# BAB 2
# TINJAUAN PUSTAKA

## 2.1 Tinjauan Sekolah

Tinjauan perusahaan dimaksudkan untuk mengetahui hal-hal yang berkaitan dengan perusahaan, antara lain profil sekolah, visi dan misi sekolah, struktur organisasi.

### 2.1.1 Profil Sekolah

SMK Widya Dirgantara yang berdiri pada tahun 1986 dan beralamat di Jl. Bojong Raya No.114A Bandung merupakan sekolah kejuruan swasta yang didirikan oleh Yayasan Pendidikan Widya Dirgantara. Sekolah ini mempunyai berbagai macam kompetensi keahlian, seperti Teknik Kendaraan Ringan, Teknik Sepeda Motor, Teknik Pemesinan, dan Teknik Komputer dan Jaringan.

#### 2.1.1.1 Sejarah Sekolah

SMK Widya Dirgantara mulai didirikan pada awal tahun 1986,namun baru mendapatkan SK Pendirian dari departemen Pendidikan Nasional pada tahun 1987 tepatnya tanggal 30 januari. Pada awalnya SMK Widya Didirikan oleh 3 orang yaitu bapak H. Padmono Subagjo, SE. Sebagai ketua yayasan, H. Sugian Noor Gani, SE. Sebagai wakilnya dan H. Iri Mamad sebagai bendaharanya, yayasan ini bergerak dibidang pendidikan yaitu sekolah menengah kejuruan.SMK Widya Dirgantara sendiri dipimpin oleh bapak Asep Junaedi, S.Pd. sebagai kepala sekolahnya dan mulai diangkat pada tahun 2008, tepatnya melalui SK Pengangkatan nomor : Skep/YPWD/08/III/Pers/2008 Tanggal 3 Maret 2008. SMK Widya Dirgantara berlokasi di Jl. Bojong Raya No. 114A Bandung, Kelurahan Cibuntu, Kecamatan Bandung Kulon, Kode Pos : 40212, Nomer Telepon : (022)6071816, Email : widyadirgantara@gmail.com

### 2.1.2 Visi dan Misi Sekolah

Visi dan Misi dari SMK Widya Dirgantara adalah sebagai berikut:

#### 2.1.2.1 Visi

Mewujudkan Lembaga Pendidikan yang bermutu, berbudaya, dan berkarakter sehingga menghasilkan generasi yang kompetitif, mandiri, dan agamis yang didukung dengan penguasaan kecakapan local dan global.

#### 2.1.2.2 Misi

1. Menghasilkan tamatan yang cerdas, terampil, kompetitif, mandiri serta agamis.
2. Mewujudkan lingkungan yang menjujung tinggi budaya bangsa dan budi pekerti luhur.
3. Menanamkan jiwa kewirausahaan (*entrepreneurship)* bagi semua lulusan.
4. Mengembangkan ilmu pengetahuan dan teknologi yang relevan dengan kebutuhan masyarakat lokal maupun global.

### 2.1.3 Badan Hukum Instansi

SMK Widya Dirgantara merupakan yayasan persekutuan *(partnership)* atau usaha kemitraan yang merupakan kombinasi terorganisir dari dua orang atau lebih untuk menjalankan suatu usaha sebagai mitra pemilik atau mitra pengelola dan dimiliki oleh dua orang atau lebih yang bekerja sama untuk mencapai tujuan bisnis. Izin perusahaan dan instansi pemerintah yang terkait telah mengikuti aturan dalam departemen pendidikan nasional.

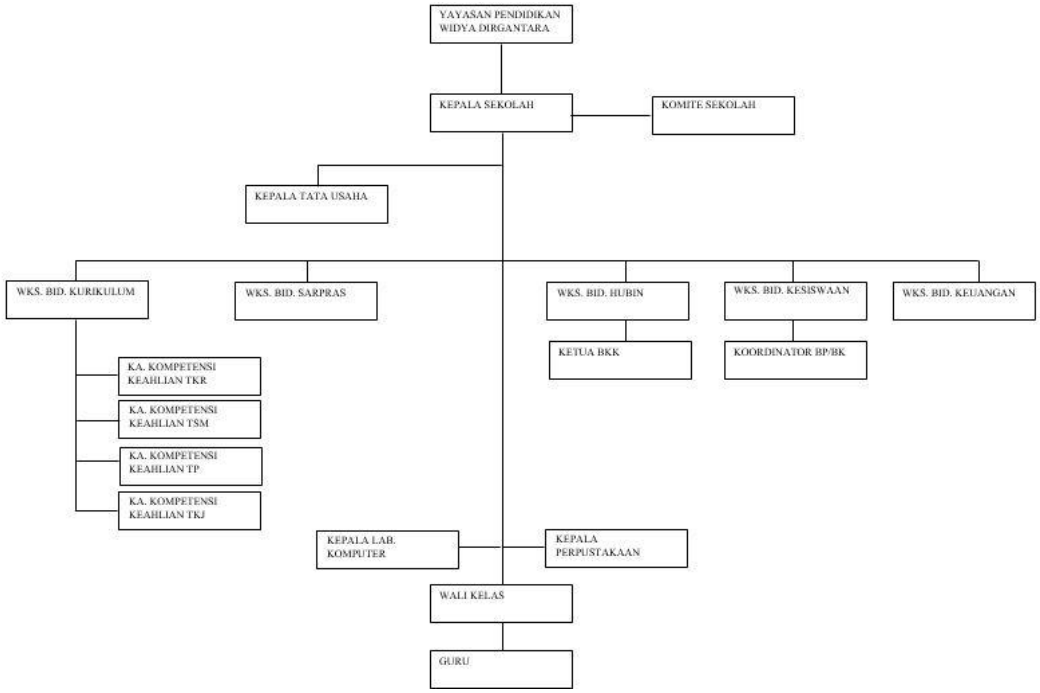
Terdapat Beberapa Legalitas yaitu :

| | | | |
|---|---|---|---|
| 1. | No. Statistik Sekolah (NSS) | : | 32.2.02.60.19.025 |
| 2. | NPSN | : | 20265547 |
| 3. | SK Pendirian | : | Kanwil Depdiknas No. 047/102/Kep/E/87 Tanggal 30 Januari 1987 |

### 2.1.4 Struktur Organisasi Sekolah

Struktur organisasi sekolah merupakan bagian yang sangat penting dalam penyelenggaraan kegiatan, karena struktur organisasi sekolah terdapat

pendelegasian wewenang terhadap setiap lini/tingkatan dalam sekolah. Berikut adalah struktur organisasi yang terdapart di SMK Widya Dirgantara:




**Gambar 2. 1 Struktur Organisasi**

## 2.2 Landasan Teori

Landasan teori yang digunakan dalam penyusunan skripsi dan model simulasi sistem pengereman mobil di SMK Widya Dirgantara adalah sebagai berikut :

### 2.2.1 Teori Tentang Model

Model adalah pola (contoh,acuan,ragam) dari sesuatu yang akan dibuat atau dihasilkan.

### 2.2.2 Teori Tentang Simulasi

Simulasi adalah sebuah metode pelatihan yang memeragakan sesuatu dalam bentuk tiruan yang mirip dengan keadaan sesungguhnya atau dengan kata lain penggambaran suatu sistem atau proses yang berupa pemeranan.

### 2.2.3 Definisi Mobil

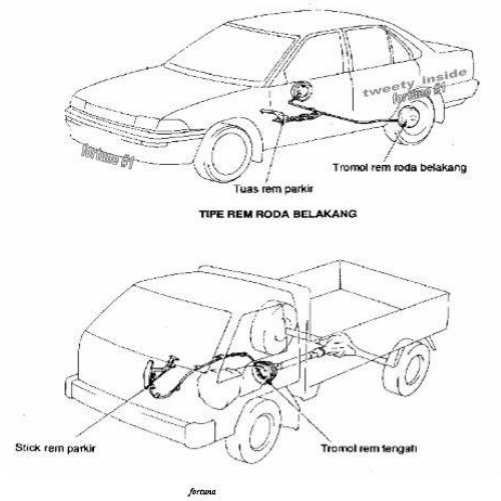
Menurut Kamus Besar Bahasa Indonesia (KBBI), mobil adalah kendaraan darat yg digerakkan oleh tenaga mesin, beroda empat atau lebih (selalu genap), biasanya menggunakan bahan bakar minyak untuk menghidupkan mesinnya. [3]

### 2.2.4 Definisi Rem

Menurut Kamus Besar Bahasa Indonesia (KBBI), Rem adalah alat untuk menahan gerakan atau mekanisme dengan jalan gesekan; alat untuk memperlambat atau untuk menghentikan gerakan atau putaran, misalnya pada roda mobil. [4]

#### 2.2.4.1 Rem Parkir

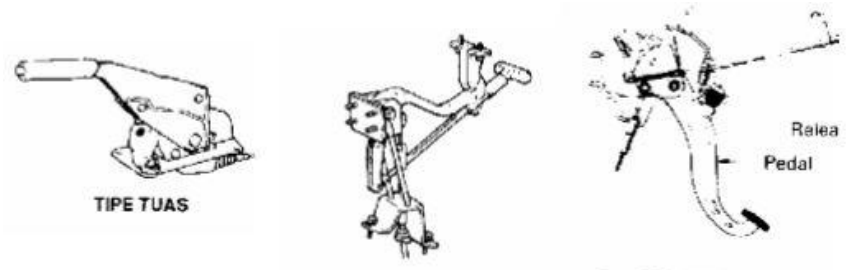
Rem parkir (parking brake) terutama digunakan untuk parkir kendaraan. Mobil penumpang dan kendaraan niaga yang kecil mempunyai rem parkir tipe roda belakang (rem kaki), atau rem parkir ekslusif yang dihubungkan dengan roda-roda belakang. Kendaraan niaga yang besar menggunakan rem parkir tipe center brake yang dipasang antara propeller shaft dan transmisi. Sistem rem parkir terdiri dari tuas rem, stick atau pedal, kabel atau tipe mekanisme batang (rod) dan tromol rem dan sepatu yang membangkitkan daya pengereman.




**Gambar 2. 2 Rem Parkir**

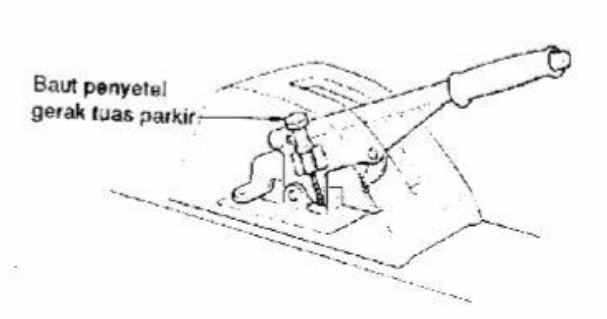
#### 2.2.4.1.1 Cara Kerja

Mekanisme kerja (operating mechanism) pada rem parkir pada dasarnya sama untuk tipe rem parkir belakang dan tipe center brake. Tuas rem parkir ditempatkan berdekatan dengan tempat duduk pengemudi dengan menarik tuas rem parkir maka rem bekerja melalui kabel yang dihubungkan dengan tuas. Ada beberapa tipe tuas rem parkir seperti diperlihatkan di bawah ini, yang digunakan bergantung pada design tempat duduk pengemudi dan sistem kerja yang dikehendaki.



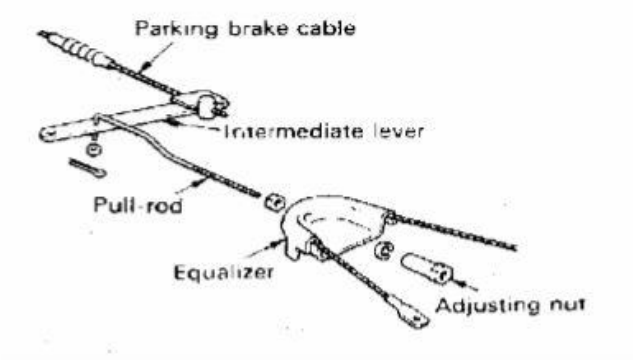

**Gambar 2. 3 Tipe Tuas Rem Parkir**

Tuas rem parkir dilengkapi dengan ratchet utnuk mengatur tuas pada suatu posisi pengetesan. Pada beberapa tuas rem parkir mur penyetelannya dekat dengan tuas rem, dengan demikian penyetelan jarak tuas dapat dengan mudah distel.




**Gambar 2. 4 Penyetelan Tuas Rem Parkir**

Kabel rem parkir memindahkan gerakan tuas ke tromol rem sub-assembly. Pada rem parkir roda belakang, dibagian tengah kabel diberi equalizer untuk menyamakan daya kerjanya tuas pada kedua roda-roda. Tuas intermediate (intermediate lever) dipasang untuk menambah daya pengoperasian.




**Gambar 2. 5 Cara Kerja Rem Parkir**

#### 2.2.4.1.2 Body Rem Parkir

a) Rem Parkir Tipe Roda Belakang

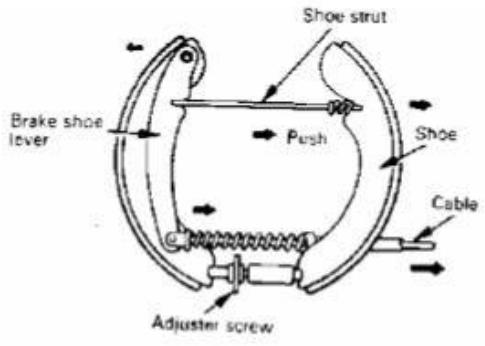
Bodi rem parkir dikelompokkan menjadi dua tipe structural bergantung pada andilnya tromol rem atau piringan rem (rem kaki) atau komponen rem yang terpisah.

1. Pelayanan Rem Tipe Sharing (Rem Kaki)

   Tipe rem parkir ini digabungkan dengan rem kaki. Hubungannya dilakukan secara mekanik dihubungkan pada sepatu rem pada kendaraan yang mempunyai tromol rem, atau pada piston pada mobil yang menggunakan disc brake.

2. Kendaraan dengan tromol rem

   Pada tipe rem parkir ini, sepatu rem akan mengembang oleh tuas sepatu rem dan shoe strut (lihat gambar). Kabel rem parkir dipasang pada tuas sepatu rem dan daya kerja dari tuas rem parkir dipindahkan melalui kabel rem parkir ke tuas sepatu rem.

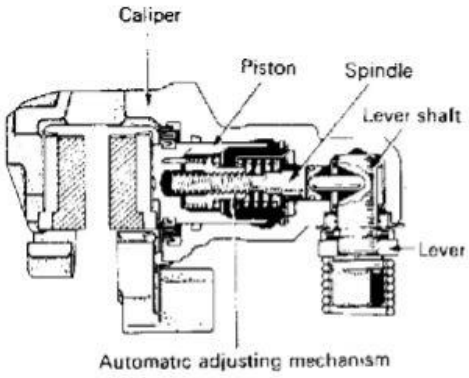



**Gambar 2. 6 Tipe Rem Parkir Tromol**

3. Kendaraan dengan rem piringan

   Dalam tipe rem parkir ini, mekanisme rem parkir disatukan dalam caliperr rem piringan. Seperti pada gambar di bawah, gerakan tuas menyebabkan poros tuas (lever shaft) berputar menyebabkan spindle menggerakkan piston. Hasilnya, pad terdorong menekan rotor piringan (disc rotor) . Pad menjadi aus dan langkah rem parkir akan bertambah dengan alasan ini, maka dilengkapi mekanisme

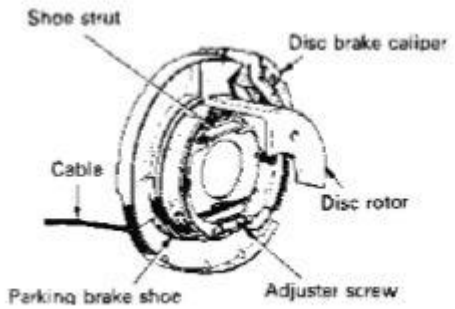
penyetelan otomatis pada mekanisme rem parkir untuk menjaga langkah spindle agar tetap konstan setiap waktu.




**Gambar 2. 7 Tipe Rem Parkir Piringan**

4. Tipe rem parker deveted

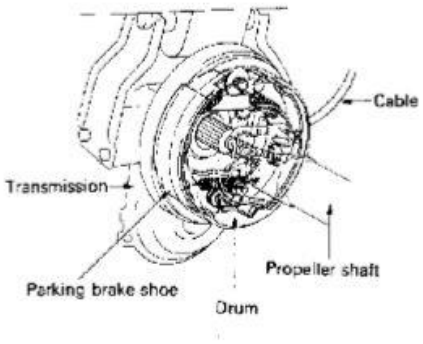
Pada tipe rem parkir ini, tromol rem parkir terpisah dari rem piringan belakang, seperti pada gambar. Cara kerjanya sama dengan tipe rem parkir seperti pada tromol rem.




**Gambar 2. 8 Tipe Rem Parkir Deveted**

b) Tipe Center Brake

Tipe center brake ini banyak digunakan pada kendaraan komersil. Tipe ini salah satu dari tipe rem tromol tetapi dipasang antara bagian belakang transmisi dan bagian depan propeller shaft. Pada rem parkir tipe center brake ini daya pengeremannya terjadi pada saat sepatu rem yang diam ditekan dari bagian dalam terhadap tromol yang berputar bersama out put shaft transmisi dan propeller shaft. Tipe rem ini bekerjanya sama dengan rem parkir tipe sharing pada kendaraan yang menggunakan rem tromol.




**Gambar 2. 9 Tipe Rem Center Brake**

#### 2.2.4.1.3 Memperbaiki Rem Tangan

Masalah yang biasa terjadi pada rem tangan adalah ketika memarkir kendaraan. Pada tempat yang menurun, kendaraan masih juga bergerak. Hal umum sebagai penyebab masalah pengereman di antaranya adalah:

1. Kawat penarik telah mulur/kendor atau karat.
2. Tempat sambungan kendor atau karat.
3. Penyetelan kurang tepat.
4. Jarak bidang pengereman antara kanvas rem/pad dan tromol/cakram terlalu besar.

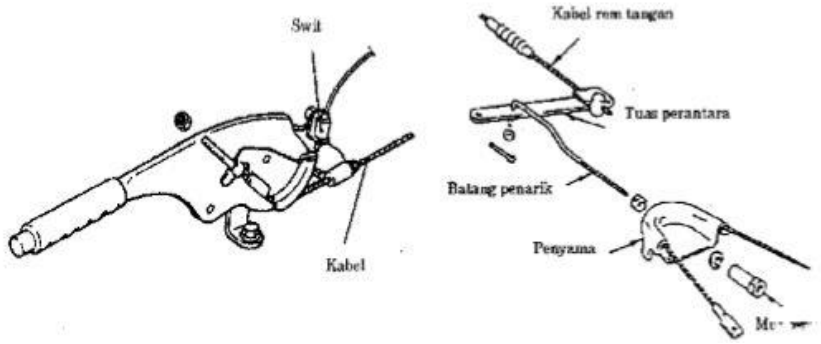
Oleh karena itu, sebelum kegiatan perbaikan, pemeriksaan terhadap komponen dan cara kerjanya harus dilakukan, yaitu:

1. Pastikan seluruh komponen berada pada kondisi normal dan dapat digunakan dengan baik.
2. Periksa gerakan tuas rem dengan cara menarik sampai kedudukan pengerem, dan terdengar suara “klik” sesuai spesifikasi. Posisi tuas rem yang benar biasanya setengah dari keseluruhan gerakan tuas.

Apabila setelah dilakukan pemeriksaan, sedangkan kerja rem tidak memperoleh hasil yang memuaskan, lakukan perbaikan dan penyetelan.

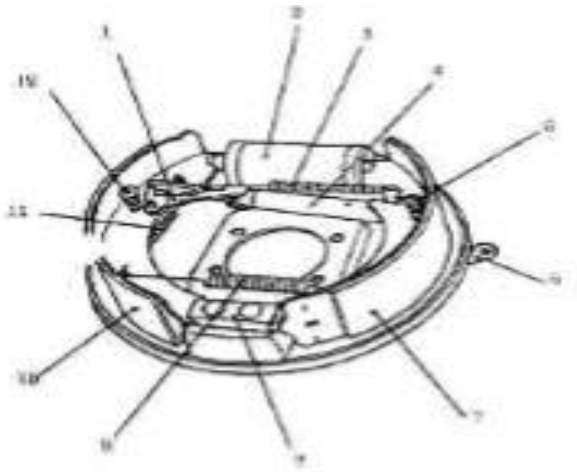
a. Bilamana tarikan kawat rem tidak lancar, berikanlah pelumasan jika masih memungkinkan.
b. Bilamana tarikan kawat melebihi spesifikasi karena kawat mulur, gantilah kawat beserta kelengkapannya.

c. Bila tarikan kawat melebihi spesifikasi karena setelan, lakukan penyetelan pada baut penyetel yang ada di tuas. Atau bilamana masih baik, dapat juga dilakukan penyetelan di bagian penyama (equalizer) di bagian bawah kendaraan.




**Gambar 2. 10 Perbaikan Rem Tangan**

d. Untuk penyetelan jarak bidang pengereman pada rem tromol tanpa penyetel otomatis, melalui pemutaran bintang (star) penyetel yang ada dalam tromol. Sedangkan, pada rem tromol dengan penyetel otomatis, jarak bidang pengereman telah dijamin oleh penyetel otomatis.



**Gambar 2. 11 Penyetelan Pada Rem Tromol**

Keterangan gambar:

| | |
|---|---|
| 1. Tuas penyetel | 7. Sepatu rem |
| 2. Silinder Roda | 8. Jangkar (Anchor) |
| 3. Pegas Pengembali | 9.Pegas pengembali |
| 4. Mur penyetel dengan penghubung berulit | 10. Sepatu rem |
| 5. Penahan | 11. Penahan |
| 6. Tuas rem parkir | 12. Mur penahan sepatu rem |

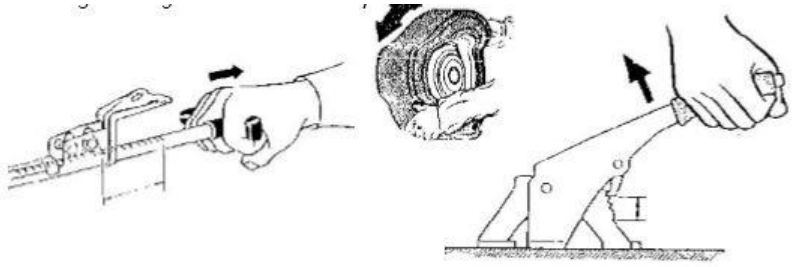
e. Untuk penyetelan jarak bidang pengereman pada rem cakram, menggunakan sekrup penyetel (3) apabila dilakukan, pengereman, tuas rem (2) karena tarikan kabel rem akan menekan piston beserta padnya melawan cakram dengan baik.

#### 2.2.4.1.4 Celah Sepatu Rem

Celah antara tromol dan kanvas yang besar akan menyebabkan kelambatan pada pengereman. Bila celah antara tromol dan kanvas terlalu kecil, rem akan terseret dan menyebabkan keausan pada tromol dan kanvas. Begitu juga, apabila celah sepatu pada keempat rodanya tidak sama pada semua roda-rodanya, maka kendaraan akan tertarik ke salah satu arah atau roda belakang kendaraan akan seperti ekor ikan (yang mengibas ke kanan dan ke kiri). Untuk mencegah kejadian ini, penting sekali untuk menyetel secara tepat celah antara tromol dan kanvas sesuai spesifikasi yang dianjurkan dan melakukan perawatan setiap saat. Pada beberapa tipe rem, penyetelannya bekerja secara otomatis, sedangkan untuk tipe lainnya celahnya harus dilakukan secara terbuka.

#### 2.2.4.1.5 Penyetelan Rem Tangan

Stel pada bagian penyetel sampai tercapai keadaan sesuai dengan gambar-gambar di bawah ini:



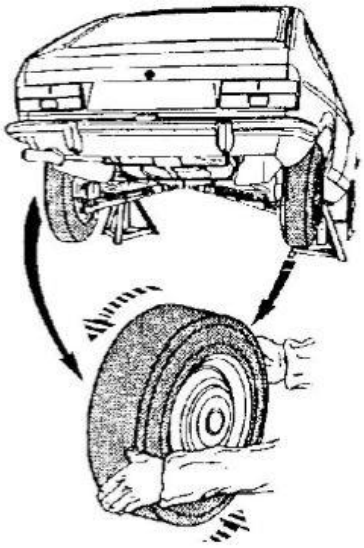
Tarik penuh, gerak batang 10 – 20 gigi
Kontrol: tarik 3 gigi, roda masih harus dapat berputar bebas

Tarik penuh, gerak tuas harus 3 – 7 gigi
Kontrol: tarik 1 gigi, roda masih harus dapat berputar bebas

**Gambar 2. 12 Penyetelan Rem Tangan**

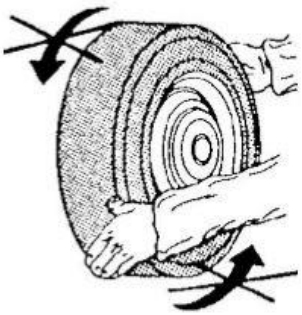
#### 2.2.4.1.6 Kontrol Kesamaan Kerja Rem Kanan dan Kiri

Tarik tuas tangan, gigi per gigi, sampai rem tangan mulai berfungsi. Kalau kondisi rem baik, hambatan gesek sama pada kedua roda.



**Gambar 2. 13 Kontrol Kesamaan Kerja Rem Kanan dan Kiri (1)**

Tarik tuas rem tangan lagi, gigi per gigi, sampai roda tak dapat diputar. Kalau rem tangan berfungsi dengan baik, hal itu terjadi dalam waktu bersamaan pada kedua roda.

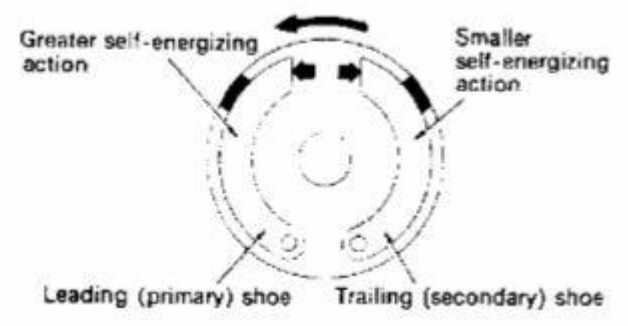


**Gambar 2. 14 Kontrol Kesamaan Kerja Rem Kanan dan Kiri (2)**

Ketidaksamaan kerja rem dapat berasal dari: - Nilai gesekan yang berbeda (tromol, kanvas) - Kelancaran jalan kabel rem tangan yang berbeda.

### 2.2.4.2 Rem Tromol

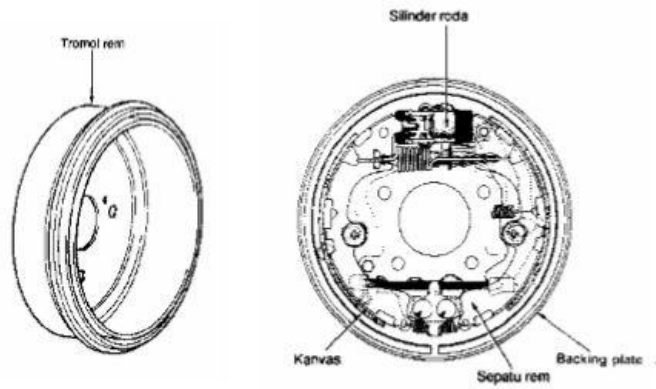
Pada tipe rem tromol kekuatan tenaga pengereman diperoleh dari sepatu rem yang diam menekan permukaan tromol bagian dalam yang berputar bersama-sama dengan roda. Karena self – energizing efect ditimbulkan oleh tenaga putar tromol dan tenaga mengembangkan sepatu, kekuatan tenaga pengereman yang besar diakibatkan oleh usaha pedal yang relatif kecil.




**Gambar 2. 15 Rem Tromol**

Komponen-komponen rem tromol:

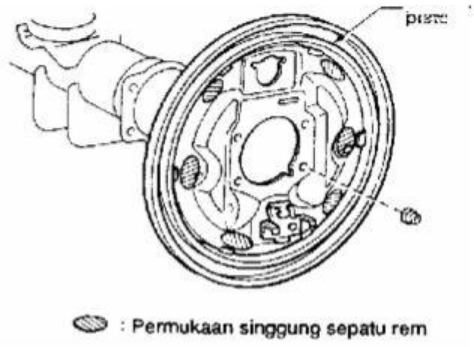
1. *Backing plate*
2. Silinder roda (*Wheel cylinder*)
3. Sepatu rem dan kanvas (*Brake shoe* dan *lining*)
4. Tromol rem (*Brake drum*)




**Gambar 2. 16 Komponen rem tromol**

1. *Backing Plate*

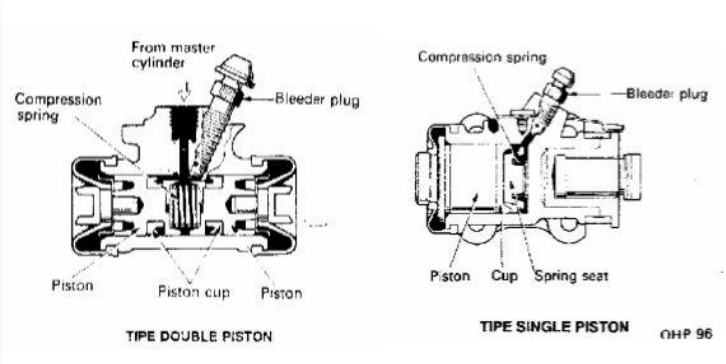
   Backing plate dibuat dari baja press yang dibaut pada axle housing atau axle carrier bagian belakang. Karena sepatu rem terkait pada backing plate maka aksi daya pengereman tertumpu pada backing plate.




***Gambar 2. 17 Backing Plate***

2. Silinder Roda

Silinder roda (*Wheel cylinder)* terdiri dari beberapa komponen seperti terlihat pada gambar di sebelah kanan. Setiap roda menggunakan satu atau dua buah silinder roda. Ada sistem yang menggunakan dua piston untuk menggerakkan kedua sepatu rem yaitu satu piston untuk setiap sisi silinder roda, sedangkan sistem yang lainnya hanya menggunakan satu piston untuk menggerakkan hanya satu sepatu rem. Bila timbul tekanan hidraulis pada master cylinder maka akan menggerakkan piston cup. Piston akan menekan kearah sepatu rem kemudian bersama-sama menekan tromol rem. Apabila rem tidak bekerja, maka piston akan kembali ke posisi semula dengan adanya kekuatan pegas pembalik sepatu rem . Bleeder plug disediakan pada silinder roda gunanya untuk membuang udara dari minyak rem.

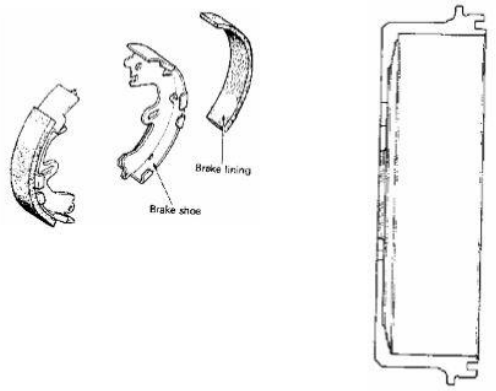



**Gambar 2. 18 Silinder Roda**

3. Sepatu rem dan Kanvas rem

   Sepatu rem (*brake shoes*) seperti juga tromol (*drum*) memiliki bentuk setengah lingkaran. Biasanya sepatu rem dibuat dari pelat baja. Kanvas rem dipasang dengan jalan dikeliling (pada kendaraan besar) atau dilem (pada kendaraan kecil) pada permukaan yang bergesekan dengan tromol.

   Kanvas ini harus dapat menahan panas dan aus serta harus mempunyai koefisien gesek yang tinggi. Koefisien tersebut sedapat mungkin tidak mudah dipengaruhi oleh keadaan turun naiknya temperatur dan kelembaban yang silih berganti. Umumnya kanvas (lining) terbuat dari campuran fiber metalic dengan brass, lead, plastik dan sebagainya dan diproses dengan ketinggian panas tertentu.

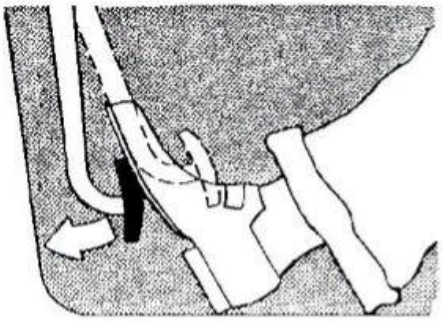



**Gambar 2. 19 Sepatu dan Kanvas Rem**

#### 2.2.4.2.1 Kontrol Fungsi Penguat Tenaga Rem (*Booster*)

Kontrol ini harus dilaksanakan, kalau pedal rem harus ditekan keras sekali untuk mencapai perlambatan/ pengereman mobil yang cukup.

1. Tekan pedal rem beberapa kali, pada saat motor mati.
2. Hidupkan motor sewaktu pedal rem ditekan. Kalau penguat tenaga berfungsi, pedal akan menurun sedikit, selama tahap tersebut.
3. Matikan motor sewaktu pedal rem ditekan. Pada tahap ini pedal tidak boleh ada reaksi. Jika peda akan terdorong kembali, katup anti-balik pada penguat tenaga harus dibersihkan/diganti.

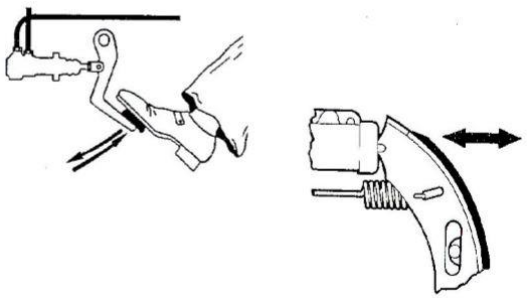


**Gambar 2. 20 Kontrol Fungsi Penguat Tenaga Rem**

#### 2.2.4.2.2 Pemeriksaan Fungsi Rem Tromol

Periksa apakah silinder rem macet. Lepas tromol hanya pada rem yang sedang diperiksa. Tromol roda-roda lain harus terpasang, agar torak-toraknya tidak tertekan keluar.

1. Minta tolong seseorang untuk menekan pedal rem, kedua kanvas ditahan dengan obeng. Torak-torak pada silinder rem yang diperiksa harus bergerak keluar tampa ada kebocoran di silinder roda. Jika terdapat kebocoran, semua silinder rem pada aksel yang diperiksa harus dioverhaul.
2. Periksa permukaan gesek pada tromol rem. Bila berwarna abu-abu sampai hitam, atau berkarat, nilai gesekannya kurang. Maka permukaan harus dibersihkan denga kertas gosok, atau lebih baik dengan dibubut/ digerinda.

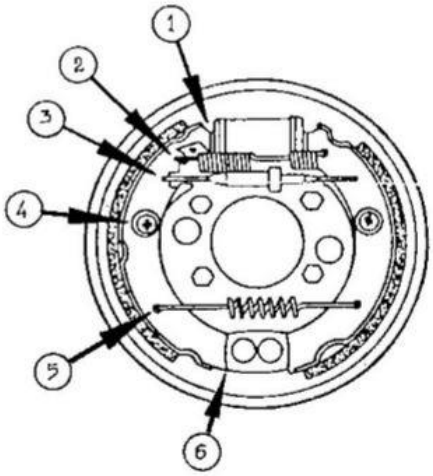


**Gambar 2. 21 Pemeriksaan Fungsi Rem Tromol**

#### 2.2.4.2.3 Pemeriksaan/ Pembersihan Bagian-bagian Rem Tromol

1. Bersihkan bagian-bagian rem dengan kuas atau sikat. Dilarang menggunakan angin, pakai air sabun jika kotor keras.

2. Periksa kondisi dan pemasangan bagian pengikat sepatu rem:

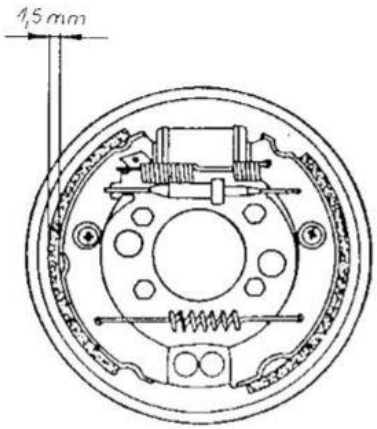



**Gambar 2. 22 Pemeriksaan/Pembersihan Rem Tromol**

Keterangan gambar:

1. Kedudukan ujung sepatu
2. Kedudukan pegas
3. Pemasangan batang penghubung
4. Pengunci sepatu
5. Kedudukan pegas
6. Kedudukan ujung sepatu

a. Periksa tebal kanvas. Jika kurang dari 1,5 mm atau keling kanvas sudah tercoret, kanvas harus diganti baru.

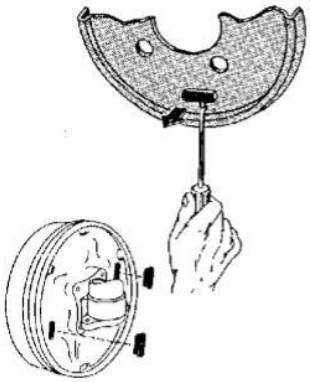



**Gambar 2. 23 Pemeriksaan/Pembersihan Rem Tromol (2)**

b. Periksa permukaan kanvas. Kalau permukaannya keras dan berkilat, nilai geseknya kurang. Kanvas harus digosok atau diganti baru agar tercapai efektifitas rem yang normal.
c. Permukaan kanvas yang kotor karena oli aksel atau cairan rem biasanya diganti baru.
d. Permukaan yang buram atau berkilat lemah menunjukkan kondisi kanvas yang normal. Tidak perlu digosok.
e. Periksa kebocoran pada sil poros aksel (hanya pada aksel rigid dengan penggerak roda). Kebocoran dapat dilihat pada piringan rem dan pada poros aksel yang basah karena oli. Sil yang bocor harus diganti baru.
f. Periksa kebocoran pada silinder rem. Jika ada, semua silinder rem pada aksel yang diperiksa harus dioverhaul atau diganti baru.
g. Untuk memeriksa kebocoran, lihat juga bagian dalam karet pelindung debu silinder rem.

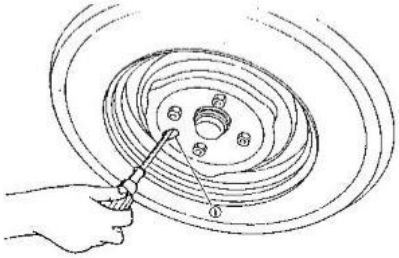
#### 2.2.4.2.4 Penyetelan Sepatu Rem

Pada sistem pengingkatan tromol dengan flens, roda harus dipasang untuk mendapat hasil penyetelan yang baik, (Jika roda tidak terpasang, tromol tertekan teratur pada flensnya) Penyetelan rem biasanya dapat dilakukan melalui lobang paa piringan rem. Lubang-lubang tersebut biasanya tertutup dengan karet.



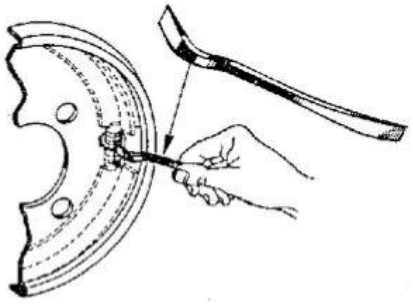
**Gambar 2. 24 Penyetelan Sepatu Rem**

Juga ada mobil dengan lubang penyetel pada tromol (misal: VW, Suzuki). Pada sistem ini, roda harus terpasang dengan posisi lubang pelg pada lubang tromol.



**Gambar 2. 25 Penyetelan Sepatu Rem (2)**

Penyetelan dapat dilakukan dengan obeng, tetapi sering lebih sederhana dengan alat khusus atau obeng yang dibengkokkan sesuai dengan keperluan.

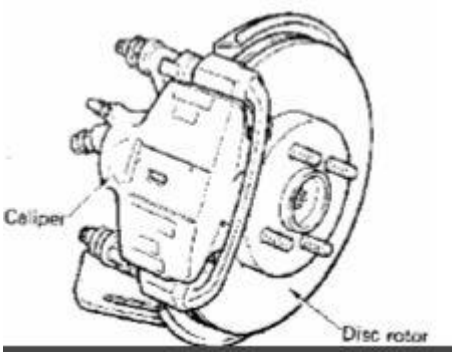


**Gambar 2. 26 Penyetelan Sepatu Rem (3)**

### 2.2.4.3 Rem Cakram

Rem cakram (*disc brake*) pada dasarnya terdiri dari cakram yang terbuat besi tuang (*disc rotor*) yang berputar dengan roda dan bahan gesek (dalam hal ini *disc pad*) yang mendorong dan menjepit cakram. Daya pengereman dihasilkan oleh adanya gesekan antara pad dan cakram (*disc*).

Karakteristik dari cakram hanya mempunyai sedikit aksi energi sendiri (*Self-energizing action)*, daya pengereman itu sedikit dipengaruhi oleh fluktuasi koefisien gesek yang menghasilkan ke stabilan tinggi. Selain itu karena permukaan bidang gesek selalu terkena udara, radiasi panasnya terjamin baik, ini dapat mengurangi dan menjamin dari terkena air. Rem cakram mempunyai batasan pembuatan pada bentuk dan ukurannya. Ukuran disc pad agak terbatas, dan ini berkaitan dengan aksi *self– energizing limited*. Sehingga perlu tambahan tekanan hidraulis yang lebih besar untuk mendapatkan daya pengereman yang efisien. Juga pad akan lebih cepat aus dari pada sepatu rem tromol. Tetapi konstruksi yang sederhana mudah pada perawatannya serta penggantian pad.

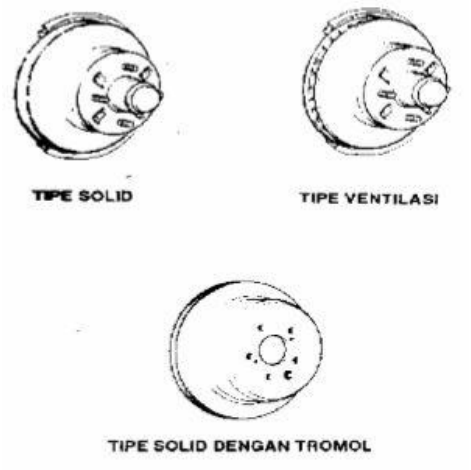



**Gambar 2. 27 Rem Cakram**

#### 2.2.4.3.1 Komponen-Komponen Rem Cakram

1. Piringan

   Umumnya cakram atau piringan (disc rotor) dibuat dari besi tuang dalam bentuk biasa (solid) berlubang-lubang untuk ventilasi. Tipe cakram lubang terdiri dari pasangan piringan yang berlubang untuk menjamin pendinginan yang baik, kedua-duanya untuk mencegah fading dan menjamin umur pad lebih panjang dan tahan lama.

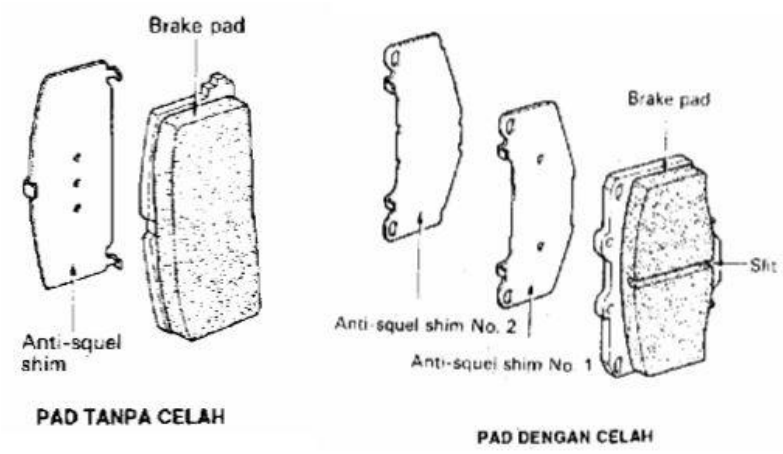



**Gambar 2. 28 Piringan Cakram**

2. Pad Rem

   Pad (*disc pad) biasanya dibuat campuran metalic fiber dan sedikit serbuk besi. Tipe ini disebut dengan "Semi Metalic disc pad".* Pada pad diberi garis celah untuk menunjukkan tebal pad (batas yang diizinkan) dengan demikian dapat mempermudah pengecekan keausan pad. Pada beberapa pad. Penggunaan

metallic plate (disebut dengan anti-squel shim) dipasangkan pada sisi piston dari pad untuk mencegah bunyi saat berlaku pengereman.



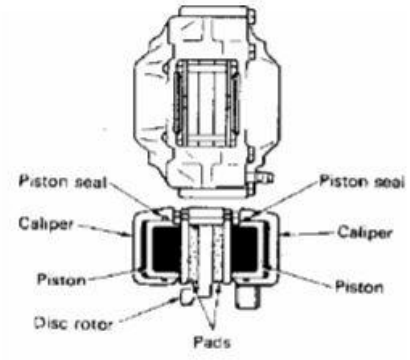

**Gambar 2. 29 Pad Rem**

3. Jenis-Jenis Kaliper

Caliper juga disebut dengan cylinder body, memegang piston-piston dan dilengkapi dengan saluran dimana minyak rem disalurkan ke silinder. Caliper dikelompokkan sebagai berikut menurut jenis pemasangannya:

1. Tipe *Fixed Caliper* (*Double Piston*)

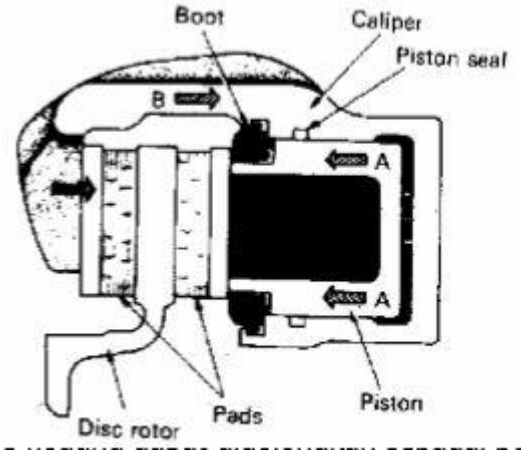
   Caliper dipasangkan tepat pada axle atau strut. Seperti digambarkan di bawah ini, pemasangan caliper dilengkapi dengan sepasang piston. Daya pengereman didapat bila pad ditekan piston secara hidraulis pada kedua ujung piringan atau cakram. Fixed caliper adalah dasar disain yang sangat baik dan dijamin dapat bekerja lebih akurat. Namun demikian radiasi panasnya terbatas karena silinder rem berada antara cakram dan velg, menyebabkan sulit tercapainya pendinginan. Untuk ini membutuhkan penambahan komponen yang banyak. Untuk mengatasi hal tersebut jenis caliper fixed ini, sudah jarang digunakan.




**Gambar 2. 30 Fixed Caliper (Double Piston)**

2. Tipe *Floating Caliper (Single Piston)*

   Seperti terlihat pada gambar piston hanya ditempatkan pada satu sisi kaliper saja. Tekanan hidraulis dari master silinder mendorong piston (A) dan selanjutnya menekan pada rotor disc (cakram) . Pada saat yang sama tekanan hidraulis menekan sisi pad (reaksi B). ini menyebabkan kaliper bergerak ke kanan dan menjepit cakram dan terjadilah usaha tenaga pengereman.




**Gambar 2. 31 Floating Caliper (Single Piston)**

Caliper tipe floating dapat digolongkan sebagai berikut:

1) Tipe *Semi-Floating* Tipe PS

2) Tipe *Full-Floating* (Tipe F, Tipe FS, Tipe AD dan Tipe PD)

Caliper tipe *semi–floating* menerima tenaga pengereman yang dibangkitkan pad bagian luar. Pada caliper tipe full–floating, kemampuannya pengeremannya

dibangkitkan oleh kedua pad dengan troque plate. *Caliper floating* banyak digunakan pada kendaraan penumpang modern.

### 2.2.5 Definisi Mikrokontroler

Menurut Setiawan [5] Mikrokontroler adalah suatu IC dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan untuk suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM/EPROM/PROM/ROM, I/O, Serial & Parallel, *Timer*, *Interupt Controller*.

Menurut Ardi Winoto [6] "*Mikrokontroler* adalah Sebuah sistem microprocessor dimana didalamnya sudah terdapat CPU, ROM, RAM, I/0, clock dan peralatan internal lainnya yang sudah terhubung dan terorganisasi dengan baik oleh pabrik pembuatannya dan dikemas dalam satu chip yang siap pakai, sehingga kita tinggal memprogram isi ROM sesuai dengan aturan penggunaan oleh pabrik pembuatannya".

Berdasarkan definisi yang dikemukakan diatas dapat disimpulkan bahwa mikrokontroller adalah suatu IC yang didesain atau dibentuk dengan kepadatan yang sangat tinggi, dimana semua bagian yang diperlukan suatu kontroler sudah dikemas dalam satu keping, biasanya terdiri dari CPU (*Central Processing Unit*), RAM (*Random Access Memory*), EEPROM/EPROM/PROM/ROM, I/O, Serial & Parallel, *Timer, Interupt Controller* dan berfungsi sebagai pengontrol rangkaian elektronik serta umunya dapat menyimpan program didalamnya.

#### 2.2.5.1 Definisi *Arduino*

*Arduino* merupakan rangkaian elektronik yang bersifat *open* source, serta memiliki perangkat keras dan lunak yang mudah untuk digunakan. *Arduino* dapat mengenali lingkungan sekitarnya melalui berbagai jenis sensor dan dapat mengendalikan lampu, motor, dan berbagai jenis aktuator lainnya. *Arduino* mempunyai banyak jenis, di antaranya *Arduino* Uno, *Arduino* Mega 2560, *Arduino* Fio, dan lainnya.

##### 2.2.5.1.1 *Arduino UNO*

*Arduino* adalah sebuah board mikrokontroller yang berbasis ATmega328. *Arduino* memiliki 14 pin input/output yang mana 6 pin dapat

digunakan sebagai output PWM, 6 analog input, crystal osilator 16 MHz, koneksi USB, jack power, kepala ICSP, dan tombol reset. *Arduino* mampu men-*support* mikrokontroller; dapat dikoneksikan dengan komputer menggunakan kabel USB. [5]




**Gambar 2. 32 *Arduino UNO***

Apakah *Arduino*? Menurut Feri Djuandi [8] *Arduino* adalah merupakan sebuah board minimum system *mikrokontroler* yang bersifat open source. Didalam rangkaian board *Arduino* terdapat *mikrokontroler* AVR seri ATMega 328 yang merupakan produk dari Atmel.

*Arduino* memiliki kelebihan tersendiri disbanding board *mikrokontroler* yang lain selain bersifat open source, *Arduino* juga mempunyai bahasa pemrogramanya sendiri yang berupa bahasa C. Selain itu dalam *board Arduino* sendiri sudah terdapat *loader* yang berupa USB sehingga memudahkan kita ketika kita memprogram *mikrokontroler* didalam *Arduino*. Sedangkan pada kebanyakan *board mikrokontroler* yang lain yang masih membutuhkan rangkaian *loader* terpisah untuk memasukkan program ketika kita memprogram *mikrokontroler*. Port USB tersebut selain untuk *loader* ketika memprogram, bisa juga difungsikan sebagai port komunikasi serial.

*Arduino* menyediakan 20 pin I/O, yang terdiri dari 6 pin input analog dan 14 pin digital input/output. Untuk 6 pin analog sendiri bisa juga difungsikan sebagai output digital jika diperlukan output digital tambahan selain 14 pin yang sudah

tersedia. Untuk mengubah pin analog menjadi digital cukup mengubah konfigurasi pin pada program. Dalam *board* kita bisa lihat pin digital diberi keterangan 0-13, jadi untuk menggunakan pin analog menjadi output digital, pin analog yang pada keterangan board 0-5 kita ubah menjadi pin 14-19. dengan kata lain pin analog 0-5 berfungsi juga sebagi pin output digital 14-16.

Sifat *open source Arduino* juga banyak memberikan keuntungan tersendiri untuk kita dalam menggunakan *board* ini, karena dengan sifat open source komponen yang kita pakai tidak hanya tergantung pada satu merek, namun memungkinkan kita bisa memakai semua komponen yang ada dipasaran.

Bahasa pemrograman *Arduino* merupakan bahasa C yang sudah disederhanakan syntax bahasa pemrogramannya sehingga mempermudah kita dalam mempelajari dan mendalami mikrokontroller.

Tabel 2. 1 Spesifikasi *Arduino UNO*

| Perangkat | Spesifikasi |
|---|---|
| *Mikrokontroler* | ATmega 328 |
| Tegangan Pengoperasian | 5 V |
| Tegangan Input yang disarankan | 7 – 12 V |
| Batas Tegangan Input | 6 – 20 V |
| Jumlah pin I/O Digital | 14 pin digital (6 diantaranya menyediakan keluaran PWM) |
| Jumlah pin input Analog | 6 pin |
| Arud DC tiap pin I/O | 40mA |
| Arus DC untuk pin 3.3 V | 50mA |
| *Memori Flash* | 32 KB (ATmega 328) sekitar 0,5 KB digunakan oleh bootloader |
| SRAM | 2 KB (ATmega 328) |
| EPROM | 1 KB (ATmega 328) |
| *Clock Speed* | 16 MHz |

Pada penelitian ini, peneliti menggunakan *Arduino uno* sebagai pusat pengontrol mobil

#### 2.2.5.2 *Power*

*Arduino* dapat diberikan *power* melalui koneksi USB atau *power supply*. *Power*nya diselek secara otomatis. *Power supply* dapat menggunakan adaptor DC atau baterai. Adaptor dapat dikoneksikan dengan mencolok *jack* adaptor pada koneksi port input *supply*.

*Board Arduino* dapat dioperasikan menggunakan *supply* dari luar sebesar 6 - 20 volt. Jika *supply*kurang dari 7V, kadangkala pin 5V akan menyuplai kurang dari 5 volt dan *board* bisa menjadi tidak stabil. Jika menggunakan lebih dari 12 V, tegangan di regulator bisa menjadi sangat panas dan menyebabkan kerusakan pada *board*. Rekomendasi tegangan ada pada 7 sampai 12 volt.

Penjelasan pada pin power adalah sebagai berikut :

a. Vin

   Tegangan input ke board *Arduino* ketika menggunakan tegangan dari luar (seperti yang disebutkan 5 volt dari koneksi USB atau tegangan yang diregulasikan). Pengguna dapat memberikan tegangan melalui pin ini, atau jika tegangan suplai menggunakan *power jack*, aksesnya menggunakan pin ini.

b. 5V

   Regulasi power *supply* digunakan untuk power mikrokontroller dan komponen lainnya pada board. 5V dapat melalui Vin menggunakan regulator pada board, atau supply oleh USB atau *supply* regulasi 5V lainnya.

c. 3V3

   Suplai 3.3 volt didapat oleh FTDI chip yang ada di board. Arus maximumnya adalah 50mA

d. Pin Ground

   Berfungsi sebagai jalur ground pada *Arduino*

e. Memori

   ATmega328 memiliki 32 KB flash memori untuk menyimpan kode, juga 2 KB yang digunakan untuk bootloader. ATmega328 memiliki 2 KB untuk SRAM dan 1 KB untuk EEPROM.

#### 2.2.5.3 Input dan Output

Setiap 14 pin digital pada *Arduino* dapat digunakan sebagai input atau output, menggunakan fungsi pinMode(), digitalWrite(), dan digitalRead(). Input/output dioperasikan pada 5 volt. Setiap pin dapat menghasilkan atau menerima maximum 40 mA dan memiliki internal pull-up resistor (disconnected oleh default) 20-50K Ohm.

Beberapa pin memiliki fungsi sebagai berikut:

a. Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirim (TX) TTL data serial. Pin ini terhubung pada pin yang koresponding dari USB ke TTL chip serial.
b. Interupt eksternal: 2 dan 3. Pin ini dapat dikonfigurasikan untuk trigger sebuah interap pada low value, rising atau falling edge, atau perubahan nilai.
c. PWM: 3, 5, 6, 9, 10, dan 11. Mendukung 8-bit output PWM dengan fungsi analogWrite().
d. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mensuport komunikasi SPI, yang mana masih mendukung hardware, yang tidak termasuk pada bahasa *Arduino*.
e. LED: 13. Ini adalah dibuat untuk koneksi LED ke digital pin 13. Ketika pin bernilai HIGH, LED hidup, ketika pin LOW, LED mati.

#### 2.2.5.4 Komunikasi

*Arduino* Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, *Arduino* lain, atau *mikrokontroler* lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). *Firmware Arduino* menggunakan USB *driver* standar COM, dan tidak ada *driver* eksternal yang dibutuhkan. Namun, pada Windows, file. Ini diperlukan. Perangkat lunak *Arduino* termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke *board Arduino*. RX dan TX LED di *board* akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke komputer.

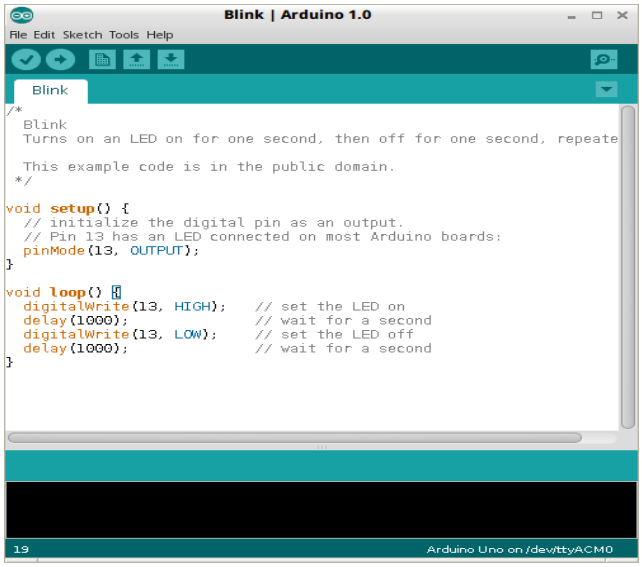
### 2.2.5.5 *Software Arduino*

*Arduino* Uno dapat diprogram dengan perangkat lunak *Arduino*. Pada ATMega328 di *Arduino* terdapat *bootloader* yang memungkinkan Anda untuk meng-*upload* kode baru untuk itu tanpa menggunakan *programmer hardware eksternal*.

IDE *Arduino* adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE *Arduino* terdiri dari:

1. Editor program, sebuah window yang memungkinkan pengguna menulis dan mengeditprogram dalam bahasa *Processing*.
2. *Compiler*, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah *mikrokontroler* tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh *mikrokontroler* adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan *Arduino*.

Sebuah kode program *Arduino* umumnya disebut dengan istilah *sketch*. Kata"*sketch*" digunakan secara bergantian dengan "kode program" dimana keduanya memiliki arti yang sama. [5]




**Gambar 2. 33 Tampilan IDE Arduino Dengan Sebuah Sketch**

### 2.2.6 Sensor *Ultrasonic*

Sensor *Ultrasonic* adalah alat elektronika yang kemampuannya bisa mengubah dari energi listrik menjadi energi mekanik dalam bentuk gelombang suara *Ultrasonic*.

Gelombang *Ultrasonic* adalah gelombang mekanik yang memiliki ciri-ciri longitudinal dan biasanya memiliki frekuensi di atas 20 Khz. Gelombong Utrasonik dapat merambat melalui zat padat, cair maupun gas. Gelombang *Ultrasonic* adalah gelombang rambatan energi dan momentum mekanik sehingga merambat melalui ketiga element tersebut sebagai interaksi dengan molekul dan sifat enersia medium yang dilaluinya. Perbedaaan waktu antara gelombang suara yang dipancarkan dengan ditangkapnya kembali gelombang suara tersebut merupakan representasi jarak. Kecepatan rambat gelombang *Ultrasonic* dipengaruhi oleh beberapa faktor antara lain adalah suhu, tekanan, kelembaban, dan sinyal radio. [9]

Hal yang harus diperhatikan ketika menggunakan sensor *Ultrasonic* terhadap suatu objek yaitu sebagai berikut.

a. Bentuk dan Ukuran Objek

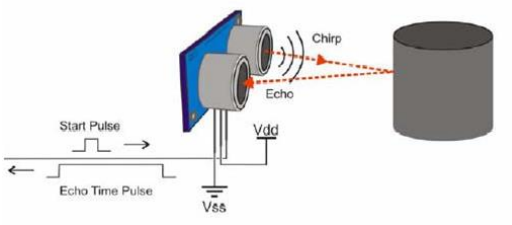
Objek dengan ukuran yang lebih besar akan memantulkan lebih banyak gelombang dibandingkan dengan objek yang berukuran kecil sehingga memberikan hasil yang lebih presisi. Bentuk suatu objek mempengaruhi banyaknya pantulan yang dipantulkan objek. Objek dengan bentuk bulat akan memantulkan gelombang ke segala arah sehingga pantulan yang ditangkap menjadi lemah. Objek dengan bentuk datar akan memantulkan gelombang dengan lebih baik.

b. Jenis Material

Objek padat yang terbuat dari besi atau baja akan memberikan pantulan yang lebih baik dibandingkan dengan objek yang terbuat dari kayu atau gabus.

c. Pola Permukaan

Objek dengan permukaan datar, halus, dan tegak lurus terhadap sinyal *Ultrasonic* yang dipancarkan akan memberikan pantulan yang lebih kuat dibandingkan dengan permukaan tidak rata.




**Gambar 2. 34 Prinsip Kerja Sensor *Ultrasonic***

Pada penelitian ini, peneliti menggunakan sensor *Ultrasonic* sebagai sensor pendeteksi jarak

### *2.2.7 Pulse Width Modulation (PWM)*

PWM merupakan suatu teknik teknik dalam mengatur kerja suatu peralatan yang memerlukan arus pull in yang besar dan untuk menghindari disipasi daya yang berlebihan dari peralatan yang akan dikontrol. PWM merupakan suatu metoda untuk mengatur kecepatan perputaran motor dengan cara mengatur prosentase lebar pulsa high terhadap perioda dari suatu sinyal persegi dalam bentuk tegangan periodik yang diberikan ke motor sebagai sumber daya. Semakin besar perbandingan lama sinyal high dengan perioda sinyal maka semakin cepat motor berputar. Sinyal PWM dapat dibangun dengan banyak cara, dapat menggunakan metode analog menggunakan rankaian op-amp atau dengan menggunakan metode digital. Dengan metode analog setiap perubahan PWM-nya sangat halus, sedangkan menggunakan metode digital setiap perubahan PWM dipengaruhi oleh resolusi dari PWM itu sendiri. Misalkan PWM digital 8 bit berarti PWM tersebut memiliki resolusi 2 pangkat 8 = 256, maksudnya nilai keluaran PWM ini memiliki 256 variasi, variasinya mulai dari 0 – 255 yang mewakili duty cycle 0 – 100% dari keluaran PWM tersebut. Pada perancangan driver ini, sinyal PWM akan diatur secara digital yang dibangkitkan oleh *mikrokontroler*.

PWM digunakan sebagai pengatur kecepatan roda pada *Motor DC*

### 2.2.8 *Bluetooth* HC-05

HC-05 Adalah sebuah modul *Bluetooth* SPP (Serial Port Protocol) yang mudah digunakan untuk komunikasi serial wireless (nirkabel) yang mengkonversi

port serial ke *Bluetooth*. HC-05 menggunakan modulasi *Bluetooth* V2.0 + EDR (Enchanced Data Rate) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz. Modul ini dapat digunakan sebagai slave maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu AT mode dan Communication mode. AT mode berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan Communication mode berfungsi untuk melakukan komunikasi *Bluetooth* dengan piranti lain. Jarak sinyal dari HC-05 adalah 30 meter, dengan kondisi tanpa halangan.

Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan driver khusus. Untuk berkomunikasi antar *Bluetooth*, minimal harus memenuhi dua kondisi berikut :

1. Komunikasi harus antara master dan slave.
2. Password harus benar saat melakukan pairing

*Bluetooth* digunakan sebagai penghubung antara *Arduino* board dan *Android*

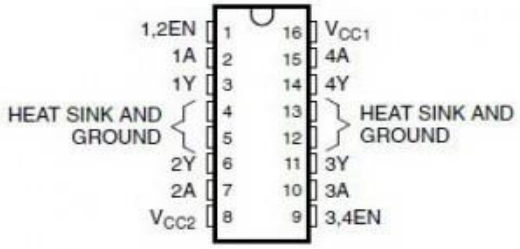


**Gambar 2. 35 Modul Bluetooth**

### 2.2.9 *Driver motor* L293D

IC L293D adlah IC yang didesain khusus sebagai *driver motor DC* dan dapat dikendalikan dengan rangkaian TTL maupun *mikrokontroler*. *Motor DC* yang dikontrol dengan driver IC L293D dapat dihubungkan ke ground maupun ke sumber tegangan positif karena di dalam driver L293D sistem driver yang digunakan adalah totem pool. Dalam 1 unit chip IC L293D terdiri dari 4 buah *driver motor DC* yang berdiri sendiri sendiri dengan kemampuan mengalirkan arus 1 Ampere tiap drivernya. Sehingga dapat digunakan untuk membuat driver H-bridge

untuk 2 buah *Motor DC*. Konstruksi pin *driver motor DC* IC l293D adalah sebagai berikut.




**Gambar 2. 36 Konfigurasi Pin Driver motor L293D**



**Gambar 2. 37 Driver motor L293D**

Fungsi Pin *Driver motor* L293D adalah sebagai berikut:

1. Pin EN (Enable, EN1.2, EN3.4) berfungsi untuk mengijinkan driver menerima perintah untuk menggerakan *Motor DC*.
2. Pin Pin In (Input, 1A, 2A, 3A, 4A) adalah pin input sinyal kendali *Motor DC*
3. Pin Out (Output, 1Y, 2Y, 3Y, 4Y) adalah jalur output masing-masing driver yang dihubungkan ke *Motor DC*
4. Pin VCC (VCC1, VCC2) adalah jalur input tegangan sumber *driver motor DC*, dimana VCC1 adalah jalur input sumber tegangan rangkaian kontrol dirver dan VCC2 adalah jalur input sumber tegangan untuk *Motor DC* yang dikendalikan.
5. Pin GND (Ground) adalah jalu yang harus dihubungkan ke ground, pin GND ini ada 4 buah yang berdekatan dan dapat dihubungkan ke sebuah pendingin kecil.

### 2.2.10 *Motor DC*

*Motor DC* adalah salah satu komponen yang dapat difungsikan sebagai penggerak roda-roda pada model simulasi sistem pengereman mobil. *Motor DC* memiliki fungsi untuk mengubah energi listrik menjadi energi mekanik. Pada

aplikasinya pergerakan pada sebuah mobil listrik umumnya menggunakan *Motor DC* sebagai komponen penggeraknya, karena jenis motor ini lebih mudah untuk dikendalikan. Kecepatan yang dihasilkan oleh *Motor DC* berbanding lurus dengan potensial yang diberikan.



**Gambar 2. 38 Motor DC**

Prinsip kerja *Motor DC* adalah motor yang bekerja atas prinsip dasar kemagnetan. Proses tolak menolak antara kutub magnet yang berseberangan dan perubahan yang cepat pada perubahan kutub akan membuat motor bergerak dengan cepat. Agar proses pergerakan tersebut dapat bergerak cepat, maka tegangan yang disalurkan harus cukup besar. Motor listrik ini menjadi titik tolak pergerakan pada robot. Tegangan yang dimiliki oleh perangkat akhirnya hampir 70% digunakan untuk pergerakan roda ini. Oleh karena itu, penghematan pada pergerakan menjadi sangat penting untuk membuat penggunaan baterai yang terpasang menjadi lebih hemat, dikarenakan pergerakan yang semakin cepat akan semakin jelas menghabiskan daya yang lebih besar. [10]

Pada penelitian ini, peneliti menggunakan *Motor DC* sebagai alat untuk penggerak roda

### 2.2.11 Motor Servo

Servo adalah motor yang mampu bekerja dua arah (*CW* dan *CCW*) dimana arah dan penggerak rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya.



**Gambar 2. 39 Motor Servo**

Jenis – jenis motor servo :

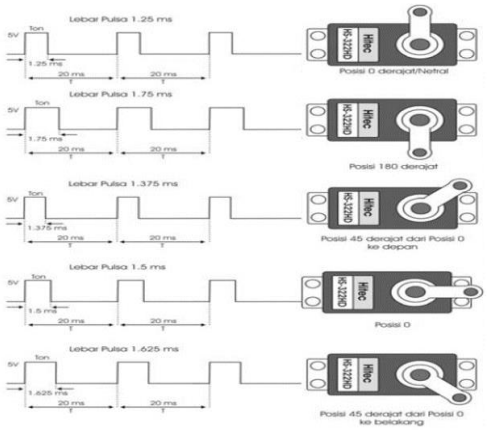
1. Servo standar 180º

   Motor servo jenis ini hanya mampu bergerak dua arah (*CW* dan *CCW*) dengan defleksi masing – masing sudut mencapai 90º sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180º.

2. Servo *continuous* 360º

   Servo ini mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

   Pulsa Kendali Servo:




**Gambar 2. 40 Pulsa Kendali Servo**

Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi *Ton duty cycle* 1.5ms, maka rotor dari motor akan berhenti tepat di tengah – tengah (sudut 0° / netral). Pada saat *Ton duty cycle* dari sinyal yang diberikan kurang dari 1.5ms, maka rotor akan berputar ke berlawanan arah jarum jam (*Counter Clock Wise, CCW*) dengan membentuk sudut yang besarnya linier terhadap besarnya *Ton duty cycle,* dan akan bertahan diposisi tersebut. Dan sebaliknya, jika *Ton duty cycle* dari sinyal yang akan diberikan lebih dari 1.5ms, maka rotor akan berputar searah jarum jam (*Clock Wise, CW*) dengan membentuk sudut yang linier pula terhadap besarnya *Ton duty cycle,* dan bertahan di posisi tersebut. [11]

Pada penelitian ini, peneliti menggunakan motor servo sebagai alat untuk sistem pengereman

### 2.2.12 Aplikasi *Mobile*

Menurut Buyens [12] aplikasi mobile berasal dari kata *application* dan *mobile*. *Application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan *mobile* dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain.

Maka aplikasi *mobile* dapat di artikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah – pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi *mobile* ini dapat di akses melalui perangkat nirkabel, pager, PDA, telepon seluler, *smartphone,* dan perangkat sejenisnya.

### 2.2.13 Karakteristik Perangkat *Mobile*

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki karakteristik yang sangat berbeda dari sistem desktop. Berikut karakteristik perangkat mobile, diantaranya :

a. Ukuran yang kecil

Perangkat mobile memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

b. Memory yang terbatas

Perangkat mobile juga memiliki memory yang kecil, yaitu primary (RAM) dan secondary (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

c. Daya proses yang terbatas

Sistem mobile tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, Anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

d. Mengkonsumsi daya yang rendah

Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

e. Kuat dan dapat diandalkan

Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.

f. Konektivitas yang terbatas

Perangkat mobile memiliki bandwith rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

g. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik dan kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka.

### 2.2.14 *Android*

*Android* merupakan sistem operasi mobile yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainnya seperti Windows Mobile, i-Phone OS, Symbian dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat hardware yang ada. Akan tetapi, sistem operasi yang ada ini berjalan dengan memprioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka.

*Android* menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. *Android* tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga.

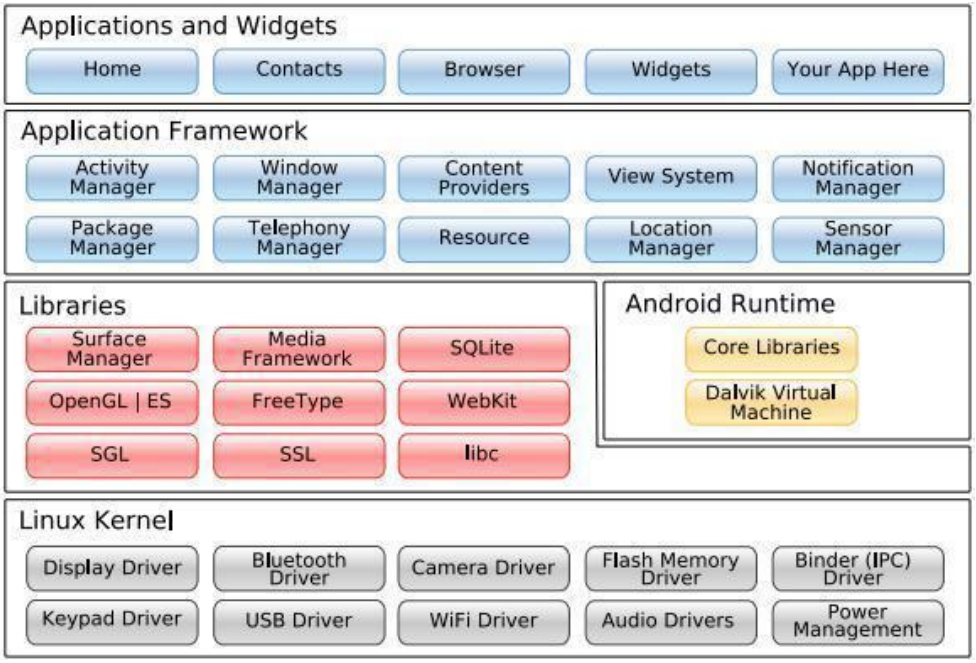
Pada penelitian ini, peneliti menggunakan *Android* sebagai media informasi dari pengujian alat

#### 2.2.14.1 Arsitektur *Android*

Arsitektur *Android* dapat digambarkan seperti pada gambar 1 dan secara garis besar Arsitektur *Android* dapat dijelaskan sebagai berikut:

a. Application dan Widgets

Application dan Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemrograman Java.




**Gambar 2. 41 Arsitektur Android**

b. *Application Framework*

*Android* adalah "*Open Development Platform*"yaitu *Android* menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan service background, dan menambah status notifications, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (reuse). Sehingga bisa kita simpulkan Application Frameworks ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi *Android*, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa sms dan panggilan telepon. Komponen-komponen yang termasuk di dalam Application Frameworks adalah sebagai berikut :

1. *Views*
2. *Content Provider*

3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager* [13]

c. *Libraries*

Libraries ini adalah layer dimana fitur-fitur *Android* berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan di atas Kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc SSL, serta :

a. *Libraries* media untuk pemutaran media audio dan video
b. *Libraries* untuk manajemen tampilan
c. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
d. *Libraries* SQLite untuk dukungan database
e. *Libraries* SSL dan WebKit terintegrasi dengan web browser dan security
f. *Libraries LiveWebcore* mencakup modern web browser dengan engine embedded webview
g. *Libraries* 3D yang mencakup implementasi OpenGL ES1.0 API's. [13]

d. *Android Run Time*

Layer yang membuat aplikasi *Android* dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. Dalvik Virtual Machine (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Android*. Di dalam *Android* Run Time dibagi menjadi dua bagian yaitu:

a. Core Libraries: Aplikasi *Android* dibangun dalam bahasa Java, sementara Dalvik sebagai virtual mesinnya bukan Virtual Machine Java, sehingga diperlukan sebuah libraries yangberfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh Core Libraries.
b. Dalvik Virtual Machine: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien,

dimana merupakan pengembangan yang mampu membuat Linux Kernel untuk melakukan threading dan manajemen tingkat rendah.

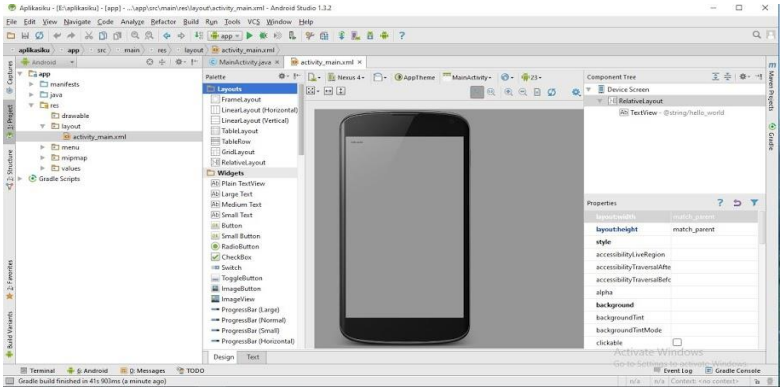
e. Linux Kernel

Linux Kernel adalah layer dimana inti dari system operasi *Android* itu berada. Berisi file-file system yang mengatur sistem processing, memory, resource, drivers, dan sistem-sistem operasi *Android* lainnya. Linux Kernel yang digunakan *Android* adalah Linux Kernel release 2.6.

### 2.2.14.2 *Android Studio*

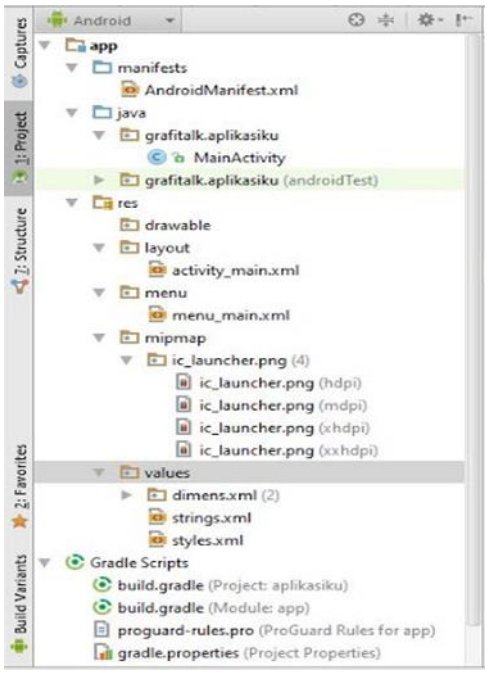
*Android studio* adalah lingkungan pengembangan terpadu (*IDE)* resmi untuk pengembangan platform *Android*,hal itu disampaikan pada tanggal 16 mei 2013 Google I/O,*Android* studio sudah tersedia secara bebas dibawah lisensi Apache 2.0.

*Android* studio pada awalnya tahap preview versi 0.1 yang dipakai pada tanggal 1 mei 2013 dan memasuki tahap beta pada bulan juni 2014 dan mulai stabil dirilis pada Desember 2014 dengan versi 1.0 ,berdasarkan jetBrains 'IDEA Intellij *Software*, *Android* studio dirancang khusus untuk pengembangan *Android* yang tersedia untuk Windows,Mac OS X dan Linux sebagai pengganti Eclipse.



**Gambar 2. 42 Tampilan Android Studio**

### 2.2.14.2.1 Struktur *Android Studio*




**Gambar 2. 43 Struktur Android Studio**

Berikut rincian dari struktur *Android* Studio:

a. App: berfungsi untuk menyimpan file file pendukung aplikasi yang akan diolah.

1. *Manifest*

Tempat dimana konfigurasi konfigurasi pendukung project serta izin (permision) untuk mengakses hardware.

2. Java

Tempat berdiamnya code/scripts java untuk menjalankan aplikasi yang akan dibuat

3. *Res*:

a. Drawable: tempat untuk menyimpan gambar atau icon pendukung aplikasi.

b. *Layout*: tempat berkumpulnya design tampilan yang akan tampak pada tampilan aplikasi anda.

c. *Menu* : berisi code berupa file xml yang didalamnya berisi array tentang menu yang akan tampil dalam project anda, misalkan file, new,open,close,dsb.

d. *Mipmap* : tempat berkumpulnya icon launcher (icon utama aplikasi) yang ukurannya menyesuaikan device yang digunakan.

e. *Values* : untuk menyimpan file .xml yang menggambarkan sumber dari berbagai jenis. Sebagai contoh folder berisi file strings.xml dengan definisi sumber string. Seperti yang akan anda lihat dari menambahkan bagian color , folder tata letak juga dapat berisi, misalnya, deskripsi warna. Di dalam file dimens.xml, terdapat string-string untuk konfigurasi dimensi aplikasi. Tidak jauh berbeda dengan dimens.xml maupun strings.xml, pada file styles.xml terdapat deklarasi variable untuk theme.

b. *Gradle scripts* : berfungsi sebagai configurasi aplikasi dengan device, serta menghubungkan library pendukung aplikasi yang dibuat.

### 2.2.15 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientas objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek. [14]

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekata terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini beragam dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi

tersebut. Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

1. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (reusable).

2. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengodean.

3. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.

4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunai nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

### 2.2.16 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dan abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi suatu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancanggan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu. [14]

Sistem berorientasi objek merupakan sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok daata dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya, dan dapat berinteraksi satu sama lain.

Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek:

1. Kelas (*Class*)

   Kelas adalah sekumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (metode/operasi), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru. Secara tekniks kelas adalah sebuah struktur dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek.

2. Objek (*object*)

   Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata benda, manusia, satua organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstra. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan. Secara teknis, sebuah kelas saat program dieksekusi makan akan dibuat sebuah objek. Objek dilihat darisegi teknis adalah elemen pada saat runtime yang akan diciptakan, dimanipulasi, dan dihancurkan saat eksekusi sehinga

sebuah objek hanya ada saat sebuah program dieksekusi. Jika masih dalam bentuk kode, disebut sebagai kelas jadi pada saat runtime (saat sebuah program dieksekusi), yang kita punya adalah objek, di dalam teks program yang kita lihat hanyalah kelas.

3. Metode (*method*)

   Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari event, aktifitas atau aksi keadaan, fungsi, atau kelakuan dunnia nyata. Contoh metode atau operasi misalnya *Read*, *Write*, *Move*, *Copy*, dan sebagainya.

4. Atribut (*attribute*)

   Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat beruoa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya berat, jenis, nama, dan sebagainya.

5. Abstraksi (*abstraction*)

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulation*)

   Pembungkusa atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerja-nya.

7. Pewarisan (*inheritance*)

   Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Antarmuka (*interface*)

   Antarmuka sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain.

9. *Reusability*

   Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

    Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalnya kelas yang lebih umum (generalisasi) adalah kendaraan darat dan kelas khususnya (spesialisasi) adalah mobil, motor, dan kereta.

11. Komunikasi Antarobjek

    Komunikasi antarobjek dilakukan lewat pesan (message) yang dikirim dari satu objek ke objek lainnya.

12. Polimorfisme (*polymorphism*)

    Kemampuan suatu objek digunakan di banyak tujuan yang berbeda dengan nama yang sehingga menghemat baris program.

13. *Package*

    Package adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan beberapa kelas yang bernama sama disimpan dalam package yang berbeda.

### *2.2.17 Unified Modeling Language (UML)*

Analisis dan desain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Pengertian berorientasi objek berarti bahwa cara mengorganisasikan perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data perilakunya.

Konsep OOAD (*Object Oriented Analysis Design*) mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu:

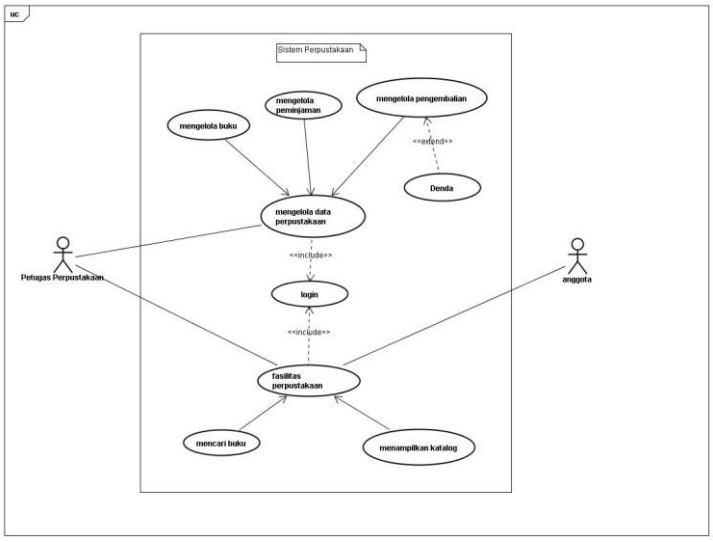
1. Analisis berorientasi objek (OOA)

Metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. Unified Modeling Language (UML) adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasi artefacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak), seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML merupakan bahasa standar untuk penulisan blueprint *software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat. Berikut ini merupakan beberapa bagian dari UML adalah sebagai berikut:

*a. Use Case Diagram*

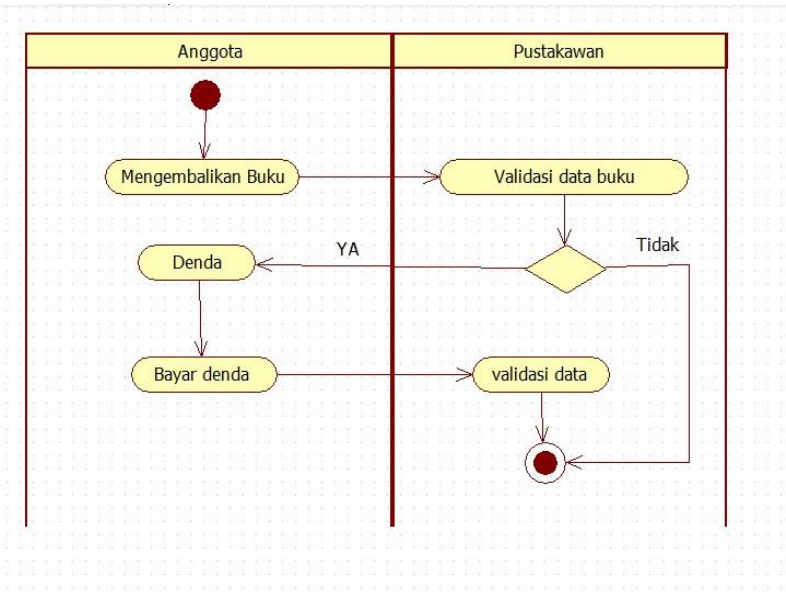
*Use Case diagram* menggambarkan sejumlah eksternal aktor dan hubungannya ke *Use Case*yang diberikan oleh sistem.




**Gambar 2. 44 Contoh Use CaseDiagram**

b. *Activity Diagram*

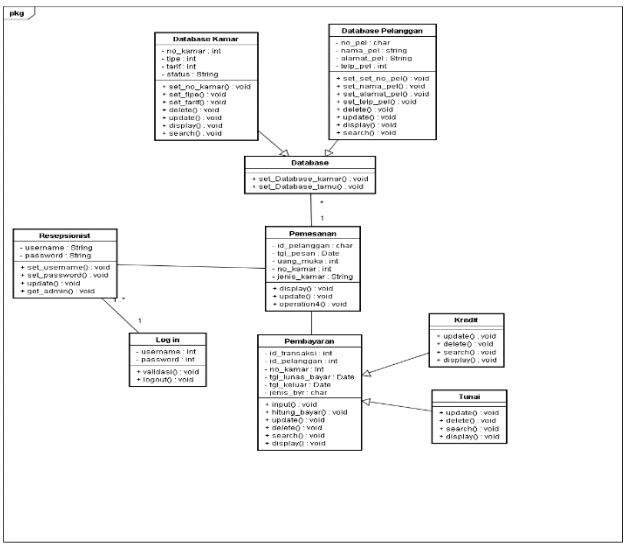
*Activity diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainya seperti *use case*.




**Gambar 2. 45 Contoh Activity Diagram**

c. *Class Diagram*

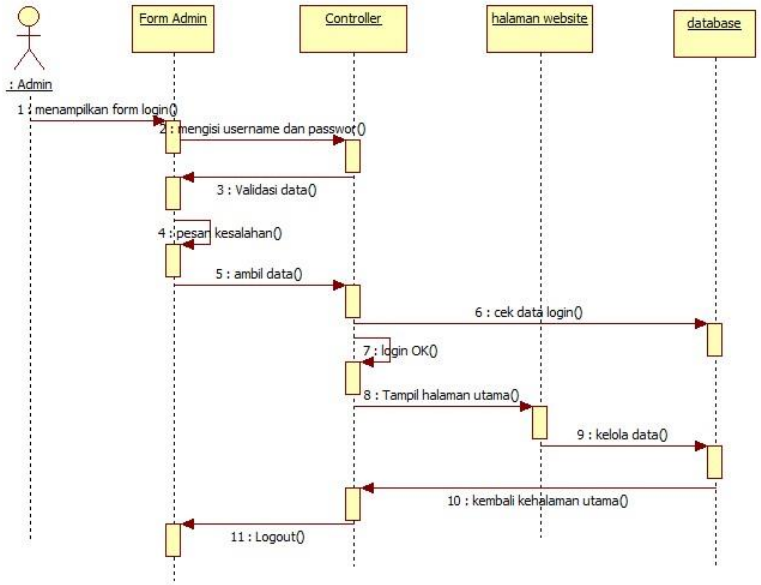
*Class diagram* menggambarkan struktur dan deskripsi *Class*, package (paket) dan objek beserta hubungan satu sama lain seperti containment penahanan), pewarisan, asosiasi dan lain-lain.




**Gambar 2. 46 Contoh Class Diagram**

*d. Sequence Diagram*

*Sequence Diagram* menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.




**Gambar 2. 47 Contoh Sequence Diagram**