

— MEMBUAT — **APLIKASI WEB PREDIKSI GAJI** **SEMUDAH MENANAK NASI**



Belajar memahami *machine learning* dan bagaimana cara mengimplementasikan dalam bentuk *web base* secara tuntas.

**SYAFRIAL FACHRI PANE, AMRI YANUAR, NUR ALAMSYAH, BACHTIAR RAMADHAN,
NUR TRI RAMADHANTI ADININGRUM**

MEMBUAT APLIKASI WEB PREDIKSI GAJI SEMUDAH MENANAK NASI

Syaafrial Fachri Pane
Amri Yanuar
Nur Alamsyah
Bachtiar Ramadhan
Nur Tri Ramadhanti Adiningrum



bukupedia
Penerbit Buku Pedia

Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi

Penulis:

Syafrial Fachri Pane
Amri Yanuar
Nur Alamsyah
Bachtiar Ramadhan
Nur Tri Ramadhanti Adiningrum

ISBN :

Editor:

Roni Habibi

Penyunting:

Roni Habibi

Desain sampul dan Tata letak:

Bachtiar Ramadhan

Penerbit:

Penerbit Buku Pedia

Redaksi:

Athena Residence Blok.E No. 1, Desa Ciwaruga,
Kec. Parongpong, Kab. Bandung Barat 40559
Tel. 628-775-2000-300
Email : penerbit@bukupedia.co.id

Distributor:

Informatics Research Center
Jl. Sariasih No. 54
Bandung 40151
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2022

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan caraapapun tanpa ijin
tertulis dari penerbit

Kata Pengantar

Puji syukur kami haturkan kehadiran Allah SWT. yang telah melimpahkan kekuatan dan kesabaran, karena berkat rahmat dan hidayah-Nya, sehingga penulisan buku dengan judul “Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi” yang telah lama dipersiapkan dapat diselesaikan dengan baik. Penyusunan buku ini mendapatkan banyak masukan dan motivasi dari berbagai pihak terutama rekan-rekan yang telah berkolaborasi dalam penulisan buku ini. Buku ini bertujuan untuk memberikan wawasan tentang penerapan *machine learning* pada bidang informatika terutama dalam memprediksi gaji karyawan. Maka dari itu, pada kesempatan ini kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak atas keikhlasannya dalam melakukan proses penyusunan buku ini hingga selesai sesuai rencana serta sesuai dengan kaedah-kaedah penulisan karya ilmiah. Penulis juga menghantarkan terima kasih kepada orang tua dan keluarga yang sangat kami cintai, yang selalu memberikan motivasi dan selalu berdoa atas setiap langkah kebenaran yang kami lakukan. Semoga buku ini kelak bermanfaat untuk penulis maupun siapa saja yang membaca buku ini.

Penulis menyadari bahwa dalam penyusunan buku ini terdapat banyak kekurangan dan jauh dari sempurna serta menyadari bahwa buku ini hanya merupakan sebagian kecil dari luasnya pengetahuan dalam algoritma regresi linier berganda. Oleh karena itu, segala bentuk masukan baik saran ataupun kritik yang berharga dari berbagai pihak untuk membangun kesempurnaan buku ini sangat kami harapkan.

Bandung, 19 Agustus 2022

Penulis

Daftar Isi

KATA PENGANTAR	I
DAFTAR ISI.....	II
DAFTAR GAMBAR.....	V
DAFTAR TABEL.....	X
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	2
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Ruang Lingkup.....	3
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	5
2.1 Teori Umum	5
2.1.1 Prediksi.....	5
2.1.2 Gaji	5
2.1.3 Pegawai.....	6
2.1.4 Aplikasi	7
2.1.5 Data	7
2.1.6 <i>Unified Modeling Language (UML)</i>	7
2.2 Bahasa Pemrograman.....	8
2.2.1 Python	8
2.2.2 HTML	14
2.2.3 CSS.....	15
2.2.4 Bootstrap.....	16
2.2.5 Django.....	16
2.2.6 <i>Model Template Views (MTV)</i>	24
2.2.7 <i>Model</i>	25
2.2.8 <i>View</i>	25
2.2.9 <i>Template</i>	26
2.3 Metode Yang Digunakan.....	27
2.3.1 Teknik Pengumpulan Data	27
2.3.2 <i>Machine learning</i>	27
2.3.3 Regresi Linier	28
2.3.4 Metode Regresi Linier Berganda.....	28
2.4 <i>Software</i>	29
2.4.1 Visual Studio Code.....	29
2.4.2 XAMPP	35

2.4.3	Jupyter Notebook.....	39
2.4.4	Tableau.....	42
2.4.5	Anaconda.....	46
2.5	Penyimpanan	49
2.5.1	<i>Database</i>	49
2.5.2	MySQL	53
2.6	Tinjauan Pustaka.....	54
2.6.1	Deskripsi Metode Yang Sama.....	54
BAB III GAMBARAN OBJEK STUDI		65
3.1	Material	65
3.2	Teknologi.....	68
BAB IV METODOLOGI PENELITIAN		71
4.1	Diagram Alur Metodologi Penelitian	71
4.1.1	Diagram Alur Metodologi Penelitian Penyelesaian Masalah.....	71
4.1.2	Diagram Alur Metodologi Pengolahan Data.....	73
BAB V PEMBUATAN APLIKASI		76
5.1	Analisis.....	76
5.1.1	Analisis Sistem Yang Sedang Berjalan.....	76
5.1.2	Analisis Sistem Yang Akan Dibangun.....	78
5.2	Perancangan Sistem (UML)	80
5.2.1	<i>Use case Diagram</i>	81
5.2.2	<i>Class diagram</i>	88
5.2.3	<i>Sequence Diagram</i>	89
5.2.4	<i>Collaboration diagram</i>	97
5.2.5	<i>Activity diagram</i>	102
5.2.6	<i>Statechart Diagram</i>	110
5.2.7	<i>Component diagram</i>	115
5.2.8	<i>Deployment Diagram</i>	116
5.3	Perancangan Basis Data/ Data Sistem.....	117
5.3.1	<i>CDM (Conceptual Data Model)</i>	117
5.3.2	<i>PDM (Physical Data Model)</i>	118
5.3.3	<i>ERD (Entity Relationship Diagram)</i>	120
5.3.4	Kamus Data	121
5.4	Struktur Menu	127
5.5	Perancangan <i>User Interface</i>	128
5.5.1	<i>User Interface</i> Halaman Depan	129
5.5.2	<i>User Interface</i> Halaman <i>Login</i>	130
5.5.3	<i>User Interface</i> Halaman Registrasi.....	131
5.5.4	<i>User Interface</i> Halaman <i>Dashboard</i>	133
5.5.5	<i>User Interface</i> Halaman Prediksi.....	134
5.5.6	<i>User Interface</i> Halaman Hasil Prediksi.....	135
5.5.7	<i>User Interface</i> Halaman Data Pegawai.....	136

5.5.8	<i>User Interface</i> Halaman Tambah Data Pegawai.....	137
5.5.9	<i>User Interface</i> Halaman Edit Data Pegawai.....	138
5.5.10	<i>User Interface</i> Halaman Visualisasi	139
5.6	Implementasi Kode Aplikasi	140
5.6.1	Implementasi Package Yang Digunakan.....	140
5.6.2	Implementasi Proses <i>Login</i>	140
5.6.3	Implementasi Proses Registrasi.....	143
5.6.4	Implementasi Proses Prediksi Gaji	145
5.6.5	Implementasi Proses Hasil Prediksi Gaji	147
5.6.6	Implementasi Proses CRUD Data Pegawai	150
5.6.7	Implementasi Proses Menampilkan Visualisasi.....	160
5.6.8	Implementasi Proses <i>Logout</i>	163
5.7	Implementasi kode CSS Aplikasi	165
5.7.1	Implementasi Kode Link Bootstrap CSS.....	165
5.7.2	Implementasi Kode CSS.....	165
5.8	Halaman Antarmuka	166
5.8.1	Implementasi Halaman Depan.....	166
5.8.2	Antarmuka Halaman <i>Login</i>	167
5.8.3	Antarmuka Halaman Registrasi	167
5.8.4	Antarmuka Halaman <i>Dashboard</i>	168
5.8.5	Antarmuka Halaman Prediksi	169
5.8.6	Antarmuka Halaman Hasil Prediksi	170
5.8.7	Antarmuka Halaman Data Pegawai	170
5.8.8	Antarmuka Halaman Tambah Data Pegawai.....	171
5.8.9	Antarmuka Halaman Edit Data Pegawai	172
5.8.10	Antarmuka Halaman Visualisasi	172
5.9	Implementasi Metode Regresi Linier Berganda	173
5.9.1	Himpunan Data.....	173
5.9.2	Proses Data Mining & Pengetahuan.....	180
5.9.3	Evaluasi Data.....	187
BAB VI	KESIMPULAN DAN SARAN.....	205
6.1	Kesimpulan.....	205
6.2	Saran.....	206
DAFTAR PUSTAKA.....	207	
BIODATA PENULIS	214	

Daftar Gambar

Gambar 2.1: Logo Python	8
Gambar 2.2: Hyperlink Web Python.org	9
Gambar 2.3: Halaman Utama Web Python.....	10
Gambar 2.4: Halaman Download Python	10
Gambar 2.5: Security Warning Desktop	11
Gambar 2.6: Instalasi Python Add Python 3.10 to PATH	11
Gambar 2.7: Instalasi Pyton Install Now	12
Gambar 2.8: Setup Progress Instalasi Python	12
Gambar 2.9: Selesai Instalasi Python.....	13
Gambar 2.10: CMD Cek Hasil Instalasi Python	13
Gambar 2.11: CMD Python Siap Menerima Perintah.....	14
Gambar 2.12: Logo HTML	14
Gambar 2.13: Logo CSS	15
Gambar 2.14: Logo Bootstrap.....	16
Gambar 2.15: Logo Django.....	16
Gambar 2.16: Anaconda Prompt Administrator	17
Gambar 2.17: Pembuatan Projek Django.....	18
Gambar 2.18: Folder Projek Django YangTerbuat	18
Gambar 2.19: Masuk Folder Projek Django	19
Gambar 2.20: Membuat Folder Aplikasi Django.....	19
Gambar 2.21: Folder Aplikasi Django	20
Gambar 2.22: Menjalankan Projek Aplikasi	20
Gambar 2.23: Running Aplikasi Pada Web browser	21
Gambar 2.24: Migrate Python Django	21
Gambar 2.25: Memasukan Perintah Python Migrate Pada Anaconda Prompt.....	22
Gambar 2.26: Menambahkan Nama Aplikasi Pada File Setting.py	22
Gambar 2.27: Membuat Folder Bernama Template.....	23
Gambar 2.28: Menambahkan Library os.....	23

Gambar 2.29: Menambahkan Kode os.path.join(BASE_DIR, ‘template’)	
Pada Template.....	24
Gambar 2.30: Arsitektur MVT Framework Django.....	25
Gambar 2.31: Isi File models.py	25
Gambar 2.32: Isi File views.py	26
Gambar 2.33: Isi File index.html Pada Folder Template	26
Gambar 2.34: Logo Kaggle.....	27
Gambar 2.35: Logo Visual Studio Code	29
Gambar 2.36: Halaman Web Download Visual Studio Code	31
Gambar 2.37: Setup Visual Studio Code	31
Gambar 2.38: Memilih Lokasi Penyimpanan Visual Studio Code	32
Gambar 2.39: Memilih Start Menu Folder.....	32
Gambar 2.40: Memilih Additional Tasks.....	33
Gambar 2.41: Memulai Proses Instalasi.....	33
Gambar 2.42: Proses Instalasi Visual Studio Code.....	34
Gambar 2.43: Instalasi Visual Studio Code Selesai	34
Gambar 2.44: Logo XAMPP.....	35
Gambar 2.45: Halaman Web Download XAMPP	36
Gambar 2.46: Setup XAMPP	37
Gambar 2.47: Memilih Komponen Yang Akan Digunakan.....	37
Gambar 2.48: Lokasi File XAMPP	38
Gambar 2.49: Proses Instalasi XAMPP	38
Gambar 2.50: Logo Jupyter.....	39
Gambar 2.51: Instal virtualenv Pada CMD	40
Gambar 2.52: Meng-upgrade pip Pada Command Prompt	41
Gambar 2.53: Instal Jupyter Notebook	41
Gambar 2.54: Running Jupyter Notebook Pada Web browser	42
Gambar 2.55: Running Python Web browser	42
Gambar 2.56: Logo Tableau	42
Gambar 2.57: Portal Customer Tableau.....	44
Gambar 2.58: Penyediaan Data Penggunaan	45
Gambar 2.59: Memilih Lokasi Penempatan Tableau	45
Gambar 2.60: Logo Anaconda	46
Gambar 2.61: Setup Anaconda	47

Gambar 2.62: Memilih Lokasi Penempatan Anaconda.....	47
Gambar 2.63: Memulai Proses Instalasi Anaconda	48
Gambar 2.64: Opsional Install Aplikasi PyCharm.....	48
Gambar 2.65: Selesai Instalasi Anaconda	49
Gambar 2.66: Logo MySQL	53
Gambar 4.1: Diagram Alur Metodologi Penelitian.....	71
Gambar 4.2: Diagram Alur Metodologi Pengolahan Data.....	73
Gambar 5.1: <i>Flowmap</i> Pemrosesan Gaji.....	77
Gambar 5.2: <i>Flowmap</i> Proses <i>Login Admin</i>	78
Gambar 5.3: <i>Flowmap</i> Proses Prediksi Gaji Pegawai.....	79
Gambar 5.4: <i>Use case Diagram</i>	81
Gambar 5.5: <i>Class diagram</i>	89
Gambar 5.6: <i>Sequence diagram Login Admin</i>	90
Gambar 5.7: <i>Sequence diagram Registrasi</i>	91
Gambar 5.8: <i>Sequence diagram</i> Melakukan Prediksi Gaji	92
Gambar 5.9: <i>Sequence diagram</i> Melakukan Pengolahan Data	94
Gambar 5.10: <i>Sequence diagram</i> Melakukan Visualisasi Data	95
Gambar 5.11: <i>Sequence diagram Logout admin</i>	96
Gambar 5.12: <i>Collaboration diagram</i> Melakukan <i>Login</i>	98
Gambar 5.13: <i>Collaboration diagram</i> Registrasi.....	99
Gambar 5.14: <i>Collaboration diagram</i> Melakukan Prediksi Gaji.....	100
Gambar 5.15: <i>Collaboration diagram</i> Melakukan Pengolahan Data Pegawai	101
Gambar 5.16: <i>Collaboration diagram</i> Melakukan Visualisasi Data...	102
Gambar 5.17: <i>Activity diagram Login Admin</i>	103
Gambar 5.18: <i>Activity diagram</i> Registrasi	104
Gambar 5.19: <i>Activity diagram</i> Melakukan Pengolahan Data	105
Gambar 5.20: <i>Activity diagram</i> Melakukan Prediksi Gaji	107
Gambar 5.21: <i>Activity diagram</i> Melakukan Visualisasi.....	108
Gambar 5.22: <i>Activity diagram Logout Admin</i>	109
Gambar 5.23: <i>Statechart Diagram Login Admin</i>	110
Gambar 5.24: <i>Statechart Diagram</i> Registrasi	111
Gambar 5.25: <i>Statechart Diagram</i> Melakukan Prediksi	112
Gambar 5.26: <i>Statechart Diagram</i> Pengolahan Data.....	113

Gambar 5.27: <i>Statechart Diagram</i> Visualisasi	114
Gambar 5.28: <i>Component diagram</i>	115
Gambar 5.29: Deployment Diagram	116
Gambar 5.30: <i>Conceptual Data Model</i>	118
Gambar 5.31: Physical Data Model	119
Gambar 5.32: <i>Entity Relationship Diagram</i>	121
Gambar 5.33: Struktur Menu Aplikasi.....	128
Gambar 5.34: Rancangan <i>User Interface</i> Dari Halaman Depan	129
Gambar 5.35: Rancangan <i>User Interface</i> Dari Halaman <i>Login</i>	130
Gambar 5.36: Rancangan <i>User Interface</i> Dari Halaman Registrasi ...	131
Gambar 5. 37: Rancangan <i>User Interface</i> Dari Halaman <i>Dashboard</i>	133
Gambar 5.38: Rancangan <i>User Interface</i> Dari Halaman Prediksi	134
Gambar 5.39: Rancangan <i>User Interface</i> Dari Halaman Hasil Prediksi	135
Gambar 5.40: Rancangan <i>User Interface</i> Dari Halaman Data Pegawai	136
Gambar 5.41: Rancangan <i>User Interface</i> Dari Halaman Tambah Data Pegawai	137
Gambar 5.42: Rancangan <i>User Interface</i> Dari Halaman Edit Data Pegawai	138
Gambar 5.43: Rancangan <i>User Interface</i> Dari Halaman Visualisasi..	139
Gambar 5.44: <i>Flowmap Login</i> Aplikasi.....	142
Gambar 5.45: <i>Flowmap Registrasi</i> Aplikasi	144
Gambar 5.46: <i>Flowmap Prediksi Gaji</i>	147
Gambar 5.47: <i>Flowmap Prediksi Gaji</i>	149
Gambar 5.48: <i>Flowmap Olah Data Pegawai</i>	159
Gambar 5.49: <i>Flowmap Menampilkan Visualisasi Data</i>	163
Gambar 5.50: <i>Flowmap Logout</i>	164
Gambar 5.51: Halaman Depan Sistem	166
Gambar 5.52: Halaman <i>Login</i> Sistem	167
Gambar 5.53: Halaman Registrasi Sistem.....	167
Gambar 5.54: Halaman <i>Dashboard</i> Sistem.....	168
Gambar 5.55: Halaman Prediksi Sistem	169
Gambar 5.56: Halaman Hasil Prediksi Sistem	170

Gambar 5.57: Halaman Data Pegawai Sistem	170
Gambar 5.58: Halaman Tambah Data Pegawai Sistem	171
Gambar 5.59: Halaman Edit Data Pegawai Sistem.....	172
Gambar 5.60: Halaman Visualisasi Sistem	172
Gambar 5.61: Visualisasi Data Training	179
Gambar 5.62: Visualisasi Data Testing.....	186
Gambar 5.63: Model OLS	187
Gambar 5.65: Model OLS.....	191
Gambar 5.66: Grafik Asumsi Linear.....	193
Gambar 5.67: Grafik Distribusi Residual.....	195
Gambar 5.68: Tabel Matriks Pearson Korelasi	198
Gambar 5.70: Homoskedastisitas.....	203

Daftar Tabel

Tabel 3.1: Dataset <i>data training</i>	66
Tabel 3.2: Dataset <i>data testing</i>	68
Tabel 3. 3: Spesifikasi <i>software</i> yang digunakan	69
Tabel 3. 4: Spesifikasi <i>hardware</i> yang digunakan	70
Tabel 4.1: Indikator capaian berdasarkan metodologi penelitian.....	72
Tabel 5.1: Definisi Aktor	82
Tabel 5.2:Definisi <i>use case</i>	82
Tabel 5.3: Skenario <i>Use case Machine learning</i>	83
Tabel 5.4: Skenario <i>Use case Login Admin</i>	84
Tabel 5.5: Skenario <i>Use case Registrasi</i>	85
Tabel 5.6: Skenario <i>Use case Melakukan Prediksi Gaji Karyawan</i>	85
Tabel 5.7: Skenario <i>Use case Melakukan Visualisasi Data</i>	86
Tabel 5.8: Skenario <i>Use case Melakukan Pengolahan Data</i>	87
Tabel 5.9: Skenario <i>Use case Logot Admin</i>	87
Tabel 5.10: Tabel django_session	122
Tabel 5.11: Tabel django_migration	122
Tabel 5.12: Tabel prediksi_pegawai	123
Tabel 5.13: Tabel auth_user_groups.....	123
Tabel 5.14: Tabel auth_group	124
Tabel 5.15: Tabel auth_user.....	124
Tabel 5.16: Tabel auth_user_user_permissions.....	125
Tabel 5.17: Tabel auth_group_permissions	125
Tabel 5.18: Tabel django_admin_log.....	126
Tabel 5.19: Tabel django_content_type	127
Tabel 5.20: Tabel auth_permission	127

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 intelektual dan internet of things sebagai tulang punggung pergerakan dan koneksi antara manusia dengan mesin(Banuprasetyo and Trisyanti, 2018). Sehingga, terdapat penggabungan teknologi digital dan internet dengan industri konvensional, yang bertujuan untuk meningkatkan produktivitas, efisiensi dan layanan konsumen secara signifikan(Prasetyo and Sutopo, 2017). Era revolusi ini akan mendisrupsi berbagai kegiatan diberbagai bidang seperti pada bidang teknologi, ekonomi, sosial, dan politik(Prasetyo and Trisyanti, 2018). Saat ini, kehidupan berada diawal revolusi yang secara mendasar mengubah cara hidup, bekerja, dan berhubungan satu sama lain (Pangaribuan and Irwansyah, 2019).

Perubahan karakteristik pekerjaan adalah salah satu dampak tersendiri dari datangnya revolusi industri 4.0(Shahroom and Hussin, 2018). Karakteristik pekerjaan yang berubah akan mendisrupsi pekerjaan yang telah ada dan menggantikannya dengan pekerjaan dengan karakteristik baru (Kergroach, 2017). Karakteristik baru pada pekerjaan juga membutuhkan kompetensi baru kepada para pekerja(Manda and Dhaou, 2019). Tentunya perusahaan harus siap untuk saling bersaing dengan perusahaan yang lain(Y. Adrianova Eka and Anyan, 2020). Selanjutnya, perusahaan perlu memiliki keunggulan dan manajemen yang efektif untuk menghadapi persaingan tersebut(Y. Adrianova Eka and Anyan, 2020). Dengan demikian salah satu aspek yang berpengaruh besar terhadap kemajuan dan keberhasilan sebuah perusahaan adalah kinerja pegawai(Y. Adrianova Eka and Anyan, 2020). Walaupun perusahaan tersebut memiliki teknologi yang canggih, namun tidak terdapat tenaga kerja didalamnya, perusahaan tidak akan dapat mencapai tujuannya(Y. Adrianova Eka and Anyan, 2020).

Oleh karena itu, penentuan gaji yang tepat oleh perusahaan kepada pegawai adalah salah satu faktor yang berpengaruh secara internal terhadap kemajuan perusahaan. Selain itu, perusahaan juga harus bersedia mengeluarkan gaji bonus bagi pegawainya 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 pegawai berdasarkan kualitas data.

Karakteristik dataset yang digunakan untuk memprediksi gaji pegawai 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 (Saputra Tamrin, Rumapea and Mambo, 2017). 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 pegawai. Tentunya hasil prediksi gaji pegawai 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 pegawai pada perusahaan?
2. Bagaimana cara menganalisis uji validitas dan korelasi dari dataset gaji pegawai pada perusahaan?
3. Bagaimana cara membuat model prediksi gaji pegawai yang tepat?
4. Bagaimana cara melakukan visualisasi data dari hasil model prediksi gaji pegawai?

1.3 Tujuan

Tujuan penilitian ini sebagai berikut :

1. Menganalisis data kepegawaian yang berkaitan dengan gaji pegawai.
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. Merancang sistem berbasis web base dengan *framework* Django.

1.4 Manfaat

Manfaat penilitian ini sebagai berikut :

1. Sebagai wadah untuk memberikan inovasi baru dalam hal melakukan prediksi gaji pegawai.
2. Bagi perusahaan, penelitian ini diharapkan dapat membantu dalam menentukan keputusan yang tepat dalam penentuan gaji pegawai 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 pegawai 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 pegawai.

1.6 Sistematika Penulisan

Dalam buku 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 pegawai.

2. BAB II LANDASAN TEORI

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

3. BAB III GAMBARAN OBJEK STUDI

Bab ini menjelaskan mengenai objek studi dilakukan pada penelitian.

4. BAB IV METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi yang digunakan dalam melakukan penyelesaian masalah.

5. BAB V PEMBUATAN APLIKASI

Bab ini menjelaskan tentang proses perancangan aplikasi prediksi menggunakan bahasa pemrograman python dengan *framework* django.

6. BAB VI KESIMPULAN DAN SARAN

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

Bab II

Landasan Teori

2.1 Teori Umum

2.1.1 Prediksi

Prediksi yaitu suatu proses untuk memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa yang akan datang, berdasarkan informasi dari masa lalu dan sekarang yang dimiliki, agar suatu kesalahan (selisih antara sesuatu yang terjadi dengan hasil perkiraan) mampu diperkecil. Prediksi tidak harus memberikan jawaban secara tepat kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin di masa yang akan terjadi. (Triyanto, Sismoro and Laksito, 2019)

Prediksi adalah salah satu unsur yang dapat digunakan untuk mendukung dalam pengambilan keputusan yang tepat. Prediksi dalam pengambilan keputusan didasarkan berdasar data yang ada pada waktu sekarang dan waktu lampau (telah terjadi), sehingga dapat digunakan untuk mendeskripsikan suatu kondisi yang sesuai dengan target yang ingin dicapai. (Indra Sanjaya and Heksaputra, 2020)

2.1.2 Gaji

Menurut Sukirno (2013:350), “Gaji dapat diartikan sebagai pembayaran (upah) kepada tiap pekerja-pekerja tetap dan tenaga kerja profesional, seperti pegawai pemerintah, dosen, guru, manajer, dan akuntan” (Sagala, 2018)

Terdapat fungsi dari gaji, diantaranya adalah sebagai berikut(Rinda Faradilla, 2021).

1. Gaji digunakan sebagai penarik seseorang ke dalam sebuah organisasi yang memiliki keahlian tertentu
2. Sebagai pendorong pegawai untuk semakin giat dan semakin berprestasi
3. Sebagai motivasi prestasi pegawai dalam jangka panjang

Adapun beberapa sistem dana pemberian gaji adalah sebagai berikut(Rinda Faradilla, 2021).

1. Berdasarkan waktu, dimana gaji dibayarkan berdasarkan seberapa lama seorang pekerja bekerja, dan dapat dihitung dalam per jam bahkan per bulan.
2. Berdasarkan satuan hasil, dimana gaji dibayarkan berdasar jumlah produk yang dihasilkan.
3. Borongan, dimana gaji diberikan didasarkan kesepakatan kedua belah pihak.
4. Upah partisipasi atau bonus, dimana upah tambahan di luar gaji pokok dengan tujuan sehingga pekerja mampu menjalankan tugas, bertanggung jawab serta menghasilkan pekerjaan yang lebih baik.
5. Mitra usaha, dimana gaji diberikan kepada organisasi yang bekerjasama dalam bentuk saham perusahaan.
6. Skala berubah atau *sliding scale*, dimana jumlah gaji diberikan berdasarkan pendapatan pada hari tersebut sehingga selalu berubah. Gaji bergantung pada hasil penjualan, jika penjualan meningkat maka pembayaran gaji juga akan bertambah dan sebaliknya.

2.1.3 Pegawai

Seorang ahli bernama Soedaryono (2000:6) yang terdapat di dalam bukunya dengan berjudul “Tata Laksana Kantor”, disebutkan bahwa pengertian dari pegawai yaitu “Seseorang yang sedang melakukan penghidupannya dengan cara bekerja atau melakukan kegiatan berbayar dalam kesatuan organisasi, baik dari kesatuan kerja pemerintah ataupun dari kesatuan kerja swasta”. Berikut, menurut Wijaya (2002:15), istilah pegawai mengandung pengertian sebagai berikut:

1. Menjadi anggota dari suatu kerjasama (organisasi) dengan maksud untuk memperoleh balas jasa/imbalan kompensasi atas jasa yang telah diberikan.
2. Berada dalam sistem kerja yang bersifat lugas/pamrih.
3. Berkedudukan selaku penerima kerja serta berhadapan dengan pihak memberi kerja.
4. Kedudukan sebagai penerima kerja itu diperoleh setelah melalui proses penerimaan dari perekutan.
5. Akan menghadapi masa pemberhentian (putusan hubungan kerja).

Lalu kemudian menurut Robbins pada tahun 2006 yang di dalam bukunya dengan berjudul “Perilaku Organisasi” pengertian pegawai didefinisikan sebagai “Orang pribadi yang melakukan kerja kepada pemberi kerja, baik sebagai pegawai tetap ataupun tidak berdasarkan kesepakatan kerja baik tertulis ataupun tidak tertulis, untuk melangsungkan suatu pekerjaan dalam jabatan atau kegiatan tertentu yang telah ditetapkan oleh pemberi kerja.” (Habibie Asnar, 2013)

2.1.4 Aplikasi

Aplikasi adalah suatu program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna program tersebut dengan tujuan mendapatkan hasil sesuai dengan tujuan dan proses bisnis. Aplikasi juga dapat diartikan sebagai salah satu pemecahan masalah dengan menggunakan Teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan maupun pemrosesan data yang diharapkan(Abdurahman, Riswya and Id, 2014).

2.1.5 Data

Data merupakan komponen utama data sistem informasi perusahaan karena semua informasi untuk pengambilan keputusan berasal dari data. Ada berbagai gagasan dalam pendefinisian data, salah satu sumber menyebutkan bahwa data adalah fakta-fakta, pemikiran, ataupun pendapat yang tidak atau belum mempunyai arti kegunaannya. Sedangkan pengertian lain dari data dapat didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili, tindakan, benda, dan sebagainya.

Data dapat terbentuk dari karakter yang dapat berupa bentukan alphabet, angka, maupun simbol khusus dan merupakan bentukan yang masih mentah, sehingga diperlukan pengolahan lebih lanjut melalui suatu model untuk menghasilkan informasi(das Sayan, Barik Rupashri and Mukherjee Ayush, 2020).

2.1.6 *Unified Modeling Language (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.

2.2 Bahasa Pemrograman

2.2.1 Python



Gambar 2.1: Logo Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Pyhton disebutkan sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standart yang besar serta komprehensif. Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis.

Seperti halnya bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai bahasa *script* meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan

dengan menggunakan bahasa *script*. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi. Saat ini kode python dapat dijalankan di berbagai *platform* sistem operasi, beberapa diantaranya adalah :

- a. Linux/Unix
- b. Windows
- c. Max OS X
- d. Java Virtual Machine
- e. Amiga
- f. Palm
- g. Symbian (untuk produk-produk Nokia)

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Lisensi Python tidak bertentangan baik menurut definisi Open Source maupun General Public License (GPL).

Bahasa pemrograman Python tentunya tidak dapat diimplementasikan atau digunakan secara langsung pada *software* teks editor ataupun terminal pada *device* seperti bahasa pemrograman PHP. Bahasa pemrograman ini akan dapat digunakan jika telah ter-install pada *device* yang digunakan.

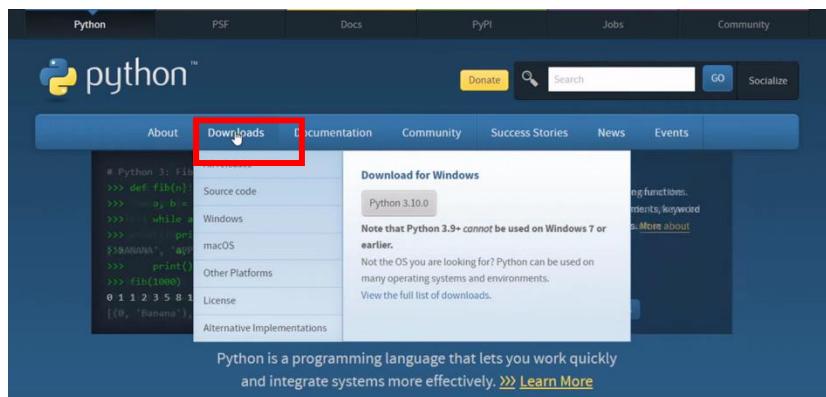
Di bawah ini dipaparkan cara untuk menginstall bahasa pemrograman Python versi 3.10.0 pada sistem operasi Windows.

1. Masuk ke alamat <https://www.python.org>



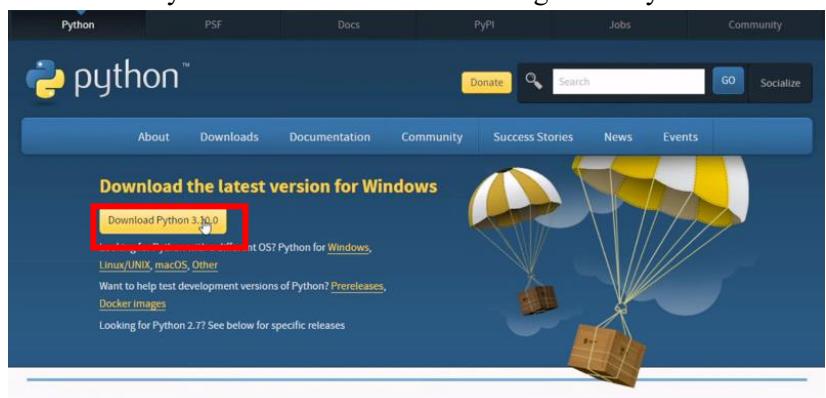
Gambar 2.2: Hyperlink Web Python.org

2. Setelah masuk ke halaman utama web tersebut, kemudian pilih tab “Downloads”



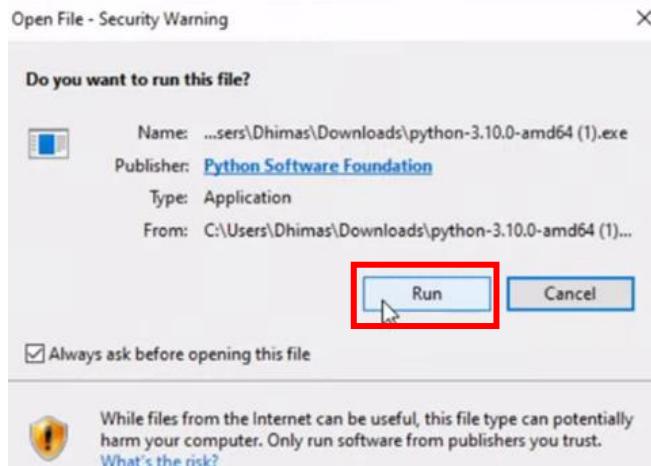
Gambar 2.3: Halaman Utama Web Python

3. Ketika telah masuk ke bagian tab Downloads, tekan tombol “Download Python 3.10.0” untuk mulai mengunduh Python.



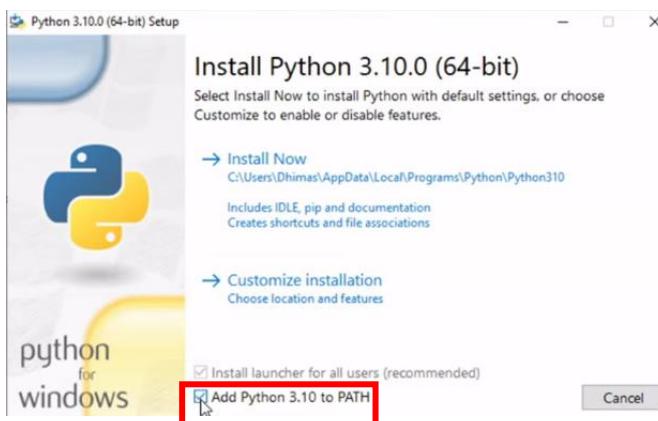
Gambar 2.4: Halaman Download Python

4. Kemudian tunggu hingga hasil unduhan hingga selesai.
5. Ketika hasil unduhan selesai, buka hasil unduhan. Ketika terdapat *Security Warning*, tekan tombol “Run”.



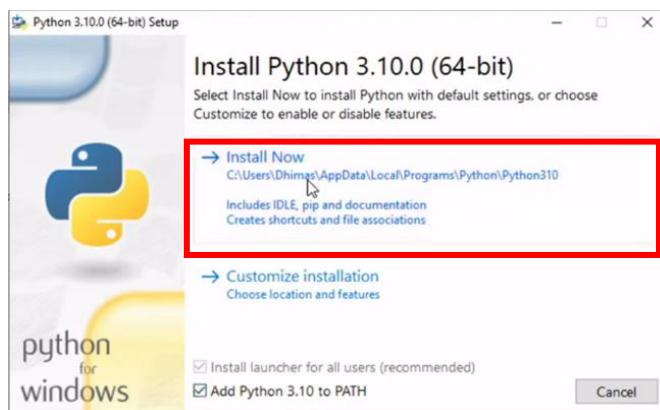
Gambar 2.5: Security Warning Desktop

6. Centang “Add Python 3.10 to PATH”. Karena jika tidak dicentang, maka harus dilakukan *setting* manual *inronment variable*.



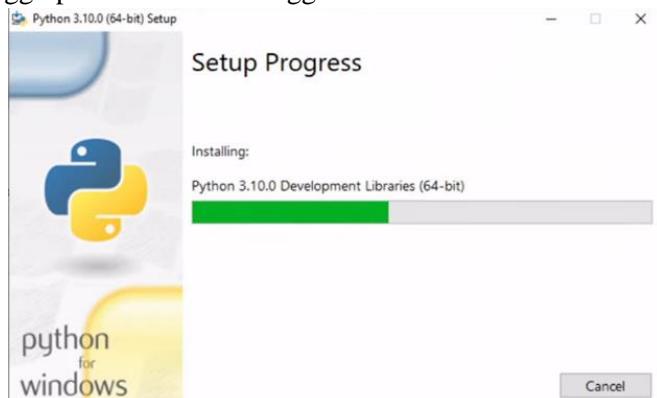
Gambar 2.6: Instalasi Python Add Python 3.10 to PATH

7. Kemudian, pilih “Install Now” untuk melakukan instalasi pada *device*.



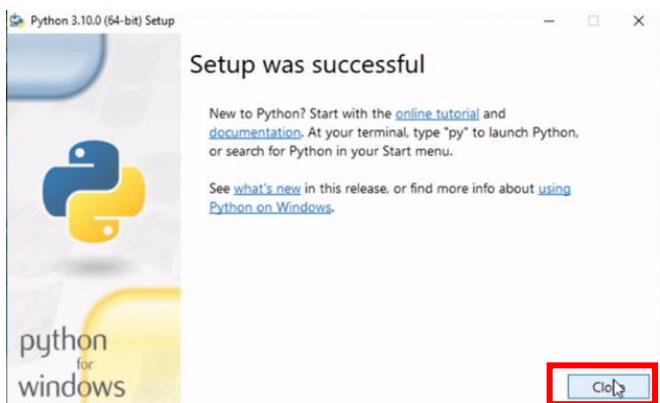
Gambar 2.7: Instalasi Pyton *Install Now*

8. Tunggu proses instalasi hingga selesai.



Gambar 2.8: Setup Progress Instalasi Python

9. Jika proses instalasi telah selesai, tekan tombol “Close”.



Gambar 2.9: Selesai Instalasi Python

10. Untuk memastikan apakah Python telah terinstall atau belum, dapat digunakan pada terminal/CMD pada *device*. Kemudian ketik perintah `python` pada CMD.

A screenshot of a Command Prompt window titled "Command Prompt - python". The window shows the following text:

```
C:\Users\win10>python
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

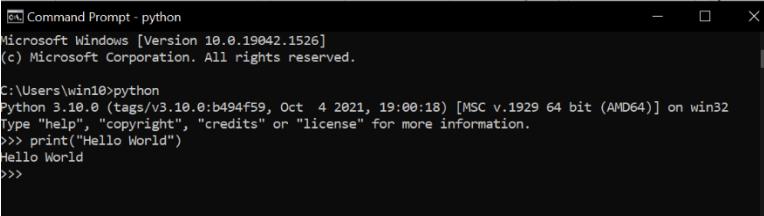
C:\Users\win10>Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

A red box highlights the "Python 3.10.0" output line.

Gambar 2.10: CMD Cek Hasil Instalasi Python

Jika Python telah terinstall, maka akan terdapat *output* versi dari Python yang telah terinstall. *Output* dari hasil install ini adalah Python 3.10.0

11. Tanda `>>>` pada terminal menunjukan bahwa Python siap menerima perintah. Dapat dilakukan perintah `print("hello world")` pada terminal.



```
PS C:\Users\win10>python
Python 3.10.0 (tags/v3.10.0:b494f59, Oct  4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello World")
Hello World
>>>
```

Gambar 2.11: CMD Python Siap Menerima Perintah

2.2.2 HTML



Gambar 2.12: Logo HTML

HTML (*Hyper Text Mark Up Language*) adalah bahasa yang dipergunakan dengan tujuan untuk mendeskripsikan struktur dari sebuah halaman web. HTML memiliki fungsi untuk mempublikasikan dokumen secara online. *Tags* adalah statement dasar dari HTML. Sebuah *tag* dinyatakan dalam kurung siku (<>). *Tags* yang ditujukan untuk sebuah dokumen atau bagian dari suatu dokumen harus dibuat dalam bentuk berpasangan. *Tag* terdiri dari dua macam yaitu *tag* pembuka dan *tag* penutup. Dimana *tag* penutup menggunakan tambahan tanda garis miring (/) di awal nama *tag*(Pahlevi, Mulyani and Khoir, 2018).

2.2.3 CSS



Gambar 2.13: Logo CSS

CSS memiliki kepanjangan *Cascading Style Sheet* adalah bahasa-bahasa yang dapat merepresentasikan halaman web. Representasi tersebut diantaranya adalah warna, *layout*, dan *font*. Dengan digunakannya CSS dalam membangun halaman web, seorang web developer dapat membuat halaman web yang dapat beradaptasi dengan berbagai macam ukuran layar. Pembuatan skrip CSS biasanya terpisah dengan halaman HTML. Meskipun skrip CSS dapat disisipkan di dalam halaman HTML. Hal ini ditujukan untuk memudahkan pengaturan halaman HTML yang memiliki rancangan yang sama (Henderson, 2009:72) (Pahlevi, Mulyani and Khoir, 2018).

2.2.4 Bootstrap



Gambar 2.14: Logo Bootstrap

Bootstrap adalah suatu *framework* untuk CSS (*Cascading Style Sheet*) dan berupa produk bentuk opensource (terbuka untuk siapapun) yang dibuat oleh Mark Otto dan Jacob Thornton. Pada awalnya Bootstrap ini dibuat untuk membuat standarisasi dalam front end untuk seluruh programmer yang ada di perusahaannya. Bootstrap telah berubah dari yang sebelumnya adalah CSS-Driven proyek ke sebuah host dari JavaScript plugins dan ikon yang dapat dengan mudah digunakan untuk formulir dan tombol.

2.2.5 Django



Gambar 2.15: Logo Django

Django ialah sebuah web *framework* berbasis bahasa pemrograman Python yang didesain untuk membuat suatu aplikasi web yang dinamis, kaya fitur dan

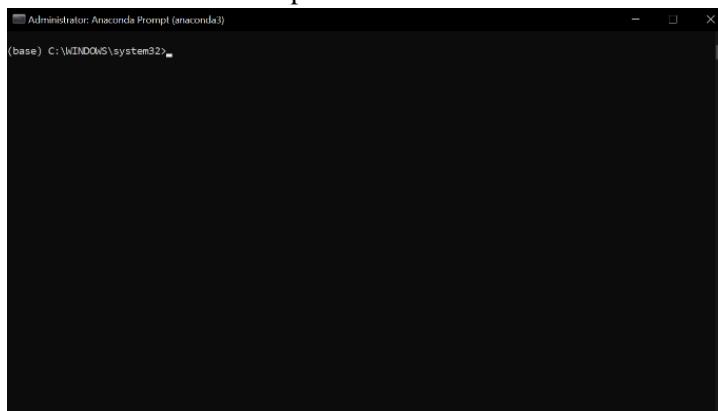
aman. (Saputra and Fathoni Aji, 2018) Django yang dikembangkan oleh Django Software Foundation terus mendapatkan perbaikan sehingga membuat web framework yang satu ini menjadi pilihan utama bagi banyak pengembang aplikasi web. (Saputra and Fathoni Aji, 2018)

Framework Django bertujuan untuk memudahkan pengembangan situs web dan basis data yang kompleks (Django Software Foundation, 2017). Salah satu keunggulan django yaitu *framework* ini telah merepresentasikan model ORM (*Object Relational Mapper*) sehingga tidak perlu lagi untuk menyesuaikan query jika terjadi perubahan *database* yang digunakan. Berikut ini adalah beberapa keuntungan menggunakan Django:

1. *Object-Relational Mapping (ORM) Support.*
2. *Automatic Admin Interface.*
3. *Elegant URL Design.*
4. *Template System.*
5. *Cache system.*
6. *Internationalization.*
7. *A light weight web server for development and test*(Rahimah, Rusdianto and Ananta, 2019).

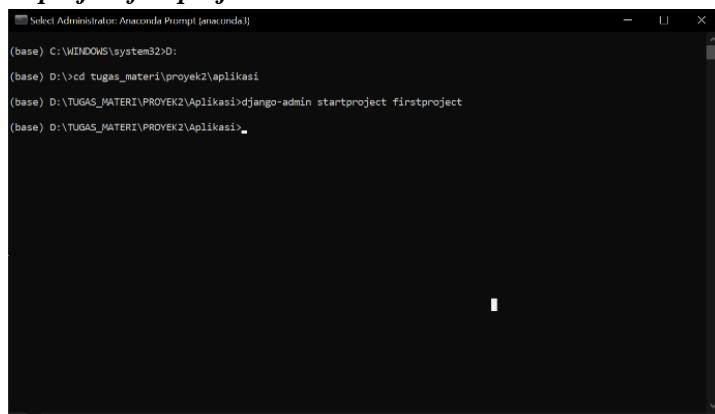
Berikut dijelaskan juga langkah-langkah instalasi *framework* Django.

1. Aktifkan Anaconda Prompt Run Administrator



Gambar 2.16: Anaconda Prompt Administrator

2. Ketikan perintah ***django-admin startproject namaproject*** untuk membuat projek. Pada kali ini contoh dari nama projek adalah firstproject. Maka perintahnya akan menjadi ***django-admin startproject firstproject***



```
Administrator: Anaconda Prompt (anaconda3)
(base) C:\Windows\system32>D:
(base) D:\>cd tugas_materi\proyek2\aplikasi
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>django-admin startproject firstproject
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>
```

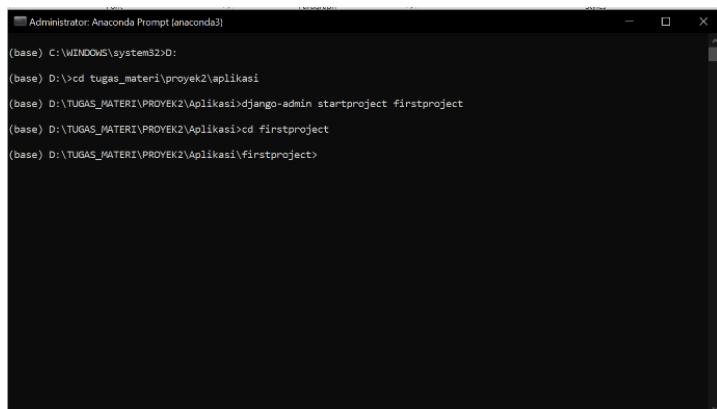
Gambar 2.17: Pembuatan Projek Django

Nantinya secara otomatis, akan terbuat folder dengan nama firstproject yang berisi folder firstproject dan file manage.py



Gambar 2.18: Folder Projek Django Yang Terbuat

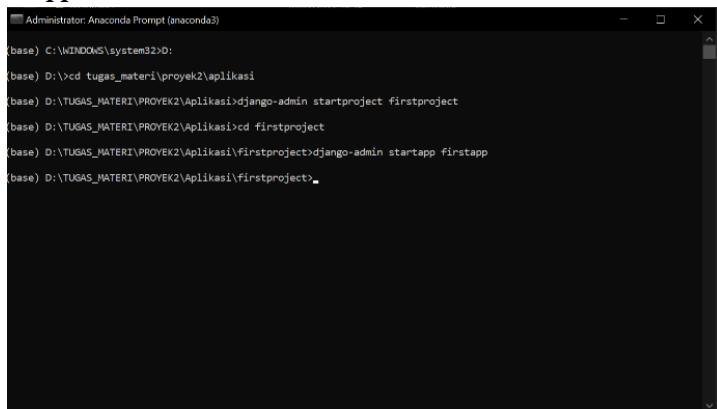
3. Masuk ke folder project yang telah dibuat sebelumnya dengan perintah ***cd firstproject***



```
(base) C:\WINDOWS\system32>D:  
(base) D:\>cd tugas_materi\proyek2\aplikasi  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>django-admin startproject firstproject  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>cd firstproject  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi\firstproject>
```

Gambar 2.19: Masuk Folder Projek Django

4. Ketikan perintah ***django-admin startapp namaaplikasi*** untuk membuat aplikasi. Pada kali ini contoh dari nama aplikasi adalah *firstapp*. Maka perintahnya akan menjadi ***django-admin startapp firstapp***



```
(base) C:\WINDOWS\system32>D:  
(base) D:\>cd tugas_materi\proyek2\aplikasi  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>django-admin startproject firstproject  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>cd firstproject  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi\firstproject>django-admin startapp firstapp  
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi\firstproject>
```

Gambar 2.20: Membuat Folder Aplikasi Django

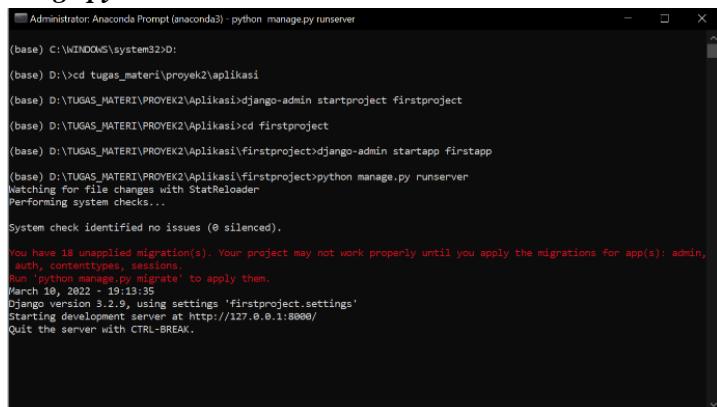
Nantinya secara otomatis, akan terbuat folder dengan nama *firstapp* di dalam folder *firstproject*.

Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi

Name	Date modified	Type	Size
firstapp	10/03/2022 19:08	File folder	
firstproject	10/03/2022 18:59	File folder	
manage.py	10/03/2022 18:59	Python file	1 KB

Gambar 2.21: Folder Aplikasi Django

5. Kemudian jalankan aplikasi dengan mengetikan perintah ***python manage.py runserver***.



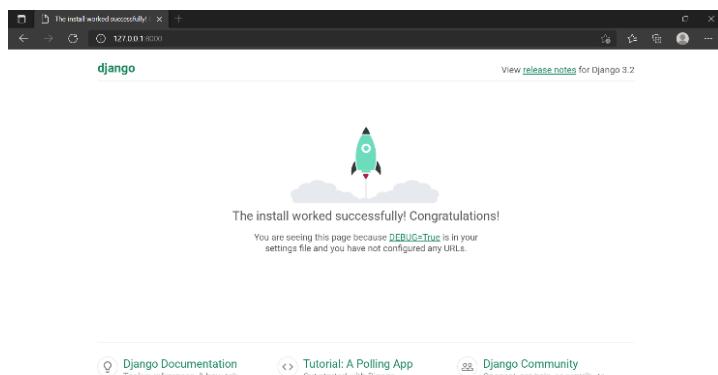
```
Administrator: Anaconda Prompt (anaconda3) - python manage.py runserver
(base) C:\WINDOWS\system32>D:
(base) D:>cd tugas_materi\proyek2\aplikasi
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>django-admin startproject firstproject
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi>cd firstproject
(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi\firstproject>django-admin startapp firstapp
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
March 10, 2022 - 19:13:35
Django version 3.2.9, using settings 'firstproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Gambar 2.22: Menjalankan Projek Aplikasi

Untuk menjalankan aplikasi, ketik alamat domain <http://127.0.0.1:8000/> pada *web browser*.

Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi



Gambar 2.23: Running Aplikasi Pada *Web browser*

6. Benahi *migrate* dengan cara mengetikan perintah ***python manage.py migrate***.

```
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions
Run 'python manage.py migrate' to apply them.
```

Gambar 2.24: *Migrate Python Django*

7. Ketik perintah **Ctrl+C** untuk menghentikan aplikasi yang berjalan, lalu ketikan perintah ***python manage.py migrate***.

```

Administrator: Anaconda Prompt [anaconda3]
Not Found: /favicon.ico
[18/Mar/2022 19:18:18] "GET /favicon.ico HTTP/1.1" 404 2116

(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi\firstproject>python manage.py migrate
Operations to perform:
  Apply all migrations: auth, admin, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002.Alter_permission_name_max_length... OK
  Applying auth.0003.Alter_user_email_max_length... OK
  Applying auth.0004_Alter_user_username_copies... OK
  Applying auth.0005_Set_user_last_login_null... OK
  Applying auth.0005_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK

(base) D:\TUGAS_MATERI\PROYEK2\Aplikasi\firstproject>

```

Gambar 2.25: Memasukan Perintah Python Migrate Pada Anaconda Prompt

Setelah itu, jalankan kembali aplikasi dengan mengetikan perintah **python manage.py runserver**.

8. Lakukan edit pada file setting.py pada kode editor untuk menyesuaikan nama aplikasi yang telah dibuat. Tambahkan 'firstapp' pada bagian **INSTALLED APP**.

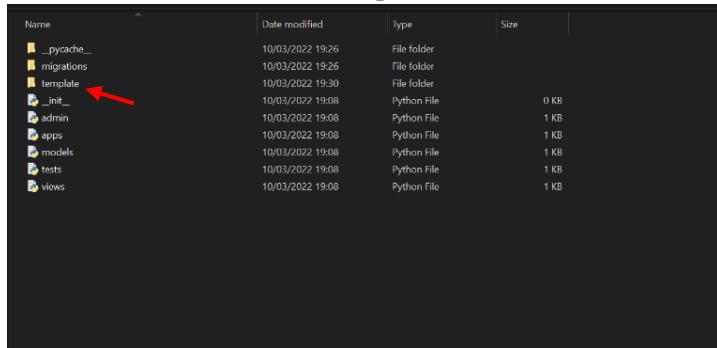
```

# File: settings.py
# Line 33: INSTALLED_APPS = [
# Line 40:     'firstapp',
# Line 41: ]
# Line 42: MIDDLEWARE = [
# Line 43:     'django.middleware.security.SecurityMiddleware',
# Line 44:     'django.contrib.sessions.middleware.SessionMiddleware',
# Line 45:     'django.middleware.common.CommonMiddleware',
# Line 46:     'django.middleware.csrf.CsrfViewMiddleware',
# Line 47:     'django.contrib.auth.middleware.AuthenticationMiddleware',
# Line 48:     'django.contrib.messages.middleware.MessageMiddleware',
# Line 49:     'django.middleware.clickjacking.XFrameOptionsMiddleware',
# Line 50: ]
# Line 53: ROOT_URLCONF = 'firstproject.urls'

```

Gambar 2.26: Menambahkan Nama Aplikasi Pada File Setting.py

- Buat folder *template* pada folder *firstapp*. Folder ini nantinya akan berisi file HTML untuk tampilan halaman web.



Gambar 2.27: Membuat Folder Bernama *Template*

- Lakukan edit pada *setting.py* dengan menambahkan kode `import os` untuk menggabungkan folder dan file, serta kode `os.path.join(BASE_DIR, 'template')` pada bagian *TEMPLATE*. Kode ini akan menghubungkan *template* dengan berbagai komponen arsitektur *framework* Django.

```

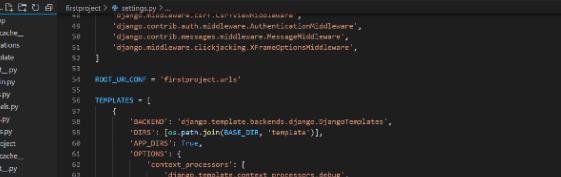
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-ovv3tukarekhl51j1nk-o@#81-wj2(r3mvv7q+18h7$e'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

```

Gambar 2.28: Menambahkan *Library* os



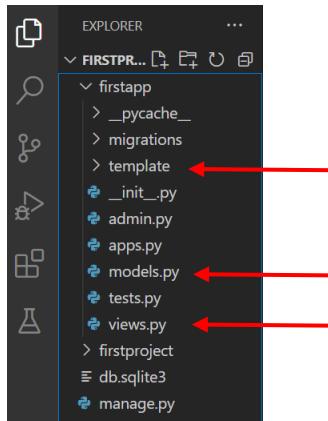
A screenshot of a Windows desktop environment. In the foreground, a code editor window for Python is open, displaying the file `firstproject/settings.py`. The code defines a `urlpatterns` list and a `TEMPLATES` configuration. The `TEMPLATES` section includes a `OPTIONS` key with settings for `BACKEND`, `DIRS`, `APP_DIRS`, `OPTIONS`, and `context_processors`. The `context_processors` list contains several Django template processors like `debug`, `request`, `auth`, and `messages`. At the bottom of the file, `WSGI_APPLICATION` is set to `'firstproject.wsgi.application'`. Below the code editor, the taskbar shows icons for File Explorer, Task View, and other system applications. The system tray at the bottom right indicates battery level, signal strength, and system status.

Gambar 2.29: Menambahkan Kode `os.path.join(BASE_DIR, 'template')` Pada Template

11. Framework telah dapat digunakan untuk membuat aplikasi dengan memerhatikan konsep arsitektur MTV (*Model Template Views*) pada framework.

2.2.6 Model Template Views (MTV)

Model Template View atau yang biasanya disingkat menjadi MTV-Architecture adalah komponen system berupa diagram arsitektur model pada framework Django. Komponen tersebut (*Model, Template, dan Views*) saling terhubung satu sama lainnya dan nantinya akan membentuk suatu halaman web. Pemodelan arsitektur ini disebut juga sebagai bentuk pemrograman terstruktur.



Gambar 2.30: Arsitektur MVT *Framework* Django

2.2.7 Model

Model pada arsitektur *framework* Django adalah suatu komponen yang bertugas sebagai media perantara antara sistem dengan basis datanya. Seluruh komunikasi dengan basis data akan dijembatani oleh komponen *Model*.

```
File Edit Selection View Go Run Terminal Help models.py - Intellisense - Visual Studio Code
FIRSTPROJECT ...
  firstapp
    > __pycache__
    > migrations
    > template
      <-- Red arrow
    > __init__.py
    > admin.py
    > apps.py
    > models.py
      <-- Red arrow
    > tests.py
    > views.py
  firstproject
  db.sqlite3
  manage.py

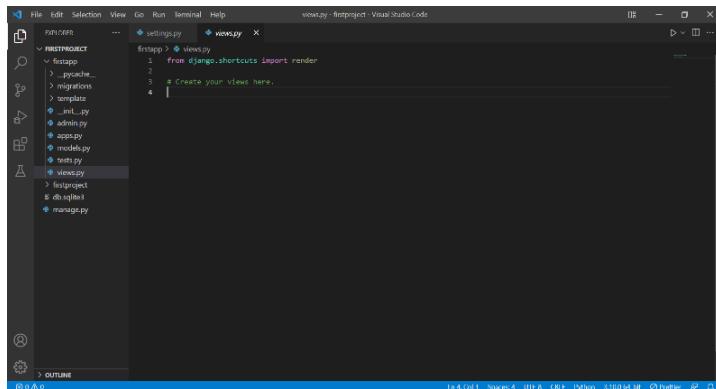
models.py
settings.py
  1  from django.db import models
  2
  3
  4 # Create your models here.
```

Gambar 2.31: Isi File models.py

2.2.8 View

View pada arsitektur *framework* Django adalah suatu komponen yang bertugas untuk menangani *request* dan *response* web dimana komponen ini akan

menerjemahkan permintaan dari web dan mengembalikannya dalam bentuk apapun (halaman HTML, kode error, atau XML)



The screenshot shows the Visual Studio Code interface with the file 'views.py' open in the center editor pane. The code contains the following:

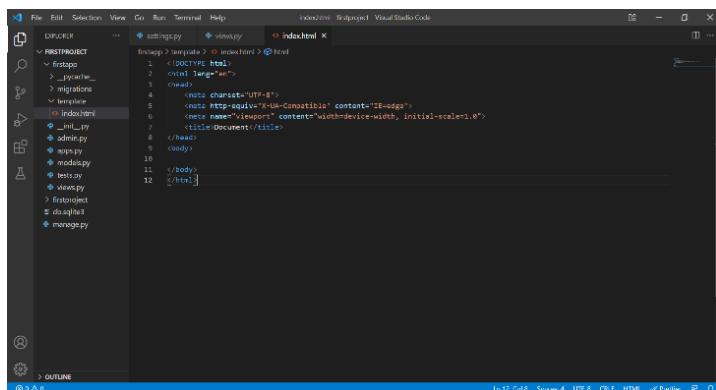
```
1 from django.shortcuts import render
2
3 # Create your views here.
4
```

The left sidebar shows the project structure under 'FIRSTPROJECT' with files like settings.py, urls.py, and manage.py.

Gambar 2.32: Isi File *views.py*

2.2.9 *Template*

Templates pada arsitektur *framework* Django adalah suatu komponen yang berisi kumpulan kode HTML yang berfungsi sebagai pengatur tampilan halaman web. Halaman ini hanya berfokus pada tampilan halaman web.



The screenshot shows the Visual Studio Code interface with the file 'index.html' open in the center editor pane. The code contains the following:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
    </head>
    <body>
        <h1>Hello World</h1>
    </body>
</html>
```

The left sidebar shows the project structure under 'FIRSTPROJECT' with files like settings.py, urls.py, and manage.py.

Gambar 2.33: Isi File *index.html* Pada Folder *Template*

2.3 Metode Yang Digunakan

2.3.1 Teknik Pengumpulan Data



Gambar 2.34: Logo Kaggle

Dalam penelitian ini, dataset yang digunakan dalam proses regresi linier berganda adalah dataset kepegawaian yang bersumber dari Kaggle. Dataset yang digunakan adalah employee_attrition_train.csv sebanyak 1029 baris dan 35 kolom, dan employee_attrition_test.csv sebanyak 441 baris dan 34 kolom. Dataset tersebut dapat dilihat pada *link* sebagai berikut : <https://www.kaggle.com/colearninglounge/employee-attrition>

2.3.2 Machine learning

Machine learning (ML) merupakan bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar (daqiqil.id, 2022). *Machine learning* dapat diartikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan dapat menghasilkan suatu prediksi di masa yang akan datang. (Homepage *et al.*, 2019) Menurut Samuel, ML berisi sebuah algoritma yang bersifat generik (umum) dimana algorima tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik. Pada intinya, algoritma yang generik tersebut ketika diberikan sejumlah data maka dapat dibangun sebuah aturan atau model atau inferensi dari data tersebut. Sebagai contoh sebuah Algoritma untuk mengenali tulisan tangan dapat digunakan untuk mendeteksi email yang berisi spam dan bukan spam tanpa mengganti kode. Algoritma yang sama

ketika diberikan data pelatihan yang berbeda menghasilkan logika klasifikasi yang berbeda(daqqiqil.id, 2022).

2.3.3 Regresi Linier

Regresi linier (*linear regression*) adalah metode yang dipakai untuk memperoleh model hubungan antara variabel dependen dan variabel independen (Walpole et al., 2012). Terdapat dua jenis regresi linier, yaitu terdiri dari regresi linier sederhana (*simple linear regression*) dan regresi linear berganda (*multiple linear regression*). Perbedaan terdapat pada banyaknya variabel independen. Regresi linier sederhana adalah metode yang dipakai untuk mendapatkan model hubungan antara satu variabel dependen dan satu variabel independen lainnya, sedangkan regresi linier ganda adalah metode yang digunakan untuk memperoleh model hubungan antara satu variabel dependen dengan lebih dari satu variabel independen lainnya (Walpole et al., 2012). (T. Adrianova Eka and Anyan, 2020)

2.3.4 Metode Regresi Linier Berganda

Regresi berganda adalah perpanjangan dari regresi linier sederhana . (Iskandar Akbar, Muttaqin and Vita Dewi Sarini, 2021) Analisis regresi linier berganda dapat digunakan untuk memprediksi hubungan antara satu variabel independen berdasarkan nilai dari dua atau lebih variabel dependen. (Iskandar Akbar, Muttaqin and Vita Dewi Sarini, 2021) Analisis regresi linier berganda juga menghasilkan persamaan matematis. (Iskandar Akbar, Muttaqin and Vita Dewi Sarini, 2021) Jika ada lebih dari dua variabel maka hubungan linier dapat dinyatakan dalam persamaan regresi linier berganda yang dikutip pada persamaan 1 dan persamaan 2 sebagai berikut :

$$Y' = b_0 + b_1X_1 + b_2X_2 + \dots + b_nX_n \quad (1)$$

$$Y'i = b_0 + b_1X_{1i} + b_2X_{2i} + \dots + b_nX_{ni} \quad (2)$$

Keterangan :

Y = nilai-nilai hasil pengamatan

Y' = nilai regresi

i = 1,2,3, k

Pada persamaan di atas ada satu variabel dependen, yaitu Y' dan ada n variabel independen X1, X2, ...Xn. (Iskandar Akbar, Muttaqin and Vita Dewi Sarini, 2021)

2.4 Software

2.4.1 Visual Studio Code



Gambar 2.35: Logo Visual Studio Code

Visual Studio Code (VS Code) adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman dengan bantuan plugin yang dapat dipasang via *marketplace* Visual Studio Code(Yudo Ardianto Agung, 2020).

Berikut fitur-fitur unggulan Visual Studio Code :

1. *Cross platform*

Dapat berjalan di Windows, Linux, dan MacOS. Artinya dapat bekerja pada sistem operasi manapun dengan menggunakan Visual Studio Code.

2. *Lightweight*

Tak perlu menunggu lama untuk memulai. Bahasa, tema, debugger, commands, dan lain-lainnya dapat dikontrol sepenuhnya sesuai keinginan.

3. *Powerfull Editor*

Source code editing yang sangat produktif, seperti membuat code snippets, *IntelliSense*, *auto correct*, dan *formatting*.

4. *Code Debugging*

Salah satu fitur bagus yang ditawarkan Visual Studio Code adalah membantu untuk melakukan debug pada kode dengan cara mengawasi kode, variabel, *call stack* dan *expression* yang mana saja.

5. *Source Control*

Visual Studio Code memiliki *intgrated source control* termasuk *Git support in-the-box* dan penyedia *source code control* lainnya di pasaran. Ini meningkatkan siklus rilis proyek secara signifikan.

6. *Intergrated Terminal*

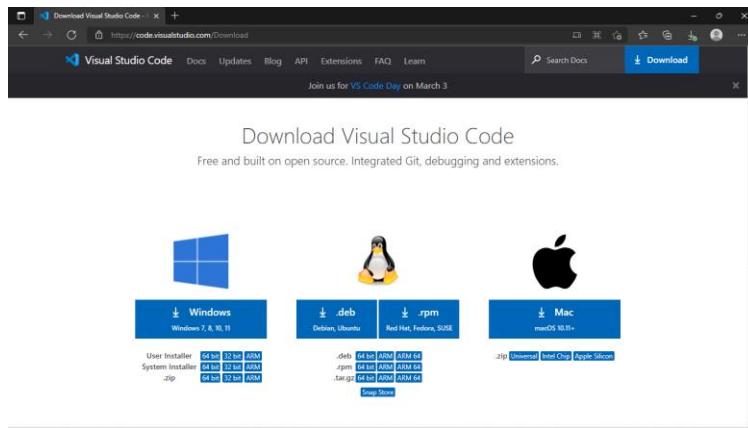
Tidak ada lagi *multiple Windows* dan *Alt-Tabs*. Perintah tersebut dapat dilakukan dengan *command-line task* sekejap dan membuat banyak terminal di dalam editor.

Berikut dijelaskan juga langkah-langkah instalasi Visual Studio Code pada Windows.

1. Download terlebih dahulu *file installer* Visual Studio Code melalui situs resminya.

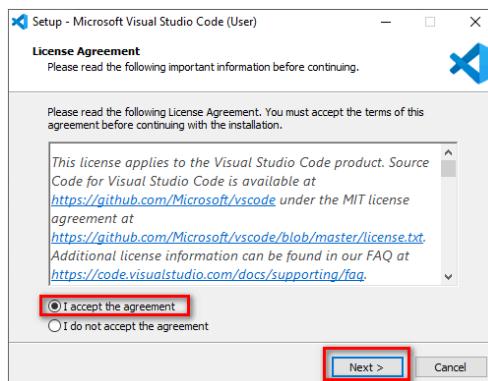
(<https://code.visualstudio.com/Download>)

Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi



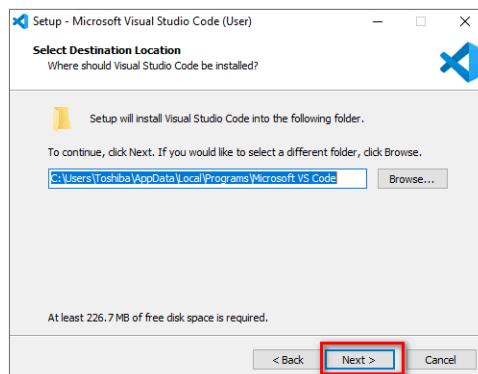
Gambar 2.36: Halaman Web Download Visual Studio Code

- Setelah proses *download* berhasil, lanjut ke proses installasi. Double klik pada *file installer* nya atau klik kanan kemudian pilih *Run as Administrator*. Jika muncul peringatan *Run as Administrator*, silahkan klik *Yes*. Pilih “*I accept the agreement*“ untuk menyetujui “*Licence Agreement*“, kemudian klik *Next*.



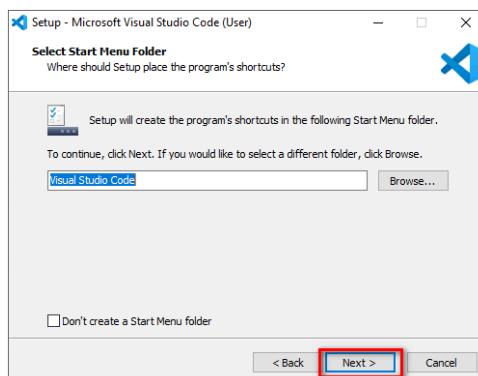
Gambar 2.37: Setup Visual Studio Code

3. Untuk *Select Destination Location* bisa dibiarkan saja jika lokasi instalasi tidak akan dirubah. Klik *Next*.



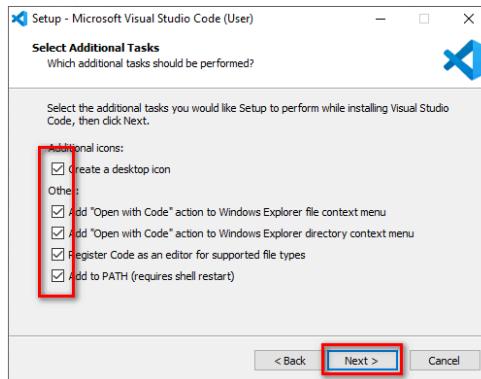
Gambar 2.38: Memilih Lokasi Penyimpanan Visual Studio Code

4. Klik *Next* jika tidak akan merubah *Start Menu Folder*.



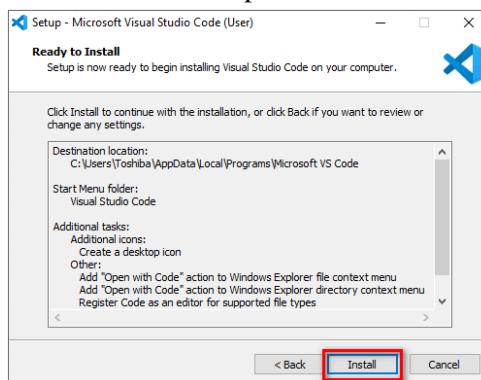
Gambar 2.39: Memilih *Start Menu Folder*

5. Di bagian *Select Additional Tasks* centang semua. Kemudian *Next*.



Gambar 2.40: Memilih *Additional Tasks*

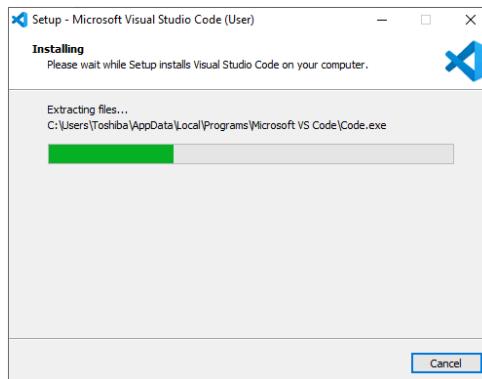
6. Lalu klik *install* untuk memulai proses instalasi.



Gambar 2.41: Memulai Proses Instalasi

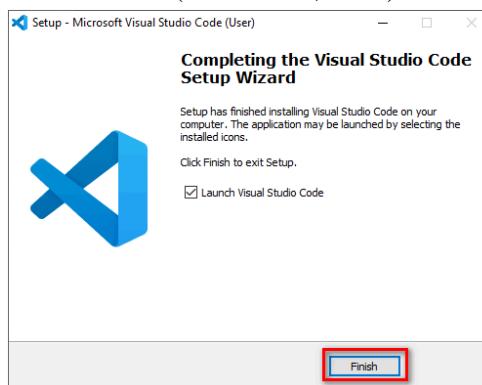
7. Tunggu sampai proses instalasi selesai.

Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi



Gambar 2.42: Proses Instalasi Visual Studio Code

8. Setelah selesai klik *Finish*(Techmania, 2019).



Gambar 2.43: Instalasi Visual Studio Code Selesai

2.4.2 XAMPP



Gambar 2.44: Logo XAMPP

XAMPP adalah sebuah perangkat lunak komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL/MariaDB, PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti Linux, Windows, MacOS, dan juga Solaris. Jika dijabarkan, masing-masing huruf yang ada di dalam nama XAMPP menurut para ahli memiliki arti sebagai berikut ini :

1. *X (Cross Platform)*

Merupakan kode penanda untuk *software cross platform* atau yang bisa berjalan di banyak sistem operasi. Jadi, ada XAMPP untuk Windows, MacOS, Linux. Semua itu bersifat *free download* XAMPP.

2. *A (Apache)*

Apache adalah aplikasi *web server* yang bersifat gratis dan bisa dikembangkan oleh banyak orang (*open source*).

3. *M (MySQL/MariaDB)*

MySQL atau MariaDB merupakan aplikasi *database server* yang dikembangkan oleh orang yang sama. MySQL berperan dalam mengolah, mengedit, dan menghapus daftar melalui *database*.

4. *P (PHP)*

Huruf “P“ yang pertama dari akronim kata XAMPP adalah inisial untuk menunjukkan eksistensi bahasa pemrograman PHP. Bahasa pemrograman ini biasanya digunakan untuk membuat *website* dinamis, contohnya dalam *website* berbasis CMS WordPress.

5. P (*Perl*)

Sementara itu, huruf P selanjutnya merupakan singkatan dari bahasa pemrograman Perl yang kerap digunakan untuk memenuhi berbagai macam kebutuhan. Perl ini bisa berjalan di dalam banyak digunakan.

Program aplikasi XAMPP berfungsi sebagai *server* lokal untuk mengampu berbagai jenis data *website* yang sedang dalam proses pengembangan. Dalam prakteknya, XAMPP digunakan untuk menguji kinerja fitur ataupun menampilkan konten yang ada di dalam *website* kepada orang lain tanpa harus terkoneksi dengan internet, cukup akses melalui XAMPP *Control panel*, atau istilahnya *website offline*. XAMPP bekerja secara *offline* layaknya *web hosting* biasa namun tidak bisa diakses oleh banyak orang.

Berikut dijelaskan juga langkah-langkah instalasi XAMPP pada Windows.

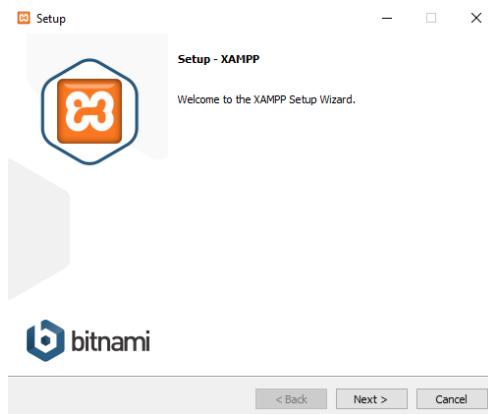
1. Siapkan file XAMPP terlebih dahulu dengan cara *download* pada *link* resminya. (<http://www.apachefriends.org/en/index.html>.)



Gambar 2.45: Halaman Web *Download* XAMPP

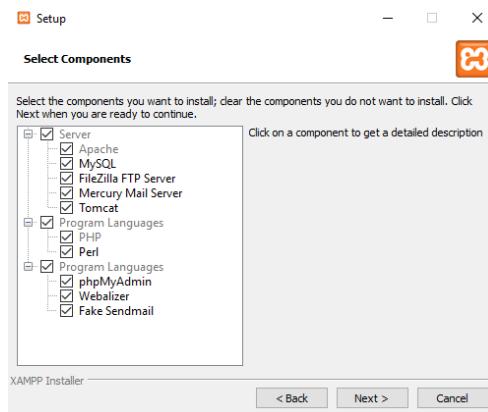
2. Setelah file berhasil *download*, langkah selanjutnya adalah menyiapkan *space/ruang harddisk* di laptop atau komputer dengan minimal masih menyisakan ROM sebesar 1 GB guna menampung gambar, teks, video, dan dokumen *website*.
3. Jalankan file XAMPP yang sudah di-*download*.
4. Setelah klik file tersebut, akan ada jendela baru yang terbuka sebagai indikasi dimulainya proses instalasi. Pada proses ini akan

diminta untuk memilih bahasa Indonesia atau English, pilih saja bahasa Indonesia kemudian klik *next*.



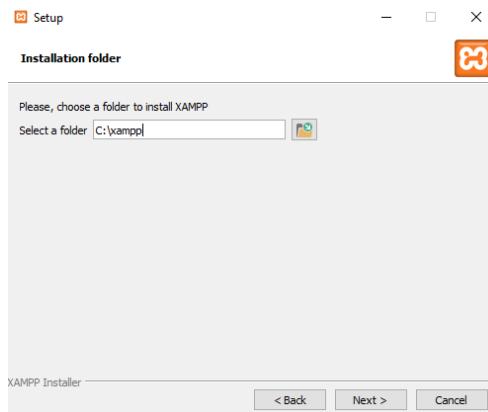
Gambar 2.46: *Setup XAMPP*

5. Setelah itu, akan diminta untuk memilih komponen yang akan digunakan. Standar untuk *server web* berbasis *CMS WordPress* menggunakan *MySQL*, *FileZilla FTP Server*, dan *Apache*, sedangkan dari bahasa pemrograman bisa ceklis pilihan *PHP MyAdmin*, *PHP*, dan *Perl*. Kemudian klik *Next*.



Gambar 2.47: Memilih Komponen Yang Akan Digunakan

6. Pada tahap ini, diminta untuk memilih lokasi file XAMPP yang akan diinstal, gunakan saja pengaturan *default* lalu klik *Next*.



Gambar 2.48: Lokasi File XAMPP

7. Tunggu beberapa saat sampai proses instalasi XAMPP selesai yang ditandai dengan *bar progress* seluruhnya berubah menjadi warna hijau.



Gambar 2.49: Proses Instalasi XAMPP

8. Setelah itu, sudah bisa dijalankan program XAMPP untuk membuat *website* secara offline menggunakan *localhost*.

2.4.3 Jupyter Notebook



Gambar 2.50: Logo Jupyter

Jupyter Notebook adalah lingkungan komputasi interaktif berbasis *web* untuk membuat dokumen *notebook* jupyter. Jupyter notebook mendukung beberapa bahasa seperti Python (Ipython), Julia, R, dan lain-lain. Sebagian besar digunakan untuk analisis data, visualisasi data dan komputasi eksplorasi interaktif lebih lanjut. Awalnya dikembangkan untuk aplikasi *data science* yang ditulis dengan Python, R, dan Julia, Jupyter Notebook berguna dalam segala macam cara untuk semua jenis proyek :

1. Visualisasi Data

Kebanyakan orang memiliki eksposur pertama mereka ke Notebook Jupyter melalui visualisasi data, sebuah *notebook* bersama yang mencakup *rendering* dari beberapa data yang ditetapkan sebagai grafik. Jupyter Notebook memungkinkan untuk membuat visualisasi, tetapi juga membagikannya dan memungkinkan perubahan interaktif pada kode dan kumpulan data yang dibagikan.

2. Berbagi Kode

Layanan *cloud* seperti GitHub dan Pastebin menyediakan cara untuk berbagi kode, tetapi sebagian besar non-interaktif. Dengan

Jupyter Notebook dapat dilihat kode, menjalankannya, dan menampilkan hasilnya langsung di *browser web*.

3. Interaksi Langsung dengan Kode

Kode Notebook Jupyter tidak statis, itu dapat diedit dan dijalankan kembali secara bertahap, dengan umpan balik yang diberikan langsung di *browser*. Notebook juga dapat menyematkan kontrol pengguna (Misalnya Slider atau bidang input teks) yang dapat digunakan sebagai sumber input untuk kode.

4. Mendokumentasikan Contoh Kode

Jika memiliki sepotong kode dan ingin menjelaskan secara bertahap cara kerjanya, dengan umpan balik langsung di sepanjang jalan, dapat menyematkannya dalam Jupyter Notebook. Kode akan tetap berfungsi penuh, dapat menambahkan interaktivitas bersama dengan penjelasan, yang ditampilkan dan diceritakan pada saat yang sama(small-business-tracker.com, 2022).

Berikut langkah-langkah untuk *install* Jupyter Notebook :

1. *Install* Python terlebih dahulu. Setelah Python terinstall, maka dilanjutkan dengan instalasi Jupyter Notebook yang dimulai dengan install pip menggunakan ***pip install virtualenv*** melalui *Command Prompt*.

```
C:\> Command Prompt
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

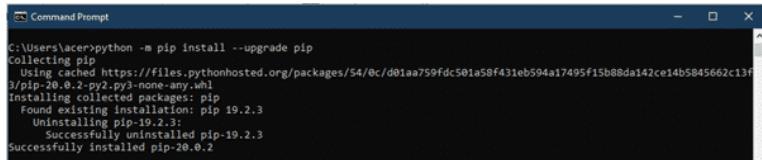
C:\Users\acer>python --version
Python 3.8.2

C:\Users\acer>pip install virtualenv
Collecting virtualenv
  Downloading https://files.pythonhosted.org/packages/23/1a/d55f5fb094564365c3d364fd41701a74ad97379f04fd548cccc3f227d98
/virtualenv-20.0.18-py2.py3-none-any.whl
    |████████| 4.6MB 56KB/s
Collecting appdirs<2,>=1.4.3 (from virtualenv)
  Using cached https://files.pythonhosted.org/packages/56/e6/b810e7008e131349edde4cbcd1b2a21e28cdf115f9af263f8ebbf8447cf1abf
/appdirs-1.4.3-py2.py3-none-any.whl
Collecting distlib<1,>=0.3.0 (from virtualenv)
  Using cached https://files.pythonhosted.org/packages/7d/29/604a3a4d7c0e1af76092e9167fbef372e0f7da055fdcfc4e1313ec21d90
/distlib-0.3.0.zip
Collecting filelock<4,>=3.0.0 (from virtualenv)
  Downloading https://files.pythonhosted.org/packages/93/83/71a2ee6158bb9f39a90c0dea1637f81dSeef866e188e1971a1biab01a35a
/filelock-3.0.0-py3-none-any.whl
Collecting six<1.14.0,>=1.12.0 (from virtualenv)
  Using cached https://files.pythonhosted.org/packages/65/e6/b1f97c97bf2390a276969c6fae1075da282f50580082d4cb10c6c51db
/six-1.14.0-py2.py3-none-any.whl
Installing collected packages: appdirs, distlib, filelock, six, virtualenv
  Running setup.py install for distlib ... done
Successfully installed appdirs-1.4.3 distlib-0.3.0 filelock-3.0.0.12 six-1.14.0 virtualenv-20.0.18
WARNING: You are using pip version 19.2.3, however version 20.0.2 is available.
It is recommended that you upgrade via the 'python -m pip install --upgrade pip' command.

C:\Users\acer>
```

Gambar 2.51: Instal virtualenv Pada CMD

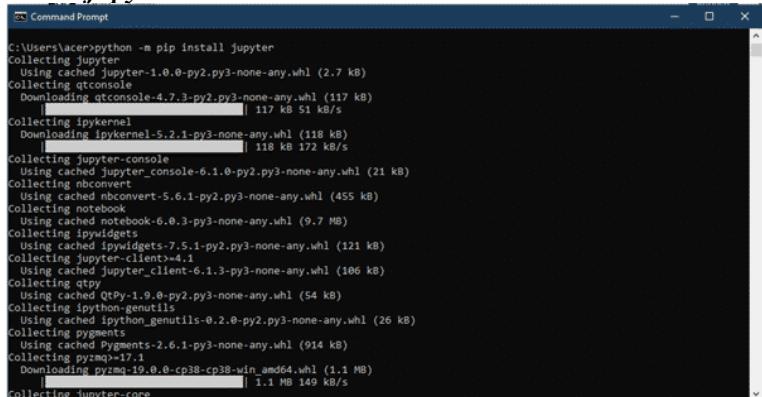
2. Jika ingin meng-*upgrade* pip ke versi terbaru gunakan ***python -m pip install -upgrade pip***.



```
C:\Users\acer>python -m pip install --upgrade pip
Collecting pip
  Using cached https://files.pythonhosted.org/packages/54/0c/d01aa759fdc501a50f431eb594a17495f15b88da142ce14b5845662c13f
  /pip-20.0.2-py2.py3-none-any.whl
Installing collected packages: pip
  Found existing installation: pip 19.2.3
    Uninstalling pip-19.2.3:
      Successfully uninstalled pip-19.2.3
Successfully installed pip-20.0.2
```

Gambar 2.52: Meng-*upgrade* pip Pada Command Prompt

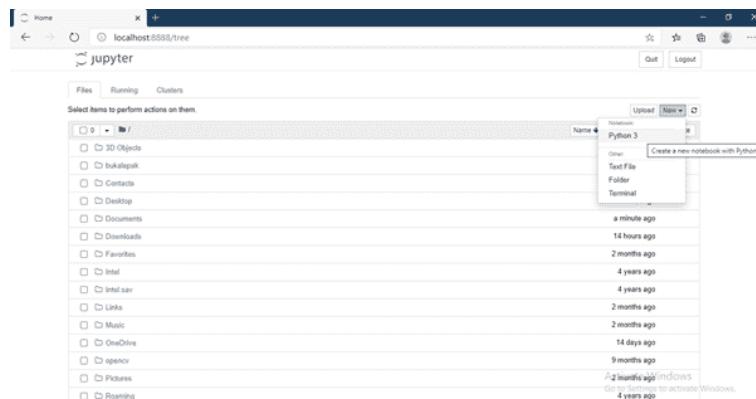
3. Kemudian *install* Jupyter Notebook menggunakan ***python -m pip install jupyter***.



```
C:\Users\laker>python -m pip install jupyter
Collecting jupyter
  Using cached jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting qtconsole
  Downloading qtconsole-4.7.3-py3-none-any.whl (117 kB) | 117 kB 51 kB/s
Collecting ipykernel
  Downloading ipykernel-5.2.1-py3-none-any.whl (118 kB) | 118 kB 172 kB/s
Collecting jupyter-console
  Using Cached jupyter_console-6.1.0-py2.py3-none-any.whl (21 kB)
Collecting nbconvert
  Using cached nbconvert-5.6.1-py2.py3-none-any.whl (455 kB)
Collecting notebook
  Using cached notebook-6.0.3-py3-none-any.whl (9.7 MB)
Collecting ipywidgets
  Using cached ipywidgets-7.5.1-py2.py3-none-any.whl (121 kB)
Collecting jupyter-client>=4.1
  Using cached jupyter_client-6.1.3-py3-none-any.whl (106 kB)
Collecting qtpy
  Using cached Qtpy-1.9.0-py2.py3-none-any.whl (54 kB)
Collecting ipython-genutils
  Using cached ipython_genutils-0.2.0-py2.py3-none-any.whl (26 kB)
Collecting pygments
  Using cached Pygments-2.6.1-py3-none-any.whl (914 kB)
Collecting pyzmq>17.1
  Downloading pyzmq-19.0.0-cp38-cp38-win_amd64.whl (1.1 MB) | 1.1 MB 149 kB/s
Collecting jupyter-core
```

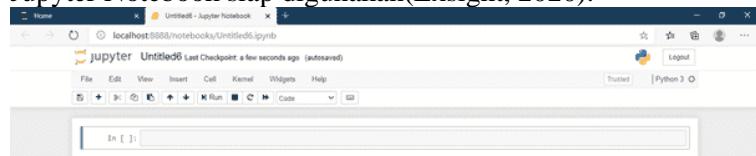
Gambar 2.53: Instal Jupyter Notebook

4. Untuk menjalankan Jupyter Notebook, ketik perintah ***jupyter notebook*** pada *Command Prompt* dan tekan *Enter*. Kemudian akan tampil jendela sebagai berikut.



Gambar 2.54: Running Jupyter Notebook Pada Web browser

5. Klik **New > Python 3** untuk membuka notebook baru. Sekarang Jupyter Notebook siap digunakan(Exsight, 2020).



Gambar 2.55: Running Python Web browser

2.4.4 Tableau



Gambar 2.56: Logo Tableau

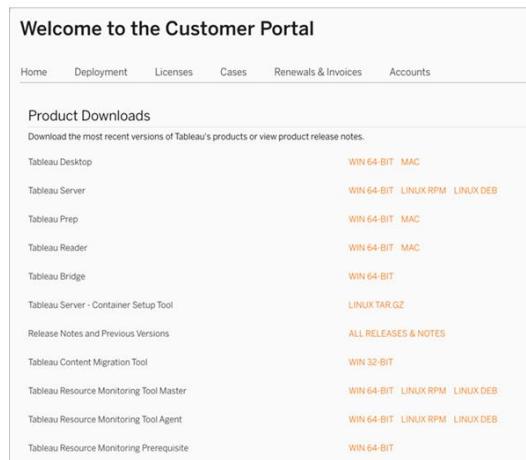
Tableau adalah *platform business intelligence* untuk membantu dalam melihat dan memahami data. *Platform* ini merupakan hasil dari proyek ilmu komputer di Stanford yang didirikan pada tahun 2003. Tujuan dibuatnya Tableau adalah untuk meningkatkan aliran analisis dan membuat data lebih dapat diakses oleh orang-orang melalui visualisasi. Berikut keunggulan menggunakan Tableau :

1. Visualisasi yang mudah dan interaktif
Keunggulan yang membuat populer Tableau adalah visualisasinya mudah, interaktif, dan menarik. Jika *tools* lain hanya menyajikan grafik biasa, *tools* ini bisa menyajikan grafik bergerak, sehingga tampilan lebih menarik.
2. *User-friendly*
Jika membuat grafik perlu adanya *coding*, Tableau tidak perlu ada *coding*. Tableau diperuntukan bagi siapapun yang bermain dengan data tapi ingin lebih sederhana dalam menyajikannya.
3. Banyak sumber data yang mampu diolah
Banyak data kompleks yang beraturan maupun tidak beraturan, dan juga berbagi sumber, Tableau bisa mengolah dari berbagai sumber. Bahkan bisa menggabungkan dari berbagai *cloud*, *spreadsheet*, dan *big data*.
4. *Dashboard mobile friendly*
Jika tidak sempat membuka laptop, Tableau bisa diakses melalui *smartphone*. Meski lewat smartphone, *user interfacenya* mudah untuk dioperasikan.
5. Terintegrasi dengan bahasa srkip(Institut Teknologi Bandung, 2022).

Berikut langkah-langkah install Tableau pada OS Windows :

1. Untuk menginstal Tableau Desktop atau Tableau Prep Builder, diperlukan dua hal yaitu kunci produk Kreator (nomor kunci dimulai dengan huruf TC) dan penginstal untuk versi yang ingin di instal.
2. Untuk versi terbaru Tableau Desktop atau Tableau Prep Builder, buka *Portal Customer*. Di bawah ***Quick Links***, klik ***Product Downloads***, kemudian klik tautan yang sesuai untuk mengunduh penginstal untuk produk dan sistem operasi.

Membuat Aplikasi Web Prediksi Gaji Semudah Menanak Nasi

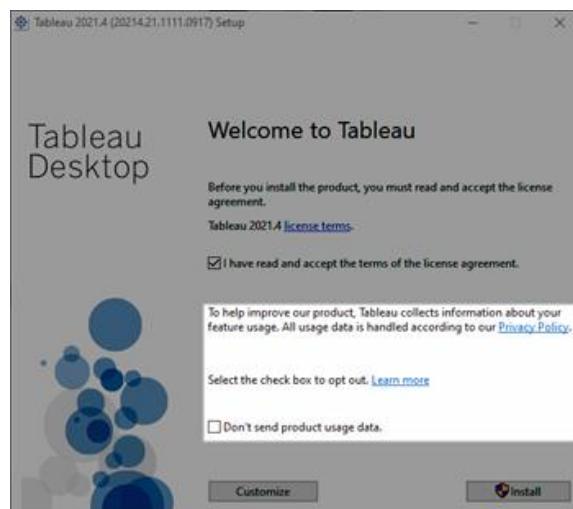


The screenshot shows the 'Welcome to the Customer Portal' page. At the top, there is a navigation bar with links: Home, Deployment, Licenses, Cases, Renewals & Invoices, and Accounts. Below the navigation bar, a section titled 'Product Downloads' is displayed. This section contains a table with two columns: product names and download links. The products listed are Tableau Desktop, Tableau Server, Tableau Prep, Tableau Reader, Tableau Bridge, Tableau Server - Container Setup Tool, Release Notes and Previous Versions, Tableau Content Migration Tool, Tableau Resource Monitoring Tool Master, Tableau Resource Monitoring Tool Agent, and Tableau Resource Monitoring Prerequisite. The download links are color-coded: WIN 64-BIT (blue), MAC (green), LINUX TAR.GZ (orange), ALL RELEASES & NOTES (yellow), WIN 32-BIT (purple), and WIN 64-BIT (blue) for the monitoring tools.

Product	Download Links
Tableau Desktop	WIN 64-BIT MAC
Tableau Server	WIN 64-BIT LINUX RPM LINUX DEB
Tableau Prep	WIN 64-BIT MAC
Tableau Reader	WIN 64-BIT MAC
Tableau Bridge	WIN 64-BIT
Tableau Server - Container Setup Tool	LINUX TAR.GZ
Release Notes and Previous Versions	ALL RELEASES & NOTES
Tableau Content Migration Tool	WIN 32-BIT
Tableau Resource Monitoring Tool Master	WIN 64-BIT LINUX RPM LINUX DEB
Tableau Resource Monitoring Tool Agent	WIN 64-BIT LINUX RPM LINUX DEB
Tableau Resource Monitoring Prerequisite	WIN 64-BIT

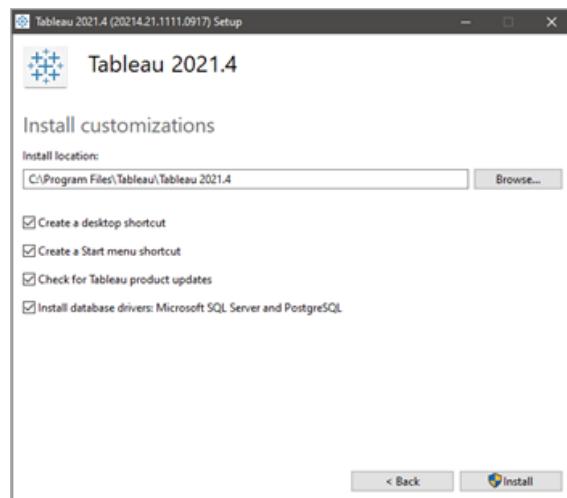
Gambar 2.57: Portal Customer Tableau

3. Setelah selesai unduh *installer* Tableau Desktop, *run installer* sebagai **administrator**. Setelah itu, untuk memilih tidak menyediakan data penggunaan, pilih kotak centang ***Don't send product usage data***.



Gambar 2.58: Penyediaan Data Penggunaan

4. Untuk menyesuaikan penginstalan, pada layar **Install Welcome Screen**, klik **Customize** dan ubah salah satu opsi berikut :



Gambar 2.59: Memilih Lokasi Penempatan Tableau

5. Setelah selesai *Customize*, klik *install* untuk memulai instalasi. Tunggu instalasi sampai selesai, dan Tableau siap digunakan(Tableau, 2022).

2.4.5 Anaconda



Gambar 2.60: Logo Anaconda

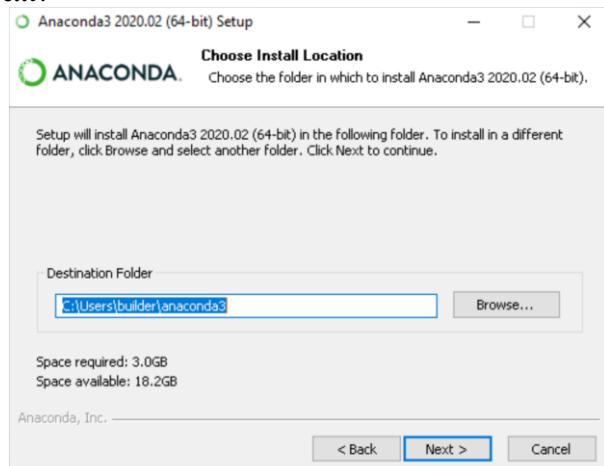
Anaconda merupakan aplikasi gratis yang memiliki banyak package dan tools atau dengan kata lain merupakan *one place solution platform* untuk *data scientist* agar memudahkan dalam mengerjakan proyek karena di dalamnya terdapat berbagai macam *software*. Anaconda bisa dikatakan sebagai paket distribusi python dari *continuum analytics* yang berisi paket *Python* ditambah beberapa paket tambahan untuk keperluan pemrograman *data science*, matematika, hingga teknik dalam satu distribusi yang *user friendly*. Berikut adalah langkah-langkah untuk menginstal Anaconda :

1. Download terlebih dahulu *installer* Anaconda pada *link* berikut :
2. Setelah menyelesaikan proses *download*, *install* Anaconda pada komputer dengan klik *installer* yang telah didownload. Setelah itu klik *Next*.



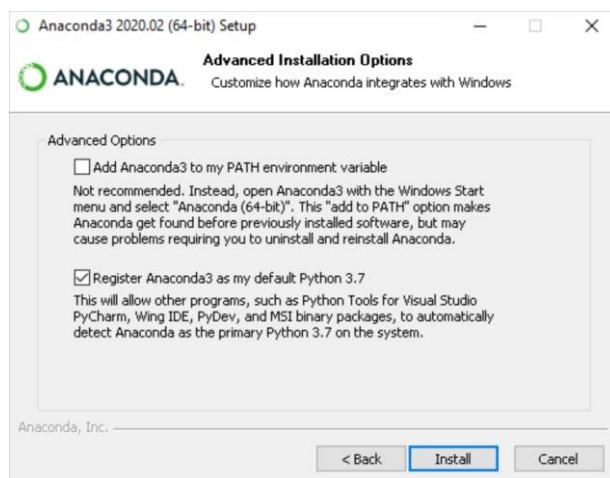
Gambar 2.61: Setup Anaconda

3. Pilih destinasi *folder* yang akan digunakan untuk tempat *software* Anaconda dengan cara klik **Browse**, kemudian pilih *folder* yang akan digunakan sebagai tempat *software* Anaconda. Setelah itu klik *Next*.



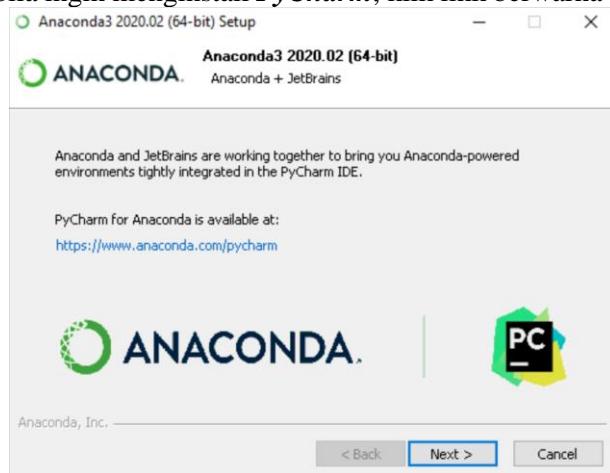
Gambar 2.62: Memilih Lokasi Penempatan Anaconda

4. Pilih Register *Anaconda3 as my default* dan klik *install*.



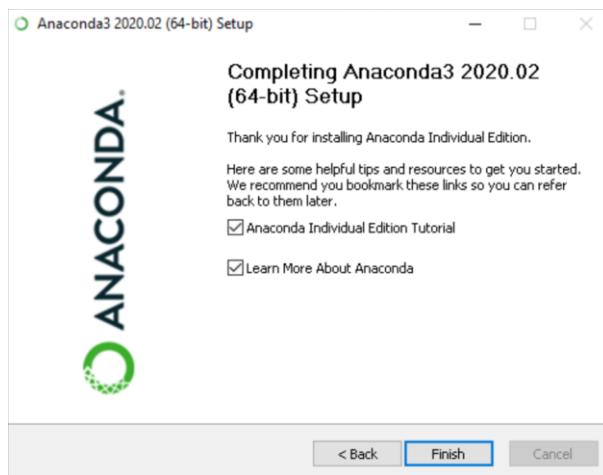
Gambar 2.63: Memulai Proses Instalasi Anaconda

5. Klik **next** dan tunggu hingga instalasi Anaconda selesai. *Optional* : Apabila ingin menginstall *PyCharm*, klik link berwarna biru.



Gambar 2.64: Opsiional Install Aplikasi PyCharm

6. Setelah itu, muncul tampilan sebagai berikut, menandakan instalasi Anaconda telah selesai dan klik **Finish**. Jika ingin membaca tentang *Anaconda Cloud* dan cara memulai dengan Anaconda, tentang kedua checkbox.



Gambar 2.65: Selesai Instalasi Anaconda

2.5 Penyimpanan

2.5.1 Database

Menurut Stephens dan Plew (2000), basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Menurut Silberschatz (2002) mendefinisikan basis data sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Menurut Ramakrishnan dan Gehrke (2003) menyatakan basis data sebagai kumpulan data, umumnya mendeskripsikan aktivitas satu organisasi atau lebih yang berhubungan. Sedangkan menurut McLeod (2001) basis data adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi(D. Tri Octafian, 2011). Dari berbagai definisi dari para ahli, dapat disimpulkan bahwa *database* atau basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Kegunaan utama sistem basis data adalah

agar pemakai mampu menyusun suatu pandangan (*view*) abstraksi data. Hal ini bertujuan untuk menyederhanakan interaksi antara pengguna dengan sistemnya dan basis data dapat mempresentasikan pandangan yang berbeda kepada para pengguna, programmer, dan administratornya.

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang , dan lain-lain. Data dinyatakan dengan nilai (angka, deretan, karakter, atau simbol). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut ;

1. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Basis data bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat sebuah basis data yang baik adalah sebagai berikut :

1. Tujuan adanya redundansi dan inkonsistensi data
Redundansi terjadi jika suatu informasi disimpan di beberapa tempat.
2. Kesulitan Pengaksesan Data
Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan *Query* ataupun dari *tool* untuk melihat tabelnya. Dengan fasilitas ini, dapat dilihat secara langsung data dari *software DBMS*-nya. Selain itu, basis data bisa dihubungkan dengan program aplikasi sehingga memudahkan pengguna dalam mengakses informasi.
3. Multiple *User*
Basis data memungkinkan pengguna data bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian *server* yang

bisa diakses ke semua pengguna dari banyak klien, hal tersebut sudah menyediakan akses ke semua pengguna dari komputer klien ke sumber informasi yaitu basis data.

Banyak manfaat yang didapat dari penggunaan basis data. Manfaat basis data diantaranya adalah :

1. Kecepatan dan kemudahan (*Speed*).

Dengan menggunakan basis data, pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar, maka penyajian informasi akan dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai.

Sebuah basis data dapat digunakan oleh banyak *user* dan banyak aplikasi. Untuk data-data yang diperlukan oleh banyak orang/bagian. Tidak perlu dilakukan pencatatan di masing-masing bagian, tetapi cukup dengan satu basis data untuk dipakai bersama.

3. Pemusatan *Control Data*.

Karena cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan di satu tempat saja.

4. Efisiensi Ruang Penyimpanan (*Space*).

Dengan pemakaian bersama, kita tidak perlu menyediakan tempat penyimpanan diberbagai tempat, tetapi cukup satu saja sehingga ini akan menghemat ruang penyimpanan data yang dimiliki oleh sebuah organisasi.

5. Keakuratan (*Accuracy*).

Penerapan secara ketat aturan tipe data, domain kata, keunikan data, hubungan antara data, dan lain-lain, dapat menekan keakuratan dalam pemasukan/penyimpanan data.

6. Ketersediaan (*Availability*).

Dengan basis data kita dapat mem-*backup* data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu kita simpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi suatu organisasi dari waktu ke waktu membutuhkan media penyimpanan yang semakin besar.

7. Keamanan (*Security*)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna diberikan hak akses yang berbeda-beda sesuai dengan pengguna dan posisinya. Basis data bisa diberikan *password*-nya untuk membatasi orang yang mengaksesnya.

8. Kemudahan dalam pembuatan program aplikasi baru

Pengguna basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS, sehingga pembuatan aplikasi tidak perlu mengurusi penyimpanan data, tetapi cukup mengatur *interface* untuk pengguna.

9. Pemakaian secara langsung

Basis data memiliki fasilitas untuk melihat datanya secara langsung dengan *tools* yang disediakan oleh DBMS. Untuk melihat data, langsung ke tabel ataupun menggunakan *query*. Biasanya yang mennggunakan fasilitas ini adalah *user* yang sudah ahli, atau *database administrator*.

10. Kebebasan data (*Data Independence*)

Jika sebuah program telah selesai dibuat, dan ternyata ada perubahan isi/struktur data. Maka dengan basis data, perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

11. *User View*

Basis data penyediaan pandangan yang berbeda-beda untuk tiap-tiap pengguna(Jatnika Hendra, 2013).

2.5.2 MySQL



Gambar 2.66: Logo MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (Bahasa Inggris: *database management system*) atau DBMS yang *multihead, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat dipakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada. Istilah semacam baris, kolom, tabel, dipakai pada MySQL. SQL sendiri merupakan suatu Bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Jadi MySQL adalah *database management system* yang menggunakan Bahasa SQL sebagai Bahasa penghubung antara perangkat lunak aplikasi dengan *database server*. MySQL sendiri sebenarnya merupakan lanjutan dari proyek UNIREG yang dikerjakan oleh Michael Montry Widenius dan TcX (perusahaan perangkat lunak asal Swedia). Sayangnya, UNIREG belum terlalu kompatibel dengan *database* dinamis yang dipakai di *website*. TcX kemudian mencari alternatif lain dan menemukan perangkat lunak yang dikembangkan oleh David Hughes, yaitu miniSQL atau mSQL. Namun, ditemukan masalah lagi karena mSQL tidak mendukung *indexing* sehingga belum sesuai dengan kebutuhan TcX. MySQL server mendukung beberapa macam sistem operasi (*cross-platform*), khususnya Linux dan Windows. Proses instalasinya pun sangat mudah. Jika menggunakan Windows, bisa menginstall XAMPP untuk menjalankan

MySQL *server* yang di dalamnya sudah terdapat juga modul untuk menjalankan Apache, PHP, FileZilla, dan Tomcat. Sedangkan di Linux, dapat menginstall MySQL secara terpisah atau menginstall LAMP (Linux, Apache, MySQL, PHP) yang sudah ada pada modul Apache dan PHP. Selain itu, XAMPP juga sudah tersedia di Linux(Hasan Bisri Isa Alfaris, Choirul Anam and Ali Masy'an, 2013).

2.6 Tinjauan Pustaka

2.6.1 Deskripsi Metode Yang Sama

1. Implementasi Model Regresi Linear Sederhana Untuk Prediksi Gaji Berdasarkan Pengalaman Lama Bekerja

Yayan Adrianova Eka Tuah, Anyan (Program Studi Pendidikan Komputer, STKIP Persada Khatulistiwa Sintang)

Suatu perusahaan tidak dapat dipisahkan dari tenaga kerja. Salah satu faktor terpenting dalam mendongkrak kinerja tenaga kerja ialah pemberian gaji atau upah yang sesuai berdasarkan waktu lama karyawan tersebut bekerja.

Penelitian ini bertujuan untuk mengetahui prediksi gaji karyawan berdasarkan tahun lama masa kerja dari seorang karyawan. Penelitian ini menggunakan metode pengumpulan data dan juga kuisioner. Teknik analisis data yaitu analisis regresi linier dalam python *machine learning* untuk mengetahui seperti apa pengaruh variabel bebas terhadap variabel terikat. Hasil dari penelitian ini ialah adanya suatu pengaruh positif dan signifikan antara gaji dengan masa kerja terhadap kinerja dari karyawan. Berdasarkan pengujian dengan menggunakan fungsi regresi linier untuk memprediksi besarnya gaji dari pengalaman bekerja, diketahui bahwa semakin bertambahnya pengalaman bekerja seorang karyawan, maka gaji karyawan tersebut juga akan bertambah setiap tahunnya. Hasil tersebut memperlihatkan bahwasannya pengalaman bekerja memberikan pengaruh positif terhadap besarnya gaji karyawan(T. Adrianova Eka and Anyan, 2020).

2. **Data Mining Menggunakan Regresi Linear Untuk Prediksi Harga Saham Perusahaan Pelayaran**

Ekka Puji Ariesanto Akhmad (Program Studi Ketatalaksanaan Pelayaran Niaga, Program Diploma Pelayaran, Universitas Hang Tuah)

Pergerakan harga penutupan saham dari PT. BULL cenderung mengalami variasi setiap harinya. Investor perlu melakukan tindakan yang tepat, sehingga risiko yang ada dapat dikurangi dengan mengetahui naik turunnya harga saham pada masa yang akan datang, dan mampu memprediksi langkah kebijakan yang optimal untuk membuat keputusan pembelian/penjualan saham yang sesuai. Tujuan penelitian ini untuk menerapkan *data mining* menggunakan regresi linier untuk prediksi harga saham pada perusahaan pelayaran. Dalam penelitian ini menggunakan metodologi berupa *Cross Industry Standard Process for Data Mining* (CRISP-DM). Hasil dari penelitian tersebut menunjukkan bahwa masih ada selisih antara harga penutupan saham luaran data testing dengan harga penutupan saham aktual yang terdapat di bursa saham. Evaluasi dari nilai *Root Mean Square Error* menunjukkan angka plus 7,522 dari data aktual harga penutupan saham periode harian PT. BULL(Ariesanto Akhmad, 2020).

3. **Machine learning Untuk Model Prediksi Harga Sembako Dengan Metode Regresi Linier Berganda**

Kandari Puteri dan Astried Silvianie (Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Institut Bisnis dan Informatika (IBI) Kosgoro 1957)

Harga sembilan bahan pokok atau yang biasa disebut sembako, setiap waktu dapat naik dan turun, serta kebutuhan akan informasi harga dari sembako harian. Oleh karena itu diperlukan suatu peramalan harga pada sembako harian untuk diketahui pada beberapa waktu ke depan. Salah satu metode yang dapat digunakan untuk melakukan prediksi harga dengan tipe numerik kontinu ialah dengan menggunakan metode regresi. Penelitian ini bertujuan untuk mengimplementasikan metode regresi linier berganda dalam memprediksi harga dari sembako. Data yang digunakan adalah menggunakan sampel data sembako di DKI Jakarta. Hasil yang didapatkan, sistem *machine learning* dapat digunakan sebagai alat bantu untuk memprediksi harga sembako harian baik harga pada

masa lampau, saat ini maupun masa yang akan datang (masa depan yang akan terjadi)(Puteri and Silvanie, 2020).

4. Prediksi Harga Rumah Menggunakan *Web Scrapping* Dan *Machine learning* Dengan Algoritma *Linear Regression*

Andi Saiful, Septi Andryana, Aris Gunaryati (Universitas Nasional; Jl. Sawo Manila No.61, Pejaten Barat, Pasar Minggu, Jakarta Selatan, DKI Jakarta 12520, Jurusan Informatika, Universitas Nasional, DKI Jakarta).

Tempat tinggal ialah kebutuhan primer yang sangat penting untuk dimiliki manusia. Maka, sangat penting dalam membuat perencanaan agar nantinya setiap keluarga dapat memiliki tempat tinggal pribadi. Dalam perencanaan tersebut dibutuhkanlah suatu prediksi harga di masa yang akan datang. Tujuan dari penelitian ini yaitu untuk membuat suatu model prediksi harga rumah menggunakan metode *machine learning* adapun algoritmanya yaitu *linear regression*. Dengan dilakukannya *web scraping* untuk mengumpulkan data, melalui beberapa *website* yang bergelut pada bidang jual beli rumah. Adapun menurut developer rumah yang ditanyakan dilapangan terkait variabel yang mempengaruhi harga rumah. Dari hasil penerapan prediksi harga rumah, disimpulkan bahwa pengolahan awal data yang dilakukan pada data set 7442 data menjadi 794 data amat mempengaruhi pada tingkat akurasi dari prediksi harga rumah tersebut. Dengan menggunakan algoritma *linear regression* untuk memprediksi harga rumah dapat memberikan hasil keakuratan prediksi harga rumah dengan baik(Saiful A, Andryana S and Gunaryati A, 2012).

5. Metode Regresi Linier Untuk Prediksi Pengadaan Inventaris Barang

Melisa Winda Pertiwi, Richardus Eko Indrajit

Teknik prediksi *data mining* pada persediaan barang diperlukan di beberapa tempat, dalam beberapa kasus ada masalah pengadaan, misalnya pada Dinas Pariwisata Pemuda Dan Olahraga Kota Tasikmalaya provinsi Jawa Barat. Hal ini disebabkan oleh kurangnya metode ilmiah untuk memprediksi pengadaan. Prediksi yang digunakan untuk pengadaan tahun depan diharapkan menghasilkan pengetahuan yang dapat digunakan untuk mengambil keputusan. Metode regresi linier merupakan salah satu metode yang dapat

memprediksi (*forecasting*). Pada *resource* ini berbasis *business intelligence* yang berfokus untuk memprediksi persediaan barang dengan menggunakan metode regresi linier berbasis pada data sebelumnya, sehingga dapat diketahui keakuratan metode regresi linier terhadap data yang digunakan oleh RMSE (*Root Mean Square Error*). Penelitian ini berbasis *business intelligence* yang mana akan muncul suatu pengetahuan baru berdasarkan proses bisnis dari studi kasus untuk melakukan peramalan khususnya pada inventaris barang. Berdasarkan uji coba dataset inventaris barang menggunakan metode regresi linear, maka dapat disimpulkan bahwa metode ini baik terhadap dataset yang digunakan dengan menunjukkan akurasi RMSE 0.94(Pertiwi and Indrajit, 2017).

6. Prediksi Kasus Covid-19 Di Indonesia Menggunakan Metode *Backpropagation* Dan Regresi Linear

Wahyudin, Heri Purwanto (Konsentrasi Teknik Informatika, Program Studi Manajemen Informatika, STMIK LPKIA Bandung) COVID-19 pertama kali menyerang manusia di Wuhan, China sedangkan di Indonesia sendiri mulai menyerang dalam lepas 2 Maret 2020 terdeteksi dua orang telah terkonfirmasi positif. Dari kasus tersebut tiap harinya mengalami pertambahan yang relatif signifikan. Hingga saat ini masih belum menemukan obat atau vaksin yang bisa digunakan untuk mengatasi penyebaran virus COVID-19. Adapun maksud dilakukannya penelitian ini yaitu untuk dapat memperkirakan jumlah kasus aktif pada penambahan kasus COVID-19 di Indonesia. Pada penelitian ini akan dicobakan dengan menggunakan metode *Backpropagation* dan Regresi Linear. Hasil prediksi kasus aktif dengan *Backpropagation* memberikan hasil penambahan dan penurunan yang tidak terlalu signifikan sedangkan hasil prediksi kasus aktif dengan Regresi Linear menunjukkan bahwa penambahan kasus untuk tiap harinya mengalami penambahan kasus aktif. Setelah penjelasan serta penjabaran pada bab sebelumnya, dapat dijelaskan bahwa, secara umum hasil prediksi kasus aktif dengan Regresi Linear menunjukkan bahwa penambahan kasus untuk tiap harinya mengalami pertambahan kasus aktif COVID-19(Wahyudin and Purwanto, 2021).

7. Perbandingan Regresi Linear, *Backpropagation* Dan Fuzzy Mamdani Dalam Prediksi Harga Emas.

Nur Nafi'iyah

Tujuan dari penelitian ini yaitu untuk melihat hasil dari regresi linear, *backpropagation* dan fuzzy mamdani dalam memprediksi harga emas. Regresi linear adalah suatu persamaan garis dari berbagai data yang dikumpulkan. Fuzzy mamdani yaitu algoritma fuzzy yang menggunakan nilai yang crips dengan rentang 0-1. Sedangkan *backpropagation* ialah suatu algoritma *neural network* yang sangat kompleks. Emas merupakan benda yang dapat digunakan untuk melakukan investasi. Sehingga apabila kita dapat memahami bagaimana pergeseran harga emas, maka kita mampu mendapatkan untung. Hasil dari ketiga metode tersebut menunjukkan bahwa korelasi yang didapatkan dari regresi linear sangat bagus yaitu 0,929. Dan nilai korelasi tertinggi dari ketiga metode berasal dari metode *backpropagation*. Hal ini dapat terbukti bahwa dalam memprediksi harga emas dengan menggunakan *backpropagation* lebih sedikit errornya $\pm 0,05$ (Nafi'iyah, 2021). [7]

8. Implementasi Algoritma Regresi Linear Sederhana Dalam Memprediksi Besaran Pendapatan Daerah (Studi Kasus : Dinas Pendapatan Kab. Deli Serdang).

Fransiskus Ginting, Efori Buulolo, Edward R Siagian.

Data Mining yaitu suatu penemuan informasi dengan cara melakukan penggalian pola informasi yang dapat berisi pencarian trend dalam sejumlah data yang sangat besar serta dapat membantu dalam proses penyimpanan data untuk mengambil suatu keputusan diwaktu yang akan datang. Dalam menentukan pola dilakukan suatu teknik klasifikasi dengan mengumpulkan *record* (*Training set*). Pendapatan daerah umumnya berasal dari pajak dan retribusi daerah. Pajak daerah merupakan salah satu dari sumber pendanaan untuk daerah secara rata-rata nasional belum mampu memberikan kontribusi yang besar dalam pembentukan pendapatan asli daerah. Dengan memanfaatkan data Pendapatan Daerah dapat menghasilkan peramalan dan prediksi penghasilan Pendapatan Daerah kedepannya agar sesuai dengan kenyataan/realitas sehingga RAPBD yang telah direncanakan dapat berjalan dengan lancar. Regresi Linear Sederhana atau SLR (*Simple Linear Regression*) adalah salah satu metode statistik yang dipergunakan dalam produksi untuk melakukan peramalan atau

prediksi mengenai karakteristik kualitas maupun kuantitas untuk menggambarkan proses yang terkait dengan pengolahan data perolehan besaran pendapatan daerah. Sehingga dalam tahap pengujian dengan *visual basic net* dapat membantu dalam mengolah data Besaran Pendapatan Daerah yang valid. Berdasarkan proses hasil perhitungan yang telah dilakukan, maka dapat diketahui nilai prediksi besaran pendapatan daerah dinas pendapatan daerah kab. Deli Serdang. Dengan menggunakan algoritma regresi linear sederhana, dinilai dapat memprediksi besaran pendapatan daerah dinas pendapatan daerah kab. deli serdang untuk tahun berikutnya sehingga program-program yang telah direncanakan sebelumnya oleh dinas pendapatan dapat berjalan dengan lancar, dan juga dapat membuat program-program yang baru agar dapat meningkatkan pendapatan daerah untuk memajukan daerah tersebut.(Ginting, Buulolo and Siagian, 2019)

9. **Prediksi Tingkat Produksi Kopi Menggunakan Regresi Linear** Petrus Katemba, Rosita Koro Djoh

Kopi dari daerah ini menjadi andalan ekspor hasil perkebunan, yang telah menembus pasar internasional dengan harga tinggi karena mutunya yang begitu baik. Namun produksi kopi cenderung menurun yang mengakibatkan permintaan akan kopi mengalami penurunan yang disebabkan oleh beberapa faktor, baik faktor alam dan sistem yang digunakan masih tradisional. Ada banyak upaya peningkatan produksi kopi telah dilakukan pemerintah, namun lemahnya teknologi pendukung menjadi salah satu kendala peningkatan produksi kopi. Tujuannya ialah untuk mengetahui apakah produksi kopi mengalami peningkatan atau justru mengalami penurunan dari waktu ke waktu. Untuk memenuhi kebutuhan kopi, maka dilakukan prediksi dengan menggunakan Regresi linear sederhana yang merupakan salah satu metode statistik yang dipergunakan dalam produksi untuk melakukan prediksi tentang karakteristik kuantitas maupun kualitas tersebut. *Simple* Regresi Linear yang terdiri dari satu buah variable bebas (x) dengan satu buah variabel terikat (y). Dengan dilakukannya prediksi menggunakan metode regresi linear, informasi dapat diberikan dan membantu para petani dan pemerintah dalam mengambil kebijakan guna meningkatkan produksi kopi di Kabupaten Manggarai. Hasil yang diperoleh dari penelitian yang dimana melibatkan 5 periode yaitu dari tahun 2011-2015, nilai

tertinggi pada tahun 2015 adalah sebesar 1.537,38 ton dan nilai terendah pada tahun 2011 adalah sebesar 1.109. Setelah dilakukannya pengujian dengan menggunakan MSE dan MAPE, diperoleh nilai MSE 43,112% dan MAPE20,001%. Sehingga pengujian menggunakan MAPE jauh lebih baik dalam menghitung akurasi prediksi produksi kopi(Katemba and Koro Djoh, 2017). [9]

10. Peramalan Penerimaan Mahasiswa Baru Universitas Samudra Menggunakan Metode Regresi Linear Sederhana.

Tri Novrita Putri, Adam Yordan, Dara Havisha Lamkaruna.

Peramalan penerimaan mahasiswa baru di Universitas Samudra menggunakan metode regresi linear sederhana. Penelitian ini bertujuan untuk meramalkan atau memprediksi jumlah penerimaan mahasiswa baru dengan menganalisis data-data yang ada pada tahun-tahun sebelumnya, yang nantinya akan dipresentasikan ke kejadian masa depan dengan model matematis, perhitungan ini dapat berupa perhitungan yang menggunakan pendekatan baik kuantitatif atau kualitatif. Data yang digunakan pada penelitian ini yaitu data penerimaan mahasiswa di Universitas Samudra pada tahun 2014, 2015, 2016, 2017 dan 2019. Pada masalah ini digunakanlah suatu metode dalam data mining yaitu regresi linear sederhana. Hasil yang didapatkan pada penelitian ini yaitu penggunaan metode regresi linear sederhana dapat dipertimbangkan karena jumlah error yang didapat pada hasil prediksi penerimaan mahasiswa baru satu tahun kedepan tidak terlalu besar yaitu hanya sebesar 21 dari 1428 mahasiswa hasil prediksi, dimana data ini diuji pada data di tahun 2018 jika diprediksi pada tahun tersebut, banyak mahasiswa yang masuk pada tahun tersebut adalah sebanyak 1449(Putri, Yordan and Lamkaruna, 2019).

11. Salary Prediction Using Regression

Sayan Das, Rupashri Barik, Ayush Mukherjee.

Tujuan dari jurnal ini adalah untuk memprediksi gaji seseorang setelah tahun tertentu. Representasi grafis dari memprediksi gaji adalah proses yang bertujuan untuk mengembangkan sistem komputerisasi untuk mempertahankan seluruh pekerjaan sehari-hari grafik pertumbuhan gaji di bidang apapun dan dapat memprediksi gaji setelah jangka waktu tertentu. Aplikasi ini dapat mengambil *database* untuk sistem penggajian dari organisasi dan membuat

grafik melalui informasi ini dari *database*. Model yang digunakan adalah model regresi linear dan regresi polinomial. Memilih gaji dari grafik x-y membutuhkan banyak percobaan karena mungkin ada lebih dari satu grafik yang sesuai. Prediksi ini benar hingga waktu tertentu. Akurasi dapat diperoleh dengan menerapkan regresi k-terdekat(das Sayan, Barik Rupashri and Mukherjee Ayush, 2020).

12. Empirical Analysis Of Regression Techniques By House Price And Salary Prediction

U. Bansal, A. Narang, A. Sachdeva, I. Kashyap, S.P. Panda.

Tujuan utama dari jurnal ini adalah untuk membandingkan kinerja dua teknik regresi yaitu algoritma Regresi Linier Sederhana (SLR) dan Regresi Linier Berganda (MLR) dengan dua kasus: memprediksi gaji karyawan setelah tahun-tahun tertentu dan memprediksi harga real estat. Gaji karyawan tergantung pada banyak faktor, seperti pengalaman total karyawan, sertifikasi, dan pengalaman keseluruhan sebagai pemimpin dan manajer. Faktor-faktor dalam memprediksi harga rumah adalah luas tanah (*sqft_living*), kondisi, tepi laut, jumlah kamar tidur, dan sebagainya. Dataset yang digunakan dalam eksperimen ini adalah *dataset open-source* dari KaggleInc. Algoritma dibandingkan menggunakan parameter seperti nilai R-kuadrat, *Mean absolute error* (MAE), *Mean Squared Error* (MSE), *Median Absolute Error* (MDAE), *Variance Score*, dan *Root Mean Square Error* (RMSE). Hasil telah menunjukkan bahwa MLR memberikan efisiensi yang lebih baik dibandingkan dengan SLR. 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(Bansal *et al.*, 2021).

13. Prediksi Kasus Covid-19 Di Indonesia Menggunakan Metode Backpropagation Dan Regresi Linear

X. Pan, X. Wan, H. Wang, dan Y. Li

Di bawah analisis tingkat psikologis pengusaha, penelitian ini berfokus pada pembahasan hubungan antara kesenjangan gaji dan efisiensi inovasi perusahaan. Ini memberikan dasar referensi untuk mempromosikan kegiatan inovasi karyawan dan meningkatkan daya saing pasar perusahaan. Dalam analisis empiris, makalah ini memilih data perusahaan manufaktur *A-share* yang terdaftar di China dari tahun 2012 hingga 2016 sebagai sampel penelitian. Variabel penjelas

termasuk gaji manajemen senior, gaji karyawan biasa dan kesenjangan gaji. Dengan menyusun model ekonometrika, penelitian ini menggunakan model regresi berganda untuk menganalisis secara empiris korelasi antara gap gaji dengan efisiensi inovasi. Koefisien regresi antara gaji moneter dan gaji ekuitas manajer senior dan kinerja inovasi perusahaan masing-masing adalah 5,545 dan 1,003. Koefisien regresi antara gaji pegawai biasa dan efisiensi inovasi perusahaan adalah 8,357. Koefisien regresi antara kesenjangan gaji internal tim manajemen senior dan efisiensi inovasi perusahaan adalah 3,552, keduanya menunjukkan korelasi positif yang signifikan pada tingkat 1%. Koefisien regresi antara kesenjangan gaji antara manajer senior dan karyawan biasa dan kinerja inovasi perusahaan adalah 3,032, yang secara signifikan berkorelasi negatif pada tingkat 5%. Perusahaan harus mengoptimalkan struktur gaji dari dua tingkat manajer senior dan karyawan biasa, untuk merangsang semangat kerja karyawan di semua tingkatan, dan mempromosikan inovasi perusahaan(Pan *et al.*, 2020).

14. Regresi Linear Bivariat Simpel Dan Aplikasinya Pada Data Cuaca Di Cilacap

Saniyah dan Budi Pratikno.

Penelitian ini bertujuan untuk membahas mengenai metode regresi linear bivariat simpel pada data cuaca di Kabupaten Cilacap untuk mengetahui hubungan unsur-unsur dalam cuaca dan iklim yaitu temperatur udara, kelembaban udara, dan curah hujan dengan model regresilinier bivariat. Model regresi linear bivariat ini dapat memprediksi dan menganalisis hubungan antar dua variabel respon sekaligus yang saling berkorelasi dan perhitungan regresi linear bivariat ini biasanya menggunakan matrik. Variabel pada model regresi linear bivariat simpel ini menggunakan dua variabel respon, yaitu variabel curah hujan dan variabel kelembaban udara suatu wilayah, dan satu variabel prediktor, temperatur udara. Metode pengujian model persamaan regresi adalah uji Wilk's Lamda, dengan nilai Wilk's Lamda = 0,881101 bernilai lebih kecil dari lambda tabel 0,903. Model peramalan untuk Y1 adalah $Y_{(1)} = -894,130 + 45,892X$ yang berarti jika tempertur udara naik sebesar satu derajat Celcius maka curah hujan akan naik sebesar 45,892mm serta nilai rata-rata

error bulanan = -0,00697mm dan $MDE_{E(1)} = 151,2132$. Model peramalan untuk Y_2 adalah $Y_{(2)} = 78,0433+0,1581X$ yang artinya jika temperatur udara naik sebesar satu derajat selsius maka kelembaban udara akan naik 0,1581 persen seta nilai rata-rata error bulanan = 0,000441 persen dan $MD_{E(2)} = 1,206636$. Hasil uji model $[Y_{(1)} \ Y_{(2)}] = [-894,130+45,892X \ 78,0433+0,1581X]$ menunjukan bahwa parameter keduanya signifikan, dengan *mean deviation error model* tersebut adalah [151, 2132 1, 206636] (Dan and Pratikno, 2014).

15. Analisis Regresi Linier Berganda Dalam Estimasi Produktivitas Tanaman Padi Di Kabupaten Karawang

Tesa Nur Padilah dan Riza Ibnu Adam.

Karawang yang sebagai pusat penanaman padi, memang sudah seharusnya produktivitas padi dari tahun ke tahun mengalami peningkatan. Sayangnya, produktivitas padi ternyata tidak konsisten, hal itu dibuktikan ketika tahun 2015 mengalami kenaikan sedangkan tahun 2016 mengalami penurunan. Oleh karena itu, diperlukannya suatu estimasi sehingga dapat diketahui bagaimana produktivitas padi untuk tahun-tahun berikutnya. Oleh karena itu, masalah produktivitas padi di Kabupaten Karawang dapat diestimasi dengan menggunakan analisis regresi linier berganda. Model regresi linier berganda dapat dinyatakan dalam bentuk perkalian matriks. Selanjutnya, perhitungan nilai-nilai koefisien regresi dapat dicari dengan menggunakan eliminasi Gauss. Data yang digunakan adalah data perbulan yang berjumlah 57data. Variabel terikat pada penelitian ini adalah produktivitas padi (kw/ha). Berdasarkan model regresi didapat 80,46% faktor-faktor produktivitas padi. Variabel-variabel yang mempengaruhi peningkatan jumlah produktivitas padi yaitu variabel produksi dan curah hujan, sedangkan variabel-variabel yang mempengaruhi penurunan jumlah produktivitas yaitu variabel luas panen, luas tanam, dan hari hujan. kesalahan relatif regresi yang diperoleh yaitu 0,04642 atau 4,642%(Padilah and Adam, 2019).

16. Model Regresi Multivariat Analisis Kesejahteraan Pedagang Kaki Lima Berdasarkan Karakteristik Sosial Ekonomi

Desak Putu Eka Nilakusmawati dan Made Susilawati.

Kurangnya penelitian terhadap pekerja sektor *informal*, khususnya PKL telah membuat diremehkannya kontribusi sektor ini terhadap pertumbuhan ekonomi dan pembangunan berkelanjutan. Sehingga penelitian ini dapat memberikan kontribusi untuk mengisi kesenjangan tersebut. Penelitian ini memiliki suatu tujuan yaitu untuk mengetahui bagaimana model tingkat kesejahteraan pedagang kaki lima berdasarkan karakteristik sosial ekonomi. Data dalam penelitian ini merupakan data kuantitatif yang diperoleh dari sumber primer, yaitu diambil secara langsung oleh peneliti menggunakan angket dan pengambilan sampel dilakukan dengan cara *purposive sampling*. Model yang digunakan dalam penelitian adalah analisis *multivariate* yaitu analisis regresi linier dan *logistic ordinal*. Variabel yang signifikan berpengaruh terhadap rata-rata pendapatan pedagang kaki lima adalah tingkat pendidikan, curahan jam kerja, dan jumlah tenaga kerja diluar tenaga kerja keluarga yang ikut membantu. Model rata-rata pendapatan pedagang kaki lima adalah $Y = -1.982 + 0.654\text{pendidikan} + 0.134\text{curahan jam kerja} + 0.817\text{Jumlah tenaga kerja non keluarga}$. 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 $G(x) = -16.308 - 0.519\text{ status kawin} + 0,739\text{ sifat layanan} + 1,19663\text{ curahan jam kerja} - 1,062\text{ Jumlah tenaga kerja non keluarga}$ (Nilakusmawati and Made, 2016).

Bab III

Gambaran Objek Studi

3.1 Material

Dalam proses melakukan penelitian ini, digunakan suatu material yang dapat mendukung penelitian yang dilakukan. Material yang dimaksud ialah berupa data yang dapat digunakan pada penelitian.

Data yang digunakan dalam penelitian merupakan data kepegawaian yang telah dipisah dalam bentuk *data train* dan *data test*. Data tersebut didapatkan melalui situs bernama Kaggle yang dapat dikunjungi melalui *link* yang dicantumkan sebagai berikut:

<https://www.kaggle.com/colearninglounge/employee-attrition>.

Data bersifat tidak berlabel dan data diolah dengan teknik *Unsupervised Learning* untuk mendapatkan hasil model prediksi dengan menggunakan metode regresi linier berganda. Berikut adalah pemaparan data yang digunakan.

1. Data kepegawaian berupa *data train*

Data kepegawaian berupa *data train* ini memiliki jumlah kolom sebanyak 35 kolom dan jumlah total *record* sebanyak 1029 *record*.

Kolom tersesbut diantaranya adalah sebagai berikut.

1. Kolom ke-1 yaitu Age
2. Kolom ke-2 yaitu Attrition
3. Kolom ke-3 yaitu BusinessTravel
4. Kolom ke-4 yaitu DailyRate
5. Kolom ke-5 yaitu Department
6. Kolom ke-6 yaitu DistanceFromHome
7. Kolom ke-7 yaitu Education
8. Kolom ke-8 yaitu EducationField
9. Kolom ke-9 yaitu EmployeeCount
10. Kolom ke-10 yaitu EmployeeNumber

11. Kolom ke-11 yaitu EnvironmentSatisfaction
12. Kolom ke-12 yaitu Gender
13. Kolom ke-13 yaitu HourlyRate
14. Kolom ke-14 yaitu JobInvolvement
15. Kolom ke-15 yaitu JobLevel
16. Kolom ke-16 yaitu JobRole
17. Kolom ke-17 yaitu JobSatisfaction
18. Kolom ke-18 yaitu MaritalStatus
19. Kolom ke-19 yaitu MonthlyIncome
20. Kolom ke-20 yaitu MonthlyRate
21. Kolom ke-21 yaitu NumCompaniesWorked
22. Kolom ke-22 yaitu Over18
23. Kolom ke-23 yaitu OverTime
24. Kolom ke-24 yaitu PercentSalaryHike
25. Kolom ke-25 yaitu PerformanceRating
26. Kolom ke-26 yaitu RelationshipSatisfaction
27. Kolom ke-27 yaitu StandardHours
28. Kolom ke-28 yaitu StockOptionLevel
29. Kolom ke-29 yaitu TotalWorkingYears
30. Kolom ke-30 yaitu TrainingTimesLastYear
31. Kolom ke-31 yaitu WorkLifeBalance
32. Kolom ke-32 yaitu YearsAtCompany
33. Kolom ke-33 yaitu YearsInCurrentRole
34. Kolom ke-34 yaitu YearsSinceLastPromotion
35. Kolom ke-35 yaitu YearsWithCurrManager

Berikut adalah *data training* yang digunakan dalam penelitian.

Tabel 3.1: Dataset *data training*

Age	Attrition	...	YearsWithCurrManager
50	No		3
36	No		1
21	Yes		0
50	No		7
...

41	No		2
22	Yes		0
29	No		3
50	No		0

2. Data kepegawaian berupa *data test*

Data kepegawaian berupa *data test* ini memiliki jumlah kolom sebanyak 34 kolom dan jumlah total *record* sebanyak 441 *record*.

Kolom tersebut diantaranya adalah sebagai berikut.

1. Kolom ke-1 yaitu Age
2. Kolom ke-2 yaitu BusinessTravel
3. Kolom ke-3 yaitu DailyRate
4. Kolom ke-4 yaitu Department
5. Kolom ke-5 yaitu DistanceFromHome
6. Kolom ke-6 yaitu Education
7. Kolom ke-7 yaitu EducationField
8. Kolom ke-8 yaitu EmployeeCount
9. Kolom ke-9 yaitu EmployeeNumber
10. Kolom ke-10 yaitu EnvironmentSatisfaction
11. Kolom ke-11 yaitu Gender
12. Kolom ke-12 yaitu HourlyRate
13. Kolom ke-13 yaitu JobInvolvement
14. Kolom ke-14 yaitu JobLevel
15. Kolom ke-15 yaitu JobRole
16. Kolom ke-16 yaitu JobSatisfaction
17. Kolom ke-17 yaitu MaritalStatus
18. Kolom ke-18 yaitu MonthlyIncome
19. Kolom ke-19 yaitu MonthlyRate
20. Kolom ke-20 yaitu NumCompaniesWorked
21. Kolom ke-21 yaitu Over18
22. Kolom ke-22 yaitu OverTime
23. Kolom ke-23 yaitu PercentSalaryHike
24. Kolom ke-24 yaitu PerformanceRating
25. Kolom ke-25 yaitu RelationshipSatisfaction
26. Kolom ke-26 yaitu StandardHours

27. Kolom ke-27 yaitu StockOptionLevel
28. Kolom ke-28 yaitu TotalWorkingYears
29. Kolom ke-29 yaitu TrainingTimesLastYear
30. Kolom ke-30 yaitu WorkLifeBalance
31. Kolom ke-31 yaitu YearsAtCompany
32. Kolom ke-32 yaitu YearsInCurrentRole
33. Kolom ke-33 yaitu YearsSinceLastPromotion
34. Kolom ke-34 yaitu YearsWithCurrManager

Berikut adalah data testing yang digunakan dalam penelitian

Tabel 3.2: Dataset *data testing*

Age	BusinessTravel	...	YearsWithCurrManager
	Travel_Rarely		3
53	Travel_Rarely		3
24	Travel_Rarely		0
45	Travel_Rarely		0
...
27	Non-Travel		4
	Travel_Rarely		2
39	Travel_Rarely		4
	Travel_Rarely		0

3.2 Teknologi

Teknologi yang digunakan untuk penelitian ini adalah penggunaan teknologi *hardware* dan *software* sebagai pendukung utama dalam membangun sistem dan penelitian yang dilakukan. *Software* dan *hardware* yang digunakan sesuai dengan kebutuhan dan digunakan berdasarkan fungsinya masing-masing.

Adapun spesifikasi teknologi yang digunakan adalah sebagai berikut:

1. *Software*

Tabel 3.3: Spesifikasi *software* yang digunakan

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

2. *Hardware*

Tabel 3.4: Spesifikasi *hardware* yang digunakan

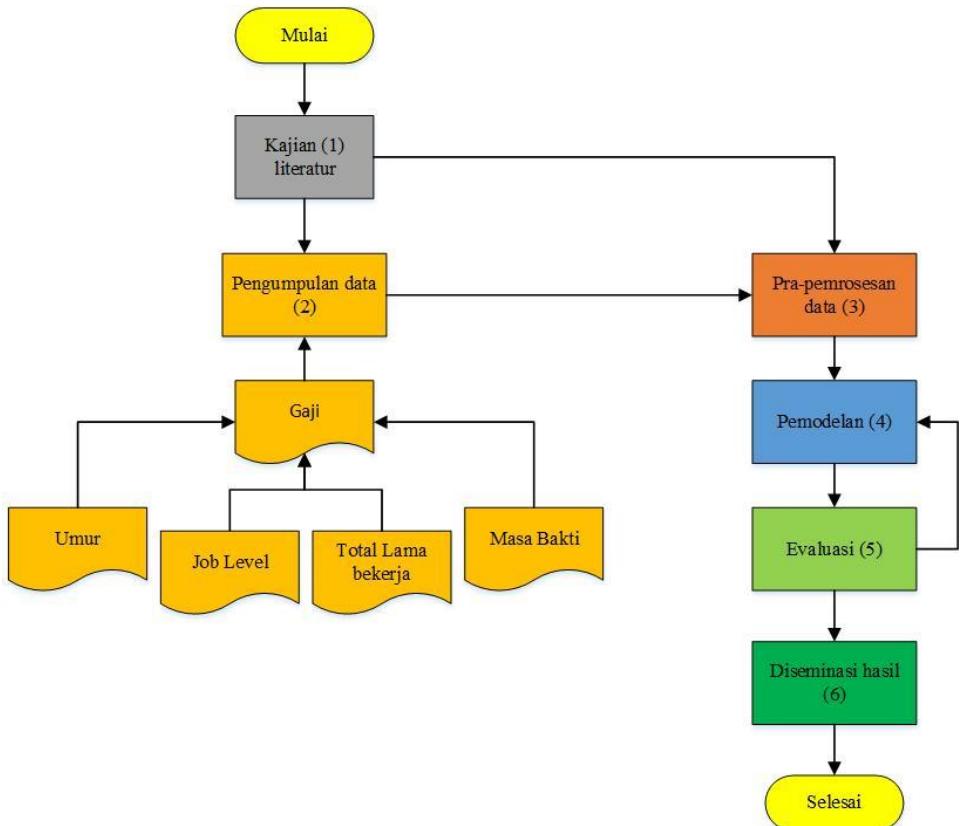
No	Nama Perangkat	Spesifikasi	Keterangan
1	Hardisk	200 GB	Media untuk menyimpan data aplikasi yang dibuat
2	Memory	4 GB	<i>Memory System</i> yang digunakan
3	Processor	AMD A4-9120 RADEON R3, 4 COMPUTE CORES 2C+2G 2.20 GHz	Untuk kecepatan transfer data dari sistem yang sangat bergantung pada kecepatan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

Bab IV

Metodologi Penelitian

4.1 Diagram Alur Metodologi Penelitian

4.1.1 Diagram Alur Metodologi Penelitian Penyelesaian Masalah



Gambar 4.1: Diagram Alur Metodologi Penelitian

Berdasarkan diagram alur metodologi penelitian diatas, terdapat indikator capaian sebagai berikut.

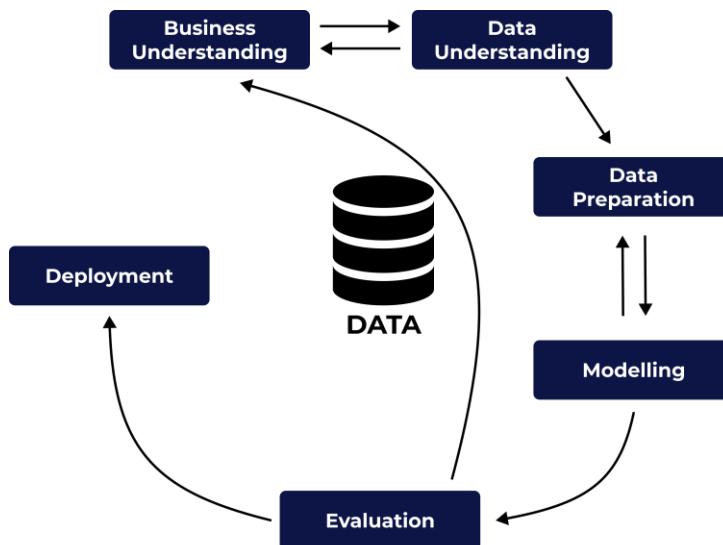
Tabel 4.1: Indikator capaian berdasarkan metodologi penelitian

No.	Tahapan		Indikator capaian
1.	Identifikasi dan perumusan masalah	→	1. <i>Mind map</i> prediksi gaji terhadap faktor-faktor spesifik berdasarkan data dan pada metode <i>machine learning</i> serta <i>framework django</i> .
2.	Studi literatur	→	2. <i>Mind map</i> prediksi gaji terhadap faktor-faktor spesifik berdasarkan data dan pada metode <i>machine learning</i> serta <i>framework django</i> .
3.	Pengumpulan data	→	3. Data mentah dari berbagai faktor (umur, job level, total lama bekerja, masa bakti)
3.	Pre-pemrosesan data	→	4. <i>Pre-processed data</i> yang sudah siap untuk pemodelan dengan tahapan pembersihan, penanganan nilai yang hilang dan transformasi.
4.	Pemodelan	→	5. Model <i>Multivariate Linier Regression</i> digunakan untuk memprediksi gaji berdasarkan data dari

			setiap faktor-faktor spesifik.
5.	Evaluasi	→	6. Performansi model
6.	Diseminasi hasil	→	7. Artikel yang diterbitkan dalam jurnal nasional terakreditasi SINTA 3, HAKI dan Buku

4.1.2 Diagram Alur Metodologi Pengolahan Data

Diagram alur metodologi pengolahan data yang digunakan ialah menggunakan metodologi CRISP-DM (*Cross Industry Standard Process for Data Mining*). Metodologi penelitian CRISP-DM digunakan dalam penelitian karena metodologi ini mampu menjabarkan setiap langkah pengolahan data dengan jelas serta mudah dipahami untuk diimplementasikan.



Gambar 4.2: Diagram Alur Metodologi Pengolahan Data

Berdasarkan diagram alur metodologi penelitian diatas, dapat dipaparkan sebagai berikut.

1. *Business Understanding*

Pada tahap ini merupakan tahap untuk menentukan seperti apa proses bisnis yang akan dibangun untuk melakukan hal apa yang ingin dicapai. Selain itu, dalam tahap ini juga diperlukan analisis apa saja alat dan bahan yang akan diperlukan untuk setiap fase proyek. Dalam hal ini, proses bisnis yang dilakukan pada penelitian adalah memprediksi gaji berdasarkan faktor-faktor spesifik. Prediksi gaji tersebut akan menggunakan data kepegawaian. Alat yang digunakan adalah berupa perangkat keras berupa laptop. Perangkat lunak yang digunakan ialah Jupyter Notebook, Anaconda, serta Microsoft Excel.

2. *Data Understanding*

Pada tahap *data understanding*, hal yang dilakukan adalah melakukan pemahaman bisnis yang berfokus untuk mengidentifikasi, mengumpulkan dan menganalisis kumpulan data untuk mencapai tujuan bisnis. Dalam hal ini, proses yang dilakukan ialah mengumpulkan data awal yang berasal dari Kaggle. Setelah data ditemukan, kemudian hal yang dilakukan adalah menjelajahi data dan melakukan verifikasi kualitas data (seberapa bersih atau otornya data yang digunakan).

3. *Data Preparation*

Tahap *data preparation* dilakukan untuk menyiapkan data akhir untuk dilakukannya pemodelan dengan cara memperbaiki masalah yang terdapat pada data. Hal yang dilakukan dalam penelitian ini adalah membersihkan data, serta memilih data yang akan digunakan dalam pemodelan. Sehingga nantinya akan didapatkan data akhir yang siap dimodelkan.

4. *Modelling*

Tahap *modelling* atau pemodelan adalah menentukan algoritma teknik pemodelan yang akan digunakan untuk menyelesaikan masalah yang ada. Dalam penelitian ini, model yang digunakan adalah menggunakan model regresi linier berganda. Karena model ini adalah model yang cocok untuk digunakan dalam melakukan prediksi terutama jika variabel yang mempengaruhi sangat banyak.

5. *Evaluation*

Tahap *evaluation* lebih berfokus untuk melihat apakah pemodelan yang dilakukan memiliki hasil yang baik atau malah sebaliknya. Pada tahap ini, evaluasi model dilakukan dengan menggunakan pengujian asumsi untuk melihat hasil-hasil evaluasi dari model.

6. *Deployment*

Yang terakhir ialah tahap *deployment*. Dimana pada tahap ini, hasil model penelitian yang telah dilakukan akan dikembangkan dan didokumentasikan. Pada tahap ini, hal yang dilakukan dalam penelitian yaitu mengembangkan model ke dalam bentuk aplikasi sehingga lebih mudah digunakan dalam melakukan proses prediksi. Selain itu, penelitian ini juga dibuatkan dalam bentuk laporan akhir untuk didokumentasikan.

Bab V

Pembuatan Aplikasi

5.1 Analisis

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.

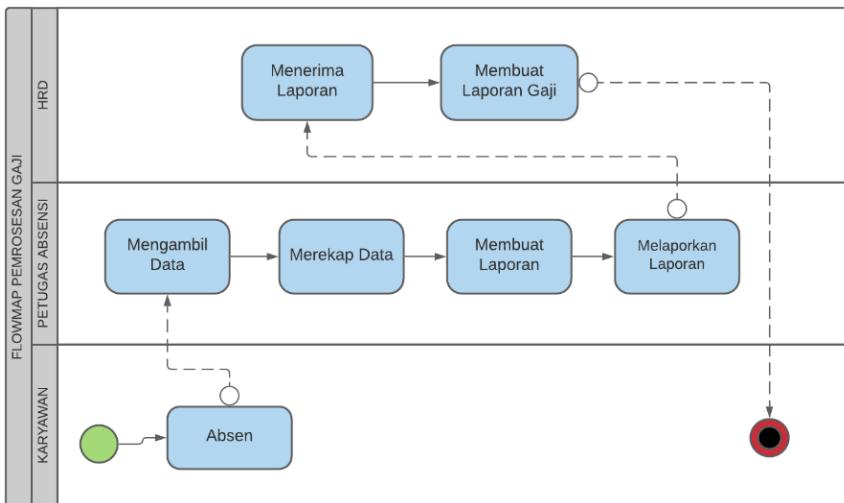
Perancangan sistem merupakan kegiatan merancang dan mendesain suatu sistem yang baik yang dimana kegiatan tersebut adalah langkah-langkah operasi dalam proses pengolahan data dan prosedur-prosedur untuk mendukung operasi sistem tersebut. Tujuan dari perancangan sistem ialah untuk memenuhi kebutuhan para 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.

5.1.1 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.

Berikut adalah *flowmap* mengenai sistem yang sedang berjalan proses prediksi gaji.



Gambar 5.1: Flowmap Pemrosesan Gaji

Keterangan :

1. Karyawan melakukan absensi setiap masuk kerja.
2. Petugas mengambil data karyawan yang masuk kerja dengan melakukan absensi.
3. Petugas merekap data kepegawaian karyawan yang bekerja pada perusahaan.
4. Petugas membuat laporan berdasarkan data kepegawaian yang ada.
5. Petugas melaporkan laporan yang telah dibuat sebelumnya.
6. HRD akan menerima laporan yang telah dibuat oleh petugas absensi.
7. HRD akan membuat laporan gaji berdasarkan laporan yang telah dibuat untuk nantinya akan digunakan dalam penggajian karyawan.

5.1.2 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*.

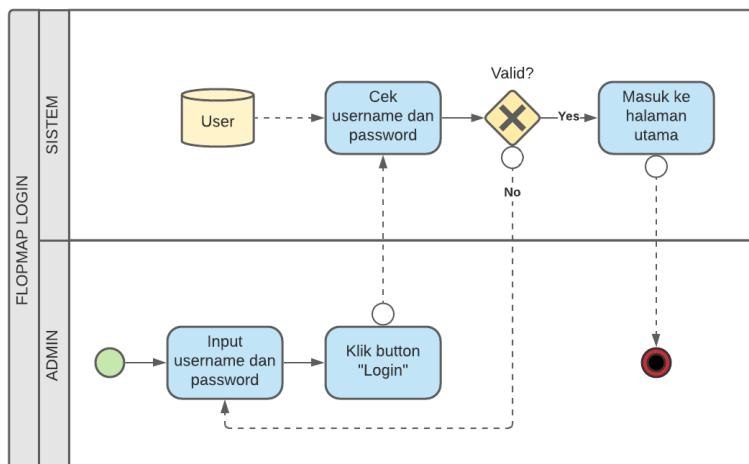
1. *Flowmap Login*

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.

Pada *flowmap* ini, terdapat dua pelaku dalam kegiatan, diantaranya adalah:

- Admin sebagai pengguna sistem.
- Sistem sebagai komponen informasi yang akan digunakan.

Berikut adalah *flowmap login* yang akan dibangun dalam pembuatan aplikasi prediksi gaji.

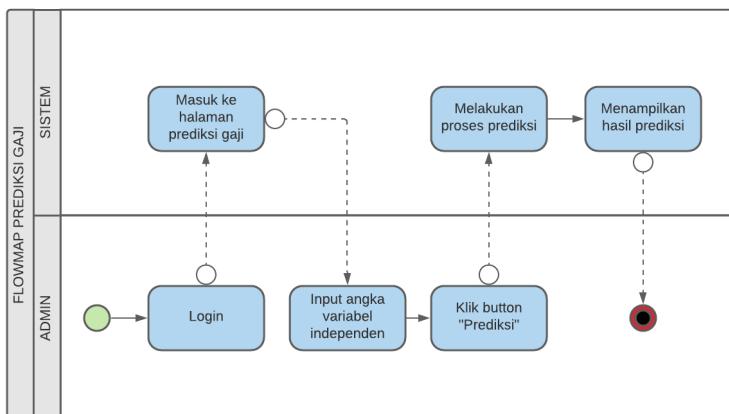


Gambar 5.2: *Flowmap* Proses *Login* Admin

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*.
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.
- Pada *flowmap* ini, terdapat dua pelaku dalam kegiatan, diantaranya adalah:
- a. Admin sebagai pengguna sistem.
 - b. Sistem sebagai komponen informasi yang akan digunakan.

Berikut adalah *flowmap* proses prediksi gaji pegawai yang akan dibangun dalam pembuatan aplikasi prediksi gaji.



Gambar 5.3: *Flowmap* Proses Prediksi Gaji Pegawai

Keterangan :

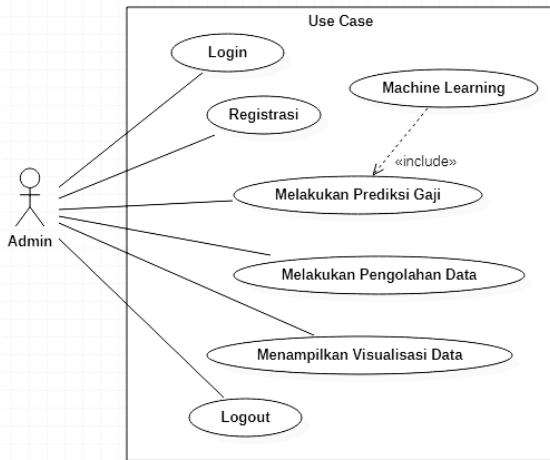
1. Admin dapat melakukan *login*.
2. Setelah admin berhasil *login*, aplikasi akan menampilkan halaman *dashboard*.
3. Setelah itu, admin dapat memilih menu Prediksi untuk beralih ke halaman prediksi.
4. Pada halaman prediksi, admin dapat menginputkan angka berupa variabel independen (usia, level pekerjaan,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

5.2 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.

5.2.1 Use case Diagram



Gambar 5.4: 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 diluar sistem (aktor) dan apa yang harus dilakukan oleh sistem yang sedang dikembangkan.

A. Definisi Aktor

Pada bagian ini akan dijelaskan aktor-aktor yang terlibat dalam Sistem.

Tabel 5.1: 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>

B. Definisi *Use case*

Tabel 5.2: 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 <i>insert</i> , <i>read</i> , <i>update</i> dan <i>delete</i> 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 skenario yang akan menjelaskan secara detail interaksi yang ada di dalamnya.

Tabel 5.3: 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
Membuat model prediksi	Melakukan olah data gaji karyawan
Kondisi Akhir	Masuk pada aplikasi

Tabel 5.4: 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
Input <i>username</i> dan <i>password</i>	Mengecek data dan menentukan aktor tersebut admin atau <i>user</i>

Kondisi Akhir	Masuk pada aplikasi
---------------	---------------------

Tabel 5.5: Skenario *Use case* Registrasi

Identifikasi	
Nomor	1
Nama	Registrasi
Tujuan	Membuat akun admin
Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Input <i>username</i> , dan <i>password</i>	Menyimpan data admin atau <i>user</i>
Kondisi Akhir	Masuk pada halaman <i>login</i>

Tabel 5.6: Skenario *Use case* Melakukan Prediksi Gaji Karyawan

Identifikasi	
Nomor	1
Nama	Melakukan Prediksi Gaji Karyawan
Tujuan	Melakukan Prediksi Gaji Karyawan

Deskripsi	
Aktor	Admin
Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Menginput Data	Melakukan Prediksi Gaji
Kondisi Akhir	Admin dapat memprediksi gaji

Tabel 5.7: 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
Input data karyawan	Menampilkan grafik visualisasi data karyawan
Kondisi Akhir	Admin dapat memvisualisasikan data

Tabel 5.8: 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
<i>Insert, read, update, dan delete</i> data karyawan.	Mengubah data karyawan yang diubah oleh admin
Kondisi Akhir	Admin dapat mengelola data

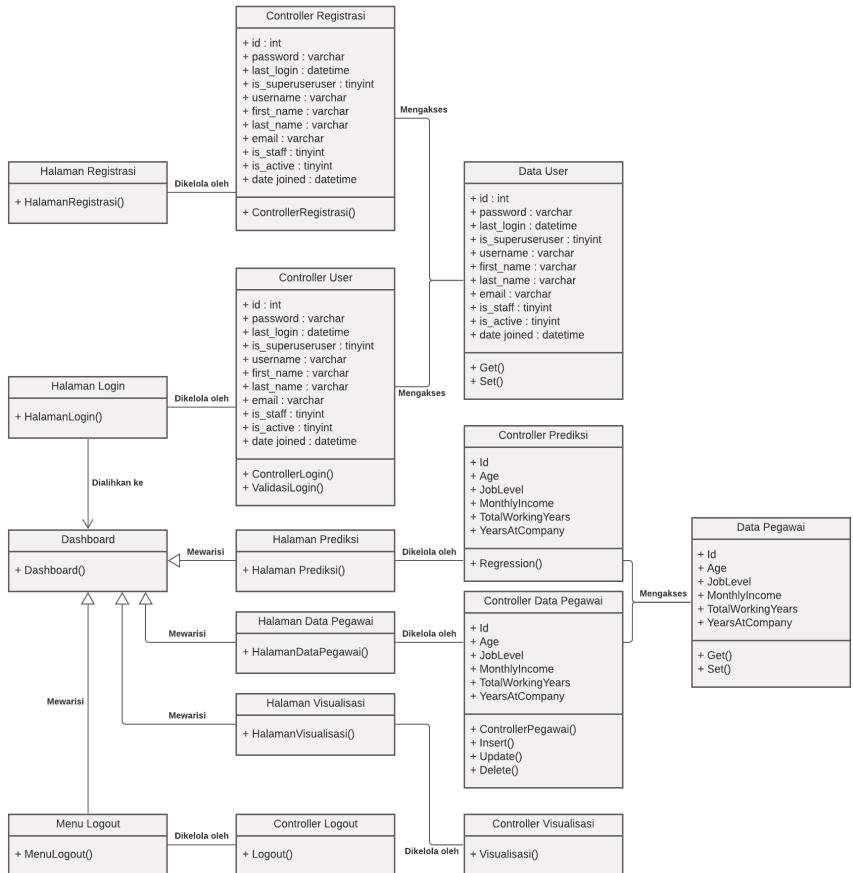
Tabel 5.9: Skenario *Use case* Logot Admin

Identifikasi	
Nomor	1
Nama	<i>Logout</i>
Tujuan	Keluar dari aplikasi
Deskripsi	
Aktor	Admin

Skenario Utama	
Kondisi Awal	
Aksi Aktor	Reaksi Sistem
Menekan tombol <i>Logout</i>	Melakukan proses keluar dari aplikasi
Kondisi Akhir	Keluar dari aplikasi

5.2.2 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 5.5: Class diagram

5.2.3 Sequence Diagram

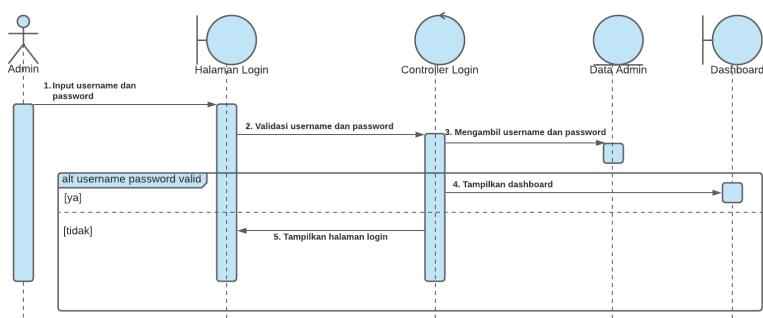
Sequence diagram adalah diagram yang mendeskripsikan serta menjelaskan suatu interaksi objek yang berdasarkan urutan waktu. Interaksi tersebut diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. *Sequence diagram* 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. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. Boundary Class : Halaman Login dan Dashboard
3. Control Class: Controller Login
4. Entity class : Data Admin
5. Percabangan ya dan tidak pada pesan objek controllerlogin
6. Pesan Objek : Lima Pesan Objek
7. Pesan : Terdapat lima pesan yang dapat dipaparkan sebagai berikut.
 - a. Input *username* dan *password*
 - b. Validasi *username* dan *password*
 - c. Mengambil *username* dan *password*
 - d. Tampilkan *dashboard*
 - e. Tampilkan halaman *login*

Berikut *Sequence diagram Login Admin* pada gambar.



Gambar 5.6: Sequence diagram Login Admin

Keterangan :

1. Admin menginputkan *username* dan *password* di halaman *login*.
2. Validasi *username* dan *password* diteruskan ke *Controller login*.
3. *Controller login* mengambil data ke *database data admin* untuk validasi *login*.
4. Jika *username* dan *password* valid pada data admin maka akan ditampilkan halaman *dashboard*.

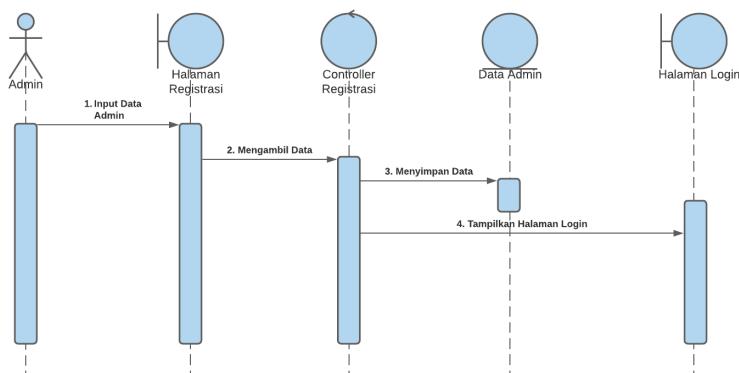
5. Jika *username* dan *password* tidak valid pada data admin maka akan ditampilkan halaman *login*.

B. *Sequence diagram* Melakukan Registrasi

Berikut ini merupakan *Sequence diagram* Registrasi Admin menjelaskan hubungan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. *Boundary Class* : Halaman Registrasi dan Halaman *Login*
3. *Control Class*: *Controller Registrasi*
4. *Entity class* : Data Admin
5. Pesan Objek : Lima Pesan Objek
6. Pesan : Terdapat empat pesan yang dapat dipaparkan sebagai berikut.
 - Input data Admin
 - Menambil Data
 - Menyimpan Data
 - Tampilkan Halaman *Login*

Berikut *Sequence diagram* Melakukan Registrasi pada gambar.



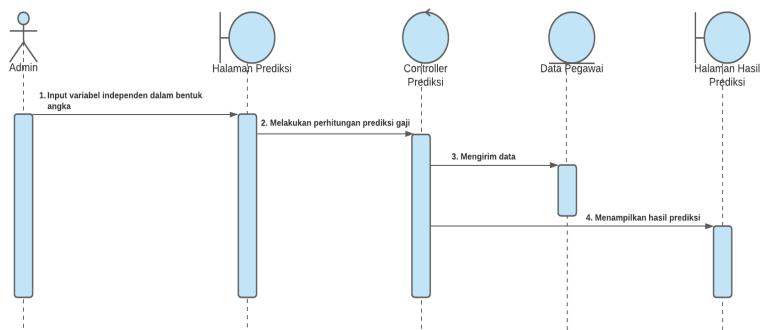
Gambar 5.7: *Sequence diagram* Registrasi

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 prediksi gaji menjelaskan hubungan antara admin dan aplikasi. Admin dapat melakukan prediksi gaji dengan cara menginput data. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.
1. Aktor : Admin
 2. *Boundary Class* : Halaman Prediksi dan Halaman Hasil Prediksi
 3. *Control Class*: *Controller Prediksi*
 4. *Entity class* : Data Pegawai
 5. Pesan Objek : Lima Pesan Objek
 6. Pesan : Terdapat empat pesan yang dapat dipaparkan sebagai berikut.
 - a. Input variabel independen dalam bentuk angka
 - b. Melakukan perhitungan prediksi gaji
 - c. Mengirim data
 - d. Menampilkan hasil prediksi

Berikut *Sequence diagram* Melakukan Prediksi Gaji pada gambar.

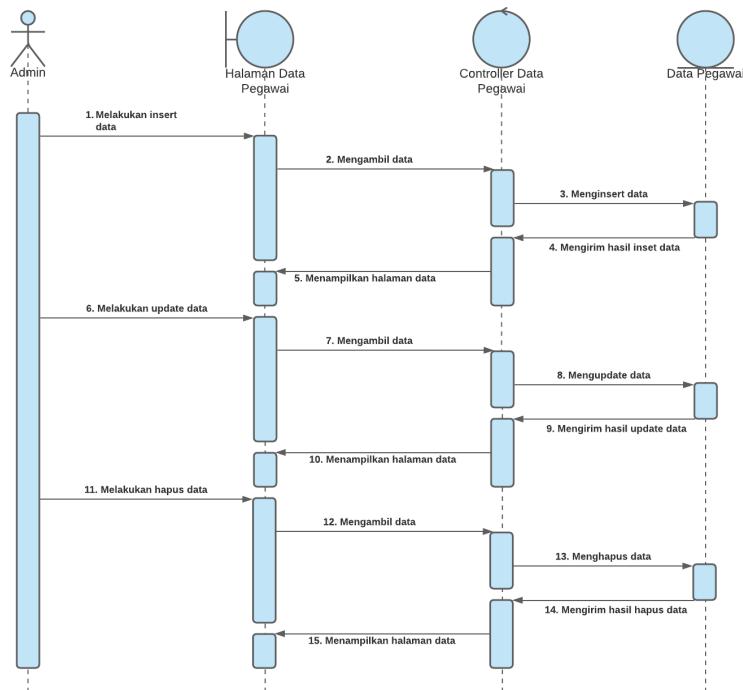


Gambar 5.8: *Sequence diagram* Melakukan Prediksi Gaji

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. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.
1. Aktor : Admin
 2. *Boundary Class* : Halaman Data Pegawai
 3. *Control Class*: *Controller* Data Pegawai
 4. *Entity class* : Data Pegawai
 5. Pesan Objek : Enam belas Pesan Objek
 6. Pesan : Terdapat lima belas pesan yang dapat dipaparkan sebagai berikut.
 - a. Melakukan *insert* data
 - b. Mengambil data (untuk *insert*)
 - c. Meng-*insert* data
 - d. Mengirim hasil *insert* data
 - e. Menampilkan halaman data (setelah *insert* data)
 - f. Meng-*update* data
 - g. Mengambil data (untuk *update*)
 - h. Meng-*update* data
 - i. Mengirim hasil *update* data
 - j. Menampilkan halaman data (setelah *update* data)
 - k. Melakukan hapus data
 - l. Mengambil data (untuk hapus)
 - m. Menghapus data
 - n. Mengirim hasil hapus data
 - o. Menampilkan halaman data (setelah hapus data)

Berikut *Sequence diagram* melakukan pengolahan data pada gambar.



Gambar 5.9: Sequence diagram Melakukan Pengolahan Data

Keterangan :

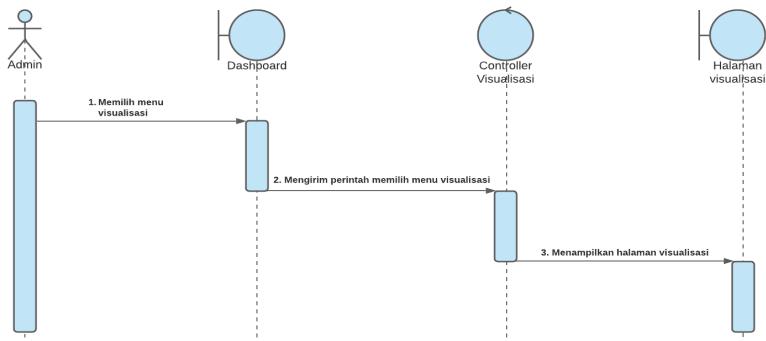
1. Admin memilih menu yang akan dikelola pada menu dihalaman data pegawai. Menu yang ada yaitu tambah data, *update* data, dan hapus data.
2. Halaman data pegawai mengambil data (baik antara input, *update* atau hapus) untuk diteruskan ke controller data pegawai.
3. Controller data pegawai mengirim hasil 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. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. *Boundary Class* : Dashboard
3. *Control Class*: Controller Visualisasi
4. Pesan Objek : Empat Pesan Objek
5. Pesan : Terdapat tigas pesan yang dapat dipaparkan sebagai berikut.
 - a. Memilih menu visualisasi
 - b. Mengirim perintah memilih menu visualisasi
 - c. Menampilkan halaman visualisasi

Berikut *Sequence diagram* Melakukan Visualisasi Data pada gambar.



Gambar 5.10: *Sequence diagram* Melakukan Visualisasi Data

Keterangan :

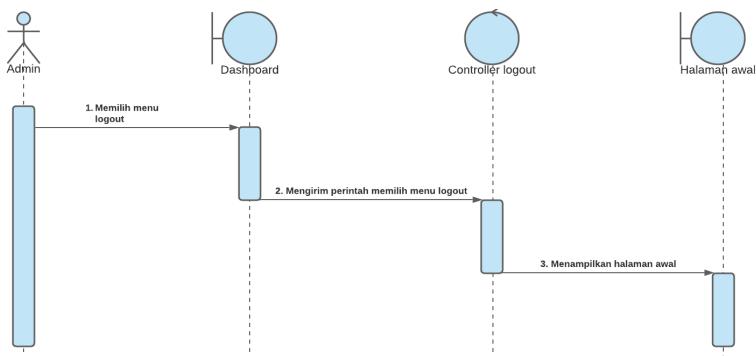
1. Admin memilih menu halaman visualisasi pada *dashboard*.
2. *Dashboard* mengirim perintah memilih menu visualisasi ke *controller visualisasi*.
3. *Controller visualisasi* menampilkan halaman visualisasi.

F. Sequence diagram Melakukan Logout

Berikut ini merupakan *Sequence diagram Logout Admin* menjelaskan hubungan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. *Boundary Class* : Dashboard dan Halaman Awal
3. *Control Class*: Controller Logout
4. Pesan Objek : Empat Pesan Objek
5. Pesan : Terdapat empat pesan yang dapat dipaparkan sebagai berikut.
 - a. Memilih menu *Logout*
 - b. Mengirim perintah memilih menu *Logout*
 - c. Menampilkan halaman awal

Berikut *Sequence diagram Logout Admin* pada gambar.



Gambar 5.11: 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.

5.2.4 *Collaboration diagram*

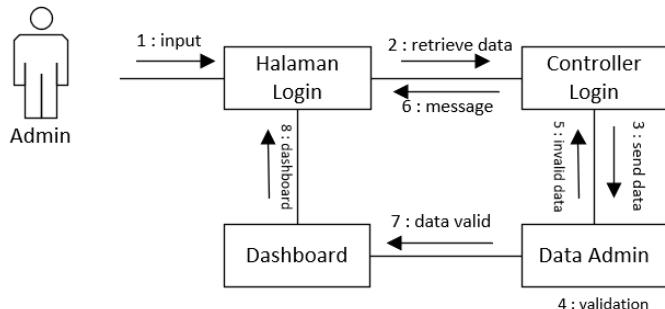
Collaboration diagram dapat dipakai untuk memodelkan interaksi antar objek yang ada di dalam sistem. Berbeda dari *Sequence diagram* yang lebih menunjukkan kronologis dari operasi-operasi yang dilakukan, *Collaboration diagram* ini lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh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

A. *Collaboration diagram* Melakukan *Login*

Berikut ini merupakan *Collaboration diagram* Melakukan *Login* Admin yang menjelaskan kronologis dari operasi-operasi yang dilakukan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. Pesan Objek : Empat Pesan Objek yang dapat dipaparkan sebagai berikut.
 - a. Halaman *Login*
 - b. *Controller Login*
 - c. Data Admin
 - d. *Dashboard*
3. Pesan/stimulus : Terdapat delapan pesan yang dapat dipaparkan sebagai berikut.
 - a. Input
 - b. *Retrieve data*
 - c. *Send data*
 - d. *Validation*
 - e. *Invalid data*
 - f. Data valid
 - g. *Dashboard*

Berikut adalah *collaboration diagram* melakukan *login* dalam bentuk gambar.



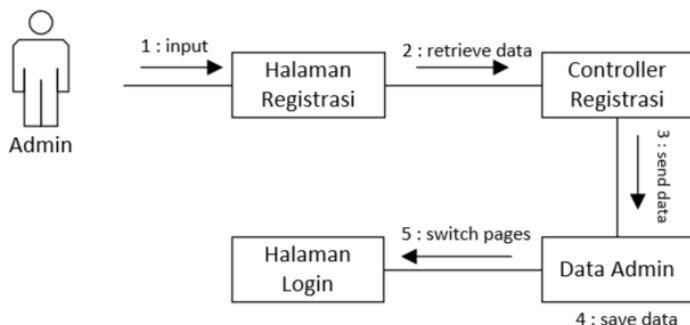
Gambar 5.12: Collaboration diagram Melakukan Login

B. Collaboration diagram Melakukan Registrasi

Berikut ini merupakan *Collaboration diagram* Melakukan Registrasi Admin yang menjelaskan kronologis dari operasi-operasi yang dilakukan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. Pesan Objek : Empat Pesan Objek yang dapat dipaparkan sebagai berikut.
 - a. Halaman Registrasi
 - b. Controller Registrasi
 - c. Data Admin
 - d. Halaman *Login*
3. Pesan/stimulus : Terdapat lima pesan yang dapat dipaparkan sebagai berikut.
 - a. Input
 - b. *Retrieve data*
 - c. *Send data*
 - d. *Save data*
 - e. *Switch page*

Berikut adalah *collaboration diagram* melakukan registrasi dalam bentuk gambar.



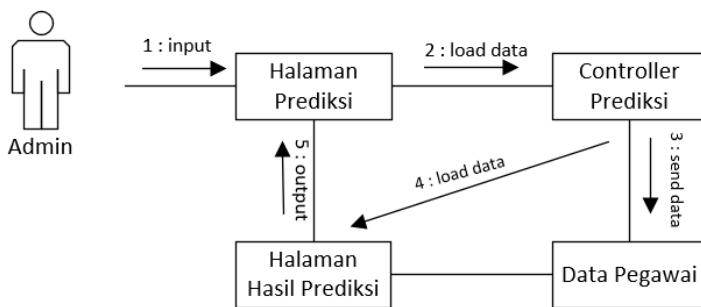
Gambar 5.13: *Collaboration diagram* Registrasi

C. *Collaboration diagram* Melakukan Prediksi Gaji

Berikut ini merupakan *Collaboration diagram* Melakukan Prediksi Gaji Pegawai yang menjelaskan kronologis dari operasi-operasi yang dilakukan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. Pesan Objek : Empat Pesan Objek yang dapat dipaparkan sebagai berikut.
 - a. Halaman Prediksi
 - b. *Controller* Prediksi
 - c. Data Pegawai
 - d. Halaman Hasil Prediksi
3. Pesan/stimulus : Terdapat lima pesan yang dapat dipaparkan sebagai berikut.
 - a. Input
 - b. *Load data*
 - c. *Send data*
 - d. *Load data*
 - e. *Output*

Berikut adalah *collaboration diagram* melakukan prediksi gaji pegawai dalam bentuk gambar.



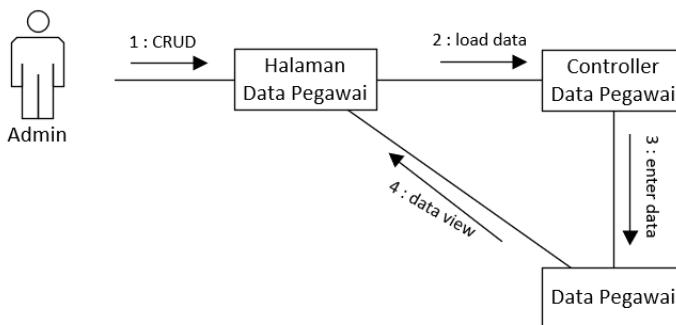
Gambar 5.14: Collaboration diagram Melakukan Prediksi Gaji

D. *Collaboration diagram* Melakukan Pengolahan Data

Berikut ini merupakan *Collaboration diagram* Melakukan Prediksi Gaji Pegawai yang menjelaskan kronologis dari operasi-operasi yang dilakukan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. Pesan Objek : Tiga Pesan Objek yang dapat dipaparkan sebagai berikut.
 - a. Halaman Data Pegawai
 - b. Controller Data Pegawai
 - c. Data Pegawai
3. Pesan/stimulus : Terdapat empat pesan yang dapat dipaparkan sebagai berikut.
 - a. CRUD
 - b. Load data
 - c. Enter data
 - d. Data View

Berikut adalah *collaboration diagram* melakukan pengolahan data pegawai dalam bentuk gambar.



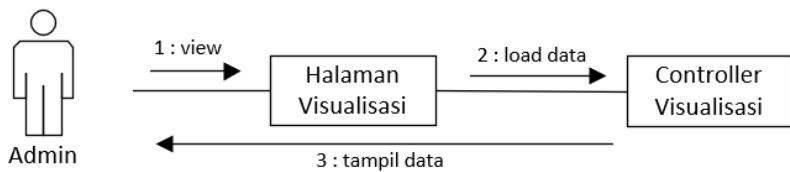
Gambar 5.15: Collaboration diagram Melakukan Pengolahan Data Pegawai

E. *Collaboration diagram* Melakukan Visualisasi Data

Berikut ini merupakan *Collaboration diagram* Melakukan Prediksi Gaji Pegawai yang menjelaskan kronologis dari operasi-operasi yang dilakukan antara admin pada aplikasi. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin
2. Pesan Objek : Dua Pesan Objek yang dapat dipaparkan sebagai berikut.
 - a. Halaman Visualisasi
 - b. Controller Visualisasi
3. Pesan/stimulus : Terdapat tiga pesan yang dapat dipaparkan sebagai berikut.
 - a. View
 - b. Load data
 - c. Tampil data

Berikut adalah *collaboration diagram* melakukan visualisasi data dalam bentuk gambar.



Gambar 5.16: Collaboration diagram Melakukan Visualisasi Data

5.2.5 Activity diagram

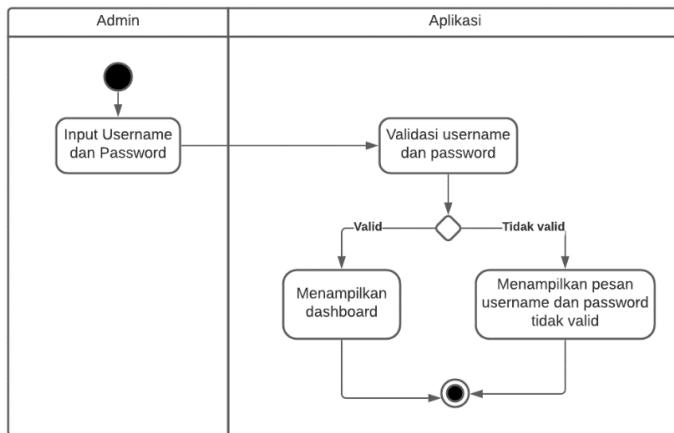
Activity diagram dapat didefinisikan sebagai diagram yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. *Activity diagram* adalah suatu teknik yang digunakan untuk mendeskripsikan logika procedural, proses bisnis serta aliran kerja dalam banyak kasus. *Activity diagram* menggambarkan 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

Berikut ini merupakan *Activity diagram Login Admin* yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat empat *activity* yang dapat dipaparkan sebagai berikut.
 - a. Input *username* dan *password*
 - b. Validasi *username* dan *password*
 - c. Menampilkan *dashboard*
 - d. Menampilkan pesan *username* dan *password* tidak valid
4. Satu percabangan untuk valid dan tidak valid
5. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan *login* admin dalam bentuk gambar.



Gambar 5.17: Activity diagram Login Admin

Keterangan :

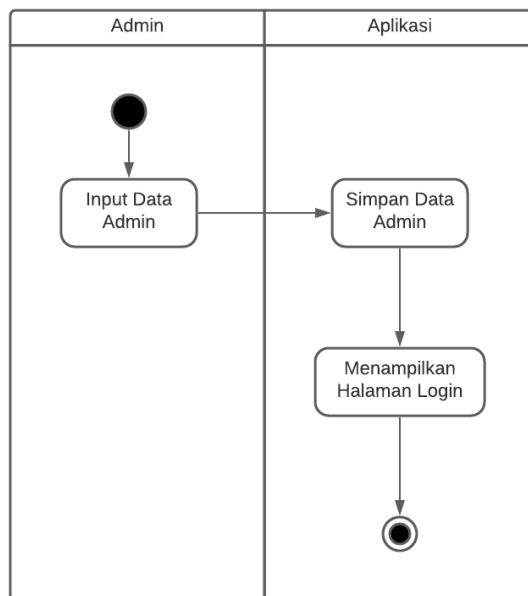
1. Admin menginput *username* dan *password*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid menampilkan pesan *username* dan *password* tidak valid.

B. Activity diagram Melakukan Registrasi Admin

Berikut ini merupakan *Activity diagram* Registrasi Admin yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu initial node
3. Activity : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - a. Input data admin
 - b. Simpan data admin
 - c. Menampilkan halaman *login*
4. Satu Activity Final Node

Berikut adalah *activity diagram* melakukan registrasi dalam bentuk gambar.



Gambar 5.18: *Activity diagram* Registrasi

Keterangan :

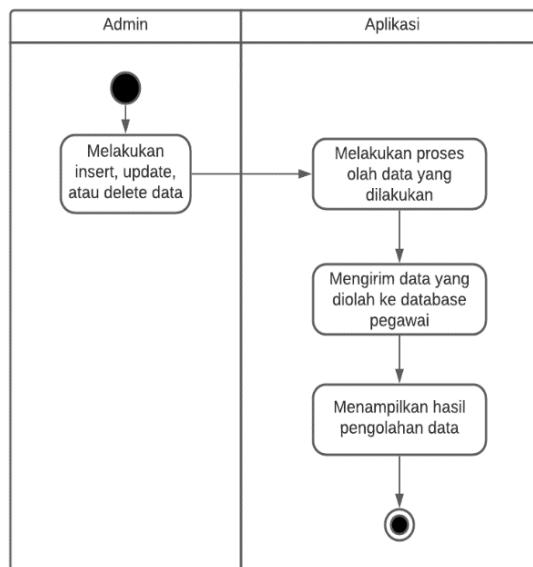
1. Admin menginput data admin.
2. Aplikasi menyimpan data admin.
3. Aplikasi menampilkan halaman *login*.

C. *Activity diagram* Melakukan Pengolahan Data

Berikut ini merupakan *Activity diagram* Pengolahan Data yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat empat *activity* yang dapat dipaparkan sebagai berikut.
 - a. Melakukan *insert*, *update*, atau *delete* data
 - b. Melakukan proses olah data yang dilakukan
 - c. Mengirim data yang diolah ke *database* pegawai
 - d. Menampilkan hasil pengolahan data
4. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan pengolahan data dalam bentuk gambar.



Gambar 5.19: *Activity diagram* Melakukan Pengolahan Data

Keterangan :

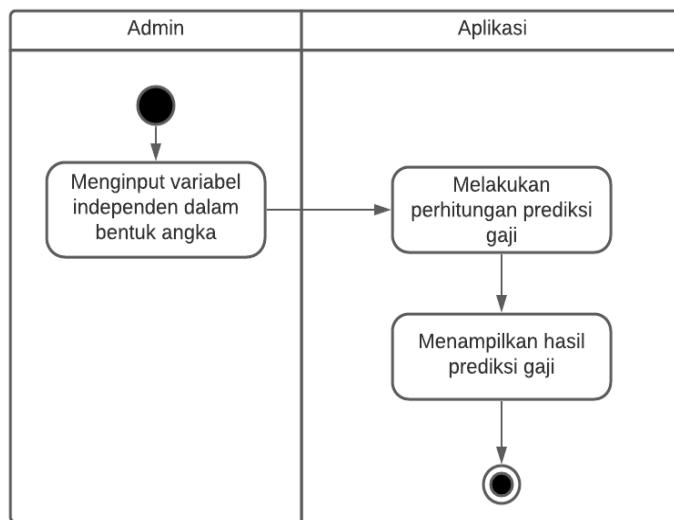
1. Admin melakukan olah data seperti *insert*, *update*, dan *delete* data.
2. Aplikasi melakukan proses olah data yang dilakukan.
3. Aplikasi mengirim data yang diolah ke *database* pegawai.
4. Aplikasi menampilkan data hasil *create/update/delete*.

D. *Activity diagram* Melakukan Prediksi Gaji

Berikut ini merupakan *Activity diagram* Melakukan Prediksi Gaji yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - Menginput variabel independen dalam bentuk angka
 - Melakukan perhitungan prediksi gaji
 - Menampilkan hasil prediksi gaji
4. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan prediksi gaji pegawai dalam bentuk gambar.



Gambar 5.20: *Activity diagram* Melakukan Prediksi Gaji

Keterangan :

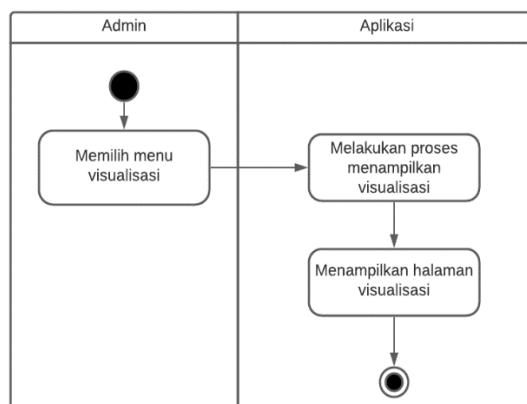
1. Admin menginputkan variabel independent dalam bentuk angka pada *form* prediksi.
2. Aplikasi melakukan proses perhitungan prediksi gaji.
3. Aplikasi menampilkan hasil prediksi.

E. *Activity diagram* Melakukan Visualisasi

Berikut ini merupakan *Activity diagram* Visualisasi Data yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - a. Memilih menu visualisasi
 - b. Melakukan proses menampilkan visualisasi
 - c. Menampilkan halaman visualisasi
4. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan pengolahan data dalam bentuk gambar.



Gambar 5.21: *Activity diagram* Melakukan Visualisasi

Keterangan :

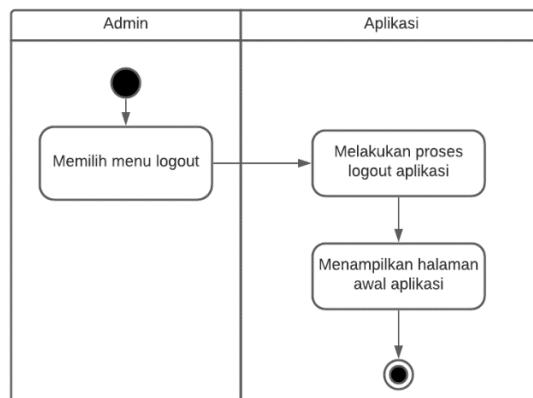
1. Admin memilih menu visualisasi.
2. Aplikasi melakukan proses menampilkan visualisasi.
3. Aplikasi menampilkan halaman visualisasi.

F. Activity diagram Logout Admin

Berikut ini merupakan *Activity diagram Logout Admin* yang menggambarkan tentang berbagai aktifitas yang terjadi pada suatu sistem. Dimana pada diagram ini, terdiri dari komponen-komponen sebagai berikut.

1. Aktor : Admin dan Aplikasi
2. Satu *initial node*
3. *Activity* : Terdapat tiga *activity* yang dapat dipaparkan sebagai berikut.
 - a. Memilih menu *Logout*
 - b. Melakukan proses *Logout* aplikasi
 - c. Menampilkan halaman awal aplikasi
4. Satu *Activity Final Node*

Berikut adalah *activity diagram* melakukan pengolahan data dalam bentuk gambar.



Gambar 5.22: *Activity diagram Logout Admin*

Keterangan :

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

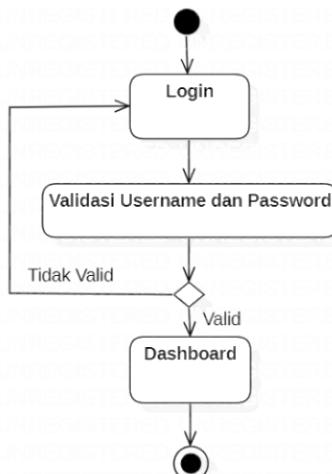
5.2.6 Statechart Diagram

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

Berikut ini merupakan *Activity diagram* *Login Admin* yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan *login* admin dalam bentuk gambar.



Gambar 5.23: Statechart Diagram Login Admin

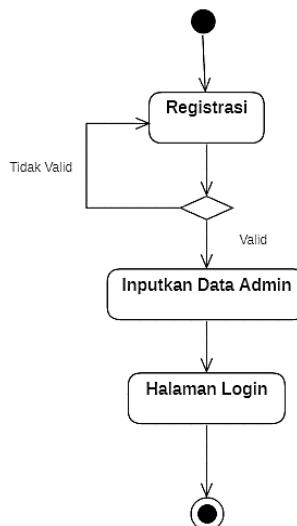
Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.

B. *Statechart Diagram* Registrasi Admin

Berikut ini merupakan *Activity diagram* Registrasi Admin yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan registrasi admin dalam bentuk gambar.



Gambar 5.24: *Statechart Diagram* Registrasi

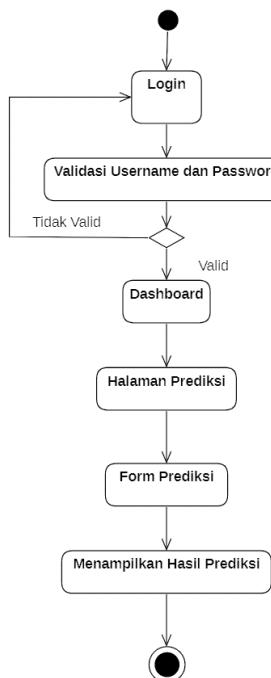
Keterangan :

1. Admin menginput data admin.
2. Jika valid maka data admin akan terinput ke *database*.
3. Jika tidak valid maka akan dikembalikan ke halaman registrasi admin.
4. Aplikasi menampilkan halaman *login*.

C. *Statechart Diagram* Melakukan Prediksi

Berikut ini merupakan *Activity diagram* Prediksi Gaji yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan prediksi gaji dalam bentuk gambar.



Gambar 5.25: *Statechart Diagram* Melakukan Prediksi

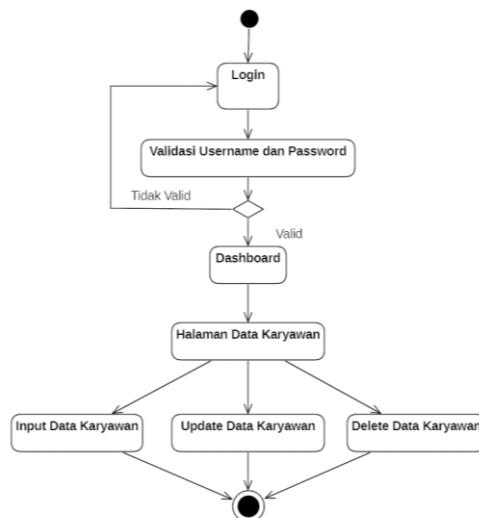
Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi menampilkan *form* prediksi pada halaman prediksi
6. Aplikasi menampilkan hasil prediksi gaji.

D. *Statechart Diagram* Pengolahan Data

Berikut ini merupakan *Activity diagram* Pengolahan Data Pegawai yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan pengolahan data dalam bentuk gambar.



Gambar 5.26: *Statechart Diagram* Pengolahan Data

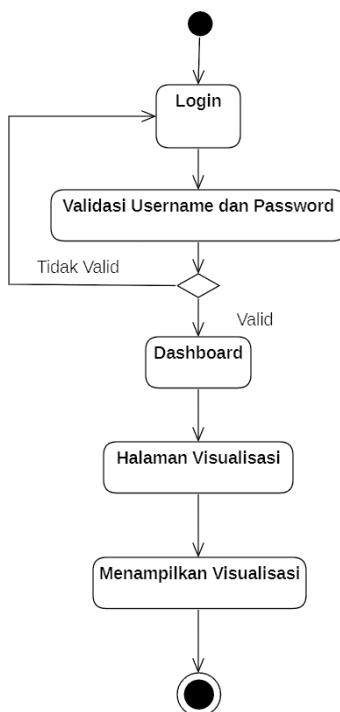
Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi akan menampilkan halaman data karyawan
6. Pada halaman data karyawan, admin dapat memilih pengolahan antara *input data karyawan*, *update data karyawan*, ataupun *delete data karyawan*

E. *Statechart Diagram* Visualisasi

Berikut ini merupakan *Activity diagram* Visualisasi Data yang menggambarkan transisi dan perubahan keadaan dari suatu objek pada sistem.

Berikut adalah *Statechart Diagram* melakukan visualisasi data dalam bentuk gambar.



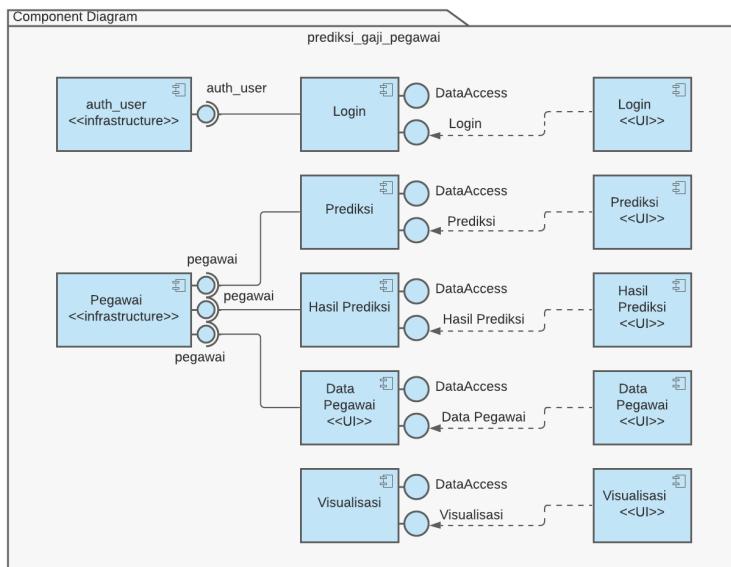
Gambar 5.27: *Statechart Diagram* Visualisasi

Keterangan :

1. Admin melakukan *login*.
2. Aplikasi melakukan validasi *username* dan *password*.
3. Jika valid akan menampilkan halaman *dashboard*.
4. Jika tidak valid akan dikembalikan ke halaman *login*.
5. Aplikasi menampilkan halaman *visualisasi*.
6. *Visualisasi* data ditampilkan oleh aplikasi

5.2.7 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.

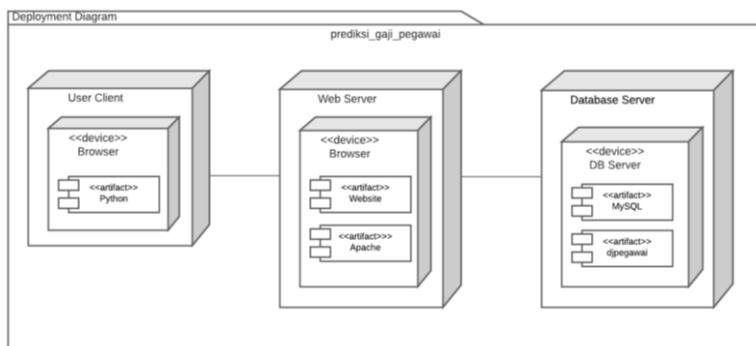


Gambar 5.28: Component diagram

5.2.8 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)

Berikut adalah *deployment diagram* aplikasi prediksi gaji pegawai dalam bentuk gambar.



Gambar 5.29: Deployment Diagram

Gambar *Deployment Diagram* diatas menjelaskan dimana aplikasi prediksi gaji pegawai diakses dengan arsitektur yang terdiri dari *User Client*, *Web Server*, dan *Database Server*. Arsitektur yang terbentuk pada diagram tersebut dapat dijelaskan sebagai berikut.

1. *User Client*

User Client atau pengguna aplikasi dapat menggunakan aplikasi prediksi gaji pada *web browser* yang tersedia pada *device*. Aplikasi ini dibuat dengan bahasa pemrograman Python.

2. *Web Server*

Web server adalah *software* (perangkat lunak) yang memberikan layanan berupa data yang dapat memenuhi permintaan klien. *Web server* terdapat pada *device* berupa *browser*, dimana

browser dapat mengakses *website* serta apache untuk melayani dan memfungsikan situs *web*.

3. *Database Server*

Database Server yang digunakan pada aplikasi adalah MySQL dengan *database* yang bernama *djpegawai*.

Hubungan arsitektur tersebut dapat dijelaskan bahwa pengguna dapat menggunakan aplikasi pada *web browser* yang dimediasi oleh *web server* berupa *website* dan apache. Untuk menjalankan aplikasi tersebut apache terhubung dengan *database server* yang telah diaktifkan.

5.3 Perancangan Basis Data/ Data Sistem

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

5.3.1 CDM (*Conceptual Data Model*)

CDM adalah suatu jenis model data yang menggambarkan hubungan antar tabel secara konseptual(Diaz, Sulistiowati and Lemantara, 2016). Sebuah *Conceptual Data Model* (CDM) menggambarkan secara keseluruhan konsep struktur basis data yang dirancang untuk suatu aplikasi(Santoso, S and Setyawan, 2013).

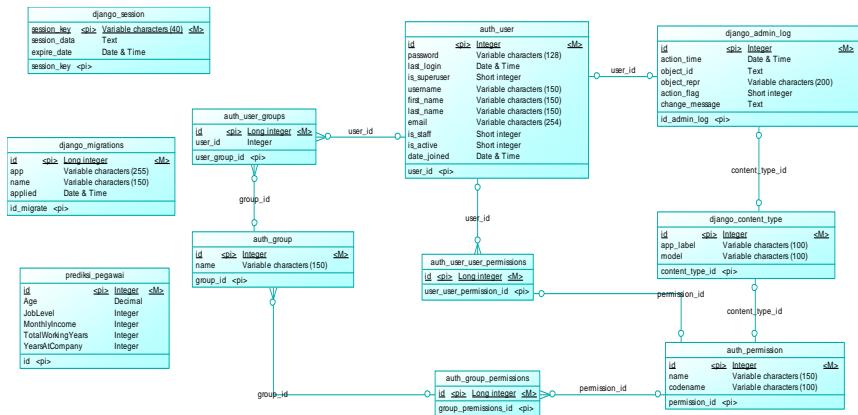
Pada CDM ini terdapat 11 (sebelas) entitas (tabel), entitas tersebut dapat dijelaskan sebagai berikut.

1. Entitas berelasi

- a. Auth_user_groups
- b. Auth_group
- c. Auth_user
- d. Auth_user_user_permissions
- e. Auth_group_permissions
- f. Django_admin_log
- g. Django_content_type
- h. Auth_permission

2. Entitas tidak berelasai
 - a. Django_session
 - b. Django_migrations
 - c. Prediksi_pegawai

Berikut adalah *Conceptual Data Model* aplikasi prediksi gaji pegawai dalam bentuk gambar.



Gambar 5.30: Conceptual Data Model

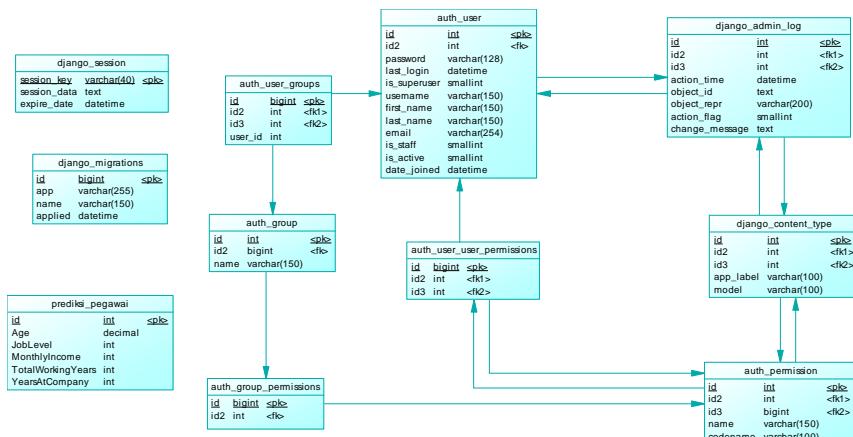
5.3.2 PDM (*Physical Data Model*)

Physical Data Model (PDM) adalah suatu jenis model data yang menggambarkan hubungan antar tabel secara fisikal(Diaz, Sulistiowati and Lemantara, 2016). Sebuah *Physical Data Model* (PDM) menggambarkan secara detail konsep rancangan struktur basis data yang dirancang untuk suatu aplikasi. PDM merupakan hasil generate dari CDM(Santoso, S and Setyawan, 2013).

Pada PDM ini terdapat 11 (sebelas) entitas (tabel), entitas tersebut dapat dijelaskan sebagai berikut.

1. Entitas berelasikan
 - a. Auth_user_groups
 - b. Auth_group
 - c. Auth_user
 - d. Auth_user_user_permissions
 - e. Auth_group_permissions
 - f. Django_admin_log
 - g. Django_content_type
 - h. Auth_permission
2. Entitas tidak berelasikan
 - a. Django_session
 - b. Django_migrations
 - c. Prediksi_pegawai

Berikut adalah *Physical Data Model* aplikasi prediksi gaji pegawai dalam bentuk gambar.



Gambar 5.31: Physical Data Model

5.3.3 ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model(Fridayanthie and Mahdiati, 2016). Didalam hubungan ini tersebut dinyatakan yang utama dari ERD adalah menunjukkan objek data (*Entity*) dan hubungan (*Relationship*), yang ada pada *Entity* berikutnya(Fridayanthie and Mahdiati, 2016).

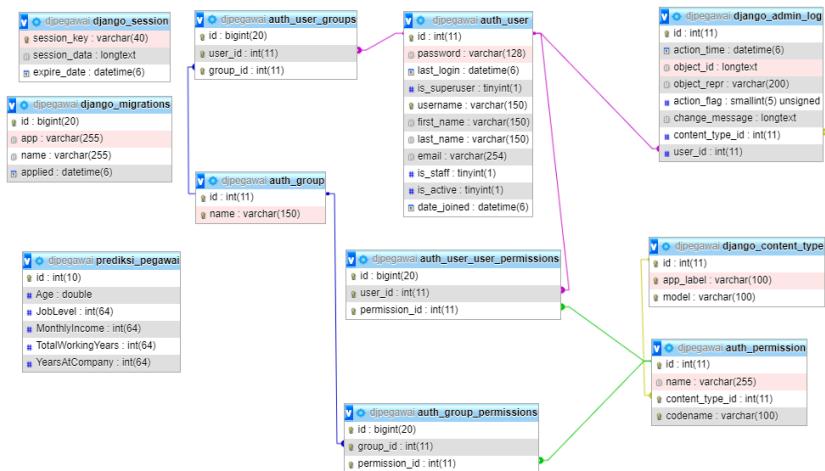
Simbol-simbol dalam ERD (*Entity Relationship Diagram*) adalah sebagai berikut(Fridayanthie and Mahdiati, 2016):

1. Entitas: suatu yang nyata atau abstrak yang mempunyai karakteristik dimana kita akan menyimpan data.
2. Atribut: ciri umum semua atau sebagian besar instansi pada entitas tertentu.
3. Relasi: hubungan alamiah yang terjadi antara satu atau lebih entitas.
4. Link: garis penghubung atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi.

Pada PDM ini terdapat 11 (sebelas) entitas (tabel), entitas tersebut dapat dijelaskan sebagai berikut.

1. Entitas berelasikan
 - a. Auth_user_groups
 - b. Auth_group
 - c. Auth_user
 - d. Auth_user_user_permissions
 - e. Auth_group_permissions
 - f. Django_admin_log
 - g. Django_content_type
 - h. Auth_permission
2. Entitas tidak berelasikan
 - a. Django_session
 - b. Django_migrations
 - c. Prediksi_pegawai

Berikut adalah *Entity Relationship Diagram* aplikasi prediksi gaji pegawai dalam bentuk gambar.



Gambar 5.32: Entity Relationship Diagram

5.3.4 Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi(Widianti, 2012). Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap(Widianti, 2012). Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem(Widianti, 2012).

Berikut adalah rancangan kamus data.

1. Nama Basis Data : djepgawai
Nama Tabel : django_session
Primary key : session_key
Foreign key : -

Tabel 5.10: Tabel django_session

No	Nama Field	Type	Length	Keterangan
1	Session_key	Varchar	40	Session key
2	Session_data	Text	-	Session data
3	Expire_date	datetime	6	Expire date

2. Nama Basis Data : djpegawai
Nama Tabel : django_migrations
Primary key : id
Foreign key : -

Tabel 5.11: Tabel django_migration

No	Nama Field	Type	Length	Keterangan
1	Id	Bigint	20	Id
2	App	Varchar	255	App
3	Name	Varchar	255	Name
4	Applied	Datetime	6	Applied

3. Nama Basis Data : djpegawai
Nama Tabel : prediksi_pegawai
Primary key : id
Foreign key : -

Tabel 5.12: Tabel prediksi_pegawai

No	Nama Field	Type	Length	Keterangan
1	Id	Int	10	Id
2	Age	Double	-	Age
3	JobLevel	Int	64	Job Level
4	MonthlyIncome	Int	64	Monthly Income
5	TotalWorkingYears	Int	64	Total Working Years
6	YearsAtCompany	Int	64	Years At Company

4. Nama Basis Data : djpegawai
Nama Tabel : auth_user_groups
Primary key : id
Foreign key : user_id, group_id

Tabel 5.13: Tabel auth_user_groups

No	Nama Field	Type	Length	Keterangan
1	Id	Bigint	20	Id
2	User_id	Int	11	User Id
3	Group_id	Int	11	Group Id

5. Nama Basis Data : djpegawai

Nama Tabel : auth_group

Primary key : id

Foreign key : -

Tabel 5.14: Tabel auth_group

No	Nama Field	Type	Length	Keterangan
1	Id	Int	11	Id
2	Name	Varchar	150	Name

6. Nama Basis Data : djpegawai

Nama Tabel : auth_user

Primary key : id

Foreign key : -

Tabel 5.15: Tabel auth_user

No	Nama Field	Type	Length	Keterangan
1	Id	Int	11	Id
2	<i>Password</i>	Varchar	128	<i>Password</i>
3	<i>Last_login</i>	Datetime	6	<i>Last Login</i>
4	Is_superuser	Tinyint	1	Is Super User
5	<i>Username</i>	Varchar	150	<i>Username</i>
6	First_name	Varchar	150	First Name
7	Last_name	Varchar	150	Last Name

8	Email	Varchar	254	Email
9	Is_staff	Tinyint	1	Is Staff
10	Is_active	Tinyint	1	Is Active
11	Date_joined	Datetime	6	Date Joined

7. Nama Basis Data : djpegawai

Nama Tabel : auth_user_user_permissions

Primary key : id

Foreign key : user_id, permission_id

Tabel 5.16: Tabel auth_user_user_permissions

No	Nama Field	Type	Length	Keterangan
1	Id	Bigint	20	Id
2	User_id	Int	11	User Id
3	Permission_id	Int	11	Permission Id

8. Nama Basis Data : djpegawai

Nama Tabel : auth_group_permissions

Primary key : id

Foreign key : group_id, permission_id

Tabel 5.17: Tabel auth_group_permissions

No	Nama Field	Type	Length	Keterangan
1	Id	Bigint	20	Id

2	Group_id	Int	11	Group Id
3	Permission_id	Int	11	Permission Id

9. Nama Basis Data : djpegawai
 Nama Tabel : django_admin_log
Primary key : id
Foreign key : user_id, content_type_id

Tabel 5.18: Tabel django_admin_log

No	Nama Field	Type	Length	Keterangan
1	Id	Int	11	Id
2	Action_time	Datetime	6	Action Time
3	Object_id	Longtext	-	Object Id
4	Object_repr	Varchar	200	Object Repr
5	Action_flag	Smallint	5	Action Flag (Unsigned)
6	Change_message	Longtext	-	Change Message
7	Content_type_id	Int	11	Content Type Id
8	User_id	Int	11	User Id

10. Nama Basis Data : djpegawai
 Nama Tabel : django_content_type
Primary key : id
Foreign key : -

Tabel 5.19: Tabel django_content_type

No	Nama Field	Type	Length	Keterangan
1	Id	Int	11	Id
2	App_label	Varchar	100	App Label
3	Model	Varchar	100	Model

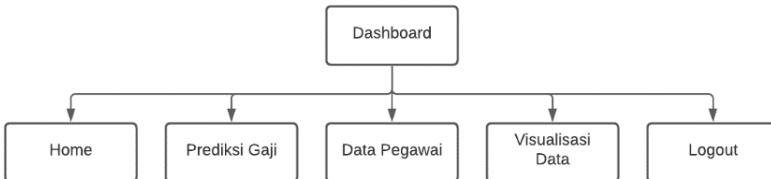
11. Nama Basis Data : djpegawai
 Nama Tabel : auth_permission
Primary key : id
Foreign key : content_type_id

Tabel 5.20: Tabel auth_permission

No	Nama Field	Type	Length	Keterangan
1	Id	Int	11	Id
2	Name	Varchar	255	Name
3	Content_type_id	Int	11	Content Type Id
4	Codename	Varchar	100	Codename

5.4 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 menumenu yang diinginkan. Berikut adalah struktur menu dari aplikasi Prediksi Gaji Pegawai secara keseluruhan.



Gambar 5.33: Struktur Menu Aplikasi

Berdasarkan gambar diatas, terdapat dijelaskan aplikasi memiliki halaman *dashboard* yang dapat mengkases menu lainnya yaitu sebagai berikut.

1. Halaman Home
Home adalah halaman awal pada aplikasi.
2. Halaman Prediksi Gaji
Halaman Prediksi Gaji adalah halaman untuk melakukan prediksi gaji pegawai.
3. Halaman Data Pegawai
Halaman Data Pegawai adalah halaman yang menampilkan data kepegawaian.
4. Halaman Visualisasi Data
Halaman Visualisasi Data adalah halaman yang menampilkan visualisasi data.
5. *Logout*.
Logout adalah menu yang digunakan oleh *user* untuk keluar aplikasi.

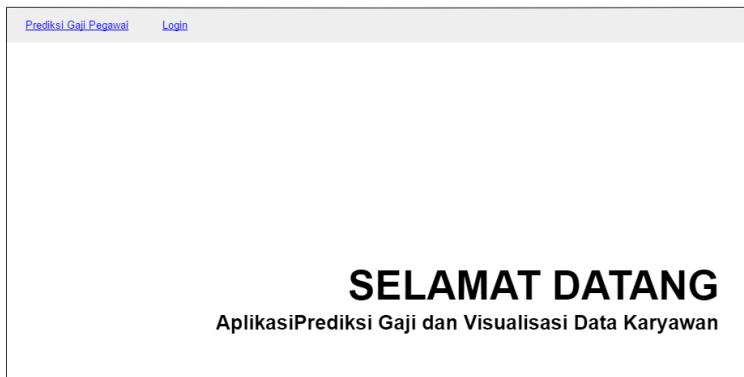
5.5 Perancangan *User Interface*

Perancangan *User Interface* adalah salah satu langkah yang penting ketika kita ingin membuat sebuah aplikasi. Dengan adanya perancangan *User Interface*, pembuatan aplikasi akan lebih terarah. Pada perancangan *User Interface* Aplikasi Prediksi Gaji, dirancang dengan *software UI* yang bernama pencil.

Berikut adalah perancangan *User Interface* aplikasi :

5.5.1 *User Interface* Halaman Depan

Berikut adalah rancangan *user interface* halaman depan aplikasi prediksi gaji pegawai.



Gambar 5.34: Rancangan *User Interface* Dari Halaman Depan

Gambar diatas adalah rancangan *user interface* dari halaman depan aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai” dan “Login”
2. Menu “Prediksi Gaji Pegawai” berfungsi untuk men-direct ke halaman depan aplikasi prediksi gaji.
3. Menu “Login” berfungsi untuk men-direct ke halaman login aplikasi prediksi gaji.
4. Terdapat teks “SELAMAT DATANG” sebagai pembuka atau penyambut pengguna. Selain itu terdapat teks “Aplikasi Prediksi Gaji dan Visualisasi Data Karyawan” sebagai nama dari aplikasi yang dirancang.

5.5.2 *User Interface Halaman Login*

Berikut adalah rancangan *user interface* halaman *login* aplikasi prediksi gaji pegawai.



Rancangan *user interface* halaman *login* aplikasi prediksi gaji pegawai. Tampilan halaman ini menampilkan:

- Teks selamat datang: "SELAMAT DATANG !".
- Keterangan: "Silahkan Inputkan Username dan Password".
- Form input:
 - Username: input box.
 - Password: input box.
 - Tombol "Login": tombol yang berfungsi untuk melakukan login.
- Pesan: "Belum memiliki akun? Silahkan daftar." diikuti dengan [Mendaftar](#).
- Tombol "Kembali": tombol yang berfungsi untuk kembali ke halaman sebelumnya.

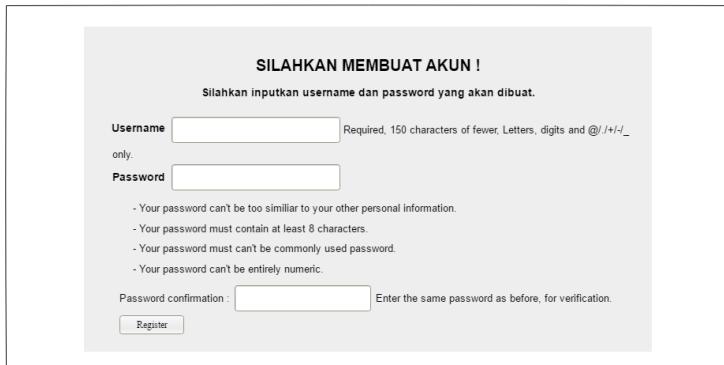
Gambar 5.35: Rancangan *User Interface* Dari Halaman *Login*

Gambar diatas adalah rancangan *user interface* dari halaman *login* aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat teks “SELAMAT DATANG !” sebagai penyambut. Kemudian terdapat teks “Silahkan Inputkan *Username* dan *Password*” yang berfungsi sebagai pemberi keterangan untuk interaksi yang harus dilakukan oleh pengguna.
2. Terdapat *form* yang berisi inputan untuk *Username* dan *Password* untuk melakukan *login* oleh pengguna.
3. Terdapat tombol “*Login*” yang berfungsi untuk *login* aplikasi setelah pengguna menginputkan *Username* dan *Password*.
4. Terdapat teks “Belum memiliki akun? Silahkan daftar.” Untuk memberi keterangan apabila pengguna belum memiliki akun. Kemudian dibawahnya terdapat *hyperlink* “*Mendaftar*” untuk men-direct ke halaman registrasi.
5. Pada bagian kanan bawah terdapat tombol “*Kembali*” untuk kembali ke halaman sebelumnya.

5.5.3 User Interface Halaman Registrasi

Berikut adalah rancangan *user interface* halaman registrasi aplikasi prediksi gaji pegawai.



The image shows a wireframe of a registration form titled "SILAHKAN MEMBUAT AKUN !". Below the title, there is a note: "Silahkan inputkan username dan password yang akan dibuat." There are two input fields: "Username" and "Password". To the right of the "Username" field, there is a note: "Required, 150 characters of fewer, Letters, digits and @/ /+/- only." Below the "Password" field, there is a note: "- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password must can't be commonly used password.
- Your password can't be entirely numeric." There is also a note below the "Password confirmation" field: "Enter the same password as before, for verification." At the bottom left is a "Register" button.

Gambar 5.36: Rancangan *User Interface* Dari Halaman Registrasi

Gambar diatas adalah rancangan *user interface* dari halaman registrasi aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat teks “SILAHKAN MEMBUAT AKUN !” sebagai keterangan halaman untuk membuat akun. Selain itu terdapat teks “Silahkan inputkan *username* dan *password* yang akan dibuat” yang berfungsi sebagai pemberi keterangan untuk interaksi yang harus dilakukan oleh pengguna.
2. Terdapat *form* yang berisi inputan untuk *Username* dan *Password* serta *Password confirmation* untuk membuat akun oleh pengguna.
3. Terdapat teks “Required, 150 characters of fewer, Letters, digits and @/ /+/- only” yang terdapat di sebelah kanan *form* input *username*, yang berfungsi sebagai pemberi keterangan bahwa *username* wajib terdiri dari kurang dari 150 karakter, huruf, angka, dan symbol.
4. Terdapat teks berbaris dibawah *form* input *password* seperti berikut.
“- Your *password* can be too similar to your other personal information

- Your *password* must contain at least 8 character
- Your *password* must can't be commonly used *password*
- Your *password* can't be entirely numeric”

Teks tersebut berfungsi untuk pemberi keterangan dalam membuat *password*, yaitu:

- Kata sandi mungkin terlalu mirip dengan informasi pribadi Anda yang lain
 - Kata sandi harus mengandung setidaknya 8 karakter
 - Kata sandi harus bukan kata sandi yang biasa digunakan
 - Kata sandi tidak boleh seluruhnya numerik
5. Terdapat teks “Enter the same *password* as before, for verification” di sebelah kanan *form* input confirmation *password*. Teks tersebut berfungsi sebagai keterangan untuk menginputkan *password* yang sebelumnya telah dibuat untuk melakukan konfirmasi *password*.
 6. Terdapat tombol “Register” yang berfungsi untuk melakukan proses registrasi ketika pengguna telah menginputkan seluruh *form* yang ada.

5.5.4 *User Interface Halaman Dashboard*

Berikut adalah rancangan *user interface* halaman *dashboard* aplikasi prediksi gaji pegawai.



Gambar 5.37: Rancangan *User Interface* Dari Halaman *Dashboard*

Gambar diatas adalah rancangan *user interface* dari halaman *dashboard* aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Menu “Prediksi Gaji Pegawai” berfungsi untuk men-direct ke halaman depan aplikasi prediksi gaji.
3. Menu “Prediksi” berfungsi untuk men-direct ke halaman prediksi gaji aplikasi prediksi gaji.
4. Menu “Data Pegawai” berfungsi untuk men-direct ke halaman daftar data pegawai aplikasi prediksi gaji.
5. Menu “Visualisasi Data” berfungsi untuk men-direct ke halaman hasil visualisasi data aplikasi prediksi gaji.
6. Menu “Logout” berfungsi *Logout* atau keluar dari aplikasi ketika pekerjaan telah selesai dilakukan.
7. Terdapat teks “SELAMAT DATANG” sebagai pembuka atau penyambut pengguna. Selain itu terdapat teks “Aplikasi Prediksi

Gaji dan Visualisasi Data Karyawan” sebagai nama dari aplikasi yang dirancang.

5.5.5 *User Interface* Halaman Prediksi

Berikut adalah rancangan *user interface* halaman prediksi gaji aplikasi prediksi gaji pegawai.



Rancangan *User Interface* halaman prediksi gaji pegawai. Di bagian atas terdapat navbar dengan menu: Prediksi Gaji Pegawai, Prediksi, Data Pegawai, Visualisasi Data, dan Logout. Di bagian tengah terdapat judul "PREDIKSI GAJI PEGAWAI" dan petunjuk "Silaikan inputkan tahun dalam bentuk angka". Berikutnya ada empat input text berturut-turut untuk "Inputkan Usia", "Inputkan Job Level", "Inputkan Total Tahun Bekerja", dan "Inputkan Total Tahun Bekerja di Perusahaan". Di bawahnya terdapat tombol "Lakukan Prediksi" dan di sebelahnya tombol "Kembali".

Gambar 5.38: Rancangan *User Interface* Dari Halaman Prediksi

Gambar diatas adalah rancangan *user interface* dari halaman prediksi gaji aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Halaman ini memiliki *form* untuk melakukan proses prediksi gaji pegawai.
3. Terdapat teks “PREDIKSI GAJI PEGAWAI” sebagai keterangan halaman untuk melakukan prediksi gaji pegawai.
4. Terdapat *form* input yang terdiri dari inputan Usia, Job Level, Total Tahun Bekerja, dan Total Tahun Bekerja Di Perusahaan.
5. Terdapat tombol “Lakukan Prediksi” untuk melakukan proses prediksi gaji ketika *form* telah diinputkan seluruhnya.
6. Pada bagian kanan bawah terdapat tombol “Kembali” untuk kembali ke halaman sebelumnya.

5.5.6 *User Interface* Halaman Hasil Prediksi

Berikut adalah rancangan *user interface* halaman depan aplikasi prediksi gaji pegawai.



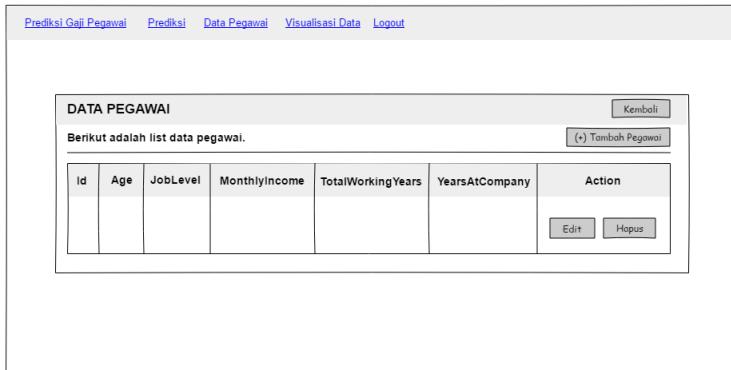
Gambar 5.39: Rancangan *User Interface* Dari Halaman Hasil Prediksi

Gambar diatas adalah rancangan *user interface* dari halaman hasil prediksi gaji pegawai dari aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Halaman ini memiliki hasil dari proses prediksi yang sebelumnya telah dilakukan.
3. Terdapat teks “HASIL PREDIKSI GAJI PEGAWAI” sebagai keterangan halaman hasil prediksi gaji pegawai. Selain itu terdapat teks “*dalam bentuk dollar (\$)” sebagai pemberi keterangan bahwa gaji miliki satuan dollar.
4. Pada bagian kanan bawah terdapat tombol “Kembali” untuk kembali ke halaman sebelumnya.

5.5.7 User Interface Halaman Data Pegawai

Berikut adalah rancangan *user interface* halaman data pegawai aplikasi prediksi gaji pegawai.



Gambar 5.40: Rancangan *User Interface* Dari Halaman Data Pegawai

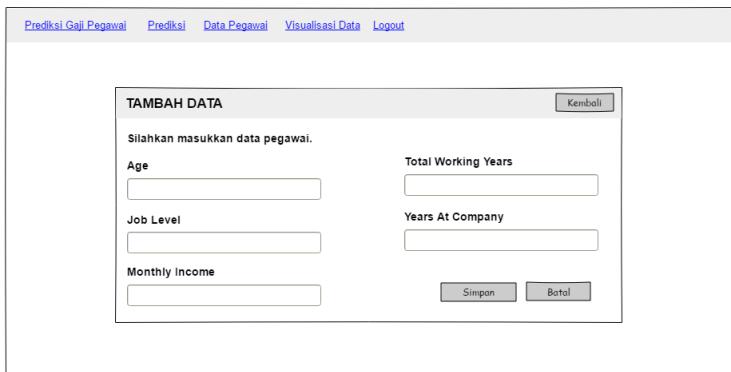
Gambar diatas adalah rancangan *user interface* dari halaman data pegawai dari aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Terdapat teks “DATA PEGAWAI” sebagai keterangan halaman data pegawai. Selain itu terdapat teks “Berikut adalah list data pegawai” sebagai pemberi keterangan bahwa tabel dibawah adalah list data pegawai.
3. Pada bagian kanan atas terdapat tombol “Kembali” untuk kembali ke halaman sebelumnya.
4. Pada bagian kanan atas yang terdapat dibawah tombol “Kembali” terdapat tombol “[+] Tambah Pegawai” yang berfungsi untuk menambah *record* data pegawai.
5. Terdapat tabel untuk list data pegawai yang terdiri dari atribut Id, Age, JobLevel, MonthlyIncome, TotalWorkingYears, YearsAtCompany, dan Action.
6. Pada bagian atribut Action, terdapat dua tombol yaitu tombol Edit dan tombol Hapus. Tombol Edit berfungsi untuk merubah

record yang dipilih, sedangkan tombol Hapus berfungsi untuk menghapus *record* pegawai yang dipilih.

5.5.8 User Interface Halaman Tambah Data Pegawai

Berikut adalah rancangan *user interface* halaman tambah data pegawai aplikasi prediksi gaji pegawai.



Rancangan *User Interface* halaman tambah data pegawai. Di bagian atas terdapat navbar dengan menu: Prediksi Gaji Pegawai, Prediksi, Data Pegawai, Visualisasi Data, dan Logout. Di bagian tengah terdapat form bertitik “TAMBAH DATA” yang meminta pengguna masukkan data pegawai. Form ini memiliki tiga baris input: Age, Job Level, dan Monthly Income di sisi kiri, dan Total Working Years dan Years At Company di sisi kanan. Di bagian bawah form terdapat dua tombol: Simpan dan Batal.

Gambar 5.41: Rancangan *User Interface* Dari Halaman Tambah Data Pegawai

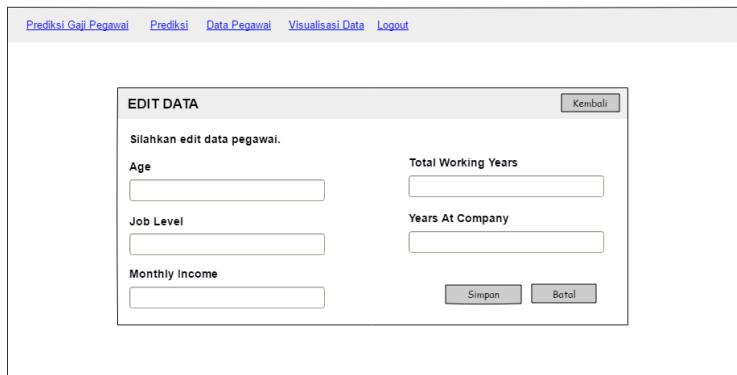
Gambar diatas adalah rancangan *user interface* dari halaman tambah data pegawai dari aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Terdapat teks “TAMBAH DATA” sebagai keterangan halaman tambah data pegawai. Selain itu terdapat teks “Silahkan masukan data pegawai.” yang berfungsi sebagai pemberi keterangan untuk interaksi yang harus dilakukan oleh pengguna.
3. Halaman ini memiliki *form* untuk menambah data pegawai.
4. Terdapat *form* input yang terdiri dari inputan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company.
5. Terdapat tombol “Simpan” untuk melakukan proses penambahan data pegawai ketika *form* telah diinputkan seluruhnya.

6. Terdapat tombol “Batal” untuk mereset seluruh data yang diinputkan.
7. Pada bagian kanan atas terdapat tombol “Kembali” untuk kembali ke halaman sebelumnya.

5.5.9 *User Interface* Halaman Edit Data Pegawai

Berikut adalah rancangan *user interface* halaman edit data pegawai aplikasi prediksi gaji pegawai.



Rancangan *User Interface* halaman edit data pegawai. Di bagian atas terdapat navbar dengan menu: Prediksi Gaji Pegawai, Prediksi, Data Pegawai, Visualisasi Data, dan Logout. Di bagian tengah terdapat form edit data pegawai dengan judul "EDIT DATA". Form ini meminta pengguna untuk mengisi data berikut: Age (usia), Total Working Years (jumlah tahun kerja), Job Level (tingkat pekerjaan), Years At Company (jumlah tahun di perusahaan), dan Monthly Income (pendapatan bulanan). Setiap inputan dilengkapi dengan tombol Simpan (simpan) dan Batal (batal).

Gambar 5.42: Rancangan *User Interface* Dari Halaman Edit Data Pegawai

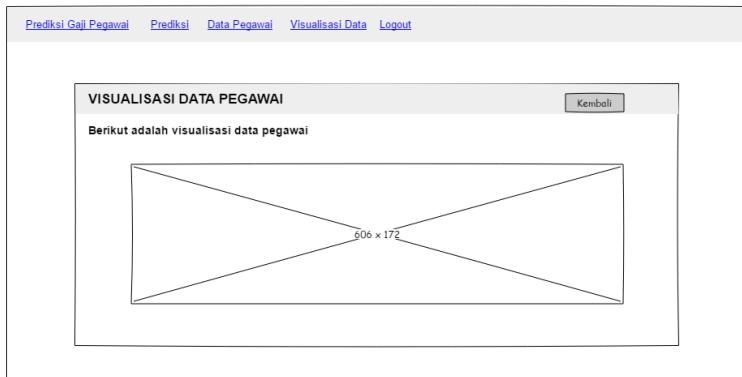
Gambar diatas adalah rancangan *user interface* dari halaman merubah atau edit data pegawai dari aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Terdapat teks “EDIT DATA” sebagai keterangan halaman edit data pegawai. Selain itu terdapat teks “Silahkan edit data pegawai.” yang berfungsi sebagai pemberi keterangan untuk interaksi yang harus dilakukan oleh pengguna.
3. Halaman ini memiliki *form* untuk mengedit data pegawai.
4. Terdapat *form* input yang terdiri dari inputan Age, Job Level, Monthly Income, Total Working Years, dan Years At Company.

5. Terdapat tombol “Simpan” untuk melakukan proses penyimpanan data pegawai yang diubah ketika *form* telah diinputkan seluruhnya.
6. Terdapat tombol “Batal” untuk mereset seluruh data yang diinputkan.
7. Pada bagian kanan atas terdapat tombol “Kembali” untuk kembali ke halaman sebelumnya.

5.5.10 User Interface Halaman Visualisasi

Berikut adalah rancangan *user interface* halaman visualisasi aplikasi prediksi gaji pegawai.



Gambar 5.43: Rancangan *User Interface* Dari Halaman Visualisasi

Gambar diatas adalah rancangan *user interface* dari halaman visualisasi data pegawai dari aplikasi prediksi gaji. Rancangan tersebut dapat dijelaskan sebagai berikut:

1. Terdapat navbar yang terdiri dari menu “Prediksi Gaji Pegawai”, “Prediksi”, “Data Pegawai”, “Visualisasi Data”, dan “Logout”
2. Terdapat teks “VISUALISASI DATA PEGAWAI” sebagai keterangan halaman data pegawai. Selain itu terdapat teks “Berikut adalah visualisasi data pegawai.” yang berfungsi sebagai pemberi keterangan bahwa visualisasi yang dipaparkan adalah visualisasi data pegawai.

5.6 Implementasi Kode 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.

5.6.1 Implementasi Package Yang Digunakan.

```
[1] from django.shortcuts import render, redirect
[2] import pandas as pd
[3] import pickle
[4] from prediksi.forms import PegawaiForm
[5] from prediksi.forms import Pegawai
[6] UserCreationForm
[7] from django.contrib.auth.decorators
[8] import login_required
```

Kode diatas adalah kode dari package yang digunakan dalam perancangan aplikasi. Package yang digunakan merupakan sesuai kebutuhan dalam perancangan aplikasi. Kode tersebut berada dibagian atas pada komponen arsitektur *views.py*.

5.6.2 Implementasi Proses Login

A. Views

```
[1] @login_required
[2] def dashboardView(request):
    return render(request, 'dashboard.html')
```

Dekorator *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*

```
[1] <h1 class="h4 text-light mb-4">
    <b>SELAMAT DATANG !</b>
</h1>
[2] <div class="sidebar-brand-text mx-3 text-light">
    Silahkan inputkan username dan password</div>
    <br/>
</div>
[3] <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>
[4] <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>
```

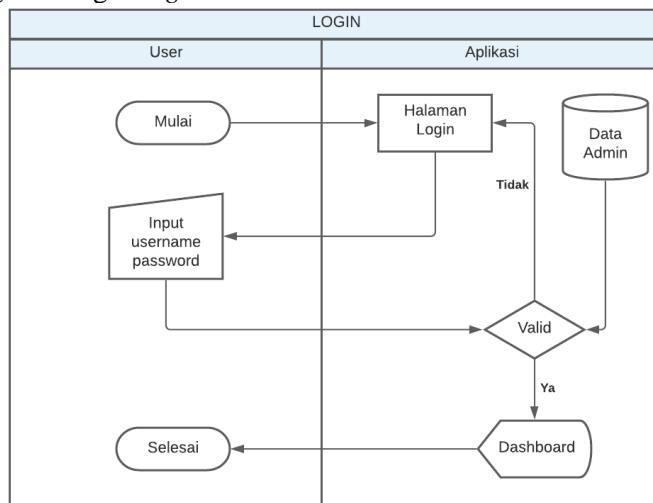
Kode diatas merupakan tampilan dari halaman *login*. Terdapat *form* yang berisikan inputan *username* dan *password*. Selain itu terdapat link “Mendaftar!” untuk melakukan registrasi.

C. URL

```
[1] path('login/', LoginView.as_view(), name="login_url"),
```

Kode diatas merupakan URL untuk mengakses halaman *login*.

D. Logika Fungsi *Login*



Gambar 5.44: 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

5.6.3 Implementasi Proses Registrasi

A. Views

```
[1] def registerView(request):
    if request.method == "POST":
        form = UserCreationForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('login_url')
        else:
            form = UserCreationForm()
[2]    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

```
[1] <b>SILAHKAN MEMBUAT AKUN !</b>
</h1>
<div class="sidebar-brand-text mx-3 text-light">
Silahkan inputkan username dan password yang akan
dibuat.</div>
[2] <br/>
</div>
<form method="POST" class="text-light">
{% csrf_token %} {{form.as_p}}
<button>Register</button>
</form>
```

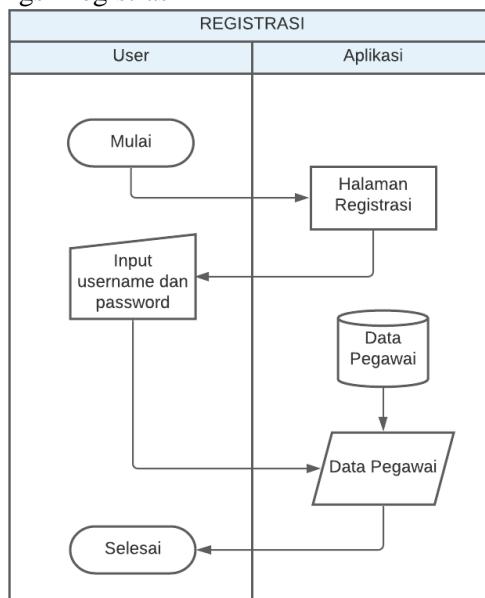
Kode diatas merupakan tampilan dari halaman register. Terdapat *form* yang berisikan inputan *username* dan *password* yang akan dibuat. Metode yang digunakan adalah POST.

C. URL

```
[1] path('register/',views.registerView,name="register_url")
```

Kode diatas merupakan URL untuk mengakses halaman registrasi.

D. Logika Fungsi Registrasi



Gambar 5.45: 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

5.6.4 Implementasi Proses Prediksi Gaji

A. Views

```
[1] 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

```
[1] <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>
[2] {% 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>
[3] <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 TahunBekerja 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>
    [4]    </div>{% endblock %}
</form>
{{ result }}
```

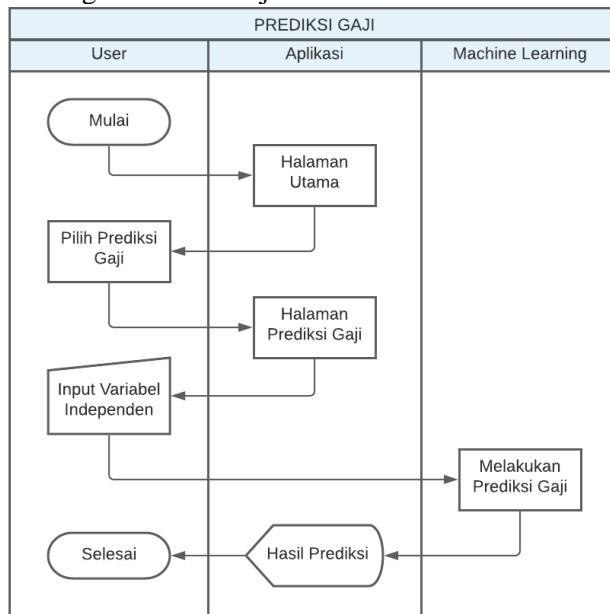
Kode diatas merupakan tampilan dari halaman prediksi gaji. Terdapat *form* yang berisikan inputan usia, job level, total tahun bekerja, dan total tahun bekerja di perusahaan. Selain itu terdapat button “Lakukan Prediksi” untuk melakukan proses prediksi.

C. URL

```
[1] url('predict', views.predict, name='predict')
```

Kode diatas merupakan URL untuk mengakses halaman prediksi gaji.

D. Logika Fungsi Prediksi Gaji



Gambar 5.46: 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.

5.6.5 Implementasi Proses Hasil Prediksi Gaji

A. Views

```
[1] def result(request):  
    print(request)
```

```
tahun = float(request.POST.get('tahun'))  
  
model= pd.read_pickle('./models/model3.pickle')  
result = model.predict([[tahun]])  
[2] 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*

```
[1] <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>  
[2] <div class="card text-center">  
<div class="card-body">  
<h4>{{ result }}</h4>  
</div>  
</div>  
<br>  
[3] <p class="text-light">*dalam bentuk dollar  
($)</p>
```

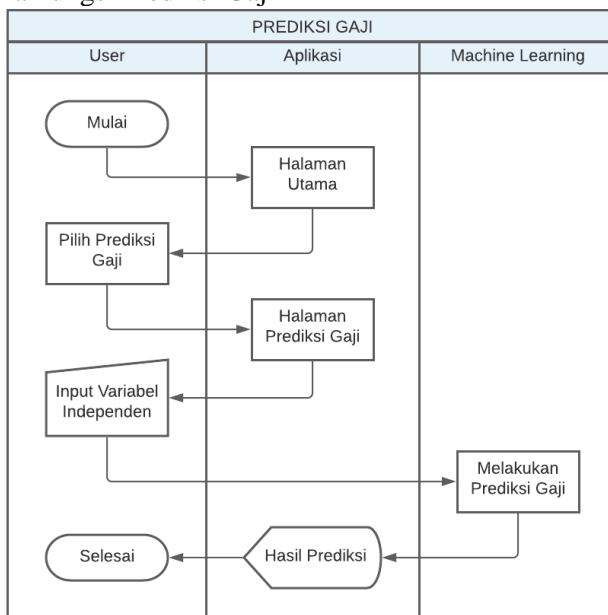
Kode diatas merupakan tampilan dari halaman hasil prediksi gaji. Terdapat *form* yang berisikan inputan *username* dan *password*. Selain itu terdapat hasil dari perhitungan prediksi gaji.

C. URL

```
[1] url('result', views.result, name='result')
```

Kode diatas merupakan URL untuk mengakses halaman hasil prediksi.

D. Logika Fungsi Prediksi Gaji



Gambar 5.47: 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.

5.6.6 Implementasi Proses CRUD Data Pegawai

1. Implementasi Menampilkan Data

A. Views

```
[1] def view(request):
    pegawai = Pegawai.objects.all()
[2]     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

```
[1] <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>
[2] <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 terdapat hasil dari perhitungan prediksi gaji.

C. URL

```
[1] path('view/', views.view),
```

Kode diatas merupakan URL untuk mengakses halaman data pegawai.

2. Implementasi Proses Tambah Data Pegawai

A. Views

[1]	def pgw(request): if request.method == "POST": form = PegawaiForm(request.POST)
[2]	if form.is_valid(): try: form.save() return redirect('/view')

	except: pass else: form = PegawaiForm() [3] 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*

```
[1] <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>  
[2] <div class="form-row">  
<div class="form-group col-md-6">  
<label for=" "><strong>Job Level :</strong></label>  
<div>{{ form.JobLevel }}</div>  
</div>  
</div>  
<div class="form-row">  
<div class="form-group col-md-13">  
<label for=" "><strong>Monthly Income :</strong></label>
```

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

<div class="float-right">
<div class="form-row">
<div class="form-group col-md-12">
<label for=" "><strong>Total Working Years : </strong></label>
<div>{{ form.TotalWorkingYears }}</div>
</div>
</div>
<div class="form-row">
<div class="form-group col-md-12">
<label for=" "><strong>Years At Company : </strong></label>
<div>{{ form.YearsAtCompany }}</div>
</div>
</div>

[3]   <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 terdapat button simpan untuk proses menyimpan data dan button batal untuk mereset data.

C. URL

```
[1] path('pgw/', views.pgw),
```

Kode diatas merupakan URL untuk mengakses halaman tambah pegawai.

3. Implementasi Proses Edit Data Pegawai

A. Views

```
[1] def edit(request, id):
    pegawai = Pegawai.objects.get(id=id)
    return render(request, 'edit.html', {'pegawai': pegawai})
[2] 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')
[3] 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*

```
[1] <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>
[2] <div>
<input    type="number"    class="form-control"
name="Age" value="{{ pegawai.Age }}">
</div>
</div>
</div>
<div class="form-row">
<div class="form-group col-md-15">
<label for="jenis_produk"><strong>Job Level :
</strong></label>
<div>
<input      type="number"      class="form-
control"      name="JobLevel"      value="{{
pegawai.JobLevel }}"/>
</div>
</div>
</div>
<div class="form-row">
<div class="form-group col-md-15">
<label for="jenis_produk"><strong>Monthly
Income : </strong></label>
```

```
<div>
<input type="number" class="form-control" name="MonthlyIncome" value="{{ pegawai.MonthlyIncome }}">
</div>
</div>
</div>
</div>
<div class="float-right">
<div class="form-row">
<div class="form-group col-md-15">
<label for="jenis_produk"><strong>Total Working Years : </strong></label>
<div>
<input type="number" class="form-control" name="TotalWorkingYears" value="{{ pegawai.TotalWorkingYears }}">
</div>
</div>
</div>
<div class="form-row">
<div class="form-group col-md-15">
<label for="jenis_produk"><strong>Years At Company : </strong></label>
<div>
<input type="number" class="form-control" name="YearsAtCompany" value="{{ pegawai.YearsAtCompany }}"/>
</div>
</div>
</div>
<hr>
<div class="form-group">
[3] <button type="submit" class="btn btn-primary">Simpan</button>
```

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

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

C. URL

```
[1] path('edit/<int:id>', views.edit),
[2] 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

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

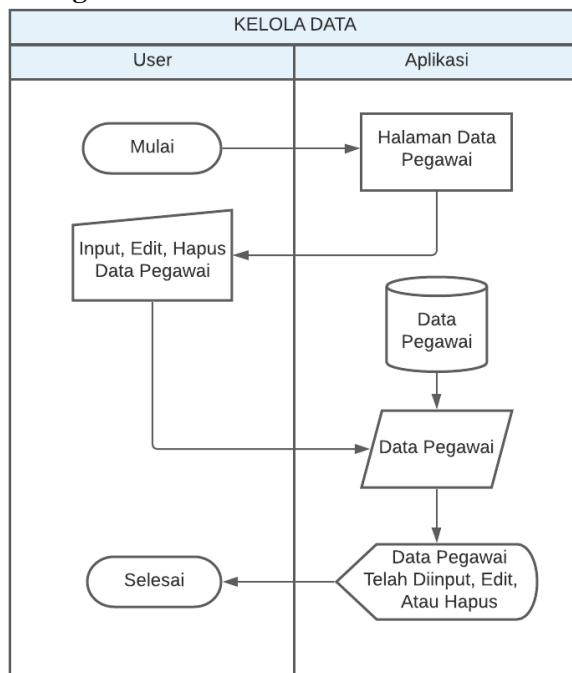
Fungsi *delete* digunakan untuk melakukan fungsi hapus data. Jika data berhasil terhapus maka hasil data yang terhapus akan dialihkan ke halaman data pegawai.

B. URL

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

Kode diatas merupakan URL untuk melakukan proses hapus data.

5. Logika Fungsi CRUD



Gambar 5.48: Flowmap Olah Data Pegawai

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

5.6.7 Implementasi Proses Menampilkan Visualisasi

A. Views

```
[1] 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

```
[1] <div class="float-left">
    <h5>Berikut adalah visualisasi data pegawai </h5>
</div>
<br>
<hr>
[2] <div class='tableauPlaceholder' id='viz164033535
7904' 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='SalaryVSYears
Experience4&#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>
[3] <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';
}
[4] var scriptElement = document.createElement
('script');
scriptElement.src = 'https://public.tableau.com/
javascripts/api/viz_v1.js';
vizElement.parentNode.insertBefore(script
Element, vizElement);
</script>
</div>
```

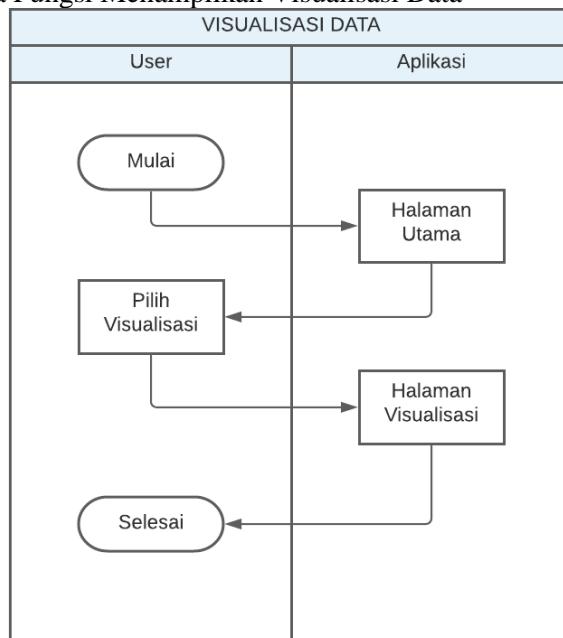
Kode diatas merupakan tampilan dari halaman visualisasi. Visualisasi tersebut dibuat dari platform Tableau yang telah dirancang sebelumnya.

C. URL

```
[1] url('visualisasi',views.visualisasi,name='
visualisasi')
```

Kode diatas merupakan URL untuk mengakses halaman visualisasi.

D. Logika Fungsi Menampilkan Visualisasi Data



Gambar 5.49: Flowmap Menampilkan Visualisasi Data

Keterangan :

1. Aplikasi menampilkan halaman utama.
2. *User* memilih menu visualisasi.
3. Aplikasi menampilkan halaman visualisasi.
4. Selesai.

5.6.8 Implementasi Proses Logout

A. Template

```
[1]  {% if user.is_authenticated %} 
[2]    <li class="nav-item">
        <a class="nav-link text-light float-right"
           href="{% url 'Logout' %}">Logout</a>
      </li>
```

```
[3]  {% endif %}
```

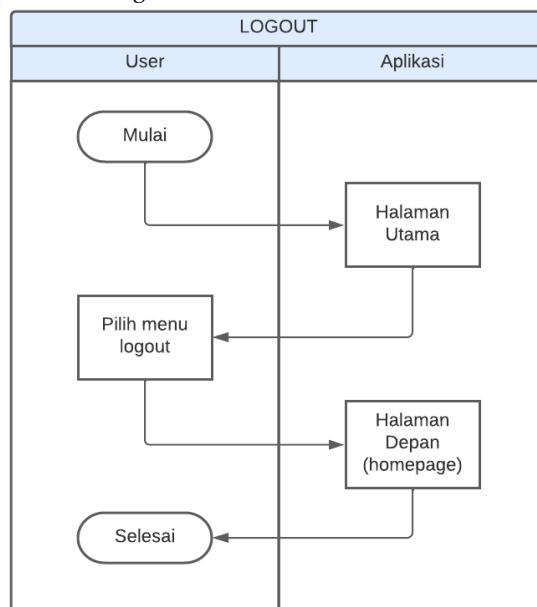
Kode diatas merupakan kode untuk tampilan dari menu *Logout*. Posisi menu *Logout* terdapat pada bagian navbar halaman web.

B. URL

```
[1]  path('Logout/', LogoutView.as_view(next_page='
homepage'), name="Logout"),
```

Kode diatas merupakan URL untuk mengakses halaman visualisasi.

C. Flowmap Proses Logout



Gambar 5.50: Flowmap Logout

Keterangan :

1. Aplikasi menampilkan halaman utama.
2. *User* memilih menu *Logout* pada navbar.
3. Aplikasi melakukan render ke halaman visualisasi.
4. Selesai.

5.7 Implementasi kode CSS Aplikasi

5.7.1 Implementasi Kode Link Bootstrap CSS

Berikut adalah kode dari link bootstrap CSS.

```
[1] <!-- Bootstrap CSS -->
<link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap
      /4.3.1/css/bootstrap.min.css" integrity="sha384-
      gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7
      x9JvoRxT2MZw1T" crossorigin="anonymous">
[2] <link rel="stylesheet" type="text/css" href="css/
      bootstrap.min.css">
[3] <link rel="stylesheet" type="text/css" href="js/
      bootstrap.min.js">
[4] <link rel="stylesheet" type="text/css" href="
      style.css">
```

Kode diatas diimplementasikan disetiap halaman pada komponen *templates framework* Django. Posisi code tersebut berada didalam tag head.

5.7.2 Implementasi Kode CSS

Berikut adalah kode dari CSS yang dibuat.

```
[1] <style>
      body{
```

```
[2] background: rgb(213, 213, 231);  
[3] }  
</style>
```

Kode diatas diimplementasikan disetiap halaman pada komponen *templates framework* Django. Posisi *code* tersebut berada didalam *tag style* yang menyatakan bahwa *tag* tersebut adalah *tag* untuk menerapkan kode CSS pada halaman *web*.

5.8 Halaman Antarmuka

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.

5.8.1 Implementasi Halaman Depan



Gambar 5.51: 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.

5.8.2 Antarmuka Halaman *Login*



Gambar 5.52: 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.

5.8.3 Antarmuka Halaman Registrasi

A screenshot of a registration page titled "SILAHKAN MEMBUAT AKUN !". It asks the user to input a new username and password. The "Username:" field has a note: "Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.". The "Password:" field has a note: "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. Your password can't be entirely numeric.". Below these fields is a "Password confirmation:" field with a note: "Enter the same password as before, for verification.". At the bottom left is a "Register" button.

Gambar 5.53: Halaman Registrasi Sistem

Gambar diatas merupakan implementasi dari halaman registrasi. Jika *user* belum memiliki akun untuk masuk sistem, maka *user* dapat

menginputkan *username* dan *password* lalu menekan tombol register.

5.8.4 Antarmuka Halaman *Dashboard*



Gambar 5.54: 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.

5.8.5 Antarmuka Halaman Prediksi

The screenshot shows a web application interface titled "PREDIKSI GAJI PEGAWAI". At the top, there is a navigation bar with links: "Prediksi Gaji Pegawai", "Prediksi", "Data Pegawai", "Visualisasi", and "Logout". Below the title, a instruction text says "Silahkan inputkan tahun dalam bentuk angka". There are four input fields: "Inputkan Usia", "Inputkan Job Level", "Inputkan Total Tahun Bekerja", and "Inputkan Total Tahun Bekerja Di Perusahaan". Below these fields is a large green button labeled "Lakukan Prediksi". In the bottom right corner of the main form area, there is a small grey button labeled "Kembali".

Gambar 5.55: 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.

5.8.6 Antarmuka Halaman Hasil Prediksi



Gambar 5.56: 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.

5.8.7 Antarmuka Halaman Data Pegawai

The screenshot shows a web application interface. At the top, there is a navigation bar with links: 'Prediksi Gaji Pegawai', 'Prediksi', 'Data Pegawai', 'Visualisasi', and 'Logout'. Below the navigation bar is a main content area with a dark header containing the text 'DATA PEGAWAI'. Underneath the header, a message states: 'Berikut adalah list data pegawai.' To the right of this message is a button labeled '(+) Tambah Pegawai'. Below the message is a table with the following data:

ID	Age	JobLevel	MonthlyIncome	TotalWorkingYears	YearsAtCompany	Action
2	38.0	2	8463	6	5	<button>Edit</button> <button>Delete</button>
3	45.0	3	9724	25	1	<button>Edit</button> <button>Delete</button>
4	36.0	2	5914	16	13	<button>Edit</button> <button>Delete</button>
5	34.0	1	2579	8	8	<button>Edit</button> <button>Delete</button>
6	38.0	1	4230	6	5	<button>Edit</button> <button>Delete</button>
7	39.0	1	2232	7	3	<button>Edit</button> <button>Delete</button>

Gambar 5.57: 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.

5.8.8 Antarmuka Halaman Tambah Data Pegawai

The screenshot shows a web application interface for adding employee data. At the top, there's a navigation bar with a logo and links for 'Prediksi Gaji Pegawai', 'Prediksi', 'Data Pegawai', 'Visualisasi', and 'Logout'. The main content is a form titled 'TAMBAH DATA' with the instruction 'Silahkan masukan data pegawai.' It contains five input fields: 'Age', 'Job Level', 'Monthly Income', 'Total Working Years', and 'Years At Company'. Below the inputs are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). The footer of the page includes a copyright notice: '© 2021 Copyright: ABACH DAN DHANTI'.

Gambar 5.58: 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*.

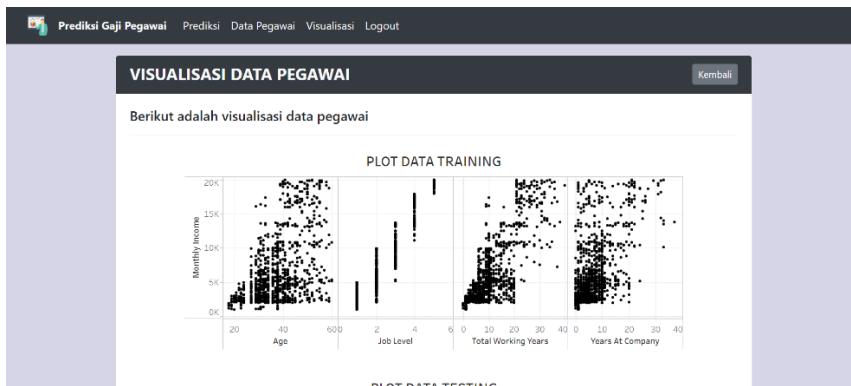
5.8.9 Antarmuka Halaman Edit Data Pegawai

The screenshot shows a web application interface titled 'EDIT DATA'. At the top right is a 'Kembali' (Back) button. The main area contains four input fields arranged in a 2x2 grid. The first row has 'Age:' with value '38,0' and 'Total Working Years:' with value '6'. The second row has 'Job Level:' with value '2' and 'Years At Company:' with value '5'. Below these is a 'Monthly Income:' field containing '8463'. At the bottom right are two buttons: 'Simpan' (Save) in blue and 'Batal' (Cancel) in red. The footer of the page includes a copyright notice: '© 2021 Copyright: ABACH DAN DHANTI'.

Gambar 5.59: 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*.

5.8.10 Antarmuka Halaman Visualisasi



Gambar 5.60: 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

5.9 Implementasi Metode Regresi Linier Berganda

5.9.1 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 :

1. Line 1 : Untuk mengimport *library* yang dibutuhkan.
2. Line 2 : Mengimport data ke python kemudian disimpan dalam variabel dengan nama df_train.
3. Line 3 : Menampilkan info detail tabel/data yang disimpan.

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

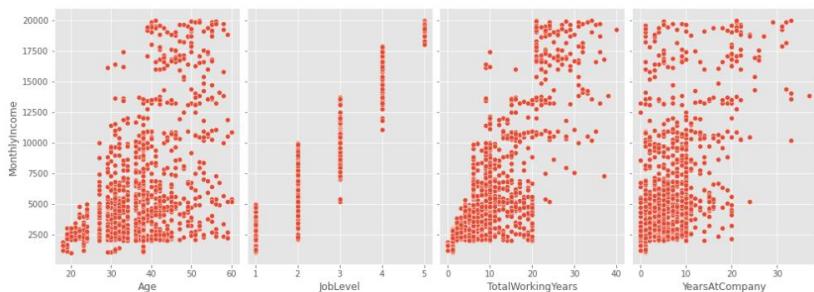
A. Visualisasi Data Training

```
[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 5.61: Visualisasi Data Training

Keterangan :

1. Line 22 : Membuat *style* untuk diagram plot.
2. 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.
3. Line 24 : Membuat heatmap untuk melihat korelasi antar variabel.

5.9.2 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.Distance  
FromHome.mean())  
  
  
# Missing Value in BusinessTravel  
  
BusinessTravel = df_test['BusinessTravel']  
  
df_test.BusinessTravel =  
df_test.BusinessTravel.fillna(value=df_test.BusinessTr  
avel.mean())  
  
  
# Missing Value in MartialStatus  
  
MaritalStatus = df_test['MaritalStatus']
```

```
df_test.MaritalStatus =  
df_test.MaritalStatus.fillna(value=df_test.MaritalStat  
us.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', 'JobSatisfacti  
on', 'MaritalStatus', 'MonthlyRate', 'NumCompaniesWorked'  
, 'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceR  
ating', 'RelationshipSatisfaction', 'StandardHours', 'Sto  
ckOptionLevel', 'TrainingTimesLastYear', 'WorkLifeBalanc  
e', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'Yea  
rsWithCurrManager'], 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 :

1. Line 25 : Melakukan *import data testing* kemudian disimpan dalam variabel df_test.
2. 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.
3. Line 28 : Cek apakah ada data yang kosong pada tabel.
4. Line 29 : Membuat variabel kemudian melakukan pengisian data kosong pada variabel dengan nilai *mean*-nya.
5. Line 30 : Cek apakah masih terdapat data yang kosong atau tidak.
6. Line 31 : Cek apakah ada data yang kosong pada tabel.
7. Line 32 : Melakukan *drop* variabel yang tidak diperlukan.
8. Line 33 : Membuat tabel korelasi setelah proses *drop* variabel.
9. Line 34 : Menetapkan variabel independent (sumbu x) yaitu variabel Age, JobLevel, TotalWorkingYears, dan YearsAtCompany. Kemudian menetapkan variabel variabel dependen (sumbu y) yaitu MonthlyIncome.
10. Line 35 : Melakukan *export* data ke format file csv.
11. 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.
12. Line 37 : membuat heatmap untuk melihat korelasi antar variabel.
13. 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.

14. Dari model tersebut didapat nilai koefisien dari variabel Independen. Nilai koefisien dari Age adalah -5,054 yang berarti tiap karyawan yang mengalami pertambahan satu tahun umur kerja, maka akan mengalami penurunan gaji sebesar 5,054. Nilai koefisien dari JobLevel adalah 3871,7530 yang berarti tiap karyawan yang mengalami pertambahan satu tingkat job level akan mengalami kenaikan gaji sebesar 3871,7530. Nilai koefisien TotalWorkingYears adalah 46,9405 yang berarti tiap karyawan yang mengalami pertambahans satu tahun pengalaman bekerja akan mengalami kenaikan kerja sebesar 46,9405. Nilai koefisien YearsAtCompany adalah -9,8460 yang berarti tiap karyawan yang mengalami pertambahan satu tahun akan mengalami penurunan gaji sebesar 9,8460.
15. Line 39 : Mencari konstanta/*intercept* menggunakan regressor, kemudian ditampilkan.
16. Intercept 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.
17. Line 40 : Melakukan prediksi *data testing* menggunakan model *machine learning*. Kemudian buat kolom baru yang bernama MonthlyIncome Prediction yang berisikan nilai prediksi.
18. Berikut persamaan umum dari model linear regresi multivariabel :

$$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_4X_4 + \dots + \beta_nX_n$$

19. β_0 adalah nilai *intersept* 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 *Intersept* didapat, maka persamaan regresi linear multivariabel sebagai berikut :

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

20. Keterangan:

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

21. Maka dapat disimpulkan, persamaan regresi linear multivariabel sebagai berikut :
- $$\text{MonthlyIncome} = -1728 - 5,054(\text{Age}) + 3871,7530(\text{JobLevel}) \\ + 46,9405(\text{TotalWorkingYears}) - 9,8460(\text{YearsAtCompany})$$
22. Line 41 : Melakukan export data setelah prediksi ke dalam format file excel.

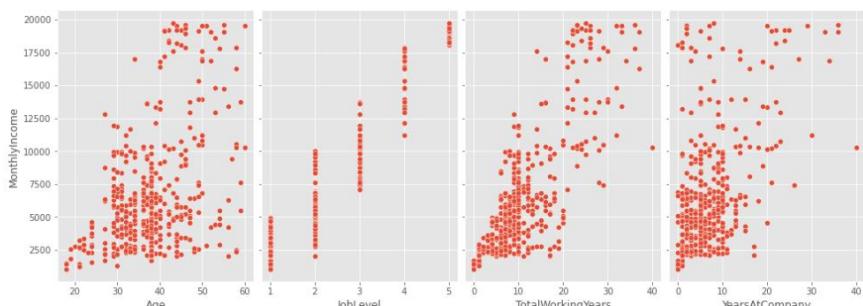
A. Visualisasi Data Testing

```
[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 5.62: Visualisasi Data Testing

Keterangan :

1. 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.
2. Line 37 : membuat heatmap untuk melihat korelasi antar variabel.

5.9.3 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  46.9466   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 5.63: Model OLS

Keterangan :

1. 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)

Uji F dikenal dengan Uji serentak atau uji Model/Uji Anova, yaitu uji untuk melihat bagaimana pengaruh semua variable bebasnya secara Bersama-sama terhadap variable terikatnya. Uji F digunakan untuk menguji apakah model regresi yang dibuat baik/signifikan atau tidak baik/non signifikan(Hidayat, 2014). 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

Keterangan :

1. 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. Fs

Fs adalah hasil akhir dari analisis ANOVA. Nilai Fs ini yang dikenal dengan F hitung dalam pengujian hipotesa dibandingkan dengan nilai p-value. Jika $F_s > P\text{-value}$, 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

Uji-t dikenal dengan uji parsial adalah uji yang digunakan untuk menguji bagaimana pengaruh masing-masing variable bebasnya secara sendiri-sendiri terhadap variable terikatnya(Hidayat, 2014). 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	$1,41965 \times 10^{-13}$
Age	$4,643777 \times 10^{-1}$
JobLevel	0
TotalWorkingYears	$6,771641 \times 10^{-5}$
YearsAtCompany	$3,13660 \times 10^{-1}$

Keterangan :

1. 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.

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

Berdasarkan uji-t dapat diambil hipotesa sebagai berikut :

Nilai variabel X1 (Age) berada di atas taraf signifikansi yang berarti terima H_1 .

Nilai variabel X2 (JobLevel) berada di bawah taraf signifikansi yang berarti terima H_0 .

Nilai variabel X3 (TotalWorkingYears) di bawah taraf signifikansi yang berarti terima H_0 .

Nilai variabel X4 (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.

C. R-Square

R-Square adalah sebuah nilai yang menyatakan seberapa sesuai hasil prediksi model mendekati data yang sebenarnya. R-Square disebut juga sebagai koefisien determinasi yang menjelaskan seberapa jauh data dependen dapat dijelaskan oleh data independent. 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. Sebagai contoh, jika R-Square bernilai 0.6 berarti 60% sebaran variable dependen dapat dijelaskan oleh variable independen. Sisanya 40% tidak dapat dijelaskan oleh variable independent atau dapat dijelaskan oleh variable diluar variable independent (komponen error). Jika nilai R-Square kecil, artinya komponen error yang besar. Sebagai contoh nilai R-Square Adjusted pada penelitian ini adalah 0.50. Nilai R-Square adjusted berfungsi untuk mengatasi masalah yang sering dijumpai pada nilai R-Square, yaitu terus bertambahnya nilai jika terdapat penambahan variable independent ke dalam model, sedangkan pada R-Square adjusted dapat mengukur tingkat keyakinan penambahan variable independent secara tepat dalam menambah daya prediksi model(Meiryani, 2021a).

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 5.64: Model OLS

Berdasarkan tabel ANOVA, nilai koefisien determinasi (R-Square) adalah 0,909 atau 90,9%. Maka, MonthlyIncome dipengarui 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.

D. 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'] = olsmod.predict(X)  
df_test_new2['residual'] = olsmod.resid  
df_test_new2
```

Keterangan :

1. 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.

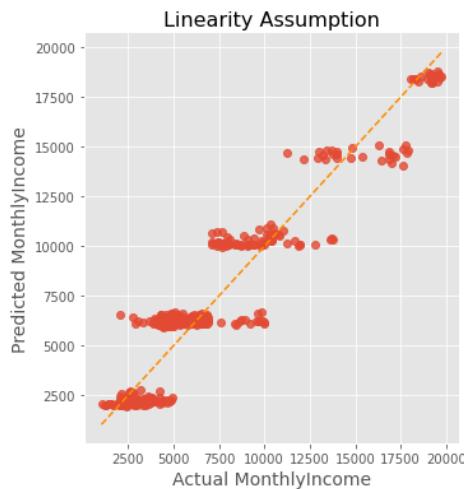
a. Linearitas

Secara sederhana, uji linearitas adalah pengujian untuk memeriksa apakah terdapat hubungan linear antara variable independent dengan variable dependent. 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 5.65: Grafik Asumsi Linear

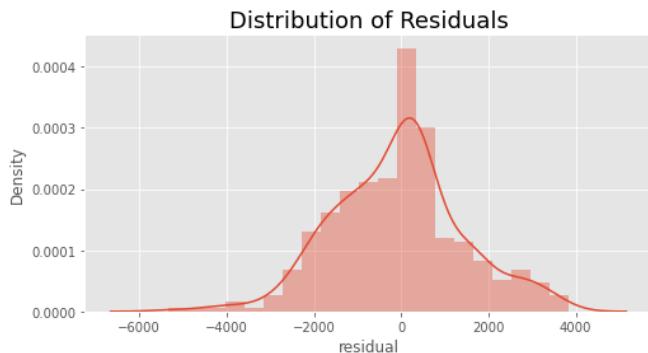
Keterangan :

1. Line 46 : Untuk menampilkan grafik linearitas dari perbandingan nilai MonthlyIncome dengan MonthlyIncomePrediction.
 2. 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
- Uji Normalitas adalah sebuah uji yang dilakukan dengan tujuan untuk menilai sebaran data pada sebuah kelompok data atau variable, apakah sebaran data tersebut berdistribusi normal ataukah tidak(AriRiz, 2012). Pendekatan *informal* untuk menguji normalitas adalah membandingkan histogram data sampel dengan kurva probabilitas normal. Distribusi empiris dari data (histogram) harus berbentuk lonceng dan menyerupai distribusi normal. Ini mungkin sulit untuk melihat apakah sampelnya kecil. Dalam memeriksa normalitas secara visual dengan memplot distribusi frekuensi, juga disebut histogram, dari data dan secara visual membandingkannya dengan distribusi normal (dilapisi merah)(Iskandar *et al.*, 2021). 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 5.66: Grafik Distribusi Residual

Keterangan :

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

Dapat diketahui hipotesa sebagai berikut :

H_0 = Residual terdistribusi normal.

H_1 = 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 H_0 terima H_1 atau dapat dikatakan residual terdistribusi secara tidak normal. Sehingga disimpulkan asumsi normalitas terpenuhi.

c. Multikolinieritas

Model regresi yang baik adalah yang tidak terjadi multikolinearitas. Uji multikolinearitas dilakukan untuk menguji apakah terdapat korelasi antar variable bebas dalam model regresi. Multikolinearitas menguji apakah terdapat linier yang sempurna antara beberapa atau semua variable yang menjelaskan model regresi (Ajija, 2011). Uji multikolinearitas dilakukan dengan tujuan untuk mengetahui apakah pada suatu model regresi ditemukan adanya korelasi antar variable independent (Ghocali, 2016). Pengujian multikolinearitas bertujuan untuk mengetahui apakah model regresi ditemuka adanya korelasi antar variable independent atau variable bebas (Ghozali, 2016). Untuk menemukan terdapat atau tidaknya multikolinearitas pada model regresi dapat diketahui dari nilai toleransi dan nilai variance inflation factor (VIF). Nilai tolerance mengukur variabilitas dari variable bebas yang terpilih yang tidak dapat dijelaskan oleh variable bebas lainnya. Jadi nilai tolerance rendah sama dengan nilai VIF tinggi, dikarenakan $VIF = 1/tolerance$, dan menunjukkan terdapat kolinearitas yang tinggi. Nilai cut off yang digunakan adalah untuk nilai tolerance 0.10 atau nilai VIF diatas angka 10. Uji Multikolinearitas bertujuan untuk menguji dan mengetahui apakah dalam suatu model regresi ditemukan adanya korelasi

yang tinggi atau sempurna antar variabel independen. Pengujian ini dapat diketahui dengan melihat nilai toleransi dan nilai variance inflation factor (VIF). Pengujian dilakukan dengan melihat nilai VIF atau *variance inflation factors*. Apabila nilai *centered VIF (Variance Inflation Factor)*. Pengujian dapat dilakukan dengan melihat nilai *Tolerance* dan *Variance Inflation Factor (VIF)* pada model regresi. Kriteria pengambilan keputusan terkait uji multikolinearitas adalah sebagai berikut (Ghozali, 2016) :

1. Jika nilai $VIF < 10$ atau nilai *Tolerance* $> 0,01$, maka dinyatakan tidak terjadi multikolinearitas.
2. Jika nilai $VIF > 10$ atau nilai *Tolerance* $< 0,01$, maka dinyatakan terjadi multikolinearitas.
3. Jika koefisien korelasi masing-masing variabel bebas $> 0,8$ maka terjadi multikolinearitas. Tetapi jika koefisien korelasi masing-masing variabel bebas $< 0,8$ maka tidak terjadi multikolinearitas(Meiryani, 2021c, 2021b).

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)

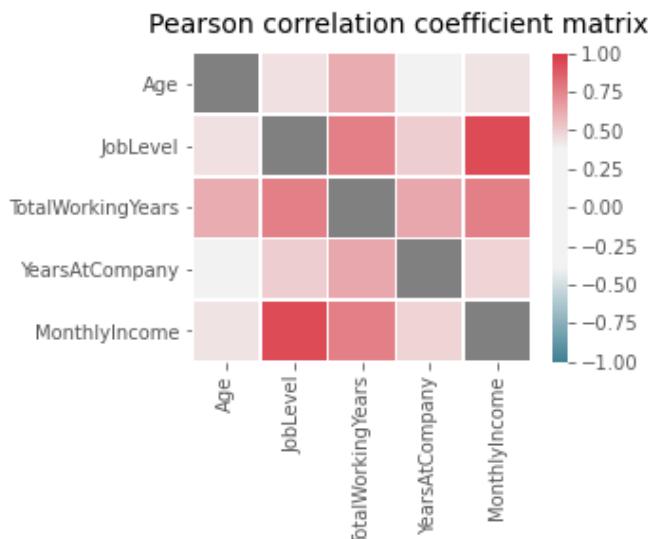
# Generate a mask for the diagonal cell
mask = np.zeros_like(corr, dtype=np.bool)
np.fill_diagonal(mask, val=True)

# 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',
labelsize=10)
# fig.tight_layout()
```



Gambar 5.67: Tabel Matriks Pearson Korelasi

Keterangan :

1. 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 VIF 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
```

No	VIF	Variable
1	28,655370	Intercept

2	1,690786	Age
3	2,489052	JobLevel
4	4,140803	TotalWorkingYears
5	1,739893	YearsAtCompany

Keterangan :

1. 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. Suatu model regresi dapat dikatakan baik ketika terbebas dari autokorelasi. Uji autokorelasi yang dapat muncul karena adanya observasi yang berurutan sepanjang waktu dan saling berkaitan satu sama lainnya (Ghozali, 2016). Permasalahan ini muncul karena residual tidak bebas pada satu observasi ke observasi lainnya. Uji autokorelasi bertujuan untuk menunjukkan korelasi anggota observasi yang diurutkan berdasarkan waktu atau ruang (Ajija, 2011). Uji autokorelasi bertujuan untuk menguji apakah dalam suatu model regresi linear ada korelasi antara kesalahan pengganggu pada periode t dengan kesalahan pada periode t-1 (sebelumnya). Jika terjadi korelasi, maka dinamakan ada problem autokorelasi. Masalah ini timbul karena residual (kesalahan pengganggu) tidak bebas dari satu observasi ke observasi lainnya. Jika data observasi di atas 100 data sebaiknya menggunakan uji Lagrange Multiplier(Meiryani, 2021b). 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 negatif 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 :

1. 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

Uji homoskedastisitas digunakan dalam menguji error atau galat dalam model statistik untuk melihat apakah varians atau keragaman dari error terpengaruh oleh faktor lain atau tidak, misalnya untuk analisis data runtun waktu, apakah keragaman error nya terpangaru oleh waktu atau tidak, atau kalau datanya cross section maka apakah varians dari error berubah-ubah setiap amatan atau tidak(Kompasiana, 2015).

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.

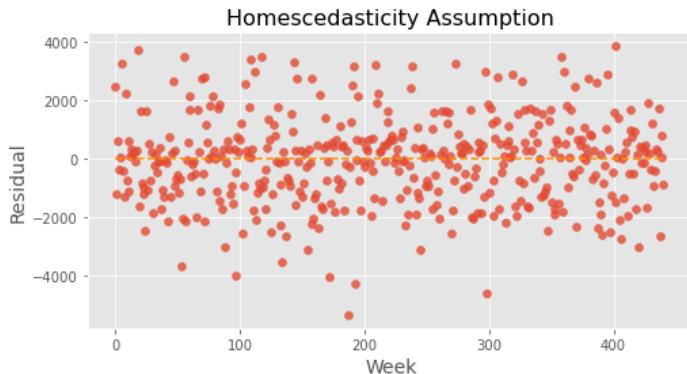
```
[51] # Plotting the residuals
      plt.subplots(figsize=(8, 4))

      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 5.68: Homoskedastisitas

Keterangan :

1. 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 :

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

Bab VI

Kesimpulan Dan Saran

6.1 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.

6.2 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

- Abdurahman, H., Riswaya, A.R. and Id, A. (2014) “Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha Bhakti STMIK Mardira Indonesia,” *Jurnal Computech & Bisnis*, 8(2), pp. 61–69. Available at: <https://jurnal.stmik-mi.ac.id/index.php/jcb/article/view/114> (Accessed: June 16, 2022).
- Adrianova Eka, Y. and Anyan (2020) “Implementasi Model Regresi Linear Sederhana Untuk Prediksi Gaji Berdasarkan Pengalaman Lama Bekerja,” *Journal Education and Technology*, 1(2), pp. 56–70. doi:10.31932/jutech.v1i2.1289.
- Ariesanto Akhmad, E.P. (2020) “Data Mining Menggunakan Regresi Linear untuk Prediksi Harga Saham Perusahaan Pelayaran,” *Jurnal Aplikasi Pelayaran dan Kepelabuhanan*, 10(2), p. 120. doi:10.30649/japk.v10i2.83.
- AriRiz (2012) *Uji Asumsi Klasik (Uji Normalitas)*, <http://fe.unisma.ac.id/>. Available at: <http://fe.unisma.ac.id/MATERI%20AJAR%20DOSEN/EKOMETRIK/AriRiz/MA%20Uji%20Normalitas.pdf> (Accessed: June 14, 2022).
- Bansal, U. *et al.* (2021) “Empirical analysis of regression techniques by house price and salary prediction,” *IOP Conference Series: Materials Science and Engineering*, 1022(1), p. 13. doi:10.1088/1757-899X/1022/1/012110.
- D. Tri Octafian (2011) “Desain Database Sistem Informasi Penjualan Barang,” *Jurnal Teknologi Dan Informatika (TEKNOMATIKA)*, 1(2), pp. 148–157. doi:10.30998/faktorexacta.v9i2.795.
- Dan, S. and Pratikno, B. (2014) “Regresi Linear Bivariat Simpel Dan Aplikasinya Pada Data Cuaca Di Cilacap,” *JMP*, 6(1), pp. 45–52. doi:10.20884/1jmp.2014.6.1.2902.
- daqiqil.id (2022) *Articles with Machine Learning*, [daqiqil.id](https://ibnu.daqiqil.id/tag/machine-learning/). Available at: <https://ibnu.daqiqil.id/tag/machine-learning/> (Accessed: June 7, 2022).
- Diaz, D.C.P., Sulistiowati and Lemantara, J. (2016) “Rancang Bangun Aplikasi Penjualan Online Pada CV. Mitra Techno Sains (Studi Kasus: CV. Mitra Techno Sains),” *JSIKA*, 5(12), pp. 1–7. Available at:

<https://jurnal.dinamika.ac.id/index.php/jsika/article/download/1480/900>
(Accessed: June 14, 2022).

Exsight (2020) *Cara Install Python Dab Jupyter Notebook Di Windows [2021]*. Available at: <https://exsight.id/blog/2021/01/10/cara-install-python-dan-jupyter-notebook-di-windows/> (Accessed: June 13, 2022).

Fridayanthie, E.W. and Mahdiati, T. (2016) “Rancang Bangun Sistem Informasi Permintaan Atk Berbasis Intranet (Studi Kasus: Kejaksaan Negeri Rangkasbitung),” *Jurnal Khatulistiwa Informatika*, 4(2), pp. 126–138. doi:10.31294/jki.v4i2.1264.

Ginting, F., Buulolo, E. and Siagian, E.R. (2019) “Implementasi Algoritma Regresi Linear Sederhana Dalam Memprediksi Besaran Pendapatan Daerah (Studi Kasus: Dinas Pendapatan Kab. Deli Serdang),” *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, 3(1), pp. 274–279. doi:10.30865/komik.v3i1.1602.

Habibie Asnar, Z. (2013) “Pengaruh Tata Ruang Kantor Terhadap Produktivitas Kerja Pegawai Di Pusat Kajian Dan Pendidikan Dan Pelatihan Aparatur III Lembaga Administrasi Negara (PKP2A III Lan) Samarinda,” *eJournal IlmuPemerintahan*, 1(4), pp. 1488–1500. Available at: <http://perpustakaan.unmul.ac.id/ejournal/index.php/um/article/view/118> (Accessed: June 16, 2022).

Hasan Bisri Isa Alfaris, Choirul Anam and Ali Masy'an (2013) “Implementasi Black Box Testing Pada Sistem Informasi Pendaftaran Santri Berbasis Web Dengan Menggunakan PHP Dan MySQL,” *Jurnal Sains dan Teknologi*, 6(1), pp. 23–38. doi:10.32764/saintekbu.v6i1.64.

Hidayat, A. (2014) *Uji F dan Uji T*, <https://www.statistikian.com/>. Available at: <https://www.statistikian.com/2013/01/uji-f-dan-uji-t.html> (Accessed: June 14, 2022).

Homepage, J. et al. (2019) “Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper,” *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), pp. 75–82. doi:10.31294/ijcit.v5i1.7951.

Indra Sanjaya, F. and Heksaputra, D. (2020) “Prediksi Rerata Harga Beras Tingkat Grosir Indonesia dengan Long Short Term Memory,” *Jurnal Teknik Informatika dan Sistem Informasi*, 7(2), pp. 163–174. doi:10.35957/jatisi.v7i2.388.

Institut Teknologi Bandung (2022) *Tableau*. Available at: <https://ditsti.itb.ac.id/en/tableau/> (Accessed: June 13, 2022).

Iskandar, A. et al. (2021) *Statistika Bidang Teknologi Informasi*. Kita Menulis. Available at: <https://kitamenulis.id/2021/04/23/statistika-bidang-teknologi-informasi/> (Accessed: June 14, 2022).

Iskandar Akbar, Muttaqin and Vita Dewi Sarini (2021) *FullBook Statistika Bidang Teknologi Informasi*.

Jatnika Hendra (2013) *Pengantar Sistem Basis Data Memahami Konsep Dasar Dan Tuntutan Praktis Pemrograman Perancangan Database*. Penerbit Andi Offset.

Katemba, P. and Koro Djoh, R. (2017) “Prediksi Tingkat Produksi Kopi Menggunakan Regresi Linear,” *Jurnal Ilmiah Flash*, 3(1), pp. 42–51. doi:10.5281/zenodo.1117136.

Kergroach, S. (2017) “Industry 4.0: New challenges and opportunities for the labour market,” *Foresight and STI Governance*, 11(4), pp. 6–8. doi:10.17323/2500-2597.2017.4.6.8.

Kompasiana (2015) *Uji Homoskedastisitas Statistik*, <https://www.kompasiana.com/>. Available at: <https://www.kompasiana.com/jokoade/54f6aa33a33311f7598b4604/uji-homoskedastisitas-statistik> (Accessed: June 14, 2022).

Manda, M.I. and Dhaou, S. ben (2019) “Responding to the challenges and opportunities in the 4th industrial revolution in developing countries,” in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*. ICST, pp. 244–253. doi:10.1145/3326365.3326398.

Meiryani (2021a) *Memahami R Square (Koefisien Determinasi) Dalam Penelitian Ilmiah*, <https://accounting.binus.ac.id/>. Available at: <https://accounting.binus.ac.id/2021/08/12/memahami-r-square-koefisien-determinasi-dalam-penelitian-ilmiah/> (Accessed: June 14, 2022).

Meiryani (2021b) *Memahami Uji Autokorelasi Dalam Model Regresi*, <https://accounting.binus.ac.id/>. Available at: <https://accounting.binus.ac.id/2021/08/06/memahami-uji-autokorelasi-dalam-model-regresi/> (Accessed: June 14, 2022).

Meiryani (2021c) *Memahami Uji Multikolineritas Dalam Model Regresi*, <https://accounting.binus.ac.id/>. Available at:

[https://accounting.binus.ac.id/2021/08/06/memahami-uji-multikolinearitas-dalam-model-regresi/#:~:text=Uji%20Multikolinearitas%20bertujuan%20untuk%20mengujii,variance%20inflation%20factor%20\(VIF\).](https://accounting.binus.ac.id/2021/08/06/memahami-uji-multikolinearitas-dalam-model-regresi/#:~:text=Uji%20Multikolinearitas%20bertujuan%20untuk%20mengujii,variance%20inflation%20factor%20(VIF).) (Accessed: June 16, 2022).

Nafi'iyah, N. (2021) "Perbandingan Regresi Linear, Backpropagation Dan Fuzzy Mamdani Dalam Prediksi Harga Emas," *Journal of Information System, Applied, Management, Accounting and Research*, 5(2), p. 331. doi:10.52362/jisamar.v5i2.420.

Nilakusmawati, D.P.E. and Made, S. (2016) "Model Regresi Multivariat Analisis Kesejahteraan Pedagang Kaki Lima Berdasarkan Karakteristik Sosial Ekonomi," *Prosiding SEMINAR NASIONAL MATEMATIKA II* , pp. 45–54. Available at: <https://erepo.unud.ac.id/id/eprint/7927/> (Accessed: June 16, 2022).

Padilah, T.N. and Adam, R.I. (2019) "Analisis Regresi Linear Berganda Dalam Estimasi Produktivitas Tanaman Padi Di Kabupaten Karawang," *Jurnal Pendidikan Matematika dan Matematika*, 5(2), pp. 117–128. doi:10.24853/fbc.5.2.117-128.

Pahlevi, O., Mulyani, A. and Khoir, M. (2018) "Sistem Informasi Inventory Barang Menggunakan Metode Object Oriented Di PT. Livaza Teknologi Indonesia Jakarta," *Jurnal PROSISKO*, 5(1), pp. 27–35. Available at: <https://ejurnal.lppmunsera.org/index.php/PROSISKO/article/view/587> (Accessed: June 16, 2022).

Pan, X. et al. (2020) "The Correlation Analysis Between Salary Gap and Enterprise Innovation Efficiency Based on the Entrepreneur Psychology," *Frontiers in Psychology*, 11, pp. 1–10. doi:10.3389/fpsyg.2020.01749.

Pangaribuan, O.C. and Irwansyah, I. (2019) "Media Cetak Indonesia di Era Revolusi Industri 4.0," *Jurnal Pewarta Indonesia*, 1(2), pp. 134–145. doi:10.25008/jpi.v1i2.11.

Pertiwi, M.W. and Indrajit, R.E. (2017) "Metode Regresi Linier Untuk Prediksi Pengadaan Inventaris Barang," *Symposium Nasional Ilmu Pengetahuan dan Teknologi (SIMNASIPTEK)*, 1(1), pp. 27–30. Available at: <https://jurnal.mdp.ac.id/index.php/jatisi/article/download/701/219/> (Accessed: June 13, 2022).

Prasetyo, B. and Trisyanti, U. (2018) “Revolusi Industri 4.0 Dan Tantangan Perubahan Sosial,” *Prosiding SEMATEKSOS* 3, pp. 22–27. doi:10.12962/j23546026.y2018i5.4417.

Prasetyo, H. and Sutopo, W. (2017) *Perkembangan Keilmuan Teknik Industri Menuju Era Industri 4.0, Seminar dan Konferensi Nasional IDEC*.

Puteri, K. and Silvanie, A. (2020) “Machine Learning Untuk Model Prediksi Harga Sembako Dengan Metode Regresi Linier Berganda,” *Jurnal Nasional Informatika*, 1(2), pp. 82–94. Available at: <https://ejournal-ibik57.ac.id/index.php/junif/article/view/134> (Accessed: June 16, 2022).

Putri, T.N., Yordan, A. and Lamkaruna, D.H. (2019) “Peramalan Penerimaan Mahasiswa Baru Universitas Samudra Menggunakan Metode Regresi Linear Sederhana,” *Jurnal Teknologi Informatika*, 2(1), pp. 2654–2617. Available at: <http://jurnal.ummu.ac.id/index.php/J-TIFA/article/view/237> (Accessed: June 16, 2022).

Rahimah, A.N., Rusdianto, D.S. and Ananta, M.T. (2019) “Pengembangan Sistem Pengelolaan Ruang Baca Berbasis Web Dengan Menggunakan Django Framework (Studi Kasus: Ruang Baca Fakultas Ilmu Komputer Universitas Brawijaya),” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3(5), pp. 4439–4446. Available at: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5227> (Accessed: June 16, 2022).

Rinda Faradilla (2021) *Gaji: Pengertian, Fungsi dan Tujuannya*, IDN Times. Available at: <https://www.idntimes.com/business/finance/rinda-faradilla/apa itu-gaji/5> (Accessed: June 7, 2022).

Sagala, E. (2018) “Analisis PPH 21 Terhadap Gaji Karyawan Pada PT. Kencana Utama Sejati,” *Jurnal Bisnis Kolega*, 4(2), pp. 55–63. Available at: <https://ejournal.pmci.ac.id/index.php/jbk/article/view/20> (Accessed: June 16, 2022).

Saiful A, Andryana S and Gunaryati A (2012) “Prediksi Harga Rumah Menggunakan Web Scrapping Dan Machine learning Dengan Algoritma Linear Regression,” *Jurnal Teknik Informatika dan Sistem Informasi*, 8(1), pp. 41–50. Available at: <https://jurnal.mdp.ac.id/index.php/jatisi/article/download/701/219/> (Accessed: June 13, 2022).

Santoso, A.B., S, M.J.D. and Setyawan, H.B. (2013) “Rancang Bangun Sistem Informasi Monitoring Dan Evaluasi Kinerja Mesin Pada Pkis Sekar Tanjung,”

Jurnal Sistem Informasi, 2(2), pp. 54–65. Available at: <https://media.neliti.com/media/publications/248370-rancang-bangun-sistem-informasi-monitori-987ff9a3.pdf> (Accessed: June 14, 2022).

Saputra, D. and Fathoni Aji, R. (2018) “Analisis Perbandingan Performa Web Service Rest Menggunakan Framework Laravel, Django Dan Ruby On Rails Untuk Akses Data Dengan Aplikasi Mobile (Studi Kasus: Portal E-Kampus STT Indonesia Tanjungpinang),” *Bangkit Indonesia*, 2(2), pp. 17–22. doi:10.52771/bangkitindonesia.v7i2.90.

Saputra Tamrin, A., Rumapea, P. and Mambo, R. (2017) “Pengaruh Profesionalisme Kerja Pegawai Terhadap Tingkat Kepuasan Pelanggan Pada Kantor PT. Taspen Cabang Manado,” *Jurnal Administrasi Publik*, 3(46), pp. 1–9. Available at: <https://ejournal.unsrat.ac.id/index.php/JAP/article/view/16283/15786> (Accessed: June 16, 2022).

das Sayan, Barik Rupashri and Mukherjee Ayush (2020) “Salary Prediction Using Regression Techniques,” *Engineering Research Network (EngRN)* [Preprint]. doi:10.2139/ssrn.3526707.

Shahroom, A.A. and Hussin, N. (2018) “Industrial Revolution 4.0 and Education,” *International Journal of Academic Research in Business and Social Sciences*, 8(9), pp. 314–319. doi:10.6007/ijarbss/v8-i9/4593.

small-business-tracker.com (2022) *Apa itu Jupyter Notebook? Analisis data menjadi lebih mudah.* Available at: <https://ind.small-business-tracker.com/what-is-jupyter-notebook-data-analysis-made-easier-574170> (Accessed: June 16, 2022).

Tableau (2022) *Install Tableau Desktop or Tableau Prep Builder from the User Interface.* Available at: https://help.tableau.com/current/desktopdeploy/en-us/desktop_deploy_download_and_install.htm (Accessed: June 16, 2022).

Techmania (2019) *Cara Install Visual Studio Code di Windows 10.* Available at: <https://www.techcelup.com/> (Accessed: June 7, 2022).

Triyanto, E., Sismoro, H. and Laksito, A.D. (2019) “Implementasi Algoritma Regresi Linear Berganda Untuk Memprediksi Produksi Padi Di Kabupaten Bantul,” *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, 4(2), pp. 66–75. doi:10.36341/rabit.v4i2.666.

Wahyudin, W. and Purwanto, H. (2021) “Prediksi Kasus Covid-19 Di Indonesia Menggunakan Metode Backpropagation Dan Regresi Linear,” *Journal of Information System, Applied, Management, Accounting and Research*, 5(2), p. 331. doi:10.52362/jisamar.v5i2.420.

Widiani, U.D. (2012) “Pembangunan Sistem Informasi Aset Di PT. Industri Telekomunikasi Indonesia (Persero) Berbasis Web,” *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 1(2), pp. 57–62. Available at: <http://komputa.if.unikom.ac.id/jurnal/pembangunan-sistem-informasi.m> (Accessed: June 14, 2022).

Yudo Ardianto Agung (2020) *Apa itu VsCode?* Available at: <https://rep.alphabetincubator.id/apa-itu-vscode/> (Accessed: June 16, 2022).

Biodata Penulis



Syafrial Fachri Pane, S.T., M.T.I., EBDP. Lahir di Medan 16 April 1988. Lulus D3 di Program Studi Teknik Informatika Politeknik Pos Indonesia tahun 2009, lulus S1 di Program Studi Teknik Informatika Universitas Pasundan tahun 2013, dan lulus S2 di Teknik Informatika Universitas Bina Nusantara. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Mengampu matakuliah Database I dan Database II. Aktif menulis di berbagai jurnal ilmiah. Bidang yang ditekuni adalah ilmu data. Saat ini sedang melanjutkan Pendidikan S3 di Telkom University.



Amri Yanuar, ST., M.OT. Lahir di Bandung, 12 Januari 1986. Lulus S1 di Program Studi Teknik Industri Universitas Pasundan tahun 2009, lulus S2 di Program Studi Management of Technology Universitas Teknologi Malaysia. Saat ini adalah dosen tetap Program Studi Logistik Bisnis di Universitas Logistik dan Bisnis Internasional. Bidang yang ditekuni adalah bidang logistik. Mengampu matakuliah Manajemen Persediaan, SAP Fundamental.



Nur Alamsyah, ST., M.KOM Lahir di Jakarta 09 Maret 1976. Lulus D3 di Program Studi Teknik Komputer STMIK Bandung, Lulus S1 Teknik Informatika STMIK Bandung tahun 2002 dan Lulus dari program studi S2 Sistem Informasi STMIK LIKMI pada tahun 2013. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Nasional Pasim mengampu matakuliah Sistem Informasi Manajemen, Pengantar Teknologi Internet dan Cyberpreneurship. Aktif menulis di berbagai

jurnal ilmiah. Bidang yang ditekuni saat ini adalah sistem informasi dan ilmu data. Saat ini sedang melanjutkan Studi S3 di Telkom University.



Bachtiar Ramadhan. Lahir di Pekanbaru, 21 Desember 2000. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang menekuni bidang ilmu data.



Nur Tri Ramadhanti Adiningrum. Lahir di Bandung, 06 Desember 2001. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang menekuni bidang ilmu data.

MEMBUAT APLIKASI WEB PREDIKSI GAJI SEMUDAH MENANAK NASI

Perkembangan ilmu pengetahuan dan teknologi pada Revolusi Industri 4.0 semakin berkembang pesat. Sehingga, terdapat penggabungan teknologi digital dan internet dengan industri konvensional, yang bertujuan untuk meningkatkan produktivitas, efisiensi dan layanan konsumen secara signifikan. Era revolusi ini akan mendisrupsi berbagai kegiatan diberbagai bidang seperti pada bidang teknologi, ekonomi, sosial, dan politik. Saat ini, kehidupan berada diawali revolusi yang secara mendasar mengubah cara hidup, bekerja, dan berhubungan satu sama lain. Perubahan karakteristik pekerjaan adalah salah satu dampak tersendiri dari datangnya revolusi industri 4.0. Karakteristik pekerjaan yang berubah akan mendisrupsi pekerjaan yang telah ada dan menggantikanya dengan pekerjaan dengan karakteristik baru. Karakteristik baru pada pekerjaan juga membutuhkan kompetensi baru kepada para pekerja. Tentunya perusahaan harus siap untuk saling bersaing dengan perusahaan yang lain. Selanjutnya, perusahaan perlu memiliki keunggulan dan manajemen yang efektif untuk menghadapi persaingan tersebut. Dengan demikian salah astu aspek yang berpengaruh besar terhadap kemajuan dan keberhasilan sebuah perusahaan adalah kinerja pegawaiya. Oleh karena itu, penentuan gaji yang tepat oleh perusahaan kepada pegawai adalah salah satu faktor yang berpengaruh secara internal terhadap kemajuan perusahaan. Sangat disayangkan, perkembangan perusahaan saat ini belum memiliki suatu media keputusan untuk melakukan prediksi gaji pegawai berdasarkan kualitas data. Karakteristik dataset yang digunakan untuk memprediksi gaji pegawai 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.

Adapun kandungan dari buku ini dimulai dari gambaran umum cakupan cara melakukan prediksi dan mengimplementasikan model tersebut ke dalam bentuk aplikasi web. Dalam setiap proses dilengkapi dengan penjelasan yang terperinci, beserta contoh kode program untuk memperjelas langkah-langkah, kinerja model *machine learning*, beserta kinerja aplikasi tersebut. Besar harapan dari penulis, pembaca diharapkan dapat mengerti cara membuat web prediksi dan dapat mengimplementasikannya ke dalam case di dunia nyata.



Syafrial Fachri Pane, S.T,M.T,LEBDP. Lahir di Medan 16 April 1988. Lulus D3 di Program Studi Teknik Informatika Politeknik Pos Indonesia tahun 2009, lulus S1 di Program Studi Teknik Informatika Universitas Pasundan tahun 2013, dan lulus S2 di Teknik Informatika Universitas Bina Nusantara. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Mengampu matakuliah Database I dan Database II. Aktif menulis di berbagai jurnal ilmiah. Bidang yang ditekuni adalah ilmu data. Saat ini sedang melanjutkan Pendidikan S3 di Telkom University.



Nur Alameyah, ST, MKOM Lahir di Jakarta 09 Maret 1978. Lulus D3 di Program Studi Teknik Komputer STMIK Bandung, Lulus S1 Teknik Informatika STMIK Bandung tahun 2002 dan Lulus dari program studi S2 Sistem Informasi STMIK LIKMI pada tahun 2013. Saat ini adalah dosen tetap Program Studi Teknik Informatika di Universitas Nasional Pasis mengampu matakuliah Sistem Informasi Manajemen, Pengantar Teknologi Internet dan Cyberpreneurship. Aktif menulis di berbagai jurnal ilmiah. Bidang yang ditekuni saat ini adalah sistem informasi dan ilmu data. Saat ini sedang melanjutkan Studi S3 di Telkom University.



Amri Yanuar, ST,MM. Lahir di Bandung, 12 Januari 1986. Lulus S1 di Program Studi Teknik Industri Universitas Pasundan tahun 2009, lulus S2 di Program Studi Management of Technology Universitas Teknologi Malaysia. Saat ini adalah dosen tetap Program Studi Logistik Bisnis di Universitas Logistik dan Bisnis Internasional. Bidang yang ditekuni adalah bidang logistik. Mengampu matakuliah Manajemen Perseleksi, SAP Fundamental.



Bachtiar Ramadhan. Lahir di Pekanbaru, 21 Desember 2000. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang meneleki bidang ilmu data.



Nur Tri Ramadhanti Adiningrum. Lahir di Bandung, 06 Desember 2001. Saat ini adalah mahasiswa Program Studi D4 Teknik Informatika di Universitas Logistik dan Bisnis Internasional. Kini sedang meneleki bidang ilmu data.

