

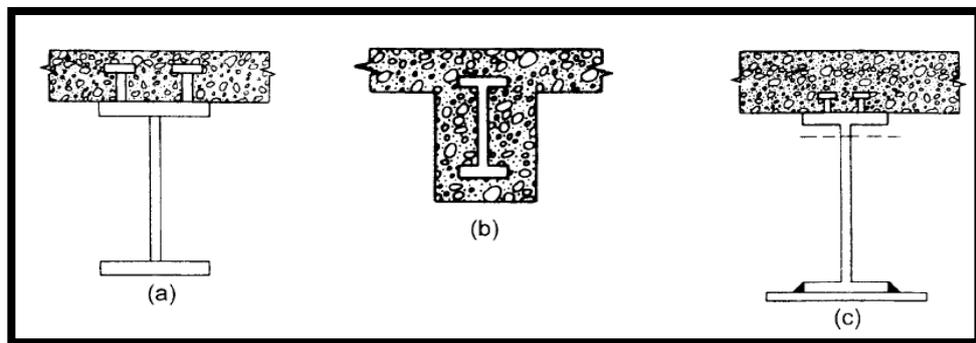
## **II. TINJAUAN PUSTAKA**

### **A. Struktur Komposit**

Penggunaan balok baja untuk menompang suatu pelat beton telah di temukan sejak lama. Namun pada saat itu pelat beton dan balok baja tidak dihubungkan dengan suatu penghubung geser sehingga yang dihasilkan adalah suatu penampang non komposit. Pada penampang non komposit, pelat beton akan mengalami lendutan yang cukup besar yang disebabkan oleh besarnya beban yang harus dipikul oleh pelat beton tersebut. Seiring berkembangnya metode pengelasan yang baik serta ditemukannya alat-alat penghubung geser yang menahan gaya geser horizontal, maka lekatan antara pelat beton dan balok baja dapat ditingkatkan. Pada akhirnya kedua material ini (baja dan beton) akan menjadi satu kesatuan komponen struktur yang disebut dengan komponen struktur komposit. Komponen struktur komposit ini dapat menahan beban sekitar 33% hingga 50% lebih besar daripada beban yang dapat dipikul oleh balok baja saja tanpa adanya perilaku komposit.

Pada awal tahun 1930 konstruksi jembatan juga sudah mulai menggunakan penampang komposit, namun baru pada tahun 1944

dikeluarkan peraturan oleh AASHTO (*American Association of State Highway and Transportation Officials*) tentang spesifikasi jembatan jalan raya dengan struktur komposit. Pada sekitar tahun 1950 penggunaan lantai jembatan komposit mulai berkembang dengan pesat (terutama di Amerika). Pada jembatan ini gaya geser longitudinal ditransfer dari balok baja kepada pelat beton bertulang dengan menggunakan penghubung geser. Hal ini mengakibatkan pelat beton tersebut akan turut serta membantu memikul momen lentur yang timbul. Penampang komposit ini dapat di lihat dalam gambar 2.1.a



Gambar 2.1. (a) Lantai jembatan komposit dengan penghubung geser, (b) Balok baja yang diselubungi beton, (c) Lantai komposit gedung dengan penghubung geser.

Pada awal tahun 1960 mulai dikembangkan penggunaan komponen struktur komposit untuk bangunan gedung yang menganut pada spesifikasi yang dikeluarkan oleh AISC (*American Institute of Steel Construction*) pada tahun 1952. Komponen struktur komposit yang digunakan dapat berupa balok baja yang diselubungi dengan beton (Gambar 2.1.b) atau berupa balok baja yang menopang pelat beton dengan penghubung geser (Gambar 2.1.c). Namun sekarang struktur balok baja yang diselubungi

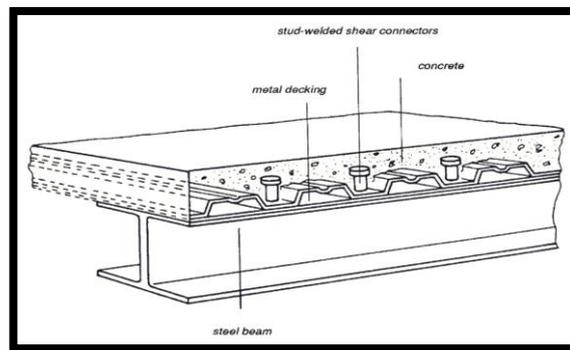
dengan beton sudah jarang digunakan, dan hampir seluruh struktur komposit untuk bangunan gedung mempunyai penampang seperti pada Gambar 2.1.c.

Perilaku komposit hanya akan terjadi jika potensi terjadinya slip antara kedua material ini dapat dicegah. Telah dijelaskan sebelumnya bahwa hal ini dapat teratasi jika gaya geser horizontal pada kedua permukaan baja dan beton dapat ditahan dengan menggunakan penghubung geser. Tipe-tipe penghubung geser yang sering digunakan dapat berupa stud, baja tulangan spiral, atau profil kanal kecil yang pendek. Penghubung geser ini selanjutnya dihubungkan pada bagian flens atas balok dengan jarak tertentu dan akan memberikan sambungan secara mekanik melalui mekanisme pengankuran dalam beton yang telah mengeras. Penghubung geser tipe stud paling banyak digunakan, dan lebih dari satu buah stud dapat dipasangkan pada tiap lokasi, jika lebar flens memungkinkan. Di samping itu pemasangan stud juga relative lebih mudah dan hanya membutuhkan tenaga kerja dalam jumlah yang sedikit.

Sejumlah penghubung geser diperlukan untuk membuat sebuah balok dapat berfungsi komposit secara penuh. Namun terkadang jumlah penghubung geser dapat dipasang lebih sedikit daripada yang diperlukan untuk menimbulkan perilaku komposit penuh, hal ini akan mengakibatkan terjadinya slip antara baja dan beton. Balok seperti ini dikatakan mengalami aksi komposit parsial.

Seiring dengan perkembangan teknologi, mulai ditemukan pula pelat baja gelombang yang digunakan dalam pembuatan struktur pelat komposit dan

terbuat dari bahan yang mempunyai tegangan tarik tinggi serta dilapisi bahan anti karat. Pelat baja gelombang ini mempunyai dua macam fungsi yaitu sebagai bekisting tetap dan sebagai penulangan positif satu arah pada lantai beton bangunan gedung bertingkat. Arah gelombang (*rib*) dari plat baja ini dapat diletakkan dalam arah tegak lurus atau sejajar terhadap balok. Namun pada sistem pelat lantai komposit, umumnya arah rib diletakkan tegak lurus terhadap balok lantai dan sejajar dengan arah balok induk. Gambar 2.2. memperlihatkan system plat lantai yang menggunakan pelat baja gelombang.



Gambar 2.2. Pelat lantai komposit dengan pelat baja gelombang

Pembahasan awal dari bab ini akan difokuskan pada komponen struktur komposit biasa dengan penghubung geser (Gambar 2.1.c) dan akan dilanjutkan dengan pembahasan mengenai pelat lantai komposit dengan menggunakan pelat baja gelombang.

Dengan menggunakan konstruksi komposit dalam desain suatu komponen ternyata dapat diperoleh beberapa keuntungan sebagai berikut:

- a. Dapat mereduksi berat profil baja yang dipakai
- b. Tinggi profil baja yang dipakai dapat dikurangi
- c. Meningkatkan kekakuan lantai

d. Dapat menambah panjang bentang layan

Reduksi berat sekitar 20-30% dapat diperoleh dengan memanfaatkan perilaku sistem komposit penuh. Dengan adanya reduksi berat ini maka secara langsung juga dapat mengurangi tinggi profil baja yang dipakai. Berkurang tinggi profil baja yang dipakai akan mengakibatkan berkurangnya tinggi bangunan secara keseluruhan, dan membawa dampak pula berupa penghematan material bangunan, terutama untuk dinding luar dan tangga.

Kekakuan dari pelat lantai komposit pada dasarnya lebih besar daripada kekakuan pelat beton dan balok baja yang beraksi non komposit. Secara normal pelat beton berperilaku sebagai pelat satu arah yang membentang di antara balok-balok penopang. Dalam desain komposit, momen inersia balok akan bertambah sehingga kekakuan pelat lantai akan meningkat. Meningkatnya kekakuan ini akan memberikan beberapa keuntungan dalam pelaksanaan konstruksi, antara lain bahwa lendutan akibat beban hidup akan berkurang dan penggunaan perancah selama proses konstruksi struktur komposit akan mampu mengurangi lendutan akibat beban mati. Disamping itu dengan menggunakan asumsi desain komposit, maka kapasitas penampang dalam menahan beban akan jauh lebih besar daripada kapasitas pelat beton atau profil baja yang berkerja sendiri-sendiri. Namun dalam daerah momen negatif, kekakuan dari sistem komposit harus dihitung kembali karena dalam daerah ini beton (yang mengalami tarik) harus diabaikan. Dalam daerah momen negatif biasanya harus disediakan tulangan tekan pada pelat beton.

## B. Kuat Lentur Nominal

Kuat lentur nominal dari suatu komponen struktur komposit (untuk momen positif), menurut SNI 03-1729-2002 pasal 12.4.2.1 ditentukan sebagai berikut:

a. Untuk  $\frac{h}{t_w} \leq \frac{1680}{\sqrt{f_{yf}}}$

$M_n$  kuat momen nominal yang dihitung berdasarkan distribusi tegangan plastis pada penampang komposit.  $\phi_b = 0,85$

b. Untuk  $\frac{h}{t_w} > \frac{1680}{\sqrt{f_{yf}}}$

$M_n$  kuat momen nominal yang dihitung dengan menggunakan distribusi tegangan plastis (memperhitungkan pengaruh tumpuan sementara). Pada kondisi ini, kekuatan lentur batas penampang ditentukan oleh terjadinya leleh pertama.  $\phi_b = 0,9$

Kuat lentur nominal yang dihitung berdasarkan distribusi tegangan plastis, dikategorikan menjadi dua kasus sebagai berikut:

- a. Sumbu netral plastis jatuh pada pelat beton

Dengan mengacu pada Gambar 2.3, maka besar gaya tekan C adalah:

$$C = 0,85 \times f'_c \times a \times b_e \quad (2.1)$$

Gaya tarik T pada profil baja adalah sebesar:

$$T = A_s \times f_y \quad (2.2)$$

Dari keseimbangan gaya  $C = T$ , maka diperoleh

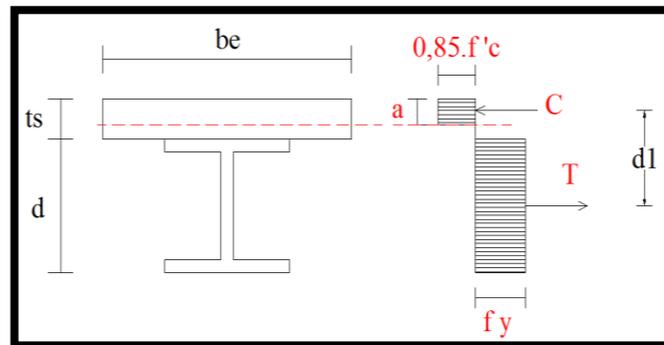
$$a = \frac{A_s \times f_y}{0,85 \times f'_c \times b_e} \quad (2.3)$$

Kuat lentur nominal dapat dihitung dari Gambar 2.3:

$$M_n = C \times d_1 \quad (2.4)$$

$$M_n = T \times d_1 \quad (2.5)$$

$$M_n = A_s \times f_y \times \left( \frac{d}{2} + t_s - \frac{a}{2} \right) \quad (2.6)$$



Gambar 2.3. Kuat lentur nominal berdasarkan distribusi tegangan plastis (sumbu netral plastis jatuh pada pelat beton)

Jika dari hasil perhitungan ternyata nilai  $a > t_s$ , maka asumsi harus diubah. Hasil ini menyatakan bahwa pelat beton tidak cukup kuat untuk mengimbangi gaya tarik yang timbul pada profil baja.

b. Sumbu netral plastis jatuh pada profil baja

Apabila ke dalam blok tegangan beton, nilai  $a$  ternyata melebihi plat beton, maka distribusi tegangan dapat ditunjukkan seperti pada Gambar 2.4. dan Gambar 2.5. Gaya tekan  $C_c$  yang berkerja pada beton adalah sebesar:

$$C_c = 0,85 \cdot f'_c \cdot b_e \cdot t_s \quad (2.7)$$

Dari keseimbangan gaya, diperoleh hubungan:

$$T' = C_c + C_s \quad (2.8)$$

Besarnya  $T'$  sekarang lebih kecil daripada  $A_s \cdot f_y$ , yaitu:

$$T' = A_s \cdot f_y - C_s \quad (2.9)$$

Dengan menyamakan persamaan (2.8) dan (2.9) diperoleh:

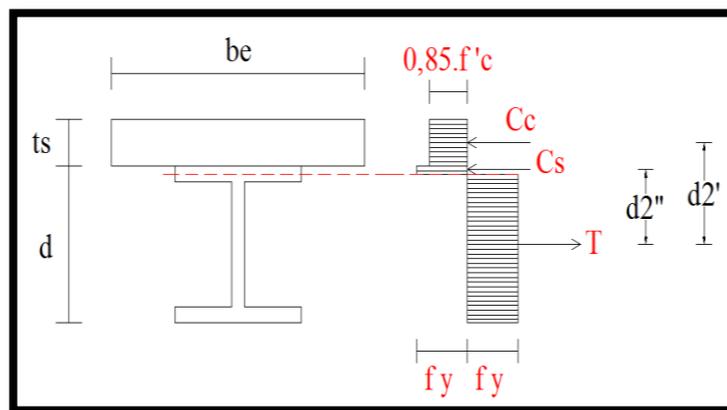
$$C_s = \frac{A_s \cdot f_y - C_c}{2} \quad (2.10)$$

Atau dengan mensubstitusikan persamaan (2.7), diperoleh bentuk:

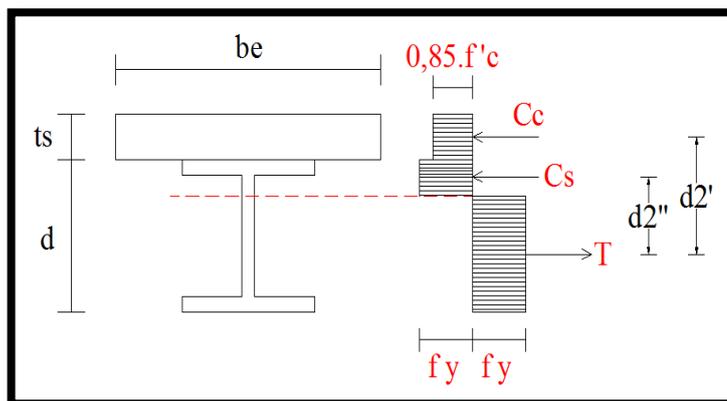
$$C_s = \frac{A_s \cdot f_y - 0,85 \cdot f'_c \cdot b_e \cdot t_s}{2} \quad (2.11)$$

Kuat lentur nominal diperoleh dengan memperhatikan Gambar 2.3:

$$M_n = C_c \cdot d'_2 + C_s \cdot d''_2 \quad (2.12)$$



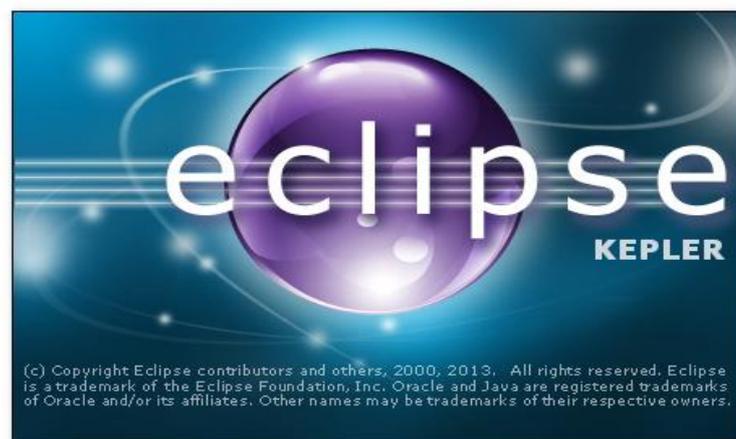
Gambar 2.4. Kuat lentur nominal berdasarkan distribusi tegangan plastis  
(sumbu netral plastis jatuh pada pelat baja di flens)



Gambar 2.5. Kuat lentur nominal berdasarkan distribusi tegangan plastis  
(sumbu netral plastis jatuh pada pelat baja di web)

### C. Pengenalan *Eclipse IDE*

Sejak berkurangnya standarisasi pengujian aplikasi pada *android*, membangun aplikasi pada *operating system* tersebut menjadi lebih mudah dan *fleksibel* di bandingkan dengan sebelumnya. Sehingga semakin banyak *programmer-programmer* baru yang lahir dengan beragam kreatifitas dalam mengembangkan aplikasi.



Gambar 2.6. Logo Eclipse Kepler

Dalam mengembangkan sebuah aplikasi, dibutuhkan *software* pendukung. Begitu pula dalam mengembangkan aplikasi pada *android*, memerlukan *software* seperti Eclipse sebagai IDE (*Integrated Development Environment*) atau program komputer dengan beberapa fasilitas yang diperlukan dalam mengembangkan perangkat lunak. Beberapa keunggulan Eclipse sebagai pengembangan perangkat lunak sebagai berikut:

- a. Eclipse tersedia secara bebas untuk merancang dan mengembangkan *android*.
- b. Eclipse merupakan IDE terpopuler, ini dapat terlihat pada banyaknya *developers* yang menggunakan Eclipse sebagai IDE dalam pengembangan aplikasinya.

- c. Eclipse memiliki *plugin android*.
- d. Eclipse mendapatkan dukungan sebagai IDE pengembang *android* dari google.

Eclipse pertama kali diluncurkan tanggal 21 juni 2004. Sejak itu, setiap tahun pada bulan juni diluncurkan versi baru dari eclipse. Umumnya, versi eclipse menggunakan nama yang berhubungan dengan astronomi.

Version Name	Date	Platform Version	Projects	Main Changes
Austin	21 June 2004	3.0 <sup>[14]</sup>		
N/A	28 June 2005	3.1		
Callisto	30 June 2006	3.2	Callisto projects <sup>[15]</sup>	
Europa	29 June 2007	3.3	Europa projects <sup>[16]</sup>	
Ganymede	25 June 2008	3.4	Ganymede projects <sup>[17]</sup>	
Galileo	24 June 2009	3.5	Galileo projects <sup>[18]</sup>	
Helios	23 June 2010	3.6	Helios projects <sup>[19]</sup>	
Indigo	22 June 2011	3.7	Indigo projects <sup>[20]</sup>	
Juno	27 June 2012	3.8 and 4.2 <sup>[21]</sup> [Notes 1]	Juno projects <sup>[24]</sup>	
Kepler	26 June 2013	4.3	Kepler projects <sup>[25]</sup>	
Luna	25 June 2014	4.4	Luna projects <sup>[26]</sup>	Integrated Java 8 support (in the previous version this was possible via a "Java 8 patch" plugin)
Mars	24 June 2015 (planned)	4.5	Mars projects <sup>[27]</sup>	

■ Old version   
 ■ Older version, still supported   
 ■ Latest version   
 ■ Future release

Gambar 2.7. Tabel nama dan tanggal rilis setiap versi Eclipse

Dalam mengembangkan aplikasi berbasis *android* menggunakan eclipse ada beberapa hal yang harus dipersiapkan antara lain spesifikasi *hardware* dan instalansi *software*. Spesifikasi minimum *hardware* yang diperlukan sebagai berikut:

- a. Sistem operasi yang dapat digunakan untuk menjalankan program *android*, antara lain:
  1. Windows XP (32 bit), Windows Vista (32 atau 64 bit), Windows 7 (32 atau 64 bit).
  2. Mac OS X 10.5.8 atau yang lebih tinggi.
  3. Linux.

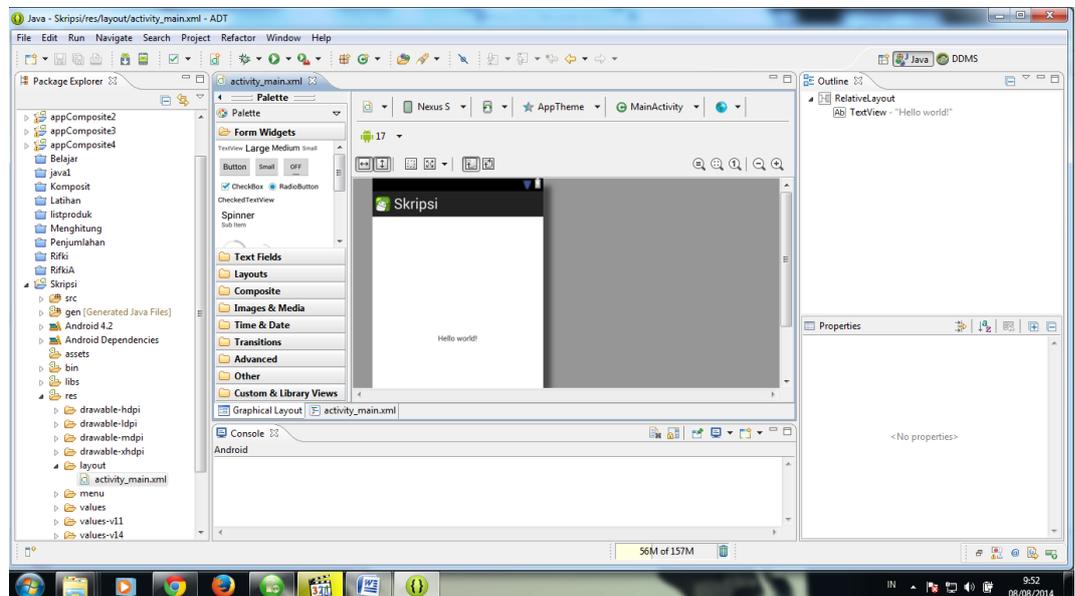
- b. Space *hard disk* yang diperlukan minimal adalah 500MB (selain JDK dan Eclipse).

Ada 2 instalansi *software* yang perlu disiapkan, antara lain:

- a. JDK (*Java Development Kit*)
- b. Paket ADT 21.X.X (*Android Development Tools*) yang berisi SDK (*Software Development Kit*) android dan eclipse.

Berikut adalah bagian-bagian yang terdapat dalam IDE Eclipse yang akan digunakan dalam pemograman *android*.

- a. *Interface Eclipse* merupakan tampilan dari *Eclipse* (Gambar 2.8.).



Gambar 2.8. *Interface Eclipse*

- b. *Menubar dan Toolbar*



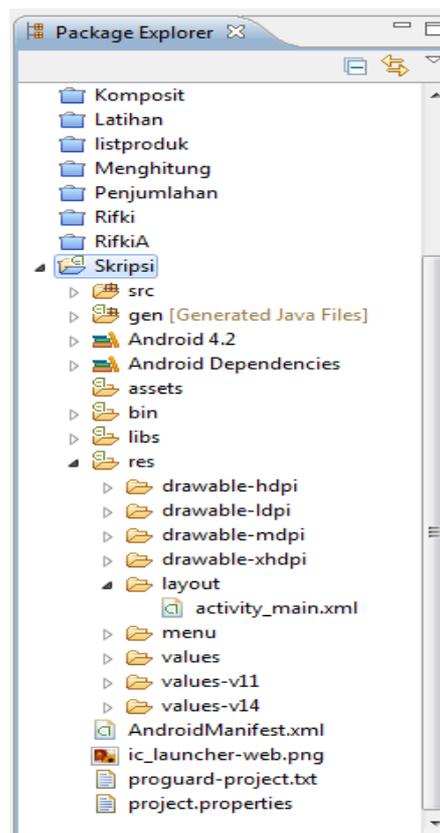
Gambar 2.9. *Jendela Menubar dan Toolbar*

*Menubar* merupakan bagian yang berisi menu serta submenu yang ada pad IDE Eclipse, digunakan untuk menjalankan eksekusi perintah

tertentu (Gambar 2.9.). *Toolbar* merupakan bagian yang berisi ikon-ikon perintah yang akan mempercepat melakukan eksekusi perintah pada eclipse (Gambar 2.9.).

c. *Package Explorer*

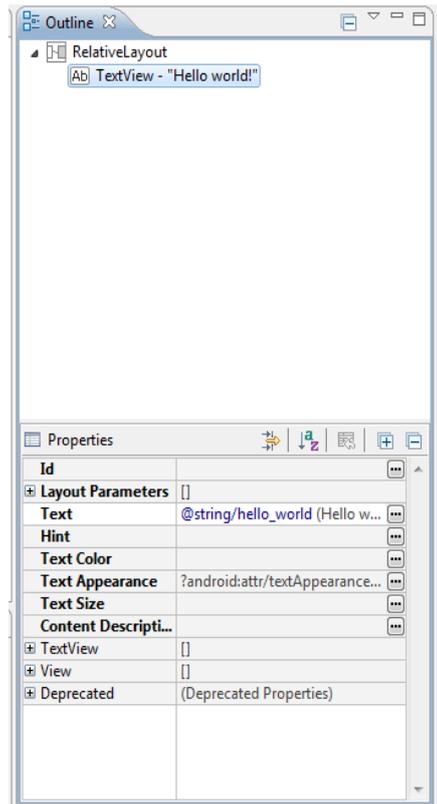
*Package Explorer* merupakan bagian yang berisi paket-paket aplikasi yang dibuat (Gambar 2.10.).



Gambar 2.10. Jendela *Package Explorer*

d. *Outline*

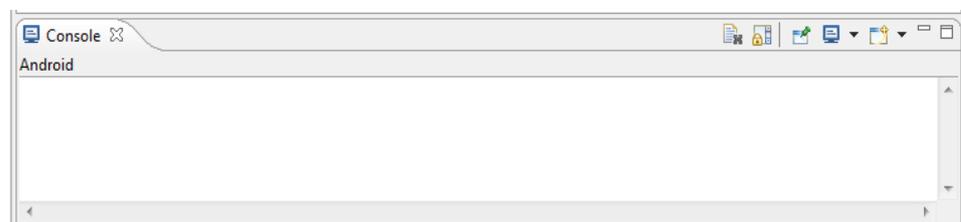
*Outline* merupakan bagian yang berisi *widget* yang digunakan didalam aplikasi (Gambar 2.11.).



Gambar 2.11. Jendela *Outline*

e. *Console*

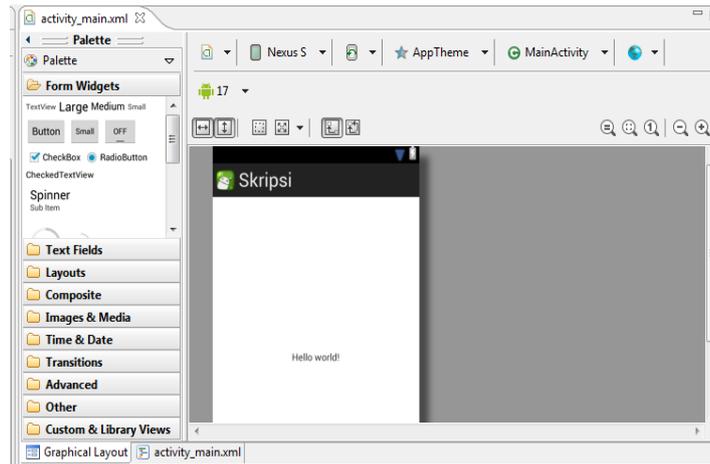
*Console* merupakan bagian yang berisi informasi status program yang dijalankan. Jika terdapat eror maka pesan akan ditampilkan pada kotak ini (Gambar 2.12.).



Gambar 2.12. Jendela *Console*

f. *Graphical Layout*

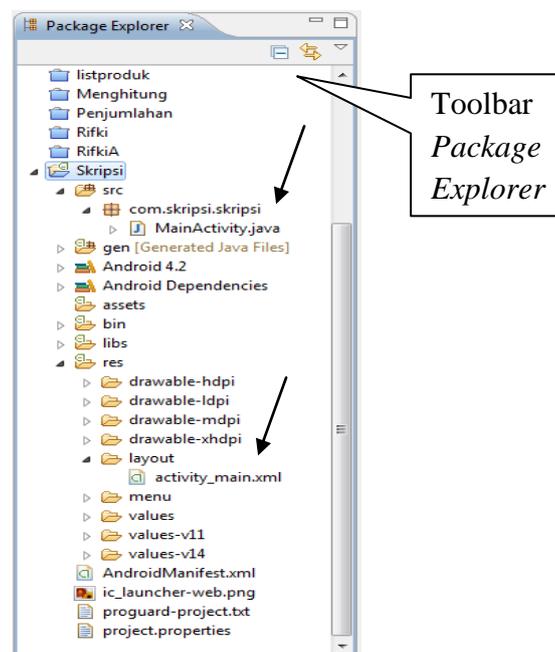
*Graphical Layout* merupakan bagian editor grafik dimana dapat memasukkan *widget* serta mengaturnya secara visual (Gambar 2.13.).



Gambar 2.13. Jendela *Graphical Layout*

g. Pengenalan *Package Explorer*

*Package Explorer* yang berada pada panel di sebelah kiri *workbench* merupakan panel hierarki element-element project java yang telah di buat. *Package Explorer* menampilkan semua project java yang pernah disimpan pada *workspace* pada folder-folder nama project. Di dalam folder tersebut, terdapat file *jar*, *library*, dan *class* dari project java yang diurutkan (Gambar 2.14.).



Gambar 2.14. *Package Explorer* Skripsi

Pada gambar tersebut (Gambar 2.8.) terdapat *toolbar button* yang dapat digunakan ketika mengakses *Package Explorer*.

Tabel 2.1. Deskripsi ikon milik *Package Explorer*

Ikon	Deskripsi
	<i>Collaps all</i> . Menampilkan semua node yang terdapat pada tree.
	<i>Link with editor</i> . Link pilihan <i>package explorer</i> untuk editor aktif
	<i>View Menu</i> . Menu-menu yang dapat digunakan dalam mengatur panel <i>package explorer</i> .
	<i>Minimize</i> . Memperkecil tampilan panel <i>package explorer</i> .
	<i>Maximize</i> . Memperbesar tampilan panel <i>package explorer</i>

Penjelasan beberapa folder *package explorer*, antara lain:

a. *Android Manifest.xml*

File konfigurasi pusat dari aplikasi.

b. *defalut.properties*

Sebuah file hasil dari pengembangan aplikasi yang digunakan oleh eclipse dan plugin ADT. Jangan pernah mengubah/edit file ini.

c. *Proguard.cfg*

Sebuah file hasil dari pengembangan aplikasi yang digunakan oleh Eclipse, ProGuard, dan plugin ADT.

d. *src/*

merupakan direktori untuk file kode aplikasi utama yang didalamnya terdapat *activity class* yang berjalan ketika aplikasi diluncurkan menggunakan ikon aplikasi.

e. *res/*

Berisikan beberapa sub-directori untuk sumber daya aplikasi berupa pewarnaan, dimensi, string, dan lainnya. Berikut ini adalah isi dari *res/*:

1. */res/drawable-hdpi/*

Merupakan direktori untuk drawable-object (seperti bitmap) yang dirancang untuk layar *high-density* (hdpi). Sedangkan direktori *drawable* lainnya mengandung aset yang dirancang untuk kepadatan layar lainnya.

2. *layout/*

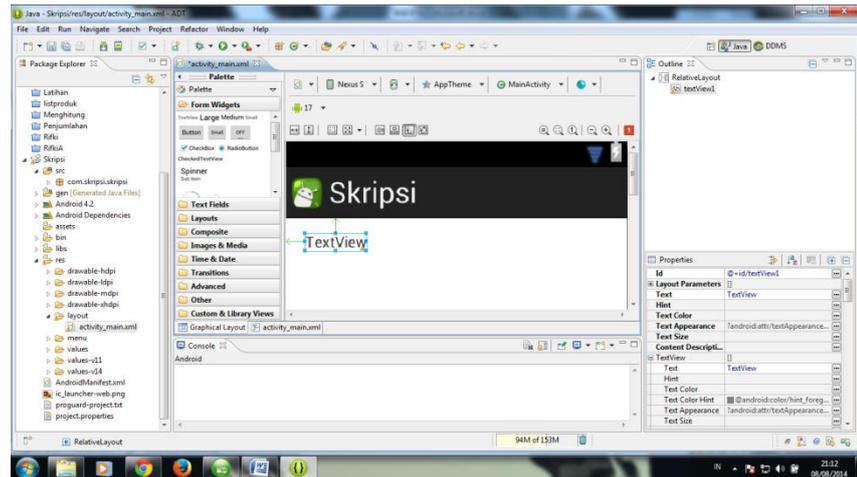
merupakan direktori file yang mendefinisikan antarmuka pengguna aplikasi.

3. *values/*

merupakan direktori untuk berbagai file XML lain yang berisi sumber daya seperti string dan definisi warna.

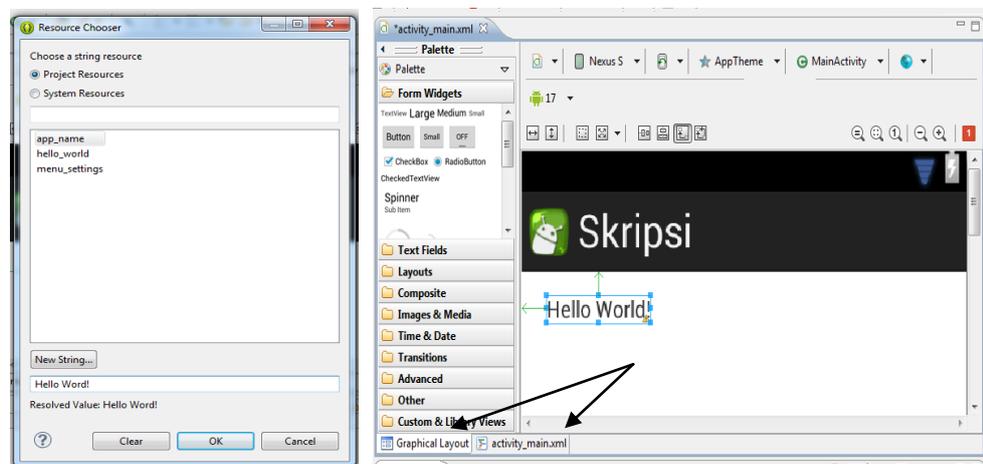
Cara melakukan setting layout pada *package explorer*. Adapun langkah-langkahnya adalah sebagai berikut:

- a. Pada *package explorer* di direktori *res/* → *layout* → *activity\_main.xml* hingga muncul tampilan seperti pada gambar 2.15.
- b. Pada tampilan tersebut, membuat kalimat “ Hello Word! ” dengan menggunakan *textview* yang berada pada form *widgets* (disebelah kiri *workbench*). Caranya dengan menarik *textview* dari form *widgets*, dan menaruh *textview* tersebut di *layout*.



Gambar 2.15. Cara membuat textview “ Hello World! “

- c. Merubah textview menjadi “ Hello Word ! “ dengan cara klik kanan pada textview → edit text. Menganti tulisan textview menjadi “ Hello Word ! “ → ok (Gambar 2.16).



a. Tampilan edit text

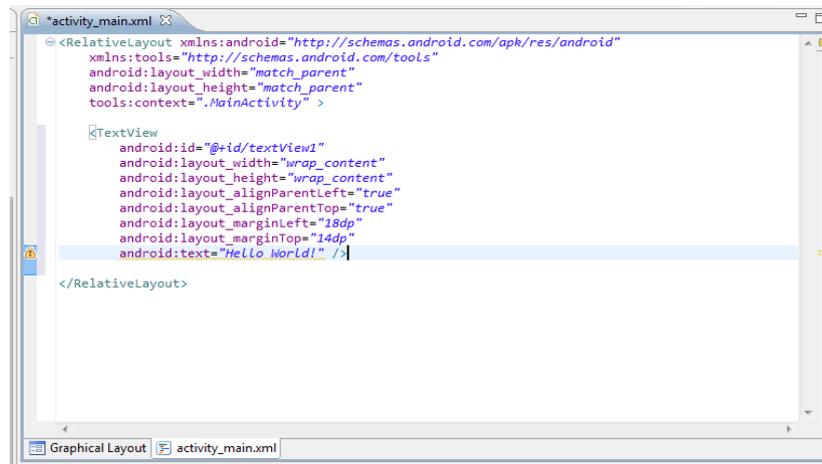
b. Hasil dari edit text

Gambar 2.16. tampilan edit text dan hasilnya edit text

- d. Ada satu lagi cara mengganti tampilan textview tersebut yaitu dari *graphical layout* ke *activity\_main.xml* yang berada di bawah *workbench* (gambar 2.16.b).

- e. Pada gambar 2.17 saat tulisan didalam perintah `android:text="Hello World!"` diganti, maka hasil di layout

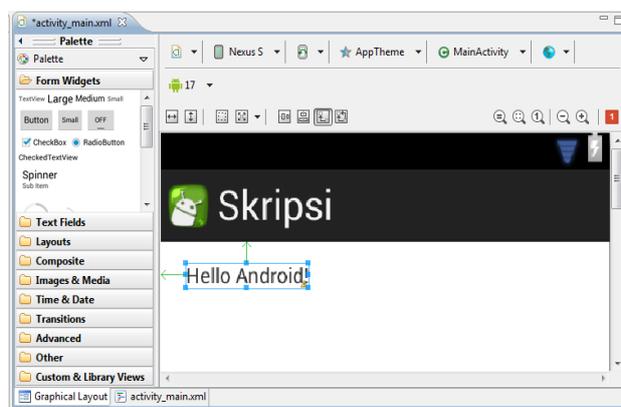
berubah. Jika perintah `android:text="Hello World!"` menjadi `android:text="Hello Android!"`



Gambar 2.17. `activity_main.xml` dengan `android:text="Hello World!"`



Gambar 2.18. `activity_main.xml` dengan `android:text="Hello Android!"`



Gambar 2.19. hasil perubahan melalui `activity_main.xml`