

BAB I PENDAHULUAN

1.1 Latar Belakang

Software yang dikembangkan dan dibuat oleh para pengembang sebagian besar digunakan oleh perusahaan atau instansi pemerintah. Pembuatan *software* tersebut bertujuan agar proses bisnis pada perusahaan atau instansi pemerintahan berjalan dengan efisien, cepat dan akurat. (Chang, Mu, & Zhang, 2011; Wahono, Suryana, & Ahmad, 2014). Salah satu ukuran kualitas perangkat lunak adalah jumlah *defect* yang ada pada produk yang dihasilkan (Turhan & Bener, 2007, p. 224). Perangkat lunak dikatakan berkualitas tinggi jika tidak ditemukan defect selama pemeriksaan dan pengujian (McDonald, Musson, & Smith, 2008, pp. 4-6).

Software Defect merupakan suatu kesalahan aktual yang dapat ditemukan pada perangkat lunak. Keberadaannya pada kode program dapat mengakibatkan perangkat lunak tidak berfungsi sebagaimana yang diharapkan. Cacat seringkali tidak terdeteksi pada saat debugging atau testing (Devi, C, A, Surendiran, B. & Kannammal, K., E. 2011). Oleh karena itu Preprocessing dapat dijadikan tahap awal yang harus dilakukan dengan tujuan Preprocessing untuk mempersiapkan data mentah sebelum dilakukan proses lain. *Preprocessing* data akan dilakukan dengan langkah mengeliminasi data yang tidak sesuai dan bisa juga dengan cara mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem. *Preprocessing* juga dilakukan untuk mendapatkan hasil yang bisa lebih akurat dan membuat nilai menjadi lebih kecil tanpa merubah informasi yang ada didalamnya. [K. Saputra and A. P. U. Siahaan, 2007].

Pada beberapa dataset terdapat rentang nilai yang berbeda disetiap atribut. Perbedaan rentang nilai pada setiap atribut menyebabkan tidak berfungsinya atribut yang memiliki nilai jauh lebih kecil dibandingkan dengan atribut-atribut lainnya. Oleh karena itu, diperlukan adanya Optimasi data untuk pencarian nilai minimum ataupun nilai maksimum dari data yang berada dalam satu variabel atau beberapa variable, (zukhri,

2014) untuk menormalisasikan data dengan menggunakan transformasi dapat menyamakan rentang nilai pada setiap atribut dengan skala tertentu. Agar dapat menghasilkan keakurasian pada software defect prediction yang lebih baik. *Transformasi* data dengan normalisasi dapat dilakukan dengan beberapa cara, yaitu *min-max normalization*, *z-score normalization*, *decimal scaling*, *sigmoid*, dan *softmax*.

Sampai saat ini *Software defect prediction* memutuskan perhatian pada: 1). Ketidakseimbangan kelas pada *software defect* 2). *Estimasi* jumlah pada *software defect* 3). Klasifikasi pada *software defect* terutama pada penentuan *defect* dan *no defects* (Song, Jia, Shepperd, Ying, & Liu, 2011). Beberapa teknik yang sering digunakan dalam data mining ialah *klasterisasi*, *asosiasi*, *estimasi* dan *klasifikasi*. (Y. Pristyanto, N. A. Setiawan, and I. Ardiyanto, 2017).

Klasifikasi merupakan salah satu bagian dari data mining yang bertujuan untuk memprediksi label kategori benda yang tidak diketahui sebelumnya berdasarkan atribut dan fitur (Ngai et al, 2011). Salah satu analisis klasifikasi yang dapat digunakan adalah *K-Nearest Neighbor* (KNN). *K-Nearest Neighbor* sendiri merupakan metode klasifikasi yang paling dasar dan sederhana (Parvin, Alizadeh, and Bidgoli (Parvin et al). Meskipun sederhana, *K-Nearest Neighbor* dianggap sebagai salah satu dari sepuluh algoritma klasifikasi data mining yang terbaik (Wu et al. 2008), bahkan dalam beberapa kasus (Kuramochi and Karypis 2005) mengemukakan bahwa *K-Nearest Neighbor* mengungguli performa *Support Vector Machine* (SVM) yang memiliki skema klasifikasi yang lebih canggih.

Siringoringo (2017) dalam penelitiannya menyatakan bahwa integrasi metode resampling pada *K-Nearest Neighbor* efektif dalam menangani ketidak seimbangan kelas dan meningkatkan nilai akurasi, sensitivitas, dan spesifisitas pada hasil klasifikasi.

1.2 Rumusan Masalah

1. Bagaimana untuk menangani ketidak seimbangan kelas pada *software defect prediction* ?
2. Berapa nilai tingkat akurasi *K-Nearest Neighbor* setelah ditangani dengan *Preprocessing* data ?

1.3 Tujuan

1. Menerapkan metode *K-Nearest Neighbor* untuk menangani ketidakseimbangan kelas pada dataset *software defect prediction*.
2. Meningkatkan tingkat akurasi *K-Nearest Neighbor* dengan penggabungan metode *z-transformasi* (normalisasi data).

1.4 Batasan Masalah

Berdasarkan permasalahan diatas akan diberikan batasan masalah sebagai berikut:

1. Pada penelitian ini hanya mengimplementasikan Algoritman *K-Nearest Neighbor* dan *Preprocessing*.
2. Pada skripsi ini hanya menggunakan dataset *software defect prediction*.

1.5 Manfaat Penelitian

Dalam penelitian ini diharapkan tingkat akurasi yang dihasilkan dengan metode *K-Nearest Neighbor* untuk tingkat akurasi *software defect* lebih baik, sehingga penelitian ini dapat digunakan sebagai alternative acuan untuk melihat tingkat *software defect* yang telah dibuat.

1.6 Sistematis Penulisan

Skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang berhubungan dengan ketidakseimbangan serta unsur-unsurnya, dan juga metode *K-Nearest Neighbor*.

3. BAB III METODOLOGI DAN PERANCANGAN SISTEM

Bab ini menjelaskan mengenai metodologi serta perancangan sistem untuk prediksi tingkat akurasi dengan metode *K-Nearest Neighbor*, dan Proses perhitungannya.

4. BAB IV PEMBAHASAN

Bab ini berisi hasil implementasi ke perangkat lunak, ujicoba sistem, dan analisa hasil.

5. BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan berdasarkan penelitian yang telah dilakukan, serta saran-saran untuk pengem