

BAB II

TINJAUAN PUSTAKA

2.1 Saluran Transmisi

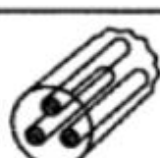
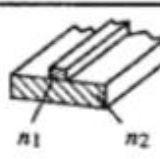
Saluran transmisi adalah penghantar, baik berupa konduktor ataupun isolator atau dielektrik, yang digunakan untuk menghubungkan suatu pembangkit sinyal, disebut juga sumber, dengan sebuah penerima/pemakai, atau disebut juga beban. Karena sinyal elektrik merambat hanya dengan kecepatan cahaya, maka sinyal elektrik juga memerlukan suatu waktu tempuh tertentu untuk merambat dari sumber ke beban.

Jika sinyal elektrik ini berubah secara cepat dengan waktu (berfrekuensi tinggi), waktu tempuh di atas menjadi signifikan. Waktu tempuh (*delay*) yang terjadi harus diperhatikan, sinyal yang keluar dari satu saluran transmisi tidaklah sama dengan apa yang dimasukkan pada bagian masukannya. Selain itu saluran transmisi yang dipergunakan biasanya mengandung kerugian, sehingga sinyal yang masuk akan mengalami peredaman (*attenuation*) dalam perambatannya, amplitudo sinyal yang melalui saluran transmisi yang mengandung kerugian itu akan mengecil (*lossy transmission line*).

Dua persoalan di atas akan menjadi lebih kompleks, jika kecepatan rambat dan peredaman tersebut merupakan fungsi dari frekuensi. Hal yang disebut dispersi ini akan menyebabkan terjadinya perubahan pada bentuk sinyal (distorsi sinyal). Kebanyakan fenomena yang bisa ditemukan di teknik elektro dan elektronika bisa dijabarkan dengan bantuan model atau rangkaian pengganti yang bersifat dekonsentrasi pada suatu elemen *single* tertentu (*lumped element*). Hal ini boleh dilakukan dan juga sebaiknya dilakukan, jika panjang gelombang dari sinyal yang diamati jauh lebih besar dari geometri rangkaian yang diamati. (Mudrik, 2009)

2.1.1 Jenis-Jenis Media Saluran Transmisi

Tabel 2.1: Jenis Saluran, Bentuk Dasar, Penggunaan, Serta Frekuensi Kerja
(Sumber: Mudrik, 2009)

Tipe	Bentuk	Penggunaan	Besaran Sinyal	Frekuensi kerja
Kabel paralel ganda	 <i>unshielded</i>  <i>shielded</i>	Transmisi energi dan sinyal	Menggunakan arus dan tegangan	Frekuensi kerja dari nol sampai tak terhingga. Tetapi ke atas dibatasi oleh peredaman yang sangat tinggi dan munculnya <i>higher order modes</i> (gelombang pengganggu)
Kabel koaxial		Transmisi energi dan sinyal serta elemen rangkaian		
Penghantar pipih (<i>strip line</i>)		Transmisi sinyal dan elemen rangkaian		
Penghantar multi konduktor (tiga fasa)		Transmisi energi		
Waveguide		Transmisi sinyal dan elemen rangkaian	Medan listrik dan magnet. Arus dan tegangan tak bermanfaat	Mulai frekuensi cut-off tertentu
Penghantar dielektris pipih		Elemen rangkaian dan penyambung optik	Medan listrik dan magnet. Arus dan tegangan tak bermanfaat	Secara teoretis untuk semua frekuensi

Tabel 2.1 menunjukkan beberapa tipe saluran transmisi yaitu ada empat tipe yang pertama, kabel paralel ganda, kabel koaxial, pengantar pipih, dan penghantar multi konduktor, memiliki jumlah konduktor yang lebih dari satu. Penghantar ini mampu untuk melewatkan sinyal yang berfrekuensi nol (sinyal arus searah). Jika frekuensi sinyal yang melaluinya masih cukup kecil, atau dengan kata lain panjang gelombangnya masih cukup besar dibanding dengan dimensi penampangnya, sinyal merambat di saluran transmisi ini dengan modulus dasar (*fundamental mode*). Tetapi jika saluran transmisi ini cukup besar, sehingga panjang gelombang sinyal sebanding dengan atau bahkan lebih kecil dari dimensi saluran transmisi, di samping modulus dasar, modulus yang lebih tinggi (*higher mode*) akan mampu merambat, biasanya kondisi ini dihindari. (Mudrik, 2009)

2.2 *Matching Impedance*

Gelombang yang merambat pada saluran transmisi yang panjangnya tak berhingga, tidak akan mempengaruhi apa yang ada di ujung saluran. Perbandingan antara tegangan dan arus di ujung masukan saluran sesungguhnya dapat dianggap sama dengan perbandingan antara tegangan dan arus setelah mencapai ujung lainnya. Arus dan tegangan di antara kedua kawat penghantar saluran itu memandang saluran transmisi sebagai suatu impedansi. Impedansi inilah yang disebut "Impedansi Karakteristik (Z_0)"

Impedansi karakteristik adalah impedansi yang diukur diujung saluran transmisi yang panjangnya tak berhingga. Bila daya dirambatkan pada saluran transmisi dengan panjang tak berhingga, maka daya itu akan diserap seluruhnya di sepanjang saluran sebagai akibat bocornya arus pada kapasitansi antar penghantar dan hilangnya tegangan pada induktansi saluran.

Besar impedansi karakteristik suatu saluran transmisi maupun bumbung gelombang berbeda-beda dan nilainya ditentukan oleh ukuran fisik penampang dan bahan dielektrik yang digunakan sebagai isolator. (Aditya, 2009)

Sebuah sumber listrik dihubungkan dengan saluran transmisi yang memiliki impedansi gelombang Z_0 yang pada ujung saluran transmisi ini dihubungkan sebuah beban dengan nilai impedansinya sama dengan impedansi gelombang saluran

transmisi tersebut $Z_L = Z_0$. Dengan menggunakan hukum Ohm maka perbandingan tegangan dan arus pada posisi beban diberikan persamaan berikut

$$\frac{V_e}{I_e} = Z_l = Z_o \dots\dots\dots(2.1)$$

Dimana:

V_e = Tegangan pada posisi beban (volt)

I_e = Arus pada posisi beban (Ampere)

Z_l = Impedansi Beban (Ohm)

Z_o = Impedansi Karakteristik (Ohm)

Konstanta integrasi adalah

$$V_1 = \frac{1}{2} (V_e + Z_o \cdot I_e) \cdot E^{\tau L} \dots\dots\dots(2.2)$$

$$V_2 = \frac{1}{2} (V_e - Z_o \cdot I_e) \cdot E^{-\tau L} \dots\dots\dots(2.3)$$

Sehingga

$$V(z) = V_e \cdot e^{\tau(L-z)} = V_e \cdot e^{\tau L} \cdot e^{-\tau z}, \text{ dan } \dots\dots\dots(2.4)$$

$$I(z) = \frac{V_e}{Z_o} \cdot e^{\tau(L-z)} = \frac{V_e}{Z_o} \cdot e^{\tau L} \cdot e^{-\tau z} \dots\dots\dots(2.5)$$

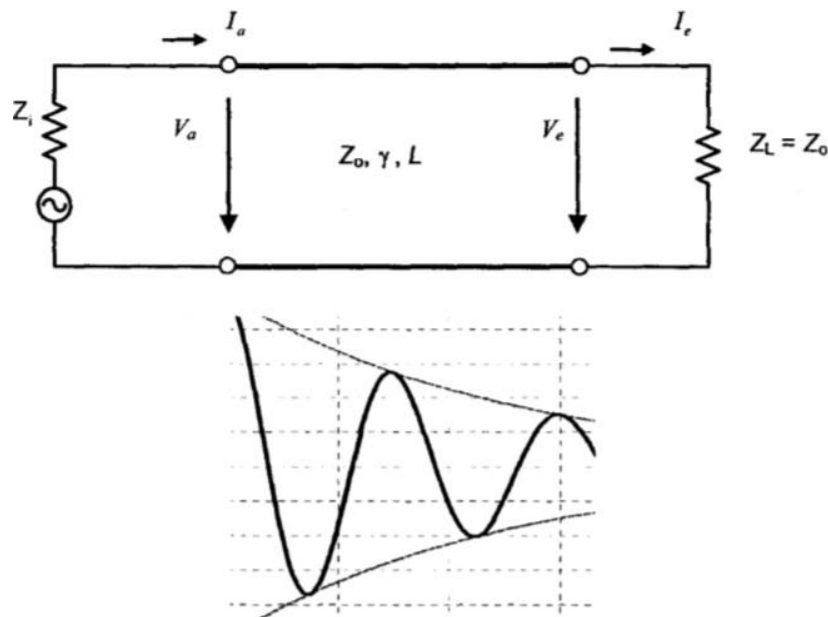
Dimana:

V_1 = Tegangan Integrasi positif (volt) (kompleks)

V_2 = Tegangan Integrasi negatif (volt) (kompleks)

$V(z)$ = Tegangan Impedansi (volt)

$I(z)$ = Arus Impedansi (volt)



Gambar 2.1: Saluran Transmisi dan *Impedance Matching*

(Sumber: Mudrik, 2009)

Selain persamaan diatas dapat digunakan alternatif persamaan berikut

$$V(z) = V_a \cdot e^{-\tau z} \text{ dan } \dots\dots\dots(2.6)$$

$$I(z) = \frac{V_a}{Z_o} \cdot e^{-\tau z} \dots\dots\dots(2.7)$$

Dimana:

V_a = Tegangan pada awal saluran (volt)

Gelombang ini merambat keluar dari sumber energi. Impedansi beban menyerap semua energi yang data kepadanya tanpa sedikitpun terpantulkan. Gelombang yang merambat di dalam saluran transmisi, selalu melihat impedansi yang besarnya adalah impedansi gelombang dari saluran itu, yang secara matematis diformulasikan dengan perbandingan tegangan arus di sana adalah Z_o . Jika pada ujung akhir saluran transmisi dipasangkan sebuah impedansi beban yang besarnya sama dengan besar impedansi saluran transmisi, maka gelombang tetap melihat impedansi yang sama pada ujung akhir saluran tersebut, gelombang itu melihat beban sebagai keberlanjutan dari saluran transmisi, sehingga saluran transmisi memiliki panjang yang tidak terhingga dan semua energi tersalurkan ke sana.

Berikut adalah perbandingan gelombang pantul dan gelombang datang di sepanjang saluran transmisi

$$\frac{V_-}{V_+} = \frac{Z_L - Z_o}{Z_L + Z_o} \cdot e^{-2\tau(L-z)} \dots\dots\dots(2.8)$$

Perbandingan tegangan pantul dan tegangan datang pada setiap titik z didefinisikan sebagai faktor refleksi di posisi tersebut

$$\Gamma(z) = \frac{V_-}{V_+} = \frac{Z_L - Z_o}{Z_L + Z_o} \cdot e^{-2\tau(L-z)} = r \cdot e^{-2\tau(L-z)} \dots\dots\dots(2.9)$$

Dimana:

$\Gamma(z)$ = Faktor Refleksi di posisi z

Sedangkan perbandingan gelombang pantul dan gelombang datang pada posisi beban $z = l$ didefinisikan sebagai faktor refleksi dari impedansi beban

$$r = \frac{Z_L - Z_o}{Z_L + Z_o} \dots\dots\dots(2.10)$$

Faktor refleksi adalah besaran yang menyatakan seberapa besar gelombang yang datang akan dipantulkan. Secara umum memiliki nilai kompleks dan bisa dituliskan dengan besar dan phasanya.

$$r = |r| \cdot e^{i\varphi} \dots\dots\dots (2.11)$$

Jika $Z_L = Z_o$, maka $r = 0$ gelombang sama sekali tidak dipantulkan.

Jika saluran transmisi pada ujungnya dibuat hubungan singkat (*Short*) atau $Z_L = 0$ didapatkan $r = -1$.

Jika saluran terbuka (*Open*) atau $Z_L = \infty$ maka $r = 1$.

Pada saluran transmisi yang ujungnya dibuat hubungan singkat (*Short*) dan terbuka (*Open*), seluruh gelombang dipantulkan baik berlawanan fasa (untuk *Short*, sehingga tegangan total pada ujung saluran transmisi menjadi nol) ataupun se-pasha (untuk *open*, tegangan menjadi maksimum pada ujung saluran transmisi, sedangkan arus menjadi nol).

Selain menggunakan faktor refleksi gelombang pantul, ada besaran lain yang dipergunakan untuk mengkuantifikasikan fenomena pantulan ini, yaitu *VSWR* (*Voltage Standing Wave Ratio*), sebagai perbandingan tegangan maksimal dengan tegangan minimal.

$$VSWR = \frac{V_{max}}{V_{min}} = \frac{1+|r|}{1-|r|} \dots\dots\dots (2.12)$$

Atau

$$|r| = \frac{VSWR-1}{VSWR+1} \dots\dots\dots (2.13)$$

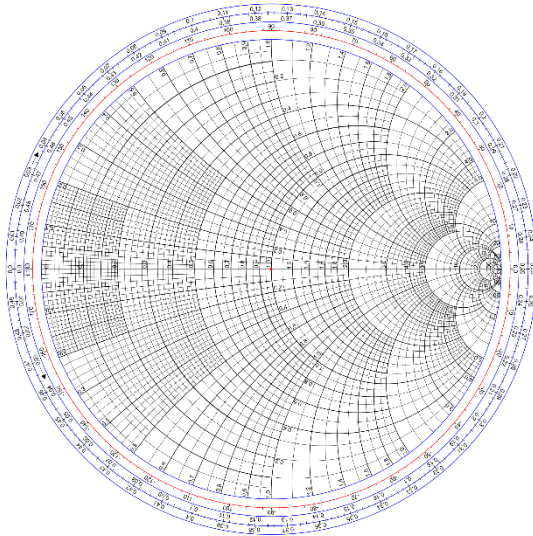
Tabel 2.2: Hubungan Faktor Refleksi dan VSWR

(Sumber: Mudrik, 2009)

$ r $	return loss	VSWR	Keterangan
0	Tak hingga	1	<i>matching</i>
1	0 dB	Tak hingga	Open/short
0,1	20 dB	1,222	Refleksi kecil
0,9	0.92 dB	19	Refleksi besar

$|r|$ mempunyai nilai yang bervariasi dari 0 sampai 1, *return loss* mempunyai nilai minimal 0 dB dan maksimal tak terhingga, sedangkan VSWR mempunyai nilai terendah 1 dan bisa membesar tak terhingga. (Mudrik, 2009)

2.3 Smithchart



Gambar 2.2: Smithchart

(Sumber: Mudrik, 2009)

Diagram Smith (*Smith's Chart*) adalah sebuah cara grafis yang akan mempermudah perhitungan yang berhubungan dengan transformasi impedansi. Dengan Diagram Smith kita bisa secara cepat mendapatkan hasil aproksimasi yang akan membantu memberikan gambaran besar yang akan didapatkan.

Smithchart adalah suatu diagram yang digunakan untuk mengetahui karakteristik dari suatu saluran transmisi. Smithchart dapat digunakan untuk menyeleraskan sebuah alat dengan saluran transmisinya agar semua gelombang dapat terkirim dan tidak ada yang dipantulkan. Dengan smithchart ini kita bisa menemukan panjang *stub* dan admitansinya untuk *me-matching* dengan metode *shunt stub matching* (Mudrik, 2009).

Smithchart digambarkan dengan koefisien refleksi dalam dua dimensi dan skala impedansi dinormalkan. Smithchart Mempunyai sumbu yang dikenal sebagai Z, Y dan YZ, Smithchart yang akan digunakan untuk masalah yang melibatkan setiap karakteristik atau sistem impedansi yang diwakili oleh titik pusat dari grafik. Normalisasi impedansi paling umum digunakan adalah 50 ohm. Setelah jawaban diperoleh melalui pengukuran pada Smithchart, lalu dilakukan konversi antara

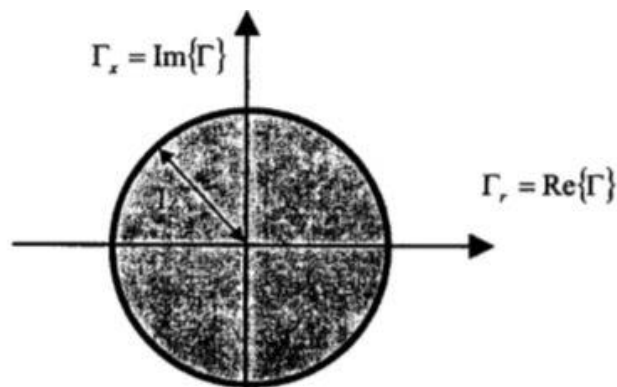
impedansi normalisasi dan nilai imajiner yang sesuai dengan mengalikan dengan impedansi karakteristik. Koefisien refleksi dapat dibaca langsung dari grafik .

Smithchart memiliki skala melingkar dalam panjang gelombang dan derajat. Skala panjang gelombang digunakan dalam masalah komponen didistribusikan dan mewakili jarak diukur sepanjang saluran transmisi dihubungkan antara generator atau sumber dan beban ke titik di bawah pertimbangan. Derajat skala mewakili sudut koefisien refleksi tegangan pada saat itu. Smith grafik juga dapat digunakan untuk pencocokan elemen dan analisis masalah disejajarkan.

Penggunaan Smithchart dan interpretasi hasil yang diperoleh dengan menggunakan memerlukan pemahaman yang baik tentang teori rangkaian AC dan teori saluran transmisi.

Sebagai impedansi dan admitansi berubah terhadap frekuensi, masalah menggunakan Smithchart hanya dapat diselesaikan secara manual menggunakan satu frekuensi pada satu waktu, hasilnya yang diwakili oleh titik. Hal ini sering memadai untuk aplikasi *Narrow Band* (biasanya sampai sekitar 5% sampai 10% *bandwidth*) tapi untuk *bandwidth* yang lebih luas biasanya diperlukan untuk menerapkan teknik Smithchart di frekuensi yang banyak pada pita frekuensi operasi. Asalkan frekuensi yang cukup dekat, yang dihasilkan Smithchart dapat bergabung dengan garis lurus untuk menciptakan sebuah lokus.

Keakuratan Smithchart berkurang untuk masalah yang melibatkan lokus besar impedansi atau admitansi, meskipun skala yang dapat diperbesar untuk daerah masing-masing untuk mengakomodasi ini (Suwanto, 2009).



Gambar 2.3: Cakram Faktor Refleksi Beban Pasif

(Sumber: Mudrik, 2009)

Pada tempat yang sama seperti visualisasi faktor refleksi di atas, akan digambarkan impedansi yang ditransformasikan dari akhir saluran transmisi ke depan, yang berjarak d , dengan persamaan

$$Z(d) = Z_o \frac{1+\Gamma(d)}{1-\Gamma(d)} \dots\dots\dots (2.14)$$

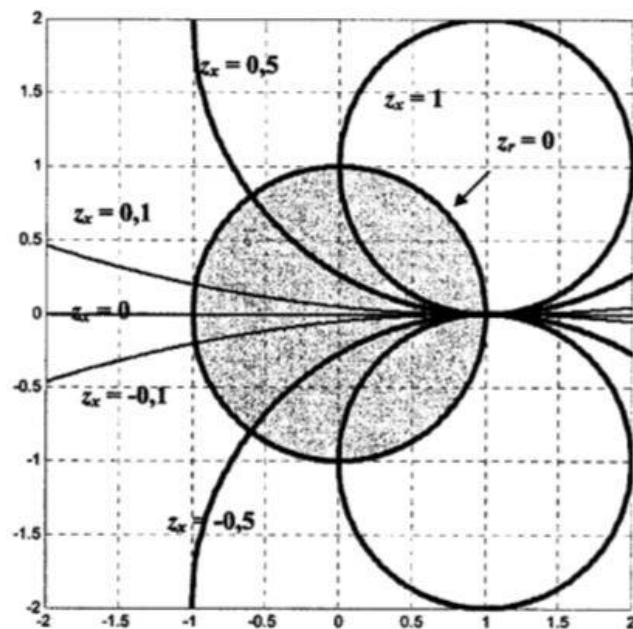
Fungsi $\Gamma(d)$ telah divisualisasikan di gambar 2.3, berikut visualisasi dari fungsi $Z(d)$, yaitu yang dinormalisasi dengan impedansi gelombangnya Z_o :

$$z(d) = \frac{Z(d)}{Z_o} = \frac{1+\Gamma(d)}{1-\Gamma(d)} \dots\dots\dots (2.15)$$

Jika impedansi ternormalisasi itu bisa dituliskan dalam bentuk koordinat kartesian, komponen riilnya sebagai resistansi ternormalisasi Z_r dan komponen imajineranya sebagai reaktansi ternormalisasi Z_x , dan dengan $\Gamma(d) = \Gamma_r + j\Gamma_x$ maka

$$Z_r = \frac{1-\Gamma_r^2-\Gamma_x^2}{(1-\Gamma_r^2)+\Gamma_x^2} \dots\dots\dots (2.16)$$

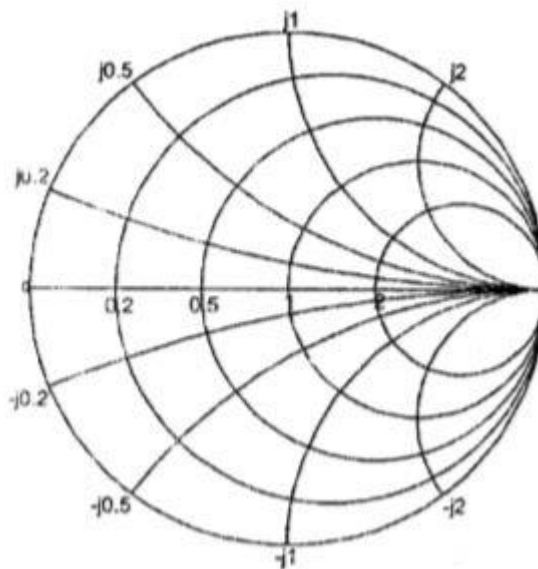
$$Z_x = \frac{2\Gamma_x}{(1-\Gamma_r^2)+\Gamma_x^2} \dots\dots\dots (2.17)$$



Gambar 2.4: Lingkaran Untuk Beberapa Nilai Z_x

(Sumber: Mudrik, 2009)

Secara lengkap sebuah Smithchart akan mempunyai bentuk seperti gambar di bawah berikut ini, sebagai kombinasi dari kurva-kurva dengan komponen riil yang konstan dan juga kurva-kurva dengan komponen imajiner yang konstan.



Gambar 2.5: Smithchart sebagai kombinasi lingkaran untuk Z_r dan Z_x konstan
(Sumber: Mudrik, 2009)

Nilai atau besar faktor refleksi bisa dihitung dengan membandingkan jarak titik tengah ke titik impedansi beban dengan radius pada smithchart yang pada bagian pinggirnya memiliki nilai faktor refleksi yaitu 1. Smithchart juga dapat menentukan impedansi masukan dari sebuah impedansi beban yang ditransformasikan ke depan dari sebuah saluran transmisi.

2.4 Gamemaker Studio



Gambar 2.6: Logo Game Maker Studio

(Sumber: <http://yoyogames.com>)

Game Maker Studio adalah sebuah *software* pembuatan sistem game yang dibuat oleh Mark Overmars dengan Bahasa pemrograman Delphi.

Game Maker dapat mengakomodasi pembuatan video game dalam berbagai platform dan berbagai genre menggunakan Bahasa pengkodean secara manual dan dapat dipelajari dengan mudah dikenal dengan Bahasa GameMaker, yang mana Bahasa pemrograman ini dapat digunakan untuk membuat tidak hanya game yang canggih, *software* ini juga dapat membuat aplikasi aplikasi pembantu yang diinginkan dengan menggunakan fitur fitur yang telah disediakan. GameMaker telah didesain untuk mengajarkan kepada para pemrogram Bahasa computer yang masih awam untuk dapat menggunakan *software* ini dengan mudah tanpa tahu banyak tentang pemrograman dengan menggunakan fitur tersebut, (<http://yoyogames.com>).

Dikarenakan mudahnya bahasa pemrograman yang digunakan pada *software* ini dan fitur fitur yang telah disediakan, *software* ini dapat membuat aplikasi pada *platform* android, maka dari itu *software* ini adalah yang paling pas dipilih untuk membuat aplikasi kali ini.

2.5 Java



Gambar 2.7: Logo Java

(Sumber: <http://java.sun.com>)

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya di kompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

2.6 Eclipse



Gambar 2.8: Logo Eclipse

(Sumber: <http://eclipse.org>)

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform independent). Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X. Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya. Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya. Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan *open source*. *Open source* berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut *plugin* (Dimarzio, 2008)

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak pengembangan IBM Visual Age for Java 4.0. Produk Eclipse ini diluncurkan oleh IBM pada tanggal 5 November 2001. IBM menginvestasikan US\$ 40 juta untuk pengembangannya. Sejak 5 November 2001, konsorsium Eclipse Foundation mengambil alih pengembangan Eclipse lebih lanjut (Dimarzio, 2008)

2.7 Android Studio



Gambar 2.9: Logo Android Studio

(Sumber: <https://wikipedia.org/>)

Android Studio merupakan lingkungan pengembangan Android terbaru berdasarkan IntelliJ IDEA untuk pembuatan aplikasi android. Mirip dengan Eclipse dengan ADT. Android Studio menyediakan alat pengembang yang lebih maju dan terintegrasi untuk pengembangan dan *debugging*.

2.8 Android



Gambar 2.10: Logo Android

(Sumber: <http://wikipedia.org/>)

Android, adalah sebuah sistem yang berbasis sistem operasi Java yang berjalan pada *Kernel* Linux 2.6. sistem ini sangat ringan dan penuh dengan fitur fiturnya.

Aplikasi Android di buat menggunakan Java dan dapat di hubungkan ke semua *platform* secara mudah. Fitur lainnya dari sebuah Android termasuk mesin grafik 3D (berbasis perangkat keras), dukungan database didukung oleh SQLite, dan sebuah peramban internet mutakhir, peletakan *User Interface* dapat di atur secara langsung melalui kode program. Android dapat mengenali dan memperbolehkan pembangunan tampilan untuk pengguna dan juga dapat mendukung tampilan pengguna berbasis XML. *Layout XML* adalah konsep baru kepada para pengembang android. Fitur lainnya yang tidak kalah pentingnya yaitu adalah arsitekturnya, aplikasi pihak ketiga yang termasuk juga aplikasi buatan rumah (*home grown*) di eksekusi di dalam prioritas sistem yang sama seperti yang di masukkan ke dalam sistem inti.

Selanjutnya ada dukungan SDK (*Software Development Kit*) dan dukungan pustaka yang telah di sediakan sedemikian rupa untuk pengembangan lebih lanjut , fitur yang paling menyenangkan pada pengembang Android adalah sekarang dapat mengakses apapun yang dapat di akses oleh sistem operasi, contohnya menelepon atau membuat aplikasi yang menggunakan GPS (jika terpasang). Potensi untuk para pengembang membuat aplikasi yang dinamis sekarang terbuka lebar (Dimarzio, 2008).

2.9 Android Jellybean

Android Jellybean mempunyai tiga versi peluncuran, berikut versi-versi tersebut dan penjelasannya.

2.9.1 Android Jellybean Versi 4.1



Gambar 2.11: Logo Android Jellybean Pertama

(Sumber: <http://www.doenggaloutletcommunity.org/>)

Jelly Bean-Android versi 4.1 yang diluncurkan pada acara Google I/O membawa fitur-fitur baru yang menawan, beberapa fitur yang diperbaharui dalam sistem operasi ini antara lain, pencarian dengan menggunakan *Voice Search* yang lebih cepat, informasi cuaca, lalu lintas, hasil pertandingan olahraga yang cepat dan tepat, selain itu versi 4.1 ini juga mempunyai fitur *keyboard* virtual yang lebih baik. Permasalahan umum yang sering ditemui pengguna Android adalah baterai, namun Baterai dalam sistem Android Jelly Bean versi 4.1 ini diklaim cukup hemat (Dugala, 2015)

2.9.2 Android Jellybean Versi 4.2



Gambar 2.12: Logo Android Jellybean Kedua

(Sumber: <http://www.doenggalroutletcommunity.org/>)

Android versi 4.2 ini merupakan versi terbaru dari versi-versi Android sebelumnya. Jelly Bean V.4.2 diklaim lebih pintar dan inovatif dibandingkan dengan pendahulunya. Beberapa fitur yang diperbaharui dalam sistem operasi ini antara lain *Notifications*, *Google Assistant*, *Face unlock* dengan *Liveness Check*, *Barrel Roll*, *Smart Widget*, *Google Now*, *Technologi Project Butter* untuk meningkatkan responsi yang sangat baik, Sistem operasi yang cepat dan Ringan, *Full Chrome browser* menjelajah internet menggunakan *Google Chrome* seperti pada perangkat komputer (Dugala, 2015)

2.9.3 Android Jellybean Versi 4.3



Gambar 2.13: Logo Android Jellybean Ketiga

(Sumber: <http://www.doenggaloutletcommunity.org/>)

Pada tanggal 22 Agustus 2013 Jellybean 4.3 telah resmi dirilis. Jellybean 4.3 lebih fokus pada pembaruan minor. Nexus 7 adalah Ponsel Pintar pertama dengan sistem operasi Android Jellybean 4.3. Beberapa fitur menarik dari sistem ini antara lain :

1. Terdapat dukungan multi user dengan *Restricted Profiles*, fitur ini memungkinkan bagi administrator untuk membuat lingkungan yang berbeda bagi setiap user, sehingga bisa mengontrol penuh penggunaan aplikasi yang tersedia untuk profil user yang dibuat. Ini sangat efektif jika pengguna memiliki anak dan membatasi penggunaan aplikasi Android.
2. Hadir dengan teknologi *Smart Bluetooth*, penggunaan *bluetooth* tanpa khawatir menghilangkan banyak daya.
3. Mendukung Open GL ES 3.0, performa grafis yang lebih bagus dan realistis.
4. Modular DRM Framework, ini berguna bagi pengembang untuk mengintegrasikan hak digital menjadi *streaming protocol*. Selain itu, Android Jelly bean 4.3 juga mempunyai performa yang bagus dibandingkan dengan pendahulunya.