revolusi yang secara mendasar mengubah cara hidup, bekerja, dan berhubungan satu sama lain [3].

Perubahan karakteristik pekerjaan adalah salah satu dampak tersendiri dari datangnya revolusi industri 4.0[4]. Karakteristik pekerjaan yang berubah akan mendisrupsi pekerjaan yang telah ada dan menggantikanya dengan pekerjaan dengan karakteristik baru [5]. Karakteristik baru pada pekerjaan juga membutuhkan kompetensi baru kepada para pekerja[6]. Tentunya perusahaan harus siap untuk saling bersaing dengan perusahaan yang lain[7]. Selanjutnya, perusahaan perlu memiliki keunggulan dan manajemen yang efektif untuk menghadapi persaingan tersebut[7]. Dengan demikian salah astu aspek yang berpengaruh besar terhadap kemajuan dan keberhasilan sebuah perusahaan adalah kinerja karyawannya[7]. Walaupun perusahaan tersebut memiliki teknologi yang canggih, namun tidak terdapat tenaga kerja didalamnya, perusahaan tidak akan dapat mencapai tujuannya[7].

Oleh karena itu, penentuan gaji yang tepat oleh perusahaan kepada karyawan adalah salah satu faktor yang berpengaruh secara internal terhadap kemajuan perusahaan. Selain itu, perusahaan juga harus bersedia mengeluarkan gaji bonus bagi karyawannya yang telah bekerja dengan maksimal dan sesuai dengan apa yang dibutuhkan oleh sebuah perusahaan. Sangat disayangkan, perkembangan perusahaan saat ini belum memliki suatu media keputusan untuk melakukan prediksi gaji karyawan berdasarkan kualitas data.

Karakteristik dataset yang digunakan untuk memprediksi gaji karyawan terdiri dari parameter-parameter berdasarkan faktor-faktor spesifik. Selanjutnya faktor-faktor tersebut akan diuji validitas dan korelasinya menggunakan pendekatan *machine learning*. Faktor-faktor tersebut akan diambil berdasarkan pedoman interpretasi koefisien korelasi [8]. Untuk menentukan faktor yang dominan terhadap prediksi gaji, maka koefisien korelasi yang akan digunakan adalah tingkat hubungan sedang, kuat, dan sangat kuat. Metode yang digunakan pada *machine learning* yaitu *regression*. *Regression* digunakan untuk melakukan prediksi gaji karyawan. Tentunya hasil prediksi gaji karyawan perlu divisualisasikan secara *realtime* untuk dapat digunakan oleh perusahaan dalam menentukan keputusan dengan cepat. Visualisasi hasil prediksi tersebut akan ditampilkan berbasis web base dengan *framework* Django.

## **Identifikasi Masalah**

Berikut adalah identifikasi permasalahan dari penelitian ini :

1. Bagaimana menganalisis karakteristik data gaji karyawan pada perusahaan?
2. Bagaimana cara menganalisis uji validitas dan korelasi dari dataset gaji karyawan pada perusahaan?
3. Bagaimana cara membuat model prediksi gaji karyawan yang tepat?
4. Bagaimana cara melakukan visualisasi data dari hasil model prediksi gaji karyawan?

## **Tujuan Penelitian**

Tujuan penilitian ini sebagai berikut :

1. Menganalisis data kepegawaian yang berkaitan dengan gaji karyawan.
2. Menganalisis uji validitas dan korelasi dari dataset gaji yang terdiri dari parameter gaji pegawai dan faktor-faktor terhadap prediksi gaji.
3. Membuat model prediksi dengan pendekatan *machine learning* menggunakan regresi.
4. Merancangan sistem berbasis web base dengan *framework* Django?

## **Manfaat Penelitian**

Manfaat penilitian ini sebagai berikut :

1. Sebagai wadah untuk memberikan inovasi baru dalam hal melakukan prediksi gaji karyawan.
2. Bagi perusahaan, penelitian ini diharapkan dapat membantu dalam menentukan keputusan yang tepat dalam penentuan gaji karyawan di masa yang akan datang.
3. Penelitian ini dapat menjadi memberikan kontribusi untuk melakukan pengembangan lebih lanjut.

## **Ruang Lingkup**

Ruang lingkup pada penelitian ini sebagai berikut :

* + 1. Dataset yang digunakan adalah gaji karyawan yang terdiri dari parameter-parameter yang spesifik.
    2. *Framework* yang digunakan untuk membuat visualisasi prediksi dari *machine learning* yaitu Django.
    3. Metode regresi digunakan untuk menguji korelasi dari parameter yang digunakan untuk memprediksi gaji karyawan.

## **Sistematika Penulisan**

Dalam laporan ini, terdapat dari lima bab yang mana setiap bab-nya memiliki pembahasan yg berbeda-beda. Berikut ini adalah pemaparan setiap bab.

1. BAB I PENDAHULUAN

Bab ini menjelaskan terkait dengan bagaimana cara untuk melakukan prediksi gaji karyawan.

1. BAB II LANDASAN TEORI

Bab ini menjelaskan keterhubungan penelitian-penelitian terdahulu yang mengkaji prediksi gaji dengan pendekatan *machine learning*.

1. BAB III ANALISIS DAN PERANCANGAN

Bab ini menjelaskan analisa dataset dalam menguji validitas dan korelasi terhadap parameter untuk memprediksi gaji karyawan dan merancang visualisasi hasil prediksi menggunakan *framework* djanggo.

1. BAB IV IMPLEMENTASI

Bab ini hasil prediksi gaji karyawan dengan menggunakan metode *machine learning* dan visualisasi menggunakan *framework* djanggo.

1. BAB V KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dan saran dari penelitian yang sudah dilakukan.

# BAB II

# LANDASAN TEORI

## **Tinjauan Studi**

Pada sebuah upaya dalam melakukan suatu analisis, maka dibutuhkan suatu panduan ataupun rujukan serta dukungan untuk setiap hasil analisis yang sudah ada sebelumnya. Yang tentunya panduan atau rujukan tersebut akan berkaitan dengan suatu analisis yang sedang dilakukan. Hasil dari penelitian-penelitian terdahulu tersebut terdiri dari topik dan pembahasan hasil penelitian yang telah dilakukan sebelumnya.

### **Penelitian Terkait**

Tabel 2. 1 Hasil Tinjauan Dari Penelitian Terkait

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Area Penelitian | Karakteristik Data | Metode | Hasil Penelitian |
| 1. | Gaji karyawan berdasar pengalaman lama bekerja.[7] | Data gaji pegawai dengan pengalaman lama bekerja, variabel gaji, variabel masa kerja. | *Machine Learning.* | Teknik model analisis data menggunakan model analisis regresi linear untuk pengetahui pengaruh variable bebas terhadap variable terikat. Dari hasil prediksi gaji karyawan berdasarkan pengalaman lama bekerja, terdapat pengaruh positif dan signifikan antara gaji dan masa kerja terhadap kinerja karyawan. |
| 2. | Harga saham perusahaan pelayaran PT. BULL.[9] | Data primer: informasi perusahaan. Data sekunder: data historis harga saham 2019-2020 dari Yahoo Finance dan Laporan Tahunan Bursa Efek Indonesia. *Purposive* *sampling*: perusahaan pelayaran PT. BULL. *Quota sampling: data time series* periode harian, variabel harga pembukaan, variabel harga tertinggi, variabel harga terendah, variabel harga penutupan, variabel volume saham selama 1 tahun 2 bulan (Juni 2019 – Juli 2020). | *Machine Learning.* | Penelitian ini menggunakan model *Cross Industry Standard Process for Data Mining* (CRISP-DM). Dari prediksi harga saham, hasil penelitian menunjukkan terdapat selisih antara harga penutupan saham luaran data testing dengan harga penutupan saham aktual yang ada di bursa saham. |
| 3. | Harga sembilan bahan pokok di DKI Jakarta.[10] | Data sembako DKI Jakarta (1 Januari 2016 – 31 Desember 2019). Variabel tanggal, variabel komoditas, variabel pasar, variabel harga. | *Machine Learning.* | Model yang digunakan dalam penelitian yaitu regresi linier berganda. Dari hasil prediksi yang telah dilakukan, persentase sumbangan pengaruh variabel bebas terhadap variabel terikat sebesar 84,2%, sedangkan sisanya sebesar 15,8% dimana dipengaruhi oleh variabel yang tak dimasukkan dalam penelitian ini. |
| 4. | Harga rumah dari 2 website jual beli rumah.[11] | Studi documenter *web scrapping* 2 website (1 Oktober 2020 – 31 Oktober 2020). Variabel luas lahan, variabel luas bangunan, variabel banyaknya kamar tidur, variabel banyaknya kamar mandi, variabel ketersediaan tempat parkir mobil. | *Machine Learning.* | Model dalam membuat prediksi adalah metode regresi linear. Dari hasil penerapan prediksi harga rumah, pengolahan awal data yang dilakukan pada data set 7442 data menjadi 794 data sangat mempengaruhi dalam tingkat akurasi prediksi harga rumah. Adapun hasil akurasi terbaik menghasilkan tingkat akurasi prediksi sebesar 0.8859830993050699 atau 88%. Namun, memiliki nilai galat cukup tinggi sebesar 259171.91 atau Rp. 259.171,91. |
| 5. | Pengadaan inventaris barang.[12] | Data Inventarsi Barang Dinas Pariwisata Pemuda dan Olahraga Kota Tasikmalaya 2012 – 2016. Variabel kode barang, variabel tahun, variabel harga barang. | *Machine Learning.* | Model yang digunakan dalam penelitian adalah metode regresi linier. Dari prediksi yang dilakukan, prediksi harga barang minimal yaitu sebesar 3011855.102, dan prediksi harga barang maksimal yaitu sebesar 23752745.511. Hasil akurasi RMSE adalah 0.934. |
| 6. | Prediksi kasus Covid-19 di Indonesia.[13] | Data yang dipakai dari databooks yang terdiri dari atribut total kasus, kasus sembuh, kasus meninggal dan kasus aktif, pada penelitian ini hanya menggunakan data kasus aktif. | *Machine Learning.* | Penelitian ini menggunakan model *backpropagation* dan regresi linear. Dari hasil simulasi Matlab dengan tiga fungsi pelatihan yaitu traincgb didapatkan nilai rata-rata error yaitu sebesar 0,017107, dengan fungsi pelatihan traingd didapatkan nilai rata-rata error sebesar -0,55116 serta fungsi pelatihan traingdx didapatkan nilai rata-rata error sebesar -3,82202 sehingga dapat disimpulkan fungsi pelatihan yang paling konvergen yaitu fungsi pelatihan terhadap traingdx denagn nilai rata-rata error paling kecil. |
| 7. | Prediksi Harga Emas.[14] | Dalam penelitian ini, data yang digunakan berasal dari internet. Kriteria atau variabel yang digunakan dalam penelitian ini, yaitu harga buka, harga beli (Input) dan harga jual (Output). | *Machine Learning* | Digunakan tiga model dalam penelitian ini, yaitu regresi linear, *backpropagation*, dan fuzzy mamdani. Hasil dari ketiga model menunjukkan bahwa korelasi dari regresi linear sangat bagus, yaitu 0,929. Dan nilai korelasi tertinggi dari ketiga metode berasal dari metode backpropagation. Hal ini terbukti bahwa dalam memprediksi harga emas menggunakan backpropagation lebih sedikit errornya ±0,05. |
| 8. | Prediksi Besaran Pendapatan Daerah.[15] | Data yang digunakan adalah data besaran pendapatan Kabupaten Deli Serdang tahun 2017 dan 2018, yang akan dibagi menjadi 2 bulan, Triwulan (3 bualn), Caturwulan (4 bulan), 1 semester (6 bulan) dan 1 tahun (12 bulan). | *Machine Learning* | Dalam penelitian ini model yang digunakan adalah regresi linear sederhana. Berdasarkan proses hasil perhitungan yang dilakukan, diketahui nilai prediksi besaran pendapatan daerah dinas pendapatan daerah Kab. Deli Serdang. Dengan menggunakan algoritma regresi linear sederhana, dinilai dapat memprediksi besaran pendapatan daerah dinas pendapatan daerah Kab. Deli Serdang untuk tahun berikutnya sehingga program-program yang telah direncanakan sebelumnya oleh dinas pendapatan dapat berjalan denagn lancar, dan juga dapat membuat program-program yang baru agar dapat meningkatkan pendapatan daerah untuk memajukan daerah tersebut. |
| 9. | Prediksi Tingkat Produksi Kopi.[16] | Data produksi kopi diperoleh dari BPS Kabupaten Manggarai dengan mengambil data produksi kopi lima tahun terakhir yaitu dari tahun 2011 – 2015. | *Machine Learning* | Dalam penelitian ini mode; yang digunakan adalah model regresi linear sederhana. Berdasarkan hasil penelitian dan pembahasan tentang prediksi tingkat produksi kopi menggunakan regresi linear sederhana maka dapat disimpukan bahwa data yang digunakan untuk memprediksi kopi merupakan data *time series*, dan setelah melakukan perhitungan prediksi pada tahun 2011 – 2015 nilai tertinggi pada tahun 2015 sebesar 1.537,38 ton dan nilai terendah pada tahun 2011 sebesar 1.109,944 ton. Setelah dilakukan pengujian menggunakan MSE dan MAPE diperoleh nilai MSE 43,112% dan MAPE 20,001% sehingga pengyjian menggunakan MAPE jauh lebih baik dalam menghitung akurasi prediksi produksi kopi. |
| 10. | Peramalan Penerimaan Mahasiswa Baru Universitas Samudra.[17] | Data didapat dari laman web Universitas Samudra. Data diambil dari data mahasiswa yang diterima 5 tahun sebelumnya. | *Machine Learning* | Model yang digunakan dalam penelitian adalah model regresi linear sederhana. Dapat disimpulkan bahwa penggunaan metode forecasting (peramalan) untuk melakukan peramalan jumlah PMB yang akan masuk ke Universitas Sambudra pada 1 tahun yang akan datang dapat dikatakan bermanfaat dan jumlah error yang didapat juga tidak terlalu besar sehingga peramalan ini dapat dipertimbangan untuk universitas agar pihak universitas dapat meningkatkan berbagai hal untuk mempertimbangkan apa-apa saja yang yang harus ditingkatkan agar penerimaan mahasiswa baru pada tahun selanjutnya dapat meningkat. |
| 11. | Prediksi gaji menggunakan teknik regresi. [18] | Data *position, level, salary.* Variabel *position*, variabel *level*, dan variabel *salary*. | *Machine Learning* | Model yang digunakan adalah model regresi linear dan regresi polinomial. emilih gaji dari grafik x-y membutuhkan banyak percobaan karena mungkin ada lebih dari satu grafik yang sesuai. Prediksi ini benar hingga waktu tertentu. Akurasi dapat diperoleh dengan menerapkan regresi k-terdekat. |
| 12. | Analisis empiris teknik regresi berdasarkan harga rumah  dan prediksi gaji. [19] | Variabel pada dataset *employee* *salary : salary, total years of experience, certification, lead.* Variabel pada dataset *house prices : price, sqft\_living, bedrooms, bathrooms, sqft\_living15, sqft\_lot, sqft\_above, sqft\_basement.* | *Machine Learning* | Model yang digunakan adalah *Simple Linear Regression* (SLR) dan *Multiple Linear Regression* (MLR). *Multiple Linear Regression* baik daripada *Simple Linear Regression*. Karena pada data House price, MLR memiliki R-Square 0,67 dan SLR 0,49. Serta pada data prediksi gaji, MLR memiliki R-Square 0,92 dan SLR 0,75. |
| 13. | Analisis korelasi antara gaji dan efisiensi inovasi enterprise berdasarkan psikologi entrepreneur.[20] | Data perusahaan manufaktur A-share China dari tahun 2012 -2016. *Explanatory variable : Monetary salary of senior manager, Senior management salary, Ordinary employee salary, Salary gap within the senior management team, Salary gap between senior managers and ordinary employees. Explained variable : Enterprise innovation efficiency. Control variable : Enterprise size, Return on assets, Asset-liability ratio, Years of establishment, Dummy variable*. | *Machine Learning* | Metode yang digunakan dalam penelitian ini yaitu *multiple regression. Monetary salary* dan *equity salary of senior managers, ordinary employee salary,* dan *the internal salary gap of the senior management team* semuanya berkorelasi positif secara signifikan pada tingkat 1% dengan efisiensi inovasi perusahaan. Korelasi antara *salary gap* antara *senior managers* dan *ordinary employees* dengan *enterprise innovation efficiency* tidak jelas. Koefisien korelasi regresi antar variabel berada dalam kisaran yang dapat diterima, menunjukkan bahwa model yang digunakan tidak memiliki multikolinearitas yang signifikan. |
| 14. | Regresi linear bivariat simpel dan aplikasinya pada data cuaca di Cilacap.[21] | Dataset dari BMKG Kabupaten Cilacap dari bulan Januari 2009 - Februari 2014. Dua variabel respon : curah hujan (Y1), kelembaban udara suatu wilayah (Y2) terhadap satu variabel respon : temperatur udara (X). | *Machine Learning* | Model dalam penelitian ini adalah menggunakan metode regresi linear bivariat simple. Model peramalan untuk Y1 adalah Y(1) = −894,130 + 45,892X yang berarti jika tempertur udara naik sebesar satu derajat Celcius maka curah hujan akan naik sebesar 45,892mm serta nilai rata-rata error bulanan = -0,00697mm dan MDEE(1) = 151,2132. Model peramalan untuk Y2 adalah Y(2) = 78,0433+0,1581X yang artinya jika tempertur udara naik sebesar satu derajat Celcius maka kelembaban udara akan naik 0,1581 persen seta nilai rata-rata error bulanan = 0,000441 persen dan MDE(2) = 1,206636. |
| 15. | Analisis regresi linier berganda dalam estimasi produktivitas tanaman padi di Kabupaten Karawang.[22] | Data Dinas Pertanian Kehutanan Perkebunan dan Peternakan Kabupaten Karawang. Variabel terikat : produktivitas padi (kw/ha). Data produktivitas padi dihasilkan dari data produksi dibagi luas panen. Variabel bebas : yaitu produksi, luas panen, luas tanam, rata-rata curah hujan, dan rata-rata hari hujan | *Machine Learning* | Model yang digunakan dalam penelitian ini adalah regresi liner berganda. Berdasarkan model regresi didapat 80,46% faktor-faktor produktivitas padi. Variabel-variabel yang mempengaruhi peningkatan jumlah produktivitas padi yaitu variabel produksi dan curah hujan, sedangkan variabel-variabel yang mempengaruhi penurunan jumlah produktivitas yaitu variabel luas panen, luas tanam, dan hari hujan. kesalahan relatif regresi yang diperoleh yaitu 4,642%. |
| 16. | Model regresi multivariat analisis kesejahteraan pedagang kaki lima berdasarkan karakteristik sosial ekonomi. [23] | Variabel bebas : Umur, Jenis kelamin, Status perkawinan, Tingkat pendidikan terakhir, Status migrasi, Tipe rumah, Sifat layanan pedagang kaki lima, Curahan jam kerja per hari, Lama menjalankan usaha, Jumlah tenaga kerja/karyawan diluar tenaga kerja keluarga yang membantu menjalankan usaha, Jumlah tenaga kerja keluarga yang membantu menjalankan usaha, Ketersediaan buku untuk pembukuan kegiatan usaha, Status Registrasi Usaha, Jenis dagangan, Sarana fisik pedagang kaki lima, Alternatif sumber pendapatan di luar pekerjaan sebagai PKL. Variabel dependen : Tingkat Kesejahteraan, dijabarkan dalam variabel: Pendapatan responden, Status Pekerjaan, dikelompokkan dalam: pekerja berusaha sendiri; berusaha sendiri dibantu oleh pekerja sementara/tidak dibayar; berusaha sendiri dibantu oleh pekerja permanen/dibayar; Karyawan/ Pekerja; Pekerja tidak dibayar. | *Machine Learning* | Model yang digunakan dalam penelitian adalah analisis *multivariate* yaitu analisis regresi linier dan *logistic ordinal*. Variabel yang signifikan berpengaruh terhadap rata-rata pendapatan pedagang kaki lima adalah tingkat pendidikan, curahan jam kerja, dan jumlah tenaga kerja diluar tenaga kerja keluarga yang ikut membantu. Model rata-rata pendapatan pedagang kaki lima adalah Y = -1.982 + 0.654pendidikan + 0.134curahan jam kerja + 0.817Jumlah tenaga kerja non keluarga. Variabel yang berpengaruh signifikan terhadap status pekerjaan pedagang kaki lima yaitu status perkawinan, sifat layanan dagangan, curahan jam kerja, serta jumlah tenaga kerja diluar tenaga kerja keluarga. Model dugaan yang menjelaskan status pekerjaan pedagang kaki lima yaitu G(x) = -16.308 - 0.519 status kawin + 0,739 sifat layanan + 1,19663 curahan jam kerja - 1,062 Jumlah tenaga kerja non keluarga. |

# BAB III

# ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan mengenai analisis sistem yang berjalan, analisis sistem yang akan dibangun, perancangan sistem (UML), dan perancangan database. Aplikasi Website untuk Prediksi Gaji Pegawai ini merupakan aplikasi yang menggunakan beberapa software yaitu Visual Studio Code, Framework Django, Jupyter Notebook, Tableau, PHPMyAdmin, dan XAMPP. Dengan aplikasi ini nantinya dapat membantu admin untuk melakukan prediksi gaji pegawai berdasarkan pengalaman lama bekerja.

## **Analisis Sistem**

Analisis sistem ialah penjabaran dari suatu sistem informasi yang utuh ke berbagai macam bagian-bagian komponennya yang bertujuan untuk mengevaluasi permasalahan atau kendala yang terjadi pada suatu sistem, sehingga nantinya dapat dilakukan perbaikan ataupun pengembangan pada sistem tersebut.

Perancangan sistem merupakan kegiatan merancang dan mendesain suatu sistem yang baik yang dimana kegiatan tersebut adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur-prosedur untuk mendukung operasi sistem tersebut. Tujuan dari perancangan sistem ialah untuk memenuhi kebutuhan para pemakai sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer dan ahli-ahli yang terlibat didalamnya.

Pada bagian ini, dibahas tentang analisis prosedur yang digambarkan dalam bentuk flowmap BPMN, pengkodean, analisis sistem fungsional, dan analisis sistem non fungsional yang meliputi perangkat keras dan perangkat lunak yang digunakan. Selain itu pada bagian ini juga akan dibahas mengenai analisis user yang terlibat dalam aplikasi tersebut. Tahapan ini sangat penting dalam membantu melanjutkan tahapan yang selanjutnya yaitu tahapan perancangan.

### **Analisis Sistem Yang Sedang Berjalan**

Pada sub bab ini akan dijelaskan mengenai proses prediksi gaji pegawai yang sedang berjalan. Analisa sistem yang sedang berjalan bertujuan untuk mengetahui dan menggambarkan lebih lanjut mengenai bagaimana cara kerja sistem tersebut. Sistem yang berjalan saat ini menjelaskan tentang flowmap proses prediksi gaji pegawai. Flowmap tersebut merupakan gambaran alur proses prediksi gaji pegawai yang sedang berjalan pada aplikasi yang nantinya akan dibangun.



Gambar 3. 1 Flowmap Pemrosesan Gaji

### **Analisis Sistem Yang Akan Dibangun**

Pada sub bab ini akan dijelaskan mengenai Aplikasi Website Prediksi Gaji Pegawai Berdasarkan Pengalaman Lama Bekerja akan dibangun. Analisa sistem yang sedang berjalan memiliki tujuan untuk memberikan gambaran dan mengetahui lebih lanjut bagaimana cara kerja sistem tersebut. Strategi yang digunakan dalam menganalisis sistem yang akan dibangun ini, adalah dengan membongkar atau menterjemahkan dalam bentuk flowmap BPMN.

#### **Flowmap Login**

Pada bagian ini menjelaskan tentang flowmap proses *login* admin untuk masuk ke aplikasi website prediksi gaji. Flowmap tersebut merupakan gambaran alur proses *login* admin yang akan dibangun pada aplikasi ini.

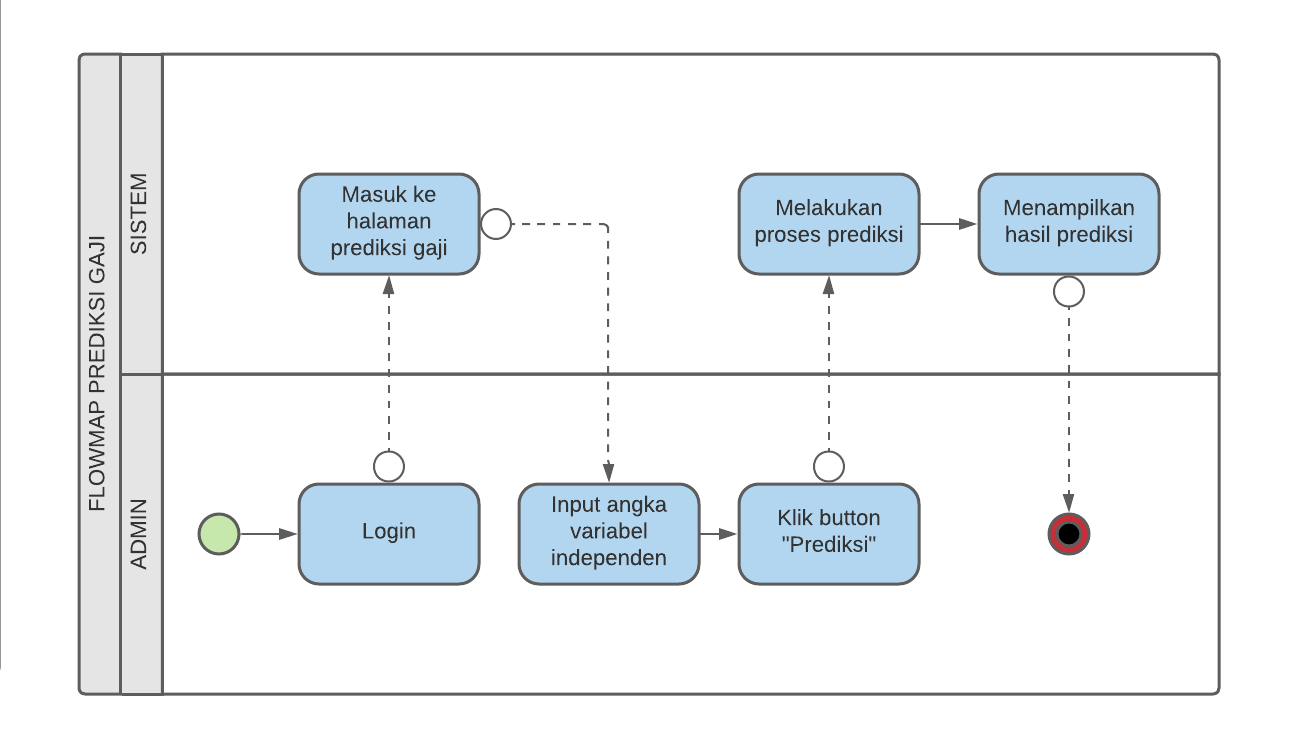
Gambar 3. 2 Flowmap Proses Login Admin

Keterangan :

1. Admin dapat membuka Aplikasi Prediksi Gaji Pegawai.
2. Aplikasi menampilkan halaman *login* .
3. Admin dapat melakukan penginputan username serta password.
4. Setelah itu, sistem dapat melakukan pengecekan apakah username serta password yang diinputkan admin valid atau tidak.
5. Jika username dan password sesuai, maka admin dapat masuk ke halaman .utama aplikasi. Sedangkan jika username dan password tidak sesuai, maka admin akan tetap berada di halaman *login.*

#### **Flowmap Prediksi Gaji**

Pada bagian ini menjelaskan tentang flowmap proses prediksi gaji pegawai yang dilakukan oleh admin. Flowmap tersebut merupakan gambaran alur proses prediksi gaji pegawai oleh admin yang akan dibangun pada aplikasi ini.



Gambar 3. 3 Flowmap Proses Prediksi Gaji Pegawai

Keterangan :

1. Admin dapat melakukan *login.*
2. Setelah admin berhasil *login*, aplikasi akan menampilkan halaman dashboard.
3. Setelah itu, admin dapat memilih menu Prediksi untuk beralih ke halaman prediksi.
4. Pada halaman prediksi, admin dapat menginputkan angka berupa variabel independent (usia, level pekerjaan, total tahun bekerja, dan tahun di perusahaan) pada form yang disediakan.
5. Setelah diinputkan, aplikasi akan menampilkan hasil prediksi gaji pegawai berdasarkan pengalaman lama bekerja.

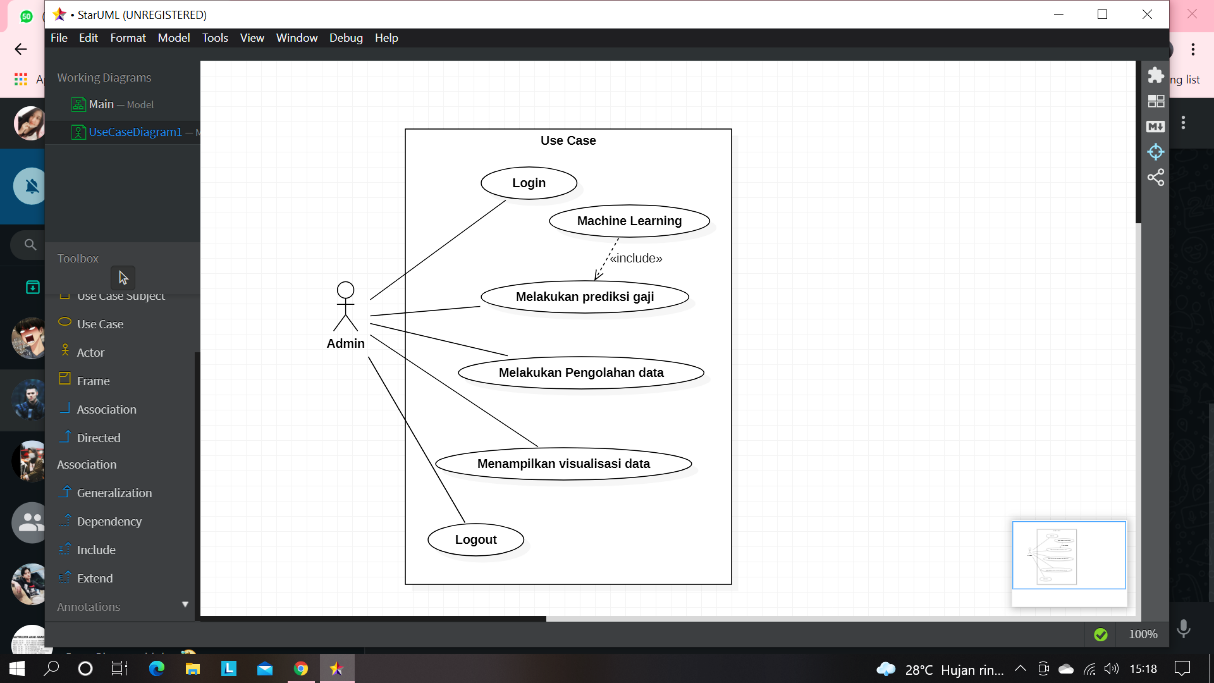
## **Perancangan Sistem (UML)**

UML adalah singkatan dari Unified Modeling Language yang didefinisikan sebagai sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau *software* berbasis objek. UML juga dapat dikatakan sebagai bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan, dan membangun suatu sistem perangkat lunak.

UML adalah suatu bentuk metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tools* untuk mendukung pengembangan sistem tersebut. Sehingga, UML juga dapat menjadi salah satu cara untuk mempermudah dalam melakukan pengembangan aplikasi yang berkelanjutan. Oleh karenanya, UML juga dapat menjadi suatu alat bantu untuk transfer ilmu tentang sistem yang akan dikembangkan dari satu developer ke developer lainya.

### **Use Case Diagram**

Use case diagram yaitu suatu gambaran graphical untuk memodelkan seluruh proses bisnis berdasarkan perspektif pengguna sistem dari beberapa atau semua aktor, use case, dan interaksi yang memperkenalkan suatu sistem. Use case diagram secara sederhana merupakan sebuah sarana bantu untuk melakukan pendefinisian apa yang ada di luar sistem (aktor) dan apa yang harus dilakukan oleh sistem yang sedang dikembangkan.



Gambar 3. 4 Use Case Diagram

1. **Definisi Aktor**

Pada bagian ini akan dijelaskan actor-aktor yang terlihat dalam (judul proyek).

Tabel 3. 1 Definisi Aktor

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Admin | * Login * Melakukan Prediksi Gaji * Melakukan Pengolahan Data * Melakukan Visualisasi Data * Logout |

1. **Definisi Use Case**

Tabel 3. 2 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Machine Learning | Merupakan aktivitas interaksi antara manusia dengan mesin. Dalam hal ini, admin membuat model prediksi *Machine Learning*. |
| 2 | Login | Merupakan aktivitas login yang dilakukan oleh admin. Sebelum admin masuk ke aplikasi ia harus melakukan login terlebih dahulu. |
| 3 | Melakukan Prediksi Gaji | Merupakan aktivitas memprediksi gaji karyawan yang dilakukan oleh admin dengan parameter lama bekerja seorang karyawan. |
| 4 | Melakukan Pengolahan Data | Merupakan aktivitas insert, read, update dan delete data karyawan yang dilakukan oleh admin. |
| 5 | Menampilkan Visualisasi Data | Merupakan aktivitas visualisasi grafik data karyawan yang dilakukan oleh admin. |
| 6 | Logout | Merupakan aktivitas logout yang dilakukan oleh admin. Admin dapat keluar dari aplikasi jika telah selesai melakukan pekerjaan pada aplikasi. |

1. **Skenario Use Case**

Skenario use case diharapkan setelah berjalannya fungsional use case. Selain itu juga diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang dilakukan oleh actor. Setiap use case akan diberikan sebuah scenario yang akan menjelaskan secara detail interaksi yang ada di dalamnya.

Tabel 3. 3 Skenario Use Case Machine Learning

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Machine Learning |
| **Tujuan** | Pembuatan model prediksi |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Membuat model prediksi | * Melakukan olah data gaji karyawan |
| **Kondisi Akhir** | Masuk pada aplikasi |

Tabel 3. 4 Skenario Use case Login Admin

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Login |
| **Tujuan** | Sebelum masuk pada aplikasi |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Input username dan password | * Mengecek data dan menentukan aktor tersebut admin atau user |
| **Kondisi Akhir** | Masuk pada aplikasi |

Tabel 3. 5 Skenario Use case Melakukan Prediksi Gaji Karyawan

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melakukan Predikis Gaji Karyawan |
| **Tujuan** | Melakukan Predikis Gaji Karyawan |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Menginput Data | * Melakukan Prediksi Gaji |
| **Kondisi Akhir** | Admi dapat memprediksi gaji |

Tabel 3. 6 Skenario Use case Melakukan Visualisasi Data

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melakukan Visualisasi Data |
| **Tujuan** | Melakukan visualisasi data karyawan |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Input data karyawan | * Menampilkan grafik visualisasi data karyawan |
| **Kondisi Akhir** | Admin dapat memvisualisasikan data |

Tabel 3. 7 Skenario Use case Melakukan Pengolahan Data

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Melakukan Pengolahan Data |
| **Tujuan** | Melakukan pengolahan data karyawan |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Insert, read, update, dan delete data karyawan. | * Mengubah data karyawan yang diubah oleh admin |
| **Kondisi Akhir** | Admin dapat mengelola data |

Tabel 3. 8 Skenario Use case Logout Admin

|  |  |
| --- | --- |
| **Identifikasi** | |
| **Nomor** | 1 |
| **Nama** | Logout |
| **Tujuan** | Keluar dari aplikasi |
| **Deskripsi** | |
| **Aktor** | Admin |
| **Skenario Utama** | |
| **Kondisi Awal** |  |
| **Aksi Aktor** | **Reaksi Sistem** |
| * Menekan tombol logout | * Melakukan proses keluar dari aplikasi |
| **Kondisi Akhir** | Keluar dari aplikasi |

### **Class Diagram**

Class diagram ialah diagram UML yang mendeskripsikan suatu struktur dari sebuah sistem yang dibuat dari kelas-kelas dengan relasi - relasinya. Class diagram juga dapat menggambarkan jenis-jenis objek yang terdapat pada sistem dan berbagai hubungan statis yang terdapat di antara mereka. Class diagram ini dapat menunjukkan properti dan operasi sebuah kelas serta batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

### **Sequence Diagram**

Sequence diagram adalah diagram yang mendeskripsikan serta menjelaskan suatu interaksi objek yang berdasarkan urutan waktu. Interaksi tersebut diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Sequence diagram dapat digunakan untuk menggambarkan urutan atau prosedur yang harus dilakukan untuk dapat menghasilkan sesuatu seperti pada use case diagram.

1. ***Sequence* Diagram *Login* Admin**

Berikut ini merupakan *sequence* diagram Login Admin menjelaskan hubungan antara admin pada aplikasi.

Berikut Sequence Diagram Login Admin pada gambar.



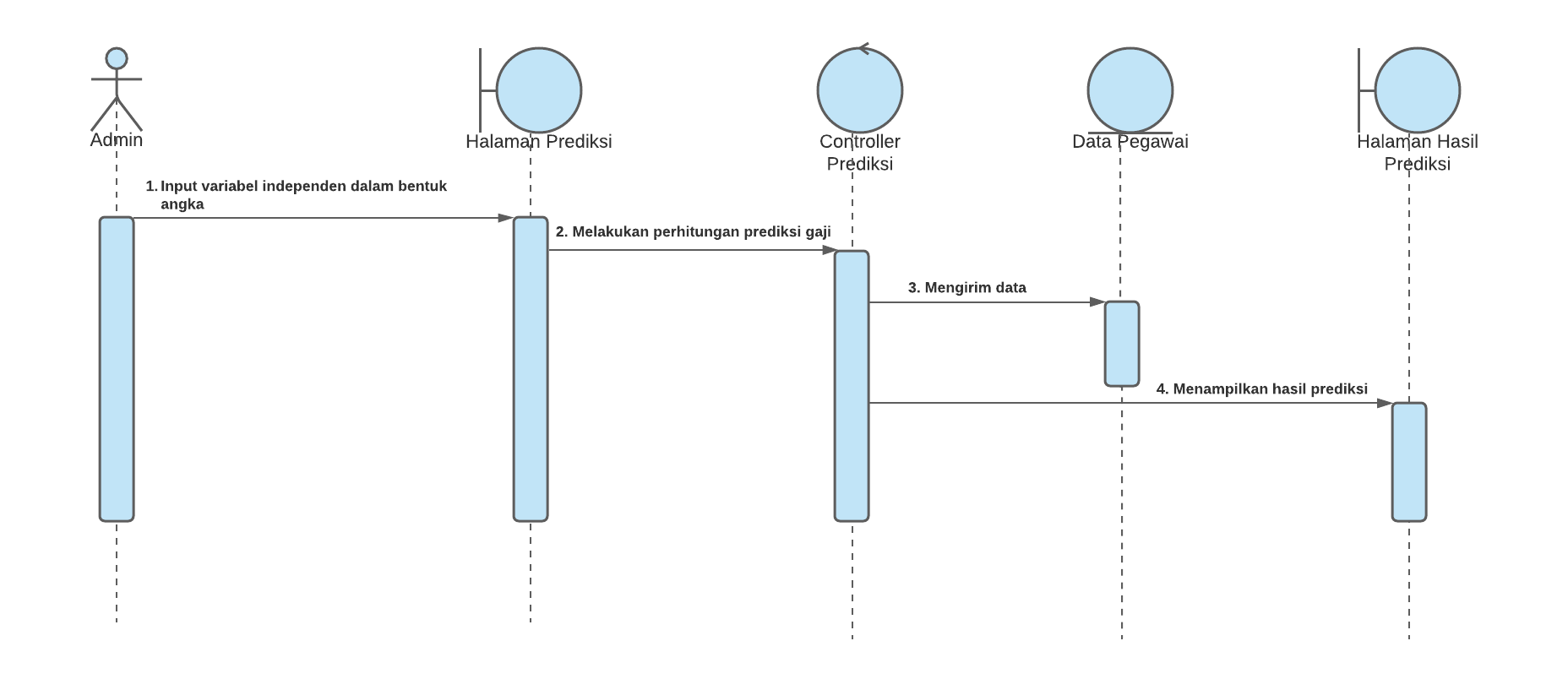
Gambar 3. 5 Sequence Diagram Login Admin

Keterangan :

1. Admin menginputkan *username* dan *password* di halaman login.
2. Validasi username dan password diteruskan ke Controller login.
3. Controller login mengambil data ke database data admin untuk validasi login.
4. Jika username dan password valid pada data admin maka akan ditampilkan halaman dashboard.
5. Jika username dan password tidak valid pada data admin maka akan ditampilkan halaman login.
6. **Sequence Diagram Melakukan Prediksi Gaji**

Berikut ini merupakan sequence diagram melakukan predikis gaji menjelaskan hubungan antara admin dan aplikasi. Admin dapat melakukan prediksi gaji dengan cara menginput data.

Berikut Sequence Diagram Melakukan Prediksi Gaji pada gambar.



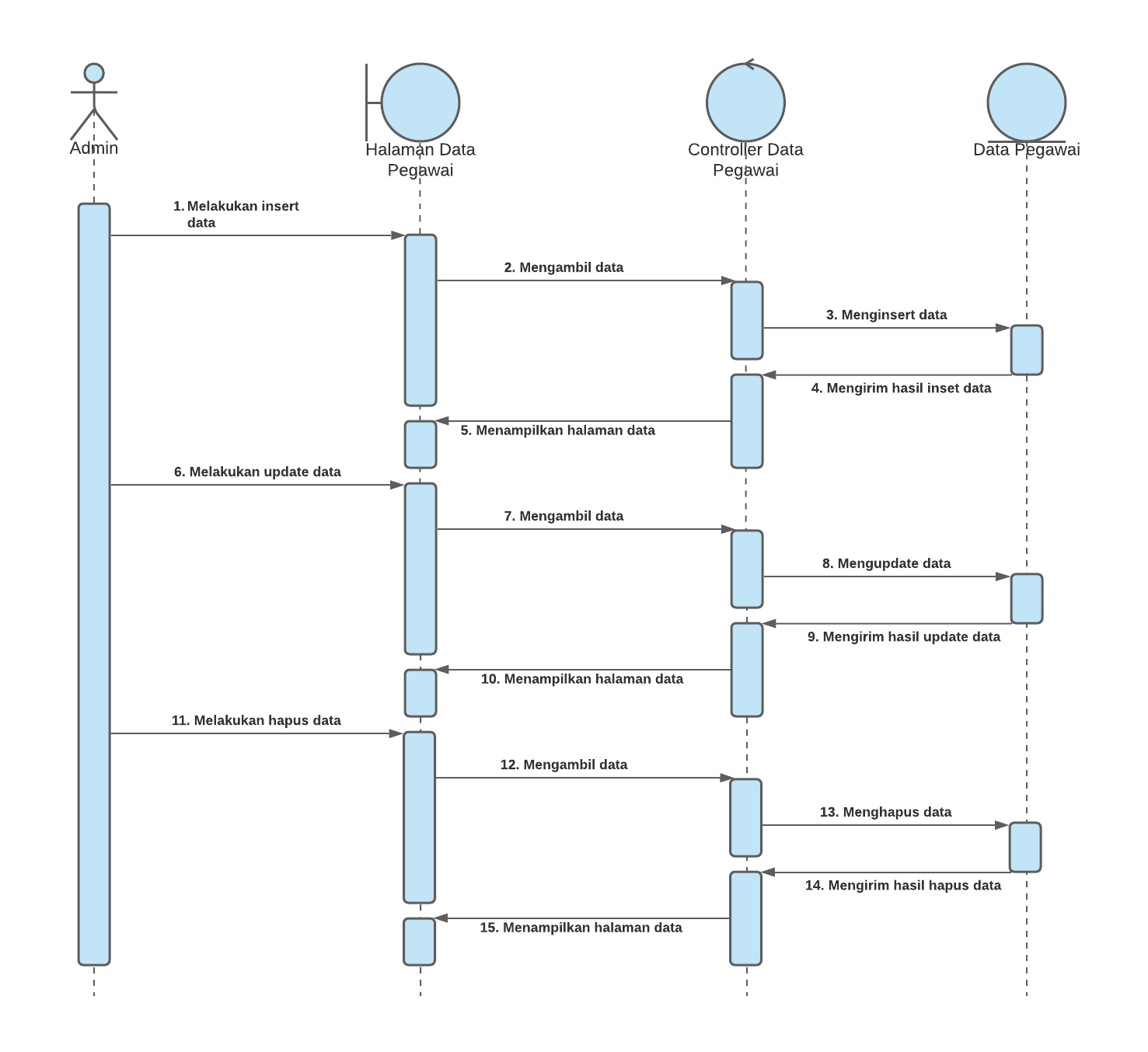
Gambar 3. 6 Sequence Diagram Melakukan Prediksi Gaji

Keterangan :

1. Admin menginputkan independen dalam bentuk angka.
2. Sistem melakukan perhitungan prediksi gaji.
3. Controller prediksi mengirim data ke database data pegawai dan memproses data.
4. Hasil data yang dihitung akan ditampilkan pada halaman hasil prediksi.
5. **Sequence Diagram Melakukan Pengolahan Data**

Berikut ini merupakan sequence diagram mengelola data menjelaskan hubungan antara admin pada aplikasi. Admin dapat insert, read, update, dan delete pada data alat.

Berikut Sequence Diagram Melakukan Pengolahan Data pada gambar.



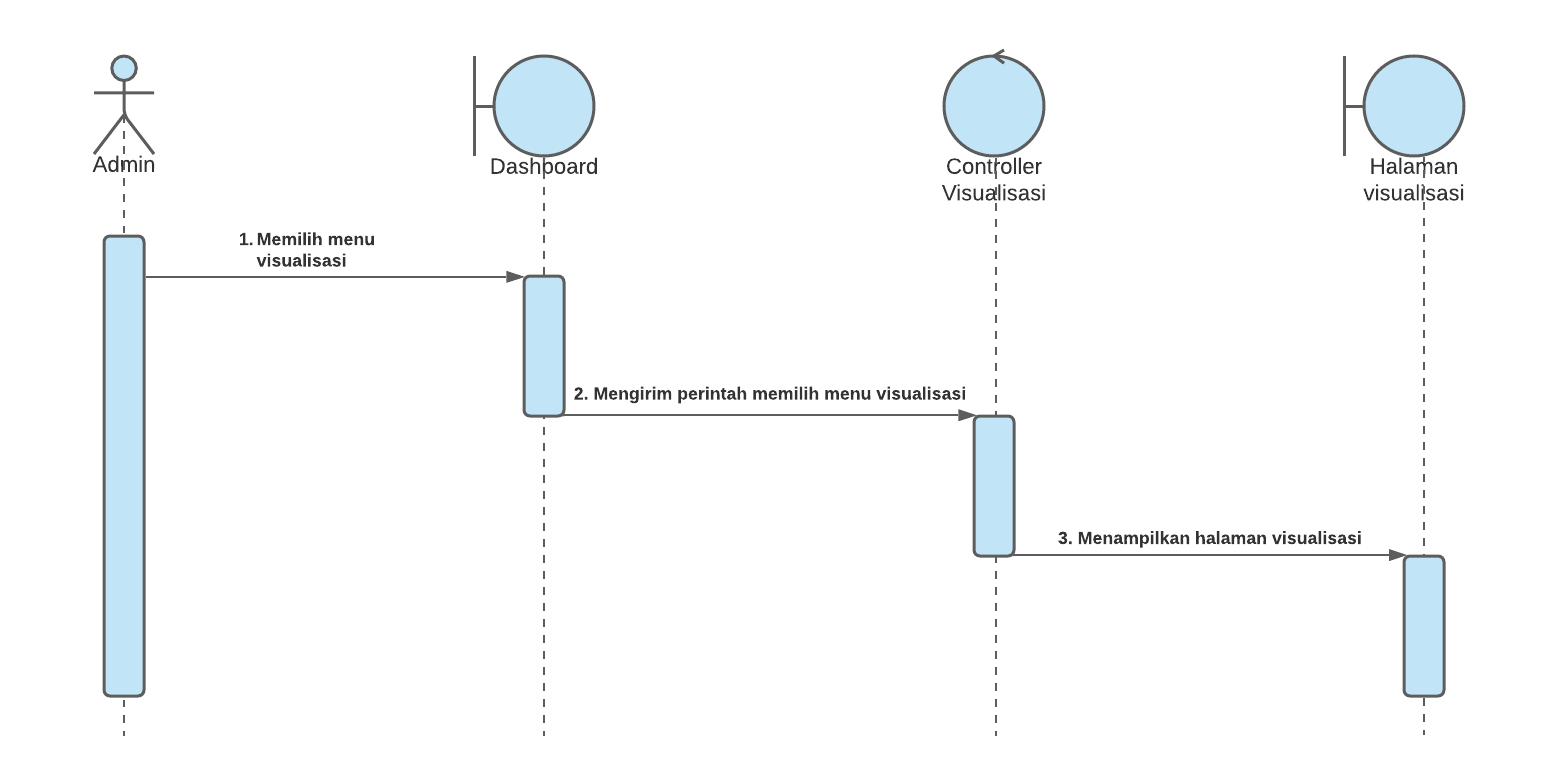
Gambar 3. 7 Sequence Diagram Melakukan Pengolahan Data

Keterangan :

1. Admin memilih menu yang akan dikelola pada menu di halaman data pegawai. Menu yang ada yaitu tambah data, update data, dan hapus data.
2. Halaman data pegawai mengambil data (baik antara input, update atau hapus) untuk diteruskan ke controller data pegawai.
3. Controller data pegawai mengirim hasil olah data ke data pegawai.
4. Data pegawai mengirim hasil olah data ke controller data pegawai.
5. Controller data pegawai menampilkan halaman data pegawai.
6. **Sequence Diagram Melakukan Visualisasi Data**

Berikut ini merupakan sequence diagram melakukan visualisasi data menjelaskan hubungan antara admin pada aplikasi.

Berikut Sequence Diagram Melakukan Visualisasi Data pada gambar.



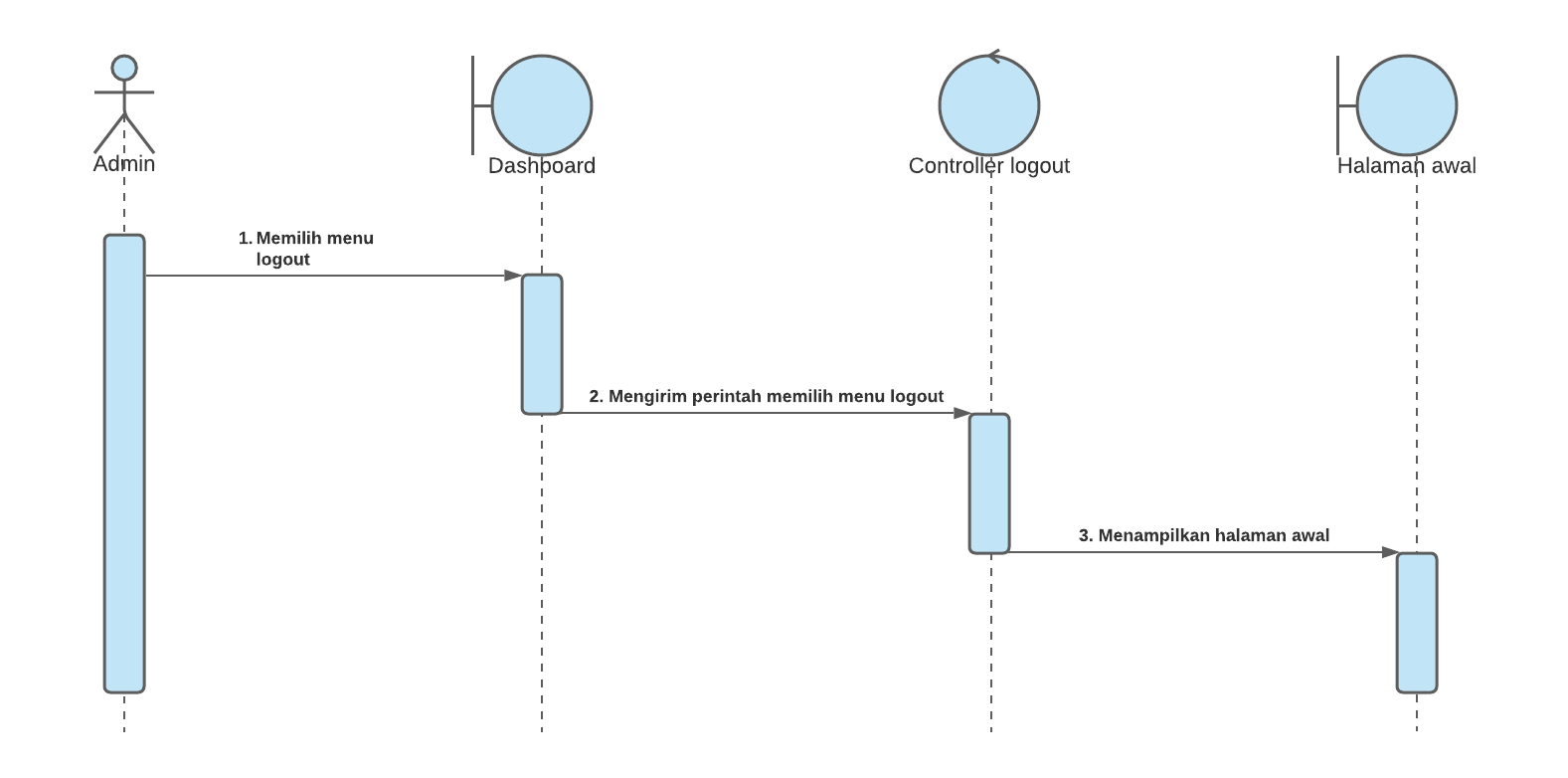
Gambar 3. 8 Sequence Diagram Melakukan Visualisasi Data

Keterangan :

1. Admin memilih menu halaman visualisasi pada dashboard.
2. Dashboard mengirim perintah memilih menu visualisasi ke controller visualisasi.
3. Controller visualisasi menampilkan halaman visualisasi.
4. **Sequence Diagram Melakukan Visualisasi Data**

Berikut ini merupakan sequence diagram *logout* Admin menjelaskan hubungan antara admin pada aplikasi.

Berikut Sequence Diagram *logout* Admin pada gambar.



Gambar 3. 9 Sequence Diagram logout admin

Keterangan :

1. Admin memilih menu logout pada dashboard.
2. Dashboard mengirim perintah memilih menu logout ke controller logout.
3. Controller logout melakukan proses logout kemudian menuju untuk menampilkan halaman awal.

### **Collaboration Diagram**

Collaboration diagram dapat dipakai untuk memodelkan interaksi antar objek yang ada di dalam sistem. Berbeda dari sequence diagram yang lebih menunjukkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram ini lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

1. **Collaboration Diagram Melakukan Login**



Gambar 3. 10 Collaboration Diagram Melakukan Login

1. **Collaboration Diagram Melakukan Prediksi Gaji**



Gambar 3. 11 Collaboration Diagram Melakukan Prediksi Gaji

1. **Collaboration Diagram Melakukan Pengolahan Data**



Gambar 3. 12 Collaboration Diagram Melakukan Visualisasi Data

1. **Collaboration Diagram Melakukan Visualisasi Data**

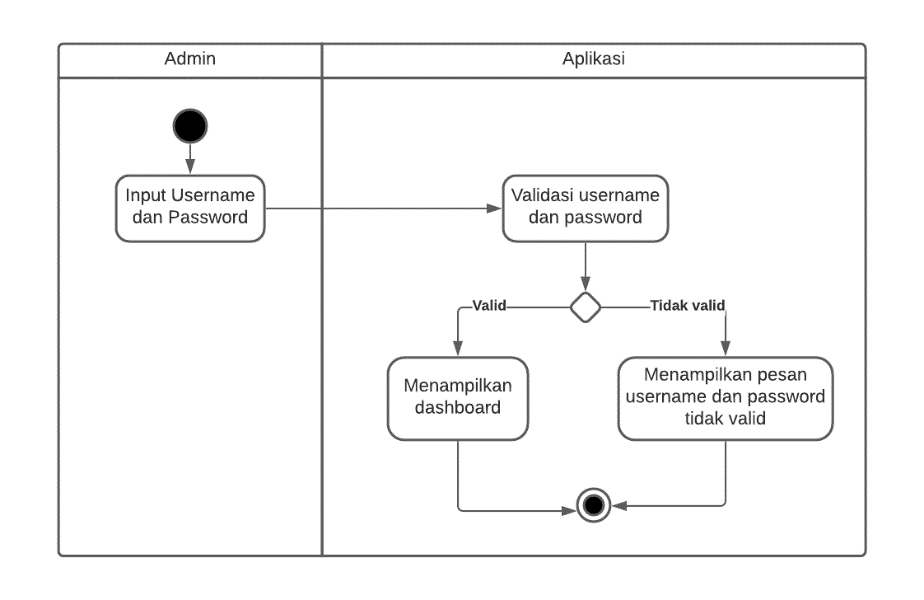


Gambar 3. 13 Collaboration Diagram Melakukan Visualisasi Data

### **Activiy Diagram**

Activity diagram dapat didefinisikan sebagai diagram yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Activity diagram adalah suatu teknik yang digunakan untuk mendeskripsikan logika procedural, proses bisnis serta aliran kerja dalam banyak kasus. Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, hasil akhir yang mungkin terjadi, hingga bagaimana mereka berakhir.

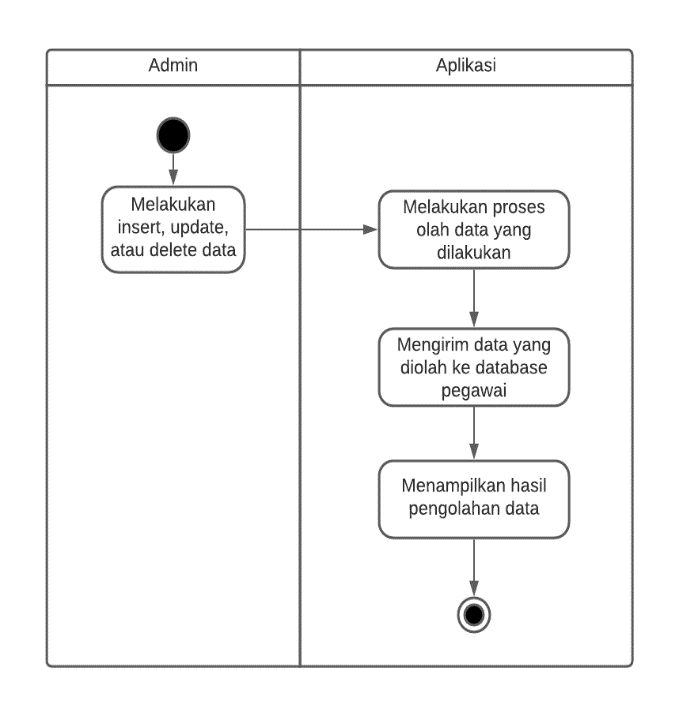
1. **Activity Diagram Login Admin**



Gambar 3. 14 Activity Diagram Login Admin

Keterangan :

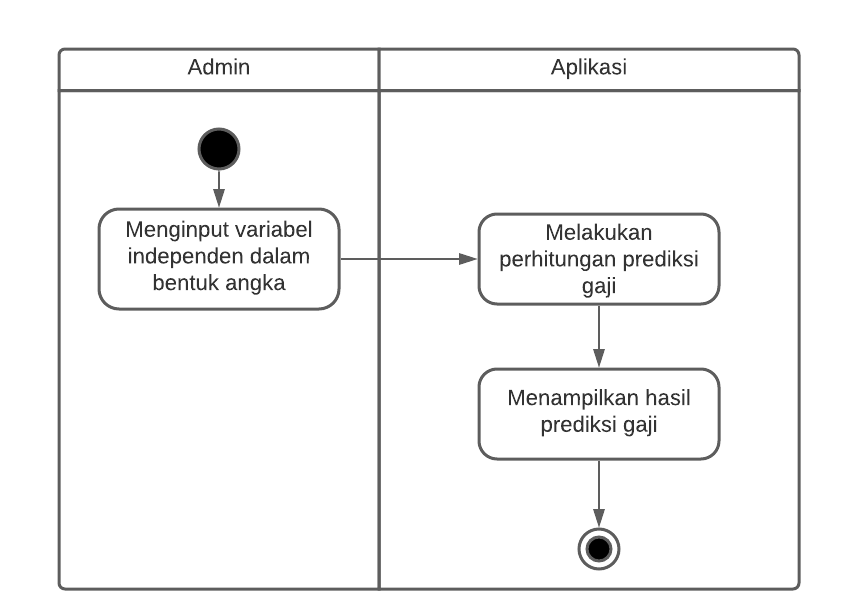
1. Admin menginput username dan password.
2. Aplikasi melakukan validasi username dan password.
3. Jika valid akan menampilkan halaman dashboard.
4. Jika tidak valid menampilkan pesan username dan password tidak valid.
5. **Activity Diagram Melakukan Pengolahan Data**



Gambar 3. 15 Activity Diagram Melakukan Pengolahan Data

Keterangan :

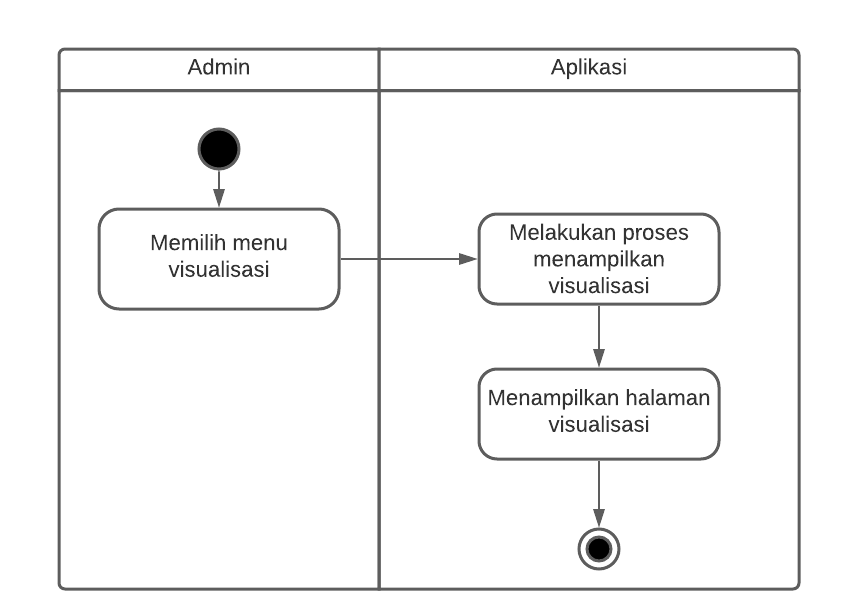
1. Admin melakukan olah data seperti insert, update, dan delete data.
2. Aplikasi melakukan proses olah data yang dilakukan.
3. Aplikasi mengirim data yang diolah ke database pegawai.
4. Aplikasi menampilkan data hasil create/update/delete.
5. **Activity Diagram Melakukan Prediksi Gaji**



Gambar 3. 16 Activity Diagram Melakukan Prediksi Gaji

Keterangan :

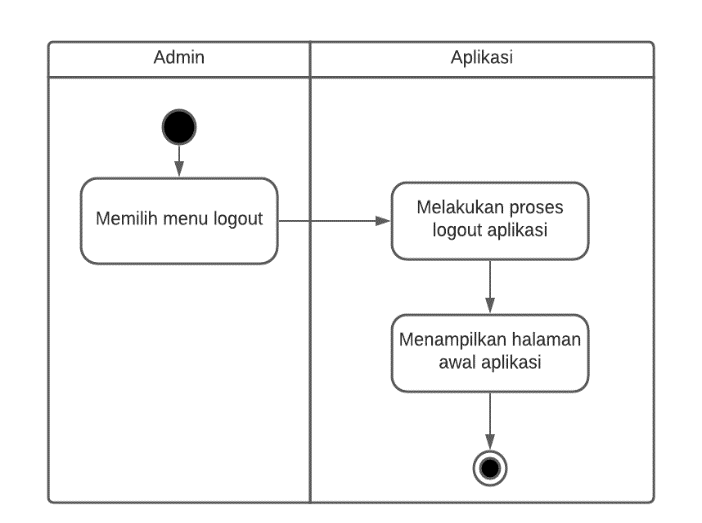
1. Admin menginputkan variabel independent dalam bentuk angka pada form prediksi.
2. Aplikasi melakukan proses perhitungan prediksi gaji.
3. Aplikasi menampilkan hasil prediksi.
4. **Activity Diagram Melakukan Visualisasi**



Gambar 3. 17 Activity Diagram Melakukan Visualisasi

Keterangan :

1. Admin memilih menu visualisasi.
2. Aplikasi melakukan proses menampilkan visualisasi.
3. Aplikasi menampilkan halamann visualisasi.
4. **Activity Diagram Logout Admin**



Gambar 3. 18 Activity Diagram Logout Admin

### **Statechart Diagram**

Statechart Diagram dapat menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) dari suatu objek pada sistem sebagai akibat dari stimuli yang diterimanya. Pada umumnya statechart diagram dapat menjelaskan atau menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

1. **Statechart Diagram Login Admin**



Gambar 3. 19 Statechart Diagram Login Admin

1. **Statechart Diagram Melakukan Prediksi**



Gambar 3. 20 Statechart Diagram Melakukan Prediksi

1. **Statechart Diagram Pengolahan Data**



Gambar 3. 21 Statechart Diagram Pengolahan Data

1. **Statechart Diagram Visualisasi**



Gambar 3. 22 Statechart Diagram Visualisasi

### **Component Diagram**

Component diagram dapat menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) diantaranya. Komponen piranti lunak atau yang biasa disebut dengan perangkat lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun runtime. Pada umumnya komponen dapat terbentuk dari beberapa class dan atau package, tapi dapat juga dibentuk dari komponen-komponen yang lebih kecil.



Gambar 3. 23 Component Diagram

### **Deployment Diagram**

Deployment/physical diagram menggambarkan detail bagaimana suatu komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), dan bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi, server, serta hal-hal lain yang bersifat fisik. Sebuah *node* terdiri dari server, *workstation*, atau piranti keras lain yang digunakan untuk melakukan *deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat pula didefinisikan dalam diagram ini. (Dharwiyanti : 2003)



Gambar 3. 24 Deployment Diagram

## **Perancangan Database**

Perancangan database merupakan proses untuk menentukan dan pengaturan data yang dibutuhkan untuk mendukung rancangan sistem, agar teciptanya pemrosesan data yang lebih efisien. Struktur tabel meliputi nama tabel, tipe data, nama atribut dan data relasi seperti primary dan foreign key.

### **CDM (Conceptual Data Model)**



Gambar 3. 25 Conceptual Data Model

### **PDM (Physical Data Model)**



Gambar 3. 26 Physical Data Model

### **ERD (Entity Relationship Diagram)**



Gambar 3. 27 Entity Relationship Diagram

## **Struktur Menu**

Berikut adalah struktur menu dari aplikasi Prediksi Gaji Pegawai secara keseluruhan :



Gambar 3. 28 Struktur Menu Aplikasi

## **Antarmuka (*Interface*)**

Impelementasi sistem memiliki bagian berupa pemaparan mengenai tampilan pada sistem serta kegunaan dari setiap halaman. Implementasi dari hasil perancanggan menggunakan bahasa pemrograman python dengan framework Django.

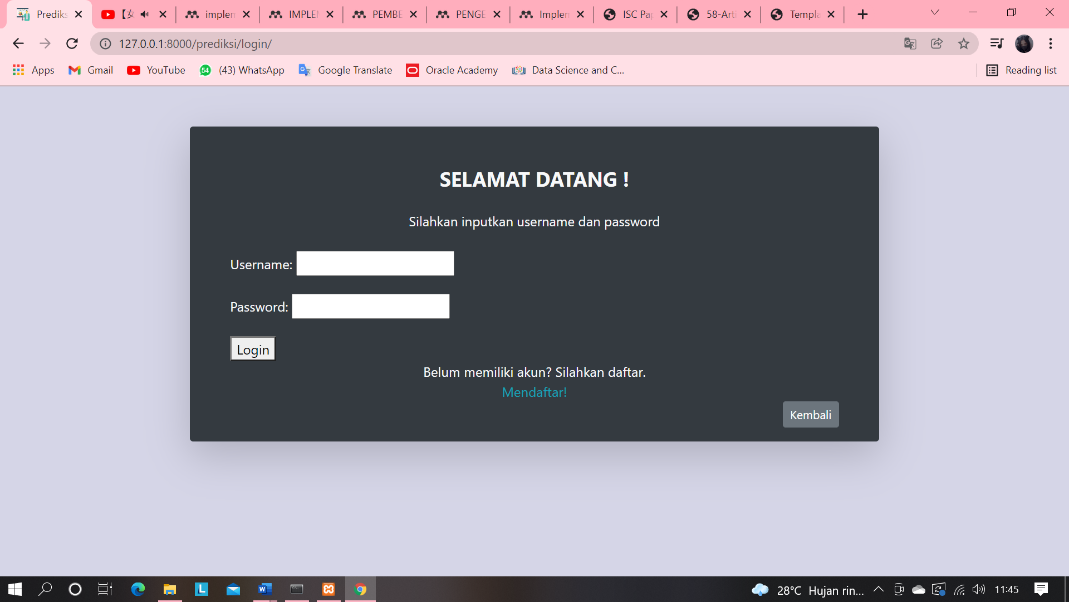
### **Antarmuka Halaman Depan**



Gambar 3. 29 Halaman Depan Sistem

Gambar diatas merupakan implementasi dari halaman depan sistem. Pada halaman ini, *user* dapat menekan button “Login” pada navbar untuk masuk ke dalam sistem.

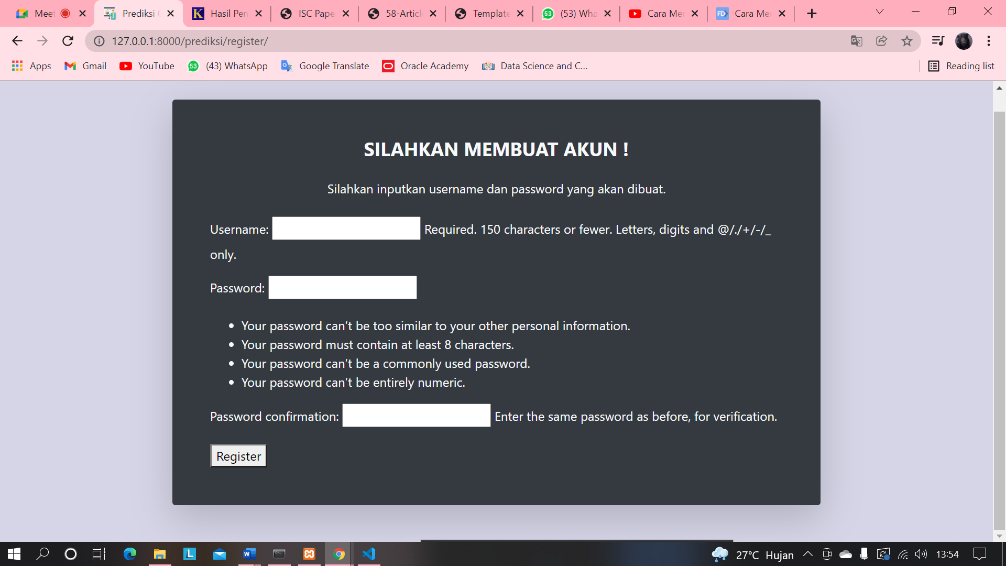
### **Antarmuka Halaman Login**



Gambar 3. 30 Halaman Login Sistem

Gambar diatas merupakan implementasi dari halaman *login*. Sebelum masuk ke halaman *dashboard*, *user* harus menginputkan *username* dan *password* yang sesuai. Jika sesuai, maka *user* akan di arahkan ke halaman *dashboard*. Jika tidak *user* akan diminta kembali memasukkan *username* dan *password* yang sesuai.

### **Antarmuka Halaman Registrasi**



Gambar 3. 31 Halaman Registrasi Sistem

Gambar diatas merupakan implementasi dari halaman registrasi. Jika *user* belum memiliki akun untuk masuk sistem, maka *user* dapat melakukan registrasi dengan cara menginputkan *username* dan *password* lalu menekan tombol register.

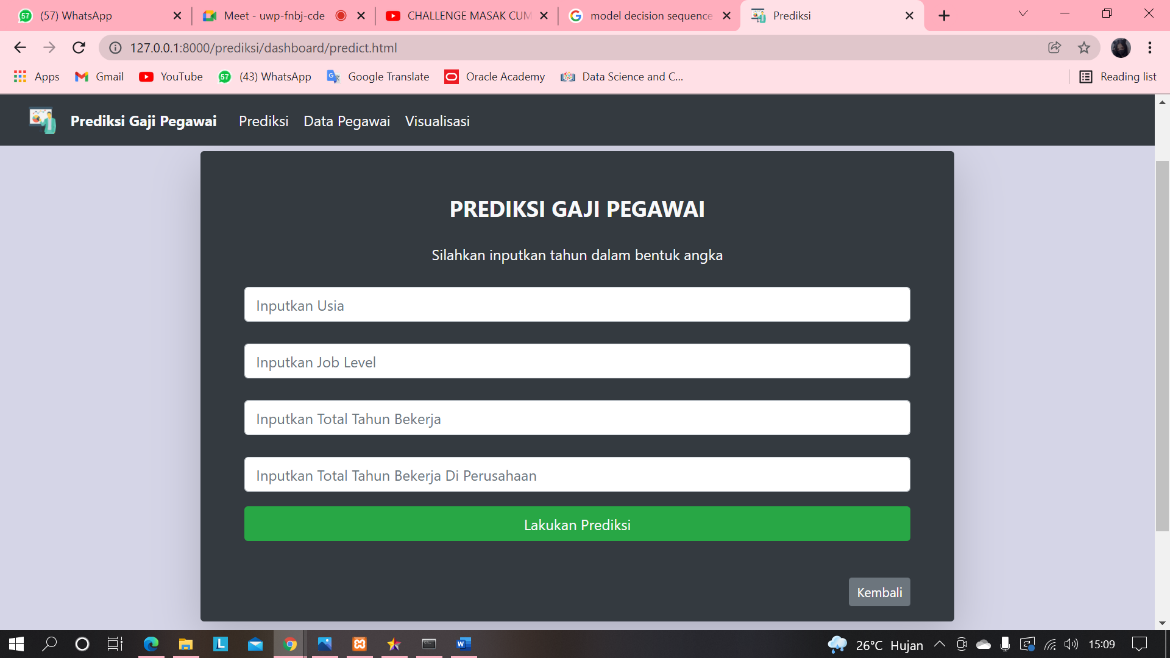
### **Antarmuka Halaman Dashboard**



Gambar 3. 32 Halaman Dashboard Sistem

Gambar diatas merupakan implementasi dari halaman *dashboard* dari sistem yang dirancang. Pada bagian navbar, terdapat pilihan submenu untuk mengakses suatu halaman yang lainnya. Pilihan submenu tersebut terdiri dari Prediksi, Data Pegawai, serta Visualisasi.

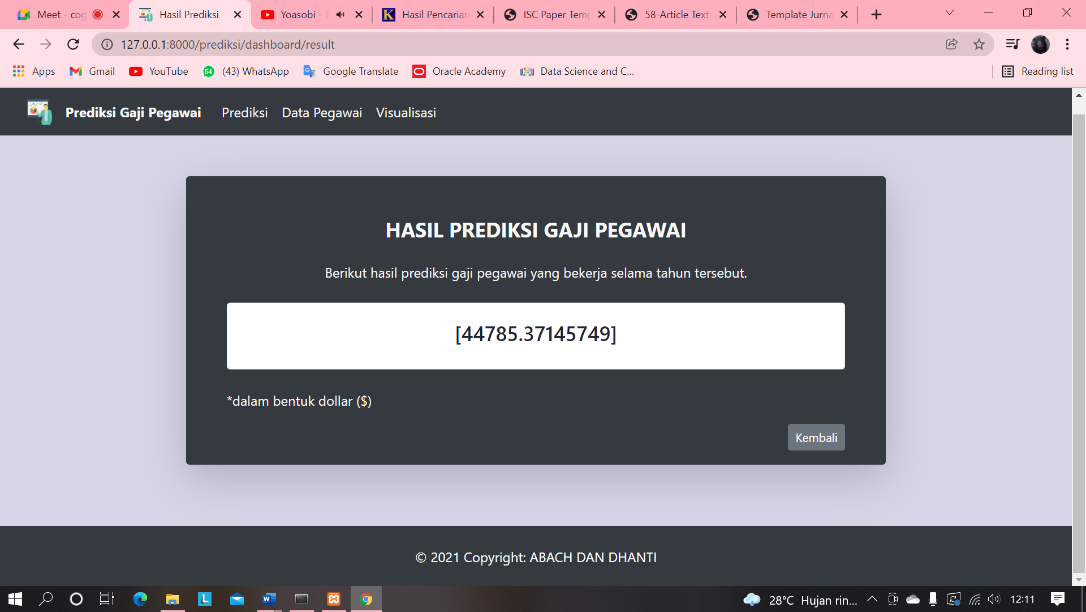
### **Antarmuka Halaman Prediksi**



Gambar 3. 33 Halaman Prediksi Sistem

Gambar diatas merupakan implementasi dari halaman prediksi dari sistem yang dirancang. Pada halaman ini, terdapat form untuk melakukan prediksi gaji pegawai. *User* dapat menginputkan angka berupa usia, job level, total tahun bekerja, dan total tahun bekerja di perusahan ke dalam form untuk mengetahui hasil prediksi gaji pegawai. Kemudian *user* dapat menekan button “Lakukan Prediksi” untuk melihat hasil prediksi berdasarkan tahun yang diinputkan.

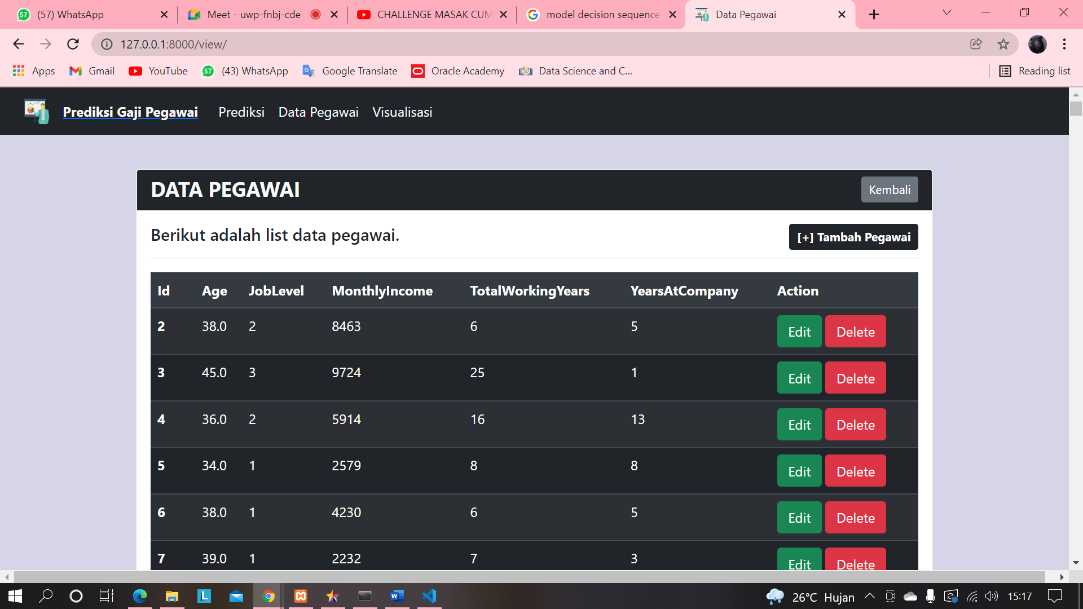
### **Antarmuka Halaman Hasil Prediksi**



Gambar 3. 34 Halaman Hasil Prediksi Sistem

Gambar diatas merupakan implementasi dari halaman hasil prediksi dari sistem yang dirancang. Pada halaman ini, akan ditampilkan hasil prediksi gaji dari user yang sebelumnya telah menginputkan angka tahun pada form prediksi.

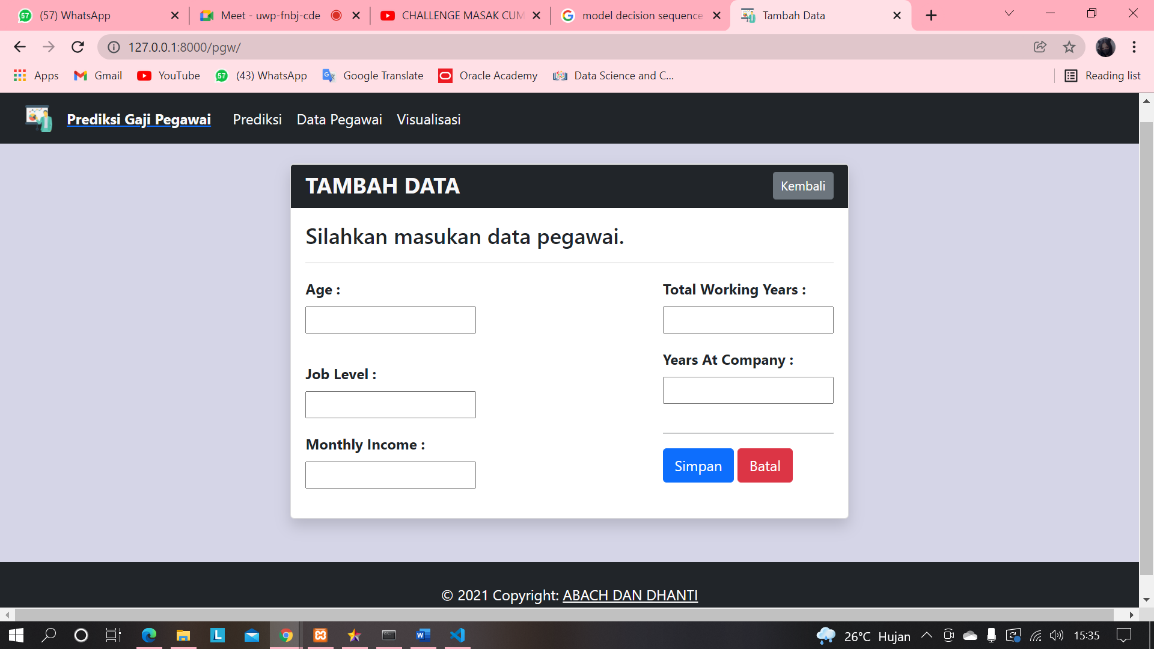
### **Antarmuka Halaman Data Pegawai**



Gambar 3. 35 Halaman Data Pegawai Sistem

Gambar diatas merupakan implementasi dari halaman data pegawai dari sistem yang dirancang. Pada halaman ini, akan ditampilkan data dalam bentuk tabel yang terdiri dari atribut Id, Age, JobLevel, MonthlyIncome, dan Action. *Record* yang ditampilkan terdiri dari 439 *record*. Terdapat *button* Tambah Pegawai yang berfungsi untuk menambah data pegawai, *button* Edit berfungsi untuk mengubah suatu *record* yang dipilih, dan *button* Delete untuk menghapus suatu *record* yang dipilih.

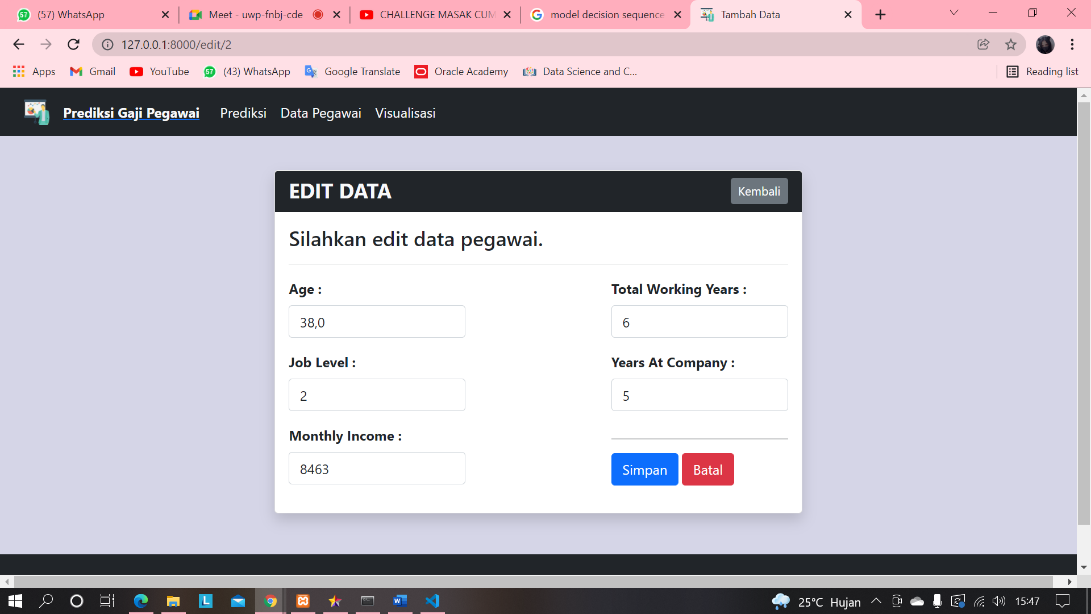
### **Antarmuka Halaman Tambah Data Pegawai**



Gambar 3. 36 Halaman Tambah Data Pegawai Sistem

Gambar diatas merupakan implementasi dari halaman tambah data pegawai dari sistem yang dirancang. Pada halaman ini, akan ditampilkan form tambah data untuk menambah data pegawai. *User* dapat menginputkan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company pada form. Lalu *user* dapat menekan *button* Simpan untuk menyimpan data yang diinputkan. *User* dapat menekan *button* Batal untuk me-*reset* data yang diinputkan pada form.

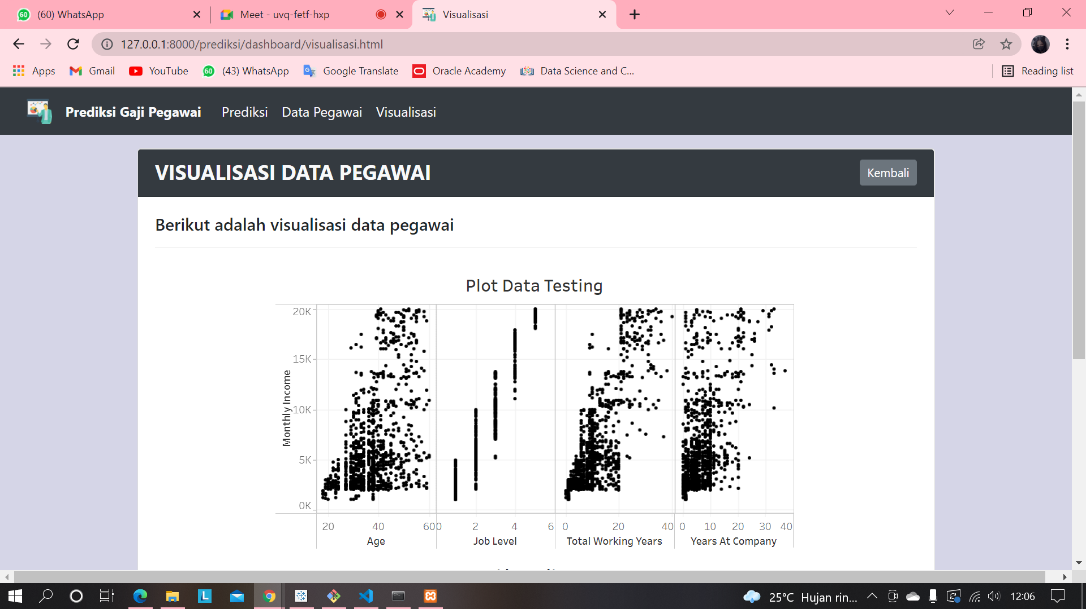
### **Antarmuka Halaman Edit Data Pegawai**



Gambar 3. 37 Halaman Edit Data Pegawai Sistem

Gambar diatas merupakan implementasi dari halaman edit data pegawai dari sistem yang dirancang. Pada halaman ini, akan ditampilkan form edit data untuk mengubah data pegawai. *User* dapat menginputkan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company pada form. Lalu *user* dapat menekan *button* Simpan untuk menyimpan data yang telah diubah. *User* dapat menekan *button* Batal untuk me-*reset* data yang diinputkan pada form.

### **Antarmuka Halaman Visualisasi**



Gambar 3. 38 Halaman Visualisasi Sistem

Gambar diatas merupakan implementasi dari halaman visualisasi dari sistem yang dirancang. Pada halaman ini, akan ditampilkan bentuk visualisasi prediksi antara Monthly Income dengan Age, Job Level, Total Working Years, Years At Company. Visualisasi terdiri dari Plot Data Testing, dan Linearitas

## **Analisis Kebutuhan**

### **Kebutuhan Fungsional (*Functional Requirements*)**

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan berbagai kebutuhan sistem yang akan dirancang. Dimana kebutuhan ini menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan fungsional yang akan dibuat yaitu terdiri dari 3 (tiga) proses sesuai dengan urutan sebagai berikut:

1. *Login* admin (user masuk ke aplikasi menggunakan username dan password);
2. Prediksi gaji pegawai yang dilakukan oleh Admin.
3. Melakukan *crate*, *read*, *update* dan *delete* pada data pegawai.

### **Kebutuhan Non-Fungsional (Non-Functional Requirements)**

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk sistem. Spesifikasi kebutuhan non-fungsional ini melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software*, serta analisis pengguna/*user*. Adapun kebutuhan non-fungsional yang didapatkan adalah sebagai berikut :

1. **Perangkat Keras (Hardware) Yang Digunakan**

Tabel 3. 9 Perangkat Keras Yang Digunakan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Perangkat** | **Spesifikasi** | **Keterangan** |
| 1 | *Hardisk* | 200 GB | Media untuk menyimpan data  aplikasi yang dibuat |
| 2 | *Memory* | 4 GB | *Memory System* yang digunakan |
| 3 | *Processor* | *AMD A4-9120 RADEON R3, 4 COMPUTE CORES 2C+2G*  *2.20 GHz* | Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan prosesor komputer |
| 4 | Infrastruktur jaringan | *PC* | Mengolah, menginput serta menghasilkan *output* data ataupun informasi yang sesuai dengan keinginan pengguna (*user*) |

1. **Perangkat Lunak (Software) Yang Digunakan**

Tabel 3. 10 Perangkat Lunak Yang Digunakan

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tools / Software** | **Fungsi** | **Keterangan** |
| 1. | *Windows 10* | Sistem Operasi | Sistem Operasi yang  digunakan |
| 2. | *XAMPP 3.2.4* | Web server | Membuka web server |
| 3. | *Python* | Bahasa Pemograman | Bahasa pemograman yang  digunakan |
| 4. | *Lucidchart* | *Software* Pendukung | Media dalam pembuatan  flowmap |
| 5. | *Visual Studio Code* | *Software* Pendukung | Media penulisan *coding* |
| 6. | *Jupyter Notebook* | *Software* Pendukung | Media penulisan *coding* |
| 7. | *PDF, Microsoft Office*  *Word* | *Document* | Media untuk membuat  laporan |
| 8. | *Google Crome* | *Browser* | Media untuk mencari  informasi |

1. Pengguna (User)

Aplikasi yang akan dirancang ini digunakan dalam lingkup bisnis sebuah perusahaan dibagian pengelolaan data gaji pegawai. Aplikasi ini melibatkan Admin sebagai pengelola data gaji pegawai pada perusahaan tersebut.

# BAB IV

# IMPLEMENTASI DAN PENGUJIAN

Setelah data dan sistem dianalisis dan didesain secara rinci, maka selanjutnya adalah tahapan implementasi. Implementasi dapat diartikan sebagai proses melaksanakan penerapan dari ide atau konsep yang telah disusun. Dengan begitu, implementasi tidak hanya sekedar melakukan penerapan suatu konsep, tetapi juga merupakan suatu rencana kegiatan yang telah tersusun untuk mencapai tujuan kegiatan. Pada tahap implementasi ini mencakup uji coba sistem dan penerapan antarmuka yang telah dirancang sebelumnya.

Perancangan sistem yang telah dibuat akan dilakukan penerapannya ke dalam bentuk web base. Dengan menggunakan *software multiplatform*, sistem dapat dirancang sesuai dengan perencanaan yang telah disusun. Perancangan ini meliputi penerapan antarmuka sistem ke dalam bentuk web base dan pengujian pada sistem yang telah dibuat. Pengujian sistem bertujuan untuk mengetahui tingkat keberhasilan berjalannya suatu sistem yang dibuat.

Tahap implementasi adalah tahapan yang penting dalam pembuatan sebuah sistem. Dengan adanya tahap ini, dapat diketahui suatu kendala dan seperti apa sistem yang telah dirancang tersebut dapat berjalan. Sehingga, jika ditemukannya suatu kendala pada sistem, sistem yang dibuat dapat diperbaiki kembali sesuai tujuan yang akan dicapai.

## **Implementasi Aplikasi**

Pada bagian implementasi aplikasi, dipaparkan kode-kode pada setiap proses berjalannya sistem yang telah dibangun. Pada bagian ini, pemaparan pola desain arsitektur dalam sistem. Arsitektur tersebut ialah Views dan Template serta URL.

### **Implementasi Proses Login**

1. Views

@login\_required

def dashboardView(request):

    return render(request, 'dashboard.html')

Dekoraktor login\_required berfungsi untuk masuk ke laman *login* pada *framework* Django. Fungsi dashboardView digunakan untuk menampilkan halaman dashboard.html yang ada di Template yaitu ke halaman login.

1. Template

<h1 class="h4 text-light mb-4">

    <b>SELAMAT DATANG !</b>

    </h1>

    <div class="sidebar-brand-text mx-3 text-light">

    Silahkan inputkan username dan password

       </div>

       <br/>

       </div>

       <form class="text-light" method="POST">

          {% csrf\_token %} {{form.as\_p}}

       <button>Login</button>

       <br/>

       <div align="center"class="sidebar-brand-text mx-3 text-light">

          Belum memiliki akun? Silahkan daftar.

       </div>

       <div align="center">

       <a href="http://127.0.0.1:8000/prediksi/register/"class="text info"> Mendaftar!</a>

       </div>

       <div class="float-right">

       <a href=<http://127.0.0.1:8000> class="btn btn-secondary btn-sm">Kembali</a>

       </div>

       </form>

Kode diatas merupakan tampilan dari halaman login. Terdapat form yang berisikan inputan username dan password. Selain itu tedapat link “Mendaftar!” untuk melakukan registrasi.

1. URL

 path('login/',LoginView.as\_view(),name="login\_url"),

Kode diatas merupakan URL untuk mengakses halaman login.

### **Implementasi Proses Registrasi**

1. Views

def registerView(request):

    if request.method == "POST":

        form = UserCreationForm(request.POST)

        if form.is\_valid():

            form.save()

            return redirect('login\_url')

    else:

        form = UserCreationForm()

    return render(request, 'registration/register.html', {'form': form})

Fungsi registerView digunakan untuk menampilkan halaman register.html yang ada di Template yaitu ke halaman registrasi. Metode yang digunakan adalah POST. Jika berhasil registrasi, maka akan dibalikan ke halaman login.

1. Template

  <b>SILAHKAN MEMBUAT AKUN !</b>

       </h1>

    <div class="sidebar-brand-text mx-3 text-light">

    Silahkan inputkan username dan password yang akan dibuat.

    </div>

    <br/>

      </div>

      <form method="POST" class="text-light">

        {% csrf\_token %} {{form.as\_p}}

      <button>Register</button>

      </form>

Kode diatas merupakan tampilan dari halaman register. Terdapat form yang berisikan inputan username dan password yang akan dibuat. Metode yang digunakan adalah POST.

1. URL

 path('register/', views.registerView, name="register\_url"),

Kode diatas merupakan URL untuk mengakses halaman registrasi.

### **Implementasi Proses Prediksi Gaji**

1. Views

def predict(request):

    context={'a':1}

    return render(request, 'predict.html', context)

Fungsi predict digunakan untuk menampilkan halaman predict.html yang ada di Template yaitu ke halaman prediksi.

1. Template

<h1 class="h4 text-light mb-4"><b>PREDIKSI GAJI PEGAWAI</b></h1>

<div class="sidebar-brand-text mx-3 text-light"> Silahkan inputkan tahun dalam bentuk angka </div><br>

</div>

{% block content %}

  <form class="user" method="POST" action="result">

  {% csrf\_token %}

  <div class="form-group">

 <input type="float" name="Age" id="Age" class="form-control"  placeholder="Inputkan Usia" autocomplete="off" required ><br>

  <input type="integer" name="JobLevel" id="JobLevel" class="form-control"  placeholder="Inputkan Job Level" autocomplete="off" required ><br>

  <input type="integer" name="YearsAtCompany" id="YearsAtCompany" class="form-control"  placeholder="Inputkan Total Tahun Bekerja" autocomplete="off" required ><br>

  <input type="integer" name="TotalWorkingYears" id="TotalWorkingYears" class="form-control"  placeholder="Inputkan Total Tahun Bekerja Di Perusahaan" autocomplete="off" required >

     </div>

     <div class="form-row">

     <div class="form-group col-md-12">

     <input type="submit" value="Lakukan Prediksi" class="btn btn-block bg-success text-light">

     </div>

     </div>

     {% endblock %}

                  </form>

                   {{ result }}

Kode diatas merupakan tampilan dari halaman prediksi gaji. Terdapat form yang berisikan inputan usia, job level, total tahun bekerja, dan total tahun bekerja di perusahan. Selain itu tedapat button “Lakukan Prediksi” untuk melakukan proses prediksi.

1. URL

url('predict', views.predict, name='predict')

Kode diatas merupakan URL untuk mengakses halaman prediksi gaji.

### **Implementasi Proses Hasil Prediksi Gaji**

1. Views

def result(request):

    print(request)

    tahun = float(request.POST.get('tahun'))

    model = pd.read\_pickle('./models/model3.pickle')

    result = model.predict([[tahun]])

    return render(request, 'result.html', {'result': result})

Fungsi result digunakan untuk menampilkan halaman result.html yang ada di Template yaitu ke halaman hasil prediksi.

1. Template

 <h1 class="h4 text-light mb-4"><b> HASIL PREDIKSI GAJI PEGAWAI</b></h1>

     <div class="sidebar-brand-text mx-3 text-light"> Berikut hasil prediksi gaji pegawai yang bekerja selama tahun tersebut.</div>

     <br>

     </div>

     <div class="card text-center">

         <div class="card-body">

             <h4>{{ result }}</h4>

         </div>

     </div>

     <br>

     <p class="text-light">\*dalam bentuk dollar ($)</p>

Kode diatas merupakan tampilan dari halaman hasil prediksi gaji. Terdapat form yang berisikan inputan username dan password. Selain itu tedapat hasil dari perhitungan prediksi gaji.

1. URL

url('result', views.result, name='result')

Kode diatas merupakan URL untuk mengakses halaman hasil prediksi.

### **mplementasi Proses Menampilkan Data Pegawai**

1. Views

def view(request):

    pegawai = Pegawai.objects.all()

    return render(request, "view.html", {'pegawai': pegawai})

Fungsi view digunakan untuk menampilkan halaman view.html yang ada di Template yaitu ke halaman data pegawai.

1. Template

<h5>Berikut adalah list data pegawai. </h5>

</div>

    <div class="float-right">

      <a href="http://127.0.0.1:8000/pgw" class="btn btn-dark btn-sm"><b>[+] Tambah Pegawai</b></a>

    </div>

    <br>

    <hr>

<table class="table table-dark table-striped">

  <thead class="thead-dark">

    <tr>

      <th scope="col">Id</th>

      <th scope="col">Age</th>

      <th scope="col">JobLevel</th>

      <th scope="col">MonthlyIncome</th>

      <th scope="col">TotalWorkingYears</th>

      <th scope="col">YearsAtCompany</th>

      <th scope="col">Action</th>

    </tr>

  </thead>

  {% for pegawai in pegawai %}

    <tr>

      <th scope="row">{{ pegawai.id }}</th>

      <td>{{ pegawai.Age }}</td>

      <td>{{ pegawai.JobLevel }}</td>

      <td>{{ pegawai.MonthlyIncome }}</td>

      <td>{{ pegawai.TotalWorkingYears }}</td>

      <td>{{ pegawai.YearsAtCompany }}</td>

      <td>

          <a href="/edit/{{ pegawai.id }}"><span class="btn btn-success">Edit</span></a>

          <a href="/delete/{{ pegawai.id }}"><span class="btn btn-danger">Delete</span></a>

      </td>

    </tr>

  {% endfor %}

  </tbody>

</table>

Kode diatas merupakan tampilan dari halaman hasil prediksi gaji. Terdapat form yang berisikan inputan username dan password. Selain itu tedapat hasil dari perhitungan prediksi gaji.

1. URL

path('view/', views.view),

Kode diatas merupakan URL untuk mengakses halaman data pegawai.

### **Implementasi Proses Tambah Data Pegawai**

1. Views

def pgw(request):

    if request.method == "POST":

        form = PegawaiForm(request.POST)

        if form.is\_valid():

            try:

                form.save()

                return redirect('/view')

            except:

                pass

    else:

        form = PegawaiForm()

    return render(request, 'haltambah.html', {'form': form})

Fungsi pgw digunakan untuk menampilkan halaman haltambah.html yang ada di Template yaitu ke halaman tambah data pegawai.

1. Template

<h4>Silahkan masukan data pegawai.</h4>

<hr>

   <form action="/pgw/" id="form-tambah" method="POST">

   {% csrf\_token %}

   <div class="float-left">

     <div class="form-row">

      <div class="form-group col-md-6">

     <label for=" "><strong>Age :</strong></label>

       <div>{{ form.Age }}</div>

       </div>

      </div>

     <div class="form-row">

      <div class="form-group col-md-6">

       <label for=" "><strong>Job Level : </strong></label>

       <div>{{ form.JobLevel }}</div>

      </div>

     </div>

     <div class="form-row">

      <div class="form-group col-md-13">

       <label for=" "><strong>Monthly Income : </strong></label>

       <div>{{ form.MonthlyIncome }}</div>

        </div>

      </div>

      </div>

       <div class="float-right">

      <div class="form-row">

     <div class="form-group col-md-12">

     <label for=" "><strong>Total Working Years : </strong></label>

      <div>{{ form.TotalWorkingYears }}</div>

      </div>

      </div>

       <div class="form-row">

       <div class="form-group col-md-12">

      <label for=" "><strong>Years At Company : </strong></label>

       <div>{{ form.YearsAtCompany }}</div>

        </div>

         </div>

         <hr>

         <div class="form-group">

         <button type="submit" class="btn btn-primary">Simpan</button>

         <button type="reset" class="btn btn-danger">Batal</button>

        </div>

       </div>

     </form>

Kode diatas merupakan tampilan dari halaman tambah data pegawai. Terdapat form yang berisikan inputan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company. Selain itu tedapat button simpan untuk proses menyimpan data dan button batal untuk mereset data.

1. URL

 path('pgw/', views.pgw),

Kode diatas merupakan URL untuk mengakses halaman tambah pegawai.

### **Implementasi Proses Edit Data Pegawai**

1. Views

def edit(request, id):

    pegawai = Pegawai.objects.get(id=id)

    return render(request, 'edit.html', {'pegawai': pegawai})

def update(request, id):

    pegawai = Pegawai.objects.get(id=id)

    form = PegawaiForm(instance=pegawai)

    if request.method == 'POST':

        form = PegawaiForm(request.POST, instance=pegawai)

        if form.is\_valid():

            form.save()

            return redirect('/view')

    return render(request, 'view.html', {'form': form})

Fungsi edit digunakan untuk menampilkan halaman edit.html yang ada di Template yaitu ke halaman edit data pegawai. Fungsi update digunakan untuk melakukan fungsi edit. Metode yang digunakan adalah POST. Jika fungsi tersebut berhasil dilakukan maka hasil update akan dialihkan ke halaman data pegawai.

1. Template

<h4>Silahkan edit data pegawai.</h4>

<hr>

<form action="/update/{{pegawai.id}}" class="post-form" method="POST">

  <input type="hidden" name="id" id="id" required maxlength="20" value="{{ pegawai.id }}"/>

   {% csrf\_token %}

   <div class="float-left">

    <div class="form-row">

    <div class="form-group col-md-15">

   <label for="kode\_produk"><strong>Age :</strong></label>

    <div>

   <input type="number" class="form-control" name="Age" value="{{ pegawai.Age }}">

   </div>

    </div>

   </div>

   <div class="form-row">

    <div class="form-group col-md-15">

   <label for="jenis\_produk"><strong>Job Level : </strong></label>

    <div>

   <input  type="number" class="form-control"  name="JobLevel" value="{{ pegawai.JobLevel }}">

   </div>

    </div>

     </div>

     <div class="form-row">

     <div class="form-group col-md-15">

   <label for="jenis\_produk"><strong>Monthly Income : </strong></label>

   <div>

   <input  type="number" class="form-control"  name="MonthlyIncome" value="{{ pegawai.MonthlyIncome }}">

   </div>

   </div>

     </div>

     </div>

     <div class="float-right">

     <div class="form-row">

     <div class="form-group col-md-15">

    <label for="jenis\_produk"><strong>Total Working Years : </strong></label>

     <div>

     <input  type="number" class="form-control"  name="TotalWorkingYears" value="{{ pegawai.TotalWorkingYears }}">

     </div>

     </div>

     </div>

     <div class="form-row">

    <div class="form-group col-md-15">

   <label for="jenis\_produk"><strong>Years At Company : </strong></label>

     <div>

     <input  type="number" class="form-control"  name="YearsAtCompany" value="{{ pegawai.YearsAtCompany }}">

      </div>

      </div>

     </div>

     <hr>

    <div class="form-group">

   <button type="submit" class="btn btn-primary">Simpan</button>

   <button type="reset" class="btn btn-danger">Batal</button>

     </div>

    </div>

   </form>

Kode diatas merupakan tampilan dari halaman edit data pegawai. Terdapat form yang berisikan inputan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company. Selain itu tedapat button simpan untuk proses menyimpan data yang telah diedit dan button batal untuk mereset data.

1. URL

path('edit/<int:id>', views.edit),

path('update/<int:id>', views.update),

Kode diatas merupakan URL untuk mengakses halaman edit pegawai, dan update untuk proses update.

### **Implementasi Proses Hapus Data Pegawai**

1. Views

def delete(request, id):

    pegawai = Pegawai.objects.get(id=id)

    pegawai.delete()

    return redirect("/view")

Fungsi delete digunakan untuk melakukan fungsi hapus data. Jika data berhasil terhapus maka hasil data yang terhapus akan dialihkan ke halaman data pegawai.

1. URL

path('delete/<int:id>', views.delete),

Kode diatas merupakan URL untuk melakukan proses hapus data.

### **Implementasi Proses Menampilkan Visualisasi**

1. Views

def visualisasi(request):

    return render(request, 'visualisasi.html')

Fungsi visualisasi digunakan untuk menampilkan halaman visualisasi.html yang ada di Template yaitu ke halaman visualisasi.

1. Template

<div class="float-left">

  <h5>Berikut adalah visualisasi data pegawai </h5>

</div>

  <br>

  <hr>

  <div class='tableauPlaceholder' id='viz1640335357904' style='position: relative'>

    <object class='tableauViz'  style='display:none;'>

    <param name='host\_url' value='https%3A%2F%2Fpublic.tableau.com%2F'/>

    <param name='embed\_code\_version' value='3' />

    <param name='site\_root' value='' />

    <param name='name' value='SalaryVSYearsExperience4&#47;Dashboard8'/>

    <param name='tabs' value='no' />

    <param name='toolbar' value='yes' />

    <param name='animate\_transition' value='yes' />

    <param name='display\_static\_image' value='yes' />

    <param name='display\_spinner' value='yes' />

    <param name='display\_overlay' value='yes' />

    <param name='display\_count' value='yes' />

    <param name='language' value='en-US' />

    <param name='filter' value='publish=yes' />

   </object>

</div>

  <script type='text/javascript'>

  var divElement = document.getElementById('viz1640335357904');

  var vizElement = divElement.getElementsByTagName('object')[0];

  if ( divElement.offsetWidth > 800 )

  {

    vizElement.style.minWidth='600px';

    vizElement.style.maxWidth='900px';

    vizElement.style.width='100%';

    vizElement.style.minHeight='127px';

    vizElement.style.maxHeight='527px';

    vizElement.style.height=(divElement.offsetWidth\*0.75)+'px';

    }

    else if ( divElement.offsetWidth > 500 )

    {

      vizElement.style.minWidth='600px';

      vizElement.style.maxWidth='900px';

      vizElement.style.width='100%';

      vizElement.style.minHeight='127px';

      vizElement.style.maxHeight='527px';

      vizElement.style.height=(divElement.offsetWidth\*0.75)+'px';

      }

      else

      {

        vizElement.style.width='100%';

        vizElement.style.height='727px';

        }

      var scriptElement = document.createElement('script');

        scriptElement.src = 'https://public.tableau.com/javascripts/api/viz\_v1.js';

        vizElement.parentNode.insertBefore(scriptElement, vizElement);

        </script>

      </div>

Kode diatas merupakan tampilan dari halaman visualisasi. Visualisasi tersebut dibuat dari platform Tableau yang telah dirancang sebelumnya.

1. URL

 url('visualisasi', views.visualisasi, name='visualisasi'),

Kode diatas merupakan URL untuk mengakses halaman visualisasi.

## **Implementasi Model *Machine Learning***

Pada bagian implementasi model *machine learning*, dipaparkan kode-kode dalam proses analisis data pegawai berdasarkan pengalaman lama bekerja. Selain analisis data, dilakukan juga pembuatan model prediksi gaji pegawai.

### **Himpunan Data**

Pada tahapan ini kita perlu untuk memahami dan mempersiapkan data yang dikenal dengan istilah *Data Preprocessing*. Metode yang digunakan dalam *Data Preprocessing* pada model ini adalah *Data Cleaning*. Berikut kode programnya :

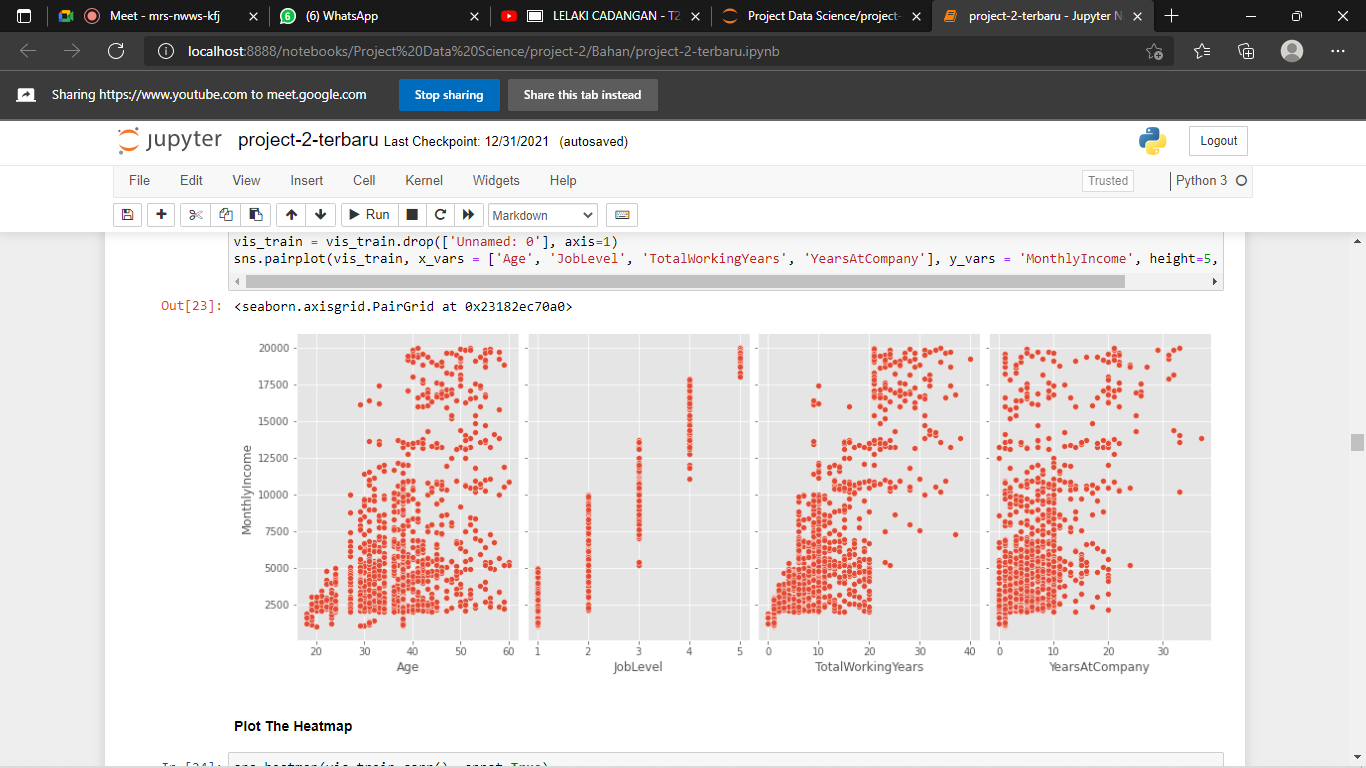
|  |  |
| --- | --- |
| [1] | # Basic Library  import pandas as pd  import numpy as np  # Data Visualization  import matplotlib.pyplot as plt  from mpl\_toolkits.mplot3d import Axes3D  import seaborn as sns  from scipy.stats import skew  # Model Building  from sklearn.linear\_model import LinearRegression  import statsmodels.api as sm |
| [2] | df\_train = pd.read\_csv('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_train.csv')  df\_train |
| [3] | df\_train.info() |
| [4] | # Encoder BusinessTravel Variable  # converting type of columns to 'category'  df\_train['BusinessTravel']= df\_train['BusinessTravel'].astype('category')  # Assigning numerical values and storing in another column  df\_train['BusinessTravel']= df\_train['BusinessTravel'].cat.codes  # Encoder Department Variable  df\_train['Department']= df\_train['Department'].astype('category')  # Assigning numerical values and storing in another column  df\_train['Department'] = df\_train['Department'].cat.codes  # Encoder EducationField Variable  df\_train['EducationField']= df\_train['EducationField'].astype('category')  # Assigning numerical values and storing in another column  df\_train['EducationField']= df\_train['EducationField'].cat.codes  # Encoder Gender Variable  df\_train['Gender'] = df\_train['Gender'].astype('category')  # Assigning numerical values and storing in another column  df\_train['Gender'] = df\_train['Gender'].cat.codes  # Encoder JobRole Variable  df\_train['JobRole'] = df\_train['JobRole'].astype('category')  # Assigning numerical values and storing in another column  df\_train['JobRole'] = df\_train['JobRole'].cat.codes  # Encoder MaritalStatus Variable  df\_train['MaritalStatus']= df\_train['MaritalStatus'].astype('category')  # Assigning numerical values and storing in another column  df\_train['MaritalStatus'] = df\_train['MaritalStatus'].cat.codes  # Encoder Over18 Variable  df\_train['Over18'] = df\_train['Over18'].astype('category')  # Assigning numerical values and storing in another column  df\_train['Over18'] = df\_train['Over18'].cat.codes  # Encoder OverTime Variable  df\_train['OverTime'] = df\_train['OverTime'].astype('category')  # Assigning numerical values and storing in another column  df\_train['OverTime'] = df\_train['OverTime'].cat.codes  df\_train |
| [5] | df\_train.isnull().values.any() |
| [6] | df\_train.isnull().sum() |
| [7] | Age = df\_train['Age']  Age.describe() |
| [8] | df\_train.Age = df\_train.Age.fillna(value=df\_train.Age.mean()) |
| [9] | DailyRate = df\_train['DailyRate']  DailyRate.describe() |
| [10] | df\_train.DailyRate= df\_train.DailyRate.fillna  (value=df\_train.DailyRate.mean()) |
| [11] | DistanceFromHome = df\_train['DistanceFromHome']  DistanceFromHome.describe() |
| [12] | df\_train.DistanceFromHome= df\_train.DistanceFromHome.fillna  (value=df\_train.DistanceFromHome.mean()) |
| [13] | df\_train\_clean = df\_train  df\_train\_clean.isnull().values.any() |
| [14] | df\_train\_clean.isnull().sum() |
| [15] | def plotCorrelationMatrix(df\_train\_clean, graphWidth):  df\_train\_clean = df\_train\_clean[[col for col in df\_train\_clean if df\_train\_clean[col].nunique() > 1]] # keep columns where there are more than 1 unique values  if df\_train\_clean.shape[1] < 2:  print(f'No correlation plots shown: The number of non-NaN or constant columns ({df\_train\_clean.shape[1]}) is less than 2')  return  corr = df\_train\_clean.corr()  plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80, facecolor='w', edgecolor='k')  corrMat = plt.matshow(corr, fignum = 1)  plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)  plt.yticks(range(len(corr.columns)), corr.columns)  plt.gca().xaxis.tick\_bottom()  plt.colorbar(corrMat)  plt.title(f'Correlation Matrix for Data Training', fontsize=15)  plt.show()  plotCorrelationMatrix(df\_train\_clean, 8) |
| [16] | df\_train\_clean.corr().abs() |
| [17] | df\_train\_clean.columns |
| [18] | df\_train\_clean = df\_train\_clean.drop(['Attrition', 'BusinessTravel', 'DailyRate', 'Department','DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount','EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement', 'JobRole', 'JobSatisfaction', 'MaritalStatus','MonthlyRate', 'NumCompaniesWorked','Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TrainingTimesLastYear', 'WorkLifeBalance','YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager'], axis=1) |
| [19] | df\_train\_clean.corr() |
| [20] | x\_train = df\_train\_clean[['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany']]  y\_train = df\_train\_clean[['MonthlyIncome']] |
| [21] | df\_train\_clean.to\_csv('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_train\_clean.csv') |

Keterangan :

* Line 1 : Untuk mengimport library yang dibutuhkan.
* Line 2 : Mengimport data ke pyhton kemudian disimpan dalam variabel dengan nama df\_train.Line 3 : Untuk mengimport library pymysql.
* Line 3 : Menampilkan info detail tabel/data yang disimpan.
* Line 4 : Melakukan encoder, karena machine learning tidak bisa membaca korelasi dari suatu variabel jika nilainya bernilai object/string.
* Line 5 : Cek apakah ada data yang kosong pada tabel..
* Line 6 : Cek berapa record yang terdapat data yag kosong.
* Line 7 : Membuat variabel Age yang berisikan atribut age, kemudian cek deskripsi variabel Age.
* Line 8 : Melakukan pengisian data kosong pada variabel Age dengan nilai mean-nya.
* Line 9 : Membuat variabel DailyRate yang berisikan atribut DailyRate, kemudian cek deskripsi variabel DailyRate.
* Line 10 : Melakukan pengisian data kosong pada variabel DailyRate dengan nilai mean-nya.
* Line 11 : Membuat variabel DistanceFromHome yang berisikan atribut DistanceFromHome, kemudian cek deskripsi variabel DistanceFromHome.
* Line 12 : Melakukan pengisian data kosong pada variabel DistanceFromHome dengan nilai mean-nya.
* Line 13 : Membuat variabel df\_train\_clean yang berisikan data df\_train, kemudian cek apakah masih terdapat data yang kosong atau tidak.
* Line 14 : Cek berapa record yang terdapat data yag kosong.
* Line 15 : Membuat heatmap antar variabel yang terdapat dalam variabel df\_train\_clean.
* Line 16 : Membuat tabel korelasi antar variabel.
* Line 17 : Menampilkan kumpulan variabel yang terdapat dalam df\_train\_clean.
* Line 18 : Melakukan drop variabel yang tidak diperlukan.
* Line 19 : Membuat tabel korelasi setelah proses drop variabel.
* Line 20 : Menetapkan variabel independent (sumbu x) yaitu variabel Age, JobLevel, TotalWorkingYears, dan YearsAtCompany. Kemudian menetapkan variabel variabel dependen (sumbu y) yaitu MonthlyIncome.
* Line 21 : Melakukan export data ke format file csv.

1. Visualisasi Data Training

|  |  |
| --- | --- |
| [22] | %matplotlib inline  plt.style.use('ggplot')  plt.rcParams['figure.figsize'] = (12,8) |
| [23] | vis\_train = pd.read\_csv('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_train\_clean.csv')  vis\_train = vis\_train.drop(['Unnamed: 0'], axis=1)  sns.pairplot(vis\_train, x\_vars = ['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany'], y\_vars = 'MonthlyIncome', height=5, aspect=0.7) |
| [24] | sns.heatmap(vis\_train.corr(), annot=True) |



Gambar 4. 1 Visualisasi Data Training

Keterangan :

* Line 22 : Membuat style untuk diagram plot.
* Line 23 : Melakukan import data kemudian disimpan dalam variabel vis\_train. Drop kolom yang tidak diperlukan. Kemudian menentukan variabel x dan y untuk diagram plot.
* Line 24 : Membuat heatmap untuk melihat korelasi antar variabel.

### **Proses Data Mining & Pengetahuan**

Pada tahapan Proses Data Mining kita perlu memilih metode yang sesuai dengan karakter data yang dikenal dengan istilah Modelling. Pada model ini digunakan Proses Data Mining Prediction. Pada tahapan Pengetahuan kita perlu memahami model dan pengetahuan yang sesuai sehingga dapat memilih model. Model yang digunakan adalah Linear Regression menggunakan Scikit Learn.

|  |  |
| --- | --- |
| [25] | df\_test = pd.read\_csv('E:\Data\_Urang\Mata Kuliah\Semester3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_test.csv')  df\_test |
| [26] | # Encoder BusinessTravel Variable  # converting type of columns to 'category'  df\_test['BusinessTravel'] = df\_test['BusinessTravel'].astype('category')  # Assigning numerical values and storing in another column  df\_test['BusinessTravel'] = df\_test['BusinessTravel'].cat.codes  # Encoder Department Variable  df\_test['Department'] = df\_test['Department'].astype('category')  # Assigning numerical values and storing in another column  df\_test['Department'] = df\_test['Department'].cat.codes  # Encoder EducationField Variable  df\_test['EducationField'] = df\_test['EducationField'].astype('category')  # Assigning numerical values and storing in another column  df\_test['EducationField'] = df\_test['EducationField'].cat.codes  # Encoder Gender Variable  df\_test['Gender'] = df\_test['Gender'].astype('category')  # Assigning numerical values and storing in another column  df\_test['Gender'] = df\_test['Gender'].cat.codes  # Encoder JobRole Variable  df\_test['JobRole'] = df\_test['JobRole'].astype('category')  # Assigning numerical values and storing in another column  df\_test['JobRole'] = df\_test['JobRole'].cat.codes  # Encoder MaritalStatus Variable  df\_test['MaritalStatus'] = df\_test['MaritalStatus'].astype('category')  # Assigning numerical values and storing in another column  df\_test['MaritalStatus'] = df\_test['MaritalStatus'].cat.codes  # Encoder Over18 Variable  df\_test['Over18'] = df\_test['Over18'].astype('category')  # Assigning numerical values and storing in another column  df\_test['Over18'] = df\_test['Over18'].cat.codes  # Encoder OverTime Variable  df\_test['OverTime'] = df\_test['OverTime'].astype('category')  # Assigning numerical values and storing in another column  df\_test['OverTime'] = df\_test['OverTime'].cat.codes  df\_test |
| [27] | df\_test.isnull().values.any() |
| [28] | df\_test.isnull().sum() |
| [29] | # Missing Value in Age  Age = df\_test['Age']  df\_test.Age = df\_test.Age.fillna(value=df\_test.Age.mean())  # Missing Value in DailyDate  DailyRate = df\_test['DailyRate']  df\_test.DailyRate = df\_test.DailyRate.fillna(value=df\_test.DailyRate.mean())  # Missing Value in DistanceFromHome  DistanceFromHome = df\_test['DistanceFromHome']  df\_test.DistanceFromHome = df\_test.DistanceFromHome.fillna(value=df\_test.DistanceFromHome.mean())  # Missing Value in BusinessTravel  BusinessTravel = df\_test['BusinessTravel']  df\_test.BusinessTravel = df\_test.BusinessTravel.fillna(value=df\_test.BusinessTravel.mean())  # Missing Value in MartialStatus  MaritalStatus = df\_test['MaritalStatus']  df\_test.MaritalStatus = df\_test.MaritalStatus.fillna(value=df\_test.MaritalStatus.mean()) |
| [30] | df\_test.isnull().values.any() |
| [31] | df\_test.isnull().sum() |
| [32] | # Eliminasi Variabel yang Tidak akan digunakan  df\_test\_clean = df\_test.drop(['BusinessTravel', 'DailyRate', 'Department','DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount','EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate','JobInvolvement','JobRole','JobSatisfaction','MaritalStatus','MonthlyRate','NumCompaniesWorked','Over18','OverTime','PercentSalaryHike','PerformanceRating','RelationshipSatisfaction','StandardHours','StockOptionLevel','TrainingTimesLastYear','WorkLifeBalance','YearsInCurrentRole','YearsSinceLastPromotion','YearsWithCurrManager'], axis=1) |
| [33] | df\_test\_clean.corr().abs() |
| [34] | x\_test = df\_test\_clean[['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany']]  y\_test = df\_test\_clean[['MonthlyIncome']] |
| [35] | df\_test\_clean.to\_csv('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_test\_clean.csv') |
| [36] | vis\_test = pd.read\_csv('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_test\_clean.csv')  vis\_test = vis\_test.drop(['Unnamed: 0'], axis=1)  sns.pairplot(vis\_test, x\_vars = ['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany'], y\_vars = 'MonthlyIncome', height=5, aspect=0.7) |
| [37] | sns.heatmap(vis\_test.corr(), annot=True) |
| [38] | regressor = LinearRegression()  persamaan = regressor.fit(x\_train, y\_train)  print(regressor.coef\_)  print(regressor.intercept\_) |
| [39] | y\_pred = regressor.predict(x\_test)  print(y\_pred) |
| [40] | vis\_test['MonthlyIncome Prediction'] = y\_pred.tolist()  vis\_test |
| [41] | vis\_test.to\_excel('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_vis\_test.xlsx') |

Keterangan :

* Line 25 : Melakukan import data testing kemudian disimpan dalam variabel df\_test.
* Line 26 : Melakukan encoder, karena machine learning tidak bisa membaca korelasi dari suatu variabel jika nilainya bernilai object/string. Line 27 : Cek apakah masih terdapat data yang kosong atau tidak.
* Line 28 : Cek apakah ada data yang kosong pada tabel.
* Line 29 : Membuat variabel kemudian melakukan pengisian data kosong pada variabel dengan nilai mean-nya.
* Line 30 : Cek apakah masih terdapat data yang kosong atau tidak.
* Line 31 : Cek apakah ada data yang kosong pada tabel.
* Line 32 : Melakukan drop variabel yang tidak diperlukan.
* Line 33 : Membuat tabel korelasi setelah proses drop variabel.
* Line 34 : Menetapkan variabel independent (sumbu x) yaitu variabel Age, JobLevel, TotalWorkingYears, dan YearsAtCompany. Kemudian menetapkan variabel variabel dependen (sumbu y) yaitu MonthlyIncome.
* Line 35 : Melakukan export data ke format file csv.
* Line 36 : Melakukan import data kemudian disimpan dalam variabel vis\_train. Drop kolom yang tidak diperlukan. Kemudian menentukan variabel x dan y untuk diagram plot.
* Line 37 : membuat heatmap untuk melihat korelasi antar variabel.
* Line 38 : Membuat variabel regressor yang isinya metode LinearRegression, kemudian membuat variabel persamaan yang isinya terdapat method regressor.fit dengan parameternya x\_train dan y\_train.

Dari model tersebut didapat nilai koefisien dari variabel Independen. Nilai koefisien dari Age adalah -5,054 yang berarti tiap karyawan yang mengalami pertambahan satu tahun umur kerja, maka akan mengalami penurunan gaji sebesar 5,054. Nilai koefisien dari JobLevel adalah 3871,7530 yang berarti tiap karyawan yang mengalami pertambahan satu tingkat job level akan mengalami kenaikan gaji sebesar 3871,7530. Nilai koefisien TotalWorkingYears adalah 46,9405 yang berarti tiap karyawan yang mengalami pertambahans satu tahun pengalaman bekerja akan mengalami kenaikan kerja sebesar 46,9405. Nilai koefisien YearsAtCompany adalah -9,8460 yang berarti tiap karyawan yang mengalami pertambahan satu tahun akan mengalami penurunan gaji sebesar 9,8460.

* Line 39 : Mencari konstanta/intercept menggunakan regressor, kemudian ditampilkan.

Intecept merupakan sebuah koefisien dari sebuah persamaan model regresi linear. Untuk kasus ini, berarti untuk variabel X yang bernilai nol atau karyawan yang belum berpengalaman kerja, karyawan tersebut akan menerima gaji sebesar 26611 per tahunnya.

* Line 40 : Melakukan prediksi data testing menggunkana model machine learning. Kemudian buat kolom baru yang Bernama MonthlyIncome Prediction yang berisikan nilai prediksi.

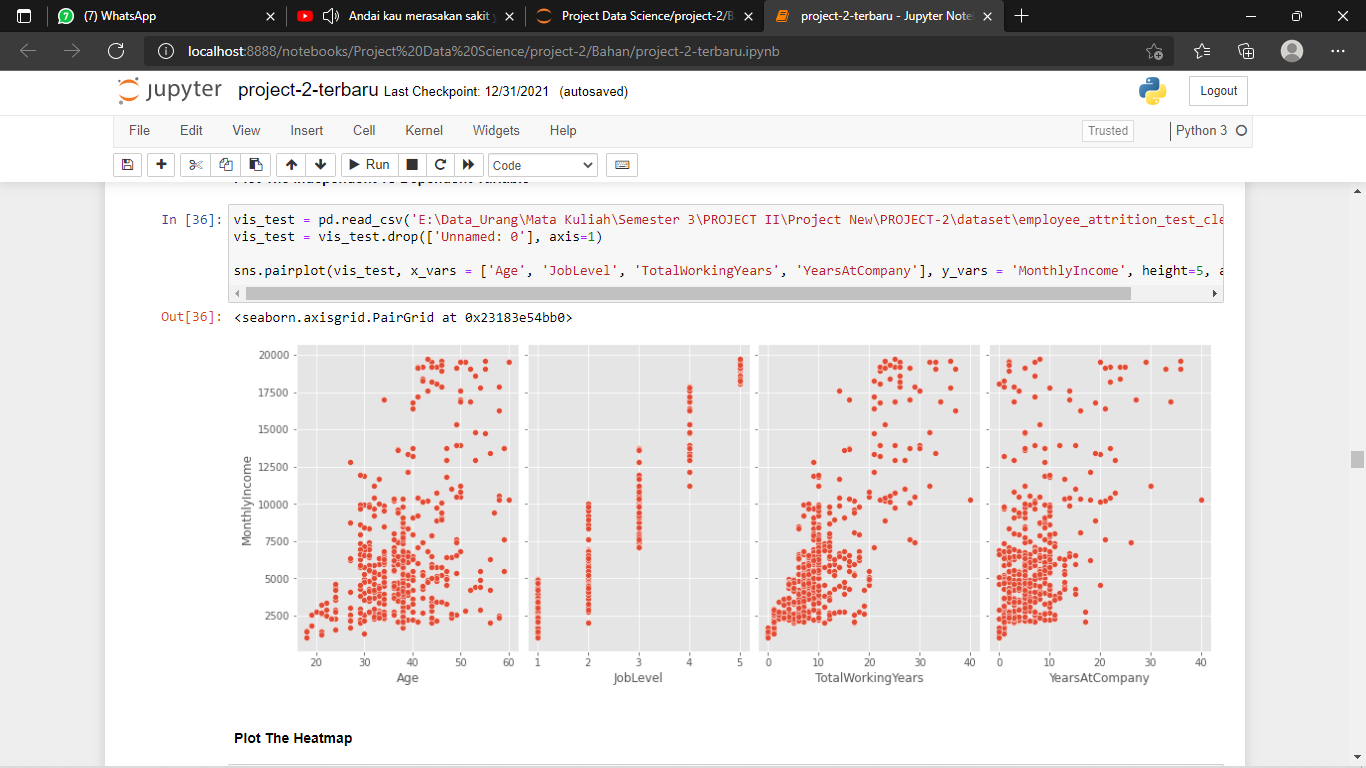
Berdasarkan nilai koefisien Years Experience dan Intersept didapat persamaan regresi linear berikut :

**MonthlyIncome = -1728 – 5,054(Age) + 3871,7530(JobLevel) + 46,9405(TotalWorkingYears) – 9,8460(YearsAtCompany)**

* Line 41 : Melakukan export data setelah prediksi ke dalam format file excel.

1. Visualisasi Data Testing

|  |  |
| --- | --- |
| [36] | vis\_test = pd.read\_csv('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_attrition\_test\_clean.csv')  vis\_test = vis\_test.drop(['Unnamed: 0'], axis=1)  sns.pairplot(vis\_test, x\_vars = ['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany'], y\_vars = 'MonthlyIncome', height=5, aspect=0.7) |
| [37] | sns.heatmap(vis\_test.corr(), annot=True) |

Keterangan :

Gambar 4. 2 Visualisasi Data Testing

* Line 36 : Melakukan import data kemudian disimpan dalam variabel vis\_train. Drop kolom yang tidak diperlukan. Kemudian menentukan variabel x dan y untuk diagram plot.
* Line 37 : membuat heatmap untuk melihat korelasi antar variabel.

### **Evaluasi Data**

Pada tahap ini kita melakukan evaluasi terhadap analisis model dan kinerja metode.

1. R-Square

R2-Square adalah sebuah nilai yang menyatakan seberapa sesuai hasil prediksi model mendekati data yang sebenarnya. Semakin besar r2\_square, maka hasil prediksi semakin dekat dengan data yang sebenarnya, artinya sama saja dengan semakin mendekati 1, maka semakin bagus model tersebut.

|  |  |
| --- | --- |
| [42] | from sklearn.metrics import r2\_score  r2 = r2\_score (y\_test, y\_pred)  print ("Nilai R2 adalah ", r2) |

Keterangan :

* Line 42 : Dari scikit learn metrics diimport metode r2\_score. Menghitung r2\_score dengan parameternya x\_train dan y\_train, kemudian hasilnya disimpan pada variable r2. Menampilkan nilai pada variabel r2.

1. Validasi Model

|  |  |
| --- | --- |
| [43] | X = df\_train\_clean[['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany']]  X = sm.add\_constant(X) # adding a constant  olsmod = sm.OLS(df\_train['MonthlyIncome'], X).fit()  print(olsmod.summary()) |

Keterangan :

* Line 43 : Membuat variabel x yang berisikan atribut Age, JobLevel, TotalWorkingYears dan YearsAtCompany. Variabel X digunakan utnuk melihat validasi model machine learning dengan model OLS.

1. Uji F (ANOVA)

F-test atau ANOVA (Analysis of Variance) dalam regresi multi-linear dapat digunakan untuk menentukan apakah model kompleks kita berkinerja lebih baik daripada model yang lebih sederhana (misalnya model dengan hanya satu variabel independent). Dengan uij-F kita dapat mengevaluasi signifikansi model kita dengan menghitung probabilitas pengamatan statistic-F yang setidaknya setinggi nilai yang diperoleh model kita. Mirip dengan skor R2, kita bisa dengan mudah mendapatkan statistic F dan probabilitas statistic F tersebut dengan mengakses atribut .fvalues dan .f\_pvalues dari model kita seperti di bawah ini.

|  |  |
| --- | --- |
| [44] | print('F-statistic:', olsmod.fvalue)  print('Probability of observing value at least as high as F-statistic:', olsmod.f\_pvalue) |

Keterangan :

* Line 44 : Untuk menampilkan nilai F-statistik dan P-value dari model machine learning dengan model OLS.

1. P-Value

P-Value adalah nilai probabilitas yang dapat diartikan sebagai besarnya peluang (probabilitas) yang diamati dari statistic uji. Variabel independen (Age, JobLevel, TotalWorkingYears, YearsAtCompany) mempunyai nilai p-value dibawah 0.05, hal ini menandakan ia signifikan melakukan prediksi variabel dependen (MonthlyIncome).

1. Fs

Fs adalah hasil akhir dari analisis ANOVA. Nilai Fs ini yang dikenal dengan F hitung dalam pengujian hipotesa dibandingkan dengan nilai p-value. Jika Fs > P-value, maka dapat dinyatakan bahwa secara simultan (bersama-sama) Salary dan Years Experience berpengaruh signifikan terhadap permintaan. Untuk menampilkan hasil perhitungan dari tabel ANOVA.

Berdasarkan tabel diatas, dapat diketahui bahwa Fs > P-value, yang artinya variabel Independen (Age, JobLevel, TotalWorkingYears, YearsAtCompany) dan MonthlyIncome berpengaruh signifikan terhadap permintaan.

1. Uji-t

Statistik-t adlaah koefisien dibagi dengan kesalahan standarnya. Kesalahan standar adalah perkiraan deviasi standar koefisien, jumlahnya bervariasi di setiap kasus. Ini dapat dianggap sebagai ukuran ketetapan yang digunakan untuk mengukur koefisien regresi. Sama seperti uji-F, nilai-p menunjukkan probabilitas untuk melihat hasil yang ekstrim seperti yang dimiliki oleh model kita. Kita juga bisa mendapatkan nilai p untuk semua variabel kita dengan memanggil .pvalues atribut pada model.

|  |  |
| --- | --- |
| [45] | print(olsmod.pvalues) |

Keterangan :

* Line 45 : Untuk menampilkan nilai p-values dari variabel independent.

1. Pengujian Asumsi

Untuk memvalidasi model machine learning dilakukan dengan analisis residual. Berikut adalah daftar pengujian atau asumsi yang akan lakukan untuk mengetahui validitas model :

|  |  |
| --- | --- |
| [46] | df\_test\_new = pd.read\_excel('E:\Data\_Urang\Mata Kuliah\Semester 3\PROJECT II\Project New\PROJECT-2\dataset\employee\_vis\_test.xlsx')  df\_test\_new2 = df\_test\_new.drop(['Unnamed: 0'], axis=1)  df\_test\_new2['MonthlyIncome Prediction'] = olsmod.predict(X)  df\_test\_new2['residual'] = olsmod.resid  df\_test\_new2 |

Keterangan :

* Line 46 : Mengimport data kemudian disimpan dalam variabel df\_test\_new. Setelah itu melakukan drop atribut. Melakukan kolom baru dengan nama MonthlyIncome Prediction yang berisikan dengan hasil prediksi. Kemudian dibuat juga kolom residual untuk menyimpan nilai residualnya.

1. Linearitas

Dengan linearitas dapat diasumsikan bahwa terdapat hubungan linear antara variabel bebas dan variabel terikat. Dengan menggunakan plot pencar dapat diketahui perbandingan nilai prediksi dengan nilai sebenarnya.

|  |  |
| --- | --- |
| [47] | # Plotting the observed vs predicted values  sns.lmplot(x='MonthlyIncome', y='MonthlyIncome Prediction', data=df\_test\_new, fit\_reg=False, size=5)  # Plotting the diagonal line  line\_coords = np.arange(vis\_test[['MonthlyIncome', 'MonthlyIncome Prediction']].min().min()-10,  vis\_test[['MonthlyIncome', 'MonthlyIncome Prediction']].max().max()+10)  plt.plot(line\_coords, line\_coords, # X and y points  color='darkorange', linestyle='--')  plt.ylabel('Predicted MonthlyIncome', fontsize=14)  plt.xlabel('Actual MonthlyIncome', fontsize=14)  plt.title('Linearity Assumption', fontsize=16)  plt.show() |



Gambar 4. 3 Grafik Asumsi Linear

Keterangan :

* Line 47 : Untuk menampilkan grafik linearitas dari perbandingan nilai MonthlyIncome dengan MonthlyIncomePrediction.

Plot sebar menenujukkan sisa yang tersebar merata di sekitar garis diagonal, sehingga kita dapat mengasumsikan bahwa ada hubungan linier antara variable independent dan dependen kita.

1. Normalitas

Berdasarkan asumsi diatas, dapat dikatakan istilah kesalahan model terdistribusi tidak normal. Memeriksa normalitas residual dengan memplotnya ke dalam histogram dan melihat nilai p dan uji normalitas Anderson-Darling dengan menggunakan normal\_ad( ) fungsi dari statsmodel untuk menghitung p-value dan kemudian membandingkannya dengan threshold 0,05. Jika p-value yang diperoleh lebih tinggi dari threshold maka dapat diasumsikan bahwa residual terdistribusi normal. Jika p-value yang diperoleh lebih kecil dari threshold, maka dapat diasumsikan bahwa residual terdistribusi tidak normal.

|  |  |
| --- | --- |
| [48] | from statsmodels.stats.diagnostic import normal\_ad  # Performing the test on the residuals  p\_value = normal\_ad(df\_test\_new2['residual'])[1]  print('p-value from the test Anderson-Darling test below 0.05 generally means non-normal:', p\_value)  # Plotting the residuals distribution  plt.subplots(figsize=(8, 4))  plt.title('Distribution of Residuals', fontsize=18)  sns.distplot(df\_test\_new2['residual'])  plt.show()  # Reporting the normality of the residuals  if p\_value < 0.05:  print('Residuals are not normally distributed')  else:  print('Residuals are normally distributed') |

Gambar 4. 4 Grafik Distribusi Residual

Keterangan :

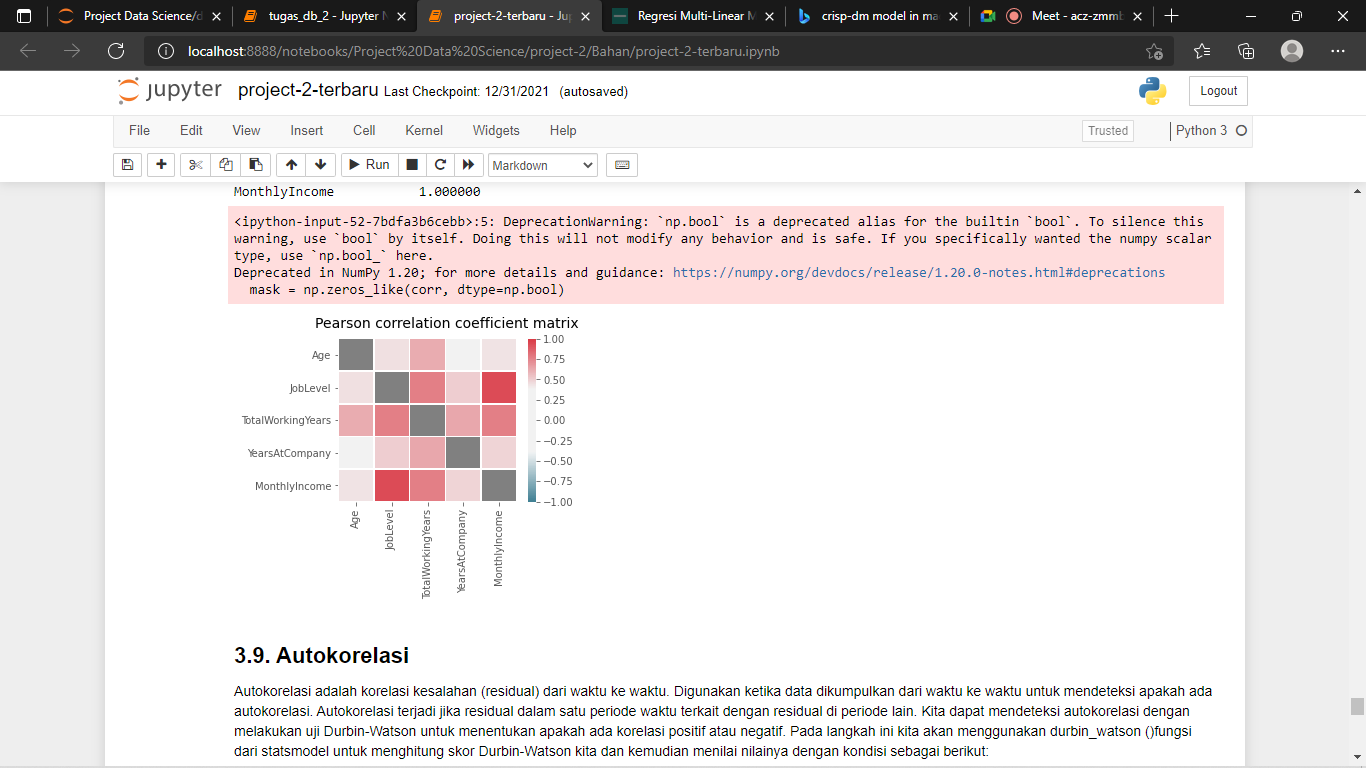
* Line 48 : Mengimport fungsi normal\_ad untuk mencari nilai normalitas. Kemudian dibuat grafik batang distribusi dari residual data.

Dari hasil perhitungan diatas, dapat diketahui bahwa nilai p-value yang dihitung menggunakan metode Anderson-Darling adalah 0,00032261. Angka tersebut berada di bawah nilai threshold yang ditentukan, yang berarti Residual terdistribusi secara tidak normal.

1. Multikolinieritas

Dari hasil asumsi diatas, dapat dikatakan bahwa prediktor yang digunakan dalam regresi berkorelasi satu sama lain. Untuk mengidentifikasi apakah ada korelasi anatara prediktor, dapat dihitung koefisien korelasi Pearson antara setiap kolom dalam data menggunakan corr( ) fungsi dari kerangka data Pandas. Kemudian kita bisa menampilkannya sebagai peta panas menggunakan heatmap( ) fungsi dari seaborn.

|  |  |
| --- | --- |
| [49] | corr = vis\_test[['Age', 'JobLevel', 'TotalWorkingYears', 'YearsAtCompany', 'MonthlyIncome']].corr()  print('Pearson correlation coefficient matrix of each variables:\n', corr)  # Generate a mask for the diagonal cell  mask = np.zeros\_like(corr, dtype=np.bool)  np.fill\_diagonal(mask, val=True)  # Initialize matplotlib figure  fig, ax = plt.subplots(figsize=(4, 3))  # Generate a custom diverging colormap  cmap = sns.diverging\_palette(220, 10, as\_cmap=True, sep=100)  cmap.set\_bad('grey')  # Draw the heatmap with the mask and correct aspect ratio  sns.heatmap(corr, mask=mask, cmap=cmap, vmin=-1, vmax=1, center=0, linewidths=.5)  fig.suptitle('Pearson correlation coefficient matrix', fontsize=14)  ax.tick\_params(axis='both', which='major', labelsize=10)  # fig.tight\_layout() |



Gambar 4. 5 Tabel Matriks Pearson Korelasi

Keterangan :

* Line 49 : Untuk mencari nilai korelasi antar variabel dan membuat tabel matriks korelasi.

1. Autokorelasi

Autokorelasi adalah korelasi kesalahan (residual) dari waktu ke waktu. Digunakan ketika data dikumpulkan dari waktu ke waktu untuk mendeteksi apakah ada autokorelasi. Autokorelasi terjadi jika residual dalam satu periode waktu terkait dengan residual di periode lain. Kita dapat mendeteksi autokorelasi dengan melakukan uij Durbin-Watson untuk menentukan apakah ada korelasi positif atau negative. Pada langkah ini kita akan menggunakan durbin\_watson() fungsi dari statsmodel untuk menghitung skor Durbin-Watson kita dan kemudian menilai nilainya dengan kondisi sebagai berikut :

1. Jika skor Durbin-Watson kurang dari 1,5 maka terdapat autokorelasi positif dan asumsi tidak terpenuhi.
2. Jika skor Durbin-Watson antara 1,5 – 2,5 maka tidak ada autokorelasi dan asumsi puas.
3. Jika skor Durbin-Watson lebih dari 2,5 maka terdapat autokorelasi negative dan asumsi tidak puas.

|  |  |
| --- | --- |
| [50] | from statsmodels.stats.stattools import durbin\_watson  durbinWatson = durbin\_watson(df\_test\_new2['residual'])  print('Durbin-Watson:', durbinWatson)  if durbinWatson < 1.5:  print('Signs of positive autocorrelation', '\n')  print('Assumption not satisfied')  elif durbinWatson > 2.5:  print('Signs of negative autocorrelation', '\n')  print('Assumption not satisfied')  else:  print('Little to no autocorrelation', '\n')  print('Assumption satisfied') |

Keterangan :

* Line 50 : Mengimport fungsi durbin\_watson, kemudian fungsi tersebut digunakan untuk mencari nilai autokorelasi dari nilai residualnya.

Didapat hasil perhitungannya adalah 2,160636228. Dapat diasumsikan bahwa terdapat sedikit atau tidak ada autokorelasi, yang berarti asumsi puas.

1. Homoskedastisitas

Dari hasil di atas, ini mengasumsikan homoskedastisitas, yang merupakan varian yang sama dalam istilah kesalahan. Heteroskedastisitas/pelanggaran homoskedastisitas terjadi ketika kita tidak memiliki varian genap di seluruh istilah kesalahan. Untuk mendeteksi homoskedastisitas, kita dapat memplot residual kita dan melihat apakah variansnya tampak seragam.

|  |  |
| --- | --- |
| [51] | # Plotting the residuals  plt.subplots(figsize=(8, 4))  plt.scatter(x=df\_test\_new2.index, y=df\_test\_new2.residual, alpha=0.8)  plt.plot(np.repeat(0, len(df\_test\_new2.index)+2), color='darkorange', linestyle='--')  plt.ylabel('Residual', fontsize=14)  plt.xlabel('Week', fontsize=14)  plt.title('Homescedasticity Assumption', fontsize=16)  plt.show() |

Keterangan :

* Line 51 : Untuk membuat penyebaran residual pada grafik apakah variansnya seragam atau tidak.

1. Prediksi Gaji

|  |  |
| --- | --- |
| [52] | # Urutan Inputan : Age, JobLevel (1-5), TotalWorkingYears, YearsAtCompany  salary\_pred = regressor.predict([[20, 1, 3, 1]])  print("Gaji yang terprediksi pada pegawai perbulan yang bekerja sepanjang tahun tersebut adalah ",salary\_pred) |

Keterangan :

* Line 52 : Menentukan prediksi gaji dengan parameter Age = 20 tahun, JobLevel = 1, TotalWorkingYears = 3 tahun, dan YearsAtCompany = 1 tahun menggunakan metode pada variabel regressor, kemudian simpan di variabel MonthlyIncome. Menampilkan nilai dari variabel MonthlyIncome.

# BAB V

# KESIMPULAN DAN SARAN

## **Kesimpulan**

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan dapat disimpulkan beberapa poin yaitu :

* 1. Hasil dari analisis karakteristik data gaji karyawan, didapatkan parameter yang berkaitan dengan gaji karyawan yaitu Age, JobLevel, TotalWorkingYears, dan YearsAtCompany. Parameter-parameter tersebut dipilih berdasarkan pedoman interpretasi koefisien korelasi dari nilai yang sedang hingga sangat kuat.
  2. Model yang dibuat berhasil melewati semua pengujian dalam langkah validasi model, sehingga kami dapat menyimpulkan bahwa model kami dapat berperforma baik untuk memprediksi gaji karyawan dengan menggunakan empat variabel independen, yaitu Age, JobLevel, TotalWorkingYears, dan YearsAtCompany. Korelasi yang didapatkan yaitu Age sebesar 0,43; JobLevel sebesar 0,94; TotalWorkingYears sebesar 0,77; dan YearsAtCompany sebesar 0, 48.
  3. Model prediksi yang dirancang dengan menggunakan *machine learning* dengan pendekatan regresi, didapatkan model prediksi gaji karyawan, dimana MonthlyIncome sebagai variabel dependen dan Age, JobLevel, TotalWorkingYears, dan YearsAtCompany variabel independen. Berdasarkan model yang diambil dari metode sckitlearn kami mendapatkan nilai akurasi sebesar 0,89. Sedangkan berdasarkan model OLS kami mendapatkan nilai akurasi sebesar 0,91. Berdasarkan kedua model akurasi yang kami gunakan, nilai akurasi yang didapatkan berkisar antara 0,89 sampai 0,91. Akurasi tersebut merupakan nilai akurasi, sehingga dapat dikatakan model *machine learning* dapat berperforma baik untuk memprediksi gaji.
  4. Visualisasi data dari hasil model prediksi gaji karyawan dapat digunakan menjadi bentuk aplikasi berbasis *web base* dengan menggunakan *framework* Django. Dengan aplikasi tersebut, admin dapat melakukan prediksi gaji karyawan dengan mudah dan dengan cepat.

## **Saran**

Saran yang dapat disampaikan pada peneliti yang akan melanjutkan dan mengembangkan penelitian ini adalah :

1. Pembuatan model prediksi yang digunakan dapat lebih beragam untuk membandingkan performa antara model satu dengan model yang lainnya.
2. Prediksi gaji pegawai dapat dihubungkan dari faktor eksternal lainnya.

# DAFTAR PUSTAKA

[1] Prasetyo B and Trisyanti U, “REVOLUSI INDUSTRI 4.0 DAN TANTANGAN PERUBAHAN SOSIAL”, *Journal of Proceedings Series*, no. 5, pp. 22-27, Nov. 2018, doi : <http://dx.doi.org/10.12962/j23546026.y2018i5.4417>

[2] H. Prasetyo and W. Sutopo, “Perkembangan Keilmuan Teknik Industri Menuju Era Industri 4.0”, *Seminar dan Konferensi Nasional IDEC*, vol. 2017, pp. 488-495, May .2017, doi : <https://idec.ft.uns.ac.id/wp-content/uploads/2017/11/Prosiding2017_ID069.pdf>

[3] O. C. Pangaribuan and I. Irwansyah, “Media Cetak Indonesia di Era Revolusi Industri 4.0,” *Jurnal Pewarta Indonesia*, vol. 1, no. 2, pp. 134–145, Oct. 2019, doi: <https://dx.doi.org/10.25008/jpi.v1i2.11>

[4] A. A. Shahroom and N. Hussin, “Industrial Revolution 4.0 and Education,” *International Journal of Academic Research in Business and Social Sciences*, vol. 8, no. 9, pp. 314-319, Oct. 2018, doi: <https://doi.org/10.24114/jh.v10i1.14138>

[5] S. Kergroach, “Industry 4.0: New Challenges And Opportunities For The Labour Market,” *Foresight and STI Governance*, vol. 11, no. 4, pp. 6–8, 2017, doi: <http://dx.doi.org/10.17323/2500-2597.2017.4.6.8>

[6] M. I. Manda and S. ben Dhaou, “Responding to the challenges and opportunities in the 4th industrial revolution in developing countries”, *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, Part F148155, pp. 244–253, 2019, doi: <http://dx.doi.org/10.1145/3326365.3326398>

[7] Y. Adrianova Eka Tuah and Anyan, “IMPLEMENTASI MODEL REGRESI LINEAR SEDERHANA UNTUK PREDIKSI GAJI BERDASARKAN PENGALAMAN LAMA BEKERJA”, *Journal Education and Technology*, vol. 1, no. 2, pp. 56-70 Dec. 2020, doi : <https://doi.org/10.31932/jutech.v1i2.1289>

[8] Tamrin A.S, Rumapea Patar, Mambo R, “PENGARUH PROFESIONALISME KERJA PEGAWAI TERHADAP TINGKAT KEPUASAN PELANGGAN PADA KANTOR PT. TASPEN CABANG MANADO”, *Jurnal Administrasi Publik*, vol. 3, no. 46, pp. 1-9 2017, doi : <https://ejournal.unsrat.ac.id/index.php/JAP/article/view/16283>

[9] E. P. Ariesanto Akhmad, “Data Mining Menggunakan Regresi Linear untuk Prediksi Harga Saham Perusahaan Pelayaran,” *Jurnal Aplikasi Pelayaran dan Kepelabuhanan*, vol. 10, no. 2, p. 120, Dec. 2020, doi: <https://dx.doi.org/10.30649/japk.v10i2.83>

[10] K. Puteri and A. Silvanie, “MACHINE LEARNING UNTUK MODEL PREDIKSI HARGA SEMBAKO DENGAN METODE REGRESI LINIER BERGANDA”, Jurnal Nasional Informatika, vol. 1, no. 2, pp. 82-94, Oct. 2020, doi : <https://ejournal-ibik57.ac.id/index.php/junif/article/view/134>

[11] A. Saiful, S. Andryana, and A. Gunaryati, “Prediksi Harga Rumah Menggunakan Web ScrappingDan Machine Learning Dengan Algoritma Linear Regression”, *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 1, pp. 41-50, Mar.2012, doi : <https://jurnal.mdp.ac.id/index.php/jatisi/article/download/701/219/>

[12] M. W. Pertiwi and R. E. Indrajit, “Metode Regresi Linier Untuk Prediksi Pengadaan Inventaris Barang”, Simposium Nasional Ilmu Pengetahuan dan Teknologi (SIMNASIPTEK) 2017, vol. 1, no. 1, pp. 27-30, 2017, doi : <https://seminar.bsi.ac.id/simnasiptek/index.php/simnasiptek-2017/article/view/114>

[13] W. Wahyudin and H. Purwanto, “PREDIKSI KASUS COVID-19 DI INDONESIA MENGGUNAKAN METODE BACKPROPAGATION DAN REGRESI LINEAR,” *Journal of Information System, Applied, Management, Accounting and Research*, vol. 5, no. 2, p. 331, May 2021, doi: <https://doi.org/10.52362/jisamar.v5i2.420>

[14] N. Nafi’iyah, “Perbandingan Regresi Linear, Backpropagation Dan Fuzzy Mamdani Dalam Prediksi Harga Emas,” *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri (SENIATI) 2016*, vol. 2, pp. 291-296, Mar. 2016, doi : <https://ejournal.itn.ac.id/index.php/seniati/article/download/840/767/>

[15] F. Ginting, E. Buulolo, and E. R. Siagian, “IMPLEMENTASI ALGORITMA REGRESI LINEAR SEDERHANA DALAM MEMPREDIKSI BESARAN PENDAPATAN DAERAH (STUDI KASUS: DINAS PENDAPATAN KAB. DELI SERDANG),” *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, Nov. 2019, doi: <http://dx.doi.org/10.30865/komik.v3i1.1602>

[16] P. Katemba and K.D. Rosita, “PREDIKSI TINGKAT PRODUKSI KOPI MENGGUNAKAN REGRESI LINEAR”, *Jurnal Ilmiah Flash,* vol. 3, pp. 42-51, Jun. 2017, doi : <http://jurnal.pnk.ac.id/index.php/flash/article/view/136/79>

[17] T. N. Putri, A. Yordan, and D. H. Lamkaruna, “Peramalan Penerimaan Mahasiswa Baru Universitas Samudra Menggunakan Metode Regresi Linear Sederhana,” *Jurnal Teknologi Informtika,* vol. 2, no. 1, Mar. 2019, doi : <http://jurnal.ummu.ac.id/index.php/J-TIFA/article/view/237/149>

[18] D. Sayan, B. Rupashri, M. Ayush, “SALARY PREDICTION USING REGRESSION TECHNIQUES.”, *Proceedings of Industry Interactive, Innovations in Science, Engineering & Technology,* Jan. 2020, doi : <https://dx.doi.org/10.2139/ssrn.3526707>

[19] U. Bansal, A. Narang, A. Sachdeva, I. Kashyap, and S. P. Panda, “Empirical Analysis Of Regression Techniques By House Price And Salary Prediction,” *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, pp. 1-13, Jan. 2021, doi: <https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012110>

[20] X. Pan, X. Wan, H. Wang, and Y. Li, “The Correlation Analysis Between Salary Gap and Enterprise Innovation Efficiency Based on the Entrepreneur Psychology,” *Frontiers in Psychology*, vol. 11, Aug. 2020, doi: <https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01749/full>

[21] S. Dan and B. Pratikno, “REGRESI LINEAR BIVARIAT SIMPEL DAN APLIKASINYA PADA DATA CUACA DI CILACAP”, *JMP,* vol. 6, no. 1, pp. 45-52, Jun. 2014, doi : <http://dx.doi.org/10.20884/1.jmp.2014.6.1.2902>

[22] T. N. Padilah and R. I. Adam, “ANALISIS REGRESI LINIER BERGANDA DALAM ESTIMASI PRODUKTIVITAS TANAMAN PADI DI KABUPATEN KARAWANG”, *Jurnal Pendidikan Matematika dan Matematika,* vol. 5, no. 2, pp. 117-128, Dec. 2019, doi : <https://jurnal.umj.ac.id/index.php/fbc/article/view/3333>

[23] P.E.N. Desak, S. Made, “UNIVERSITAS UDAYANA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM JURUSAN MATEMATIKA”, *Conference: Prosiding Seminar Nasional Matematika II,* vol. 2, pp. 43-54, Oct. 2016, doi : <https://simdos.unud.ac.id/uploads/file_penelitian_1_dir/24a473ea40f085c51398cd477f586a3a.pdf>

# LAMPIRAN

## **CODE COVERAGE**

## **GLOSARIUM**

### **Glosarium Non-Teknis**

|  |  |  |
| --- | --- | --- |
| No | Istilah | Keterangan |
| 1 | OOP | Object Oriented Programing adalah paradigma pemrograman yang berorientasi objek |
| 2 |  |  |
| 3 |  |  |

### **Glosarium Teknis**

|  |  |  |
| --- | --- | --- |
| No | Istilah | Keterangan |
| Python | | |
| 1 | import | Digunakan untuk mengimpor sebagian isi modul serta bukan keseluruhan isi modul. |
| 2 | fields | Sebuah ruang disebuah GUI window yang berfungsi untuk memasukkan sebuah teks atau angka. |
| 3 | class | Sebuah *blueprint* (cetakan) dari objek (atau instance) yang dibuat. |
| 4 | path() | Memberi tahu baris perintah folder mana yang perlu dicari saat mencari file. |
| 5 | url() | Pintu pertama masuk user ke aplikasi pada python. |
| 6 | def | Fungsi yang berisi perintah atau baris kode yang dikelompokkan menjadi satu kesatuan. |
| 7 | render() | Mengambil RequestContext |
| 8 | request | Melempar parameter dalam URL sampai mengirim header khusus dan verifikasi SSL |
| Mysql | | |
| 1 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# TABEL - TABEL