

**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,

Ketua Kelompok

Yang membuat pernyataan,

Anggota Kelompok

Bachtiar Ramadhan

1.20.4.077

Nur Tri Ramadhanti Adiningrum

NPM 1.20.4.061

ABSTRAK

Perusahaan tidak dapat dipisahkan dengan tenaga kerja. Salah satu aspek yang berpengaruh terhadap kemajuan sebuah perusahaan adalah kinerja karyawannya. Pemberian gaji yang sesuai adalah salah satu faktor penting untuk mendorong kinerja tenaga kerja. Sangat disayangkan, perkembangan perusahaan saat ini belum memiliki suatu media keputusan untuk melakukan prediksi gaji karyawan berdasarkan kualitas data. Penelitian ini bertujuan untuk mengetahui prediksi gaji karyawan berdasarkan faktor-faktor spesifik. Dalam penelitian ini, faktor-faktor yang dilakukan pengujian diantaranya variabel independen berupa Age, JobLevel, TotalWorkingYears, dan YearsAtCompany. Kemudian variabel dependen berupa MonthlyIncome. Teknik analisis data menggunakan analisis regresi linear multivariat yang digunakan untuk melakukan prediksi gaji karyawan. Hasil prediksi gaji karyawan akan ditampilkan berbasis *web base*. Model yang dibuat berhasil melewati semua pengujian dalam langkah validasi model, sehingga dapat disimpulkan bahwa model yang dibuat dapat berperforma baik untuk memprediksi gaji karyawan. Hasil 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.

Kata Kunci : Prediksi Gaji, Regresi linear Multivariat, Faktor Spesifik, *Web base*

ABSTRACT

The company cannot be separated from the workforce. One aspect that affects the progress of a company is the performance of its employees. Providing an appropriate salary is one of the important factors to boost the performance of the workforce. Unfortunately, the current development of the company does not have a decision media to predict employee salaries based on data quality. This study aims to determine the prediction of employee salaries based on specific factors. In this study, the factors that were tested included independent variables in the form of Age, JobLevel, TotalWorkingYears, and YearsAtCompany. Then the dependent variable is MonthlyIncome. The data analysis technique used multivariate linear regression analysis which was used to predict employee salaries. The results of employee salary predictions will be displayed on a web-base. The model created successfully passed all the tests in the model validation step, so it can be concluded that the model created can perform well for predicting employee salaries. The results of employee salary predictions can be used as a form of web-based application using the Django framework. With this application, admins can predict employee salaries easily and quickly.

Keywords : Salary Prediction, Multivariate Linear Regression, Specific Faktors, Web base

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran 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 *MACHINE LEARNING*” 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

LEMBAR PERSETUJUAN.....	ii
SURAT PERNYATAAN TIDAK MELAKUKAN PLAGIARISME.....	iii
ABSTRAK.....	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	3
1.5 Ruang Lingkup.....	3
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI	5
2.1. Tinjauan Studi	5
2.1.1 Penelitian Terkait.....	5
BAB III ANALISIS DAN PERANCANGAN.....	20
3.1 Analisis Sistem.....	20
3.1.1 Analisis Sistem Yang Sedang Berjalan.....	21
3.1.2 Analisis Sistem Yang Akan Dibangun.....	21
3.2 Perancangan Sistem (UML)	23
3.2.1 <i>Use case</i> Diagram.....	24
3.2.2 <i>Class diagram</i>	31
3.2.3 <i>Sequence Diagram</i>	31
3.2.4 <i>Collaboration Diagram</i>	37
3.2.5 <i>Activity Diagram</i>	39
3.2.6 <i>Statechart Diagram</i>	43

3.2.7	<i>Component Diagram</i>	46
3.2.8	<i>Deployment Diagram</i>	47
3.3	Perancangan Database	47
3.3.1	CDM (<i>Conceptual Data Model</i>)	47
3.3.2	PDM (<i>Physical Data Model</i>)	48
3.3.3	ERD (<i>Entity Relationship Diagram</i>)	48
3.4	Struktur Menu	48
3.5	Antarmuka (<i>Interface</i>)	49
3.5.1	Antarmuka Halaman Depan	49
3.5.2	Antarmuka Halaman <i>Login</i>	50
3.5.3	Antarmuka Halaman Registrasi	50
3.5.4	Antarmuka Halaman <i>Dashboard</i>	51
3.5.5	Antarmuka Halaman Prediksi	51
3.5.6	Antarmuka Halaman Hasil Prediksi	52
3.5.7	Antarmuka Halaman Data Pegawai	52
3.5.8	Antarmuka Halaman Tambah Data Pegawai	53
3.5.9	Antarmuka Halaman Edit Data Pegawai	53
3.5.10	Antarmuka Halaman Visualisasi	54
3.6	Analisis Kebutuhan	54
3.6.1	Kebutuhan Fungsional (<i>Functional Requirements</i>)	54
3.6.2	Kebutuhan Non-Fungsional (<i>Non-Functional Requirements</i>)	55
BAB IV IMPLEMENTASI DAN PENGUJIAN		57
4.1	Implementasi Aplikasi	57
4.1.1	Implementasi Proses <i>Login</i>	57
4.1.2	Implementasi Proses Registrasi	59
4.1.3	Implementasi Proses Prediksi Gaji	61
4.1.4	Implementasi Proses Hasil Prediksi Gaji	63
4.1.5	Implementasi Proses CRUD Data Pegawai	65
4.1.6	Implementasi Proses Menampilkan Visualisasi	72
4.2	Implementasi Model <i>Machine learning</i>	75
4.2.1	Himpunan Data	75
4.2.2	Proses Data Mining & Pengetahuan	82
4.2.3	Evaluasi Data	89

4.3	Pengujian dan Hasil Pengujian	104
4.3.1	Pengujian dan Hasil Pengujian Proses <i>Login</i> Admin	104
4.3.2	Pengujian dan Hasil Pengujian Proses Pengolahan Data	105
4.3.3	Pengujian dan Hasil Pengujian Proses Prediksi Gaji	106
4.3.4	Pengujian dan Hasil Pengujian Proses Visualisasi	108
4.3.5	Pengujian dan Hasil Pengujian Proses <i>Logout</i>	109
BAB V KESIMPULAN DAN SARAN		110
5.1	Kesimpulan	110
5.2	Saran	111
DAFTAR PUSTAKA		112
LAMPIRAN.....		115
CODE COVERAGE		115
Code Coverage Folder Prediksi		115
Code Coverage Folder PrediksiGaji		115
GLOSARIUM.....		115
Glosarium Non-Teknis		115
Glosarium Teknis		116
TABEL - TABEL.....		117

DAFTAR GAMBAR

Gambar 3. 1 Flowmap Pemrosesan Gaji	21
Gambar 3. 2 Flowmap Proses Login Admin	22
Gambar 3. 3 Flowmap Proses Prediksi Gaji Pegawai	23
Gambar 3. 4 Use case Diagram	24
Gambar 3. 5 Class diagram	31
Gambar 3. 6 Sequence Diagram Login Admin.....	32
Gambar 3. 7 Sequence Diagram Registrasi	33
Gambar 3. 8 Sequence Diagram Melakukan Prediksi Gaji.....	34
Gambar 3. 9 Sequence Diagram Melakukan Pengolahan Data	35
Gambar 3. 10 Sequence Diagram Melakukan Visualisasi Data.....	36
Gambar 3. 11 Sequence Diagram logout admin.....	37
Gambar 3. 12 Collaboration Diagram Melakukan Login	37
Gambar 3. 13 Collaboration Diagram Registrasi.....	38
Gambar 3. 14 Collaboration Diagram Melakukan Prediksi Gaji	38
Gambar 3. 15 Collaboration Diagram Melakukan Visualisasi Data	38
Gambar 3. 16 Collaboration Diagram Melakukan Visualisasi Data	39
Gambar 3. 17 Activity Diagram Login Admin.....	39
Gambar 3. 18 Activity Diagram Registrasi	40
Gambar 3. 19 Activity Diagram Melakukan Pengolahan Data.....	41
Gambar 3. 20 Activity Diagram Melakukan Prediksi Gaji.....	42
Gambar 3. 21 Activity Diagram Melakukan Visualisasi	42
Gambar 3. 22 Activity Diagram Logout Admin.....	43
Gambar 3. 23 Statechart Diagram Login Admin.....	44
Gambar 3. 24 Statechart Diagram Registrasi.....	44
Gambar 3. 25 Statechart Diagram Melakukan Prediksi.....	45
Gambar 3. 26 Statechart Diagram Pengolahan Data.....	45
Gambar 3. 27 Statechart Diagram Visualisasi.....	46
Gambar 3. 28 Component Diagram.....	46
Gambar 3. 29 Deployment Diagram.....	47
Gambar 3. 30 Conceptual Data Model	47
Gambar 3. 31 Physical Data Model.....	48
Gambar 3. 32 Entity Relationship Diagram	48

Gambar 3. 33 Struktur Menu Aplikasi	49
Gambar 3. 34 Halaman Depan Sistem.....	49
Gambar 3. 35 Halaman Login Sistem	50
Gambar 3. 36 Halaman Registrasi Sistem	50
Gambar 3. 37 Halaman Dashboard Sistem.....	51
Gambar 3. 38 Halaman Prediksi Sistem.....	51
Gambar 3. 39 Halaman Hasil Prediksi Sistem.....	52
Gambar 3. 40 Halaman Data Pegawai Sistem.....	52
Gambar 3. 41 Halaman Tambah Data Pegawai Sistem	53
Gambar 3. 42 Halaman Edit Data Pegawai Sistem.....	53
Gambar 3. 43 Halaman Visualisasi Sistem.....	54
Gambar 4. 1 Flowmap Login Aplikasi	59
Gambar 4. 2 Flowmap Registrasi Aplikasi.....	61
Gambar 4. 3 Flowmap Prediksi Gaji	64
Gambar 4. 4 Flowmap Prediksi Gaji	72
Gambar 4. 5 Flowmap Menampilkan Visualisasi Data	75
Gambar 4. 6 Visualisasi Data Training	81
Gambar 4. 7 Visualisasi Data Testing	88
Gambar 4. 8 Model OLS	89
Gambar 4. 9 Nilai P-Values Dari Variabel Independen.....	92
Gambar 4. 10 Model OLS	93
Gambar 4. 11 Grafik Asumsi Linear	95
Gambar 4. 12 Grafik Distribusi Residual	96
Gambar 4. 13 Tabel Matriks Pearson Korelasi.....	99
Gambar 4. 14 Tabel VIF	100
Gambar 4. 15 Homoskedastisitas	103
Gambar 4. 16 Pengujian dan Hasil Pengujian Proses Login Admin.....	104
Gambar 4. 17 Pengujian dan Hasil Pengujian Proses Pengolahan Data	105
Gambar 4. 18 Pengujian dan Hasil Pengujian Proses Prediksi Gaji	106
Gambar 4. 19 Pengujian dan Hasil Pengujian Proses Visualisasi.....	108
Gambar 4. 20 Pengujian dan Hasil Pengujian Proses Logout.....	109

DAFTAR TABEL

Tabel 2. 1 Hasil Tinjauan Dari Penelitian Terkait	5
Tabel 3. 1 Definisi Aktor.....	24
Tabel 3. 2 Definisi Use case.....	25
Tabel 3. 3 Skenario Use case Machine learning	26
Tabel 3. 4 Skenario Use case Login Admin	27
Tabel 3. 5 Skenario Use case Registrasi.....	27
Tabel 3. 6 Skenario Use case Melakukan Prediksi Gaji Karyawan	28
Tabel 3. 7 Skenario Use case Melakukan Visualisasi Data	29
Tabel 3. 8 Skenario Use case Melakukan Pengolahan Data	29
Tabel 3. 9 Skenario Use case Logout Admin	30
Tabel 3. 10 Perangkat Keras Yang Digunakan.....	55
Tabel 3. 11 Perangkat Lunak Yang Digunakan.....	56
Tabel 4. 1 Hasil pengujian proses login admin.....	104
Tabel 4. 2 Hasil pengujian proses pengolahan data.....	105
Tabel 4. 3 Hasil pengujian proses prediksi gaji.....	107
Tabel 4. 4 Hasil pengujian proses visualisasi	108
Tabel 4. 5 Hasil pengujian proses logout.....	109

BAB I

PENDAHULUAN

1.1 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 satu 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 memiliki 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.

1.2 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?

1.3 Tujuan Penelitian

Tujuan penelitian 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?

1.4 Manfaat Penelitian

Manfaat penelitian 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.

1.5 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.

1.6 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.

2. BAB II LANDASAN TEORI

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

3. 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* django.

4. BAB IV IMPLEMENTASI

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

5. BAB V KESIMPULAN DAN SARAN

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

BAB II

LANDASAN TEORI

2.1. 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.

2.1.1 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.	<i>Machine learning.</i>	Teknik model analisis data menggunakan model analisis regresi linear untuk mengetahui pengaruh variable bebas terhadap variabel 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. <i>Purposive sampling:</i> perusahaan pelayaran PT. BULL. <i>Quota sampling: data time series</i> 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).	<i>Machine learning.</i>	Penelitian ini menggunakan model <i>Cross Industry Standard Process for Data Mining</i> (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.	<i>Machine learning.</i>	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 <i>web scrapping</i> 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.	<i>Machine learning.</i>	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

				<p>sebesar 0.8859830993050699 atau 88%. Namun, memiliki nilai galat cukup tinggi sebesar 259171.91 atau Rp. 259.171,91.</p>
5.	Pengadaan inventaris barang.[12]	Data Inventaris Barang Dinas Pariwisata Pemuda dan Olahraga Kota Tasikmalaya 2012 – 2016. Variabel kode barang, variabel tahun, variabel harga barang.	<i>Machine learning.</i>	<p>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.</p>
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	<i>Machine learning.</i>	<p>Penelitian ini menggunakan model <i>backpropagation</i> dan regresi linear. Dari hasil simulasi Matlab dengan tiga fungsi pelatihan yaitu <i>traincgb</i> didapatkan nilai rata-rata error yaitu sebesar</p>

		menggunakan data kasus aktif.		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 dengan 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).	<i>Machine learning</i>	Digunakan tiga model dalam penelitian ini, yaitu regresi linear, <i>backpropagation</i> , 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 $\pm 0,05$.
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 bulan), Caturwulan (4 bulan), 1 semester (6 bulan) dan 1 tahun (12 bulan).	<i>Machine learning</i>	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

				<p>direncanakan sebelumnya oleh dinas pendapatan dapat berjalan dengan lancar, dan juga dapat membuat program-program yang baru agar dapat meningkatkan pendapatan daerah untuk memajukan daerah tersebut.</p>
9.	<p>Prediksi Tingkat Produksi Kopi.[16]</p>	<p>Data produksi kopi diperoleh dari BPS Kabupaten Manggarai dengan mengambil data produksi kopi lima tahun terakhir yaitu dari tahun 2011 – 2015.</p>	<p><i>Machine learning</i></p>	<p>Dalam penelitian ini model yang digunakan adalah model regresi linear sederhana. Berdasarkan hasil penelitian dan pembahasan tentang prediksi tingkat produksi kopi menggunakan regresi linear sederhana maka dapat disimpulkan bahwa data yang digunakan untuk memprediksi kopi merupakan data <i>time series</i>, dan setelah melakukan</p>

				<p>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 pengujian menggunakan MAPE jauh lebih baik dalam menghitung akurasi prediksi produksi kopi.</p>
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.	<i>Machine learning</i>	<p>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</p>

				<p>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 mempertimbangan apa-apa saja yang harus ditingkatkan agar penerimaan mahasiswa baru pada tahun selanjutnya dapat meningkat.</p>
11.	<p>Prediksi gaji menggunakan teknik regresi. [18]</p>	<p>Data <i>position</i>, <i>level</i>, <i>salary</i>. Variabel <i>position</i>, variabel <i>level</i>, dan variabel <i>salary</i>.</p>	<p><i>Machine learning</i></p>	<p>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</p>

				diperoleh dengan menerapkan regresi k-terdekat.
12.	Analisis empiris teknik regresi berdasarkan harga rumah dan prediksi gaji. [19]	Variabel pada dataset <i>employee salary</i> : <i>salary</i> , <i>total years of experience</i> , <i>certification</i> , <i>lead</i> . Variabel pada dataset <i>house prices</i> : <i>price</i> , <i>sqft_living</i> , <i>bedrooms</i> , <i>bathrooms</i> , <i>sqft_living15</i> , <i>sqft_lot</i> , <i>sqft_above</i> , <i>sqft_basement</i> .	<i>Machine learning</i>	Model yang digunakan adalah <i>Simple Linear Regression</i> (SLR) dan <i>Multiple Linear Regression</i> (MLR). <i>Multiple Linear Regression</i> baik daripada <i>Simple Linear Regression</i> . 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. <i>Explanatory variable</i> : <i>Monetary salary of senior manager</i> , <i>Senior management salary</i> , <i>Ordinary</i>	<i>Machine learning</i>	Metode yang digunakan dalam penelitian ini yaitu <i>multiple regression</i> . <i>Monetary salary</i> dan <i>equity salary of senior managers</i> , <i>ordinary employee salary</i> , dan <i>the internal salary gap of the senior management team</i>

		<p><i>employee salary, Salary gap within the senior management team, Salary gap between senior managers and ordinary employees.</i></p> <p><i>Explained variable : Enterprise innovation efficiency. Control variable : Enterprise size, Return on assets, Asset-liability ratio, Years of establishment, Dummy variable.</i></p>		<p>semuanya berkorelasi positif secara signifikan pada tingkat 1% dengan efisiensi inovasi perusahaan. Korelasi antara <i>salary gap</i> antara <i>senior managers</i> dan <i>ordinary employees</i> dengan <i>enterprise innovation efficiency</i> tidak jelas. Koefisien korelasi regresi antar variabel berada dalam kisaran yang dapat diterima, menunjukkan bahwa model yang digunakan tidak memiliki multikolinearitas yang signifikan.</p>
14.	<p>Regresi linear bivariat simpel dan aplikasinya pada data cuaca di Cilacap.[21]</p>	<p>Dataset dari BMKG Kabupaten Cilacap dari bulan Januari 2009 - Februari 2014. Dua variabel respon : curah hujan (Y1), kelembaban udara suatu wilayah (Y2) terhadap satu</p>	<i>Machine learning</i>	<p>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</p>

		variabel respon : temperatur udara (X).		derajat Celcius maka curah hujan akan naik sebesar 45,892mm serta nilai rata-rata error bulanan = - 0,00697mm dan $MDE_{E(1)} = 151,2132$. 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 $MD_{E(2)} = 1,206636$.
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	<i>Machine learning</i>	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 dihasilkan dari data produksi dibagi luas panen. Variabel bebas : yaitu produksi, luas panen, luas tanam, rata-rata curah hujan, dan rata-rata hari hujan		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	<i>Machine learning</i>	Model yang digunakan dalam penelitian adalah analisis <i>multivariate</i> yaitu analisis regresi linier dan <i>logistic ordinal</i> . 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

	<p>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:</p> <p>Pendapatan responden, Status Pekerjaan, dikelompokkan dalam: pekerja berusaha sendiri; berusaha sendiri dibantu oleh pekerja</p>	<p>rata-rata pendapatan pedagang kaki lima adalah $Y = -1.982 + 0.654\text{pendidikan} + 0.134\text{curahan jam kerja} + 0.817\text{Jumlah tenaga kerja non keluarga}$. 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 $G(x) = -16.308 - 0.519 \text{ status kawin} + 0,739 \text{ sifat layanan} + 1,19663 \text{ curahan jam kerja} - 1,062 \text{ Jumlah tenaga kerja non keluarga}$.</p>
--	--	---

		sementara/tidak dibayar; berusaha sendiri dibantu oleh pekerja permanen/dibayar; Karyawan/ Pekerja; Pekerja tidak dibayar.		
--	--	---	--	--

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.

3.1 Analisis Sistem

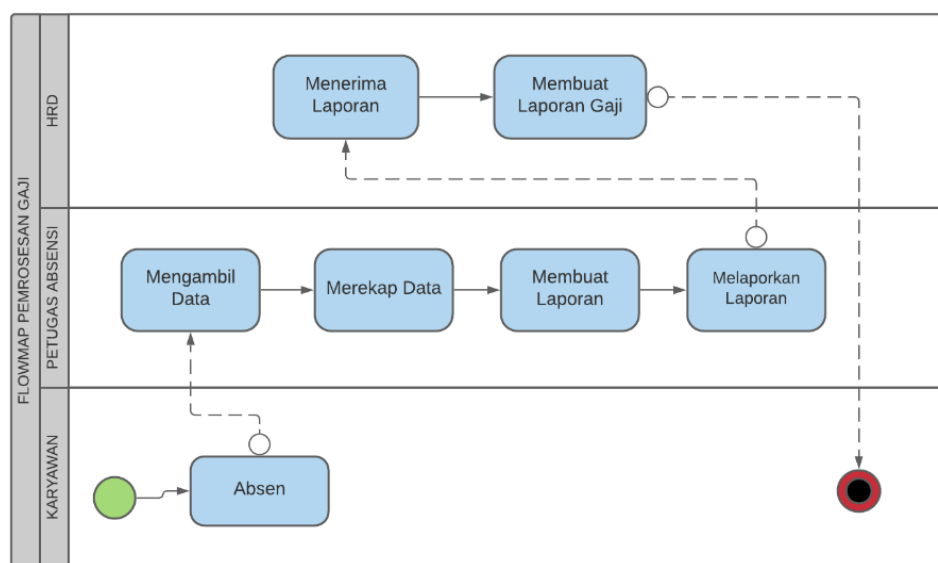
Analisis sistem ialah penjabaran dari suatu sistem informasi yang utuh ke berbagai macam bagian-bagian komponennya yang bertujuan untuk mengevaluasi permasalahan atau kendala yang terjadi pada suatu sistem, sehingga nantinya dapat dilakukan perbaikan ataupun pengembangan pada sistem tersebut.

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.

3.1.1 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.



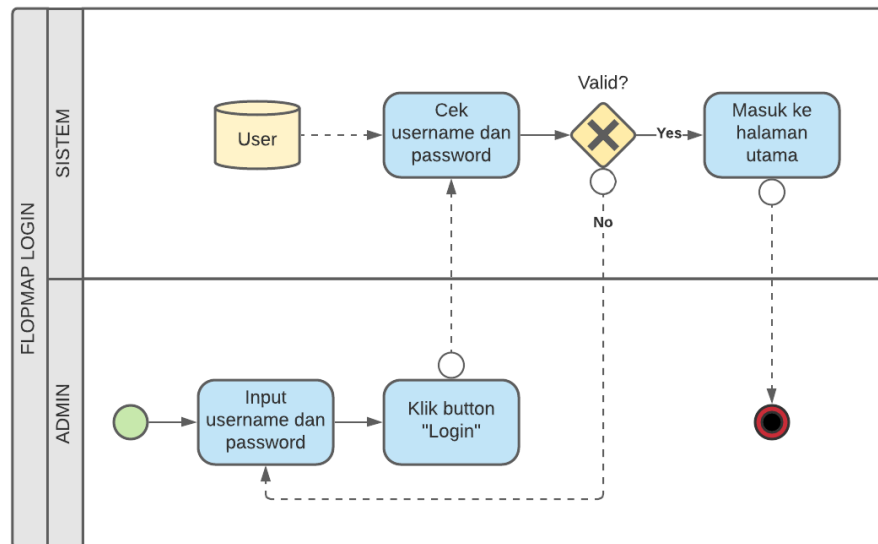
Gambar 3. 1 *Flowmap* Pemrosesan Gaji

3.1.2 Analisis Sistem Yang Akan Dibangun

Pada sub bab ini akan dijelaskan mengenai aplikasi website prediksi gaji pegawai berdasarkan faktor-faktor spesifik 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.

3.1.2.1 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.



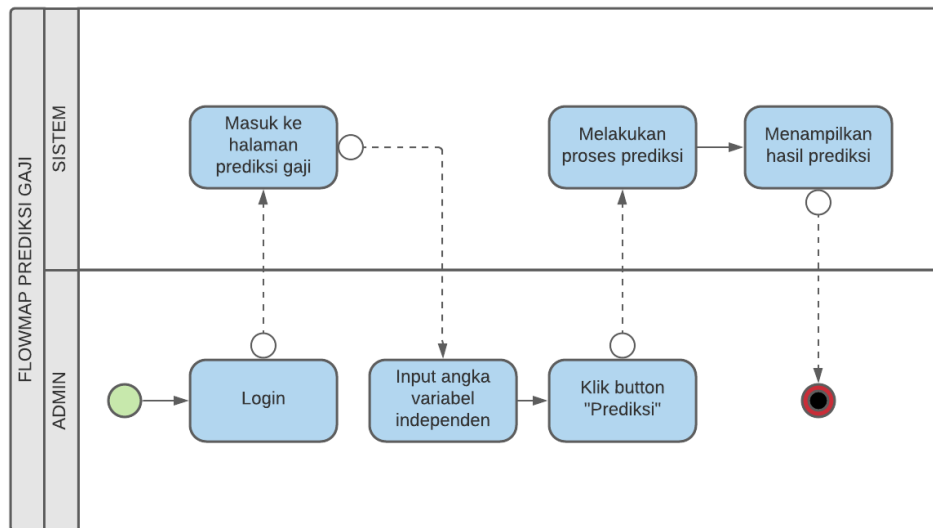
Gambar 3. 2 Flowmap Proses Login Admin

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*.

3.1.2.2 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.

3.2 Perancangan Sistem (UML)

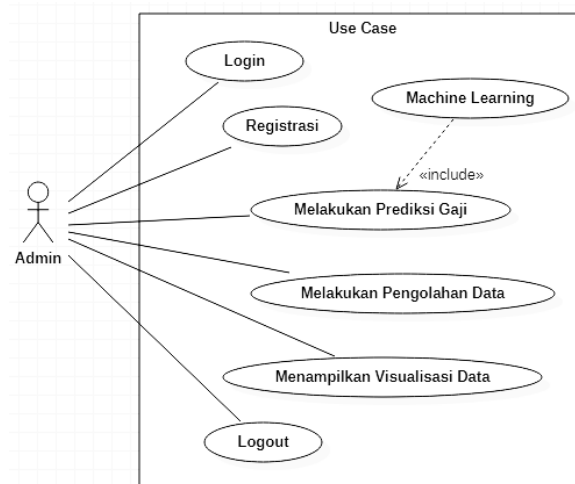
UML adalah singkatan dari Unified Modeling Language yang didefinisikan sebagai sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau *software* berbasis objek. UML juga dapat dikatakan sebagai bahasa spesifikasi standar yang digunakan untuk mendokumentasikan, menspesifikasikan,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

3.2.1 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

a. Definisi Aktor

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

Tabel 3. 1 Definisi Aktor

No	Aktor	Deskripsi
1	Admin	<ul style="list-style-type: none"> • <i>Login</i> • Melakukan Prediksi Gaji • Melakukan Pengolahan Data

		<ul style="list-style-type: none"> • Melakukan Visualisasi Data • <i>Logout</i>
--	--	---

b. Definisi *Use case*

Tabel 3. 2 Definisi *Use case*

No	Aktor	Deskripsi
1	<i>Machine learning</i>	Merupakan aktivitas interaksi antara manusia dengan mesin. Dalam hal ini, admin membuat model prediksi <i>Machine learning</i> .
2	<i>Login</i>	Merupakan aktivitas <i>login</i> yang dilakukan oleh admin. Sebelum admin masuk ke aplikasi ia harus melakukan <i>login</i> terlebih dahulu.
3	Registrasi	Merupakan aktivitas registrasi yang dilakukan oleh admin. Apabila admin belum memiliki akun, maka admin harus melakukan registrasi terlebih dahulu.
4	Melakukan Prediksi Gaji	Merupakan aktivitas memprediksi gaji karyawan yang dilakukan oleh admin dengan parameter lama bekerja seorang karyawan.
5	Melakukan Pengolahan Data	Merupakan aktivitas insert, read, update dan delete data karyawan yang dilakukan oleh admin.

6	Menampilkan Visualisasi Data	Merupakan aktivitas visualisasi grafik data karyawan yang dilakukan oleh admin.
7	<i>Logout</i>	Merupakan aktivitas <i>logout</i> yang dilakukan oleh admin. Admin dapat keluar dari aplikasi jika telah selesai melakukan pekerjaan pada aplikasi.

c. 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 aktor. 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	<i>Machine learning</i>
Tujuan	Pembuatan model prediksi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	

Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Membuat model prediksi 	<ul style="list-style-type: none"> Melakukan olah data gaji karyawan
Kondisi Akhir	Masuk pada aplikasi

Tabel 3. 4 Skenario *Use case Login Admin*

Identifikasi	
Nomor	1
Nama	<i>Login</i>
Tujuan	Sebelum masuk pada aplikasi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Input <i>username</i> dan <i>password</i> 	<ul style="list-style-type: none"> Mengecek data dan menentukan aktor tersebut admin atau user
Kondisi Akhir	Masuk pada aplikasi

Tabel 3. 5 Skenario *Use case Registrasi*

Identifikasi	
Nomor	1
Nama	Registrasi

Tujuan	Membuat akun admin
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Input <i>username</i>, dan <i>password</i> 	<ul style="list-style-type: none"> Menyimpan data admin atau user
Kondisi Akhir	Masuk pada halaman <i>login</i>

Tabel 3. 6 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
<ul style="list-style-type: none"> Menginput Data 	<ul style="list-style-type: none"> Melakukan Prediksi Gaji
Kondisi Akhir	Admi dapat memprediksi gaji

Tabel 3. 7 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
<ul style="list-style-type: none"> Input data karyawan 	<ul style="list-style-type: none"> Menampilkan grafik visualisasi data karyawan
Kondisi Akhir	Admin dapat memvisualisasikan data

Tabel 3. 8 Skenario *Use case* Melakukan Pengolahan Data

Identifikasi	
Nomor	1
Nama	Melakukan Pengolahan Data
Tujuan	Melakukan pengolahan data karyawan
Deskripsi	
Aktor	Admin

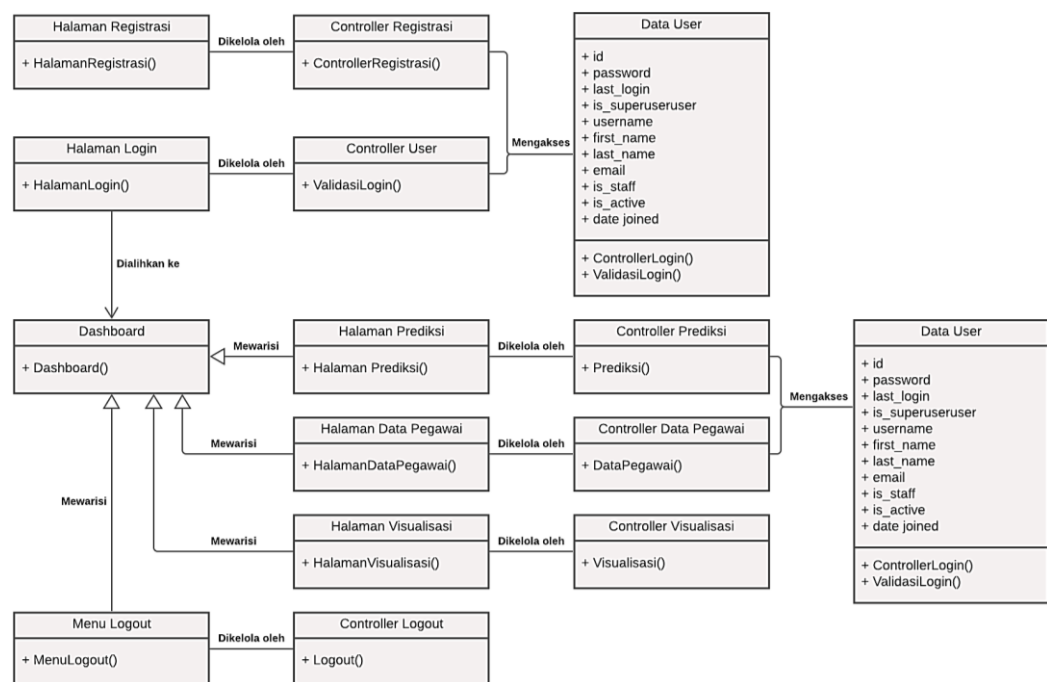
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Insert, read, update, dan delete data karyawan. 	<ul style="list-style-type: none"> Mengubah data karyawan yang diubah oleh admin
Kondisi Akhir	Admin dapat mengelola data

Tabel 3. 9 Skenario *Use case Logout* Admin

Identifikasi	
Nomor	1
Nama	<i>Logout</i>
Tujuan	Keluar dari aplikasi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Menekan tombol <i>logout</i> 	<ul style="list-style-type: none"> Melakukan proses keluar dari aplikasi
Kondisi Akhir	Keluar dari aplikasi

3.2.2 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.



Gambar 3.5 Class diagram

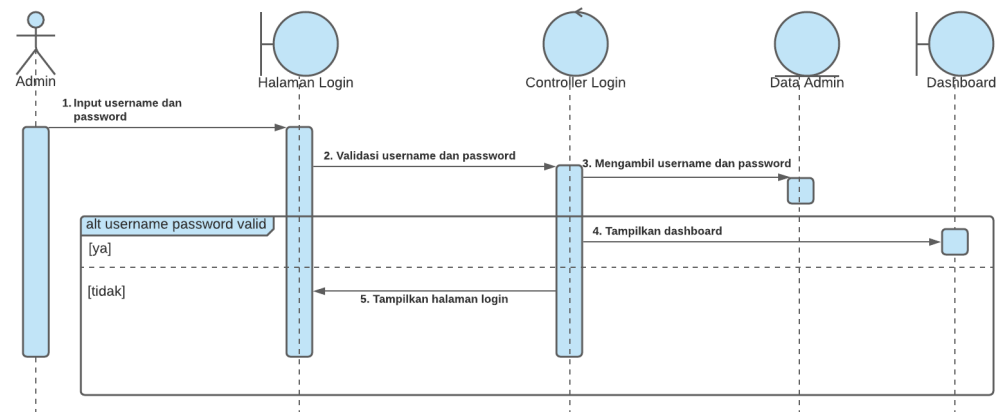
3.2.3 Sequence Diagram

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*.

A. 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. 6 *Sequence Diagram Login Admin*

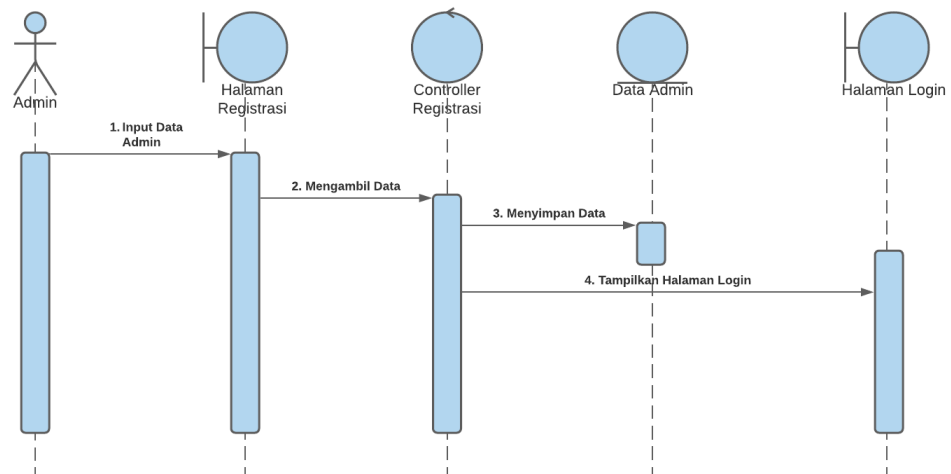
Keterangan :

1. Admin memasukkan *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*.

B. *Sequence Diagram* Melakukan Registrasi

Berikut ini merupakan *Sequence Diagram* Registrasi Admin menjelaskan hubungan antara admin pada aplikasi.

Berikut *Sequence Diagram* Melakukan Registrasi pada gambar.



Gambar 3. 7 *Sequence Diagram* Registrasi

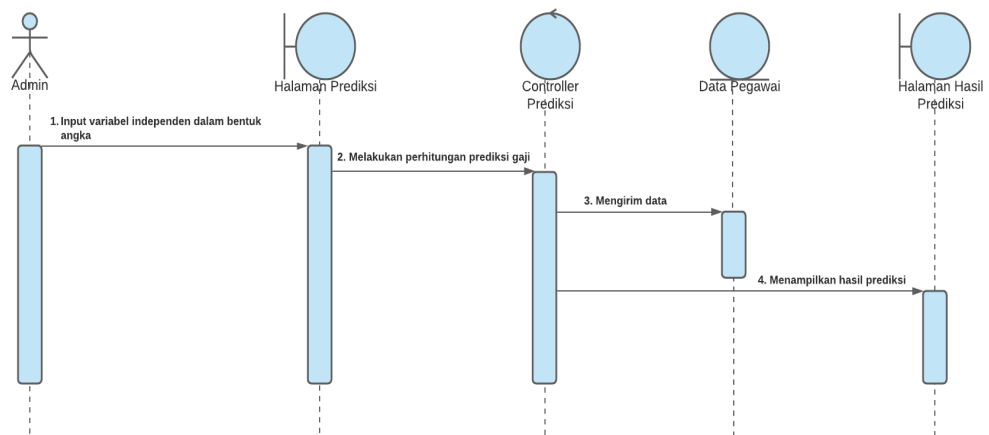
Keterangan :

1. Admin menginputkan data admin di halaman registrasi.
2. Mengambil data oleh Controller registrasi.
3. Controller *login* mengirimkan data ke data admin kemudian data tersebut disimpan.
4. Jika data admin telah tersimpan pada data admin maka akan ditampilkan halaman *login*.

C. *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. 8 *Sequence Diagram* Melakukan Prediksi Gaji

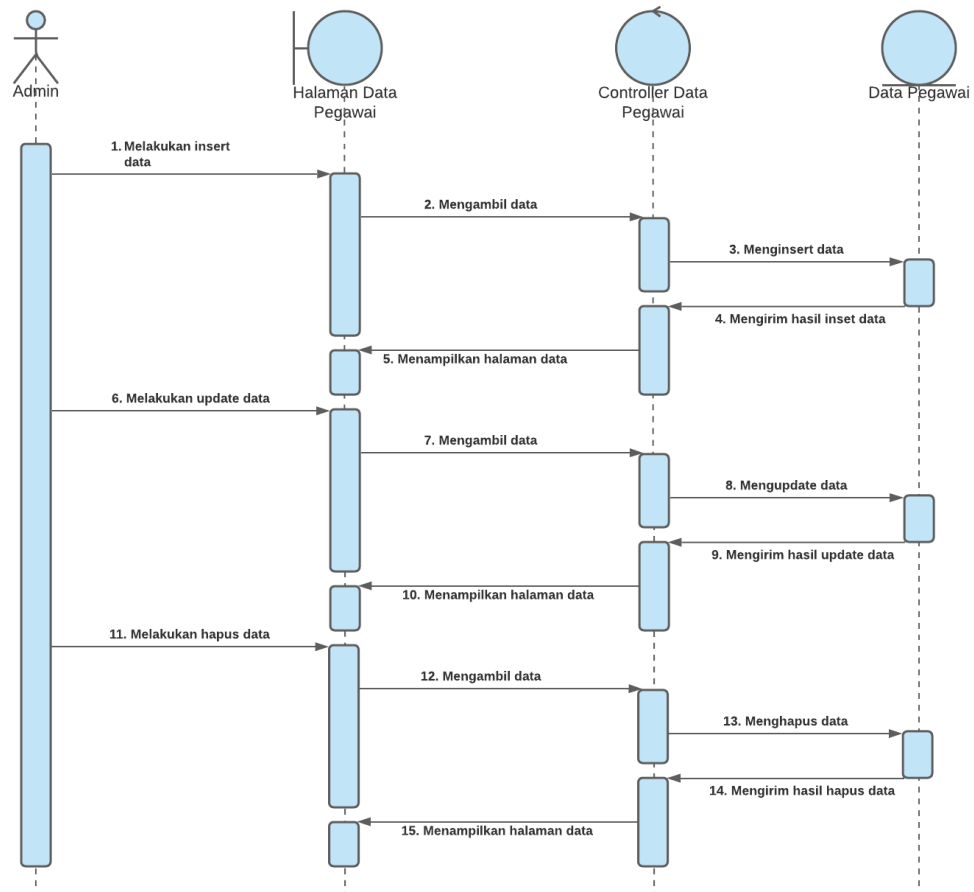
Keterangan :

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.

D. 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. 9 *Sequence Diagram* Melakukan Pengolahan Data

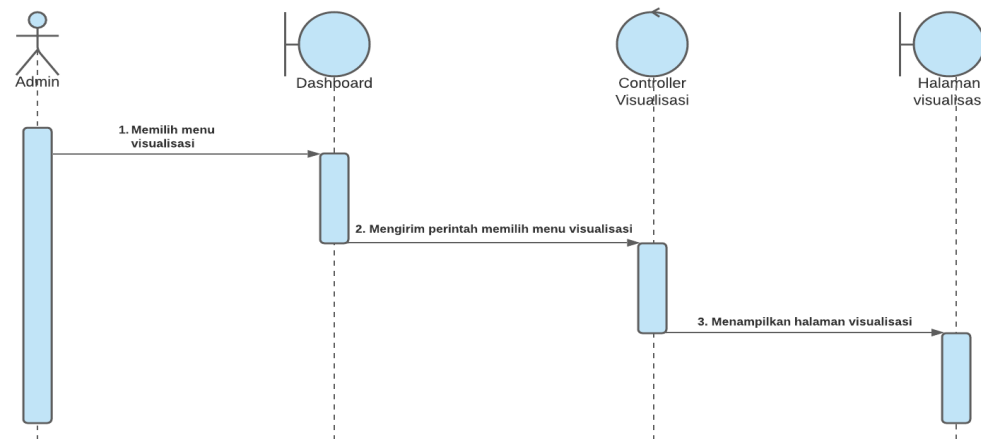
Keterangan :

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.

E. *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. 10 *Sequence Diagram* Melakukan Visualisasi Data

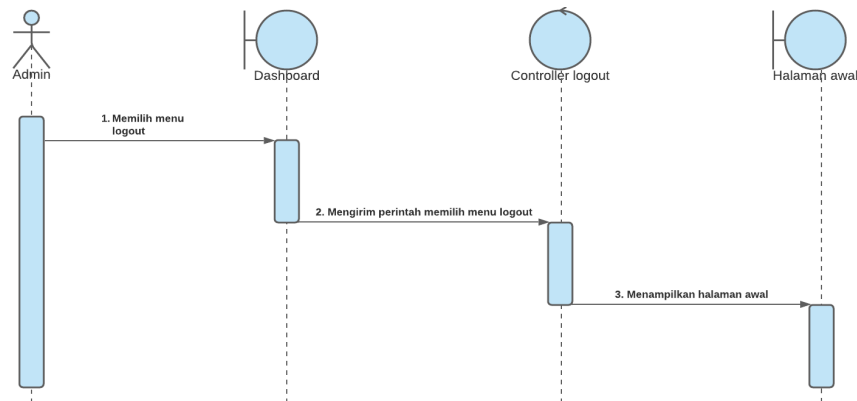
Keterangan :

1. Admin memilih menu halaman visualisasi pada *dashboard*.
2. *Dashboard* mengirim perintah memilih menu visualisasi ke controller visualisasi.
3. Controller visualisasi menampilkan halaman visualisasi.

F. *Sequence Diagram Melakukan Logout*

Berikut ini merupakan *Sequence Diagram logout* Admin menjelaskan hubungan antara admin pada aplikasi.

Berikut *Sequence Diagram logout* Admin pada gambar.



Gambar 3. 11 *Sequence Diagram* logout admin

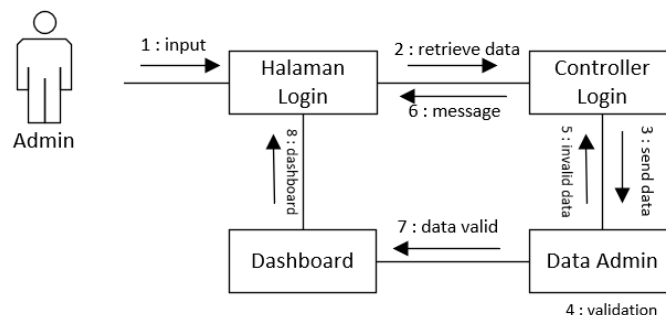
Keterangan :

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.

3.2.4 *Collaboration Diagram*

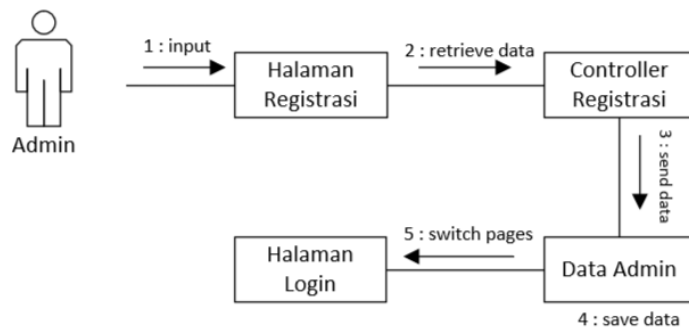
Collaboration Diagram dapat dipakai untuk memodelkan interaksi antar objek yang ada di dalam sistem. Berbeda dari *Sequence Diagram* yang lebih menunjukkan kronologis dari operasi-operasi yang dilakukan, *Collaboration Diagram* ini lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh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

A. *Collaboration Diagram* Melakukan *Login*



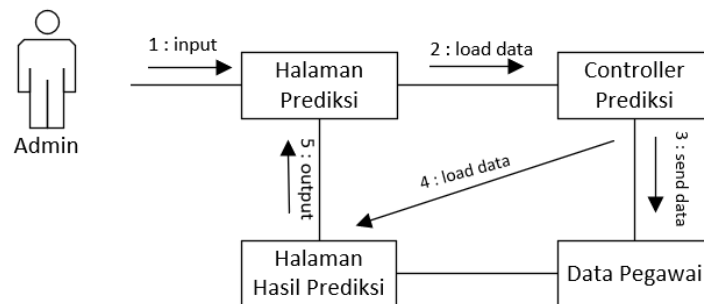
Gambar 3. 12 *Collaboration Diagram* Melakukan *Login*

B. Collaboration Diagram Melakukan Registrasi



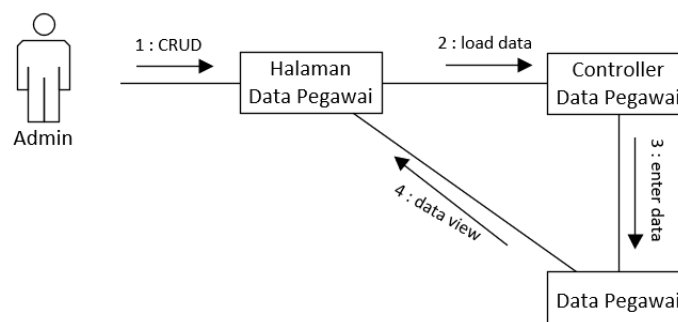
Gambar 3. 13 Collaboration Diagram Registrasi

C. Collaboration Diagram Melakukan Prediksi Gaji



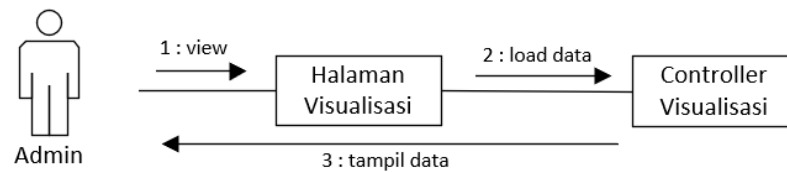
Gambar 3. 14 Collaboration Diagram Melakukan Prediksi Gaji

D. Collaboration Diagram Melakukan Pengolahan Data



Gambar 3. 15 Collaboration Diagram Melakukan Visualisasi Data

E. Collaboration Diagram Melakukan Visualisasi Data

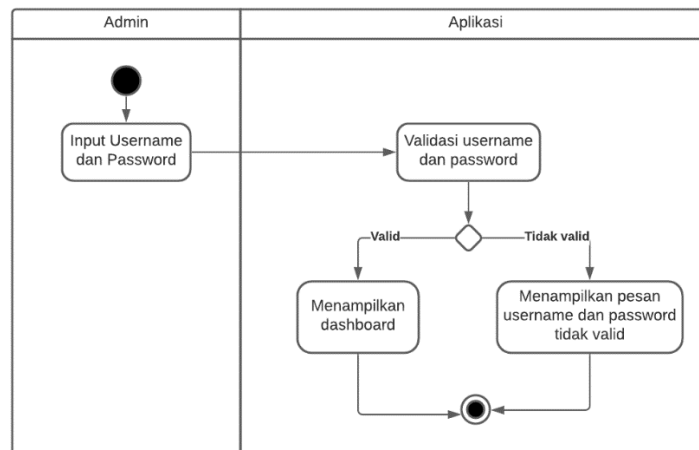


Gambar 3. 16 Collaboration Diagram Melakukan Visualisasi Data

3.2.5 Activity 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.

A. Activity Diagram Login Admin



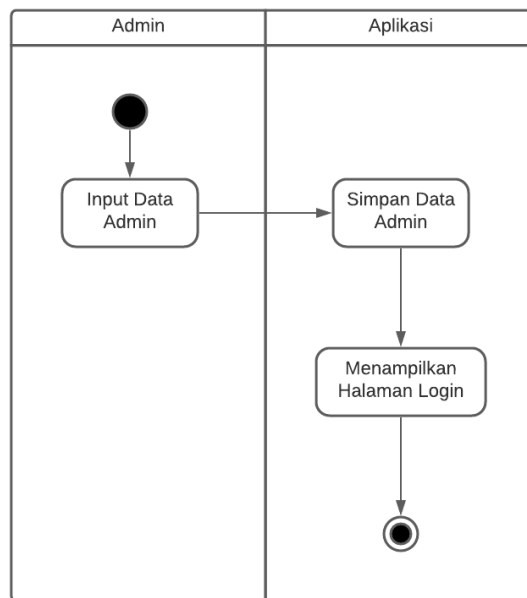
Gambar 3. 17 Activity Diagram Login Admin

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.

B. Activity Diagram Melakukan Registrasi Admin

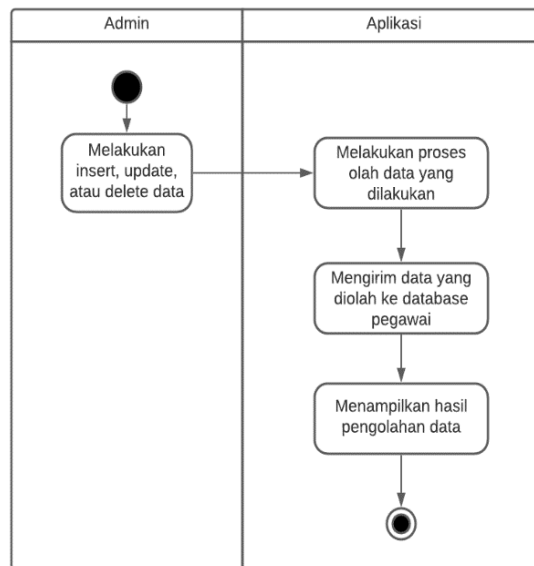


Gambar 3. 18 Activity Diagram Registrasi

Keterangan :

1. Admin menginput data admin.
2. Aplikasi menyimpan data admin.
3. Aplikasi menampilkan halaman *login*.

C. Activity Diagram Melakukan Pengolahan Data

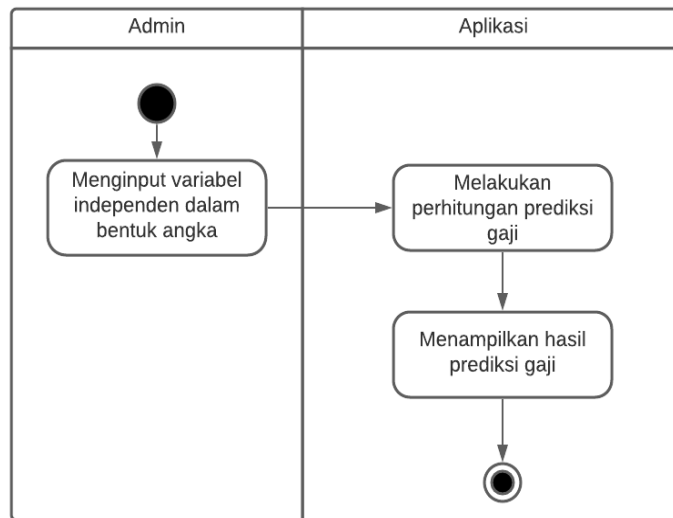


Gambar 3. 19 Activity Diagram Melakukan Pengolahan Data

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.

D. Activity Diagram Melakukan Prediksi Gaji

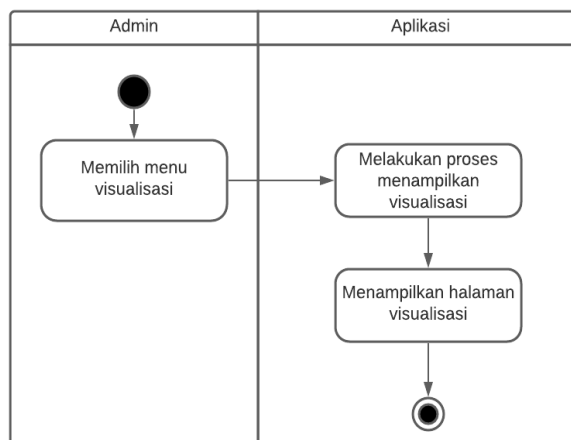


Gambar 3. 20 Activity Diagram Melakukan Prediksi Gaji

Keterangan :

1. Admin menginputkan variabel independent dalam bentuk angka pada form prediksi.
2. Aplikasi melakukan proses perhitungan prediksi gaji.
3. Aplikasi menampilkan hasil prediksi.

E. Activity Diagram Melakukan Visualisasi

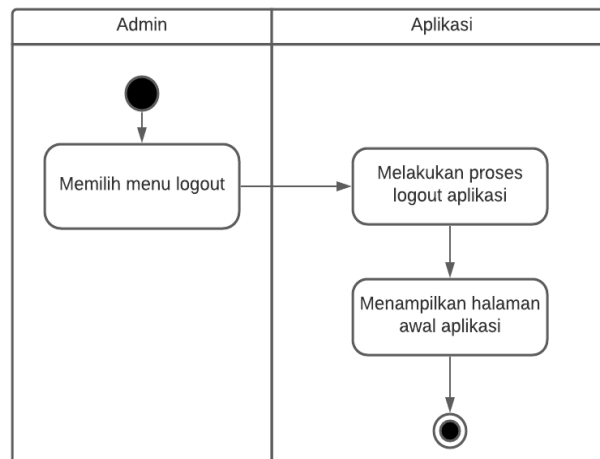


Gambar 3. 21 Activity Diagram Melakukan Visualisasi

Keterangan :

1. Admin memilih menu visualisasi.
2. Aplikasi melakukan proses menampilkan visualisasi.
3. Aplikasi menampilkan halamann visualisasi.

F. Activity Diagram Logout Admin



Gambar 3. 22 Activity Diagram Logout Admin

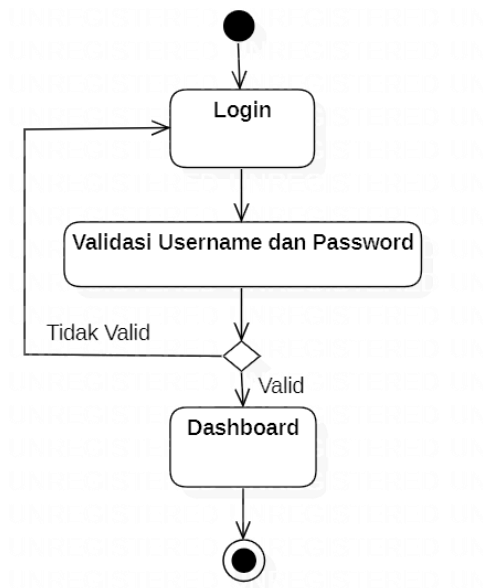
Keterangan :

1. Admin memilih menu *logout*.
2. Aplikasi melakukan proses *logout* aplikasi.
3. Aplikasi menampilkan halaman awal aplikasi.

3.2.6 Statechart Diagram

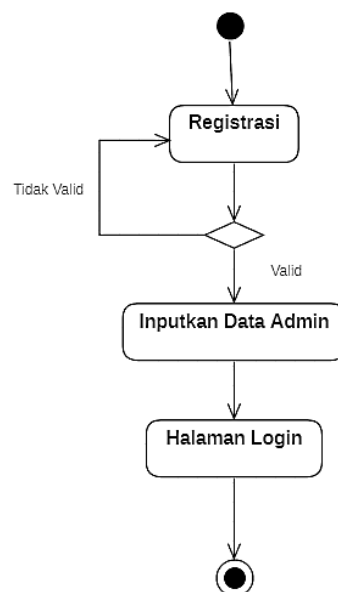
Statechart Diagram dapat menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) dari suatu objek pada sistem sebagai akibat dari stimuli yang diterimanya. Pada umumnya *Statechart Diagram* dapat menjelaskan atau menggambarkan class tertentu (satu class dapat memiliki lebih dari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

A. Statechart Diagram Login Admin



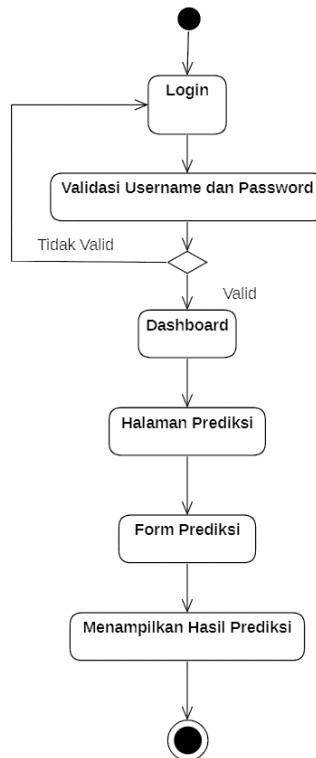
Gambar 3. 23 Statechart Diagram Login Admin

B. Statechart Diagram Registrasi Admin



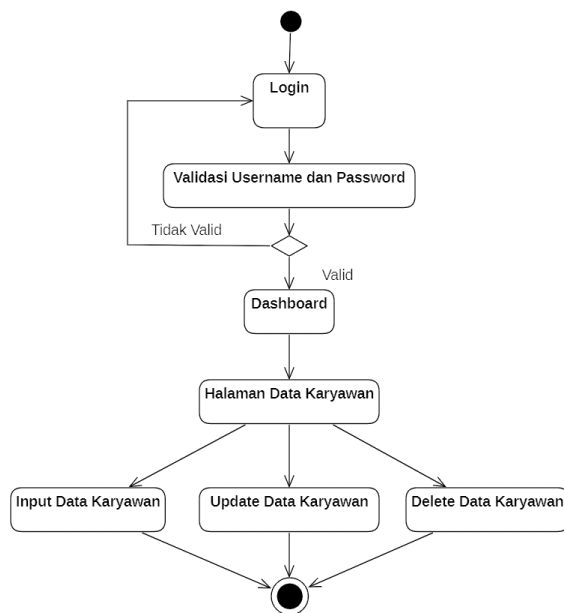
Gambar 3. 24 Statechart Diagram Registrasi

C. Statechart Diagram Melakukan Prediksi



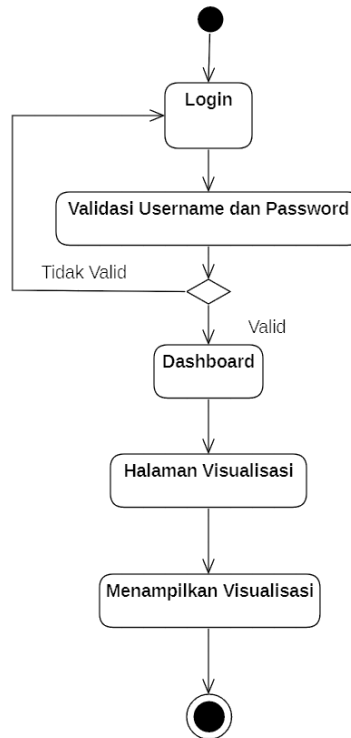
Gambar 3. 25 Statechart Diagram Melakukan Prediksi

D. Statechart Diagram Pengolahan Data



Gambar 3. 26 Statechart Diagram Pengolahan Data

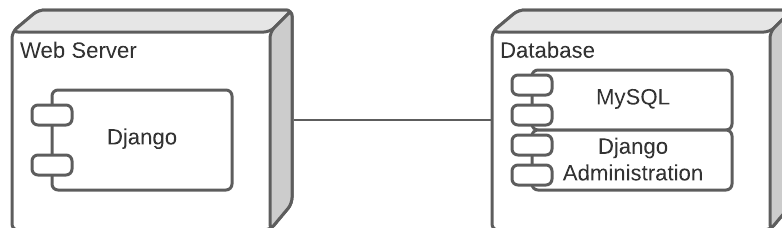
E. Statechart Diagram Visualisasi



Gambar 3. 27 Statechart Diagram Visualisasi

3.2.7 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. 28 Component Diagram

3.2.8 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)

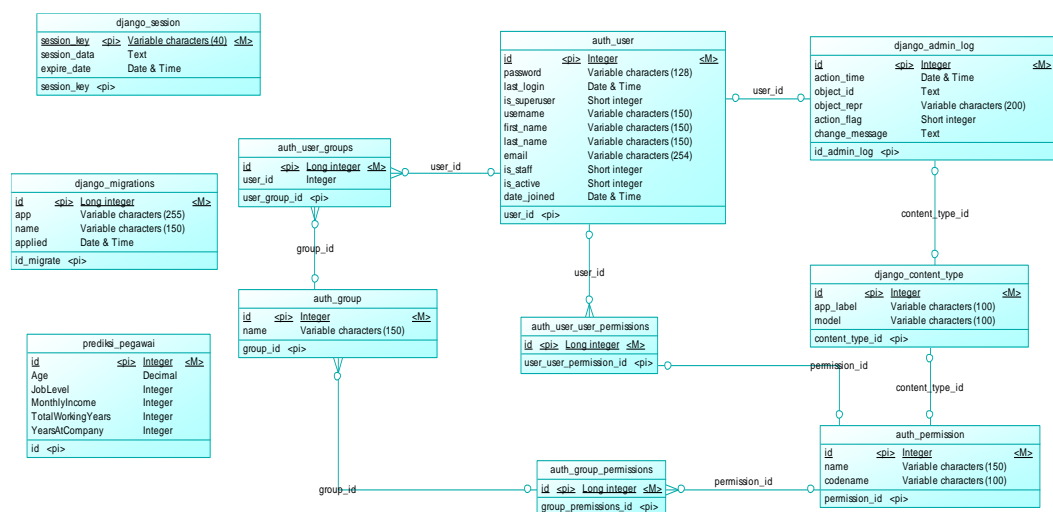


Gambar 3. 29 *Deployment Diagram*

3.3 Perancangan Database

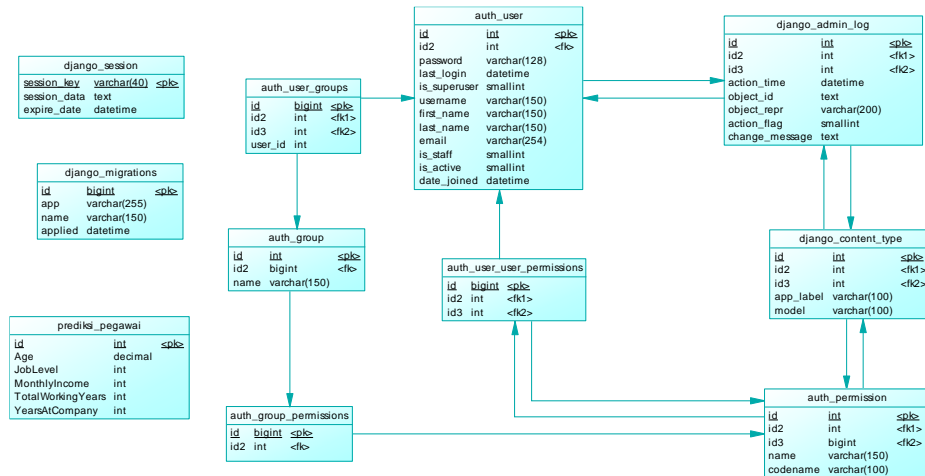
Perancangan database merupakan proses untuk menentukan dan pengaturan data yang dibutuhkan untuk mendukung rancangan sistem, agar terciptanya pemrosesan data yang lebih efisien. Struktur tabel meliputi nama tabel, tipe data, nama atribut dan data relasi seperti primary dan foreign key.

3.3.1 CDM (*Conceptual Data Model*)



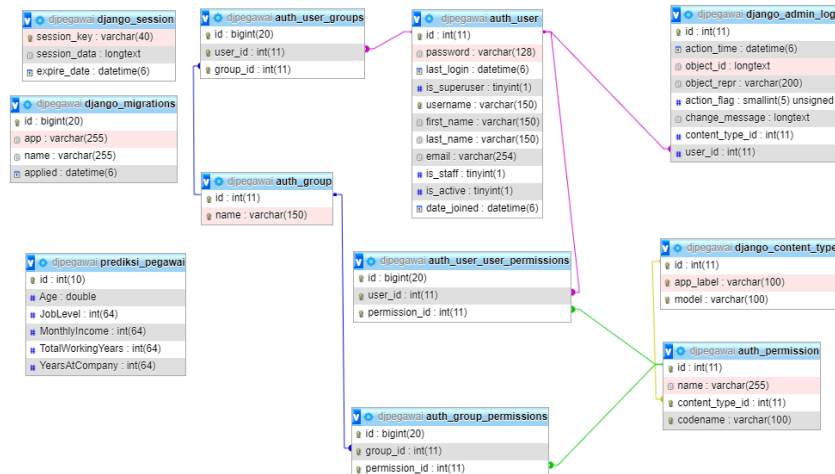
Gambar 3. 30 *Conceptual Data Model*

3.3.2 PDM (Physical Data Model)



Gambar 3. 31 Physical Data Model

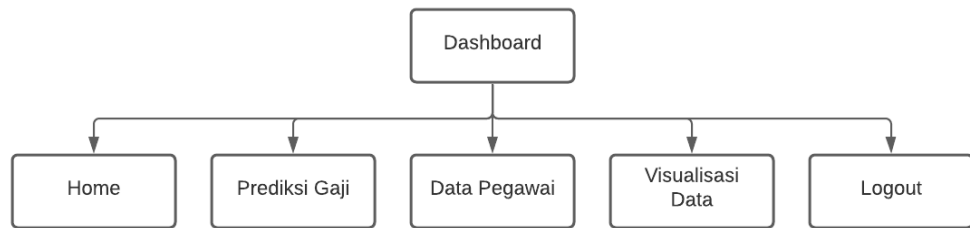
3.3.3 ERD (Entity Relationship Diagram)



Gambar 3. 32 Entity Relationship Diagram

3.4 Struktur Menu

Struktur menu ialah bentuk umum dari suatu rancangan aplikasi dalam memudahkan pengguna untuk menjalankan aplikasi. Sehingga saat menjalankan aplikasi, pengguna tidak mengalami kesulitan dalam memilih menu yang diinginkan. Berikut adalah struktur menu dari aplikasi Prediksi Gaji Pegawai secara keseluruhan :



Gambar 3. 33 Struktur Menu Aplikasi

3.5 Antarmuka (*Interface*)

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

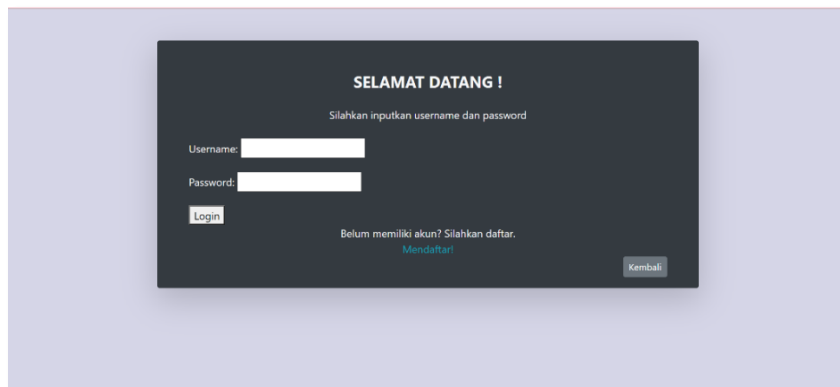
3.5.1 Antarmuka Halaman Depan



Gambar 3. 34 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.

3.5.2 Antarmuka Halaman *Login*

The image shows a login interface with a dark gray background. At the top, it says "SELAMAT DATANG !" in white. Below that, it says "Silahkan inputkan username dan password" in a smaller white font. There are two white input fields: "Username:" and "Password:". Below the "Password:" field is a "Login" button. At the bottom, it says "Belum memiliki akun? Silahkan daftar." in white, with a blue link "Mendaftar!". In the bottom right corner, there is a "Kembali" button.

Gambar 3. 35 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.

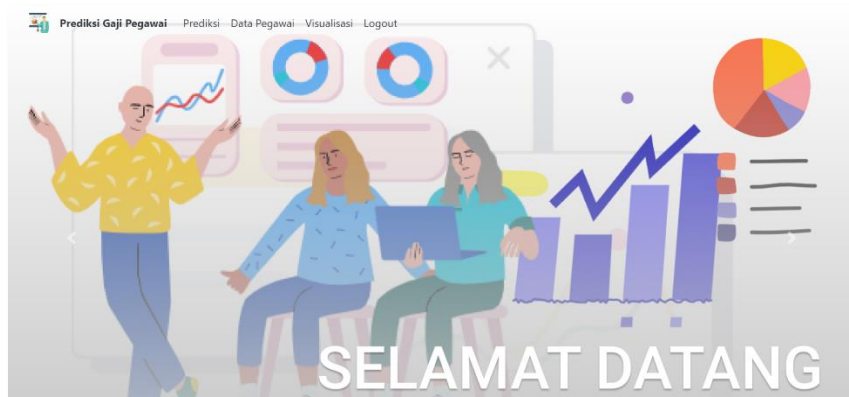
3.5.3 Antarmuka Halaman Registrasi

The image shows a registration interface with a dark gray background. At the top, it says "SILAHKAN MEMBUAT AKUN !" in white. Below that, it says "Silahkan inputkan username dan password yang akan dibuat." in a smaller white font. There are two white input fields: "Username:" and "Password:". Below the "Password:" field is a "Register" button. Below the "Register" button, there are four bullet points: "Your password can't be too similar to your other personal information.", "Your password must contain at least 8 characters.", "Your password can't be a commonly used password.", and "Your password can't be entirely numeric.". Below these bullet points is a "Password confirmation:" field and a text "Enter the same password as before, for verification.".

Gambar 3. 36 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.

3.5.4 Antarmuka Halaman *Dashboard*



Gambar 3. 37 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, Visualisasi, dan *Logout*.

3.5.5 Antarmuka Halaman *Prediksi*

A screenshot of the 'Prediksi Gaji Pegawai' form. The form has a dark background with white text. At the top, it says 'PREDIKSI GAJI PEGAWAI' and 'Silahkan inputkan tahun dalam bentuk angka'. Below this, there are four input fields: 'Inputkan Usia', 'Inputkan Job Level', 'Inputkan Total Tahun Bekerja', and 'Inputkan Total Tahun Bekerja Di Perusahaan'. At the bottom, there is a green button labeled 'Lakukan Prediksi' and a smaller button labeled 'Kembali'.

Gambar 3. 38 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 perusahaan ke dalam form untuk mengetahui hasil prediksi gaji pegawai. Kemudian *user* dapat menekan button “Lakukan

Prediksi” untuk melihat hasil prediksi berdasarkan tahun yang diinputkan.

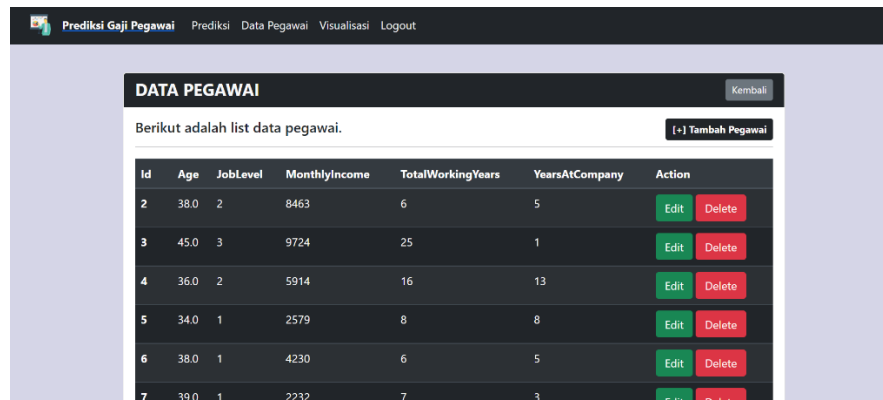
3.5.6 Antarmuka Halaman Hasil Prediksi



Gambar 3. 39 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.

3.5.7 Antarmuka Halaman Data Pegawai

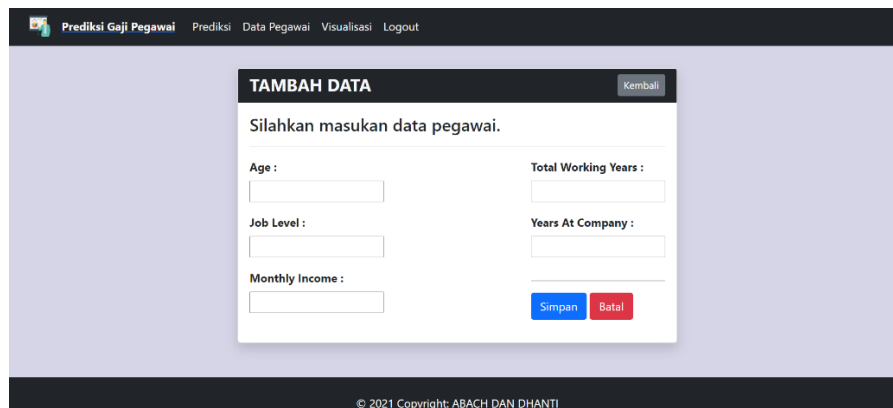


Gambar 3. 40 Halaman Data Pegawai Sistem

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.

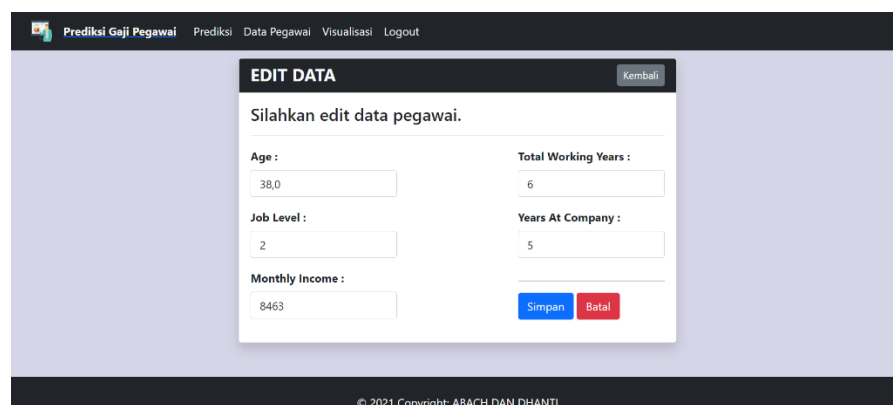
3.5.8 Antarmuka Halaman Tambah Data Pegawai



Gambar 3. 41 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.

3.5.9 Antarmuka Halaman Edit Data Pegawai

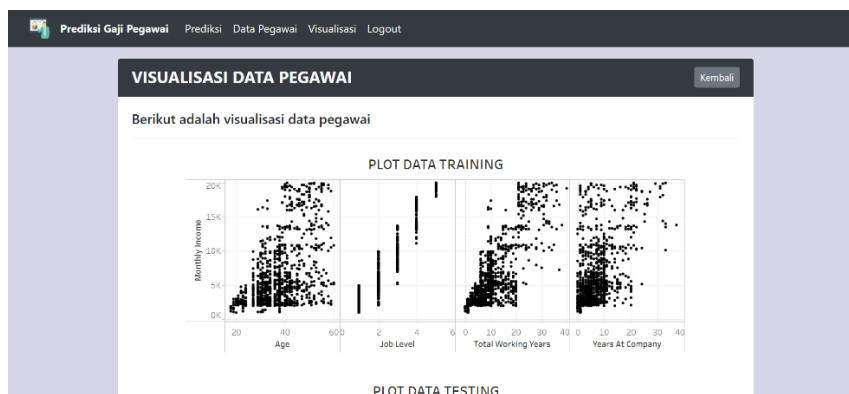


Gambar 3. 42 Halaman Edit Data Pegawai Sistem

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.

3.5.10 Antarmuka Halaman Visualisasi



Gambar 3. 43 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

3.6 Analisis Kebutuhan

Analisis kebutuhan adalah langkah awal untuk menentukan gambaran perangkat yang akan dihasilkan ketika pengembang melaksanakan sebuah proyek pembuatan perangkat lunak. Selain itu, Analisa kebutuhan adalah suatu proses untuk mendapatkan informasi, mode, spesifikasi tentang perangkat lunak.

3.6.1 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.

3.6.2 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 :

A. Perangkat Keras (*Hardware*) Yang Digunakan

Tabel 3. 10 Perangkat Keras Yang Digunakan

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

B. Perangkat Lunak (*Software*) Yang Digunakan

Tabel 3. 11 Perangkat Lunak Yang Digunakan

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

C. 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.

4.1 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.

4.1.1 Implementasi Proses *Login*

A. 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*.

B. 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>
```

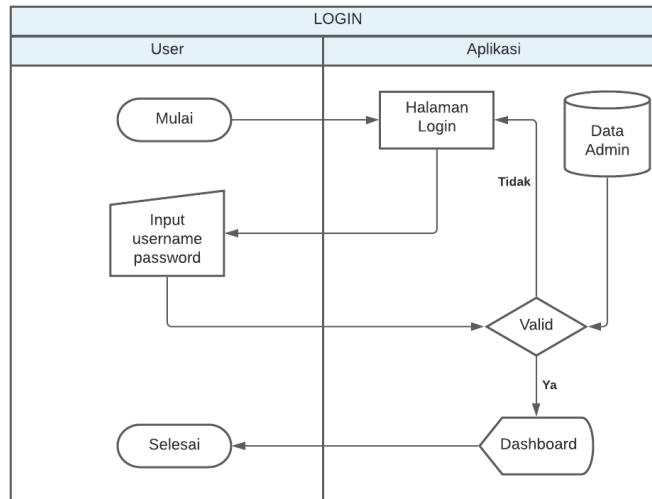
Kode diatas merupakan tampilan dari halaman *login*. Terdapat form yang berisikan inputan *username* dan *password*. Selain itu tedapat link “Mendaftar!” untuk melakukan registrasi.

C. URL

```
path('Login/', LoginView.as_view(), name="login_url"),
```

Kode diatas merupakan URL untuk mengakses halaman *login*.

D. Logika Fungsi *Login*



Gambar 4. 1 Flowmap Login Aplikasi

Keterangan :

1. Aplikasi menampilkan halaman *login*.
2. User menginputkan *username* dan *password*.
3. Aplikasi melakukan validasi *username* dan *password* dengan menyesuaikan pada database data admin.
4. Jika valid akan menampilkan halaman *dashboard*.
5. Jika tidak valid akan menampilkan kembali ke halaman *login*.
6. Selesai

4.1.2 Implementasi Proses Registrasi

A. 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 dibalikkan ke halaman *login*.

B. 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.

C. URL

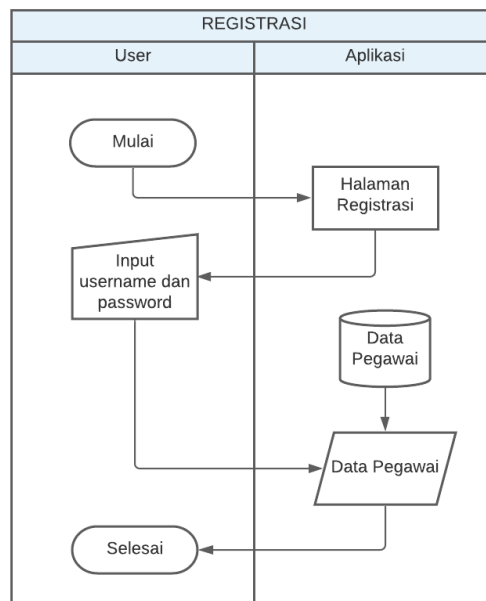
```

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

```

Kode diatas merupakan URL untuk mengakses halaman registrasi.

D. Logika Fungsi Registrasi



Gambar 4. 2 Flowmap Registrasi Aplikasi

Keterangan :

1. Aplikasi menampilkan halaman registrasi.
2. User menginputkan *username* dan *password*.
3. Aplikasi akan menyimpan data *username* dan *password* ke database data pegawai.
4. Selesai

4.1.3 Implementasi Proses Prediksi Gaji

A. 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.

B. 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 perusahaan. Selain itu tedapat button “Lakukan Prediksi” untuk melakukan proses prediksi.

C. URL

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

Kode diatas merupakan URL untuk mengakses halaman prediksi gaji.

4.1.4 Implementasi Proses Hasil Prediksi Gaji

A. 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.

B. 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>
```

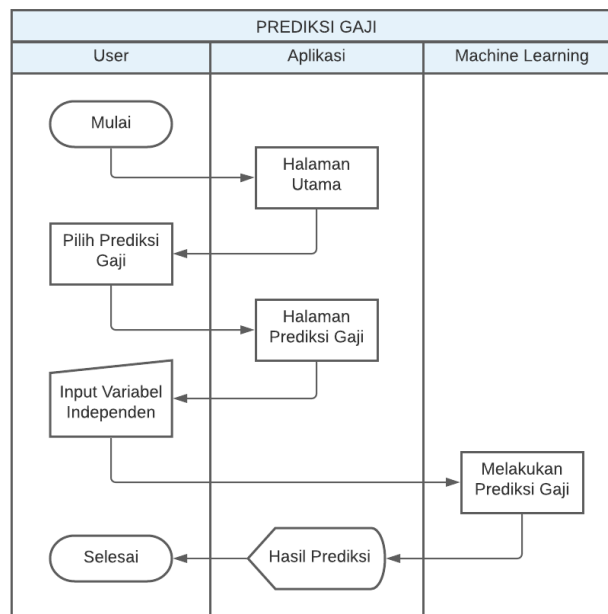
Kode diatas merupakan tampilan dari halaman hasil prediksi gaji. Terdapat form yang berisikan inputan *username* dan *password*. Selain itu tedapat hasil dari perhitungan prediksi gaji.

C. URL

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

Kode diatas merupakan URL untuk mengakses halaman hasil prediksi.

D. Logika Fungsi Prediksi Gaji



Gambar 4. 3 *Flowmap* Prediksi Gaji

Keterangan :

1. Aplikasi menampilkan halaman utama.
2. User memilih menu prediksi gaji.
3. Aplikasi menampilkan halaman prediksi gaji.
4. User menginputkan variabel independen pada form di halaman prediksi gaji.
5. Aplikasi melakukan proses perhitungan prediksi gaji.
6. Aplikasi menampilkan hasil prediksi.
7. Selesai.

4.1.5 Implementasi Proses CRUD Data Pegawai

1. Implementasi Menampilkan Data

A. 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.

B. 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.

C. URL

```

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

```

Kode diatas merupakan URL untuk mengakses halaman data pegawai.

2. Implementasi Proses Tambah Data Pegawai

A. 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.

B. 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 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.

C. URL

```

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

```

Kode diatas merupakan URL untuk mengakses halaman tambah pegawai.

3. Implementasi Proses Edit Data Pegawai

A. 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.

B. 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>
    <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 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 terdapat button simpan untuk proses menyimpan data yang telah diedit dan button batal untuk mereset data.

C. 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.

4. Implementasi Proses Hapus Data Pegawai

A. Views

```
def delete(request, id):  
    pegawai = Pegawai.objects.get(id=id)  
    pegawai.delete()  
    return redirect("/view")
```

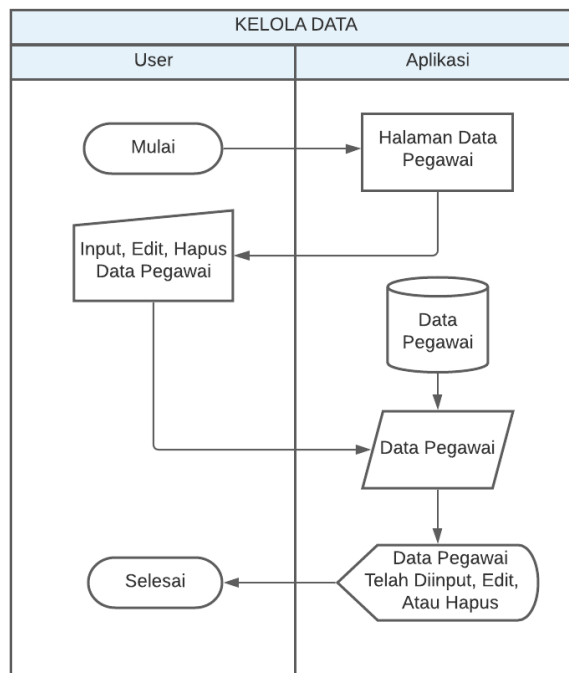
Fungsi delete digunakan untuk melakukan fungsi hapus data. Jika data berhasil terhapus maka hasil data yang terhapus akan dialihkan ke halaman data pegawai.

B. URL

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

Kode diatas merupakan URL untuk melakukan proses hapus data.

5. Logika Fungsi CRUD



Gambar 4. 4 Flowmap Prediksi Gaji

Keterangan :

1. Aplikasi menampilkan halaman data pegawai.
2. User melakukan kelola data seperti insert, edit, dan hapus data pegawai.
3. Aplikasi melakukan proses kelola data pegawai yang dilakukan.
4. Aplikasi mengirim data yang dikelola ke database data pegawai.
5. Aplikasi menampilkan data hasil insert, edit, dan hapus data pegawai.
6. Selesai

4.1.6 Implementasi Proses Menampilkan Visualisasi

A. 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.

B. 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>

```

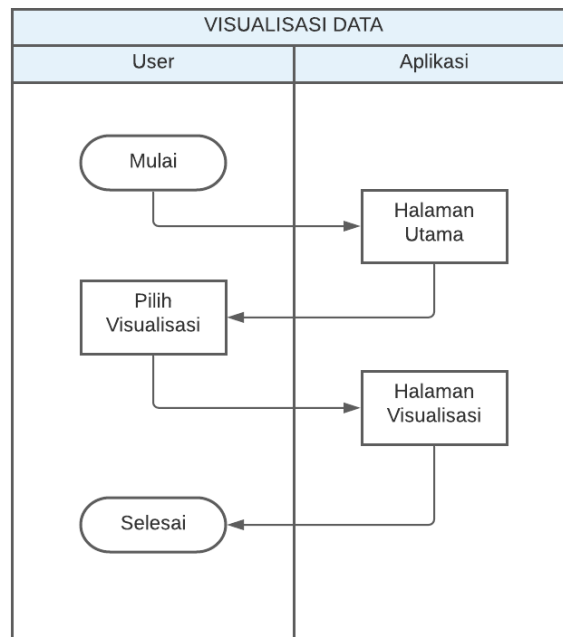
Kode diatas merupakan tampilan dari halaman visualisasi. Visualisasi tersebut dibuat dari platform Tableau yang telah dirancang sebelumnya.

C. URL

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

Kode diatas merupakan URL untuk mengakses halaman visualisasi.

D. Logika Fungsi Menampilkan Visualisasi Data



Gambar 4. 5 *Flowmap* Menampilkan Visualisasi Data

Keterangan :

4. Aplikasi menampilkan halaman utama.
5. User memilih menu visualisasi.
6. Aplikasi menampilkan halaman visualisasi.
7. Selesai.

4.2 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.

4.2.1 Himpunan Data

Pada tahap ini, hal yang dilakukan adalah 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 yang 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 yang 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.

A. Visualisasi Data Training

```
[22] %matplotlib inline

plt.style.use('ggplot')

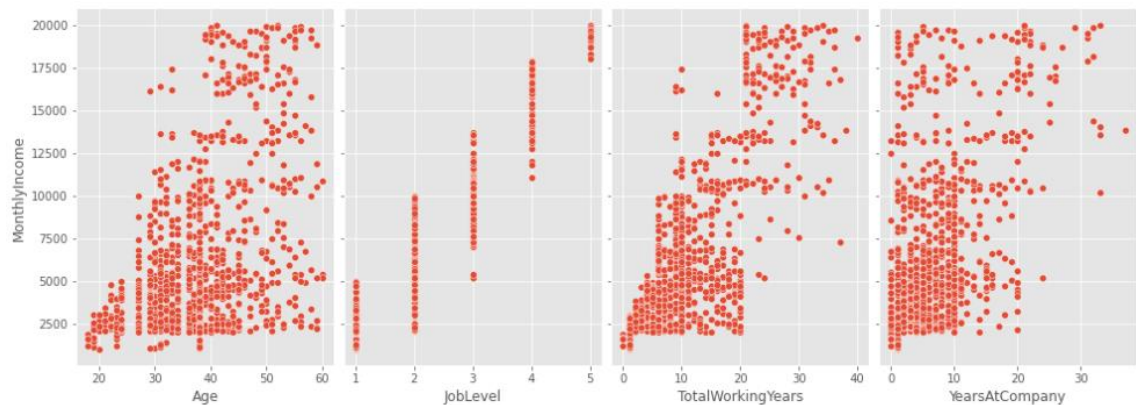
plt.rcParams['figure.figsize'] = (12,8)

[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. 6 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.

4.2.2 Proses Data Mining & Pengetahuan

Pada tahapan Proses Data Mining hal yang dilakukan adalah memilih metode yang sesuai dengan karakter data yang dikenal dengan istilah Modelling. Pada model ini digunakan Proses Data Mining Prediction. Pada tahapan Pengetahuan hal yang dilakukan adalah 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 MaritalStatus

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', 'MaritalSta
tus', 'MonthlyRate', 'NumCompaniesWorked', 'Over18', 'OverTime', 'PercentS
alaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StandardHo
urs', 'StockOptionLevel', 'TrainingTimesLastYear', 'WorkLifeBalance', 'Ye
arsInCurrentRole', '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 penambahan 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 menggunakan model *machine learning*. Kemudian buat kolom baru yang bernama MonthlyIncome Prediction yang berisikan nilai prediksi.

Berikut persamaan umum dari model linear regresi multivariabel :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \dots + \beta_n X_n$$

β_0 adalah nilai intercept dari persamaan linear, dan $\beta_1, \beta_2, \beta_3$ sampai dengan β_n adalah konstanta dari variabel independen. Berdasarkan nilai koefisien variabel independen dan Intercept didapat, maka persamaan regresi linear multivariabel sebagai berikut :

$$Y = -1728 - 5,054X_1 + 3871,7530X_2 + 46,9405X_3 - 9,8460X_4$$

Y = Variabel Dependen yaitu MonthlyIncome

X_1 = Variabel Independen pertama yaitu Age

X_2 = Variabel Independen kedua yaitu JobLevel

X_3 = Variabel Independen ketiga yaitu TotalWorkingYears

X_4 = Variabel Independen keempat yaitu YearsAtCompany

Maka dapat disimpulkan, persamaan regresi linear multivariabel sebagai berikut :

$$\text{MonthlyIncome} = -1728 - 5,054(\text{Age}) + 3871,7530(\text{JobLevel}) + 46,9405(\text{TotalWorkingYears}) - 9,8460(\text{YearsAtCompany})$$

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

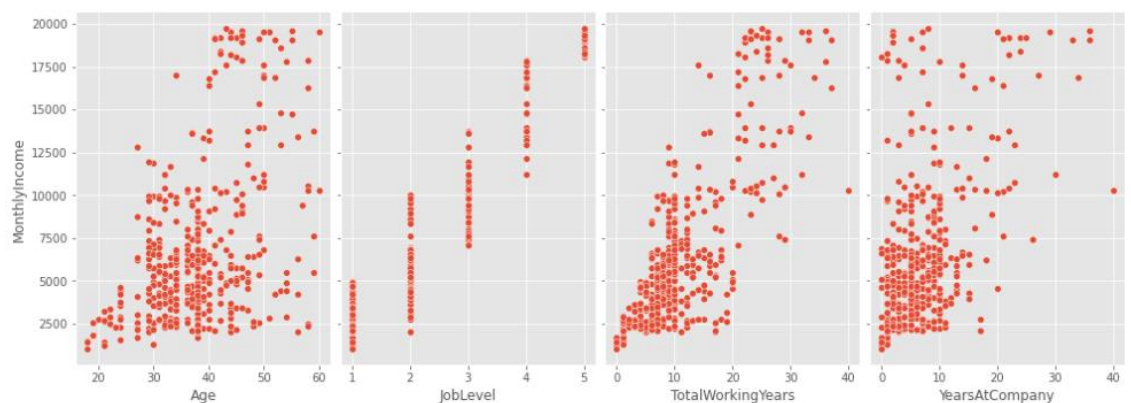
A. Visualisasi Data Testing

```
[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)
```



Gambar 4. 7 Visualisasi Data Testing

Keterangan :

- 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.

4.2.3 Evaluasi Data

Pada tahap ini hal yang dilakukan adalah melakukan evaluasi terhadap analisis model dan kinerja metode.

A. Validasi Model

```
[42] 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())
```

```

=====
                    OLS Regression Results
=====
Dep. Variable:      MonthlyIncome    R-squared:      0.909
Model:              OLS              Adj. R-squared: 0.909
Method:             Least Squares    F-statistic:    2571.
Date:               Mon, 10 Jan 2022  Prob (F-statistic): 0.00
Time:               10:58:11          Log-Likelihood: -8944.9
No. Observations:   1029              AIC:             1.790e+04
Df Residuals:       1024              BIC:             1.792e+04
Df Model:           4
Covariance Type:    nonrobust
=====
                    coef    std err          t      P>|t|      [0.025      0.975]
=====
const             -1728.5202    230.587     -7.496     0.000    -2180.998    -1276.043
Age                -5.0543         6.905     -0.732     0.464     -18.605         8.496
JobLevel           3871.7530         65.635    58.989     0.000     3742.958     4000.548
TotalWorkingYears  46.9406         11.733     4.001     0.000      23.917      69.965
YearsAtCompany     -9.8460          9.767     -1.008     0.314     -29.012         9.320
=====
Omnibus:             12.798    Durbin-Watson:      2.069
Prob(Omnibus):        0.002    Jarque-Bera (JB):    15.262
Skew:                 -0.182    Prob(JB):            0.000485
Kurtosis:              3.472    Cond. No.             213.
=====

```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Gambar 4. 8 Model OLS

Keterangan :

- Line 42 : Membuat variabel x yang berisikan atribut Age, JobLevel, TotalWorkingYears dan YearsAtCompany. Variabel X digunakan untuk melihat validasi model *machine learning* dengan model OLS.

B. Uji F (ANOVA)

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

```
[43] print('F-statistic:', olsmod.fvalue)

      print('Probability of observing value at least as high as
      F-statistic:', olsmod.f_pvalue)
```

```
F-statistic: 2570.622889791836
Probability of observing value at least as high as F-statistic: 0.0
```

Keterangan :

- Line 43 : Untuk menampilkan nilai F-statistik dan P-value dari model *machine learning* dengan model OLS.

a. 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).

b. F_s

F_s adalah hasil akhir dari analisis ANOVA. Nilai F_s ini yang dikenal dengan F hitung dalam pengujian hipotesa dibandingkan dengan nilai p-value. Jika $F_s > P\text{-value}$, maka dapat dinyatakan bahwa secara simultan (bersama-sama) variabel dependen dan variabel independen berpengaruh signifikan terhadap permintaan. Hipotesa yang didapat dari tabel ANOVA di atas adalah :

H_0 = Variabel independen secara simultan bukan penjelas yg signifikan terhadap variabel dependen (Model tidak cocok).

H_1 = Variabel independen secara simultan merupakan penjelas yang signifikan terhadap variabel dependen (Model cocok).

Berdasarkan tabel diatas, dapat diketahui bahwa $F_s > P\text{-value}$, yang artinya hipotesa yang dapat diambil adalah terima H_1 dan tolak H_0 . Dapat dikatakan, variabel Independen (Age, JobLevel, TotalWorkingYears, YearsAtCompany) dan MonthlyIncome berpengaruh signifikan terhadap permintaan. Pada taraf signifikansi 5% (0,05), H_0 ditolak karena nilai probabilitasnya yaitu 0,00 yang berarti dibawah dari 5%. Maka dapat disimpulkan, model yang dipakai cocok.

C. Uji-t

Statistik-t adalah 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 yang dibuat. Selain itu, uji-t juga bisa mendapatkan nilai p untuk seluruh variabel dengan memanggil .pvalues atribut pada model.

```
[44] print(olsmod.pvalues)
```

```

const          1.419659e-13
Age            4.643777e-01
JobLevel       0.000000e+00
TotalWorkingYears 6.771641e-05
YearsAtCompany 3.136608e-01
dtype: float64

```

Gambar 4. 9 Nilai P-Values Dari Variabel Independen

Keterangan :

- Line 44 : Untuk menampilkan nilai p-values dari variabel independen.

Hipotesa yang dapat diambil adalah :

H_0 = Variabel independen tidak berpengaruh signifikan

H_1 = Variabel independen berpengaruh signifikan.

$\alpha = 0,05$ (Tarf signifikansi)

Berdasarkan uji-t dapat diambil hipotesa sebagai berikut :

- Nilai variabel X_1 (Age) berada di atas taraf signifikansi yang berarti terima H_1 .
- Nilai variabel X_2 (JobLevel) berada di bawah taraf signifikansi yang berarti terima H_0 .
- Nilai variabel X_3 (TotalWorkingYears) di bawah taraf signifikansi yang berarti terima H_0 .
- Nilai variabel X_4 (YearsAtCompany) di atas taraf signifikansi yang berarti terima H_1 .

Berdasarkan hipotesa di atas, dapat disimpulkan variabel independen JobLevel dan TotalWorkingYears adalah variabel yang tidak mempengaruhi variabel dependen. Sedangkan variabel independen Age dan YearsAtCompany adalah variabel yang mempengaruhi variabel dependen.

D. 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 baik model tersebut.

OLS Regression Results			
Dep. Variable:	MonthlyIncome	R-squared:	0.909
Model:	OLS	Adj. R-squared:	0.909
Method:	Least Squares	F-statistic:	2571.
Date:	Mon, 10 Jan 2022	Prob (F-statistic):	0.00
Time:	10:58:11	Log-Likelihood:	-8944.9
No. Observations:	1029	AIC:	1.790e+04
Df Residuals:	1024	BIC:	1.792e+04
Df Model:	4		
Covariance Type:	nonrobust		

Gambar 4. 10 Model OLS

Berdasarkan tabel ANOVA, nilai koefisien determinasi (R-Square) adalah 0,909 atau 90,9%. Maka, MonthlyIncome dipengaruhi oleh faktor Age dan YearsAtCompany sebesar 0,909 atau 90,9%. Nilai sisa dari koefisien determinasi adalah 0.091 atau 9,1% dipengaruhi oleh faktor lain yang tidak diketahui.

E. Pengujian Asumsi

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

```
[45] 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 45 : 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.

a. 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.

```
[46] # 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

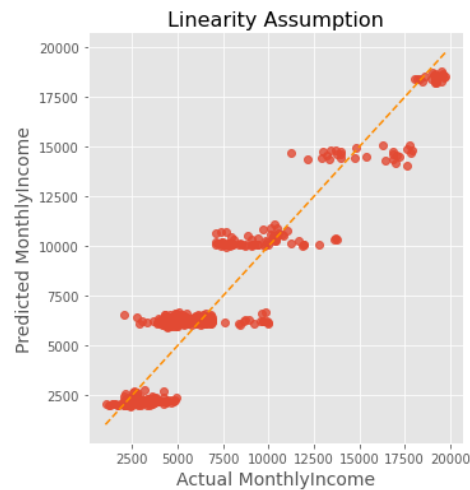
color='darkorange', linestyle='--')

plt.ylabel('Predicted MonthlyIncome', fontsize=14)

plt.xlabel('Actual MonthlyIncome', fontsize=14)

plt.title('Linearity Assumption', fontsize=16)

plt.show()
```



Gambar 4. 11 Grafik Asumsi Linear

Keterangan :

- Line 46 : Untuk menampilkan grafik linearitas dari perbandingan nilai MonthlyIncome dengan MonthlyIncomePrediction.

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

b. 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.

```
[47] 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

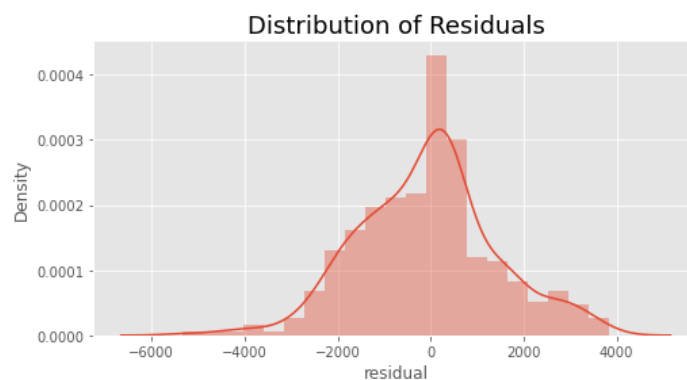
if p_value < 0.05:

    print('Residuals are not normally distributed')

else:

    print('Residuals are normally distributed')

```



Gambar 4. 12 Grafik Distribusi Residual

Keterangan :

- Line 47 : Mengimport fungsi normal_ad untuk mencari nilai normalitas. Kemudian dibuat grafik batang distribusi dari residual data.

Dapat diketahui hipotesa sebagai berikut :

H0 = Residual terdistribusi normal.

H1 = Residual terdistribusi secara tidak normal.

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 yaitu 0,05, yang berarti tolak H0 terima H1 atau dapat dikatakan residual terdistribusi secara tidak normal. Sehingga disimpulkan asumsi normalitas terpenuhi.

c. Multikolinieritas

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

```
[48] 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')

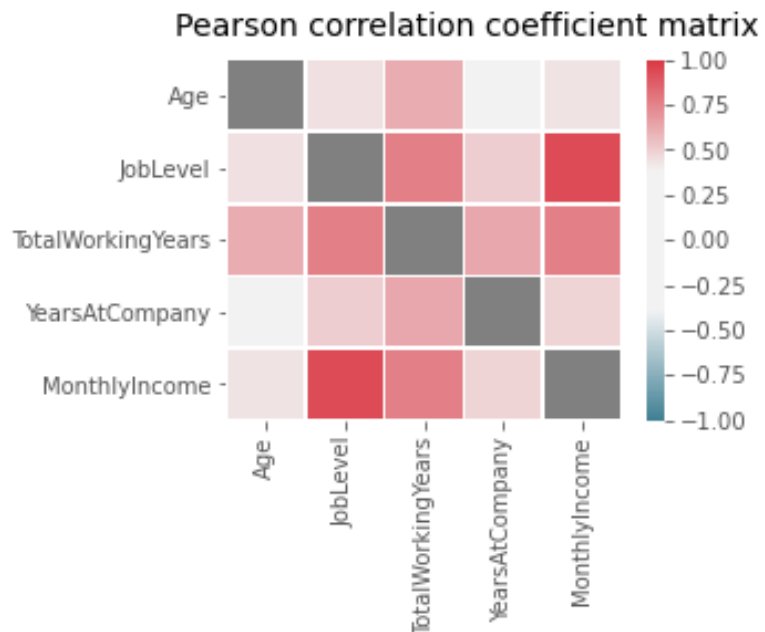

# Draw the heatmap with the mask and correct
aspect ratio

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',
labels=10)

# fig.tight_layout()
```



Gambar 4. 13 Tabel Matriks Pearson Korelasi

Keterangan :

- Line 48 : Untuk mencari nilai korelasi antar variabel dan membuat tabel matriks korelasi.

Masalah multikolinearitas itu muncul jika terdapat hubungan yang sempurna pada satu ada lebih variabel independen dalam model. Pengujian multikolinearitas dapat dilakukan dengan melihat nilai koefisien P-if nya.

```
[49] from patsy import dmatrices

      from statsmodels.stats.outliers_influence import
      variance_inflation_factor

      #find design matrix for linear regression model using
      'rating' as response variable

      y, X = dmatrices('MonthlyIncome ~
      Age+JobLevel+TotalWorkingYears+YearsAtCompany',
      data=vis_test, return_type='dataframe')
```

```
#calculate VIF for each explanatory variable

vif = pd.DataFrame()

vif['VIF'] = [variance_inflation_factor(X.values, i)
for i in range(X.shape[1])]

vif['variable'] = X.columns


#view VIF for each explanatory variable

vif
```

	VIF	variable
0	28.655370	Intercept
1	1.690786	Age
2	2.489052	JobLevel
3	4.140803	TotalWorkingYears
4	1.739893	YearsAtCompany

Gambar 4. 14 Tabel VIF

Keterangan :

- Line 49 : Untuk mencari nilai VIF variabel independen dan VIF intersep.

Berdasarkan gambar di atas dapat dilihat nilai variabel Age, JobLevel, TotalWorkingYears, YearsAtCompany memiliki nilai kurang dari 10 sehingga dengan menggunakan tingkat signifikansi sebesar 0,05 dapat disimpulkan bahwa pada data tersebut tidak terdapat multikolinearitas pada variabel-variabel prediktor.

d. 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. Autokorelasi dapat dideteksi dengan melakukan uji Durbin-Watson untuk menentukan apakah ada korelasi positif atau negative. Pada langkah ini akan dilakukan perhitungan skor Durbin-Watson menggunakan `durbin_watson()` fungsi dari `statsmodel` yang dibuat, kemudian menilainya 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.

e. Homoskedastisitas

Dari hasil di atas, ini mengasumsikan homoskedastisitas, yang merupakan varian yang sama dalam istilah kesalahan. Heteroskedastisitas/pelanggaran homoskedastisitas terjadi ketika model tidak memiliki varian genap di seluruh istilah kesalahan. Untuk mendeteksi homoskedastisitas, dapat dilakukan berupa memplot residual 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)

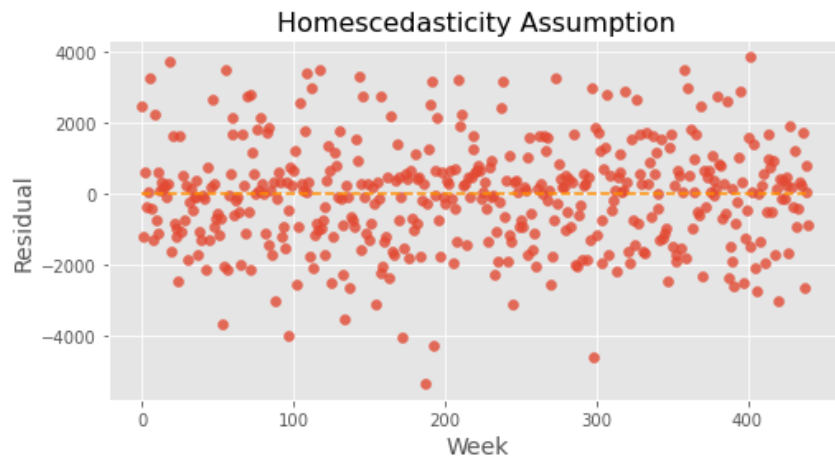
plt.plot(np.repeat(0, len(df_test_new2.index)+2),
         color='darkorange', linestyle='--')

plt.ylabel('Residual', fontsize=14)

plt.xlabel('Week', fontsize=14)

plt.title('Homoscedasticity Assumption', fontsize=16)

plt.show()
```



Gambar 4. 15 Homoskedastisitas

Keterangan :

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

Dari grafik scatterplot di atas, terlihat titik-titik menyebar secara acak, serta tersebar baik di atas maupun di bawah angka 0 (nol) pada sumbu Y. Maka dapat diambil kesimpulan bahwa tidak terdapat gejala heteroskedastisitas pada model regresi yang digunakan.

F. 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`.

4.3 Pengujian dan Hasil Pengujian

Pada bagian pengujian dan hasil pengujian aplikasi, dipaparkan pengujian pada setiap proses berjalannya sistem yang telah dibangun. Dalam melakukan proses pengujian ini digunakan extension Selenium IDE. Berikut adalah pengujian dan hasil pengujian sistem :

4.3.1 Pengujian dan Hasil Pengujian Proses *Login Admin*

	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/	
2	✓ set window size	1382x784	
3	✓ click	linkText=Login	
4	✓ type	id=id_username	abahchan
5	✓ click	id=id_password	
6	✓ type	id=id_password	abah211200
7	✓ click	css=button	

Gambar 4. 16 Pengujian dan Hasil Pengujian Proses *Login Admin*

Pada gambar diatas menunjukkan bahwa pengujian proses *login* admin pada aplikasi prediksi gaji pegawai tidak mengalami error yang ditandai deklarasi command, target dan *value*-nya berwarna hijau. Artinya proses *login* admin pada aplikasi ini telah berjalan dengan baik. Berikut disertakan tabel hasil pengujian proses *login* admin :

Tabel 4. 1 Hasil pengujian proses *login* admin

No.	Proses	Status
1.	open on http://127.0.0.1:8000/	OK (20:21:04)
2.	setWindowSize on 1382x784	OK (20:21:04)
3.	click on linkText= <i>Login</i>	OK (20:21:04)
4.	type on id=id_username with value abahchan	OK (20:21:07)
5.	click on id=id_password	OK (20:21:08)
6.	type on id=id_password with value abah211200	OK (20:21:09)
7.	click on css=button	OK (20:21:10)

8.	'Login Admin' completed successfully20:21:10	20:21:10
----	--	----------

4.3.2 Pengujian dan Hasil Pengujian Proses Pengolahan Data

	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/	
2	✓ set window size	1382x784	
3	✓ click	linkText=Login	
4	✓ type	id=id_username	abahchan
5	✓ click	id=id_password	
6	✓ type	id=id_password	abah211200
7	✓ click	css=button	
8	✓ click	linkText=Data Pegawai	
9	✓ double click	linkText=Data Pegawai	
10	✓ click	css= btn > b	
11	✓ click	id=id_Age	
12	✓ type	id=id_Age	32
13	✓ click	id=id_JobLevel	

Gambar 4. 17 Pengujian dan Hasil Pengujian Proses Pengolahan Data

Pada gambar diatas menunjukkan bahwa pengujian proses pengolahan data pada aplikasi prediksi gaji pegawai tidak mengalami error yang ditandai deklarasi command, target dan *value*-nya berwarna hijau. Artinya proses pengolahan data pada aplikasi ini telah berjalan dengan baik. Berikut disertakan tabel hasil pengujian proses pengolahan data :

Tabel 4. 2 Hasil pengujian proses pengolahan data

No.	Proses	Status
1.	open on http://127.0.0.1:8000/	OK (21:33:17)
2.	setWindowSize on 1382x784	OK (21:33:18)
3.	click on linkText= <i>Login</i>	OK (21:33:19)
4.	type on id=id_username with value abahchan	OK (21:33:21)
5.	click on id=id_password	OK (21:33:22)
6.	type on id=id_password with value abah211200	OK (21:33:23)
7.	click on css=button	OK (21:33:24)

8.	click on linkText=Data Pegawai	OK (21:33:26)
9.	doubleClick on linkText=Data Pegawai	OK (21:33:28)
10.	click on css=.btn > b	OK (21:33:30)
11.	click on id=id_Age	OK (21:33:31)
12.	type on id=id_Age with value 32	OK (21:33:32)
13.	click on id=id_JobLevel	OK (21:33:32)
14.	type on id=id_JobLevel with value 2	OK (21:33:33)
15.	click on id=id_MonthlyIncome	OK (21:33:34)
16.	click on id=id_TotalWorkingYears	OK (21:33:35)
17.	type on id=id_TotalWorkingYears with value 5	OK (21:33:36)
18.	click on id=id_YearsAtCompany	OK (21:33:37)
19.	type on id=id_YearsAtCompany with value 2	OK (21:33:37)
20.	click on id=id_MonthlyIncome	OK (21:33:37)
21.	type on id=id_MonthlyIncome with value 5436	OK (21:33:37)
22.	click on css=.btn-primary	OK (21:33:38)
23.	'Kelola Data' completed successfully20	21:33:38

4.3.3 Pengujian dan Hasil Pengujian Proses Prediksi Gaji

	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/	
2	✓ set window size	1382x784	
3	✓ click	linkText=Login	
4	✓ type	id=id_username	abahchan
5	✓ click	id=id_password	
6	✓ type	id=id_password	abah211200
7	✓ click	css=button	
8	✓ click	linkText=Prediksi	
9	✓ click	id=Age	
10	✓ type	id=Age	32
11	✓ click	id=JobLevel	
12	✓ type	id=JobLevel	2
13	✓ click	id=YearsAtCompany	
14	✓ type	id=YearsAtCompany	5

Gambar 4. 18 Pengujian dan Hasil Pengujian Proses Prediksi Gaji

Pada gambar diatas menunjukkan bahwa pengujian proses prediksi gaji pada aplikasi prediksi gaji pegawai tidak mengalami error yang ditandai deklarasi command, target dan *value*-nya berwarna hijau. Artinya proses prediksi gaji pada aplikasi ini telah berjalan dengan baik. Berikut disertakan tabel hasil pengujian proses prediksi gaji :

Tabel 4. 3 Hasil pengujian proses prediksi gaji

No.	Proses	Status
1.	open on http://127.0.0.1:8000/	OK (21:39:03)
2.	setWindowSize on 1382x784	OK (21:39:04)
3.	click on linkText= <i>Login</i>	OK (21:39:04)
4.	type on id=id_username with value abahchan	OK (21:39:06)
5.	click on id=id_password	OK (21:39:07)
6.	type on id=id_password with value abah211200	OK (21:39:07)
7.	click on css=button	OK (21:39:07)
8.	click on linkText=Prediksi	OK (21:39:07)
9.	type on id=Age with value 32	OK (21:39:12)
10.	click on id=JobLevel	OK (21:39:13)
11.	type on id=JobLevel with value 2	OK (21:39:14)
12.	click on id=YearsAtCompany	OK (21:39:15)
13.	type on id=YearsAtCompany with value 5	OK (21:39:16)
14.	click on id=TotalWorkingYears	OK (21:39:17)
15.	type on id=TotalWorkingYears with value 2	OK (21:39:18)
16.	click on css=.btn-block	OK (21:39:19)
17.	'Prediksi Gaji' completed successfully	21:39:20

4.3.4 Pengujian dan Hasil Pengujian Proses Visualisasi

	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/	
2	✓ set window size	1382x784	
3	✓ click	linkText=Login	
4	✓ type	id=id_username	abahchan
5	✓ click	id=id_password	
6	✓ type	id=id_password	abah211200
7	✓ click	css=button	
8	✓ click	linkText=Visualisasi	

Gambar 4. 19 Pengujian dan Hasil Pengujian Proses Visualisasi

Pada gambar diatas menunjukkan bahwa pengujian proses visualisasi pada aplikasi prediksi gaji pegawai tidak mengalami error yang ditandai deklarasi command, target dan *value*-nya berwarna hijau. Artinya proses visualisasi pada aplikasi ini telah berjalan dengan baik. Berikut disertakan tabel hasil pengujian proses visualisasi :

Tabel 4. 4 Hasil pengujian proses visualisasi

No.	Proses	Status
1.	open on http://127.0.0.1:8000/	OK (21:44:48)
2.	setWindowSize on 1382x784	OK (21:44:48)
3.	click on linkText= <i>Login</i>	OK (21:44:48)
4.	type on id=id_username with value abahchan	OK (21:44:50)
5.	click on id=id_password	OK (21:44:51)
6.	type on id=id_password with value abah211200	OK (21:44:52)
7.	click on css=button	OK (21:44:53)
8.	click on linkText=Visualisasi	OK (21:44:54)
9.	'Visualisasi Data' completed successfully	21:44:56

4.3.5 Pengujian dan Hasil Pengujian Proses *Logout*

	Command	Target	Value
1	✓ <i>open</i>	http://127.0.0.1:8000/	
2	✓ <i>set window size</i>	1366x768	
3	✓ <i>click</i>	linkText=Login	
4	✓ <i>type</i>	id=id_username	abahchan
5	✓ <i>click</i>	id=id_password	
6	✓ <i>type</i>	id=id_password	abah211200
7	✓ <i>click</i>	css=button	
8	✓ <i>click</i>	linkText=Logout	

Gambar 4. 20 Pengujian dan Hasil Pengujian Proses *Logout*

Pada gambar diatas menunjukkan bahwa pengujian proses *logout* pada aplikasi prediksi gaji pegawai tidak mengalami error yang ditandai deklarasi command, target dan *value*-nya berwarna hijau. Artinya proses *logout* pada aplikasi ini telah berjalan dengan baik. Berikut disertakan tabel hasil pengujian proses *logout* :

Tabel 4. 5 Hasil pengujian proses *logout*

No.	Proses	Status
1.	open on http://127.0.0.1:8000/	OK (11:40:46)
2.	setWindowSize on 1382x784	OK (11:40:46)
3.	click on linkText= <i>Login</i>	OK (11:40:46)
4.	type on id=id_username with value abahchan	OK (11:40:49)
5.	click on id=id_password	OK (11:40:50)
6.	type on id=id_password with value abah211200	OK (11:40:51)
7.	click on css=button	OK (11:40:52)
8.	click on linkText=Visualisasi	OK (11:40:53)
9.	'Visualisasi Data' completed successfully	11:40:56

BAB V

KESIMPULAN DAN SARAN

5.1 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 model OLS kami mendapatkan nilai akurasi sebesar 0,909. Akurasi tersebut merupakan nilai akurasi yang baik, sehingga dapat dikatakan model *machine learning* dapat berperforma baik untuk memprediksi gaji.
4. Berdasarkan uji validitas, nilai akurasi 0,909 menunjukkan bahwa MonthlyIncome dipengaruhi oleh faktor independen (Age, YearsAtCompany) sebesar 0,909 atau 90,9%. Nilai sisa dari akurasi tersebut adalah 0,091 atau 9,1% yang artinya MonthlyIncome dipengaruhi oleh faktor lain yang tidak diketahui sebesar 9,1%.

5. 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.

5.2 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. Sumber data yang digunakan kurang maksimal. Pada penelitian ini, hanya didapatkan real yang berasal dari Kaggle. Diharapkan kedepannya dapat menggunakan data real langsung dari perusahaan.

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 Scrapping Dan *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 Informatika*, vol. 2, no. 1, Mar. 2019, doi : <http://jurnal.umm.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.imp.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

Code Coverage Folder Prediksi

No	Halaman	Running Statement	Missing Statement	Excluded	Percent
1	Views.py	4	65	0	6%
2	Urls.py	2	2	0	50%
3	Tests.py	1	0	0	100%
4	Models.py	8	2	0	80%
5	Forms.py	2	4	0	33%
6	Apps.py	4	0	0	100%
7	Admin.py	1	0	0	100%

Code Coverage Folder PrediksiGaji

No	Halaman	Running Statement	Missing Statement	Excluded	Percent
1	Asgi.py	4	0	0	100%
2	Settings.py	21	0	0	100%
3	Urls.py	3	3	0	50%
4	Wsgi.py	4	0	0	100%

GLOSARIUM

Glosarium Non-Teknis

No	Istilah	Keterangan
1	OOP	Object Oriented Programing adalah paradigma pemrograman yang berorientasi objek
2	Linear Regresi Berganda	Linear Regresi Berganda adalah metode statistik yang digunakan untuk melakukan peramalan dengan banyak faktor.

3	Linear Regresi Sederhana	Linear Regresi Sederhana adalah metode statistic yang digunakan untuk melakukan peramalan dengan 2 faktor.
---	--------------------------	--

Glosarium Teknis

No	Istilah	Keterangan
Python		
1	import	Digunakan untuk mengimpor sebagian isi modul serta bukan keseluruhan isi modul.
2	fields	Sebuah ruang disebut GUI window yang berfungsi untuk memasukkan sebuah teks atau angka.
3	class	Sebuah <i>blueprint</i> (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

TABEL - TABEL

1. Tabel Definisi Aktor

No	Aktor	Deskripsi
1	Admin	<ul style="list-style-type: none"> • <i>Login</i> • Melakukan Prediksi Gaji • Melakukan Pengolahan Data • Melakukan Visualisasi Data • <i>Logout</i>

2. Tabel Definisi *Use case*

No	Aktor	Deskripsi
1	<i>Machine learning</i>	Merupakan aktivitas interaksi antara manusia dengan mesin. Dalam hal ini, admin membuat model prediksi <i>Machine learning</i> .
2	<i>Login</i>	Merupakan aktivitas <i>login</i> yang dilakukan oleh admin. Sebelum admin masuk ke aplikasi ia harus melakukan <i>login</i> terlebih dahulu.
3	Registrasi	Merupakan aktivitas registrasi yang dilakukan oleh admin. Apabila admin belum memiliki akun, maka admin harus melakukan registrasi terlebih dahulu.
4	Melakukan Prediksi Gaji	Merupakan aktivitas memprediksi gaji karyawan yang dilakukan oleh admin dengan parameter lama bekerja seorang karyawan.

5	Melakukan Pengolahan Data	Merupakan aktivitas insert, read, update dan delete data karyawan yang dilakukan oleh admin.
6	Menampilkan Visualisasi Data	Merupakan aktivitas visualisasi grafik data karyawan yang dilakukan oleh admin.
7	<i>Logout</i>	Merupakan aktivitas <i>logout</i> yang dilakukan oleh admin. Admin dapat keluar dari aplikasi jika telah selesai melakukan pekerjaan pada aplikasi.

3. Tabel Skenario *Use case Machine learning*

Identifikasi	
Nomor	1
Nama	<i>Machine learning</i>
Tujuan	Pembuatan model prediksi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Membuat model prediksi 	<ul style="list-style-type: none"> Melakukan olah data gaji karyawan
Kondisi Akhir	Masuk pada aplikasi

4. Tabel Skenario *Use case Login*

Identifikasi	
Nomor	1
Nama	Registrasi
Tujuan	Membuat akun admin
Deskripsi	

Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Input <i>username</i>, dan <i>password</i> 	<ul style="list-style-type: none"> Menyimpan data admin atau user
Kondisi Akhir	Masuk pada halaman <i>login</i>

5. Tabel Skenario *Use case* Registrasi

Identifikasi	
Nomor	1
Nama	Registrasi
Tujuan	Membuat akun admin
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Input <i>username</i>, dan <i>password</i> 	<ul style="list-style-type: none"> Menyimpan data admin atau user
Kondisi Akhir	Masuk pada halaman <i>login</i>

6. Tabel 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
<ul style="list-style-type: none"> Menginput Data 	<ul style="list-style-type: none"> Melakukan Prediksi Gaji

Kondisi Akhir	Admi dapat memprediksi gaji
----------------------	-----------------------------

7. Tabel 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
<ul style="list-style-type: none"> Input data karyawan 	<ul style="list-style-type: none"> Menampilkan grafik visualisasi data karyawan
Kondisi Akhir	Admin dapat memvisualisasikan data

8. Tabel 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
<ul style="list-style-type: none"> Insert, read, update, dan delete data karyawan. 	<ul style="list-style-type: none"> Mengubah data karyawan yang diubah oleh admin
Kondisi Akhir	Admin dapat mengelola data

9. Tabel Skenario *Use case Logout*

Identifikasi	
Nomor	1
Nama	<i>Logout</i>
Tujuan	Keluar dari aplikasi
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
<ul style="list-style-type: none"> Menekan tombol <i>logout</i> 	<ul style="list-style-type: none"> Melakukan proses keluar dari aplikasi
Kondisi Akhir	Keluar dari aplikasi