**BAB II KAJIAN PUSTAKA**

**A. Kajian Teori**

(Bagian 1-3 tetap, karena bersifat umum)

1. **Konsep Media Pembelajaran**
2. **Penelitian dan Pengembangan (R&D)**
3. **Model Pengembangan ADDIE**

(Bagian 4-8 akan diperluas dan ditambahkan poin-poin penting)

1. **Robotika**
   * 4.3 Autonomous Mobile Robot (AMR)
     + ... (seperti sebelumnya)
     + 4.3.4 Sistem Sensor pada AMR (Pendahuluan)
2. **Swerve Drive**
   * 5.2 Komponen Swerve Drive
     + **Motor Penggerak Roda (BLDC):**
       - Prinsip kerja BLDC.
       - Kelebihan: Efisiensi, torsi, *low maintenance*.
       - Kekurangan: Membutuhkan *controller* kompleks.
       - **Penting:** Kontrol kecepatan dan torsi BLDC (biasanya dengan *field-oriented control* - FOC). Anda *tidak perlu* menjelaskan FOC secara detail, cukup sebutkan.
     + **Motor Penggerak Kemudi (PG45 Brushed):**
       - Prinsip kerja *brushed motor*.
       - Kelebihan dan kekurangan.
       - **Penting:** Kontrol posisi *brushed motor* (biasanya dengan PID).
     + **AS5600 (Magnetic Rotary Encoder):**
       - Prinsip kerja *magnetic rotary encoder*.
       - *Absolute* vs. *Incremental encoder*.
       - Spesifikasi AS5600 (resolusi, akurasi, interface).
       - **Penting:** Bagaimana output AS5600 (sudut absolut) digunakan dalam kontrol *steering*.
     + Sistem Transmisi dan Struktur Mekanik.
   * 5.3 Kinematika Swerve Drive
     + **Forward Kinematics:**
       - Persamaan (matriks).
       - Input: Kecepatan dan sudut setiap roda.
       - Output: Kecepatan linear (Vx, Vy) dan kecepatan angular (ω) robot.
     + **Inverse Kinematics:**
       - Persamaan (matriks) - *ini yang paling penting*.
       - Input: Kecepatan linear (Vx, Vy) dan kecepatan angular (ω) robot yang *diinginkan*.
       - Output: Kecepatan dan sudut *setiap roda* yang diperlukan.
       - **Penting:** Jelaskan *bagaimana* persamaan ini diturunkan (geometri). Gunakan ilustrasi yang jelas.
       - **Penting:** Bahas *singularitas* kinematika (jika ada). Singularitas terjadi ketika konfigurasi roda tertentu tidak dapat menghasilkan gerakan robot tertentu.
     + **Parameter-parameter:** Jelaskan semua parameter yang terlibat (lebar robot, panjang robot, jarak antar roda, dll.).
     + **Contoh Perhitungan:** Berikan contoh perhitungan *inverse kinematics* yang sederhana.
     + **Penting:** Hubungkan dengan *wheel odometry* dan AS5600. Bagaimana data dari sensor-sensor ini digunakan dalam perhitungan kinematika?
3. **Navigasi Robot**
   * 6.2 Algoritma Path Planning
     + **Pure Pursuit:**
       - Prinsip kerja (ilustrasi).
       - Parameter *lookahead distance*.
       - Kelebihan dan kekurangan.
       - **Penting:** Bagaimana output Pure Pursuit (curvature, target velocity) digunakan sebagai input untuk *inverse kinematics* *swerve drive*. Ini adalah *jembatan* antara *path planning* dan *low-level control*.
     + AI Pathfinding (A\*, D\*, RRT, dll.) - sebutkan saja.
   * 6.3 Path Tracking
     + **PID Controller:**  
       \* **Penting:** Karena Anda akan menggunakan PID, jelaskan *prinsip kerja PID controller* secara detail.
       - Jelaskan apa itu *proportional (P)*, *integral (I)*, dan *derivative (D)* control.
       - Jelaskan bagaimana ketiga komponen ini bekerja sama untuk mengoreksi error.
       - Jelaskan *efek* dari masing-masing parameter (Kp, Ki, Kd) terhadap respons sistem (overshoot, settling time, steady-state error).
       - Jelaskan *tuning* PID (bagaimana cara memilih nilai Kp, Ki, Kd yang optimal). Sebutkan beberapa metode *tuning* (misalnya, Ziegler-Nichols, trial and error).
       - **Penting:** Di mana PID digunakan dalam sistem Anda? Kemungkinan:  
         \* Kontrol kecepatan BLDC motor.  
         \* Kontrol posisi *steering motor*.  
         \* *Path tracking* (mengikuti jalur yang dihasilkan oleh Pure Pursuit). Jika Anda menggunakan PID untuk *path tracking*, jelaskan *bagaimana* error dihitung (misalnya, *cross-track error*, *heading error*).  
         \* Anda *tidak perlu* menjelaskan algoritma *path tracking* yang kompleks (seperti Model Predictive Control - MPC). Cukup fokus pada PID.
4. **Sensor untuk Navigasi AMR**
   * 7.1 Wheel Odometry
     + Prinsip kerja.
     + *Wheel encoder* (BLDC internal atau eksternal).
     + Kelebihan dan kekurangan (akumulasi error, slip).
     + **Penting:** Rumus perhitungan posisi (x, y) dan orientasi (θ) robot berdasarkan data *encoder*.
   * 7.2 Inertial Measurement Unit (IMU)
     + Prinsip kerja (accelerometer, gyroscope, magnetometer).
     + Jenis-jenis IMU (3-DOF, 6-DOF, 9-DOF).
     + Parameter penting (akurasi, drift, sampling rate).
     + **Penting:** Bagaimana data dari accelerometer dan gyroscope digunakan untuk memperkirakan kecepatan dan orientasi robot.
     + **Penting:** Masalah *drift* dan cara mengatasinya (sensor fusion, Kalman filter - sebutkan saja).
5. **Simulator (Unity)**
   * 8.1 Definisi Simulator
   * 8.2 Jenis-jenis Simulator
   * 8.3 Manfaat Simulator dalam Pendidikan Teknik
   * 8.4 Contoh Simulator Robotika (V-REP, Gazebo, Webots, CoppeliaSim)
   * 8.5 Kriteria Simulator yang Baik
   * **8.6 Unity sebagai Platform Simulator:**
     + **Penting:** Jelaskan *mengapa* Anda memilih Unity.
     + **Penting:** Sebutkan *fitur-fitur Unity* yang relevan untuk simulasi robotika:
       - **Physics Engine:** Mensimulasikan gaya, torsi, tumbukan, gravitasi, dll. (Penting untuk dinamika robot).
       - **3D Graphics:** Visualisasi model robot dan lingkungan.
       - **Scripting (C#):** Memprogram perilaku robot, algoritma navigasi, kontrol PID, dll.
       - **Asset Store:** Aset siap pakai (model 3D, script, dll.).
       - **UI System:** Membuat antarmuka pengguna untuk simulator.
       - **Input System:** Menangani input dari pengguna (keyboard, mouse, joystick).
     + **Penting (Jika Relevan):** Jika Anda menggunakan *package* atau *plugin* Unity tertentu (misalnya, *ROS#*, *URDF Importer*, *ML-Agents Toolkit*), sebutkan dan jelaskan secara singkat.

**B. Penelitian yang Relevan:**

* ... (seperti sebelumnya)

**Poin-poin Tambahan (Opsional, tapi sangat disarankan):**

* **Diagram Blok Sistem:** Buat diagram blok yang menunjukkan *seluruh sistem* Anda, termasuk:
  + Robot *swerve drive* (dengan motor, *encoder*, IMU).
  + *Controller* (PID).
  + Algoritma navigasi (Pure Pursuit).
  + Simulator Unity.
  + Input dan output dari setiap komponen.
  + Ini akan memberikan gambaran yang jelas tentang arsitektur sistem Anda.
* **Flowchart Algoritma:** Buat *flowchart* untuk:
  + Algoritma Pure Pursuit.
  + Algoritma kontrol PID (jika Anda menggunakannya untuk *path tracking*).
  + Logika utama simulator (bagaimana data mengalir dari input pengguna ke model robot, ke sensor virtual, ke *controller*, dan kembali ke model robot).

Dengan menambahkan poin-poin ini, kajian teori Anda akan menjadi sangat lengkap dan memberikan dasar yang kuat untuk pengembangan simulator Anda.