

# Variabel Penelitian & Hipotesis

## Metodologi Penelitian Kuantitatif

Hendri Karisma, M.T.

Dosen Teknik Informatika, STMIK Tazkia, Bogor, Indonesia

VP Engineering, Jejakin.com

`hendri@stmik.tazkia.ac.id` — `hendri.karisma@jejakin.com`

Program Studi Teknik Informatika  
STMIK Tazkia, Bogor, Indonesia

2026

Setelah pertemuan ini, mahasiswa mampu:

- 1 Memahami konsep variabel dalam riset kuantitatif *Computer Science*.
- 2 Mengidentifikasi variabel independen, dependen, dan kontrol.
- 3 Menyusun tabel operasionalisasi variabel secara lengkap.
- 4 Memahami perbedaan hipotesis penelitian dan hipotesis statistik.
- 5 Merumuskan pasangan  $H_0$  dan  $H_1$  yang tepat.

# Apa Itu Variabel Penelitian?

## Definisi

Variabel adalah atribut, karakteristik, atau faktor yang dapat **diukur**, **dimanipulasi**, atau **diamati** dalam suatu penelitian.

Keunikan variabel dalam riset *Computer Science*:

- **Deterministik** – Dapat direproduksi jika kondisi eksperimen sama.
- **Terukur secara presisi** – Menggunakan *profiler*, *benchmark tool*, dan *log* sistem.
- **Dapat dikontrol ketat** – *Hardware*, *software*, dan dataset dapat distandarkan.

*Berbeda dengan riset sosial yang berbasis kuesioner, riset CS berbasis parameter komputasi.*

# Tiga Jenis Variabel dalam Eksperimen

## Independen ( $X$ )

Faktor yang **dimanipulasi** oleh peneliti.

*“Apa yang saya ubah?”*

Contoh: jenis algoritma, nilai parameter, teknik *preprocessing*.

## Dependen ( $Y$ )

Metrik yang **diukur** sebagai hasil eksperimen.

*“Apa yang saya ukur?”*

Contoh: akurasi, *F1-Score*, *running time*, *throughput*.

## Kontrol ( $Z$ )

Faktor yang **dijaga konstan** selama eksperimen.

*“Apa yang saya jaga tetap?”*

Contoh: *hardware*, dataset, *random seed*.

# Variabel Independen – Contoh dari Berbagai Domain

<b>Domain</b>	<b>Variabel Independen</b>	<b>Nilai</b>
<i>Machine Learning</i>	Jenis algoritma klasifikasi	SVM, RF, KNN
<i>Deep Learning</i>	Fungsi aktivasi	ReLU, Sigmoid, Tanh
<i>Deep Learning</i>	Nilai <i>learning rate</i>	0.001, 0.01, 0.1
<i>Computer Vision</i>	Arsitektur CNN	VGG-16, ResNet-50
NLP	Metode <i>embedding</i>	Word2Vec, BERT
Keamanan Jaringan	Algoritma enkripsi	AES-128, ChaCha20
IoT	Protokol komunikasi	MQTT, CoAP, HTTP
<i>Cloud Computing</i>	Strategi <i>load balancing</i>	Round Robin, Least Conn.
<i>Data Mining</i>	Teknik <i>feature selection</i>	PCA, Chi-Square
<i>Software Engineering</i>	Teknik pengujian	<i>White-box, Black-box</i>

# Variabel Dependen – Metrik Performa Standar

<b>Metrik</b>	<b>Satuan</b>	<b>Domain</b>
<i>Accuracy</i>	%	Klasifikasi ML, <i>Computer Vision</i>
<i>Precision</i>	%	<i>Information Retrieval</i> , NLP
<i>Recall</i>	%	Deteksi anomali, diagnosa medis AI
<i>F1-Score</i>	0–1	Kelas tidak seimbang
<i>AUC-ROC</i>	0–1	Klasifikasi biner
MSE / RMSE	Bervariasi	Regresi, prediksi numerik
<i>Running Time</i>	detik, ms	Algoritma, optimasi
<i>Throughput</i>	MB/s	Jaringan, enkripsi
<i>Latency</i>	ms	Sistem <i>real-time</i> , IoT
<i>Memory Usage</i>	MB, GB	Efisiensi algoritma
<i>FPS</i>	frame/s	<i>Computer Vision real-time</i>

## Prinsip Utama

Ubah **hanya satu** variabel pada satu waktu (*ceteris paribus*).

Semua kondisi lain harus identik antara kelompok kontrol dan eksperimen.

Variabel Kontrol	Contoh Nilai
Spesifikasi <i>hardware</i>	Intel i7-12700, RTX 3060, 16 GB RAM
Dataset	UCI Heart Disease (303 record)
Rasio <i>train-test split</i>	80% train, 20% test
<i>Random seed</i>	random_state=42
Versi <i>library</i>	Python 3.10, Scikit-learn 1.2.2
Jumlah <i>epoch</i>	100 <i>epoch</i> , <i>batch size</i> 32
Metode validasi	<i>10-Fold Cross Validation</i>

## Definisi

Proses mendefinisikan variabel secara **konkret dan terukur** sehingga dapat diamati dalam eksperimen.

Komponen tabel operasionalisasi:

- 1 **Variabel:** Nama variabel yang diteliti.
- 2 **Indikator:** Aspek spesifik yang diukur.
- 3 **Skala Pengukuran:** Nominal, ordinal, interval, atau rasio.
- 4 **Instrumen / Alat Ukur:** Cara pengumpulan data.

*Operasionalisasi menjawab: “Bagaimana tepatnya variabel ini diukur?”*

# Contoh: Operasionalisasi Deteksi *Malware*

**Judul:** Perbandingan Random Forest dan XGBoost untuk Deteksi *Malware* Android

Variabel	Jenis	Indikator	Skala	Instrumen
Algoritma	Indepen.	RF vs XGBoost	Nominal	Scikit-learn
Akurasi	Depen.	Rasio prediksi benar	Rasio	<code>accuracy_score()</code>
<i>F1-Score</i>	Depen.	Harmonik <i>Prec. &amp; Rec.</i>	Rasio	<code>f1_score()</code>
Waktu	Depen.	Durasi <i>training</i>	Rasio	<code>time.time()</code>
Dataset	Kontrol	CICMalDroid 2020	–	Tetap
<i>Seed</i>	Kontrol	Nilai = 42	–	<code>random_state</code>

# Hipotesis Penelitian vs Hipotesis Statistik

## Hipotesis Penelitian

Pernyataan verbal tentang hubungan antar variabel.

*"Penggunaan SMOTE mampu meningkatkan akurasi klasifikasi Decision Tree pada dataset tidak seimbang."*

## Hipotesis Statistik

Formulasi matematis yang dapat diuji secara kuantitatif.

$$H_0: \mu_{SMOTE} \leq \mu_{tanpa}$$

$$H_1: \mu_{SMOTE} > \mu_{tanpa}$$

## Penting

Hipotesis statistik **selalu** ditulis berpasangan:  $H_0$  dan  $H_1$ .

# $H_0$ dan $H_1$ : Pasangan yang Tidak Terpisahkan

## $H_0$ – Null Hypothesis (Status Quo)

- Menyatakan **tidak ada** perbedaan / pengaruh / hubungan signifikan.
- Diterima jika  $p\text{-value} > \alpha$  (biasanya  $\alpha = 0.05$ ).

## $H_1$ – Alternative Hypothesis (Klaim Riset)

- Menyatakan **ada** perbedaan / pengaruh / hubungan signifikan.
- Diterima jika  $p\text{-value} \leq \alpha$ .

## Catatan Filosofis

Kita tidak pernah “membuktikan”  $H_1$  secara absolut. Kita hanya **menolak**  $H_0$  berdasarkan bukti statistik yang cukup kuat.

# Hipotesis Satu Arah vs Dua Arah

## Dua Arah (*Two-tailed*)

Hanya menguji **ada tidaknya** perbedaan.

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

*“Terdapat perbedaan akurasi antara SVM dan Random Forest.”*

## Satu Arah (*One-tailed*)

Menguji **arah spesifik** perbedaan.

$$H_0 : \mu_1 \leq \mu_2$$

$$H_1 : \mu_1 > \mu_2$$

*“Akurasi CNN dengan augmentasi lebih tinggi daripada tanpa augmentasi.”*

## Kapan Menggunakan Satu Arah?

Hanya jika ada **dasar teori yang kuat** dari literatur untuk memprediksi arah.

# Contoh 1: Komparasi Dua Algoritma

**Topik:** Perbandingan SVM dan Random Forest untuk klasifikasi sentimen.

- **Var. Independen:** Jenis algoritma (SVM vs Random Forest)
- **Var. Dependen:** Akurasi (%), *F1-Score*
- **Var. Kontrol:** Dataset, *split ratio*, *seed*, *hardware*

## Hipotesis (Dua Arah)

$H_0 : \mu_{SVM} = \mu_{RF}$  (Tidak ada perbedaan akurasi)

$H_1 : \mu_{SVM} \neq \mu_{RF}$  (Terdapat perbedaan akurasi)

Uji statistik: *Paired t-test* atau *Wilcoxon Signed-Rank Test*.

## Contoh 2: Pengaruh *Parameter Tuning*

**Topik:** Pengaruh *learning rate* terhadap akurasi model LSTM.

- **Var. Independen:** Nilai *learning rate* (0.001, 0.01, 0.1)
- **Var. Dependen:** Akurasi prediksi (%)

### Hipotesis (Lebih dari 2 Kelompok)

$$H_0 : \mu_{0.001} = \mu_{0.01} = \mu_{0.1}$$

$$H_1 : \text{Minimal satu } \mu_i \neq \mu_j$$

Uji statistik: *One-Way ANOVA* (jika data normal) atau *Kruskal-Wallis* (jika tidak normal).

## Contoh 3: Pengaruh *Preprocessing* (Satu Arah)

**Topik:** Pengaruh teknik SMOTE terhadap *F1-Score* pada dataset tidak seimbang.

- **Var. Independen:** Penerapan SMOTE (dengan vs tanpa)
- **Var. Dependen:** *F1-Score*

### Hipotesis (Satu Arah)

$$H_0 : \mu_{\text{SMOTE}} \leq \mu_{\text{tanpa}} \quad (F1\text{-Score tidak lebih tinggi)}$$

$$H_1 : \mu_{\text{SMOTE}} > \mu_{\text{tanpa}} \quad (F1\text{-Score lebih tinggi)}$$

Dasar teori: Literatur menunjukkan SMOTE efektif untuk *imbalanced class*.

# Studi Kasus Lengkap: Deteksi Kendaraan *Real-time*

**Judul:** Evaluasi YOLOv8 vs SSD MobileNetV2 untuk Deteksi Kendaraan

**Variabel:**

- **Independen:** Model deteksi (YOLOv8 vs SSD MobileNetV2)
- **Dependen:**  $mAP@0.5$  (%),  $FPS$ ,  $inference\ time$  (ms)
- **Kontrol:** Dataset COCO Traffic, resolusi  $640 \times 640$ , GPU RTX 3060, CUDA 11.8

**Hipotesis 1 ( $mAP$ ):**  $H_0: \mu_{YOLO} = \mu_{SSD}$      $H_1: \mu_{YOLO} \neq \mu_{SSD}$

**Hipotesis 2 ( $FPS$ ):**  $H_0: \mu_{FPS(YOLO)} = \mu_{FPS(SSD)}$      $H_1: \mu_{FPS(YOLO)} \neq \mu_{FPS(SSD)}$

Uji: *Independent t-test* atau *Mann-Whitney U Test*.

# Kesalahan Umum yang Harus Dihindari

## 1 Mencampur variabel independen dan kontrol

Variabel independen = yang diubah; kontrol = yang dijaga tetap.

## 2 Hipotesis terlalu umum

“Metode A lebih baik” → Lebih baik *apa?* Spesifikkan metriknya!

## 3 Tidak menulis $H_0$

$H_0$  dan  $H_1$  **wajib** ditulis berpasangan.

## 4 Satu arah tanpa justifikasi

Hipotesis satu arah memerlukan dasar teori yang kuat.

## 5 Variabel dependen tidak terukur

“Performa sistem” → Metrik spesifik apa yang diukur?

## 6 Mengabaikan variabel kontrol

*Hardware*, versi *software*, parameter tetap wajib dicantumkan.

## Ringkasan Pertemuan 4

Aspek	Keterangan
Var. Independen ( $X$ )	Faktor yang dimanipulasi (algoritma, parameter)
Var. Dependen ( $Y$ )	Metrik yang diukur (akurasi, waktu, <i>throughput</i> )
Var. Kontrol ( $Z$ )	Faktor dijaga konstan (dataset, <i>hardware</i> , <i>seed</i> )
$H_0$	Tidak ada perbedaan/pengaruh signifikan
$H_1$	Ada perbedaan/pengaruh signifikan
Dua arah	$\mu_1 \neq \mu_2$
Satu arah	$\mu_1 > \mu_2$ atau $\mu_1 < \mu_2$

### Kunci Sukses

Identifikasi variabel yang jelas  $\rightarrow$  Operasionalisasi yang konkret  $\rightarrow$  Hipotesis yang *testable*.

# Tugas Pertemuan 4

Berdasarkan topik penelitian yang telah Anda pilih:

- 1 Identifikasi **variabel independen**, **dependen**, dan **kontrol**.
- 2 Buat **tabel operasionalisasi variabel** lengkap (variabel, indikator, skala, instrumen).
- 3 Rumuskan **hipotesis penelitian** (narasi verbal).
- 4 Rumuskan **hipotesis statistik** ( $H_0$  dan  $H_1$ ) dengan notasi matematis.
- 5 Tentukan apakah hipotesis Anda **satu arah** atau **dua arah**, serta berikan alasannya.

## Referensi Wajib

Oates, B. J. (2005). *Researching Information Systems and Computing*. SAGE.

Wohlin, C. et al. (2012). *Experimentation in Software Engineering*. Springer.