

## Matriks 1 — Kumpulan Insight Informatif (bukan problem)

Area	Insight Informatif	Bukti/Angka	Dampak Teknis
Struktur & labeling	Struktur folder <b>clean</b> : 1 folder = 1 kelas (defect/longberry/peaberry/premium), tanpa nested/noise folder	Struktur dataset eksplisit	Loader bisa simpel (ImageFolder/custom), label mapping statik & reproducible
Dataset size	Total <b>1211</b> images (kecil–menengah)	1211	Perlu augmentasi + regulasi agar tidak overfit
Class balance	Distribusi <b>near-perfect balanced</b>	defect 300, longberry 301, peaberry 310, premium 300 (ratio max/min 1.033)	Tidak perlu class weighting/oversampling; baseline CE loss cukup
Integrity	<b>0 corrupt images</b>	0 corrupt	Pipeline training bisa clean (tanpa skip/try-except khusus)
Visual semantics per kelas	longberry & peaberry punya identitas morfologi kuat; premium relatif “ideal”; defect berisi abnormalitas	Observasi visual STEP 3	Model perlu menangkap fitur bentuk global (shape) + tekstur lokal
Task nature	Problem bersifat <b>hybrid</b> : shape-dominant (longberry/peaberry/premium) vs texture/irregularity-dominant (defect)	STEP 3.5 (blur/edge)	Pilih backbone yang mampu global+local (ResNet/EfficientNet, dsb.)
Resolusi & aspect	Semua image sudah <b>256×256</b> dan aspect <b>1:1</b> tanpa variasi	width=256, height=256, ar=1.0, std=0	Tidak perlu strategi resize kompleks; input size bisa fixed; fokus ke augmentasi & regularisasi
Color & lighting (global)	Dataset <b>bright-biased</b> karena background putih dominan	RGB condong 200–255	Warna bukan sinyal utama; risiko shortcut dari lighting/background
Warna per kelas	Tidak ada <b>color signature</b> kuat yang memisahkan kelas	Histogram per kelas overlap	RGB tetap disarankan (sinyal halus), tapi warna hanya secondary cue
Grayscale vs RGB	Grayscale cukup untuk baseline, namun RGB lebih aman untuk performa maksimal	Grayscale mirip luminance global	Boleh uji baseline grayscale, tapi default tetap RGB

Area	Insight Informatif	Bukti/Angka	Dampak Teknis
Tekstur proxy	Edge density mean antar kelas mirip, namun defect punya <b>variance &amp; tail ekstrem</b>	mean defect~0.0168; max defect~0.0506; std defect tinggi	Tekstur bukan pemisah linear; yang penting pola spasial edge (retak/pecah)
Duplicates (exact)	Ada exact duplicates <b>intra-class</b> (bukan leakage)	11 grup; cross-class 0	Boleh dedup untuk mengurangi redundancy/bias efektif
Near-duplicates	Near-duplicate <b>lintas kelas</b> cukup banyak → sinyal ambiguity	64 pairs; 33 cross-class (~52%)	Harus group-aware split agar evaluasi tidak bias/leakage-like

---

## Matriks 2 — Kumpulan Masalah (Observasi → Risiko → Hipotesis → Saran)

Observasi (Problem)	Risiko jika tidak ditangani	Hipotesis	Saran Penanganan ( $\geq 1$ )
<b>Boundary defect ↔ premium blur</b> (premium kadang punya cacat ringan; defect sangat beragam)	Confusion tinggi defect/premium; evaluasi jadi “terlihat buruk” atau tidak stabil; model tampak gagal padahal label ambigu	Banyak error model akan terkonsentrasi di defect↔premium dan pada sampel borderline	(1) Fokus metrik <b>per-class recall</b> + confusion matrix; (2) Error analysis terstruktur: kumpulkan borderline; (3) Pertimbangkan label smoothing / calibration; (4) Dokumentasikan “upper bound” realistis
<b>Defect sangat heterogen</b> (kumpulan failure modes)	Model underfit defect ekstrem atau overfit defect ringan; recall defect jadi bottleneck	Variansi intra-class defect membuat representasi sulit dipelajari oleh model kecil	(1) Backbone lebih kuat (ResNet/EfficientNet); (2) Augmentasi terkontrol (rot/zoom/brightness); (3) Tambah regularisasi (dropout/weight decay/early stopping); (4) Curate “hard examples” untuk diagnosis
<b>Near-duplicate lintas kelas signifikan</b> (pHash $\leq 4$ , 33 cross-class)	Leakage-like effect: validasi bias/overoptimistic atau justru label conflict; split random menjadi tidak kredibel	Banyak pasangan mirip secara visual tetapi beda label → split acak bisa menaruh “kembar” di train vs val/test	(1) <b>Group-aware split</b> berbasis pHash bucket/cluster; (2) Buat “duplicate group id” lalu split by group; (3) Laporkan evaluasi dengan & tanpa group-split untuk transparansi
<b>Exact duplicates intra-class</b> (11 grup, terutama peaberry)	Redundancy membuat model “menghafal”; effective sample size turun; training bias terhadap sampel tertentu	Duplicates berasal dari copy file-level (xx & xx(1))	(1) Deduplicate exact (MD5/pHash=0); (2) Simpan log file yang di-drop; (3) Jika ingin tetap, pastikan duplicates tidak tersebar lintas split
<b>Background putih &amp; posisi centered konsisten</b>	Shortcut learning: model belajar bayangan/lighting/background, bukan biji	Model akan overfit cepat dan generalisasi buruk jika kondisi background berubah	(1) Color jitter ringan + brightness/contrast; (2) Random rotation/translation kecil; (3) (Opsional) Random erasing/cutout ringan; (4) Pastikan pipeline normalisasi konsisten

Observasi (Problem)	Risiko jika tidak ditangani	Hipotesis	Saran Penanganan ( $\geq 1$ )
<b>Color bukan sinyal utama (overlap besar)</b>	Jika terlalu mengandalkan warna (augmentasi ekstrem / normalisasi buruk) $\rightarrow$ performa turun	Warna hanya secondary cue; shape+texture lebih dominan	(1) Pertahankan RGB tapi augmentasi warna <b>ringan</b> ; (2) Jangan aggressive hue/saturation; (3) Uji baseline grayscale sebagai pembanding (bukan default)
<b>Tekstur proxy overlap (edge density tidak memisahkan kelas)</b>	Feature engineering berbasis edge/variance akan mengecewakan; interpretasi tekstur “salah arah”	Yang penting bukan jumlah edge, tapi <b>spatial pattern</b> (retak internal vs outline)	(1) Jangan fokus handcrafted texture; (2) Pakai CNN dengan receptive field bertahap; (3) Tambahkan analisis misclass + Grad-CAM nanti untuk melihat pattern
<b>Dataset kecil-menengah (1211)</b>	Overfitting cepat, performa tidak stabil antar run	Model akan sensitif terhadap split/seed	(1) Augmentasi + regularisasi; (2) Gunakan stratified/group-aware split; (3) Laporkan $\text{mean} \pm \text{std}$ dari beberapa seed; (4) (Opsional) transfer learning backbone pretrained