

INTISARI

Penelitian ini dilatarbelakangi oleh kebutuhan proses sortasi buah kopi pascapanen yang efisien dan objektif guna meningkatkan mutu produk. Proses sortasi manual sangat rentan terhadap kesalahan manusia (*human error*) akibat kelelahan fisik, sedangkan sistem otomasi terdahulu masih terbatas pada jumlah kategori dan penggunaan jalur pemrosesan tunggal yang kurang efisien. Penelitian ini bertujuan merancang bangun sistem pemilah kematangan buah kopi berbasis Arduino dengan arsitektur master-slave untuk meningkatkan kapasitas, konsistensi, dan keberhasilan pemilahan. Sistem ini dibangun dengan tiga jalur pemrosesan paralel untuk mengklasifikasikan buah kopi ke dalam kategori matang, belum matang, dan busuk. Subsistem akuisisi data memanfaatkan tiga sensor warna TCS3200 dan enam sensor *infrared* E18-D80NK yang dikelola oleh tiga Arduino Uno sebagai *slave*. Arduino Mega 2560 bertindak sebagai *master* untuk mengoordinasikan aktuator *solenoid* dan motor servo, serta merekapitulasi data dari tiga sensor *load cell* pembaca berat. Berdasarkan hasil pengujian, rancang bangun ini menunjukkan tingkat keberhasilan yang sangat baik dalam mengotomatisasi proses sortasi. Arsitektur *master-slave* terbukti mampu mengeksekusi pemrosesan data paralel dari ketiga jalur secara bersamaan tanpa mengalami tabrakan data (*data collision*). Hasil pengujian menunjukkan sistem mampu mengklasifikasikan buah kopi dengan tingkat keberhasilan di atas 99%. Proses pemilahan tersebut tersinkronisasi secara akurat dengan sistem penimbangan *load cell*, di mana sistem mampu menimbang massa dengan rata-rata tingkat kesalahan (*error rate*) penimbangan individu di bawah 5%. Selain itu, kesalahan total sortasi tercatat sangat presisi, yakni kurang dari 1% (0,10% pada percobaan pertama dengan akumulasi 98,0 gram dan 0,68% pada percobaan kedua dengan akumulasi 87,4 gram). Secara keseluruhan, sistem otomasi ini terbukti andal, responsif, dan berhasil mereduksi kesalahan penyortiran secara efektif.

Kata Kunci: Pemilah kopi, Arduino, *Master-slave*, Sensor TCS3200, *Load cell*.

ABSTRACT

This research was motivated by the need for an efficient and objective post-harvest coffee cherry sorting process to improve product quality. Manual sorting is highly susceptible to human error due to physical fatigue, while previous automated systems were limited in the number of categories they could handle and relied on a single, less-than-efficient processing line. This study aims to design and build an Arduino-based coffee cherry ripeness sorting system with a master-slave architecture to improve sorting capacity, consistency, and success rates. The system is built with three parallel processing lines to classify coffee cherries into ripe, unripe, and rotten categories. The data acquisition subsystem utilizes three TCS3200 color sensors and six E18-D80NK infrared sensors managed by three Arduino Uno boards acting as slaves. An Arduino Mega 2560 acts as the master to coordinate the solenoid actuators and servo motors, as well as to aggregate data from three load cell sensors that measure weight. Based on the test results, this design demonstrated an excellent success rate in automating the sorting process. The master-slave architecture proved capable of executing parallel data processing across all three lanes simultaneously without experiencing data collisions. The test results showed that the system was able to classify coffee cherries with a success rate exceeding 99%. The sorting process is accurately synchronized with the load cell weighing system, where the system is capable of weighing mass with an average individual weighing error rate of less than 5%. In addition, the total sorting error was recorded as highly precise, at less than 1% (0.10% in the first experiment with an accumulation of 98.0 grams and 0.68% in the second experiment with an accumulation of 87.4 grams). Overall, this automation system has proven to be reliable, responsive, and effective at reducing sorting errors.

Keywords: Coffee sorter, Arduino, Master-slave, TCS3200 sensor, Load cell.