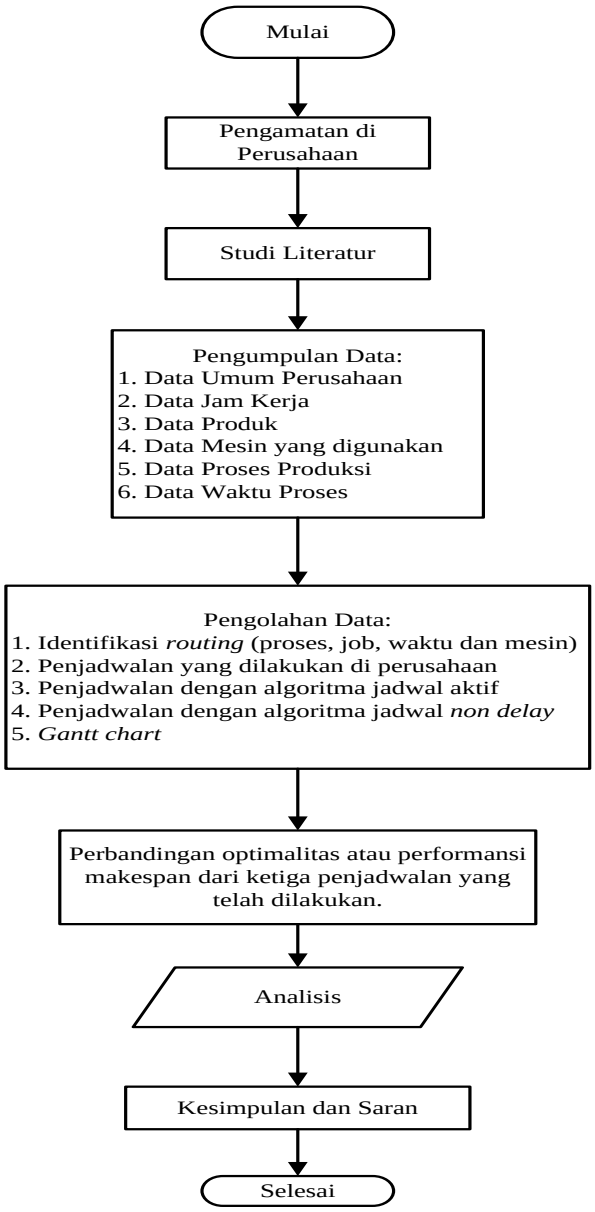
Bab 3
Metodologi Penelitian

3.1. *Flowchart* Pemecahan Masalah

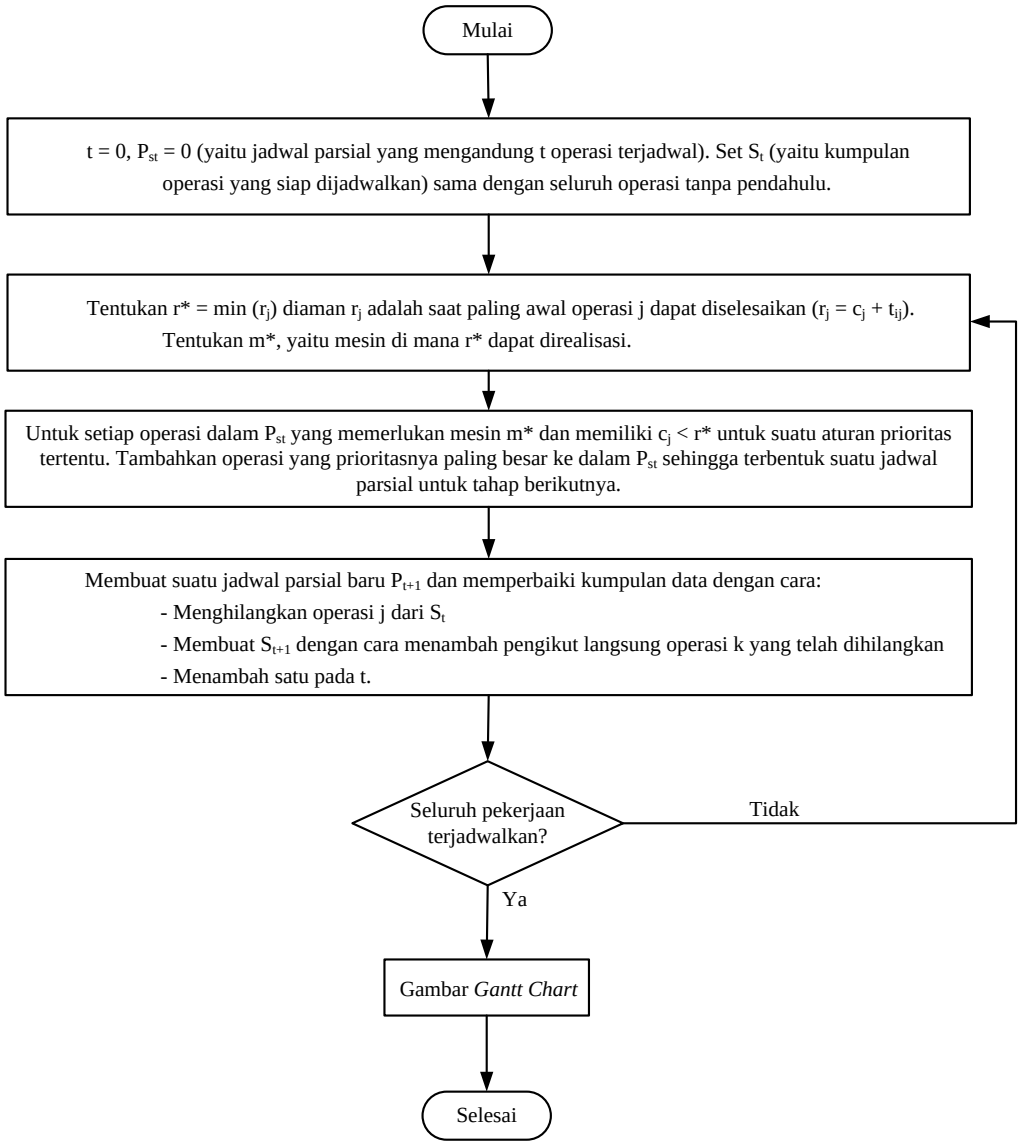
3.1.1. *Flowchart* Penelitian

Berikut ini adalah *flowchart* atau metodologi penelitian yang dilakukan dalam melakukan penelitian:



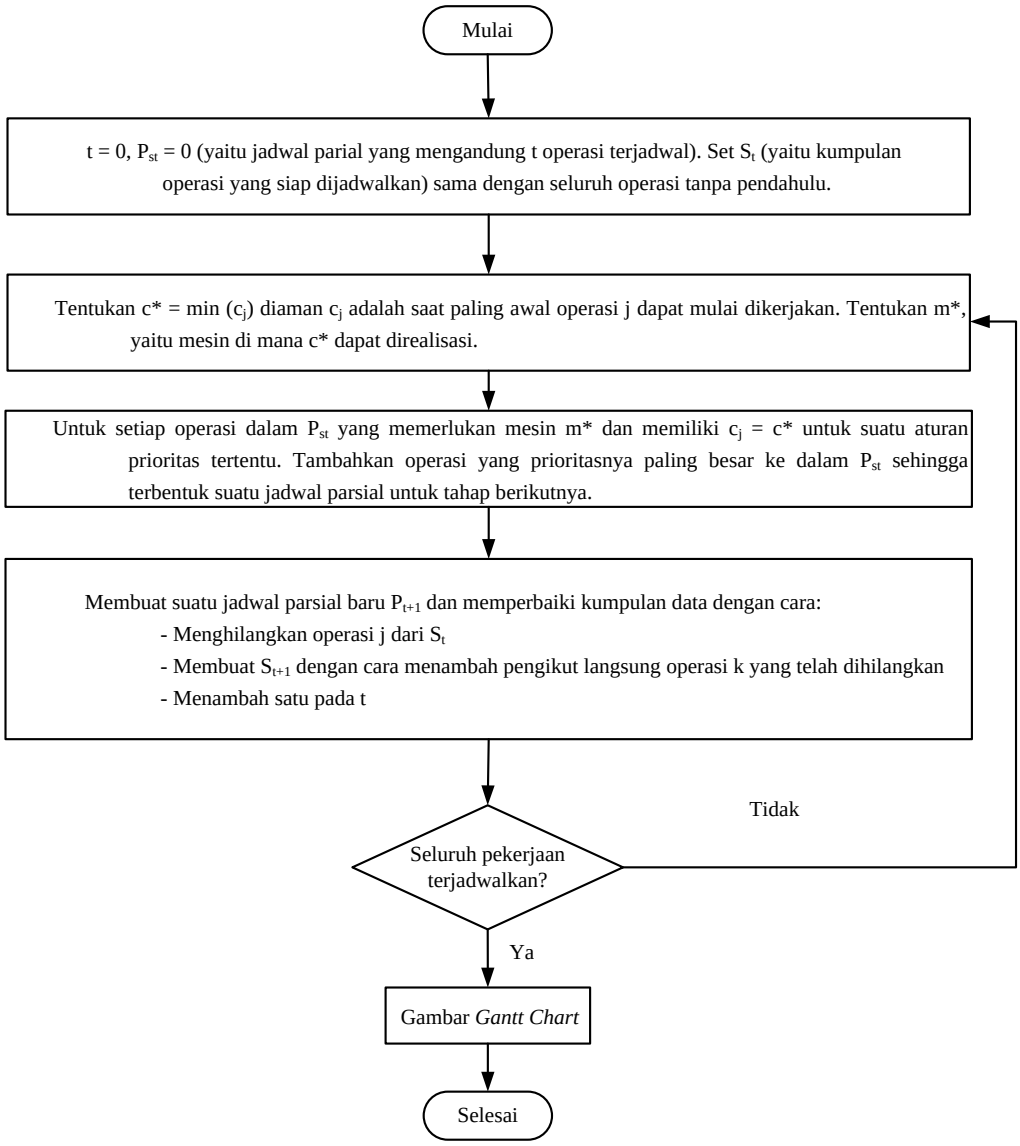
Gambar 3.1. *Flowchart* Pemecahan Masalah

3.1.2. *Flowchart* Metode Penjadwalan *Priority Dispatching* Algoritma Jadwal Aktif Berdasarkan SPT (*Shortest Processing Time*)



Gambar 3.2. *Flowchart* Metode Penjadwalan *Priority Dispatching* Algoritma Jadwal Aktif Berdasarkan SPT (*Shortest Processing Time*)

3.1.3. *Flowchart* Metode Penjadwalan *Priority Dispatching* Algoritma Jadwal *Non Delay*



Gambar 3.2. *Flowchart* Metode Penjadwalan *Priority Dispatching* Algoritma Jadwal *Non Delay*

3.2. Langkah-langkah Pemecahan Masalah

3.2.1. Langkah-langkah Pemecahan Masalah Penelitian

Untuk mencapai tujuan penelitian yang telah ditentukan diperlukan adanya tahapan-tahapan kegiatan penelitian yang disusun secara terstruktur dan sistematis. Untuk itu pada penelitian ini dilakukan tahapan kegiatan sebagai berikut:

1. Pengamatan di perusahaan, pada tahap ini peneliti melakukan penelitian langsung di perusahaan PT. SINAR SAKTI MATRA NUSANTARA, yaitu mempelajari dan memahami tentang proses pembuatan produk *Ass* Mesin Jahit dan Baut *Bearing* yang berhubungan dengan penelitian. Serta melakukan pengidentifikasian dan perumusan masalah, penentuan tujuan penelitian dan ruang lingkup penelitian.
2. Melakukan studi literatur yang mengacu dari berbagai sumber, baik dari buku maupun tugas akhir yang dijadikan referensi untuk memperoleh data dan teori-teori yang dibutuhkan untuk mendukung dalam melakukan penelitian.
3. Melakukan pengumpulan data yang berhubungan dengan masalah penjadwalan ini. Adapun data-data yang diperlukan yaitu sebagai berikut:
    - Data umum perusahaan.
    - Data jam kerja.
    - Data produk yang dibuat pada periode tertentu sesuai dengan *order* .
    - Data mesin-mesin yang dipakai.
    - Data proses produksi.
    - Data waktu proses.
4. Melakukan pengolahan data terhadap data yang telah dikumpulkan, dengan langkah-langkah dari metode ini adalah sebagai berikut:
    - Identifikasi *routing* (proses, *job*, waktu dan mesin).
    - Penjadwalan awal yang dilakukan oleh perusahaan.
    - Penjadwalan pekerjaan (*job*) mesin dengan metode algoritma jadwal aktif. Metode algoritma jadwal aktif adalah kumpulan jadwal *feasible* dimana tidak satupun operasi dapat dipindahkan lebih awal tanpa menunda operasi lain dengan aturan prioritas SPT (*Short Processing*

Time) yaitu prioritas diberikan kepada pekerjaan yang memiliki waktu paling singkat dijadwalkan pada urutan pertama, kemudian diikuti *job* yang terbesar.

- Penjadwalan pekerjaan (*job*) mesin dengan metode algoritma jadwal *non delay*. Metode algoritma jadwal *non delay* adalah kumpulan jadwal *feasible* dimana tidak satupun mesin dibiarkan menganggur jika pada saat yang sama terdapat operasi yang memerlukan mesin tersebut. Jika ada mesin yang digunakan secara bersama-sama maka aturan yang digunakan adalah aturan SPT.
- Menggambar *Gantt Chart*. *Gantt Chart* (*Bar Chart*) terdiri dari dua koordinat aksis dimana satu aksis mempresentasikan waktu yang telah dilalui dan aksis lainnya mempresentasikan pekerjaan atau aktivitas yang dilakukan. Pekerjaan ini dinyatakan dalam bentuk batang.

5. Melakukan analisis secara deskriptif terhadap hasil dari pengolahan data yang telah dilakukan pada penjadwlana proses pembuatan produk dan yaitu penjadwalan yang dilakukan perusahaan, penjadwalan dengan algoritma jadwal aktifdengan aturan prioritas SPT dan penjadwalan dengan algoritma jadwal *non delay*.
6. Menyimpulkan mengenai penelitian yang telah dilakukan yaitu berisi rangkuman dari pengolahan data dan analisis, serta berisikan saran penelitian bagi perusahaan.

3.2.2. Langkah-langkah Metode Penjadwalan *Priority Dispatching* Algoritma Jadwal Aktif Berdasarkan SPT (*Shortest Processing Time*)

1. Mulai pada t = 0, P_{st} = 0 (yaitu jadwal parsial yang mengandung t operasi terjadwal). Set S_t (yaitu kumpulan operasi yang siap dijadwalkan) sama dengan seluruh operasi tanpa pendahulu.
2. Tentukan r* = min (r_j) diaman r_j adalah saat paling awal operasi j dapat diselesaikan ($r_j = c_j + t_{ij}$). Tentukan m*, yaitu mesin di mana r* dapat direalisasi.
3. Untuk setiap operasi dalam P_{st} yang memerlukan mesin m* dan memiliki $c_j <$ r* untuk suatu aturan prioritas tertentu. Tambahkan operasi yang prioritasnya paling besar ke dalam P_{st} sehingga terbentuk suatu jadwal parsial untuk tahap berikutnya.
4. Membuat suatu jadwal parsial baru P_{t+1} dan memperbaiki kumpulan data dengan cara:
   - Menghilangkan operasi j dar S_t
   - Membuat S_{t+1} dengan cara menambah pengikut langsung operasi k yang telah dihilangkan
   - Menambah satu pada t.
5. Kembali ke langkah 2 sampai seluruh pekerjaan terjadwalkan.
6. Menggambar *Gantt chart.*

3.2.3. Langkah-langkah Metode Penjadwalan *Priority Dispatching* Algoritma Jadwal *Non Delay*

1. t = 0, P_{st} = 0 (yaitu jadwal parial yang mengandung t operasi terjadwal). Set S_t (yaitu kumpulan operasi yang siap dijadwalkan) sama dengan seluruh operasi tanpa pendahulu.
2. Tentukan c* = min (c_j) diaman c_j adalah saat paling awal operasi j dapat mulai dikerjakan. Tentukan m*, yaitu mesin di mana c* dapat direalisasi.
3. Untuk setiap operasi dalam P_{st} yang memerlukan mesin m* dan memiliki c_j = c* untuk suatu aturan prioritas tertentu. Tambahkan operasi yang prioritasnya paling besar ke dalam P_{st} sehingga terbentuk suatu jadwal parsial untuk tahap berikutnya.
4. Membuat suatu jadwal parsial baru P_{t+1} dan memperbaiki kumpulan data dengan cara:
   - Menghilangkan operasi j dar S_t
   - Membuat S_{t+1} dengan cara menambah pengikut langsung operasi k yang telah dihilangkan
   - Menambah satu pada t.
5. Kembali ke langkah 2 sampai seluruh pekerjaan terjadwalkan.
6. Menggambar *Gantt chart*.