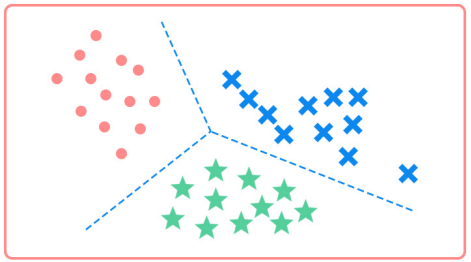
# Pemograman Web Lanjut

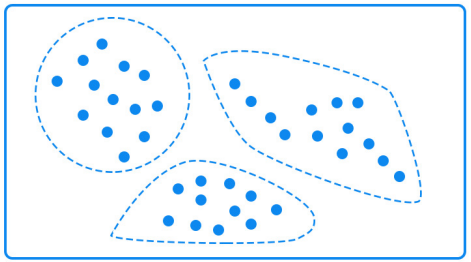
Febri Damatraseta Fairuz, S.T, M.Kom

# Machine Learning & Computer Vision

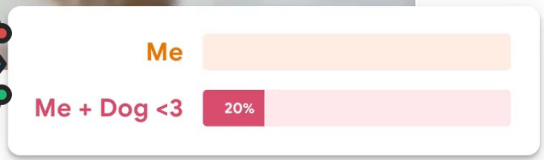
MACHINE LEARNING



Supervised learning



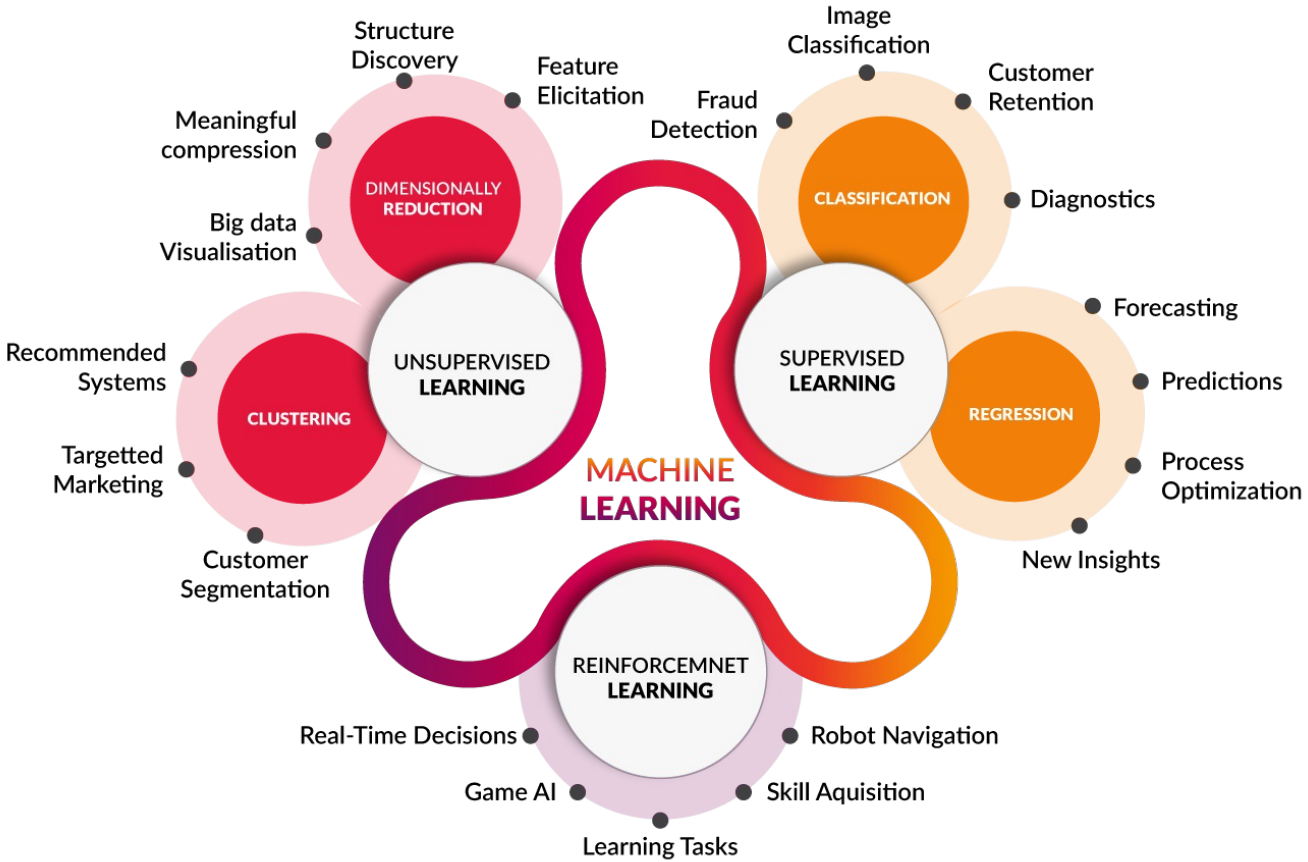
Unsupervised learning



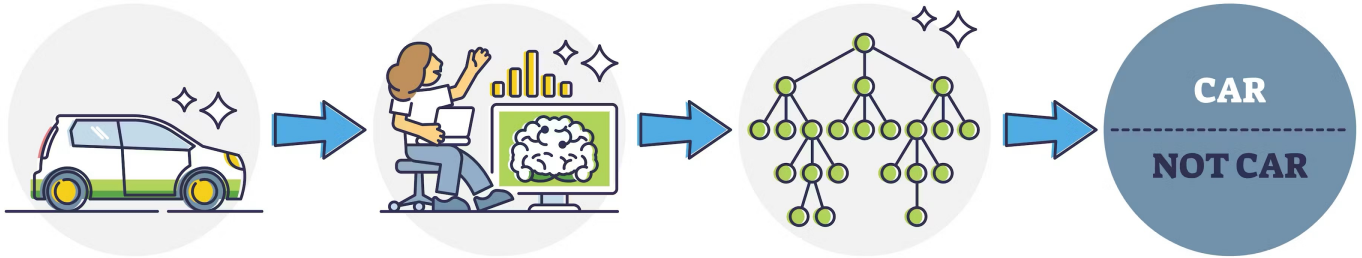

COMPUTER VISION

# Machine Learning

# Taxonomi of Machine Learning

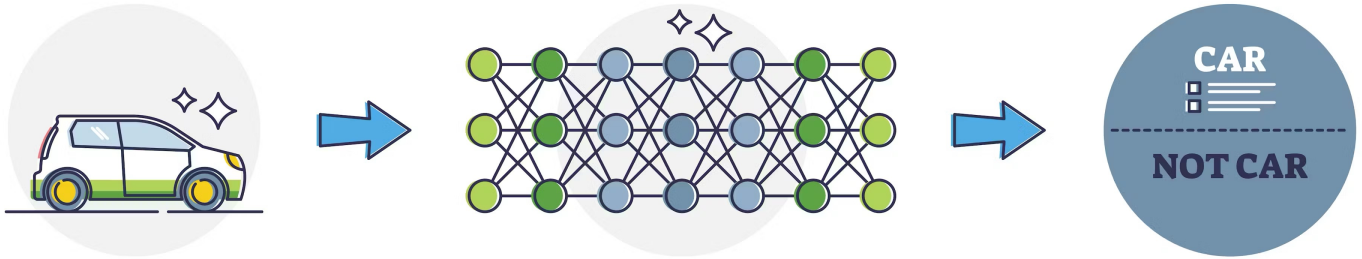


## MACHINE LEARNING




INPUT | FEATURE EXTRACTION | CLASSIFICATION | OUTPUT

## DEEP LEARNING

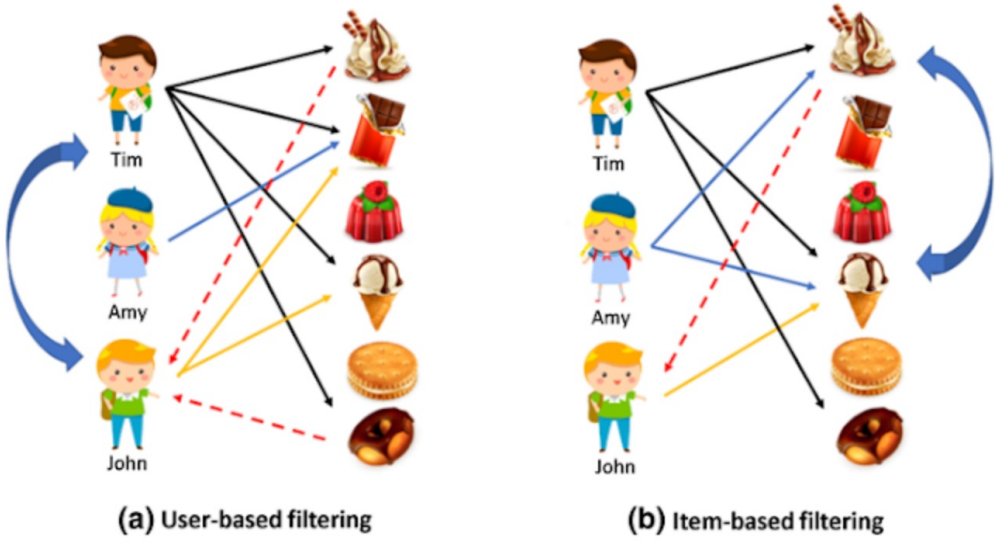



INPUT | FEATURE EXTRACTION + CLASSIFICATION | OUTPUT

# Example ML vs DL

Sistem rekomendasi makanan:

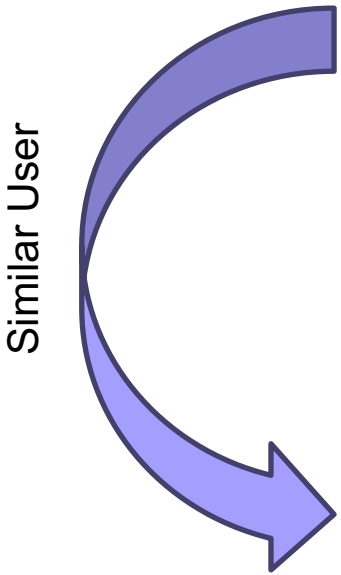
Teknik ML yang dipakai Collaborative Filtering (CF)



# Feature Extraction

## User-based

**collaborative filtering:**





Septian


Jemy


Anton


Nasi Padang


Ayam Gemes


Ice Cream


Mie Goreng

# Feature Extraction

## User-based

**collaborative filtering**

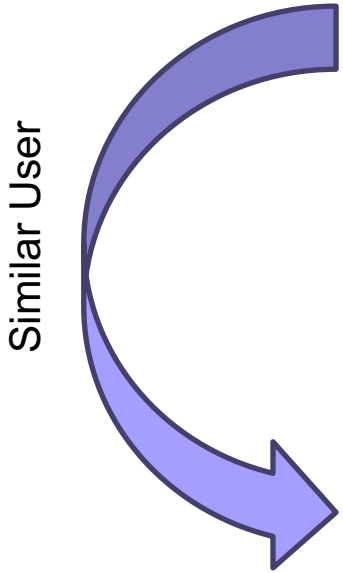
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

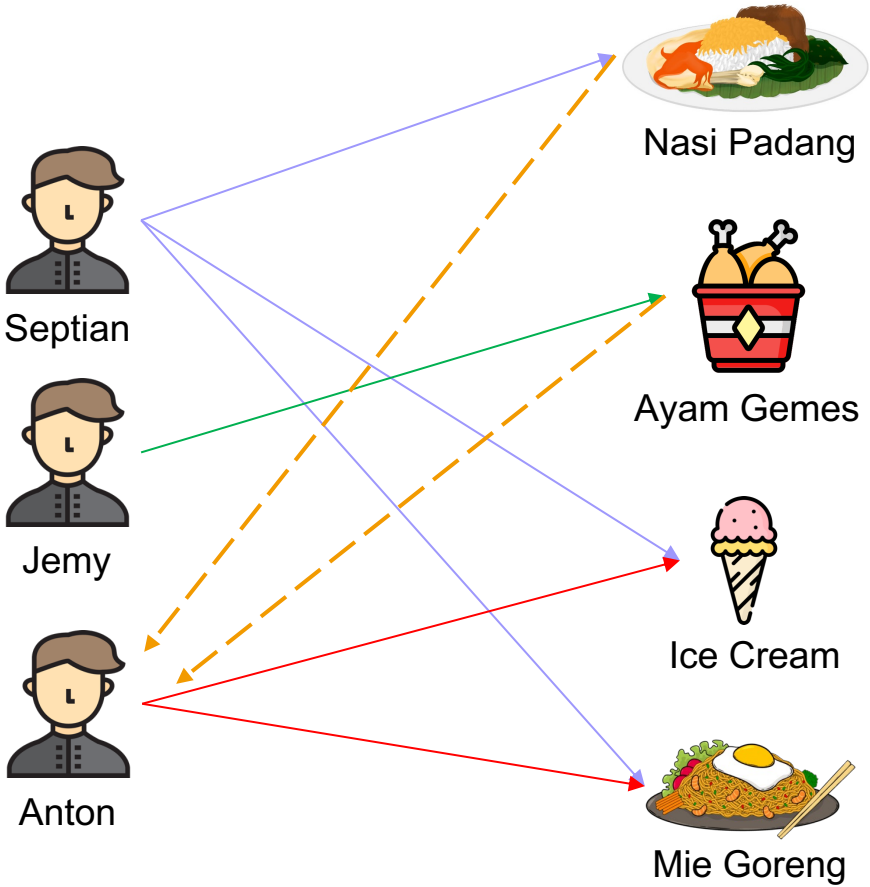
| | Nasi Padang | Ayam Gemes | Ice cream | Mie goreng | Mean |
|---|---|---|---|---|---|
| **Septian** | 5 | 0 | 1 | 6 | 3 |
| **Jemy** | 0 | 3 | 0 | 0 | 0.75 |
| **Anton** | ? | ? | 4 | 3 | 1.75 |

# Classification

## User-based

**collaborative filtering:**

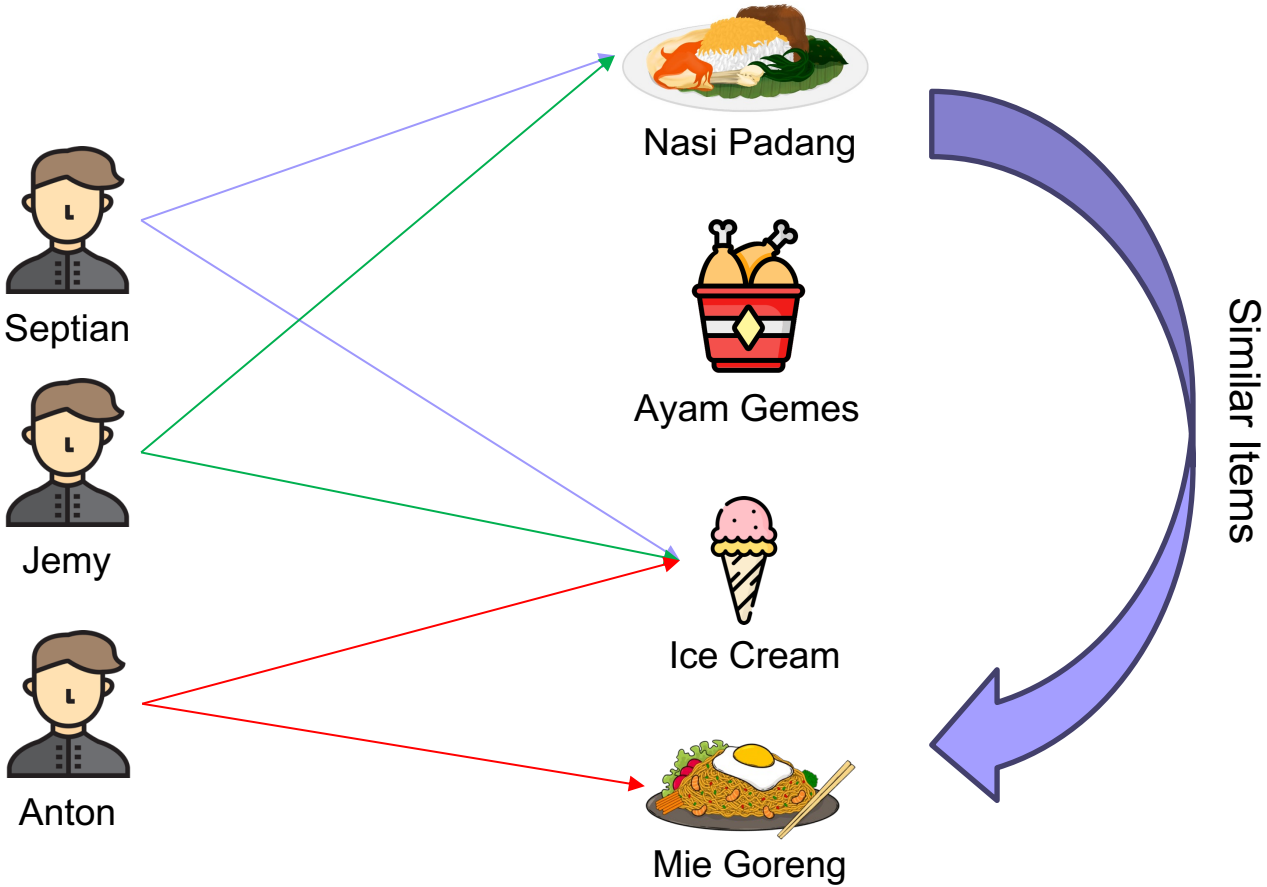


# Feature Extraction

## Item-based

**collaborative filtering:**



# Feature Extraction

## Item-based

**collaborative filtering**

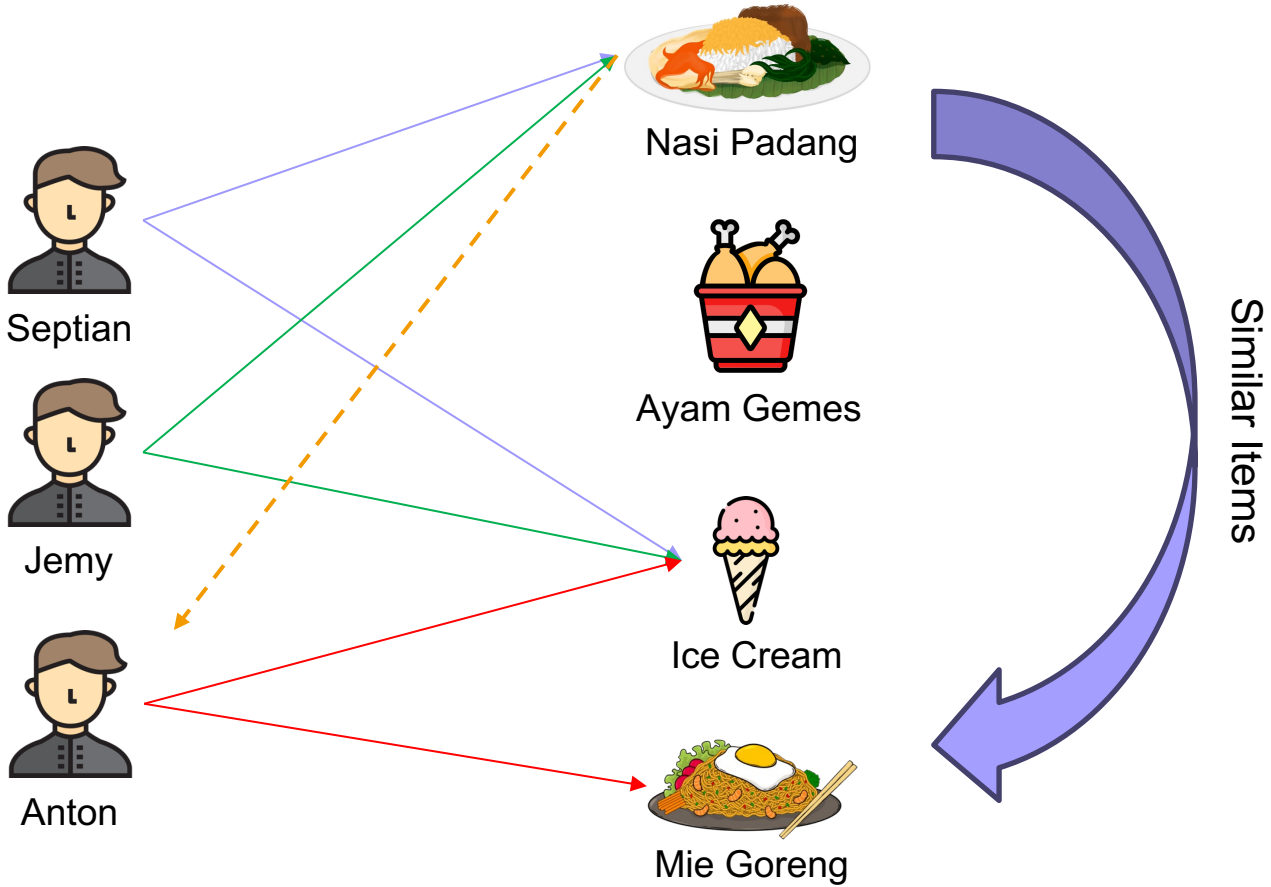
$$r(A,i2) = \frac{\left((r(A,i1) * sim(i2,i1)) + (r(A,i3) * sim(i2,i3)\right)}{sim(i2,i1) + sim(i2,i3)}$$

$$r(A,i2) = \frac{\left((2 * 0.904) + (3 * 0.868)\right)}{0.904 + 0.868} = 2.49$$

# Classification

## Item-based

**collaborative filtering:**

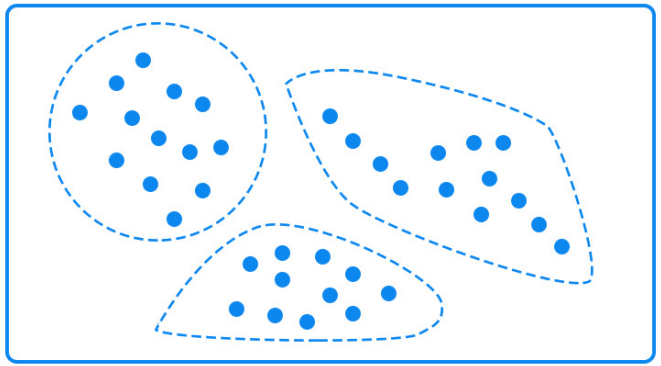


# Implementation CF

```
const users = [
    { id: 1, name: "Septian", ratings: { 1: 5, 2: 0, 3: 1, 4: 6 } },
    { id: 2, name: "Jemy", ratings: { 1: 0, 2: 3, 3: 0, 4: 0 } },
    { id: 3, name: "Anton", ratings: { 1: 0, 2: 0, 3: 4, 4: 3 } },
];

const foods = [
    { id: 1, name: "Nasi Padang" },
    { id: 2, name: "Ayam Gemes" },
    { id: 3, name: "Ice Cream" },
    { id: 4, name: "Mie Goreng" },
];
```

**User-based**

**collaborative filtering**



Unsupervised learning

# Implementation CF

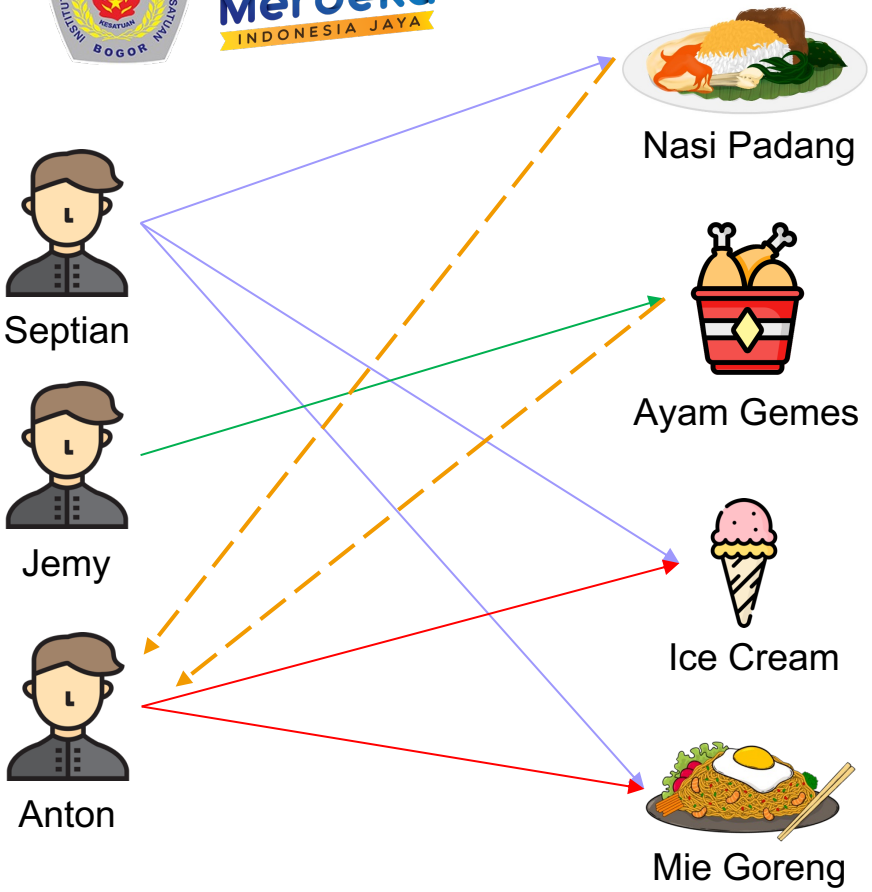
## User-based

collaborative filtering

```
const recommendationSimilarity = (userId) => {
const user = users.find((u) => u.id === userId);
if (user) {
    const similarities = users
    .filter((u) => u.id !== userId)
    .map((otherUser) => {
       const commonRatings = Object.keys(user.ratings)
       .filter((foodId) => otherUser.ratings[foodId] !== undefined)
       .map((foodId) => ({
       foodId: parseInt(foodId),
       rating: otherUser.ratings[foodId],
    }));

    const similarity = commonRatings.reduce((acc, cur) => {
      return acc + Math.abs(user.ratings[cur.foodId] - cur.rating);
    }, 0);

    return {
       user: otherUser,
       similarity: 1 / (1 + similarity),};
    });

    similarities.sort((a, b) => b.similarity - a.similarity);

    const bestMatch = similarities[0].user;
    const recommendations = Object.keys(bestMatch.ratings)
    .filter((foodId) => user.ratings[foodId] === 0).map((foodId) => ({
      food: foods.find((f) => f.id === parseInt(foodId)),
      rating: bestMatch.ratings[foodId],
    }));

    return { similarities, recommendations };
   }
};
```

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

# Implementation CF




User:

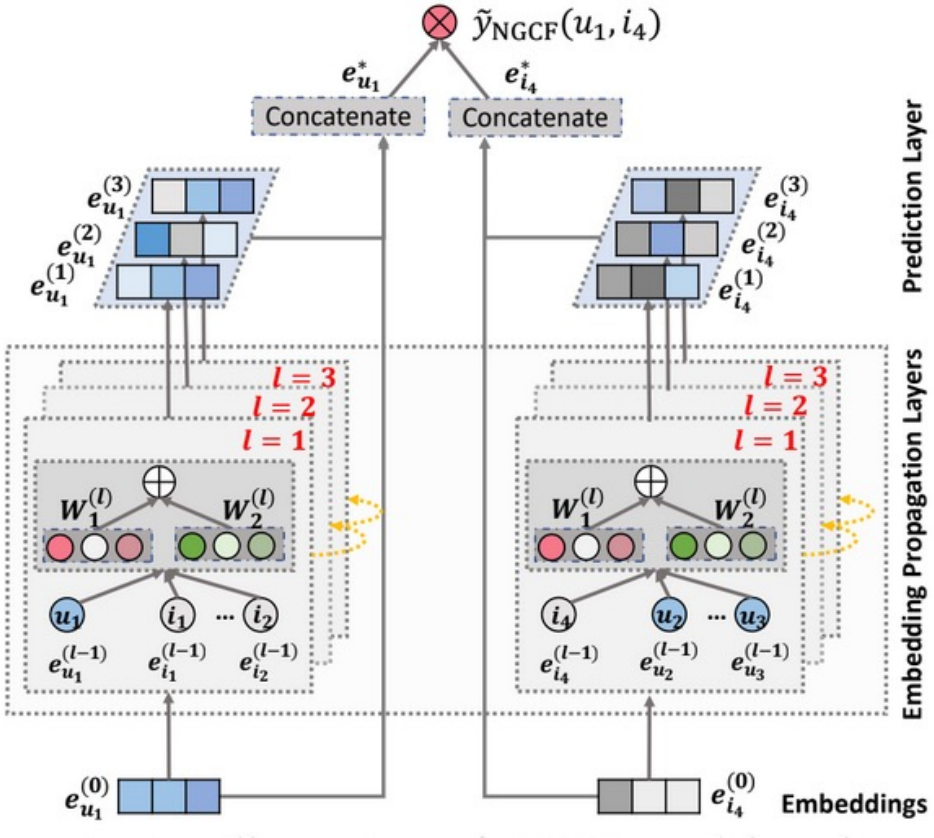
Anton

**Recommended Food:**

1. Nasi Padang
2. Ayam Gemes

# Deep Learning

# Neural Collaborative Filtering



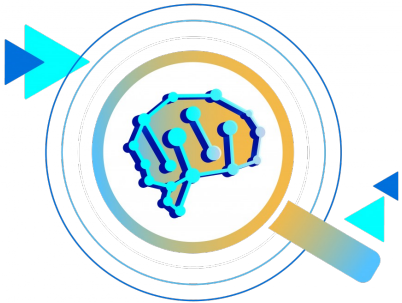

$$\mathbf{x} = \mathbf{p}_u \odot \mathbf{q}_i$$
$$\hat{y}_{ui} = \alpha(\mathbf{h}^\top \mathbf{x}),$$

$$z^{(1)} = \phi_1(\mathbf{U}_u, \mathbf{V}_i) = [\mathbf{U}_u, \mathbf{V}_i]$$
$$\phi^{(2)}(z^{(1)}) = \alpha^1(\mathbf{W}^{(2)} z^{(1)} + b^{(2)})$$
$$\dots$$
$$\phi^{(L)}(z^{(L-1)}) = \alpha^L(\mathbf{W}^{(L)} z^{(L-1)} + b^{(L)}))$$
$$\hat{y}_{ui} = \alpha(\mathbf{h}^\top \phi^L(z^{(L-1)}))$$

$$\hat{y}_{ui} = \sigma(\mathbf{h}^\top [\mathbf{x}, \phi^L(z^{(L-1)})]).$$

# Deep Learning

Deep learning is a subset of machine learning that uses multilayered neural networks,  called deep neural networks, to simulate the complex decision-making power of the human brain.
Some form of deep learning powers most of the artificial intelligence (AI) applications in our lives today.



Deep learning is an aspect of data science that drives many applications and services that improve automation, performing analytical and physical tasks without human intervention. This enables many everyday products and services—such as digital assistants, voice-enabled TV remotes, credit card fraud detection, self-driving cars and generative AI

# Types of Deep Learning models

1. Convolutional neural networks (CNNs or ConvNets)
2. Recurrent neural networks (RNNs)
3. Generative adversarial networks (GANs)
4. Diffusion models
5. Transformer models

# DL – Tech Stack

**Libraries:**





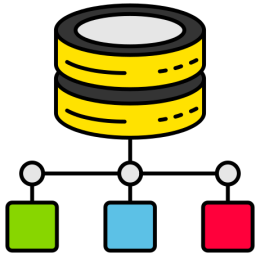
**Datasets:**



**Backend Language:**

# Step of Deep Learning

1. Pre – Processing
2. Pasca – Processing
3. Post - Processing



Data Modeling

# Step of Deep Learning

**1. Pre – Processing**
Mempersiapkan pembuatan dataset.

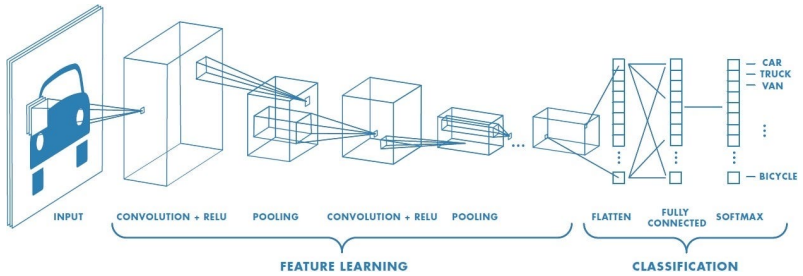
A. Pengumpulan Data
B. Pembersihan Data
C. Normalisasi Data
D. Pembagian Data
E. Augmentasi Data
F. Dimensi Data

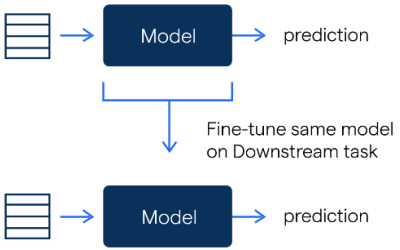
# Step of Deep Learning

**2. Pasca – Processing**

Menentukan data model yang akan dibangun.

A. Menentukan model Deep Learning
B. Penyesuaian model (train, validation dan testing)
C. Fine-tuning
D. Deploying model



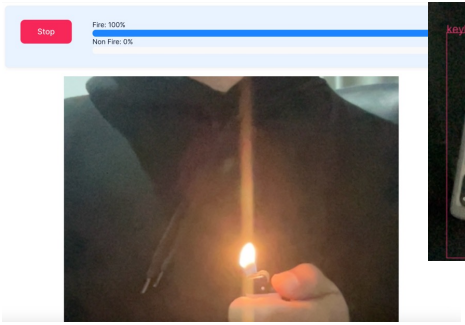



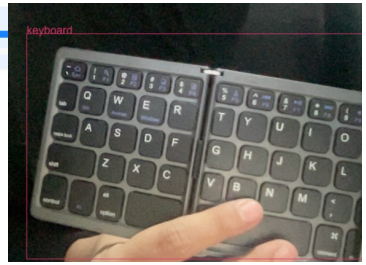
# Step of Deep Learning

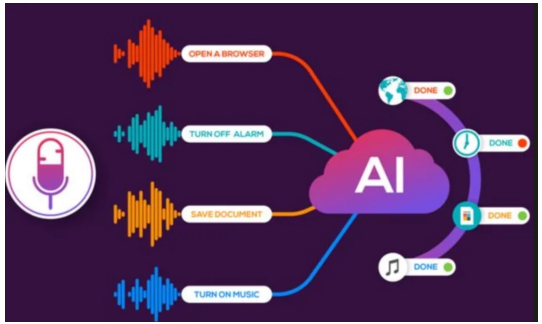
**2. Post – Processing**

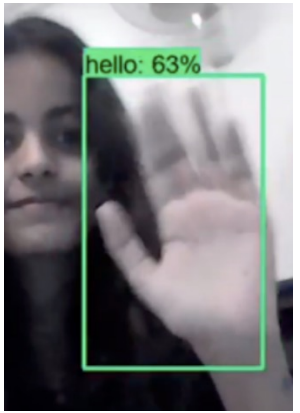
Membangun sistem prototipe AI.

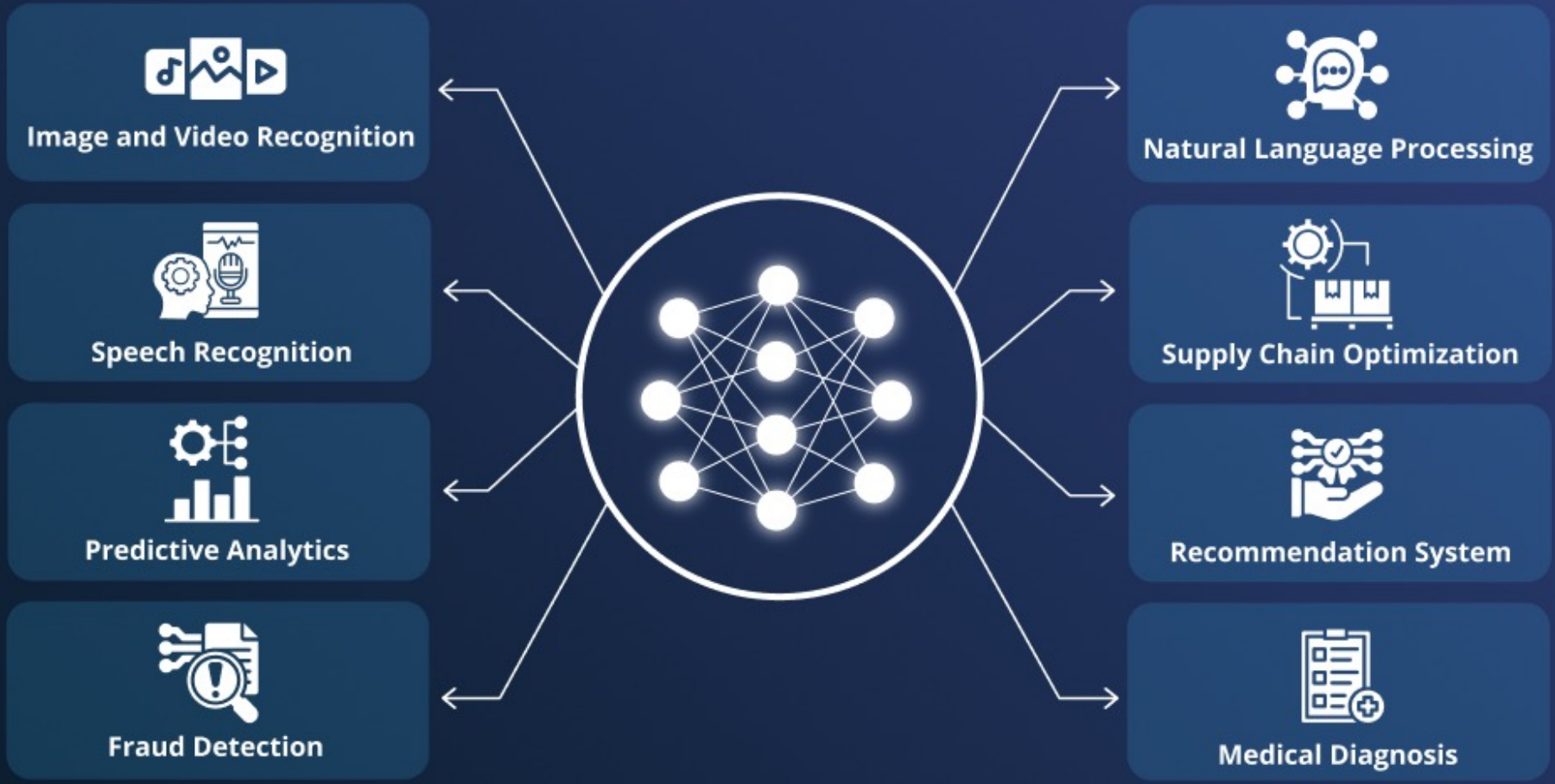
A. Evaluasi Hasil Data Model
B. Penyaringan Hasil data model
C. Rescaling
D. Integrasi data model dengan sistem






Leewayhertz

# THANK YOU