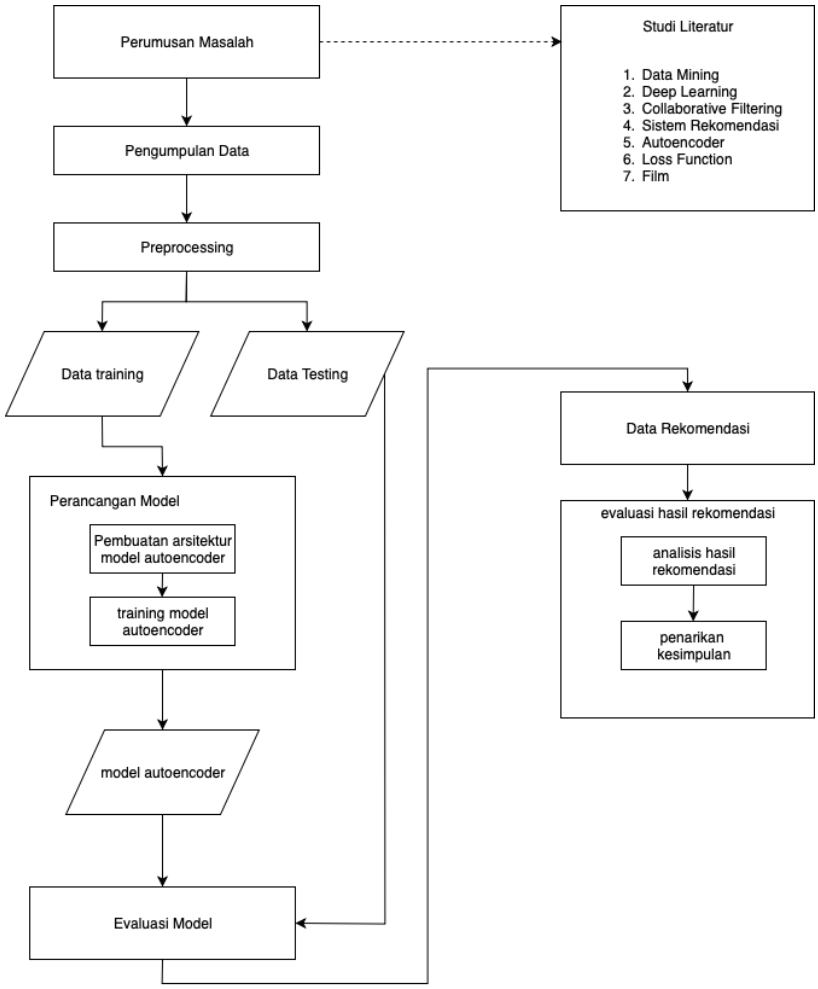
BAB III
METODELOGI PENELITIAN

Bab ini menjelaskan mengenai metodelogi penelitian, mulai dari desain penelitian, perangkat dan data penelitian.

3.1 Desain Penelitian

Pada sub bab ini, akan dipaparkan desain penelitian dari program yang dibuat dalam skripsi ini. Alur penelitian pada gambar 3.1 akan dijelaskan pada sub bab selanjutnya.



Gambar 3.1 Desain Penelitian

3.1.1 Perumusan Masalah

Merupakan tahap awal penelitian. Proses yang terjadi di tahap persiapan yaitu dimulai dari mengidentifikasi masalah yang

akan dibahas, kemudian merumuskan masalah, lalu menentukan metode atau algoritma yang akan digunakan untuk menyelesaikan masalah tersebut, dan yang terakhir adalah menentukan model penelitian untuk membantu penyelesaian masalah.

1.1.2 Studi Literatur

Selanjutnya penulis melakukan studi literatur berkaitan dengan topik yang telah disetujui pada tahap pertama. Pada tahap ini dilakukan studi literatur tentang ilmu *Data Mining, Deep Learning, Collaborative Filtering,* sistem rekomendasi, *Autoencoder, Loss Function,* dan film. Dalam mempelajari tentang bahasan di atas penulis mempelajari dari beberapa sumber, seperti buku, jurnal, juga internet, ataupun bahan bacaan lainnya yang didapat dari berbagai sumber yang kredibel.

1.1.3 Pengumpulan Data

Pengumpulan dataset yang akan digunakan pada penilitian ini diunduh dari movielens. Bahan yang digunakan dalam penelitian ini adalah sekumpulan data set yang didapatkan dari database website MovieLens yang nantinya akan digunakan pada saat penyisipan data ke model yang sudah di rancang. Data set diambil dari Movielens.org sebuah lembaga riset yang memfokuskan kajian tentang mesin rekomendasi (https://movielens.org/). Data set yang dipilih adalah Movielens, data ini terdiri dari 6040 user, 3883 item (film), 1.000.209 rating, serta info konten item berupa genre. Nilai rating yang terdapat pada data set adalah 1, 2, 3, 4, dan 5. Dari 1.000.000 rating.

1.1.4 Preprocessing

Pada proses ini data yang dikumpulkan akan diproses terlebih dahulu agar dapat digunakan untuk proses *training* dan *testing*. Data akan dibagi menjadi 90%-10% untuk data *training* dan data *testing*. Untuk tahap praproses ini menggunakan library *pandas* untuk memasukan mengunggah data training berupa .csv menjadi dataframe , keluaran pada proses ini yaitu data *training* dan data

testing. Semua indeks akan dikurangi satu agar index dimulai dari 0. Pada tahap ini digunakan library *sklearn.data_train_test_split* untuk membagi data test dan data train.

Sebelum membuat arsitektur model, terlebih dahulu dataframe akan dirubah kedalam bentuk matriks agar dapat diaplikasikan kedalam model. Matriks berupa MxN dimana R(i, j) adalah rating yang diberikan oleh user i terhadap item j.

Proses merubah dataframe menjadi matriks yaitu:

1. load dataframe rating yang berisi userid, movieid, dan rating.

user_emb_id	movie_emb_id	rating	timestamp
4032	452	4	965513126
4291	3696	4	965274731
3312	355	4	968914033
3789	777	4	966040424
5462	3460	4	959900645

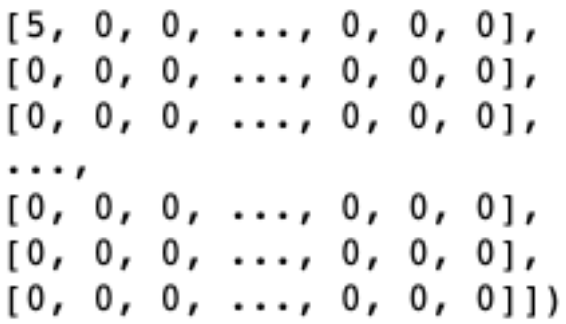
Gambar 3.2 Dataframe rating

2. Membuat matriks dengan baris sejumlah user, dan kolom sejumlah item.

Users	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6	···
User1	0	0	4	0	1	0	···
User2	2	5	2	0	0	2	···
User3	0	0	5	3	2	4	···
User4	1	0	0	4	0	0	···
User5	2	3	0	0	0	0	···
···	···	···	···	···	···	···	···

Gambar 3.3 Matriks rating yang berisi userid dan movieid

3. Mengisi kolom dan baris dengan rating tiap user dan item.
4. Keluaran berupa matriks MxN.



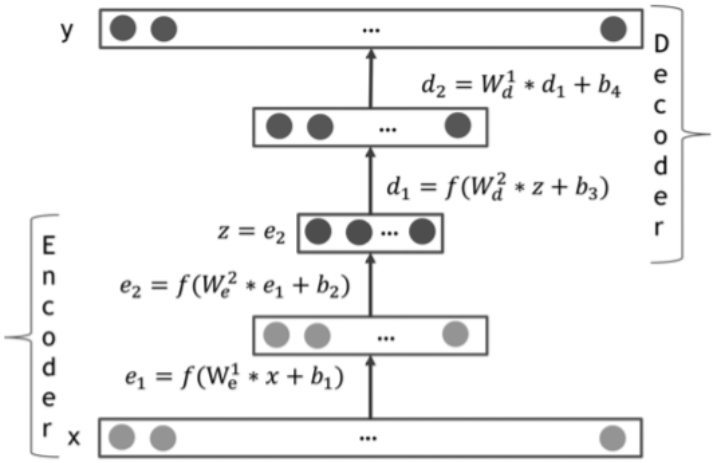
Gambar 3.4 Keluaran matriks

3.1.4 Perancangan Model Deepautoencoder

Pada tahap ini dipersiapkan untuk membangun model deepautoencoder, Seperti yang dijelaskan pada bab 2 Autoencoder adalah model neural network yang memiliki input dan output yang sama (Liou, Cheng, Liou, & Liou, 2014). Autoencoder mempelajari data input dan berusaha untuk melakukan rekonstruksi terhadap data input tersebut. (Wang & Yeung, 2013). Input yang digunakan dalam kasus ini yaitu matriks user dan item dari dataset yang telah melalui tahap praproses sebelumnya. Lalu input akan masuk ke dalam layer encoder untuk dienstraksi fitur dari data tersebut, setelah itu masuk ke proses decoder untuk merekonstruksi ulang data yang telah di-*encode*, keluarannya berupa hasil rekonstruksi input oleh model.

Model yang dibangun merupakan modifikasi dari penelitian model autoencoder yang dilakukan oleh Sedhain,dkk (Sedhain, Menony, Sannery, & Xie, 2015). Penelitian tersebut menggunakan 1 hidden layer dengan fungsi aktivasi *sigmoid* untuk proses ekstraksi fitur data, namun pada penelitian ini digunakan 3 hidden layer dengan asumsi hasil training lebih baik dari penelitian sebelumnya. Ketiga layer tersebut adalah encoder layer dengan jenis aktivasi SELU, latentspace layer, dan decoder layer dengan jenis aktivasi

SELU dengan menambahkan fungsi *dropout* untuk menghindari *overfitting*.



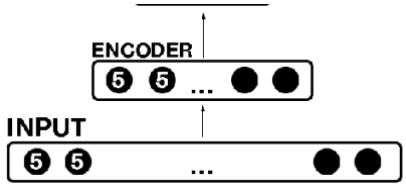
Gambar 3.5 Arsitektur deep autoencoder dengan 5 layer (Sedhain et al., 2015).

Tahapan pada arsitektur deep auto encoder yaitu:

1. Memasukan *layer input* sejumlah matriks user dan item sebanyak 6040 *node*. *User* pertama akan masuk ke dalam jaringan *encoder*, dengan *input x* = (r_1, r_2, …, r_m) berisi nilai rating dari semua item.

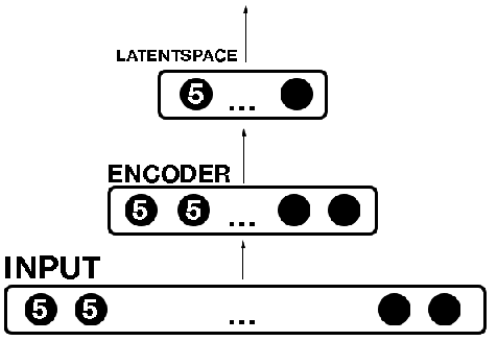
Gambar 3.6 Input rating dari matriks.

2. *input vector x* akan di-*encode* ke bentuk *vector* e_1 oleh fungsi aktivasi *f*. $e_1 = f(W * x + b_1)$ dimana W adalah nilai bobot dan *b* adalah nilai bias. Jumlah *node* nya diperkecil untuk mengektraksi fitur abstraksi yang penting.



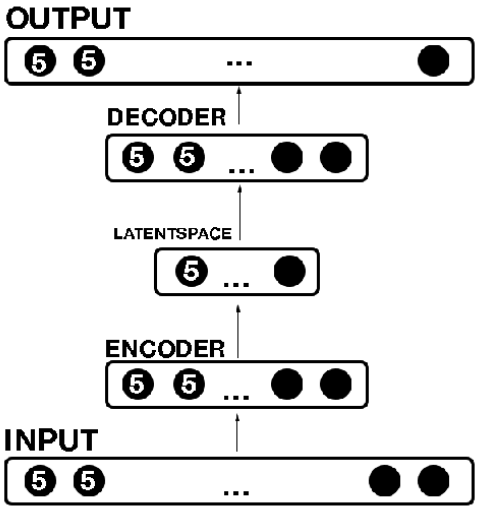
Gambar 3.7 *Input* masuk ke *layer encoder*

3. Layer *latentspace* akan memilah kembali fitur terbaik dari hasil *encode* sebelumnya.



Gambar 3.8 *Input* masuk ke *layer latentspace*

4. Layer *Decoder* akan merekonstruksi ulang hasil dari layer *encoder* sebelumnya. Nilai *error* akan dihitung antara nilai *output* dan nilai *input*, tujuannya yaitu untuk meminimalisir tingkat error.



Gambar 3.9 Rekonstruksi nilai *input* menjadi *input.*

5. Dilakukan *back-propagation* untuk menghitung tingkat *error*, jika tingkat *error* masih tinggi maka bobot akan di-*update*.

3.1.5 Evaluasi dan Analisis Hasil Model

Ditahap ini penulis mengevaluasi dan analisis hasil yang sudah didapatkan berdasarkan tahap pengujian atau training dalam bentuk grafik garis. Analisa hasil model berupa perbandingan hasil prediksi pada dataset yang digunakan pada model yang dibuat menggunakan RMSE. RMSE digunakan untuk mengukur error antara rating prediksi dengan rating sebenarnya.

$$RMSE = \sqrt{\frac{m_i*(r_i - y_i)^2}{\sum_{i=0}^{i=n} m_i}} \qquad \textbf{(4)}$$

Agar hasil prediksi tidak bernilai 0 maka akan dibuat sebuah *mask function* dimana *mask function* adalah m_i akan bernilai 1 jika r_i atau rating sebenarnya tidak sama dengan 0.

r_i adalah rating sebenarnya dan y_i adalah rating prediksi. Dua nilai tersebut akan dikuadratkan agar hasilnya tidak minus lalu dikali m_i, Lalu semua nilai tersebut akan dibagi dengan jumlah *masked function*.

3.1.6 Data Rekomendasi

Data testing akan berupada hasil rekomendasi yang memuat Userid, Moviesid, dan Rating. Akan ditampilkan 10 nilai tertinggi yang didapatkan hasil *testing*.

Proses penampilan data rekomendasi yaitu:

1. Memasukan rating awal
2. Inisialisasi matriks sejumlah data user x item
3. Mengubah rating masukkan menjadi matriks

4. Input user rating awal dimasukan ke dalam matriks yang sudah diinisialisasi
5. Matriks akan diprediksi oleh hasil model training
6. Hasil berupa keluaran *dataframe* prediksi

1.1.7 Evaluasi Hasil Rekomendasi

Pada tahap ini dilakukan analisis terhadap hasil rekomendasi yang dibuat dan pernarikan kesimpulan dari hasil penelitian tersebut.

3.2 Perangkat dan Data Penelitian

Berikut alat dan bahan yang digunakan dalam penelitian ini.

3.2.1 Perangkat Penelitian

1. Perangkat keras (*Hardware)* yaitu sebuah macbook air dengan spesifikasi:
- Processor Intel Core i5-5200U
- Random Acces Memory (RAM) 8 GB
- SSD 256 GB
2. Perangkat lunak (*software)* sebagai berikut :
- Jupyter Notebook
- Microsoft Word 2013
- Sublime Text 3
3. Library yang digunakan dalam penelitian ini yaitu pandas, sckit-learn, dan keras.

Panda Dataframe, menggunakan sistem dataframe, sehingga dapat membuat sebuah file ke dalam tabel virtual seperti spreadsheet dengan menggunakan pandas. Dengan menggunakan pandas, data juda dapat diolah sperti joi, distinct, group by, agregasi, dan teknik lain seperti di SQL. Pandas juga dapat membaca file dari berbagai format seperti .txt, .csv, .tsv, dan lainnya.

Scikit-learn, library untuk machine learning yang mendukung melakukan beragam pekerjaan dalam data science seperti regresi,

klasifikasi, clustering, data preprocessing, dimensionality reduction, dan model selection (perbandingan, validasi, dan pemilihan parameter maupun model). Scikit-learn ditulis dalam bahasa python, dan menggunakan numpy secara ekstensif untuk perhitungan aljabar linier dan operasi array.

Keras, adalah *library* berbasis *open source* yang dirancang untuk menyederhanakan model dari kerangka Deep Learning. Saat ini, Keras dianggap sebagai salah satu *library* pembelajaran mesin terbaik di Python. Keras juga menyediakan beberapa utilitas terbaik dalam hal menyusun model, memproses dataset, memvisualisasikan grafik, dan hal lainnya.

3.2.2 Data Penelitian

Bahan yang digunakan dalam penelitian ini adalah sekumpulan data set yang didapatkan dari database website *MovieLens* yang nantinya akan digunakan pada saat penyisipan data ke model yang sudah di rancang. Data set yang dipilih adalah Movielens, data ini terdiri dari 6040 user, 3883 item (film), 1.000.209 rating, serta info konten item berupa genre. Nilai rating yang terdapat pada data set adalah 1, 2, 3, 4, dan 5. Dari 1.000.000 rating.