

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

2.1.1 Pengertian Sistem

Dalam bahasa Latin, "sistem" dan "sustema", istilah "sistem" mengacu pada satu set yang terdiri dari komponen atau elemen yang dihubungkan untuk memudahkan aliran materi, energi, atau informasi. Karena istilah "system" banyak digunakan dalam forum, percakapan sehari-hari, dan file ilmiah, artinya beragam. Sistem adalah sekumpulan benda-benda yang saling berhubungan. Menurut (Rima Rizqi Wijayanti, Wada Kaligula Budiargo, 2020), Sistem adalah kumpulan atau himpunan dari elemen, komponen, atau variabel yang saling tergantung, berinteraksi, dan terpadu.

Sistem juga dapat didefinisikan sebagai kumpulan atau grup dari bagian atau komponen apapun, baik fisik maupun non-fisik, yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai suatu tujuan tertentu. Jika ada komponen sistem yang tidak bermanfaat, komponen tersebut tidak termasuk dalam sistem informasi, sehingga seseorang dapat lebih memahami data yang digunakan. (Ferdin, 2020). Menurut Wikipedia Untuk memahami dan mengembangkan suatu sistem, kita harus memahami komponen yang membentuknya. Berikut adalah beberapa hal yang membedakan satu sistem dari yang lain:

1. Batasan (*Boundary*).
Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam system dan mana yang diluar system.
2. Lingkungan (*Environment*).
Segala sesuatu diluar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*Input*).
Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*Output*).
Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*).
Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*), komponen ini berupa subsistem dari sebuah sistem.
6. Penghubung (*Interface*).
Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*Storage*).
Area yang dikuasai untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya.

2.1.2 Pengertian Informasi

Informasi seperti darah yang terus mengalir di dalam tubuh manusia, jadi sangat penting bagi sebuah organisasi. Karena data berasal dari ribuan informasi yang selalu di dengar, penting untuk belajar tentang data sebelum mendefenisikan apa itu informasi. Data adalah fakta atau kenyataan yang menggambarkan apa yang terjadi di sekitar lingkungan. Menurut (Ferdin, 2020) Sistem informasi terdiri dari empat komponen utama: *hardware*, *software*, sumber daya manusia yang telah dilatih, dan infrastruktur.

Informasi adalah hasil pemrosesan data dari setiap komponen sistem menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan orang lain untuk lebih memahami situasi. Oleh karena itu, keakuratan data sangat penting untuk menentukan kualitas data karena gangguan atau kesengajaan dapat mengubah dan merusak keaslian sumber data. kualitas sebuah informasi sangat dipengaruhi oleh 3 hal pokok, yaitu :

1. Akurat
 - 1) Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan.
 - 2) Akurat juga berarti informasi harus jelas dan mencerminkan maksudnya.
2. Tepat waktu
 - 1) Informasi yang datang ke penerima tidak boleh terlambat.
Informasi yang
 - 2) sudah lama atau using tidak akan mempunyai nilai lagi

3. Relevan

Informasi harus mempunyai manfaat bagi yang memakainya. Relevan bagi tiap-tiap orang yang membutuhkannya.

2.1.3 Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Teknologi dan sistem informasi tidak dapat dipisahkan dari manajemen perusahaan atau organisasi (Riyadli et al., 2020). Menurut (Cahyadi Agustin, 2021) Sistem informasi merupakan gabungan di dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan Sumber Daya Manusia (SDM) yang terlatih.

Definisi sistem informasi juga bisa didefinisikan kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (output), guna mencapai sasaran-sasaran perusahaan. Menurut (Rima Rizqi Wijayanti, Wada Kaligula Budiargo, 2020) Sistem informasi adalah sistem dalam suatu organisasi yang mendukung operasi manajemen dan kegiatan strategis serta memberikan laporan yang diperlukan oleh pihak luar.



Gambar 2. 1 Model umum suatu sistem

Pada dasarnya sistem informasi mempunyai tiga kegiatan utama yaitu: menerima data sebagai masukan, kemudian memprosesnya dengan melakukan perhitungan, penggabungan unsur-unsur data dan akhirnya dapat diperoleh informasi yang diperlukan sebagai keluaran. Prinsip tersebut berlaku baik bagi sistem informasi manual maupun sistem informasi modern dengan penggunaan perangkat komputer.

2.1.4 Rancang Bangun

Tahap awal dalam membangun sebuah situs Penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa komponen yang terpisah ke dalam suatu unit yang utuh dan berfungsi disebut rancang bangun (Arnomo, 2023). Analisis diri dan sketsa teknis yang dapat membantu proses pengambilan keputusan dikenal sebagai rancang bangun (Halijah & Arnomo, 2023). Python adalah bahasa pemrograman dan postgresql adalah database yang akan digunakan untuk membangun web ini. Selanjutnya adalah membuat diagram alir, bagan alir, atau aliran data dari situs web. Diagram alir, juga disebut diagram alir, adalah diagram yang menunjukkan struktur hirarki dan isi setiap halaman. Membuat diagram alir pada suatu situs dapat mempermudah pemeliharaan dan pembaharuan isi situs. Hal ini sangat penting untuk dilakukan karena konten situs web harus terus diperbarui untuk menjaga pengunjung tidak bosan.

Proses perancangan program mencakup merencanakan dan mengorganisasi struktur, fungsi, dan alur kerja suatu program komputer sebelum memulai proses penulisan kode. Tujuan dari proses ini adalah untuk membuat rancangan yang baik

dan efisien sebelum masuk ke tahap pengembangan yang lebih mendalam. Berikut adalah beberapa teori dasar perancangan program:

1. Pemahaman Masalah

Sebelum merancang program, pemahaman yang baik tentang masalah yang akan diselesaikan sangat penting. Identifikasi *input*, *proses*, dan *output* yang diperlukan.

2. *Decomposition* (Pemecahan Masalah)

Pemecahan masalah melibatkan membagi masalah besar menjadi masalah-masalah kecil yang lebih mudah dikelola. Ini melibatkan pengidentifikasian modul atau fungsi-fungsi yang dapat bekerja secara mandiri.

3. Algoritma

Algoritma adalah urutan langkah-langkah logis untuk memecahkan suatu masalah. Algoritma harus jelas, terstruktur, dan dapat dipahami dengan mudah.

4. Struktur Data

Struktur data adalah cara menyimpan dan mengorganisasi data dalam program. Pemilihan struktur data yang tepat (seperti *array*, *linked list*, *stack*, *queue*, *dll.*) sangat penting karena dapat mempengaruhi efisiensi program.

5. Modularitas

Modularitas melibatkan pembagian program menjadi modul-modul atau fungsi-fungsi kecil yang memiliki tanggung jawab tertentu. Setiap modul harus mandiri dan dapat digunakan kembali untuk meningkatkan keterbacaan dan pemeliharaan kode.

6. Abstraksi

Abstraksi melibatkan penyederhanaan kompleksitas dengan mengisolasi detail yang tidak perlu dan fokus pada aspek-aspek penting dari suatu masalah. Ini membantu dalam memahami dan merancang program dengan lebih baik.

7. Kebacaan (*Readability*) dan Keterawatan (*Maintainability*)

Kode program harus mudah dibaca oleh orang lain (dan oleh Anda sendiri di masa mendatang). Gunakan nama variabel, fungsi, dan komentar yang deskriptif untuk menjelaskan tujuan dari setiap bagian kode.

8. Pengujian

Rencanakan pengujian program sejak awal perancangan. Identifikasi kasus uji yang mungkin, dan pastikan program dapat mengatasi berbagai situasi yang mungkin terjadi.

9. Pertimbangan Kinerja

Pertimbangkan efisiensi dan kinerja program. Pemilihan algoritma dan struktur data yang tepat dapat mengoptimalkan kinerja program.

2.1.5 UML (*Unified Modelling Language*)

UML pertama kali diperkenalkan pada tahun 1990-an oleh Grady Booch dan Ivar Jacobson serta James Rumbaugh memadukan metode masing-masing yang kemudian dinamakan dengan *Unified Modeling Language (UML)*. UML muncul karena adanya pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML versi 2.0 diterima pada tahun 2003 dan mendefinisikan 14 teknik pembuatan diagram yang digunakan untuk menggambarkan sistem. (Cahyadi Agustin, 2021) Bahasa Unified Modeling (UML) adalah sebuah bahasa yang digunakan dalam sistem pengembangan perangkat lunak yang berbasis OOP (Object Oriented Programming) yang dimaksudkan untuk mendokumentasikan, visualisasi, menspesifikasikan, dan membangun perangkat lunak. Dua kelompok besar termasuk dalam diagram ini: pemodelan struktur dan pemodelan perilaku. Kelompok pertama menggambarkan diagram struktur yang mencakup kelas, objek, paket, *deployment*, komponen, dan diagram gabungan struktur.

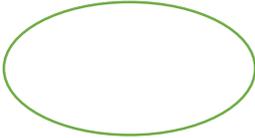
Sedangkan diagram-diagram yang menggambarkan perilaku adalah activity, sequence, communication, interaction overview timing, behavior state machine, protocol state machine, dan use case diagram. Bahasa pemodelan *Unified Modeling Language (UML)* adalah untuk sistem yang berparadigma berorientasi objek. Tujuan pemodelan ini adalah untuk membuat masalah yang kompleks menjadi lebih mudah dipelajari dan dipahami. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML termasuk:

1. *Use Case Diagram*

Secara grafis *Use Case Diagram* menggambarkan secara sederhana fungsi dari sistem dan interaksi antara sistem dengan pengguna serta sistem eksternal. Maksudnya adalah mendeskripsikan siapa yang akan menggunakan sistem dan bagaimana pengguna itu berinteraksi dengan sistem.

Tabel 2. 1 *Use Case Diagram*

NAMA	FUNGSI	NOTASI
Actor	Menggambarkan tokoh atau sistem yang memperoleh keuntungan dan berada di luar dari sistem.	

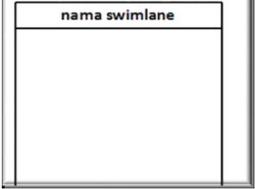
Use Case	Mewakili sebuah bagian dari fungsionalitas sistem dan ditempatkan dalam sistem.	
Subject	Menyatakan lingkup dari subjek	
Association Relationship	Menghubungkan actor untuk berinteraksi dengan use case.	
System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.	

2. Activity Diagram.

Pada bagian ini menjelaskan segala aliran aktivitas baik proses bisnis maupun use case. Diagram ini juga digunakan untuk memodelkan action yang akan dilakukan pada saat sebuah operasi dieksekusi dan salah satu cara untuk memodelkan event-event yang akan terjadi di dalam sebuah use case.

Tabel 2. 2 Activity Diagram

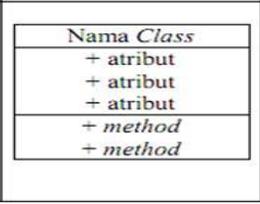
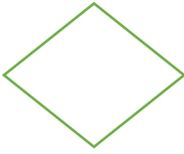
Nama	Deskripsi	Notasi
Status Awal	Sebuah awal dari aktivitas	

	Sistem.	
Aktivitas	Kegiatan-kegiatan apa saja yang dilakukan sistem.	
Percabangan	Menggambarkan apabila ada aktivitas lebih dari satu.	
Penggabungan	Lebih dari satu aktivitas digabungkan menjadi satu.	
Status Akhir	Akhir dari aktivitas sistem yang dilakukan.	Status akhir 
Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.	

3. Class Diagram

Class Diagram adalah sekumpulan class dan interface lengkap dengan kolaborasi dan hubungan antara mereka untuk membangun sebuah sistem. Class Diagram menggambarkan keadaan (atribut / properti) suatu sistem, struktur serta deskripsi class, package, dan objek serta hubungan satu sama lain seperti containment, pewarisan, dan asosiasi.

Tabel 2.3 *Class Diagram*

Nama	Deskripsi	Notasi
Kelas	Himpunan dari objek-objek yang sama.	
Nary Association	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.	
Generalization	Hubungan dimana objek descendent berbagi perilaku dan struktur data dari objek ancestor.	
Collaboration	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem menghasilkan suatu hasil yang	

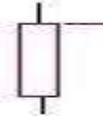
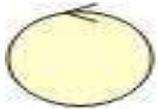
	terukur bagi suatu <i>actor</i> .	
Realization	Operasi yang benar-benar dilakukan oleh objek.	
Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri.	

4. *Sequence Diagram*.

Secara grafis diagram ini menggambarkan bagaimana objek berinteraksi satu sama lain melalui pesan pada eksekusi sebuah *use case* atau operasi. *Sequence Diagram* biasanya digunakan untuk menggambarkan skenario yang dilakukan sebagai respon dari sebuah event untuk menghasilkan sebuah output.

Tabel 2. 4 *Sequence Diagram*

Nama	Deskripsi	Notasi
<i>Actor</i>	Melambangkan seorang user/pengguna	
<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi	

		
<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form	
<i>Controls class</i>	Melambangkan hubungan Antara <i>boundary</i> dengan <i>table</i>	
<i>Entinity class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan	
<i>Message</i>	Spesifikasi komunikasi antara objek yang memuat informasi tentang aktifitas yang terjadi.	

2.2 Tinjauan Teori Khusus

2.2.1 Teknisi

Teknisi mesin adalah orang yang tahu dan bisa merawat, memperbaiki, mengoperasikan, dan mengatasi masalah berbagai mesin dan peralatan mekanik. Mereka biasanya bekerja di bidang yang membutuhkan peralatan mekanik, seperti

manufaktur, konstruksi, industri otomotif, energi, dan bidang lainnya yang membutuhkan peralatan mekanik.

Tugas dan Tanggung Jawab

1. Melakukan pemeliharaan rutin pada mesin dan peralatan produksi.
2. Mengidentifikasi dan memperbaiki gangguan dalam operasi mesin.
3. Melakukan perbaikan darurat jika mesin mengalami masalah saat produksi.
4. Berkoordinasi dengan departemen produksi untuk menjadwalkan pemeliharaan yang tidak mengganggu jadwal produksi.
5. Memeriksa dan mengganti suku cadang yang aus atau rusak.
6. Memastikan peralatan keselamatan berfungsi dengan baik.
7. Mencatat semua perawatan dan perbaikan yang dilakukan pada mesin.
8. Mengikuti pedoman keselamatan dan prosedur operasional standar.

2.2.2 Maintenance Mesin

Maintenance mesin adalah serangkaian tindakan perawatan dan perbaikan yang dilakukan pada mesin atau peralatan industri untuk memastikan kinerjanya tetap optimal dan mencegah kerusakan atau kegagalan yang tidak terduga. Menurut (Prabowo et al., 2020) Perawatan harus dioptimalkan untuk komponen dan sistem peralatan mesin produksi agar kondisi mereka selalu berada dalam kondisi terbaik.

Perawatan mesin sangat penting untuk memperpanjang umur mesin, mengurangi *downtime* produksi, dan meningkatkan efisiensi operasional. Menurut (Maulana, 2002) Pemeliharaan mesin produksi diperlukan untuk menjaga kinerja mesin agar hasil produksi tetap stabil. Dalam perencanaan pemeliharaan terencana,

harus membuat daftar sarana dan kemudian menentukan bagaimana sarana atau aset ini dipelihara (Santoso et al., 2021).

Maintenance mesin mencakup beberapa jenis perawatan, antara lain:

1. *Maintenance Preventif*: Ini melibatkan tindakan-tindakan perawatan terencana yang dilakukan secara rutin untuk mencegah potensi kerusakan atau kegagalan. Contohnya termasuk pemeriksaan berkala, pelumasan, penggantian suku cadang yang aus, dan kalibrasi.
2. *Maintenance Prediktif*: Dalam *maintenance prediktif*, teknologi seperti sensor, pemantauan kecerdasan buatan, dan analisis data digunakan untuk memantau kondisi mesin secara terus-menerus. Data ini digunakan untuk memprediksi waktu kerusakan atau kegagalan sehingga perbaikan dapat dilakukan sebelum terjadi kerusakan serius.
3. *Maintenance Korektif (Reaktif)*: Ini adalah tindakan perbaikan yang diambil setelah mesin mengalami kerusakan atau kegagalan. Meskipun *maintenance* ini diperlukan, tindakan reaktif biasanya lebih mahal dan dapat menyebabkan *downtime* yang signifikan.
4. *Maintenance Terencana (Planned Maintenance)*: *Maintenance* terencana melibatkan perawatan yang dijadwalkan secara teratur berdasarkan pengalaman dan pedoman produsen mesin. Ini termasuk pemeriksaan berkala, penggantian komponen, dan servis rutin.
5. *Maintenance Darurat*: Ini melibatkan tindakan perbaikan mendesak yang harus dilakukan segera setelah mesin mengalami kerusakan yang parah atau kegagalan yang menghentikan proses produksi.

6. *Total Productive Maintenance (TPM)*: *TPM* adalah pendekatan holistik terhadap *maintenance* yang melibatkan semua anggota tim produksi. Tujuan *TPM* adalah mencapai efisiensi optimal melalui perbaikan sistematis, pelibatan karyawan dalam perawatan mesin, dan eliminasi kerugian produksi.
7. Penting untuk mencatat bahwa *maintenance* mesin harus disesuaikan dengan jenis mesin, lingkungan kerja, intensitas penggunaan, dan kebutuhan produksi perusahaan. Oleh karena itu, strategi *maintenance* harus dirancang dengan hati-hati dan dikelola secara efisien untuk memastikan mesin beroperasi dengan optimal tanpa mengalami *downtime* yang tidak perlu.

2.2.3 Mesin

Mesin memfasilitasi pekerjaan manusia dengan mengubah energi. Pemeliharaan dilakukan untuk mempertahankan umur dan kehandalan alat karena jika digunakan secara terus menerus, umur dan kehandalan alat akan menurun (Ninny Siregar & Munthe, 2019).

2.2.4 Mesin SPEA

Mesin SPEA adalah peralatan pengujian elektronik yang digunakan dalam industri manufaktur elektronik, terutama dalam produksi papan sirkuit cetak (PCB) dan perangkat semikonduktor. SPEA adalah singkatan dari "*Società Prodotti Elettronici ed Affini*," yang merupakan nama perusahaan asal Italia yang memproduksi peralatan ini.

Menurut (Chinmoy Ghorai , Student Member, IEEE, Swapan Shakhari , Student Member & and Indrajit Banerjee, Member, 2020) Metode ini menggunakan

solusi berbasis Algoritme Evolusi Kekuatan Pareto (SPEA) untuk menemukan rute komunikasi yang efektif dan meningkatkan persyaratan Kualitas Layanan (QoS). Menemukan rute yang stabil antara sumber dan tujuan target merupakan masalah penting dengan banyak tantangan karena mobilitas VANet yang tinggi dan penyimpangan topologi yang sering terjadi. Mesin SPEA melakukan banyak hal dan digunakan untuk berbagai proses manufaktur elektronik:

1. Uji Fungsi Elektronik (*In-Circuit Testing - ICT*): Mesin SPEA dapat melakukan pengujian fungsi elektronik pada PCB. Ini melibatkan pengukuran dan pengujian sirkuit elektronik pada PCB untuk memastikan bahwa komponen elektronik berfungsi seperti yang diharapkan.
2. Uji Kontinuitas: Mesin SPEA dapat memeriksa kontinuitas jalur listrik pada PCB untuk memastikan tidak ada koneksi yang terputus atau hubungan yang tidak diinginkan.
3. Uji Jangkauan Signal (*Boundary Scan Testing*): Mesin SPEA dapat melakukan uji boundary scan untuk mengidentifikasi kesalahan dalam desain sirkuit terpadu (IC) yang sulit dijangkau oleh teknik pengujian tradisional.
4. Pengujian Pengkabelan: Mesin SPEA dapat memeriksa pengkabelan PCB untuk memastikan bahwa konektor dan kabel terpasang dengan benar dan sesuai dengan spesifikasi.
5. Pengujian Komponen Pasif dan Aktif: Mesin SPEA dapat menguji *resistor, kapasitor, induktor, transistor*, dan komponen elektronik

lainnya pada PCB untuk memastikan nilai-nilai komponen sesuai dengan spesifikasi.

6. Pengujian Kecepatan Sinyal: Dalam perangkat semikonduktor yang membutuhkan pengujian kecepatan sinyal (seperti mikroprosesor atau FPGA), mesin SPEA dapat mengukur kecepatan dan waktu delay sinyal pada PCB.
7. Uji Beban (*Load Testing*): Mesin SPEA dapat menguji performa PCB di bawah beban maksimum untuk memastikan bahwa komponen dan sirkuit dapat menangani beban operasional yang diharapkan.

Mesin SPEA dan peralatan pengujian elektronik serupa sangat penting dalam proses produksi elektronik karena mereka membantu memastikan kualitas produk akhir dan mengidentifikasi kesalahan atau cacat sejak dini dalam proses produksi. Ini membantu mengurangi pemborosan material dan biaya, meningkatkan efisiensi produksi, dan meningkatkan kepuasan pelanggan dengan produk yang dihasilkan. SPEA digunakan dalam perutean PCB protokol perutean terdistribusi yang menghitung penundaan ujung ke ujung untuk total jalur perutean sebelum mengirim paket (Chinmoy Ghorai , Student Member, IEEE, Swapan Shakhari , Student Member & and Indrajit Banerjee, Member, 2020).



Gambar 2. 2 Mesin SPEA

2.2.5 Mesin Laser *Marking*

Mesin penanda laser adalah alat yang menggunakan sinar laser untuk menempatkan marka atau tanda pada berbagai jenis material. Mereka bekerja dengan menggunakan cahaya yang difokuskan pada sebuah area atau titik (Dikky Antonius, 2020). Mesin ini memiliki beberapa komponen utama dan berbagai teknologi untuk mencapai hasil marking yang akurat dan berkualitas. Berbagai produk elektronik, seperti amplifier, perangkat audio, dan peralatan musik, memiliki PCB yang dilabeli dengan informasi penting, seperti kode produk, nomor seri, tanggal produksi, atau informasi lainnya yang diperlukan selama proses produksi, pengujian, dan perawatan (Setiowati et al., 2023).

Berikut adalah komponen dan cara kerja umum mesin laser marking. Ditinjau dari metodenya, mesin marking dapat dibedakan menjadi beberapa macam, antara lain metode ketuk (*Dot-Peen*), gores (*Scribing*), tekan (*Stamp*) dan juga laser (*Light Amplification by Stimulated Emission of Radiation*). Masing-masing metode mempunyai kelemahan dan keunggulannya tersendiri (Dikky Antonius, 2020)

Komponen Mesin Laser Marking:

1. Sumber Laser: Mesin ini dilengkapi dengan sumber laser yang dapat berupa *CO2 laser*, *fiber laser*, atau *Nd:YAG laser*. Sumber laser ini menghasilkan sinar laser yang digunakan untuk marking.
2. *Galvo Scanner*: *Galvo scanner* adalah sistem cermin yang dapat mengarahkan sinar laser dengan cepat dan akurat ke lokasi yang diinginkan pada permukaan material. Ini memungkinkan marking berlangsung dengan cepat dan presisi.
3. Fokus Optik: Fokus optik difungsikan untuk memfokuskan sinar laser menjadi titik kecil yang sangat terkonsentrasi pada permukaan material. Fokus optik ini dapat disesuaikan untuk menghasilkan marka dengan berbagai ukuran dan ketajaman.
4. Kontroler (biasanya komputer) mengontrol arah, intensitas, kecepatan gerakan, dan penyebaran sinar laser yang diarahkan ke permukaan material (Dikky Antonius, 2020).
5. Perangkat Lunak (*Software*): Perangkat lunak pengendali digunakan untuk merancang dan mengatur marka yang akan dibuat. Ini

memungkinkan operator untuk membuat teks, gambar, kode bar, atau grafik lainnya yang akan di-mark.

Cara Kerja Mesin Laser Marking:

1. **Pemilihan Material:** Pertama, operator harus memilih material yang akan di-mark. Mesin laser marking dapat digunakan pada berbagai material termasuk logam, plastik, kaca, karet, kulit, dan banyak lagi.
2. **Pengaturan Parameters:** Operator mengatur parameter laser seperti daya laser, kecepatan pemindahan *galvo scanner*, dan fokus optik sesuai dengan karakteristik material yang akan di-mark.
3. **Desain Marka:** Menggunakan perangkat lunak pengendali, operator merancang marka yang akan diaplikasikan pada material. Desain ini dapat berupa teks, grafik, nomor seri, atau informasi lainnya.
4. **Pemindahan *Galvo Scanner*:** *Galvo scanner* mengarahkan sinar laser ke lokasi yang diinginkan pada permukaan material sesuai dengan desain yang telah dibuat.
5. **Pemberian Marka:** Sinar laser diarahkan ke permukaan material, menciptakan perubahan sifat material, misalnya, oksidasi pada logam atau perubahan warna pada plastik. Proses ini menciptakan marka pada material.
6. **Verifikasi dan Inspeksi:** Setelah marking selesai, marka dapat diperiksa secara visual atau dengan sistem inspeksi otomatis untuk memastikan bahwa marka terbaca dengan benar dan berkualitas.



Gambar 2. 3 Mesin Laser *Marking*

2.2.6 Robot Solder

Robot solder digunakan dalam industri elektronik karena lebih aman, presisi, dan konsistensi dalam melakukan proses soldering pada komponen elektronik atau material logam. Proses soldering melibatkan peleburan logam untuk menghubungkan dua atau lebih komponen elektronik secara permanen. Menurut (Hao Wu , Wenbin Gao, Xiangrong Xu, Anggota, 2019) Selain itu, metode klasifikasi tradisional robot solder, seperti jaringan saraf dan statistik, hanya dapat mengklasifikasikan jenis cacat; metode pencocokan templat hanya dapat mencocokkan posisi objek.

Robot solder memiliki sistem penglihatan dan kontrol yang memungkinkan mereka menghasilkan sambungan dengan kualitas yang baik dengan kecepatan tinggi. Beberapa robot solder memiliki teknologi pemrosesan gambar 3D yang

memungkinkan mereka mengenali dan menyesuaikan diri dengan permukaan dan bentuk 3D dari komponen yang akan di-solder. Mereka juga dapat mengenali posisi dan orientasi komponen secara otomatis dan menyesuaikan pergerakan pembakaran solder (pembakar solder) sesuai dengan kebutuhan. Robot solder juga memiliki sistem pengendalian suhu yang akurat untuk memastikan suhu yang tepat digunakan untuk soldering. Memungkinkan proses produksi berjalan secara efisien, robot solder dapat diintegrasikan ke dalam sistem otomatisasi pabrik yang lebih luas. Ketika robot solder digunakan, kecelakaan kerja yang terjadi pada pekerja manusia selama proses soldering diminimalkan.

Dalam metode solder juga ada beberapa hal yang perlu diperhatikan seperti (Jaringan Inspeksi Masker) untuk deteksi cacat sambungan solder memiliki tiga cabang tugas yang berbeda: yaitu, penentuan posisi kotak pembatas, klasifikasi kotak pembatas, dan segmentasi objek. Struktur utama jaringan menggunakan ResNet dan RPN yang terhubung ke lapisan konvolusional terakhir ResNet dan menghasilkan ROI. Untuk setiap ROI, fitur terkait diekstraksi dan dikumpulkan ke ukuran tetap, yang digunakan sebagai masukan ke setiap cabang (Hao Wu , Wenbin Gao, Xiangrong Xu, Anggota, 2019). Di tempat lain, penggunaan robot solder meningkatkan konsistensi, kecepatan, dan ketepatan proses soldering.



Gambar 2. 4 Robot Soldering

2.2.7 Indikator Pemeriksaan

Tergantung pada jenis mesin dan fungsinya, indikator pemeriksaan pada mesin dapat melibatkan berbagai metode pengukuran dan pemantauan untuk melacak dan mengevaluasi kinerja mesin. Indikator ini memberikan petunjuk kepada operator atau teknisi tentang kondisi mesin dan memungkinkan mereka untuk menemukan masalah potensial atau kebutuhan perawatan.

Berikut indicator pengecekan pada mesin :

1. Suhu

Pengukuran suhu mesin dapat mengindikasikan apakah mesin bekerja pada suhu yang aman. Suhu yang berlebihan dapat menunjukkan masalah dalam pendinginan Atau overhead.

2. Tekanan

Indikator tekanan seperti tekanan oli, tekanan udara, atau tekanan hidrolis digunakan untuk memastikan bahwa mesin beroperasi pada tekanan yang sesuai untuk fungsi-fungsinya.

3. Kecepatan

Pengukuran kecepatan rotasi mesin atau komponen tertentu seperti motor, poros, atau penggerak lainnya dapat memberikan informasi mengenai performa mesin.

4. Getaran

Pemantauan getaran pada mesin dapat membantu mendeteksi kerusakan pada komponen mesin atau ketidakseimbangan yang dapat menyebabkan keausan lebih cepat.

5. Level Bahan Bakar atau Cairan

Indikator level bahan bakar atau cairan pendingin memberikan informasi mengenai tingkat bahan bakar atau cairan pendingin yang tersedia.

6. Kelembaban

Kelembaban mesin dapat mempengaruhi kinerjanya, khususnya pada mesin yang rentan terhadap korosi atau kerusakan akibat kelembaban.

7. Keausan dan Kerusakan

Penggunaan sensor dan teknologi pemantauan visual dapat membantu mendeteksi keausan atau kerusakan pada komponen mesin seperti gigi, bantalan, atau sabuk.

8. Kualitas dan Pemisahan Produk

Pada mesin produksi, indikator seperti dimensi produk, kualitas permukaan, atau deteksi cacat digunakan untuk memastikan bahwa produk yang dihasilkan memenuhi standar kualitas.

9. Monitoring Energi

Indikator pemakaian energi dapat membantu mengidentifikasi mesin yang membutuhkan perawatan atau penggantian komponen untuk meningkatkan efisiensi energi.

10. Sistem Alarm

Pengaturan sistem alarm untuk memperingatkan operator atau teknisi ketika indikator pemeriksaan mencapai batas tertentu, menunjukkan keadaan darurat atau masalah yang memerlukan perhatian segera.

2.2.8 Pengertian *Prepventive Maintenance*

Proses merencanakan dan menjadwalkan kegiatan perawatan rutin pada peralatan atau mesin untuk mencegah kerusakan, memperpanjang umur pakai, dan memastikan bahwa peralatan berfungsi dengan efisiensi terbaik dikenal sebagai pemeliharaan pencegahan. Menurut (Marasabessy et al., 2020) Menyusun rencana perawatan adalah salah satu cara untuk meminimalkan *downtime* Manajemen

perawatan mesin produksi sangat penting untuk menjaga keandalan (reability) mesin agar tetap beroperasi dengan baik.

Tujuan utama *preventive maintenance* adalah mencegah kerusakan dan kegagalan peralatan. Dengan melakukan pemeliharaan teratur, potensi masalah bisa terdeteksi lebih awal, mencegah kerusakan yang lebih serius yang mungkin menyebabkan *downtime* produksi yang mahal, Untuk mencetak laporan perawatan pencegahan, beberapa langkah harus dilakukan.

1. Memperpanjang Umur Peralatan

Pemeliharaan yang teratur dapat memperpanjang umur pakai peralatan. Peralatan yang dirawat dengan baik cenderung bertahan lebih lama dan memiliki nilai investasi yang lebih baik.

2. Meningkatkan Efisiensi Operasional

Mesin yang dirawat dengan baik biasanya lebih efisien dalam operasinya. Pemeliharaan preventif mengurangi risiko kerusakan produksi dan meningkatkan produktivitas dengan menjaga peralatan beroperasi secara optimal.

3. Menjaga Standar Kualitas

Standar kualitas yang tinggi sangat penting dalam beberapa industri, terutama manufaktur. *Maintenance* preventif membantu memastikan bahwa peralatan beroperasi sesuai dengan spesifikasi yang ditentukan, sehingga menjaga kualitas produk.

4. Meningkatkan Keamanan Kerja

Pemeliharaan preventif dapat meningkatkan keselamatan di tempat kerja. Mesin yang dijaga dengan baik cenderung memiliki risiko kecelakaan yang lebih rendah.

5. Optimalkan Waktu dan Biaya

Dengan merencanakan pemeliharaan pada waktu yang tepat dan memastikan penggunaan suku cadang yang tepat, perusahaan dapat mengoptimalkan biaya dan waktu yang dikeluarkan untuk pemeliharaan, mengurangi biaya *downtime* yang tidak terduga.

6. Mendukung Kepatuhan Regulasi

Penjadwalan pemeliharaan preventif membantu bisnis mematuhi peraturan dan standar industri yang berlaku.

7. Meminimalkan Kerugian Akibat *Downtime*

Pemeliharaan yang teratur mencegah *downtime* yang tidak terduga karena kerusakan mesin. Dengan menjadwalkan pemeliharaan pada waktu yang tepat, perusahaan dapat mengurangi kerugian finansial karena waktu henti produksi.

8. Meningkatkan Keandalan Peralatan

Peralatan menjadi lebih handal dengan pemeliharaan yang teratur. Penggunaan rutin dan pemantauan yang cermat membantu menemukan masalah potensial sebelum menjadi masalah yang serius.

9. Mendukung Pelacakan dan Analisis

Jadwal pemeliharaan pencegahan menyimpan informasi penting tentang pemeliharaan. Dalam bidang manajemen aset, data ini dapat

digunakan untuk menganalisis kinerja peralatan, merencanakan penggantian peralatan, dan membuat keputusan strategis.

Perusahaan dapat mencapai tujuan ini dengan menjadwalkan perawatan preventif untuk mengurangi biaya, meningkatkan efisiensi operasional, dan memastikan keberlanjutan operasi dalam jangka panjang.

2.2.9 Pengertian Odoo

Odoo adalah software aplikasi bisnis yang menggabungkan *CRM (Customer Relationship Management)*, yang, sesuai namanya, adalah strategi bisnis yang menggabungkan teknologi, proses, dan manusia. Selain itu, *software* ini mencakup manajemen proyek, manufaktur, gudang, penjualan, dan keuangan. ODOO adalah aplikasi berbasis open source yang menggabungkan sistem *Enterprise Resource Planning (ERP)* untuk mengelola komponen bisnis perusahaan (Ismail Palandeng & Retnoningsih, 2021). ERP adalah sistem yang mendukung proses bisnis dengan menyediakan informasi yang terintegrasi secara realtime antar divisi fungsional perusahaan. Dengan menggunakan ERP, perusahaan dapat mengolah dan menampilkan sejumlah besar data atau informasi dengan mudah (Suminten et al., 2019).

Menurut (Githa & Raharja, 2021) odoo adalah aplikasi web berbasis open source, yang dapat digunakan untuk membangun sistem e-SCM. Modul-modul Odoo dapat disesuaikan dengan kebutuhan sistem yang sedang dibangun, dalam hal ini sistem e-SCM. Lebih mendalam Odoo adalah sistem atau software manajemen open source yang mudah digunakan dan diintegrasikan. Bentuknya termasuk berbasis web, desktop, dan mobile. Selain itu, banyak kelebihan software ini,

termasuk biaya yang terjangkau, pemasangan yang mudah, didukung oleh banyak komunitas, dan lengkap dan terintegrasi.

2.2.10 Pengertian *PostgreSql*

Database postgresql adalah database relasional sumber terbuka kelas perusahaan tingkat lanjut yang mendukung kueri *SQL (relasional)* dan JSON (non-relasional). Ini adalah sistem manajemen basis data yang sangat stabil, *PostgreSQL* digunakan sebagai penyimpanan data utama atau gudang data untuk banyak aplikasi web, seluler, geospasial, dan analitik. Menurut (Aminullah et al., 2018) *PostgreSQL* merupakan sebuah ObjectRelational Database Management System (ORDBMS) dan masih memiliki fitur-fitur khas DBMS tradisional tetapi dengan sejumlah perbaikan untuk suatu pekerjaan dan fungsional sebagaimana juga bisa ditemukan pada sistem-sistem DBMS generasi pada saat ini.

PostgreSQL memiliki sejarah yang kaya dalam mendukung tipe data tingkat lanjut, dan mendukung tingkat optimalisasi kinerja yang umum di seluruh database komersial, seperti *Oracle* dan *SQL Server*. *AWS* mendukung *PostgreSQL* melalui layanan database yang terkelola sepenuhnya dengan *Amazon Relational Database Service (RDS)*. *Amazon Aurora* dengan kompatibilitas *PostgreSQL* juga dibangun menggunakan *PostgreSQL*.

2.2.11 Pengertian Python

Python diciptakan oleh *Guido van Rossum* pertama kali di *Centrum Wiskunde & Informatica (CWI)* di Belanda pada awal tahun 1990-an. Bahasa python terinspirasi dari bahasa pemrograman ABC. Sampai sekarang, *Guido* masih menjadi penulis utama untuk *python*, meskipun bersifat open source sehingga ribuan orang juga berkontribusi dalam mengembangkannya. Di

tahun 1995, Guido melanjutkan pembuatan *python* di Corporation for National Research Initiative (CNRI) di Virginia Amerika, di mana dia merilis beberapa versi dari *python*. Pada Mei 2000, Guido dan tim Python pindah ke BeOpen.com dan membentuk tim BeOpen PythonLabs. Di bulan Oktober pada tahun yang sama, tim *python* pindah ke Digital Creation. Pada tahun 2001, dibentuklah Organisasi *Python* yaitu *Python Software Foundation (PSF)* (Cahyadi Agustin, 2021).

2.2.12 Pengertian XML (*eXtensible Markup Language*)

XML, atau *eXtensible Markup Language*, adalah sebuah bahasa markup yang mendefinisikan seperangkat aturan untuk mengkodekan dokumen dalam format yang dapat dibaca baik oleh manusia maupun oleh mesin. Tujuannya adalah untuk menyimpan dan mentransportasikan data dengan fokus pada kemudahan pemahaman dan keumuman. XML umumnya digunakan untuk pertukaran data antara sistem, file konfigurasi, dan merepresentasikan data hirarkis dalam berbagai aplikasi dan industri.

Komponen dari file XML

File Extensible Markup Language (XML) adalah dokumen berbasis teks yang dapat Anda simpan dengan ekstensi *.xml*. Anda dapat menulis XML mirip dengan *file* teks lainnya. Untuk membuat atau mengedit *file* XML, Anda dapat menggunakan salah satu hal berikut:

1. Editor teks seperti Notepad atau Notepad++
2. Editor XML *online*
3. Peramban web

Setiap file XML menyertakan komponen berikut.

1. Dokumen XML

Tanda `<xml></xml>` digunakan untuk menandai awal dan akhir dari sebuah *file* XML. Konten dalam tanda ini juga disebut dokumen XML. Itu adalah tanda pertama yang akan dicari oleh perangkat lunak apa pun untuk memproses kode XML.

2. Deklarasi XML

Sebuah dokumen XML dimulai dengan beberapa informasi tentang XML itu sendiri. Misalnya, dokumen itu mungkin menyebutkan versi XML yang diikutinya. Pembukaan ini disebut deklarasi XML. Berikut adalah contohnya.

```
<?xml version="1.0" encoding="UTF-8"?>
```

3. elemen XML

Semua tanda lain yang Anda buat dalam dokumen XML disebut dengan elemen

XML. Elemen XML dapat berisi fitur berikut:

1. Teks
2. Atribut
3. Elemen lainnya

Semua dokumen XML dimulai dengan tanda primer, yang disebut dengan elemen *root*.

Misalnya, perhatikan *file* XML di bawah.

```
<InvitationList>
```

```
<family>
```

```
<aunt>
```

```

<name>Christine</name>
  <name>Stephanie</name>
</aunt>
</family>
</InvitationList>

```

<InvitationList> adalah elemen *root*; *family* (keluarga) dan *aunt* (bibi) adalah nama elemen lainnya.

4. Atribut XML

Elemen XML dapat memiliki deskriptor lain yang disebut atribut. Anda dapat menentukan nama atribut Anda sendiri dan menulis nilai atribut dalam tanda kutip seperti yang ditunjukkan di bawah.

```
<person age="22">
```

5. Konten XML

Data dalam *file* XML juga disebut konten XML. Misalnya, dalam *file* XML, Anda mungkin melihat data seperti ini.

```

<friend>
  <name>Lusy</name>
  <name>Karlina</name>
</friend>

```

Nilai data *Lusi* dan *Karlina* adalah kontennya.

2.2.13 Pengertian PgAdmin

PgAdmin merupakan aplikasi atau interface database postgresql yang dapat digunakan untuk melakukan desain dan manajemen secara komprehensif,

pgAdmin tersedia dalam versi Windows dan Linux. PgAdmin menggunakan lisensi Artistic License yang tetap dapat digunakan dan di sebar luaskan secara gratis.

Menurut (Aminullah et al., 2018) PgAdmin merupakan aplikasi atau interface database postgresql yang dapat digunakan untuk melakukan desain dan manajemen secara komprehensif, pgAdmin tersedia dalam versi Windows dan Linux. PgAdmin menggunakan lisensi Artistic License yang tetap dapat digunakan dan di sebar luaskan secara gratis. Selain itu pgAdmin juga sub aplikasi perangkat lunak pendukung pengembangan dan alat bantu administrasi DBMS *PostgreSQL* yang memiliki banyak fungsi

2.2.14 Penelitian Terdahulu

Tabel 2. 5 Penelitian Terdahulu

No	Nama Peneliti	Judul	ISSN/ISBN	Kesimpulan/Hasil
1	Hao Wu , Wenbin Gao, Xiangrong Xu, Anggota, IEEE	Solder joint recognition using Mask R- CNN method	51505002, 51605004	Penerapan solder baru yang memanfaatkan metode jaringan inspeksi Mask R- CNN untuk mengklasifikasikan dan mengelompokkan k esalahan sambungan

				solder secara bersamaan.
2	Chinmoy Ghorai, Anggota Mahasiswa, IEEE, Tukar Shakhari, Anggota Mahasiswa, IEEE, Indrajit Banerjee, Anggota, IEEE	A SPEA-Based Multimetric Routing Protocol for Intelligent Transportation Systems	1254-9050	Pembuatan perutean multimetric untuk mencari jalur komunikasi yang efisien dengan bantuan Solusi berbasis Strength Pareto <i>Evoluntionary Algorithm (SPEA)</i>
3	Rima Rizqi Wijayanti, Wada Kaligula Budiargo, Abdurrasyid	Rancang Bangun Sistem Informasi Borescope Inspection Report Pada Engine <i>Maintenance</i> PT. GMF Aeroasia, Tbk	2549-0710 2722-2713	Perancangan sistem report/pelaporan hasil borescope yang awalnya melakukan penginputan secara manual menjadi report berbasis web dengan menggunakan <i>framework laravel</i>

4	Dudi Awalludin, Donny Apdian, Vina Kristiani	Analisis dan Perancangan Sistem Informasi Pembuatan Daily Report Produksi Dies Berbasis Web	2302-7339 1412-3622	Pembuatan daily report produksi yang sedang berjalan dengan metode pengembangan sistem menggunakan SDLC dengan model waterfall dengan tahapan System Engineering, Requirement <i>Analysis</i> dan Design
5	Dikky Antonius, Cyrilus Pandu	Analisa Parameter Laser Marking pada Material Stainless Steel Terhadap Struktur Mikro dan Kedalaman Marking	2502-3829	peningkatan power dan frekuensi pada laser marking berpengaruh pada melebarnya diameter dari hasil marking dikarenakan peningkatan

				divergence yang dipengaruhi energi yang diaplikasikan
6	Cahyadi Agustin, Firman Rizaldi Yusup	Perancangan Sistem Informasi Human Resource Development Berbasis Web menggunakan <i>Flamework Odo</i> di PT. Nuro Intelia Prospera Jakarta	2722-4147 2339-188X	Pembuatan aplikasi sistem Human <i>Resource Development</i> yang dapat memudahkan dalam proses pengelolaan data karyawan dan absensi berbasis web dengan metode <i>waterfall</i>