

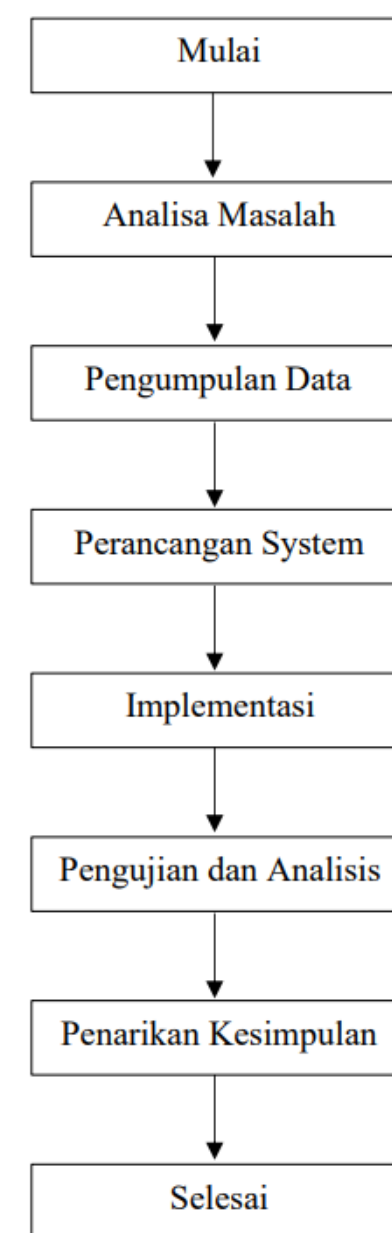
# ANALISIS TOPIC MODELLING JOURNAL ARXIV MENGUNAKAN METODE K-MEANS DENGAN ALGORITMA DIMENSIONALITY REDUCTION DAN T-SNE MODEL

DEMA RIANDI AKHMAD

201710370311116

## ABSTRAK

Dengan menggunakan k-means untuk mengkategorikan data teks ke dalam kelompok yang mewakili topik berbeda dan selanjutnya memanfaatkan PCA untuk reduksi dimensi, metodologi gabungan ini memberdayakan peneliti untuk mengungkap dan memahami tema laten secara efektif. Kesimpulannya, penggabungan *Cluster K-Means* dengan PCA mewakili jalan yang menjanjikan bagi para peneliti untuk mencari wawasan berbeda dari kumpulan data tekstual. Pendekatan terpadu ini memfasilitasi ekstraksi topik yang komprehensif, membantu penemuan pengetahuan dan proses pengambilan keputusan di berbagai domain. Penelitian kali ini menunjukkan antara metode K-Means dengan menggunakan algoritma pereduksi dimensi PCA memberikan hasil yang sangat memuaskan dengan mengkombinasi visualisasi antara algoritma t-SNE dan UMAP.



## HASIL PENGUJIAN

```
[ ] from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier

# random forest classifier instance
forest_clf = RandomForestClassifier(n_estimators=100, random_state=42, n_jobs=4)

# cross validation on the training set
forest_scores = cross_val_score(forest_clf, X_train, y_train, cv=3, n_jobs=4)

# print out the mean of the cross validation scores
print("Accuracy: ", '{:,.3f}'.format(float(forest_scores.mean()) * 100), "%")
Accuracy: 91.186 %

[ ] from sklearn.model_selection import cross_val_predict
from sklearn.metrics import precision_score, recall_score, f1_score

# cross validate predict on the training set
forest_train_pred = cross_val_predict(forest_clf, X_train, y_train, cv=3, n_jobs=4)

# print precision and recall scores
print("Precision: ", '{:,.3f}'.format(float(precision_score(y_train, forest_train_pred, average='macro')) * 100), "%")
print(" Recall: ", '{:,.3f}'.format(float(recall_score(y_train, forest_train_pred, average='macro')) * 100), "%")
print(" F1 score: ", '{:,.3f}'.format(float(f1_score(y_train, forest_train_pred, average='macro')) * 100), "%")
Precision: 92.141 %
Recall: 88.072 %
F1 score: 89.706 %
```

