# BAB I

# PENDAHULUAN

## **Latar Belakang**

Perkembangan ilmu pengetahuan dan teknologi pada Revolusi Industri 4.0 semakin berkembang pesat. Revolusi Industri 4.0 sendiri mulai terjadi melalui rekayasa intelegensia dan *internet of thing* sebagai tulang punggung pergerakan dan konektivitas antara manusia dengan mesin[1] (Prasetyo & Trisyanti, 2018). Sehingga, terdapat penggabungan teknologi digital dan internet dengan industri konvensional, yang bertujuan untuk meningkatkan produktivitas, efisiensi dan layanan konsumen secara signifikan[2] (Prasetyo & Sutopo, 2017). Era revolusi ini akan mendisrupsi berbagai kegiatan diberbagai bidang seperti pada bidang teknologi, ekonomi, sosial, dan politik[1] (Prasetyo & Trisyanti). Saat ini, kehidupan berada diawal revolusi yang secara mendasar mengubah cara hidup, bekerja, dan berhubungan satu sama lain. [3](Pangaribuan & Irwansyah, 2019).

Perubahan karakteristik pekerjaan adalah salah satu dampak tersendiri dari datangnya revolusi industry 4.0[4] (Shahroom & Hussin, 2018). Karakteristik pekerjaan yang berubah akan mendisrupsi pekerjaan yang telah ada dan menggantikanya dengan pekerjaan dengan karakteristik baru [5](Kergroach, 2017). Karakteristik baru pada pekerjaan juga membutuhkan kompetensi baru kepada para pekerja[6] (Manda & Dhaou, 2019; Sharma, 2019). Tentunya perusahaan harus siap untuk saling bersaing dengan perusahaan yang lain[7]. Selanjutnya, perusahaan perlu memiliki keunggulan dan manajemen yang efektif untuk menghadapi persaingan tersebut[7]. Dengan demikian salah astu aspek yang berpengaruh besar terhadap kemajuan dan keberhasilan sebuah perusahaan adalah kinerja karyawannya[7]. Walaupun perusahaan tersebut memiliki teknologi yang canggih, namun tidak terdapat tenaga kerja didalamnya, perusahaan tidak akan dapat mencapai tujuannya[7].

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

Karakteristik dataset yang digunakan untuk memprediksi gaji karyawan terdiri dari parameter-parameter berdasarkan faktor-faktor spesifik. Selanjutnya faktor-faktor tersebut akan di uji validitas dan korelasinya menggunakan mendekatan *machine learning*. Faktor-faktor tersebut akan diambil berdasarkan pedoman interpretasi koefisien korelasi yang dikemukakan oleh Sugiyono (2008 : 184) [8]. Untuk menentukan faktor yang dominan terhadap prediksi gaji, maka koefisien korelasi yang akan digunakan adalah tingkat hubungan sedang, kuat, dan sangat kuat. Metode yang digunakan pada *machine learning* yaitu *regression*. *Regression* digunakan untuk melakukan prediksi gaji karyawan. Tentunya hasil prediksi gaji karyawan perlu divisualisasikan secara *realtime* untuk dapat digunakan oleh perusahaan dalam menentukan keputusan dengan cepat. Visualisasi hasil prediksi tersebut akan ditampilkan berbasis web base dengan *framework* Django.

## **Identifikasi Masalah**

Berikut adalah identifikasi permasalahan dari penelitian ini :

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

## **Tujuan Penelitian**

Tujuan penilitian ini sebagai berikut :

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

## **Manfaat Penelitian**

Manfaat penilitian ini sebagai berikut :

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

## **Ruang Lingkup**

Ruang lingkup pada penelitian ini sebagai berikut :

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

## **Sistematika Penulisan**

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

1. BAB I PENDAHULUAN

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

1. BAB II LANDASAN TEORI

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

1. BAB III ANALISIS DAN PERANCANGAN

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

1. BAB IV IMPLEMENTASI

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

1. BAB V KESIMPULAN DAN SARAN

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

# BAB II

# LANDASAN TEORI

## **Tinjauan Studi**

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

### **Penelitian Terkait**

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

# BAB III

# ANALISIS DAN PERANCANGAN

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

## **Analisis Sistem**

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

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

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

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

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



Gambar 1 Flowmap Pemrosesan Gaji

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

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

#### **Flowmap Login**

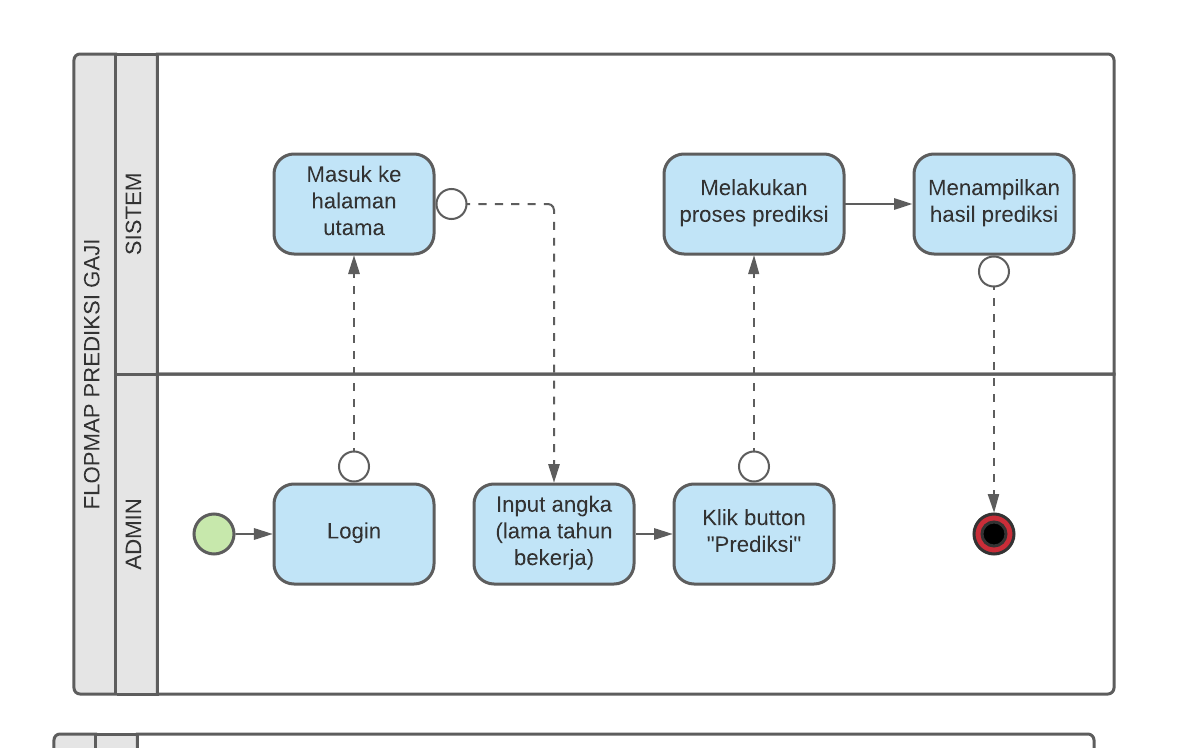
Pada bagian ini menjelaskan tentang flowmap proses *login* admin untuk masuk ke aplikasi website prediksi gaji. Flowmap tersebut merupakan gambaran alur proses *login* admin yang akan dibangun pada aplikasi ini.

Gambar 2 Flowmap Proses Login Admin

Keterangan :

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

#### **Flowmap Prediksi Gaji**

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

Gambar 3 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 berapa lamanya (dalam tahun) seorang pegawai telah bekerja, pada form yang disediakan
5. Setelah diinputkan, aplikasi akan menampilkan hasil prediksi gaji pegawai berdasarkan pengalaman lama bekerja

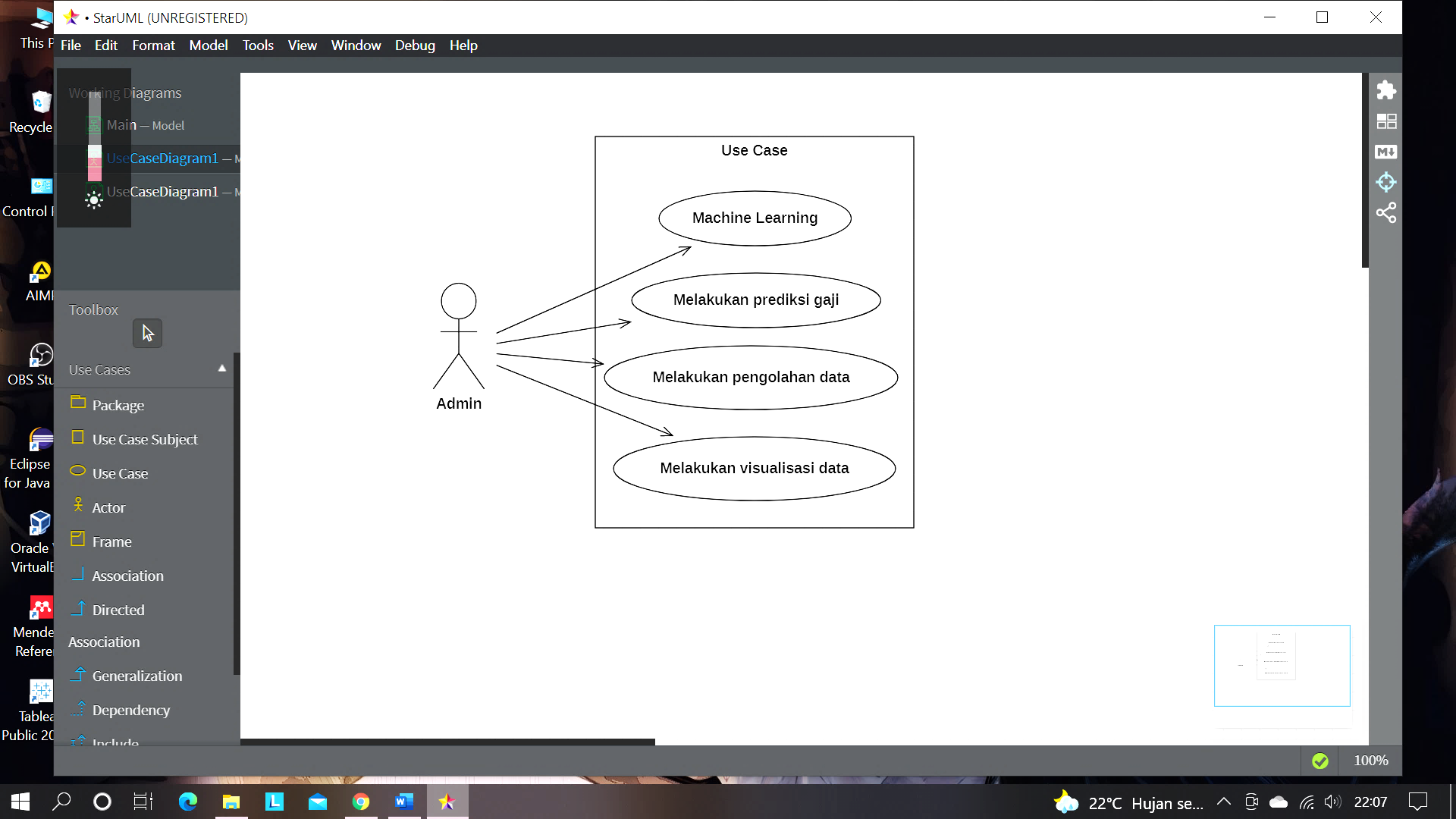
## **Perancangan Sistem (UML)**

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

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

### **Use Case Diagram**

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



Gambar 4 Use Case Diagram

1. **Definisi Aktor**

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

Tabel 1 Definisi Aktor

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

1. **Definisi Use Case**

Tabel 2 Definisi Use Case

|  |  |  |
| --- | --- | --- |
| **No** | **Aktor** | **Deskripsi** |
| 1 | Login | Merupakan aktivitas login yang dilakukan oleh admin. Sebelum admin masuk ke aplikasi ia harus melakukan login terlebih dahulu. |
| 2 | Melakukan Prediksi Gaji | Merupakan aktivitas memprediksi gaji karyawan yang dilakukan oleh admin dengan parameter lama bekerja seorang karyawan. |
| 3 | Melakukan Pengolahan Data | Merupakan aktivitas insert, read, update dan delete data karyawan yang dilakukan oleh admin. |
| 4 | Melakukan Visualisasi Data | Merupakan aktivitas visualisasi grafik data karyawan yang dilakukan oleh admin. |

1. **Skenario Use Case**

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

Tabel 3 Skenario Use case Login Admin

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

Tabel 4 Skenario Use case Melakukan Prediksi Gaji Karyawan

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

Tabel 5 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 6 Skenario Use case Melakukan Pengolahan Data

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

### **Class Diagram**

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

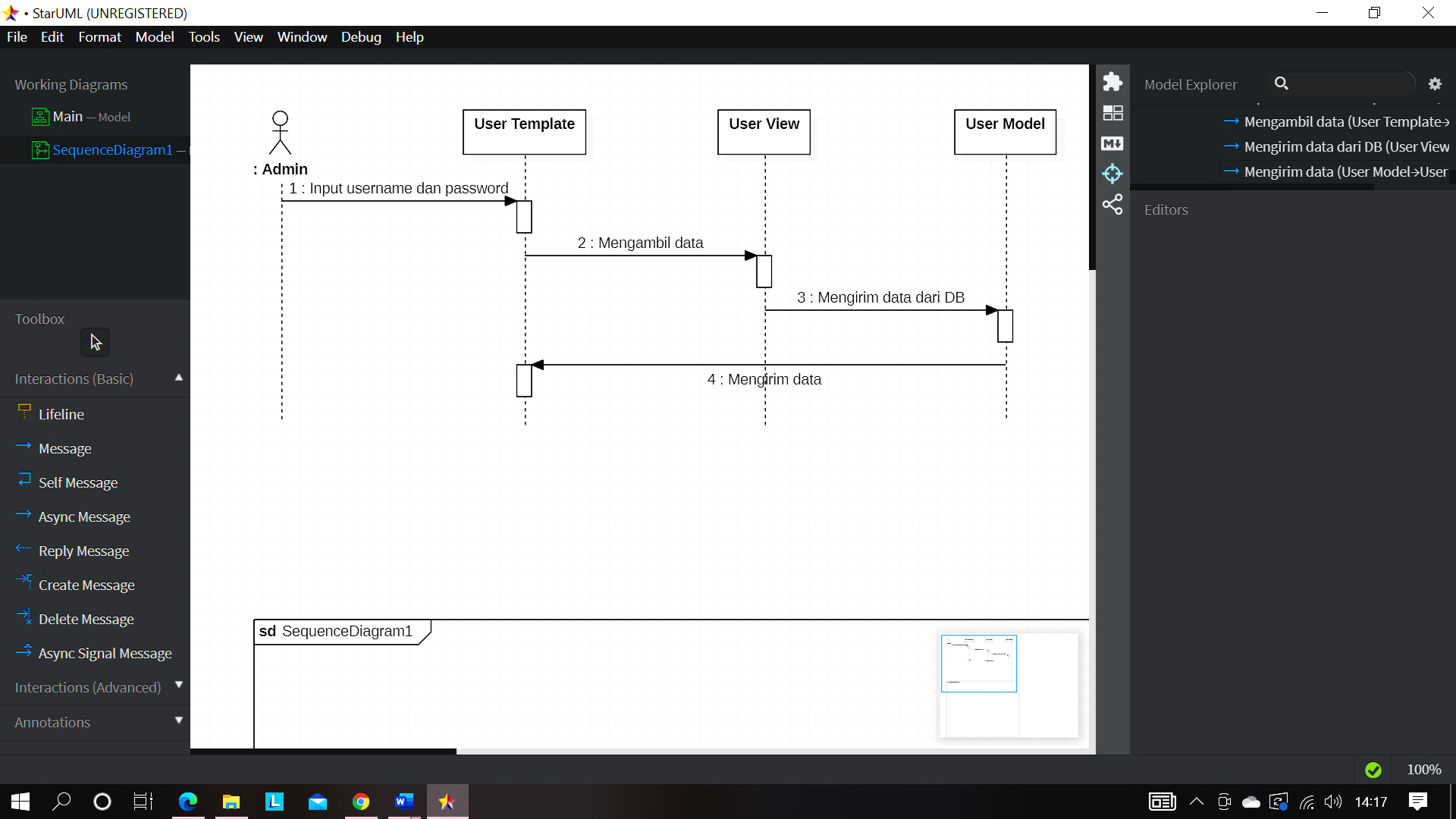
### **Sequence Diagram**

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

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

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

Berikut Sequence Diagram Login Admin pada gambar.



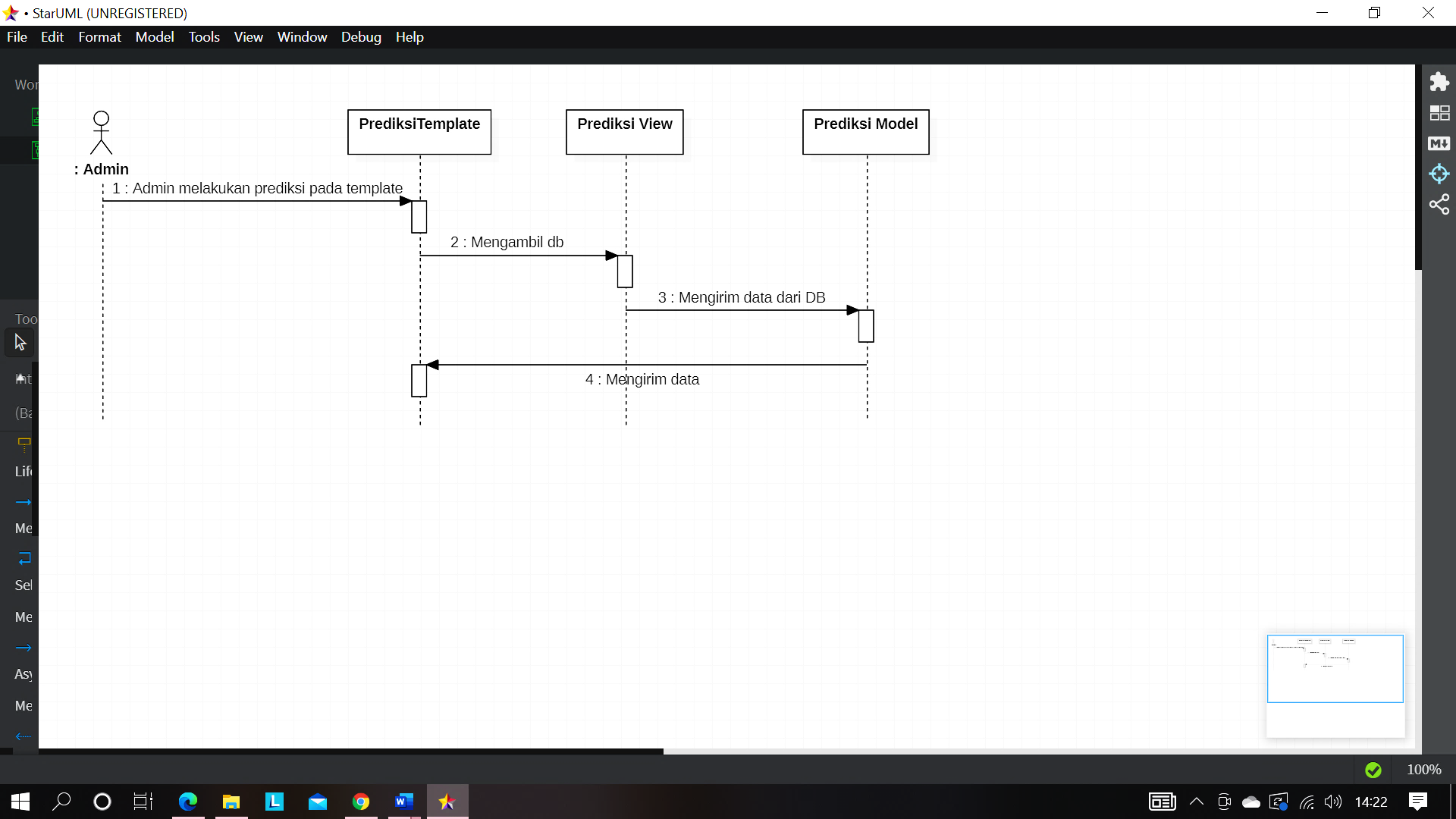
Gambar 5 Sequence Diagram Login Admin

Keterangan :

1. Admin menginputkan *username* dan *password*.
2. User template mengambil data pada user view.
3. User view mengirim data dari database ke user model.
4. User model mengirim data pada user template untuk ditampilkan pada aplikasi.
5. **Sequence Diagram Melakukan Prediksi Gaji**

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

Berikut Sequence Diagram Melakukan Prediksi Gaji pada gambar.



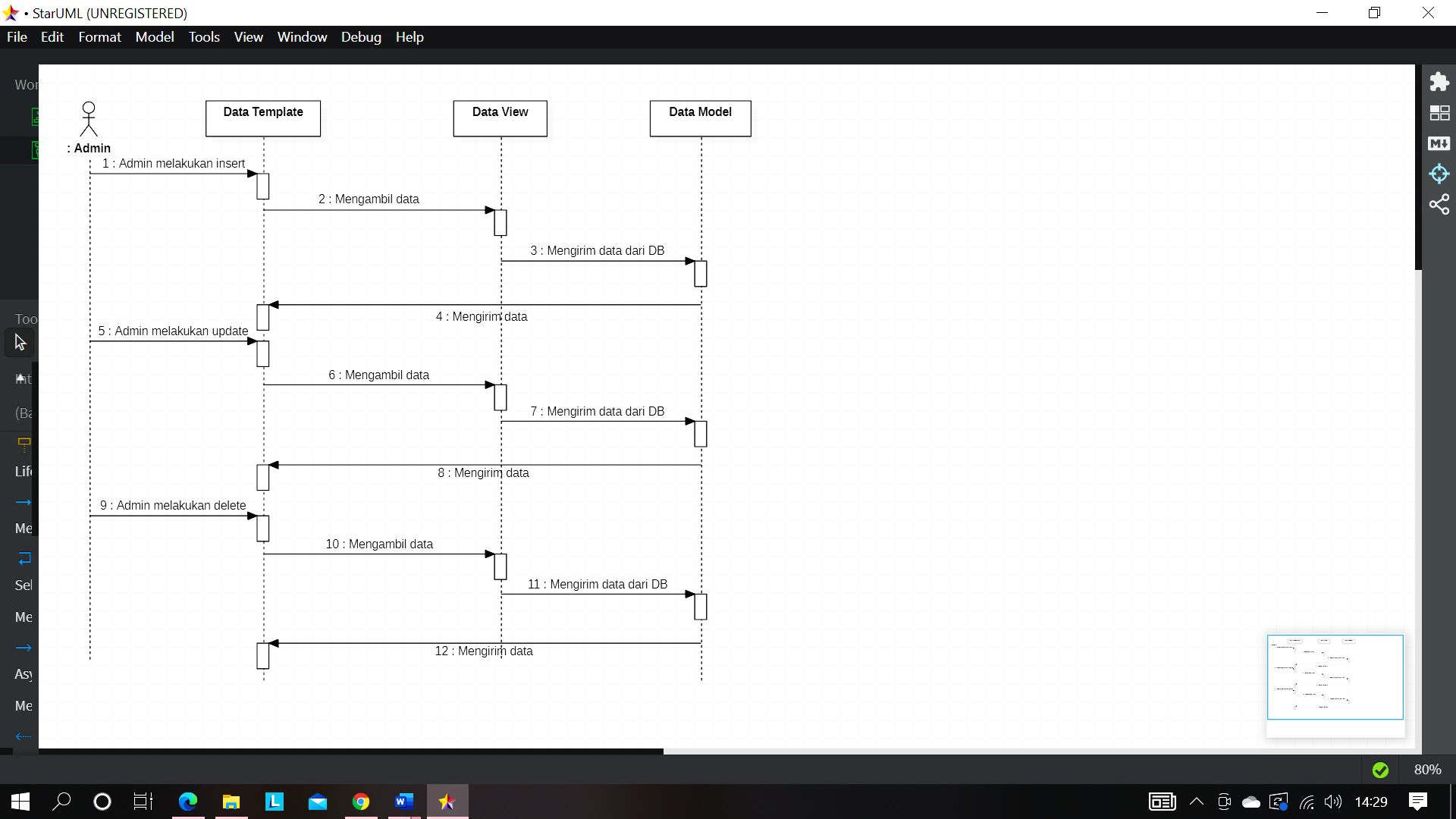
Gambar 6 Sequence Diagram Melakukan Prediksi Gaji

Keterangan :

1. Admin melakukan template prediksi.
2. Prediksi template mengambil data pada prediksi view.
3. Prediksi view mengirim data dari database ke prediksi model.
4. Prediksi model mengirim data pada prediksi template untuk ditampilkan pada aplikasi.
5. **Sequence Diagram Melakukan Pengolahan Data**

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

Berikut Sequence Diagram Melakukan Pengolahan Data pada gambar.



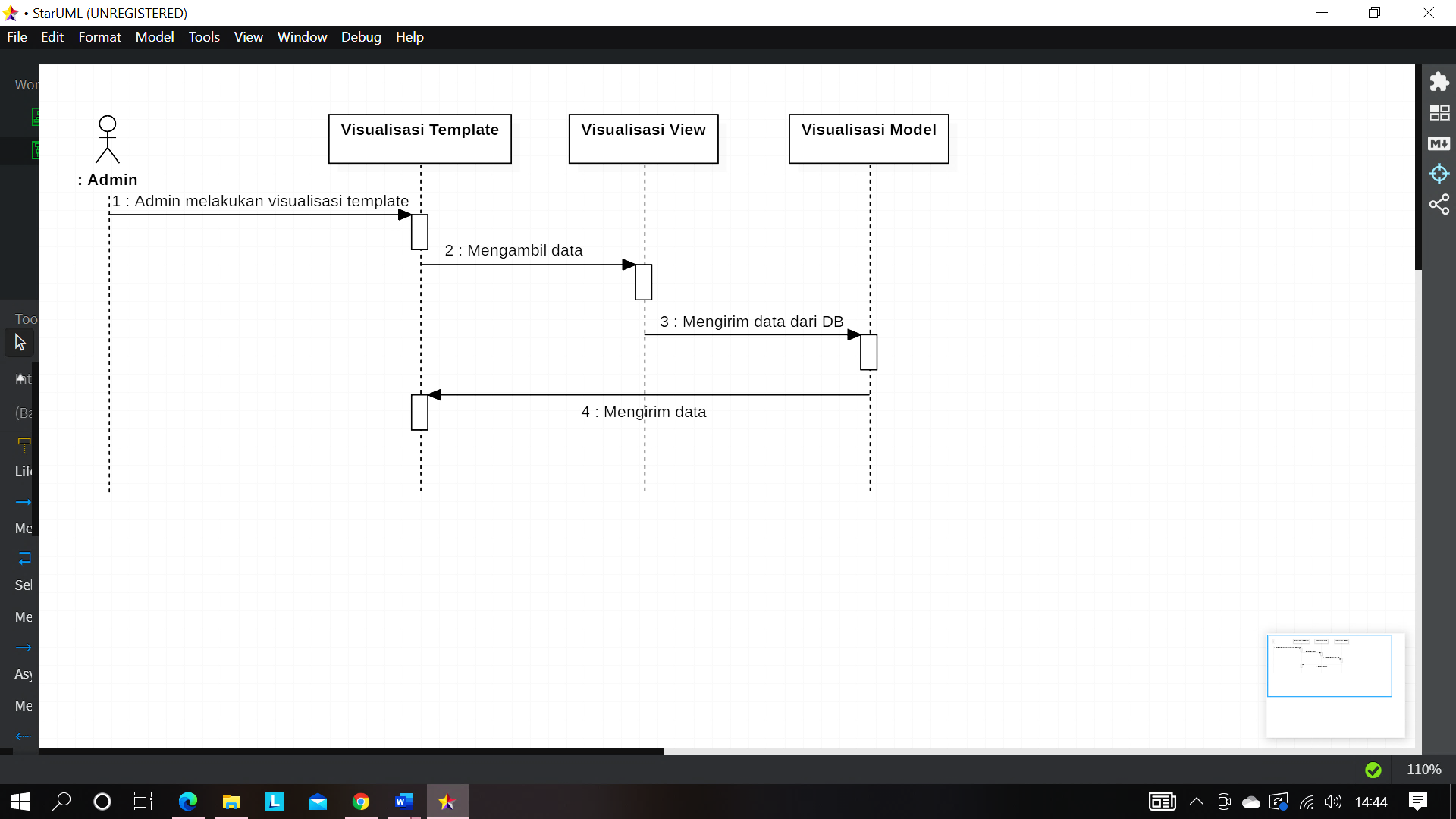
Gambar 7 Sequence Diagram Melakukan Pengolahan Data

Keterangan :

1. Admin memilih menu yang akan dikelola pada menu alat.
2. Data template mengambil data pada data view.
3. Data view mengirim data dari db ke data model.
4. Data model mengirim data pada template data untuk ditampilkan pada aplikasi.
5. **Sequence Diagram Melakukan Visualisasi Data**

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

Berikut Sequence Diagram Melakukan Visualisasi Data pada gambar.



Gambar 8 Sequence Diagram Melakukan Visualisasi Data

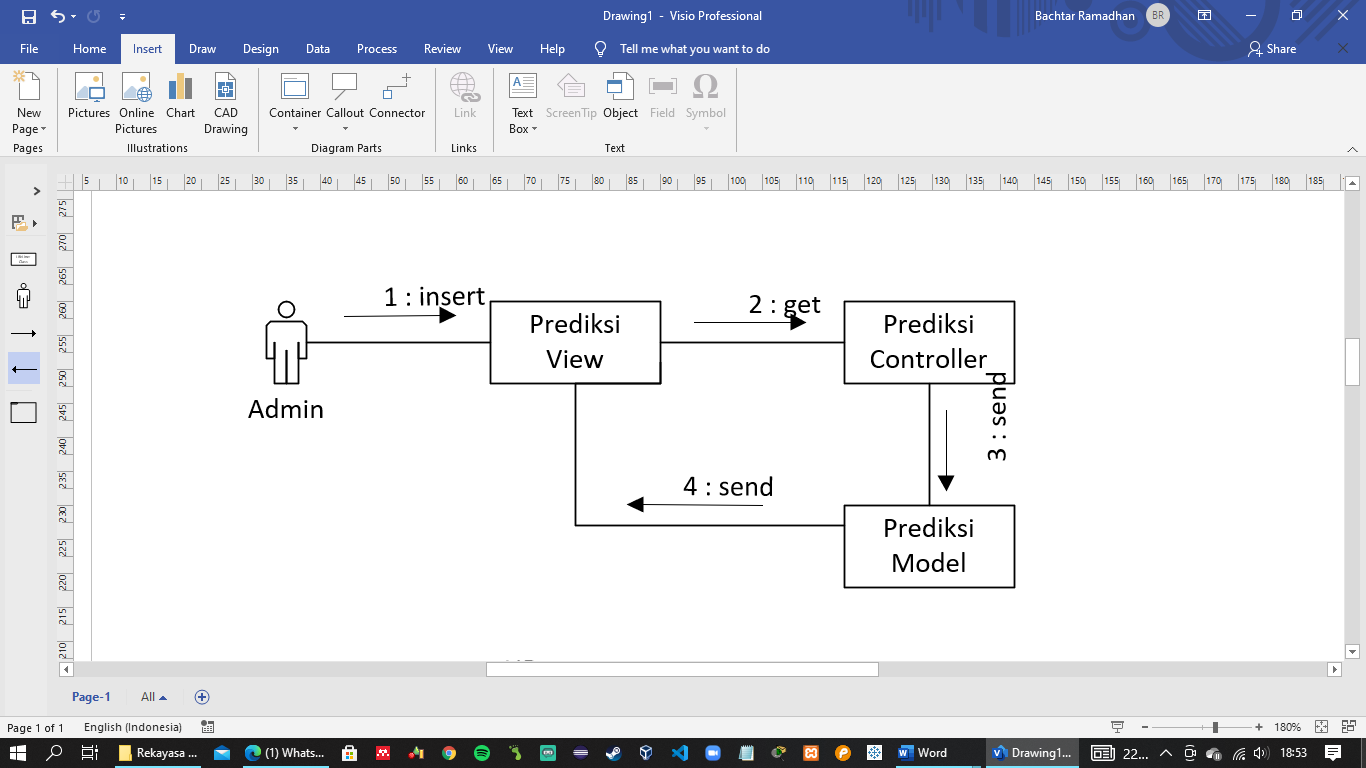
Keterangan :

1. Admin melakukan visualisai template.
2. Visualisasi template mengambil data pada visualisasi view.
3. Visualisasi view mengirim data dari database ke visualisasi model.
4. Visualisasi model mengirim data pada visualisasi template untuk ditampilkan pada aplikasi.

### **Collaboration Diagram**

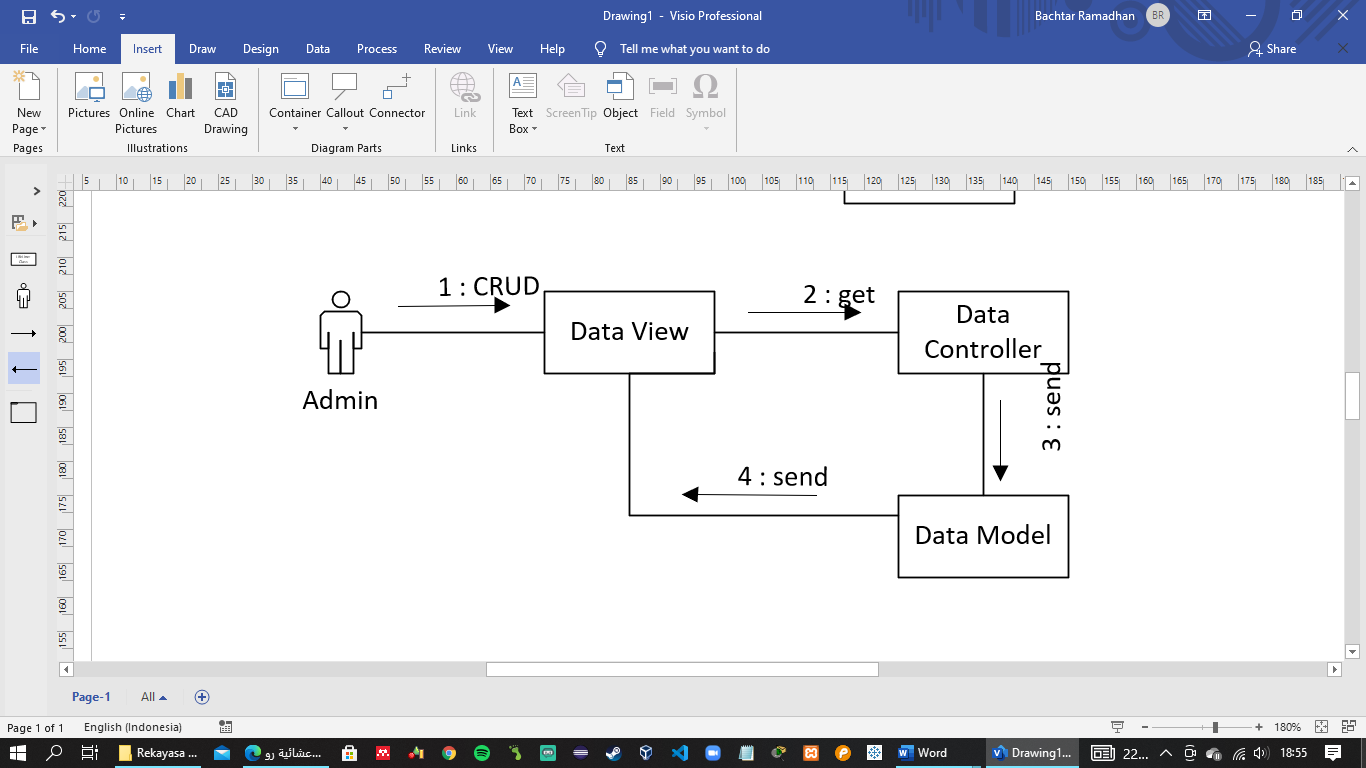
Collaboration diagram dapat dipakai untuk memodelkan interaksi antar objek yang ada di dalam sistem. Berbeda dari sequence diagram yang lebih menunjukkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram ini lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

1. **Collaboration Diagram Melakukan Prediksi Gaji**

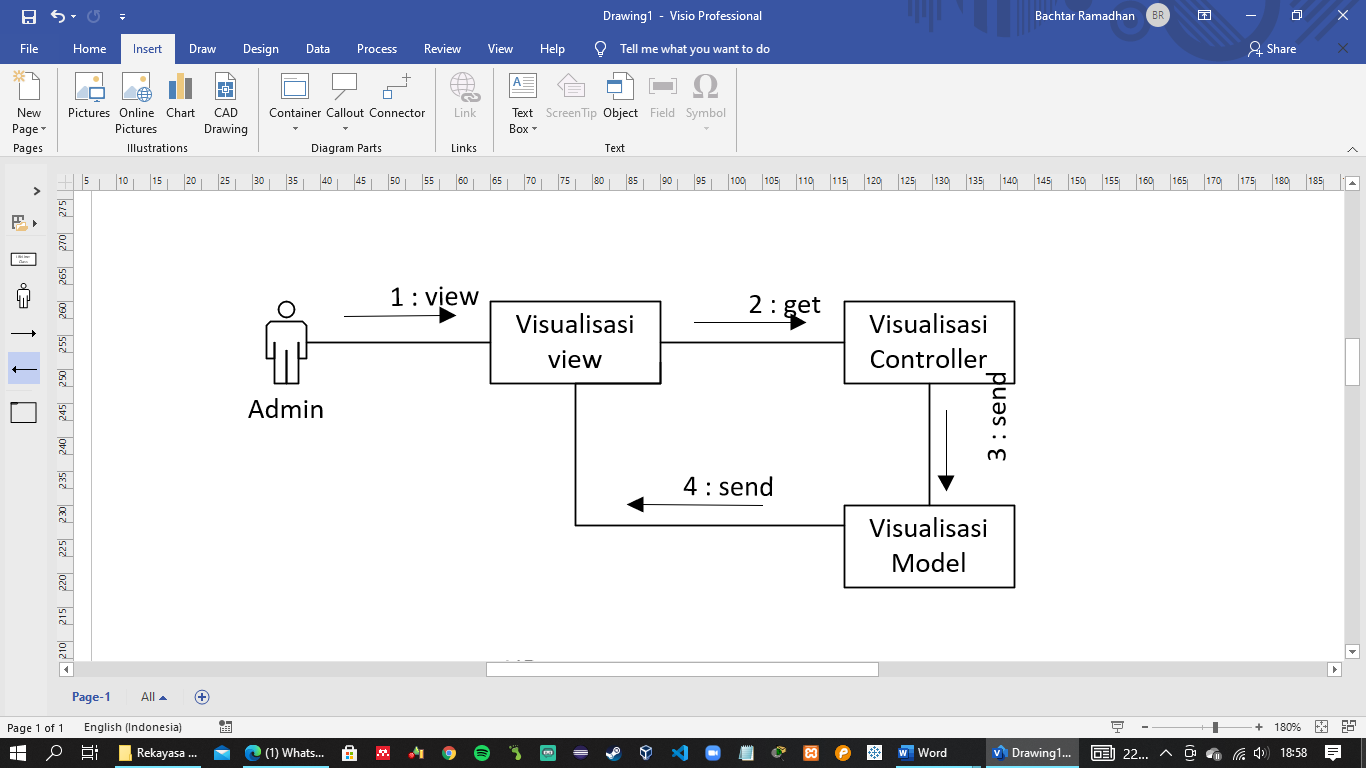


Gambar 9 Collaboration Diagram Melakukan Prediksi Gaji

1. **Collaboration Diagram Melakukan Pengolahan Data**



Gambar 10 Collaboration Diagram Melakukan Pengolahan Data

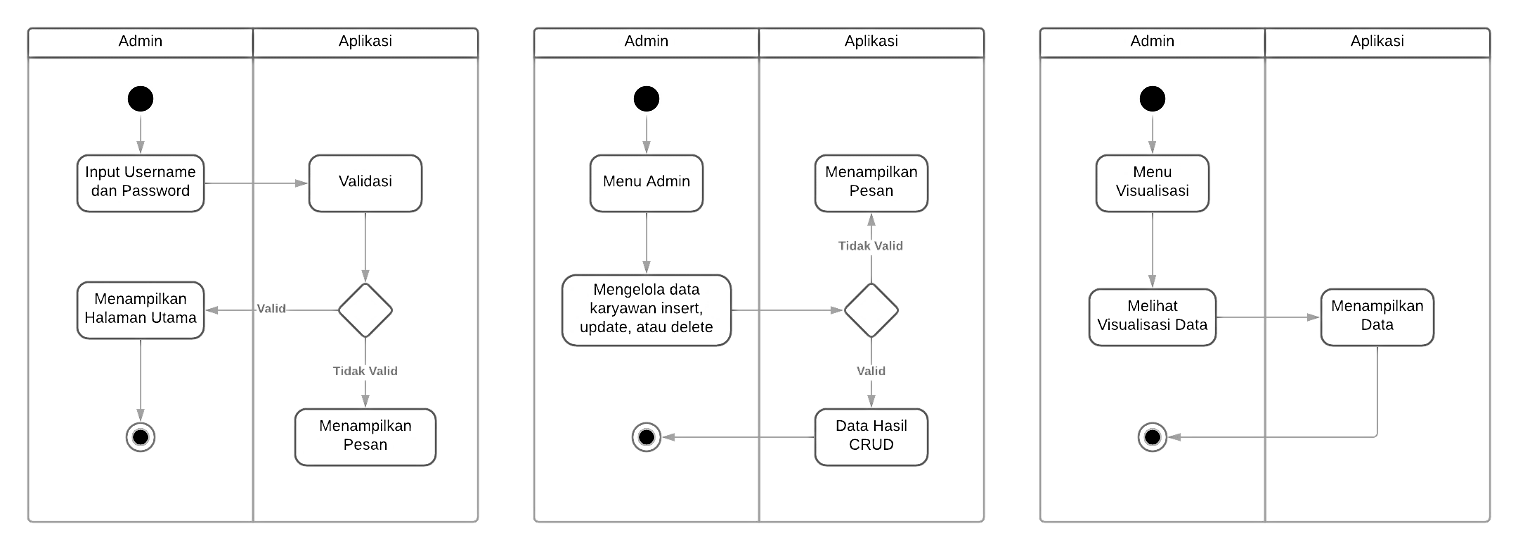
1. **Collaboration Diagram Melakukan Visualisasi Data**

Gambar 11 Collaboration Diagram Melakukan Visualisasi Data

### **Activiy Diagram**

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

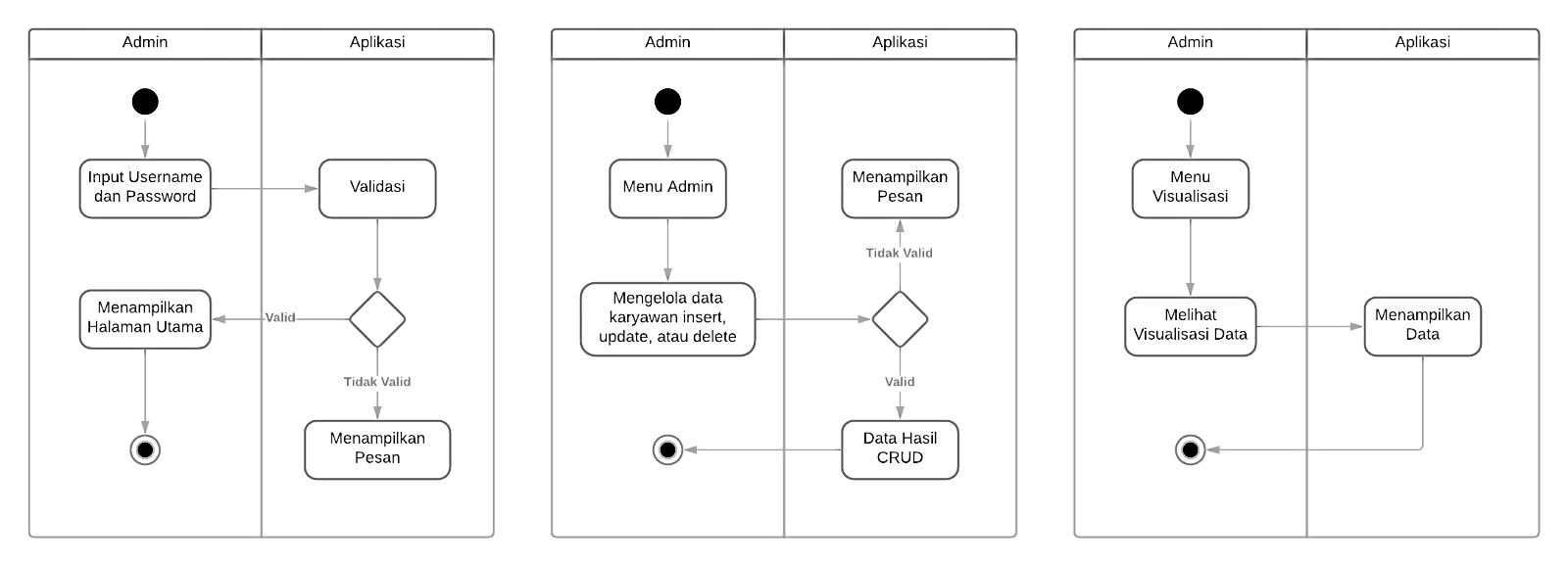
1. **Activity Diagram Login Admin**



Gambar 12 Activity Diagram Login Admin

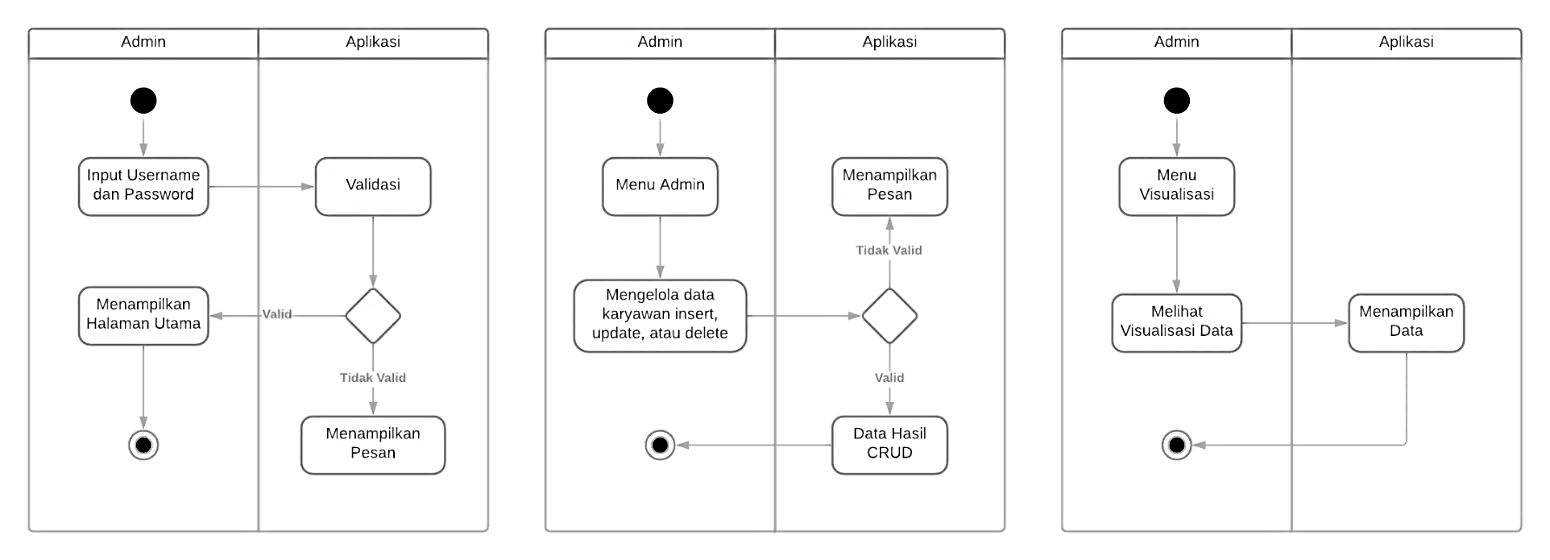
Keterangan :

1. Menginput username dan password.
2. Melakukan pengecekan data.
3. Jika valid akan menampilkan halaman utama.
4. Jika tidak valid menampilkan pesan.
5. **Activity Diagram Melakukan Pengolahan Data**



Gambar 13 Activity Diagram Melakukan Pengolahan Data

Keterangan :

1. Admin berada pada menu admin.
2. Admin akan melakukan creare/update/delete.
3. Melakukan pengecekan data.
4. Jika valid maka akan menampilkan data hasil create/update/delete.
5. **Activity Diagram Melakukan Prediksi Gaji**

Gambar 14 Activity Diagram Melakukan Prediksi Gaji

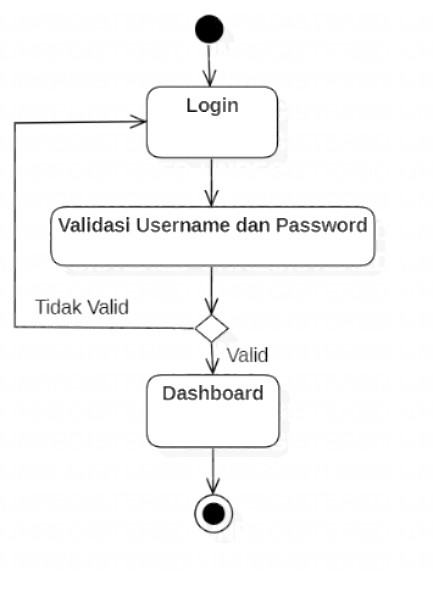
Keterangan :

1. Admin berada pada menu visualisasi.
2. Admin melihat visualisasi data.
3. Aplikasi menampilkan data.

### **Statechart Diagram**

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

1. **Statechart Diagram Login Admin**



Gambar 15 Statechart Diagram Login Admin

1. **Statechart Diagram Melakukan Prediksi**



Gambar 16 Statechart Diagram Melakukan Prediksi

1. **Statechart Diagram Pengolahan Data**



Gambar 17 Statechart Diagram Pengolahan Data

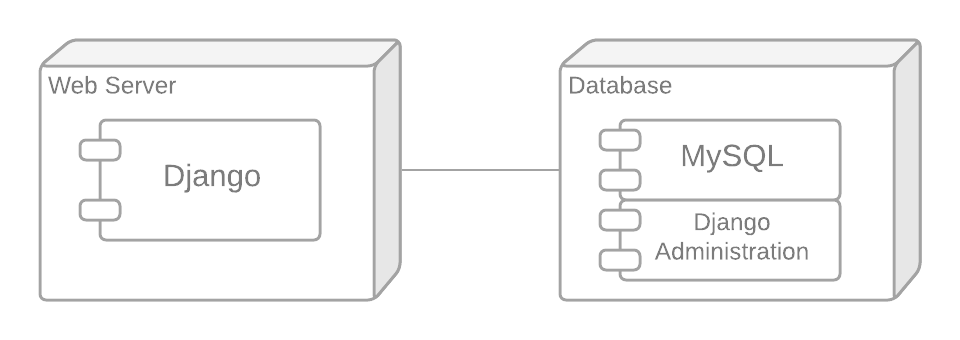
1. **Statechart Diagram Visualisasi**



Gambar 18 Statechart Diagram Visualisasi

### **Component Diagram**

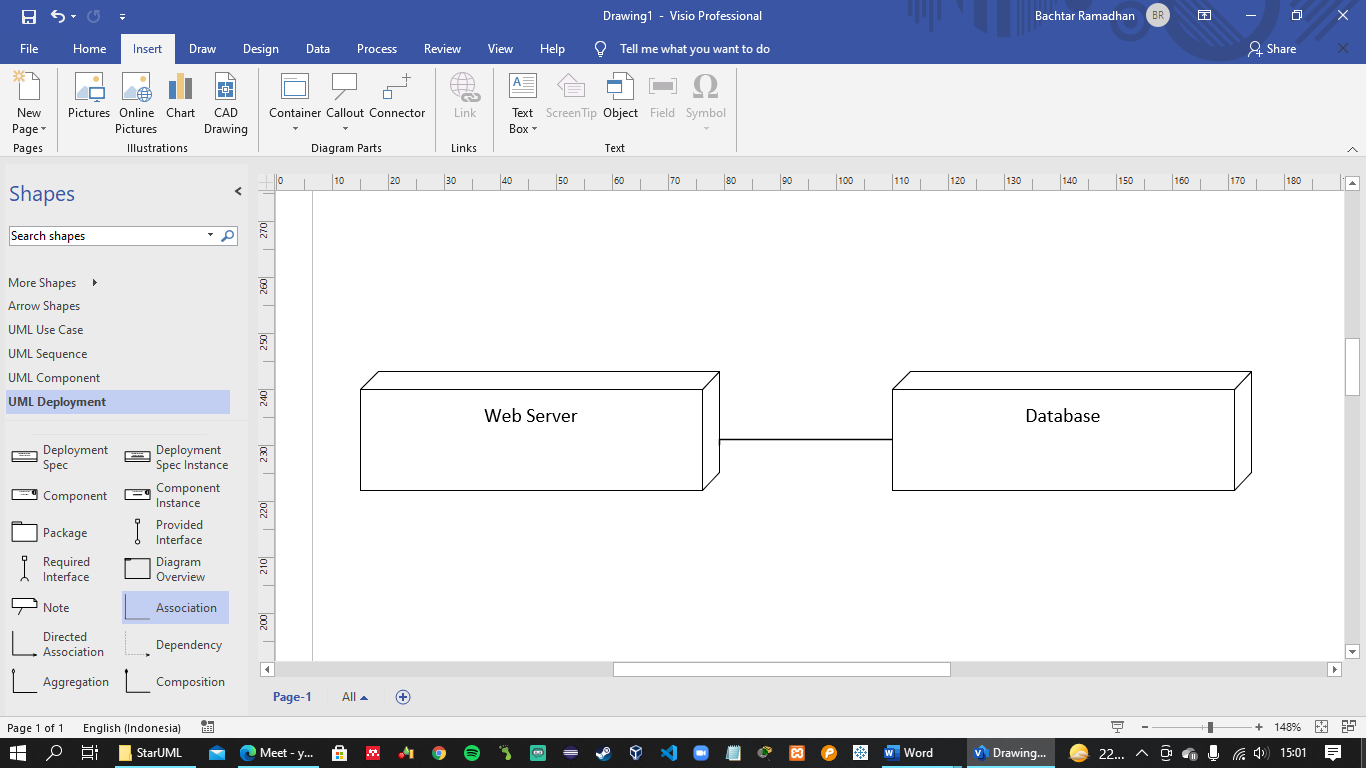
Component diagram dapat menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) diantaranya. Komponen piranti lunak atau yang biasa disebut dengan perangkat lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable,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



Gambar 19 Component Diagram

### **Deployment Diagram**

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



Gambar 20 Deployment Diagram

## **Perancangan Database**

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

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

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

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

## **Analisis Kebutuhan**

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

Analisis kebutuhan fungsional merupakan suatu kebutuhan yang berhubungan dengan berbagai kebutuhan sistem yang akan dirancang. Dimana kebutuhan ini menjabarkan mengenai fungsi-fungsi yang dapat mendukung jalannya sistem, adapun kebutuhan fungsional yang akan dibuat yaitu terdiri dari 2 (dua) proses sesuai dengan urutan sebagai berikut:

1. *Login* admin (user masuk ke aplikasi menggunakan username dan password);
2. Prediksi gaji pegawai yang dilakukan oleh Admin.

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

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

1. Perangkat Keras (Hardware) Yang Digunakan

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

1. Perangkat Lunak (Software) Yang Digunakan

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

1. Pengguna (User)

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

# BAB IV

# IMPLEMENTASI DAN PENGUJIAN

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

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

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

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

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

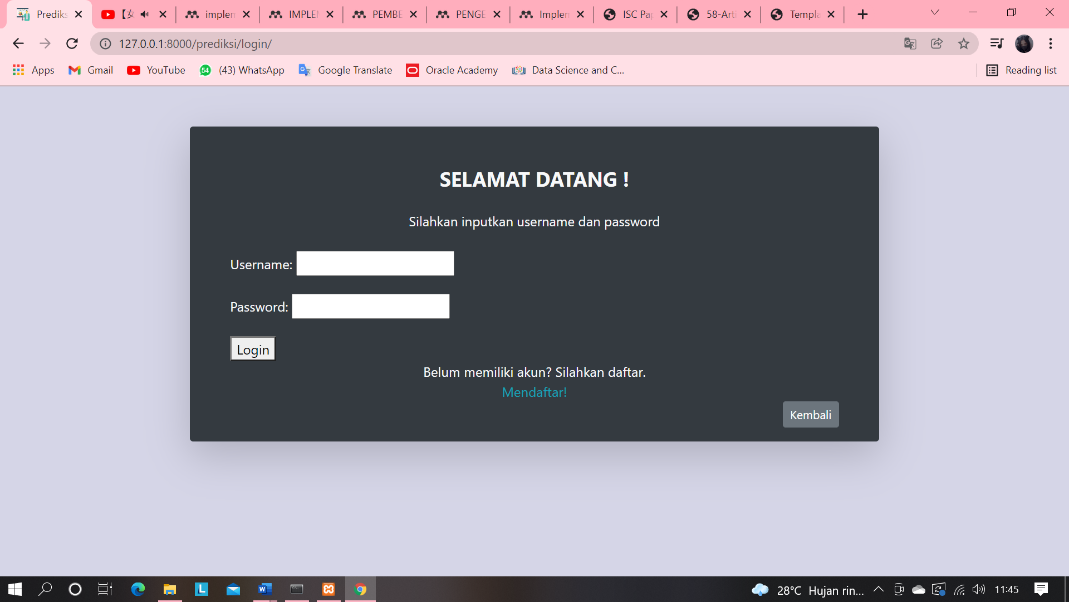
### **Implementasi Halaman Depan**



Gambar 21 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.

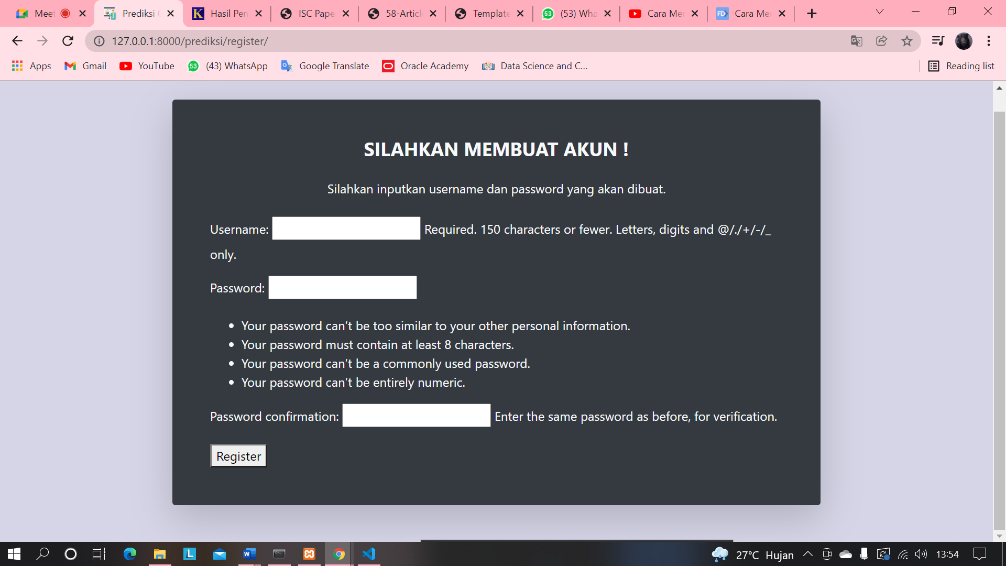
### **Implementasi Halaman *Login***



Gambar 22 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.

### **Implementasi Halaman Registrasi**



Gambar 23 Halaman Registrasi Sistem

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

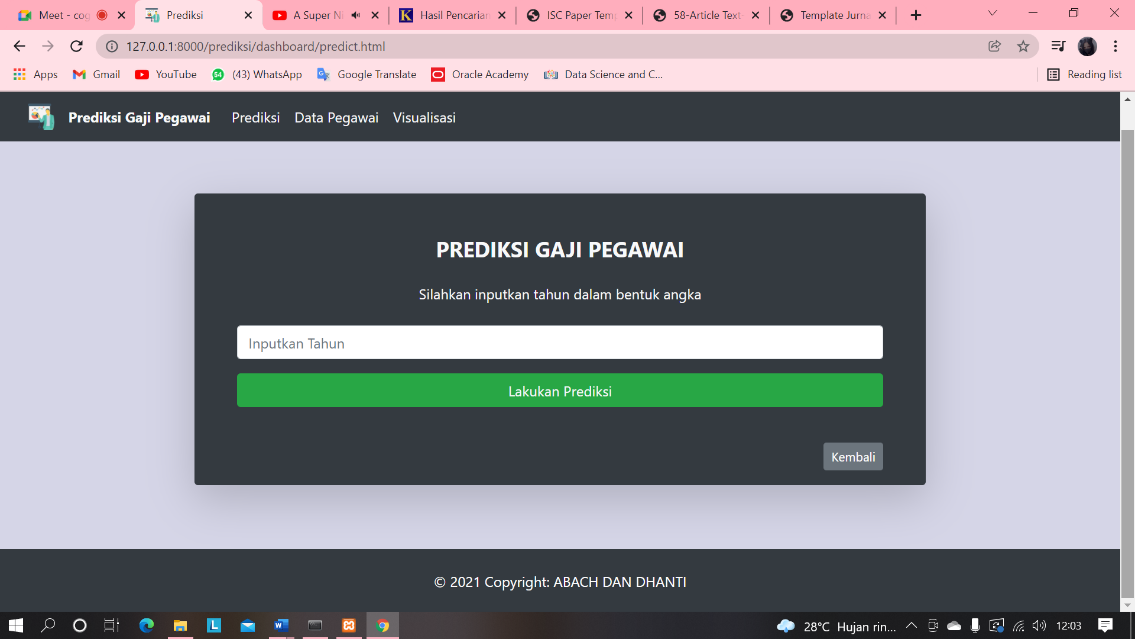
### **Implementasi Halaman *Dashboard***



Gambar 24 Halaman Dashboard Sistem

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

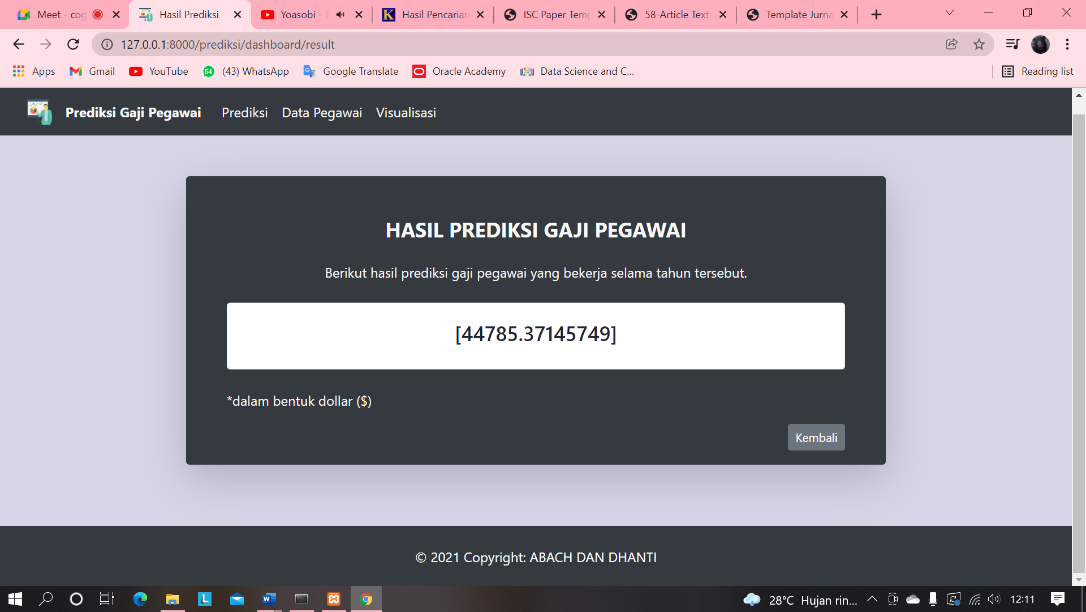
### **Implementasi Halaman Prediksi**



Gambar 25 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 ke dalam form untuk mengetahui hasil prediksi gaji pegawai. Kemudian *user* dapat menekan button “Lakukan Prediksi” untuk melihat hasil prediksi berdasarkan tahun yang diinputkan.

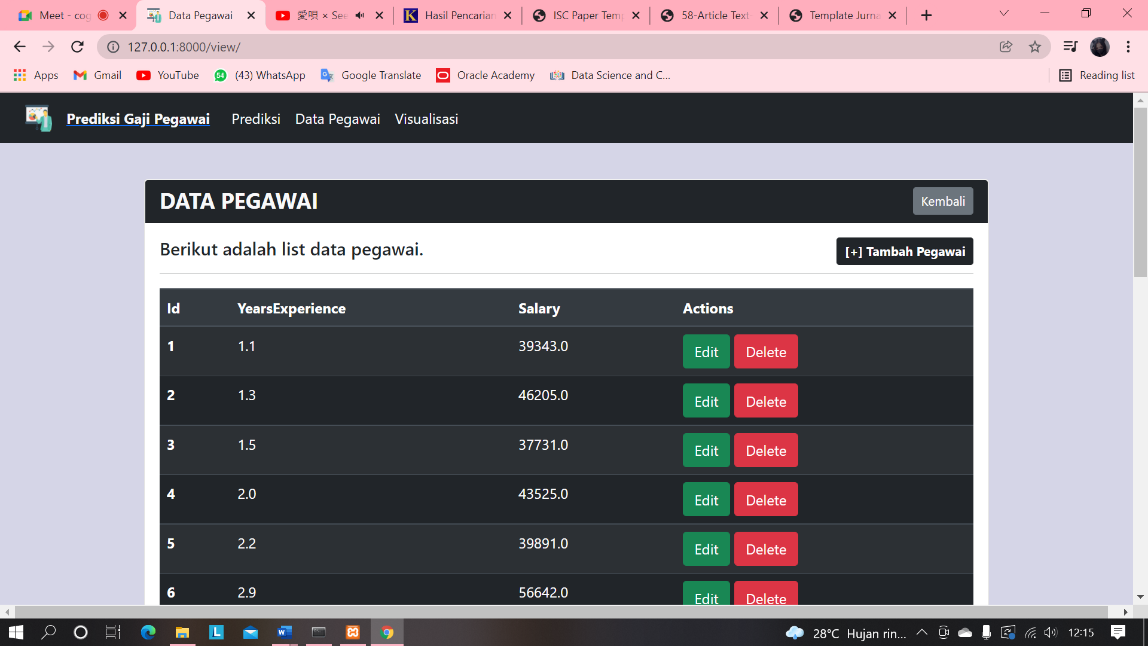
### **Implementasi Halaman Hasil Prediksi**



Gambar 26 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.

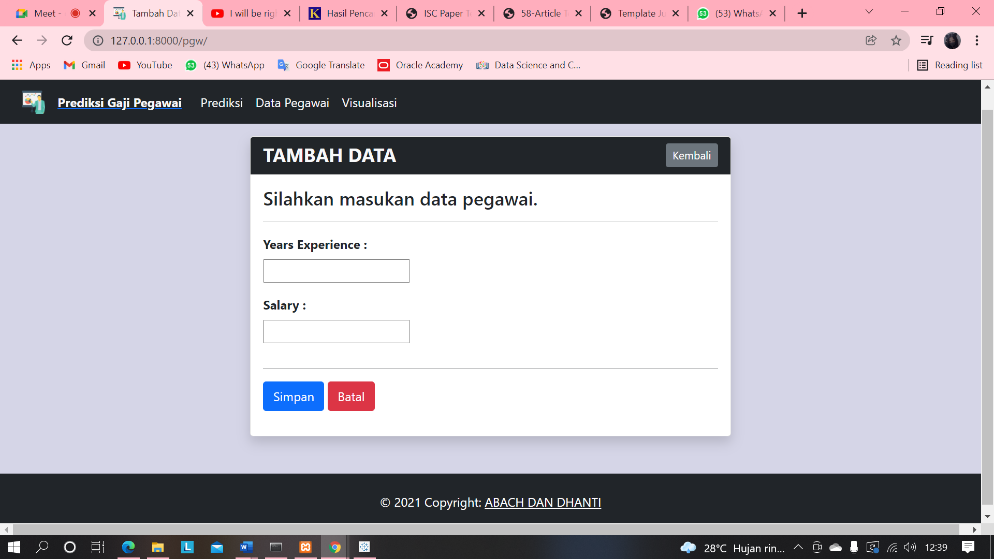
### **Implementasi Halaman Data Pegawai**



Gambar 27 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, YearsExperience, Salary, dan Action. *Record* yang ditampilkan terdiri dari 22 *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.

### **Implementasi Halaman Tambah Data Pegawai**



Gambar 28 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 Years Experience dan Salary 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.

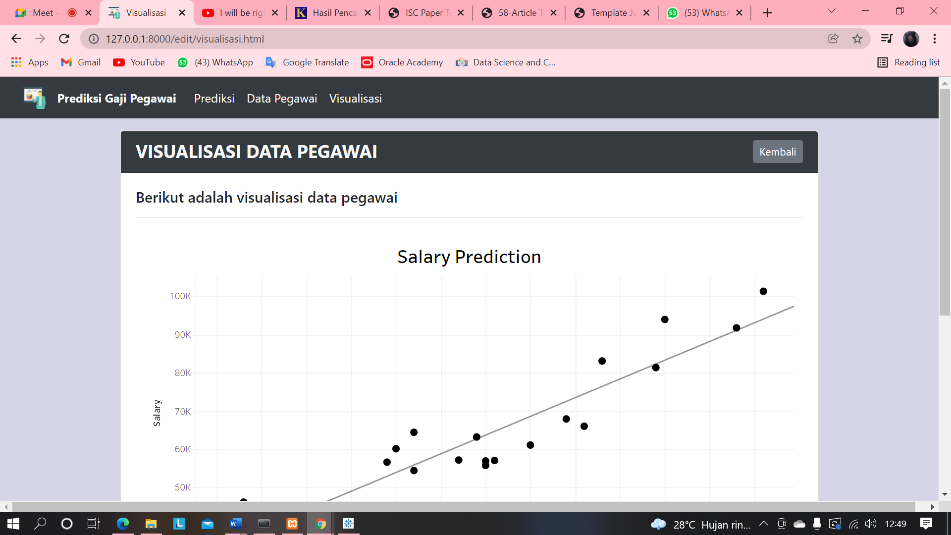
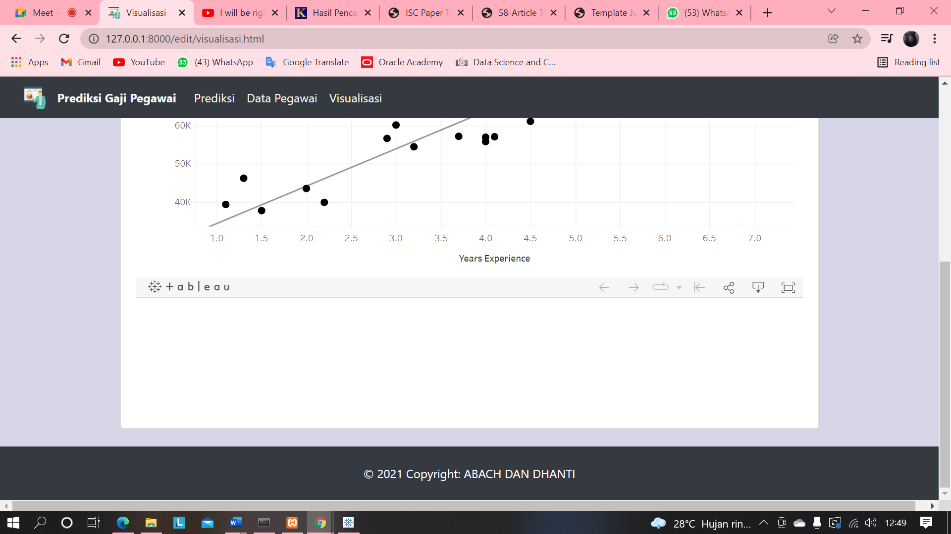
### **Implementasi Halaman Edit Data Pegawai**



Gambar 29 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 Years Experience dan Salary pada form. Lalu *user* dapat menekan *button* Simpan untuk menyimpan data yang telah diubah.

### **Implementasi Halaman Visualisasi**



Gambar 30 Halaman Visualisasi Sistem

Gambar diatas merupakan implementasi dari halaman visualisasi dari sistem yang dirancang. Pada halaman ini, akan ditampilkan bentuk visualisasi prediksi antara gaji (*salary*) dengan pengalaman lama bekerja (*Years Experience*).

## **Implementasi Aplikasi**

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

### **Implementasi Proses Login**

1. Views

@login\_required

def dashboardView(request):

    return render(request, 'dashboard.html')

1. Template

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

    <b>SELAMAT DATANG !</b>

    </h1>

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

    Silahkan inputkan username dan password

       </div>

       <br/>

       </div>

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

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

       <button>Login</button>

       <br/>

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

          Belum memiliki akun? Silahkan daftar.

       </div>

       <div align="center">

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

       </div>

       <div class="float-right">

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

       </div>

       </form>

1. URL

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

### **Implementasi Proses Registrasi**

1. Views

def registerView(request):

    if request.method == "POST":

        form = UserCreationForm(request.POST)

        if form.is\_valid():

            form.save()

            return redirect('login\_url')

    else:

        form = UserCreationForm()

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

1. Template

  <b>SILAHKAN MEMBUAT AKUN !</b>

       </h1>

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

    Silahkan inputkan username dan password yang akan dibuat.

    </div>

    <br/>

      </div>

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

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

      <button>Register</button>

      </form>

1. URL

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

### **Implementasi Proses Prediksi Gaji**

1. Views

def predict(request):

    context={'a':1}

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

1. Template

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

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

    <br>

     </div>

     {% block content %}

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

        {% csrf\_token %}

       <div class="form-group">

       <input type="float" name="tahun" id="tahun" class="form-control"  placeholder="Inputkan Tahun" autocomplete="off" required >

       </div>

       <div class="form-row">

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

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

        </div>

       </div>

         {% endblock %}

      </form>

         {{ result }}

1. URL

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

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

1. Views

def result(request):

    print(request)

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

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

    result = model.predict([[tahun]])

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

1. Template

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

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

     <br>

     </div>

     <div class="card text-center">

         <div class="card-body">

             <h4>{{ result }}</h4>

         </div>

     </div>

     <br>

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

1. URL

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

### **Implementasi Proses Menampilkan Data Pegawai**

1. Views

def view(request):

    pegawai = Pegawai.objects.all()

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

1. Template

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

    </div>

    <div class="float-right">

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

    </div>

    <br>

    <hr>

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

  <thead class="thead-dark">

    <tr>

      <th scope="col">Id</th>

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

      <th scope="col">Salary</th>

      <th scope="col">Actions</th>

    </tr>

  </thead>

  {% for pegawai in pegawai %}

    <tr>

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

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

      <td>{{ pegawai.Salary }}</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>

1. URL

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

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

1. Views

def pgw(request):

    if request.method == "POST":

        form = PegawaiForm(request.POST)

        if form.is\_valid():

            try:

                form.save()

                return redirect('/view')

            except:

                pass

    else:

        form = PegawaiForm()

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

1. Template

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

    <hr>

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

      {% csrf\_token %}

    <div class="form-row">

       <div class="form-group col-md-8">

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

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

       </div>

    </div>

    <div class="form-row">

        <div class="form-group col-md-8">

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

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

        </div>

    </div>

    <hr>

    <div class="form-group">

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

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

    </div>

    </form>

1. URL

 path('pgw/', views.pgw),

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

1. Views

def edit(request, id):

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

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

def update(request, id):

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

    form = PegawaiForm(instance=pegawai)

    if request.method == 'POST':

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

        if form.is\_valid():

            form.save()

            return redirect('/view')

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

1. Template

 <h4>Silahkan edit data pegawai.</h4>

   <hr>

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

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

      {% csrf\_token %}

   <div class="form-row">

       <div class="form-group col-md-8">

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

   <div>

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

             </div>

          </div>

      </div>

      <div class="form-row">

         <div class="form-group col-md-8">

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

          <div>

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

            </div>

         </div>

      </div>

      <hr>

      <div class="form-group">

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

      </div>

</form>

1. URL

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

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

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

1. Views

def delete(request, id):

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

    pegawai.delete()

    return redirect("/view")

1. URL

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

### **Implementasi Proses Menampilkan Visualisasi**

1. Views

def visualisasi(request):

    return render(request, 'visualisasi.html')

1. Template

<div class="float-left">

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

</div>

  <br>

  <hr>

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

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

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

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

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

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

    <param name='tabs' value='no' />

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

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

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

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

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

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

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

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

   </object>

</div>

  <script type='text/javascript'>

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

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

  if ( divElement.offsetWidth > 800 )

  {

    vizElement.style.minWidth='600px';

    vizElement.style.maxWidth='900px';

    vizElement.style.width='100%';

    vizElement.style.minHeight='127px';

    vizElement.style.maxHeight='527px';

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

    }

    else if ( divElement.offsetWidth > 500 )

    {

      vizElement.style.minWidth='600px';

      vizElement.style.maxWidth='900px';

      vizElement.style.width='100%';

      vizElement.style.minHeight='127px';

      vizElement.style.maxHeight='527px';

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

      }

      else

      {

        vizElement.style.width='100%';

        vizElement.style.height='727px';

        }

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

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

        vizElement.parentNode.insertBefore(scriptElement, vizElement);

        </script>

      </div>

1. URL

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

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

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

### **Himpunan Data**

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

|  |  |
| --- | --- |
| [1] | import pandas as pd |
| [2] | from sqlalchemy import create\_engine |
| [3] | import pymysql |
| [4] | #mysql+pymysql://<username>:<password>@<host>/<dbname>[?<options>]  sqlEngine=create\_engine('mysql+pymysql://root:@127.0.0.1',pool\_recycle=3600) |
| [5] | dbConnection = sqlEngine.connect() |
| [6] | df = pd.read\_sql('SELECT \* FROM djpegawai.prediksi\_pegawai ', dbConnection) |
| [7] | pd.set\_option('display.expand\_frame\_repr', False) |
| [8] | print(df) |
| [9] | dbConnection.close() |
| [10] | df\_clean = df.drop('id', axis=1) |
| [11] | x = df2.iloc[:, :-1] |
| [12] | y = df2.iloc[:, -1] |

Keterangan :

* Line 1 : Untuk mengimport library pandas kemudian diinisialisasi dengan varibel pd
* Line 2 : Dari sqlalchemy diimport metode create\_engine
* Line 3 : Untuk mengimport library pymysql.
* Line 4 : Membuat engine database MySQL kemudian disimpan dalam variabel sqlEngine.
* Line 5 : Membuat koneksi dengan metode sqlEngine agar terhubung dengan database, kemudian simpan dalam variabel dbConnection.
* Line 6 : Membaca data pada database djpegawai tabel prediksi\_pegawai dan dikoneksikan dengan dbConnections, kemudian disimpan pada variabel df.
* Line 7 : Representasi dataframe keseluruhan
* Line 8 : Menampilkan data pada variabel df.
* Line 9 :
* Line 10 : Menghapus variabel id kemudian disimpan pada variabel df\_clean.
* Line 11 : Untuk menampilkan data yang terdapat dalam variabel df2.
* Line 12 : Membuat variabel X yang di dalamnya column yang diambil itu column Salary.
* Line 13 : Membuat variabel Y yang di dalamnya column yang diambil itu column YearsExperience

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

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

|  |  |
| --- | --- |
| [13] | from sklearn.model\_selection import train\_test\_split |
| [14] | x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=1/3, random\_state=0) |
| [15] | #Modelling Machine Learning  from sklearn.linear\_model import LinearRegression |
| [16] | regressor = LinearRegression() |
| [17] | persamaan = regressor.fit(x\_train, y\_train) #training |
| [18] | print(regressor.coef\_) |
| [19] | print(regressor.intercept\_) |
| [20] | y\_pred = regressor.predict(x\_test) |

Keterangan :

* Line 13 : Dari scikit learn yang dimana menggunakan model selection untuk mengimport metode train\_test\_split.
* Line 14 : Membuat variabel x\_train, x\_test, y\_train, dan y\_test menggunakan metode train\_test\_split yang dimana variabel yang digunakan adalah x, y, dan untuk data testnya sebanyak 1/3 total data yang ada dengan pemilihan data secara acak.
* Line 15 : Dari scikit learn linear model diimport metode LinearRegression.
* Line 16 : Membuat variabel regressor yang isinya metode LinearRegression.
* Line 17 : Membuat variabel persamaan yang isinya terdapat method regressor.fit dengan parameternya x\_train dan y\_train.
* Line 18 : Mencari koefisien dari x menggunakan regressor, kemudian ditampilkan.
* Pada bagian Years Experience memiliki nilai koefisien 9087. Nilia tersebut memiliki arti bahwa tiap karyawan yang mengalami pertambahan satu tahun kerja, maka kenaikan gaji yang diterima sebesesar 9087.
* Line 19 : Mencari konstanta/intercept menggunakan regressor, kemudian ditampilkan.

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

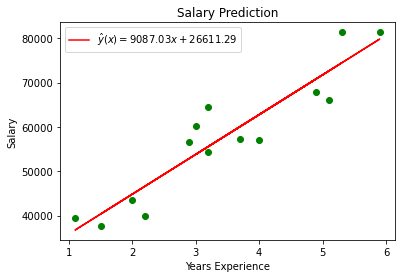
Berdasarkan nilai koefisien Years Experience dan Intersept didapat persamaan regresi linear berikut :

**Salary = 9087\*Years Experience + 26611**

* Line 20 : Menentukan variabel y\_pred yang didalamnya mencari prediksi mengggunakan variabel equation dengan parameternya x\_test.

1. Visualisasi Data Train

|  |  |
| --- | --- |
| [21] | import matplotlib.pyplot as plt |
| [22] | plt.scatter(x\_train, y\_train, color="green") |
| [23] | plt.plot(x\_train, regressor.predict(x\_train), color="red") |
| [24] | plt.title("Salary Prediction") |
| [25] | plt.xlabel("Years Experience") |
| [26] | plt.ylabel("Salary") |
| [27] | plt.legend(('$\hat{y}(x)=9087.03x+26611.29$',),loc='upper left') |
| [28] | plt.show() |



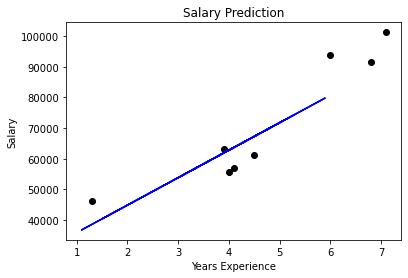
Gambar 31 Hasil Regresi Data Training

Keterangan :

* Line 21 : Mengimport library matplotlib.pyplot.
* Line 22 : Menampilkan titik/scatter x\_train dan y\_train berwarna hijau.
* Line 23 : Menampilkan plot/garis pada grafik berwarna merah.
* Line 24 : Membuat judul grafik dengan nama Salary vs Years Experience.
* Line 25 : Membuat label sumbu x dengan nama Years Experience.
* Line 26 : Membuat label sumbu y dengan nama Salary.
* Line 27 : Menampilkan fungsi linear y = 9087.03x+26611.29 pada grafik.
* Line 28 : Menampilkan visualisasi grafik.
* Line 29 : Dari scikit learn metrics diimport metode r2\_score.
* Line 30 : Menghitung r2\_score dengan parameternya x\_train dan y\_train, kemudian hasilnya disimpan pada variable r2.
* Line 31 : Menampilkan nilai pada variabel r2.

1. Visualisasi Data Test

|  |  |
| --- | --- |
| [29] | plt.scatter(x\_test, y\_test, color="black") |
| [30] | plt.plot(x\_train, regressor.predict(x\_train), color="blue") |
| [31] | plt.title("Salary Prediction") |
| [32] | plt.xlabel("Years Experience") |
| [33] | plt.ylabel("Salary") |
| [34] | plt.show() |



Gambar 32 Hasil Regresi Data Test

Keterangan :

* Line 29 : Menampilkan titik/scatter x\_test dan y\_test berwarna hitam.
* Line 30 : Menampilkan plot/garis pada grafik berwarna biru.
* Line 31 : Membuat judul grafik dengan nama Salary Prediction.
* Line 32 : Membuat label sumbu x dengan nama Years Experience.
* Line 33 : Membuat label sumbu y dengan nama Salary.
* Line 34 : Menampilkan visualisasi grafik.

### **Evaluasi Data**

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

1. R-Square

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

|  |  |
| --- | --- |
| [35] | from sklearn.metrics import r2\_score |
| [36] | r2 = r2\_score (x\_train, y\_train) |
| [37] | print ("Nilai R2 adalah ", r2) |

Keterangan :

* Line 35 : Dari scikit learn metrics diimport metode r2\_score.
* Line 36 : Menghitung r2\_score dengan parameternya x\_train dan y\_train, kemudian hasilnya disimpan pada variable r2.
* Line 37 : Menampilkan nilai pada variabel r2.

1. Tabel ANOVA

|  |  |
| --- | --- |
| [38] | import numpy as np |
| [39] | import pandas as pd |
| [40] | import scipy |
| [41] | from scipy import stats |
| [42] | def ANOVATAB(y,yhat,n,m):  dfn=n  dfd=m-n-1  ybar=np.average(y)    SSR = sum((yhat-ybar)\*\*2)  SSE = sum((y-yhat)\*\*2)  print (SSE)  print(dfd)  SST = sum((y-ybar)\*\*2)  MSR = SSR/dfn  MSE = SSE/dfd    Fs = MSR/MSE  ks =1-scipy.stats.f.cdf(Fs,dfn,dfd)  data\_table = {  'SS' : [SSR,SSE,SST],  'df' : [dfn,dfd,m-1],  'MS' : [MSR,MSE,'-'],  'Fs' : [Fs,'-','-'],  'pval' : [ks, '-', '-']  }  return pd.DataFrame(data\_table) |
| [43] | n=1  m=len(y\_pred)  print('TABEL ANOVA')  result=ANOVATAB(y\_test,y\_pred,n,m)  result |

Keterangan :

* Line 38 : Untuk mengimport libraty numpy kemudian diinisialisasi dengan variabel np.
* Line 39 : Untuk mengimpoty library pandas kemudian diinisialisasi dengan variabel pd.
* Line 40 : Untuk mengimport library scipy.
* Line 41 : Dari Scipy diimport metode stats.
* Line 42 : Untuk membuat rumus perhitungan pada tabel ANOVA.

Tabel ANOVA (Analysis of Variance) adalah tabel untuk menguji penerimaan (acceptability) model dari perspektif statistic dalam bentuk analisis sumber keragaman. ANOVA ini sering juga diterjemahkan sebagai analisis ragam dan output dari ANOVA adalam penjelasan uji regresi. Pada tabel tersebut tertera nilai dari SS, df, MS, FS, dan Pval.

1. SS (Sum of Square)

Kolom SS diperoleh dari penjumlahan kuadrat dari prediksi variabel terikat (permintaan) dikurangi dengan nilai rata-rata permintaan dari dara sebenarnya. Jadi secara manual kit acari dulu rata-rata permintaan dari data asli kita. Kemudian masing-masing prediksi permintaan (lihat pada tabel residual output) dikurangi dengan rata-rata tersebut kemudian dikuadratkan. Selanjutnya, seluruh hasil perhitungan tersebut dijumlahkan.

1. Df (Degree of Freedom)

Df atau derata bebas dari total adalah n-1, dimana n adalah banyaknya observasi.

1. MS (Mean of Square)

MS atau rata-rata jumlah kuadrat adalah hasil bagi antara kolom SS dengan kolom df. Dari perhitungan MS ini, selanjutnya bagi MS Regresi dengan MS Residual yang kemudian akan mendapatkan nilai F.

1. P-Value

P-Value adalah nilai probabilitas yang dapat diartikan sebagai besarnya peluang (probabilitas) yang diamati dari statistic uji. Variabel independent (Years Experience) mempunyai nilai p-value dibawah 0.05, hal ini menandakan ia signifikan melakukan prediksi variabel dependen (Salary).

1. Fs

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

* Line 43 : Untuk menampilkan hasil perhitungan dari tabel ANOVA.

Berdasarkan tabel diatas, dapat diketahui bahwa Fs > P-value, yang artinya variabel Salary dan Years Experience berpengaruh signifikan terhadap permintaan.

1. Prediksi Gaji

|  |  |
| --- | --- |
| [44] | salary\_pred = regressor.predict([[7]]) |
| [45] | print("Gaji yang terprediksi pada pegawai yang bekerja sepanjang tahun tersebut adalah ",salary\_pred) |

Keterangan :

* Line 44 : Menentukan prediksi gaji dengan parameter YearsExperience nya 7 tahun menggunakan metode pada variabel regressor, kemudian simpan di variabel salary\_pred.
* Line 45 : Menampilkan nilai dari variabel salary\_pred.

# DAFTAR PUSTAKA

[1] D. Tantangan, P. Sosial Banuprasetyo, and D. Trisyanti, “Prosiding SEMATEKSOS 3 ‘Strategi Pembangunan Nasional MenghadapiRevolusiIndustri 4.0’ REVOLUSI INDUSTRI 4.0.”

[2] H. Prasetyo and W. Sutopo, “Perkembangan Keilmuan Teknik Industri Menuju Era Industri 4.0,” 2017.

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

[4] A. A. Shahroom and N. Hussin, “Industrial Revolution 4.0 and Education,” *International Journal of Academic Research in Business and Social Sciences*, vol. 8, no. 9, Oct. 2018, doi: 10.6007/ijarbss/v8-i9/4593.

[5] S. Kergroach, “Industry 4.0: New challenges and opportunities for the labour market,” *Foresight and STI Governance*, vol. 11, no. 4, pp. 6–8, 2017, doi: 10.17323/2500-2597.2017.4.6.8.

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

[7] Y. Adrianova Eka Tuah, P. Studi Pendidikan Komputer, and S. Persada Khatulistiwa Sintang, “IMPLEMENTASI MODEL REGRESI LINEAR SEDERHANA UNTUK PREDIKSI GAJI BERDASARKAN PENGALAMAN LAMA BEKERJA,” 2020.

[8] “AGUNG SAPUTRA TAMRIN PATAR RUMAPEA RULLY MAMBO.”

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

[10] K. Puteri and A. Silvanie, “MACHINE LEARNING UNTUK MODEL PREDIKSI HARGA SEMBAKO DENGAN METODE REGRESI LINIER BERGANDA 1),” 2020. [Online]. Available: www.data.jakarta.go.id.

[11] A. Saiful, S. Andryana, and A. Gunaryati, “10 th,” 2012, [Online]. Available: http://jurnal.mdp.ac.id

[12] M. W. Pertiwi and R. E. Indrajit, *Metode Regresi Linier Untuk Prediksi Pengadaan Inventaris Barang*.

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

[14] N. Nafi’iyah, “Perbandingan Regresi Linear, Backpropagation Dan Fuzzy Mamdani Dalam Prediksi Harga Emas,” p. 2016.

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

[16] A. Dosen STIKOM Uyelindo, J. Perintis Kemerdekaan -Kayu Putih, P. Katemba, and R. Koro Djoh, “PREDIKSI TINGKAT PRODUKSI KOPI MENGGUNAKAN REGRESI LINEAR.”

[17] T. N. Putri, A. Yordan, and D. H. Lamkaruna, “Peramalan Penerimaan Mahasiswa Baru Universitas Samudra Menggunakan Metode Regresi Linear Sederhana,” 2019. [Online]. Available: https://data.unsam.ac.id/?op=pmb,

[18] “SALARY PREDICTION USING REGRESSION TECHNIQUES.”

[19] U. Bansal, A. Narang, A. Sachdeva, I. Kashyap, and S. P. Panda, “Empirical analysis of regression techniques by house price and salary prediction,” in *IOP Conference Series: Materials Science and Engineering*, Jan. 2021, vol. 1022, no. 1. doi: 10.1088/1757-899X/1022/1/012110.

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

[21] S. Dan and B. Pratikno, “REGRESI LINEAR BIVARIAT SIMPEL DAN APLIKASINYA PADA DATA CUACA DI CILACAP.”

[22] J. P. Matematika, D. Matematika, T. N. Padilah, and R. I. Adam, “ANALISIS REGRESI LINIER BERGANDA DALAM ESTIMASI PRODUKTIVITAS TANAMAN PADI DI KABUPATEN KARAWANG”.

[23] K. Bukit Jimbaran and B. -Gedung UKM Lantai, “UNIVERSITAS UDAYANA FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM JURUSAN MATEMATIKA.”