BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan implementasi dan hasil pengujian sistem deteksi malware Android menggunakan *Large Language Model* (Google Gemini 2.5 Pro), diperoleh kesimpulan yang menjawab rumusan masalah sebagai berikut:

1. Kemampuan Analisis Semantik LLM: Penelitian ini membuktikan bahwa LLM dapat dimanfaatkan secara efektif sebagai mesin analisis statis. Model mampu memahami konteks "niat jahat" (*malicious intent*) dengan menghubungkan izin sensitif dan potongan kode API, sehingga berhasil mendeteksi ancaman yang menggunakan teknik *obfuscation* yang biasanya sulit dikenali oleh metode berbasis *signature*.
2. Metodologi Pra-pemrosesan: Metode pra-pemrosesan yang paling optimal untuk analisis berbasis LLM ditemukan dengan menggunakan kombinasi Apktool untuk mengekstrak *AndroidManifest.xml* dan Jadx untuk mendekompilasi *bytecode* menjadi format *Pseudo-Java*. Format Java ini terbukti memberikan konteks keterbacaan yang lebih baik bagi LLM dibandingkan format Smali, sehingga meningkatkan akurasi interpretasi logika program.
3. Strategi Prompt Engineering: Penerapan teknik *Prompt Engineering* terstruktur yang mencakup komponen *Persona*, *Defense*, *Context*, dan *Instruction* berhasil mengarahkan LLM untuk melakukan penalaran bertahap (*Chain-of-Thought*). Strategi ini efektif meminimalisir halusinasi AI dan menghasilkan klasifikasi ancaman yang disertai justifikasi logis.
4. Kinerja Sistem: Berdasarkan pengujian pada dataset campuran, sistem menunjukkan karakteristik sensitivitas tinggi (*High Sensitivity*) dengan nilai Recall mencapai 100%, yang berarti tidak ada sampel malware yang lolos dari deteksi. Namun, sistem masih memiliki keterbatasan pada aspek presisi dengan nilai Akurasi 50%, yang disebabkan oleh kecenderungan model yang konservatif (*paranoid*) dalam menilai izin berisiko pada

aplikasi aman, sehingga menghasilkan tingkat *False Positive* yang cukup tinggi

5.2 Saran

1. Integrasi Analisis Hibrida: Disarankan menggabungkan analisis statis LLM dengan analisis dinamis (Sandbox). LLM bertugas mendeteksi potensi niat, sementara Sandbox memverifikasi apakah niat tersebut benar-benar dieksekusi, guna mengurangi *False Positive*.
2. Fine-Tuning Dataset: Diperlukan pelatihan ulang (fine-tuning) model dengan dataset aplikasi aman yang lebih besar agar model dapat belajar membedakan penggunaan izin yang sah vs berbahaya.