

IMPLEMENTASI LITERASI TERHADAP APLIKASI *EASILY APPLICABLE GRAPHICAL LAYOUT EDITOR* BERBASIS PENDIDIKAN ISLAM

Arief Goeritno^{1)*}, Dian Anwar²⁾, Syaiful³⁾, Ritzkal⁴⁾, Ahya Ibadina Syahida⁵⁾, Rezi Setria⁶⁾

¹ Program Studi Teknik Elektro, Universitas Ibn Khaldun (UIKA) Bogor

² Politeknik Logistik Elektronika APP Depok

³ Program Studi Teknik Sipil, UIKA Bogor

⁴ Program Studi Teknik Informatika, UIKA Bogor

⁵ Program Studi Agribisnis, Fakultas Pertanian, Universitas Jenderal Soedirman (Unsoed)

⁶ Program Studi Manajemen Pendidikan Islam, Sekolah Pascasarjana (SPs), UIKA Bogor

* E-mail: arief.goeritno@uika-bogor.ac.id

Abstract

On Islamic education for students of SMK Elektronika, it has been done summary material about application with image brands (brand image) Easily Applicable Graphical Layout Editor (EAGLE) by literacy approach. EAGLE application is one of the principals of fundamental Islamic education in the material form. The literacy of EAGLE application material, including general explanation and important notes to the EAGLE application. A common explanation of the EAGLE application which is one of the many applications for the draft in the manufacture of pieces of printed boards (printed circuit board, PCB) computer-assisted (computer aided design, CAD), so as layout editor is easy to use to help draft board printed circuit (PCB). EAGLE application consists of three main modules, namely the Layout Editor, Schematic Editor, and Autoroutes embedded in a single user interface. All three can be accessed from EAGLE's User Interface (UI) or user interface on EAGLE application, so there is no need to change netlists between the scheme and layout. Some advantages of the EAGLE application, namely (i) cross-program (cross platform), where EAGLE can operate on the operating system (Operating System), either Windows, Mac, or Linux, (ii) the need for storage is small, there are editions free use (free of charge) or can also have the low cost, and (iii) support of the community. Important notes for implementation literacy to EAGLE application, grouped in five vital records, namely (a) view basis, (b) creation with project new and schematic, (c) reinforcement board layout, (d) loading pathways among parts (routes of parts), and (e) checking rules for design (design rule check, drc).

Keywords: *Computer Application Literacy, Easily Applicable Graphical Layout Editor (EAGLE), Islamic Education*

Abstrak

Dalam pendidikan Islam untuk siswa SMK Teknik Elektronika, telah dilakukan ringkasan materi tentang aplikasi dengan citra merk (*brand image*) *Easily Applicable Graphical Layout Editor (EAGLE)* dengan pendekatan secara literasi. Aplikasi *EAGLE* merupakan salah satu pokok dari sejumlah pokok pendidikan islami berupa materi. Literasi terhadap materi aplikasi *EAGLE*, meliputi penjelasan umum dan catatan penting pada aplikasi *EAGLE*. Penjelasan umum terhadap aplikasi *EAGLE* yang merupakan salah satu dari sekian banyak aplikasi untuk rancangan dalam pembuatan keping papan tercetak (*printed circuit board, PCB*) berbantuan komputer (*computer aided design, CAD*), sehingga dikatakan sebagai *editor* tata letak yang mudah digunakan untuk bantuan rancangan papan rangkaian tercetak (*PCB*). Aplikasi *EAGLE* terdiri atas tiga modul utama, yaitu *Editor* Tata Letak, *Editor Schematic*, dan *Autorouter* yang disematkan dalam satu antarmuka pengguna. Ketiganya dapat diakses dari *EAGLE's User Interface (UI)* atau antarmuka pengguna pada *EAGLE*, sehingga tidak perlu pengubahan *netlists* antara skema dan tata letak. Sejumlah keunggulan aplikasi *EAGLE*, yaitu (i) lintas program (*cross platform*), dimana aplikasi *EAGLE* dapat beroperasi pada sistem

pengoperasian (*Operating System, OS*), baik Windows, Mac, atau Linux, (ii) kebutuhan untuk penyimpanan yang kecil, terdapat edisi bebas pakai (*free of charge*) atau dapat juga dipilih yang berbiaya murah (*low cost*), dan (iii) dukungan masyarakat (*community support*). Catatan penting untuk implementasi literasi terhadap aplikasi *EAGLE*, dikelompokkan pada lima catatan penting, yaitu (a) pandangan (*view*) dasar, (b) berkreasi dengan proyek baru dan skematik, (c) pembuatan *board layout*, (d) pembuatan jalur-jalur penghubung antar bagian (*routes of parts*), dan (e) pengecekan aturan untuk rancangan (*design rule check, drc*).

Kata kunci: Literasi Aplikasi Komputer, *Easily Applicable Graphical Layout Editor (EAGLE)*, Pendidikan Islam

PENDAHULUAN

Penciptaan aplikasi dengan citra merk (*brand image*) *Easily Applicable Graphical Layout Editor* atau *EAGLE* (CadSoft, 2010) merupakan salah satu karya manusia. Dalam perspektif keislaman, karya manusia merupakan nikmat sekaligus karunia dari Allah SWT (*God's Grace*), sebagaimana ungkapan *إِنَّ كُلَّ عَمَلِ الْإِنْسَانِ فَهُوَ نِعْمَةٌ مِنَ اللَّهِ تَعَالَى*. Lebih lanjut dalam al-Qur'an pada Surat An-Nahl (surat ke-16, ayat 53) Allah SWT berfirman:

وَمَا بِكُمْ مِنْ نِعْمَةٍ فَمِنَ اللَّهِ ثُمَّ إِذَا مَسَّكُمُ الضُّرُّ فَإِلَيْهِ تَجْأَرُونَ

“Dan apa saja nikmat yang ada pada kamu, maka dari Alloh-lah (datangnya), dan apabila kamu ditimpa oleh kemudharatan, maka hanya kepada-Nya-lah kamu meminta pertolongan”.

Berdasarkan keberadaan karya manusia yang berkaitan erat dengan sejumlah pokok pada pendidikan islami, maka aplikasi *EAGLE* merupakan unsur materi yang diperlukan untuk siswa pada Program Studi Teknik Elektronika di Sekolah Menengah Kejurusan (SMK). Pokok-pokok pada pendidikan islami (Thalib, 2001), meliputi (i) dasar, (ii) tujuan, (iii) pendidik, (iv) objek didik, (v) materi, (vi) pengetahuan dasar, (vii) sistem, (viii) azas, (ix) metode, (x) sifat, (xi) seni, (xii) bentuk, (xiii) masa, (xiv) jenjang, (xv) penanggung jawab, (xvi) dana, (xvii) sarana, (xviii) tempat, (xix) sikap pendidik, dan (xx) sanksi.

Sejumlah aplikasi dengan peruntukan simulasi kebijakan (Syahida, 2017), perkiraan kondisi, atau keperluan khusus, seperti aplikasi *EAGLE* (Clarke, 2008; CadSoft, 2010; Aono, 2011) telah banyak beredar di pasaran. Aplikasi *EAGLE* merupakan salah satu dari sembilan aplikasi untuk rancangan *printed circuit board (PCB)* paling top (Gong, 2017) yang berkaitan dengan pembuatan diagram skematis rangkaian elektronika dengan sasaran akhir berupa tata letak (*layout*) komponen-komponen elektronika pada sebuah *board*.

Integrasi dalam perolehan diagram skematis rangkaian elektronika berbasis sistem mikrokontroler merupakan sejumlah tahapan penting dan krusial dalam pembuatan *board* untuk tata letak komponen sistem, penempatan dan pengawatan antara sensor dan sistem mikrokontroler, dan penyambungan pada sistem mikrokontroler untuk sistem keluaran (Goeritno, 2014). *Board* berfungsi sebagai penghubung keberadaan seluruh komponen penyusun sebuah sistem mikrokontroler (Effendi, 2015; Goeritno, 2016) atau mikrokomputer.

Rangkaian elektronika pada sistem mikrokontroler dibuat untuk pembentukan jalur antar komponen dengan bantuan aplikasi *EAGLE* (Clarke, 2008; CadSoft, 2010; Aono, 2011), agar *pin* pada mikrokontroler diketahui penggunaannya (Goeritno, 2016b; Goeritno, 2017). Tata letak komponen merupakan langkah untuk pengubahan diagram skematis menjadi bentuk rangkaian pada *board* dengan perintah *switch to board* pada program aplikasi *EAGLE* (Clarke, 2008). Penggunaan komponen-komponen dipilih sesuai dengan kebutuhan dalam pembuatan *board* mikrokontroler (Goeritno, 2016).

Keberadaan *board* (Mathivanan, 2006; Bates, 2011) pada sistem utama untuk mikrokontroler didasarkan kepada capaian berupa bentuk fisis sistem minimal (*minimum system*) terhadap suatu modul peranti elektronika untuk sistem pengontrolan suhu (Saefurrochman, 2015) pada Program Studi Teknik Elektronika di Sekolah Menengah Kejuruan (Goeritno, 2016). Perolehan bentuk *minimum system* berbasis mikrokontroler untuk berbagai keperluan, juga dilalui dengan pemanfaatan aplikasi *EAGLE* (Clarke, 2008; CadSoft, 2010; Aono, 2011), seperti i) pengontrolan suhu dan humiditas (Goeritno, 2014), ii) konsepsi perancangan sistem pengontrolan untuk analogi *smart greenhouse* (Prakoso, 2016; Goeritno, 2016, Goeritno, 2017), iii) pemantauan awal konduktivitas dan kualitas air (Effendi, 2015; Goeritno, 2016), iv) pengendalian *actuator* dengan sensor *voice recognition* (Ginting, 2016; Goeritno, 2016c; Goeritno, 2017), v) pemantauan kondisi instalasi kelistrikan di otobis (Mustofa, 2016a; Mustofa, 2016b; Goeritno, 2017), vi) sistem pengontrolan beban-beban listrik dengan sistem *payload data handling* (Sopyandi, 2016; Goeritno, 2017), dan vii) prototipe sistem pengaman tambahan pada kendaraan roda empat atau lebih (Afandi, 2017).

Berdasarkan uraian-uraian tersebut yang berkaitan erat dengan karya manusia dan pokok-pokok pada pendidikan islami, maka tujuan implementasi literasi terhadap

aplikasi *EAGLE* untuk siswa SMK di Program Studi Teknik Elektronika berbasis pendidikan Islam ini, meliputi literasi penjelasan umum dan catatan penting pada aplikasi *EAGLE*. Keterkaitan penjelasan terhadap implementasi literasi aplikasi *EAGLE* di SMK Teknik Elektronika berbasis pendidikan Islam lebih ditekankan kepada objek didik, materi, dan jenjang. Objek didik berupa siswa pendidikan formal, materi berupa aplikasi komputer, dan jenjang pada pendidikan di sekolah menengah kejuruan.

PEMBAHASAN

Penjelasan Umum pada Aplikasi *EAGLE*

Aplikasi *EAGLE* merupakan salah satu dari sekian banyak aplikasi untuk rancangan dalam pembuatan keping papan tercetak (*printed circuit board, PCB*) berbantuan komputer (*computer aided design, CAD*), sehingga dikatakan sebagai *editor* tata letak yang mudah digunakan bantuan rancangan papan rangkaian tercetak (*PCB*). Aplikasi *EAGLE* terdiri atas tiga modul utama, yaitu *Editor Tata Letak, Editor Schematic*, dan *Autorouter* yang disematkan dalam satu antarmuka pengguna.

Ketiganya dapat diakses dari *EAGLE's User Interface (UI)* atau antarmuka pengguna pada *EAGLE*, sehingga tidak perlu perubahan *netlists* antara skema dan tata letak. Fitur utama aplikasi dicakup juga anotasi maju dan mundur *online*, fungsi bantuan sensitif untuk konteks, beberapa jendela untuk papan, skematik dan kepustakaan, bahasa pengguna yang *powerful*, editor teks terpadu, dan tidak terdapat perlindungan salinan perangkat keras (Clarke, 2008; CadSoft, 2010; Aono, 2011).

Sejumlah keunggulan aplikasi *EAGLE*, karena beberapa alasan, yaitu: lintas program (*cross platform*), kebutuhan untuk penyimpanan yang kecil, terdapat edisi bebas pakai (*free of charge*) atau dapat juga dipilih yang berbiaya murah (*low cost*), dan dukungan masyarakat (*community support*). Alasan *cross platform*, yaitu aplikasi *EAGLE* dapat beroperasi pada sistem pengoperasian (*Operating System, OS*), baik Windows, Mac, atau Linux. Kebutuhan untuk penyimpanan yang kecil (50-200MB ruang *disk*), merupakan alasan dalam pengasangan (*peng-install-an*), sehingga hanya perlu beberapa menit. Disediakan dengan edisi *free of charge* dengan penyediaan utilitas yang memadai. Dukungan komunitas pengguna dan penggemar, telah menjadikan aplikasi *EAGLE* termanfaatkan dalam perancangan *PCB*.

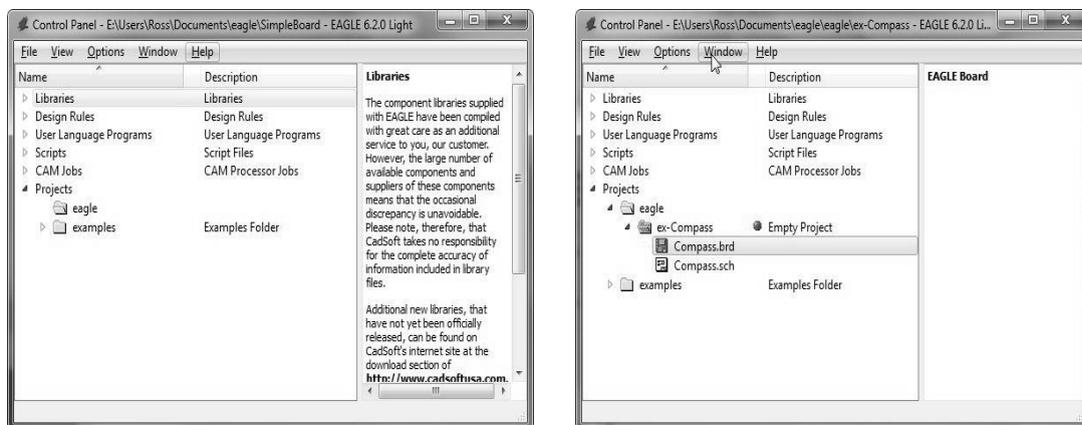
Catatan Penting pada Aplikasi *EAGLE*

Antarmuka untuk pengguna (*user interface, UI*) pada *EAGLE* dirancang dengan apa yang disebut antarmuka modal, dengan pilihan satu mode, pengerjaan berkali-kali, berlawanan dengan pilihan objek dan penerapan satu operasi pada satu waktu, bahwa aplikasi *EAGLE* (Anonymous, 2012) dengan empat *view* dasar, yaitu panel kontrol (*control panel*), kepustakaan (*library*), skematik (*schematic*), dan papan (*board*). *Control Panel* merupakan jendela utama, untuk peluncuran segala sesuatu yang lain dan saat penutupan, yaitu semua jendela bawahan ditutup. *Library* untuk pengaturan dan pengeditan bagian. *Schematic* untuk penggambaran skema untuk proyek, berupa pendefinisian bagian-bagian yang dimiliki dalam proyek, dan *pin* mana pada bagian-bagiannya harus dihubungkan. *Board* untuk perletakan potongan-potongan proyek dan secara fisik sebagai penghubung *pin* yang benar, seperti didefinisikan dalam skema (Anonymous, 2012).

Pandangan (*view*) dasar

a) *Control panel*

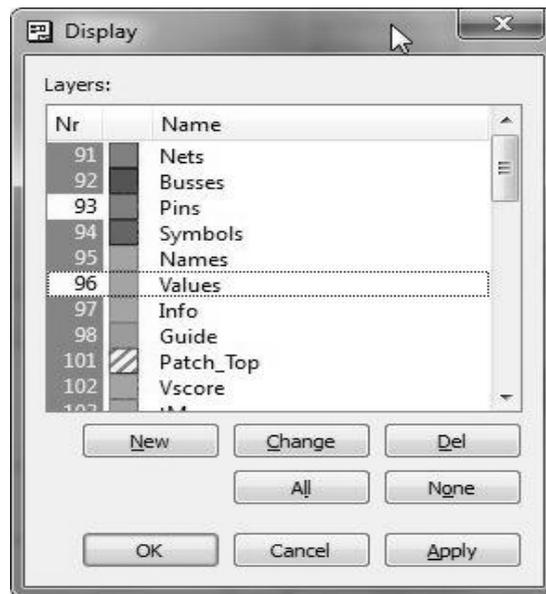
Tampilan pada *control panel* seperti ditunjukkan pada Gambar 1.



Gambar 1. Tampilan pada *control panel*

Berdasarkan Gambar 1 dapat dijelaskan, bahwa berbagai kategori di *control panel*, yaitu:

- *File* kepustakaan (.lbr files) sebagai penyimpan bagian-bagian individu yang ditambahkan ke papan;
- Aturan rancangan (.dru) merupakan *checker* aturan rancangan;

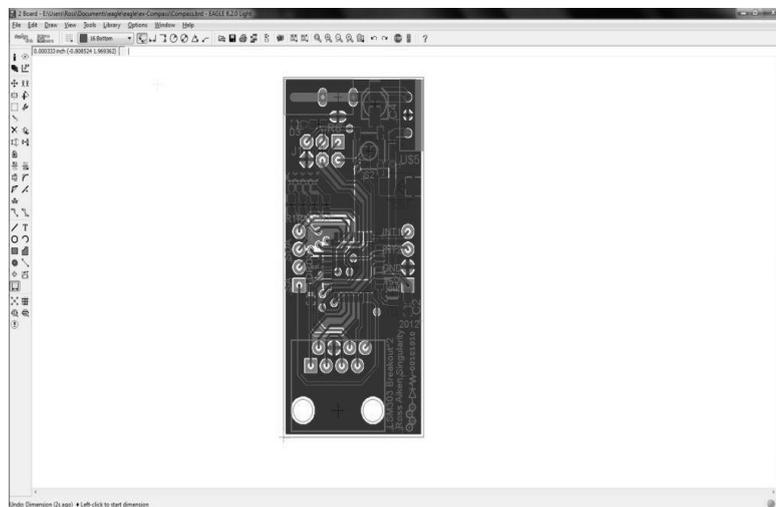


Gambar 2. Tampilan jendela *schematic*

Berdasarkan Gambar 2 dapat dijelaskan, bahwa jendela skematis sebagai tempat pembuatan dan peng-*edit*-an skematik.

c) Jendela papan (*board*)

Tampilan jendela *board*, seperti ditunjukkan pada Gambar 3 (Anonymous, 2012).



Gambar 3. Tampilan jendela *board*

d) Sejumlah definisi

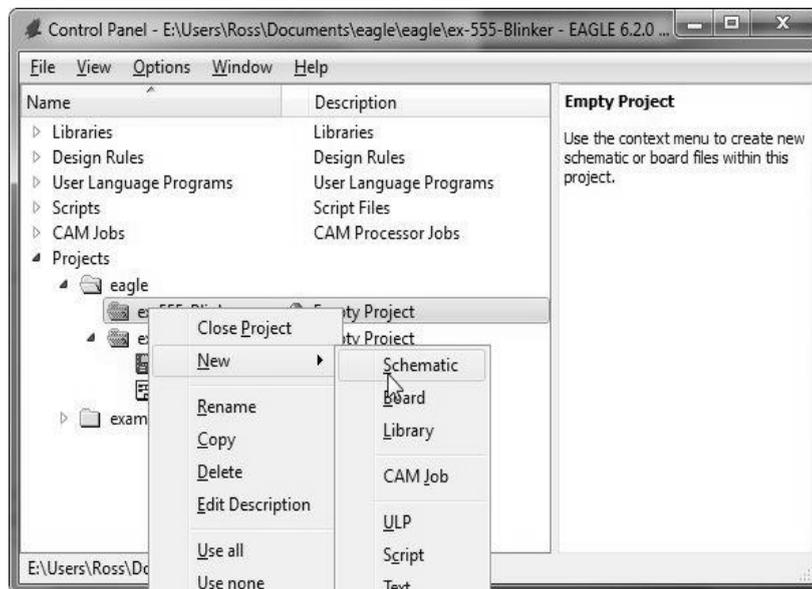
Sejumlah definisi yang perlu diketahui sebelum mulai pengerjaan, yaitu:

- *Part* (bagian), fisis komponen pada *PCB*, terdapat di perpustakaan,
- *Net* (jaring), sambungan listrik (dalam tampilan skematik),

- *Trace*, intinya sebuah kawat pada PCB (board view), digunakan untuk pembuatan koneksi yang didefinisikan oleh *Net*,
- *Plane* (bidang), biasanya dihubungkan ke *ground* (dalam beberapa kasus dihubungkan sebaliknya, biasanya untuk situasi pembawa arus tinggi),
- *Wire* (kawat), dalam tampilan skematik atau *board*, ini tidak dibagikan di antara keduanya, *Label*, penunjuk *Name* pada *Net*,
- *Name*, pengenal unik untuk setiap bagian, dan
- *Value* (nilai), nilai fisis komponen.

Berkreasi dengan proyek baru dan skematik

Tampilan awal saat berkreasi dengan proyek baru dan skematik seperti ditunjukkan pada Gambar 4.

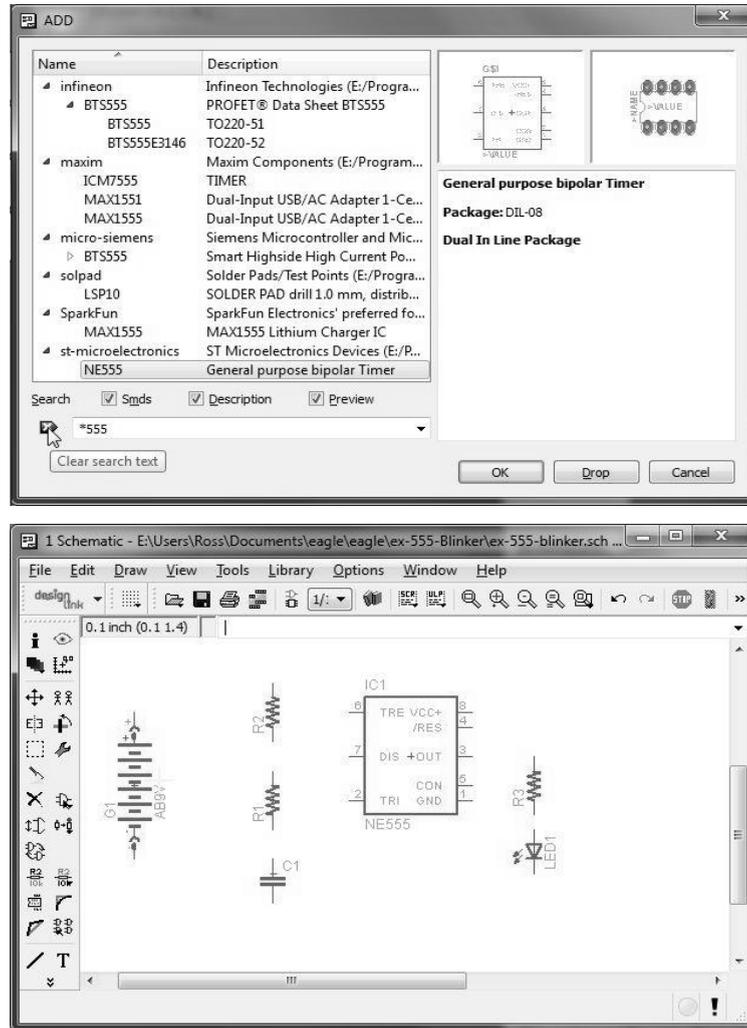


Gambar 4. Tampilan awal saat berkreasi dengan proyek baru dan skematik

Berdasarkan Gambar 4 dijelaskan, bahwa dikreasikan sebuah rangkaian *LED* berkedip dengan penggunaan *chip* pewaktu *IC 555* yang merupakan *chip* sederhana dengan multi fungsi. Pengoperasian mode stabil dengan hasil gelombang bentuk persegi pada kaki (*pin*) keluaran (*output*). Langkah-langkah lanjutan berupa:

- Klik *File* > *New* > *Project*
- Pemberian nama proyek secara deskriptif (misalnya: *ex-555-Blinker*)
- Klik kanan pada proyek, dan klik *New* > *Schematic*
- Beri nama skematik itu (misalnya: *ex-555-Blinker.sch*).

Tampilan pada penambahan bagian-bagian (*parts*) (Anonymous, 2012), seperti ditunjukkan pada Gambar 5.



Gambar 5. Tampilan pada penambahan bagian-bagian (*parts*)

Berdasarkan Gambar 5 dijelaskan, bahwa klik pada tombol *Add* (atau ketik *Add*), kemudian ketik *555 di kotak pencarian. Diinginkan salah satu dari *st-microelectronins library* dan dengan penggunaan proses tersebut, ditambahkan bagian-bagian lain, yaitu:

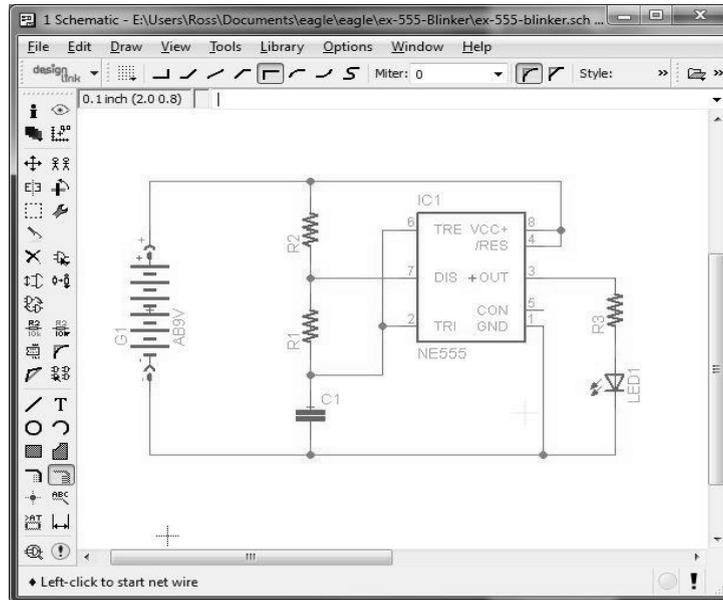
R1, R2, dan R3 - 'R0805' dari *library* 'resistor';

C1 - 'C0805' dari *library* 'capasitor';

LED1 - 'CHIPLED_0805' bagian dari *library* 'led'; dan

G1 - 'AB9V' bagian dari *library* 'battery'.

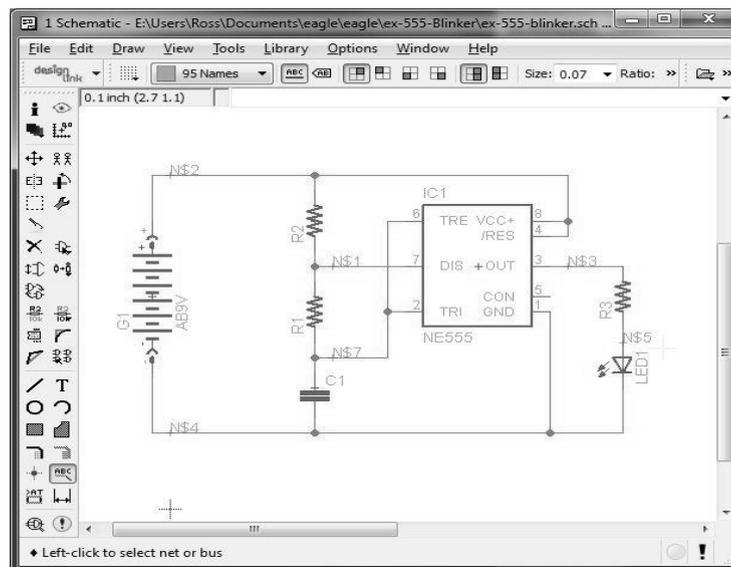
Penghubungan masing-masing bagian yang telah dipilih dari *library* (Anonymous, 2012), seperti ditunjukkan pada Gambar 6.

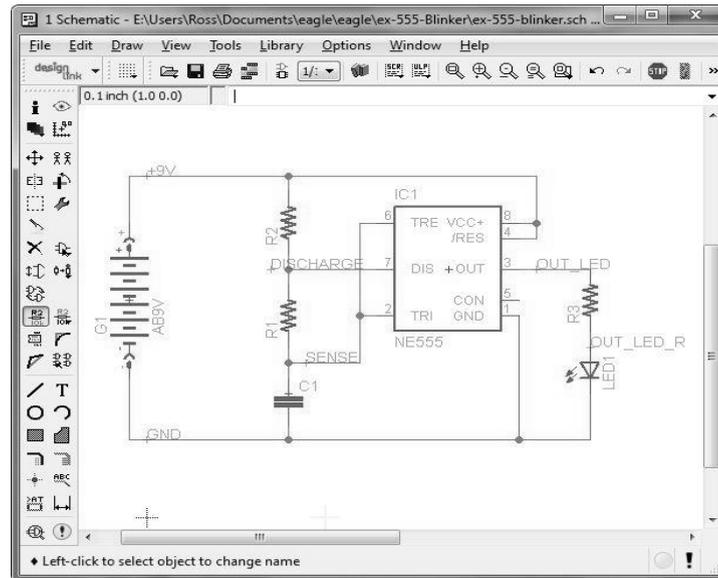


Gambar 6. Penghubungan masing-masing bagian yang telah dipilih dari *library*

Berdasarkan Gambar 6 dijelaskan, bahwa setelah diperoleh semua bagian yang tampak, kemudian dilakukan penghubungan melalui screenshot dengan penggunaan perintah 'net'.

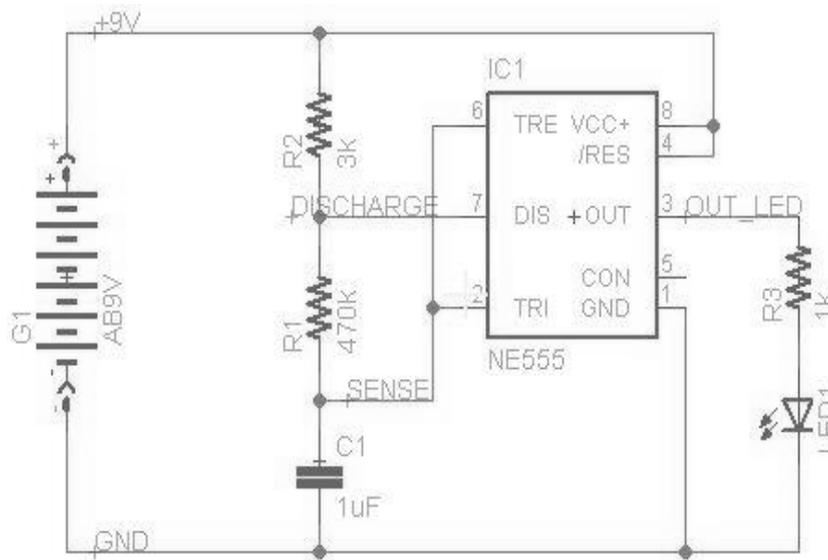
Tampilan pada pemberian *Label* dan *Name* semua bagian pada *net* (Anonymous, 2012), seperti ditunjukkan pada Gambar 7.





Gambar 7. Tampilan pada pemberian *Label* dan *Name* semua bagian pada *net*

Berdasarkan Gambar 7 ditunjukkan, bahwa dengan penggunaan perintah 'Label', klik masing-masing jaring (*net*) untuk pembuatan nama jala-jala muncul di kabel. Setelah semua nama ditampilkan, digunakan perintah 'Name' untuk pemberian nama yang sesuai dengan nama-nama *parts*. Pemberian nama pada rangkaian (Anonymous, 2012), seperti ditunjukkan pada Gambar 8.



Gambar 8. Pemberian nama pada rangkaian

Berdasarkan Gambar 8 ditunjukkan, bahwa penggunaan 'Value' (tepat di sebelah perintah 'Name'), diberi 'Label' semua resistor dan kapasitor dengan nilai yang sesuai. Nilai R3 sebagai penaham untuk pembatasan arus masuk ke *LED*. Nilai R3 sebesar 1 k Ω sebagai nilai perkiraan konservatif (tergantung pada spesifikasi *LED*).

Nilai C1, R1, dan R2 sebagai penentu tingkat *blinkiness LED*. Penggunaan persamaan (1) digunakan untuk penghitungan frekuensi kedip.

$$f = \frac{1}{[\ln 2 \times C \times (R1 + R2)]}$$

Untuk kondisi dimana nilai R1 = 470 kΩ, R2 = 3 kΩ, dan C1 = 1 μF, maka dinyalakan dan dimatikan 3 kali per detik.

Perolehan rangkaian sesuai Gambar 8, mutlak diperlukan pengecekan berdasarkan kaidah kelistrikan. Tampilan untuk pengecekan (Anonymous, 2012), seperti ditunjukkan pada Gambar 9.



Gambar 9. Tampilan untuk pengecekan

Berdasarkan Gambar 9 ditunjukkan, bahwa eksekusi terhadap ERC berakibat pada area terlihat "terjadi" kesalahan, sehingga perlu diperiksa pada *output* untuk baris satu demi satu. Penjelasan terhadap tampilan berupa uraian.

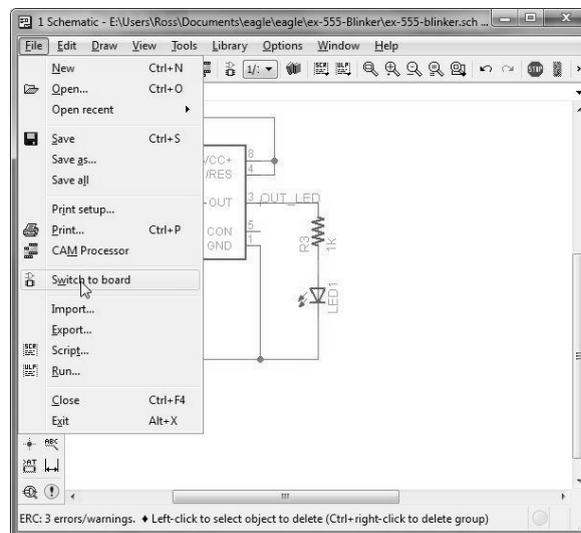
- Kesalahan (1): sebagai penjaminan, agar berhati-hati.

Tidak terdapat hubungan masukan (*input*) pada pin IC1 CON. Secara umum, *pin* pada masukan tidak terhubung. Dalam hal ini, *pin* CON adalah tegangan referensi yang dapat ditetapkan secara manual, namun tidak terdapat keburukan terjadi, jika dibiarkan tidak terhubung (*floating*, mengambang).

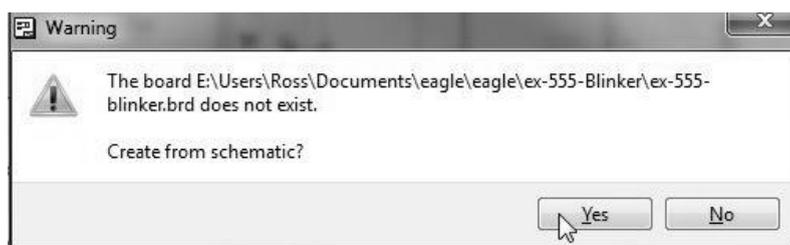
- Peringatan (2) - Ini tidak sepenting kesalahan, namun tetap dibutuhkan pandangan sepiantas. Satu peringatan yang harus dicari adalah tentang jaring (net) hanya punya satu simpul/pin. Itu berarti tidak terdapat hubungan jaring tersebut di kedua ujungnya.
 - Pin POWER IC1 VCC + terhubung ke + 9V-, dalam hal ini *EAGLE* beri peringatan, bahwa setiap kali dilakukan penghubungan tegangan yang berbeda secara bersama-sama (jika dilakukan penghubungan jalur catu daya 12 V dan jalur catu daya 5 V, hal buruk pasti terjadi). Dalam kasus ini, itu hanya perbedaan nomenklatur, jadi tidak apa-apa untuk disetujui.
 - Bagian LED1 tidak bernilai: perlu diberi nilai terhadap LED1, tetapi sampai saat itu, peringatan ini akan ada.
- Disetujui (0): setelah di-klik tombol 'Approve' pada peringatan/kesalahan, maka masuk ke tahapan selanjutnya.

Pembuatan *board layout*

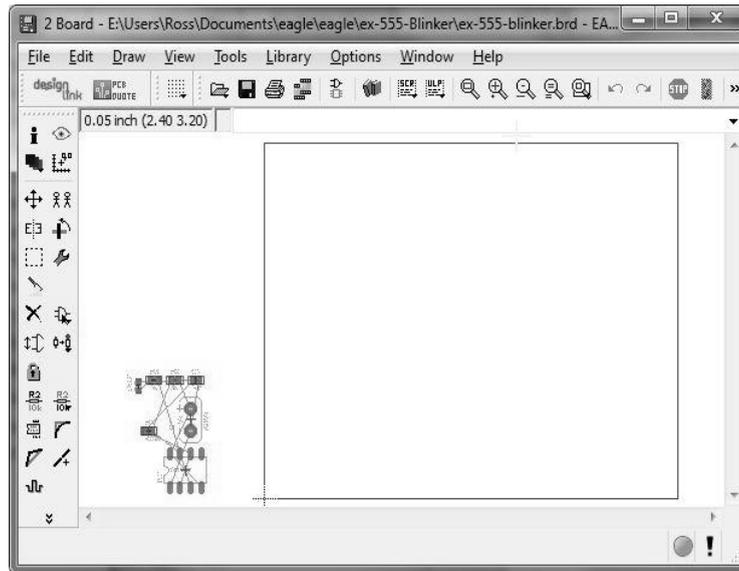
Tampilan langkah-langkah dalam pembuatan tata letak papan (*board layout*) (Anonymous, 2012), seperti ditunjukkan pada Gambar 10.



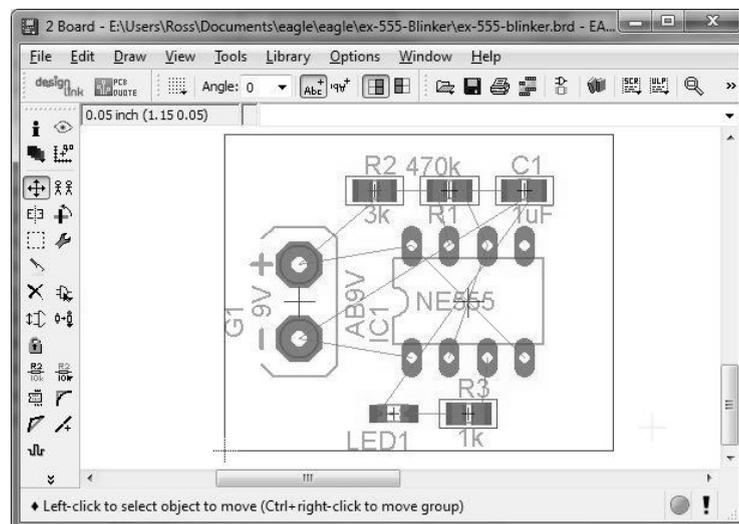
(10.a)



(10.b)



(10.c)



(10.d)

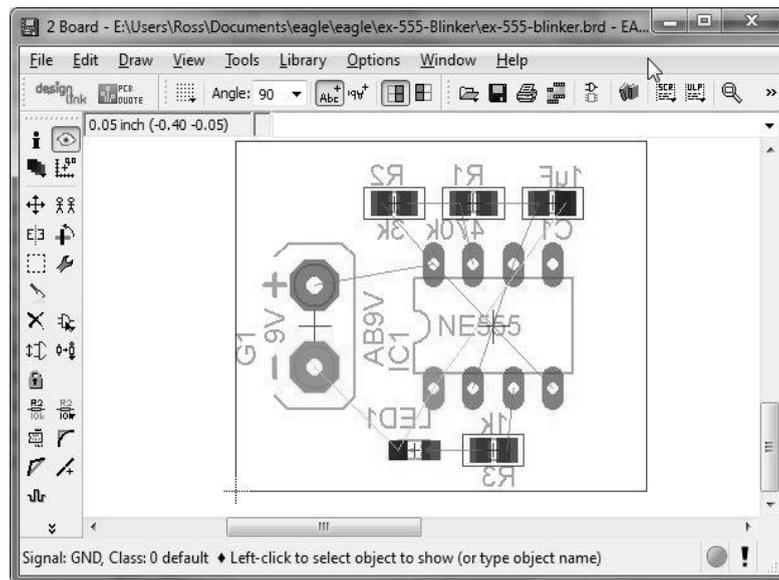
Gambar 10. Tampilan langkah-langkah dalam pembuatan *board layout*

Berdasarkan Gambar 10 ditunjukkan, bahwa pembuatan dasar tata letak (*layout*) PCB sisi bawah (*left side*) dari skematik itu sesuatu yang mudah. Pilih *File > Switch to board* dan timbul peringatan bahwa papan tidak ada, maka klik Yes untuk pembuatan papan (*board*) dari skematis. Saat *file board* muncul, muncul kotak di layar, dengan semua bagian di sebelah kirinya, kemudian dipindahkan pada keterwakilan area dengan penempatan bagian dalam versi gratis dari *EAGLE*.

Perlu diperhatikan, bahwa semua komponen yang ditambahkan ke skematik di luar area yang dapat ditempatkan, dihapus *EAGLE*. Setelah pemindahan bagian dari tempat semula, maka harus disimpan di dalam kotak ukuran 4" x 3.2" atau 100 x 80

mm. Semua bagian dipindahkan ke dalam konfigurasi yang mirip dengan gambar terakhir (10.d). Perlu perhatian, bahwa langkah tersebut perlu sejumlah pemikiran untuk penyelamatan diri di kemudian hari dengan asing-masing garis emas itu terwakili jejak yang tidak berubah.

Pembuatan tata letak PCB sisi sebaliknya (tempat penyolderan, *right side*), dilakukan dengan cara dan pola yang sama sebagaimana untuk sisi sebelumnya (tempat keberadaan komponen, *left side*). Tampilan tata letak papan pada *right side* (Anonymous, 2012), seperti ditunjukkan pada Gambar 11.

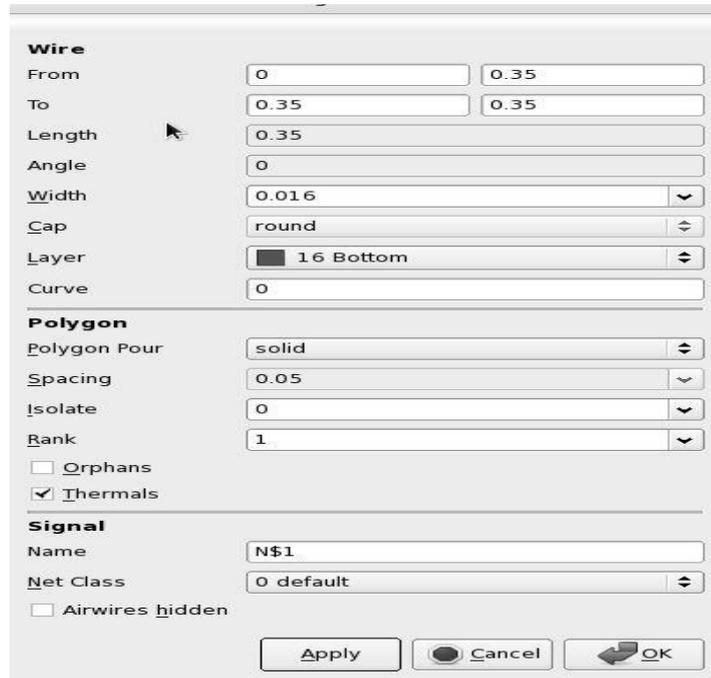


Gambar 11. Tampilan tata letak papan pada *right side*

Berdasarkan Gambar 11 ditunjukkan, bahwa tampilan papan rangkaian tercetak (PCB), merupakan sekeping PCB dengan dua sisi (*two sides*). Untuk logistik penyolderan dilakukan melalui lubang-lubang, sehingga diperlukan penggunaan bagian bawah PCB (*right side*). Guna keperluan tersebut digunakan perintah *Mirror* dan klik pada permukaan bagian *mount* untuk beralih ke lapisan bawah.

Untuk kondisi dimana diperlukan perbaikan orientasi komponen, maka digunakan perintah *Rotate* or *Move*. Kepemilikan semua bagian yang ditata, dijalankan perintah *Ratsnest*. Perintah *Ratsnest* sebagai penghitung ulang jalur terpendek untuk semua kabel yang tidak dilepas (*airwires*), yang harus dibersihkan dari kesemrawut di layar dengan jumlah yang wajar.

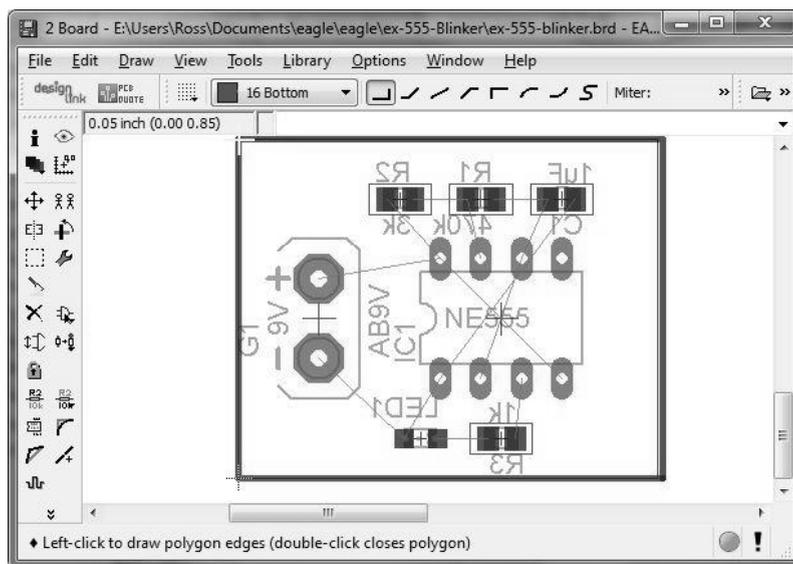
Perintah pembuatan bidang dasar (*ground plane*) (Anonymous, 2012), seperti ditunjukkan pada Gambar 12.



Gambar 12. Perintah pembuatan bidang dasar (*ground plane*)

Berdasarkan Gambar 12 ditunjukkan, bahwa dalam pembuatan bidang dasar diperlukan langkah-langkah penyetelan atau pengaturan dimensi, agar waktu peng-etsa-an dapat dikurangi saat pembuatan *PCB*.

Tampilan jendela untuk pembuatan bidang dasar (*ground plane*) (Anonymous, 2012), seperti ditunjukkan pada Gambar 13.



Gambar 13. Tampilan jendela untuk pembuatan bidang dasar (*ground plane*)

Berdasarkan Gambar 13 ditunjukkan, bahwa inti dari pembuatan bidang dasar (*ground plane*), adalah pengambilan semua area yang tidak terpakai di *PCB* dan dihubungkan ke *ground net*. Setelah pembuatan bidang pengoperasian *ratsnest*, luas area bidang yang tertinggal harus turun drastis. Untuk pembuatan bidang dasar (*ground plane*) di *EAGLE*, dioperasikan dengan perintah *Polygon* saat pengaturan *Layer to Bottom* (warna biru) dan perlacakan kotak di sekitar bagian luar.

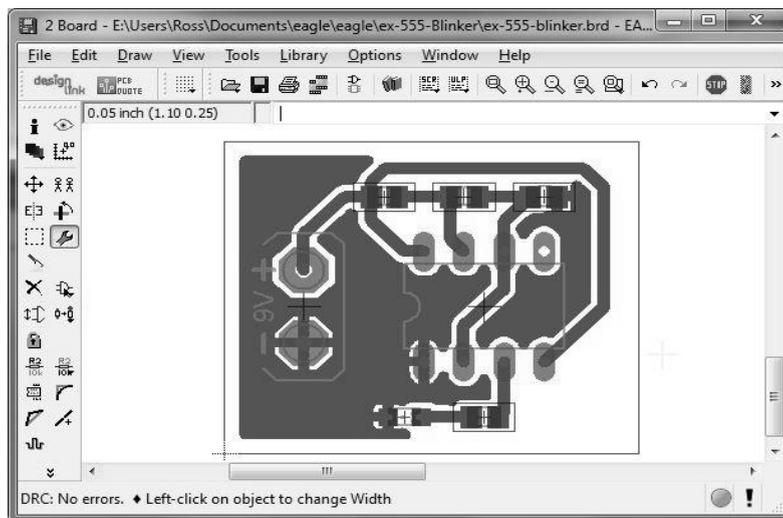
Penggunaan perintah *Name*, di-klik pada garis putus-putus warna biru untuk penggantian nama bidang menjadi *GND*. Pengoperasian dengan perlakuan klik kanan pada garis putus-putus warna biru dan klik *Properties*, jendela yang mirip dengan penunjukan pada langkah ditampilkan. Hal itu sebagai penanda kemunculan masing-masing kotak dengan penjelasan, bahwa:

- *From, To, Length, Angle* (Dari, Untuk, Panjang, Sudut) sebagai penjas segmen garis yang di-klik;
- *Width* (lebar) merupakan penjelasan tentang lebar minimum (dalam *mils*) pada bidang dasar;
- *Cap* sebagai tanda tidak diberlakukan;
- *Curve* sebagai Bend dari segmen garis itu, kecuali diketahui apa yang dilakukan, maka tinggalkan hal ini di nilai 0;
- *Polygon Pour* sebagai penjaga agar tetap solid, hal itu berpengaruh pada pola;
- *Spacing* sebagai jarak antara garis isi saat dituangkan ke '*Hatch*';
- *Isolate* sebagai jarak antara isi dan jejak apapun (jika kurang dari aturan rancangan, digunakan jarak yang ditentukan dalam aturan perancangan);
- *Orphans* sebagai hal yang tidak terkendali, *EAGLE* hanya pengisian wilayah bersebelahan yang kosong terbesar. Untuk kondisi dicentang, area dapat diisi tanpa harus terjadi pelanggaran aturan rancangan;
- *Thermal* - Ini agak sulit untuk dijelaskan, perhatikan saja bantalan / pin pada bagian dan ubah pengaturannya. Pada dasarnya, setelah mengaktifkan ini membuatnya lebih mudah untuk menyolder bagian-bagian yang terhubung ke pesawat besar, dengan mengorbankan kinerja dengan jejak saat ini sangat tinggi.
- *Name* - Nama poligon (meski Anda bisa mengedit kotak ini, Eagle akan berteriak pada Anda jika Anda mencoba mengganti namanya)
- *Net Class* sebagai tanda penambahan kelas jala tambahan (missal: *Power*) dengan persyaratan berbeda yang dimungkinkan untuk dipilih; dan

- *Airwires* sebagai tanda untuk pencentangan yang dapat dilakukan pada kotak ini, agar semua *airwires* agar tersembunyikan, meskipun bidang tidak terhubung.

Pembuatan jalur-jalur penghubung antar bagian (*routes of parts*)

Berpedomana pada jendela sebagaimana ditunjukkan pada Gambar 13, maka dengan penggunaan perintah *Route* dan di-klik pada *airwire* apapun, maka jejak warna biru harus muncul di simpul *airwire* yang paling dekat dengan tempat penge-klik-an. Tampilan pembuatan jalur-jalur penghubung antar bagian (Anonymous, 2012), seperti ditunjukkan pada Gambar 14.



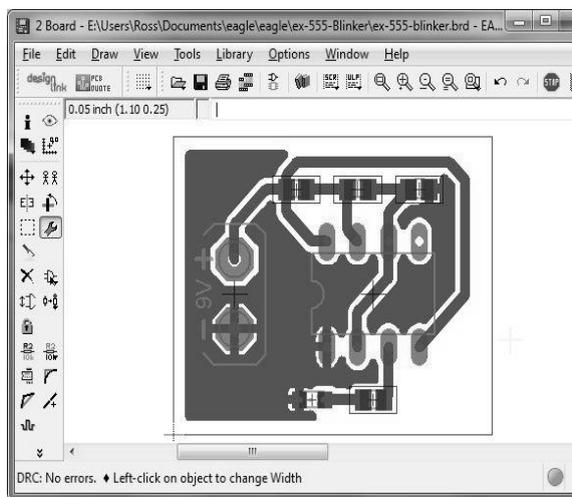
Gambar 14. Tampilan pembuatan jalur-jalur penghubung antar bagian

Berdasarkan Gambar 14 dapat dijelaskan, bahwa jika muncul jejak warna merah, maka hal itu bukan yang diinginkan. Perubahan dilakukan pada *layer* dari atas (warna merah) ke arah bawah (warna biru). Penge-klik-an dilakukan kembali di tempat lain untuk jangkar kawat pada saat itu. Perubahan terhadap sudut belokan dilakukan dengan penge-klik-an kanan (ctrl-klik kanan untuk beralih di antara putaran sudut yang sama), dan di-klik di tengah untuk pembuatan lapisan antara lapisan atas dan bawah.

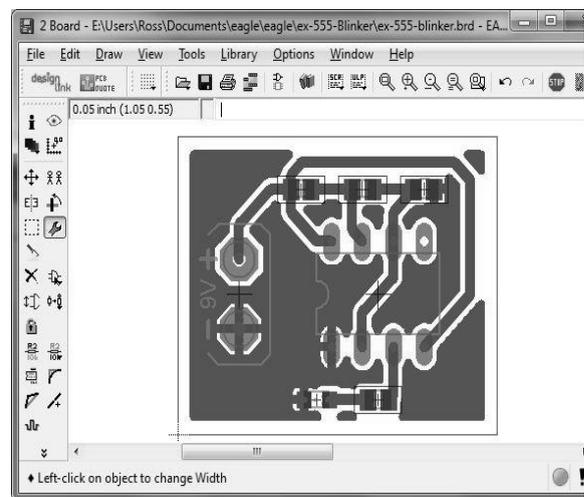
Untuk kondisi dimana terdapat halangan pada bidang dasar (ground plane), maka dapat dioperasikan *Ratsnest* untuk pembuatan penghitungan ulang ke mana arahnya atau di-klik '*Ripup*', kemudian di-klik di perbatasan bidang. Pemotongan terhadap batas bidang tetap beroperasi sampai dioperasikan *Ratsnest*. Saat dilakukan perutean dengan penggunaan sudut belokan 45 derajat, bukan putaran 90 derajat seperti yang digunakan dalam skema.

Berdasarkan pembuatan rangkaian dengan kecepatan rendah, hal itu merupakan satu dari sedikit konsesi yang harus dibuat, agar fenomena fisika pada tikungan tajam sebagai penyebab keseluruhan masalah *funky*, mulai dari elektron yang bocor dari titik, kerancuan saat di sudut dalam, hingga ketidakcocokan nilai impedans sebagai penyebab derau, pantulan, dan keseluruhan masalah lain. Pembuatan contoh *routing* tersebut bukanlah satu-satunya cara yang dapat dilakukan, juga bukan cara terbaik, tetapi inilah hasil yang dapat diterima.

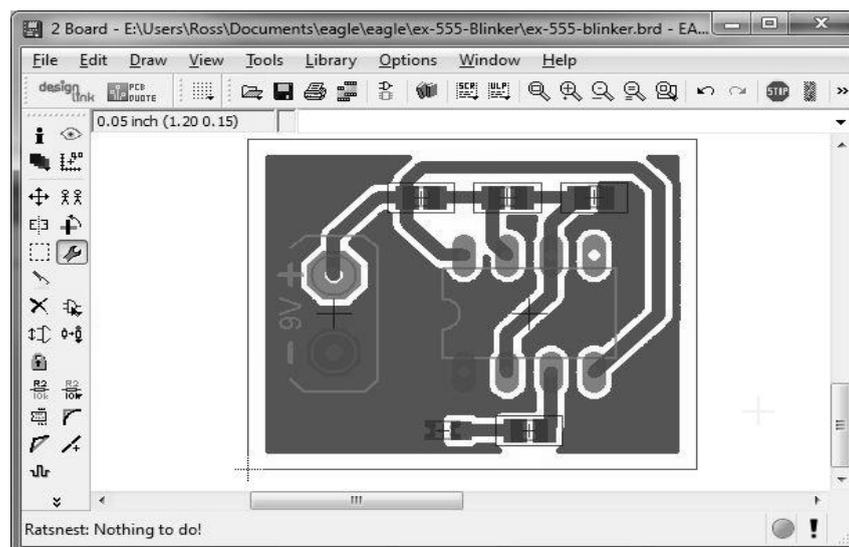
Peninjauan kembali terhadap jalur-jalur untuk pembuangan panas dan berdiri sendiri (Anonymous, 2012), seperti ditunjukkan pada Gambar 15.



(15.a)



(15.b)



(15.c)

Gambar 15. Peninjauan kembali terhadap jalur-jalur untuk pembuangan kalor (energi panas) dan berdiri sendiri

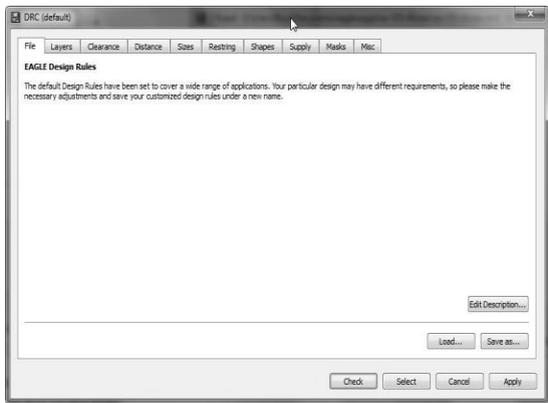
Berdasarkan Gambar 15 ditunjukkan, bahwa terdapat tiga *screencaps* berbeda, yaitu (a) papan dengan pembuangan kalor dan tidak terdapat jalur berdiri sendiri, (b) papan dengan pembuangan kalor dan terdapat jalur berdiri sendiri, dan (c) papan dengan jalur berdiri sendiri, tetapi tidak terdapat jalur pembuangan kalor.

Pengecekan aturan untuk rancangan (*design rule check, drc*)

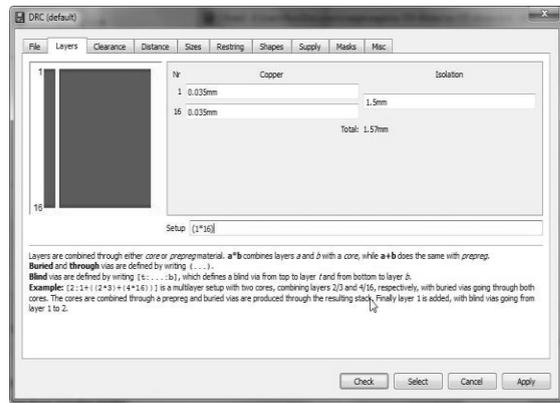
Pengecekan aturan untuk rancangan dilakukan dalam dua tahapan, yaitu pengaturan dan perolahan hasil.

- a) Tahapan pengaturan melalui pemilihan nilai-nilai pada pengecekan aturan untuk rancangan

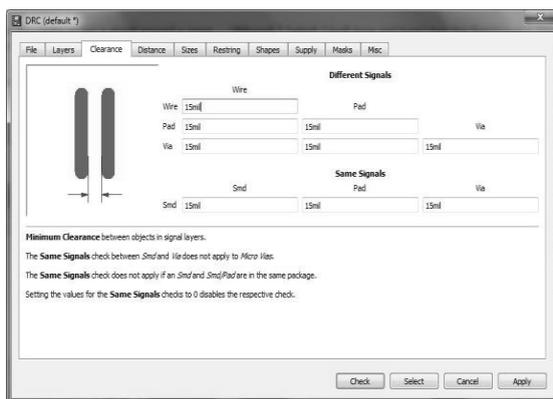
Tampilan untuk pengecekan aturan rancangan (*drc*) (Anonymous, 2012), seperti ditunjukkan pada Gambar 16.



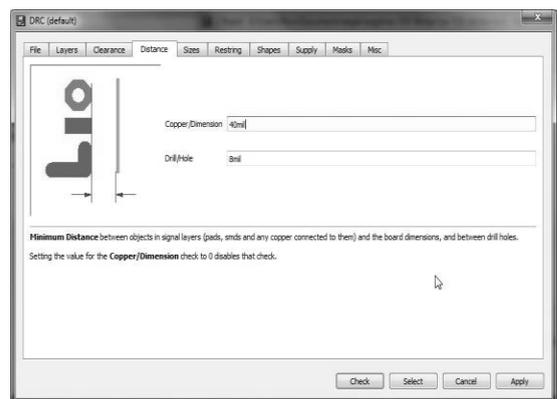
(16.a)



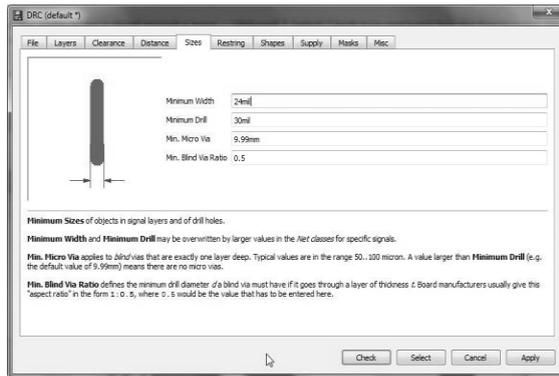
(16.b)



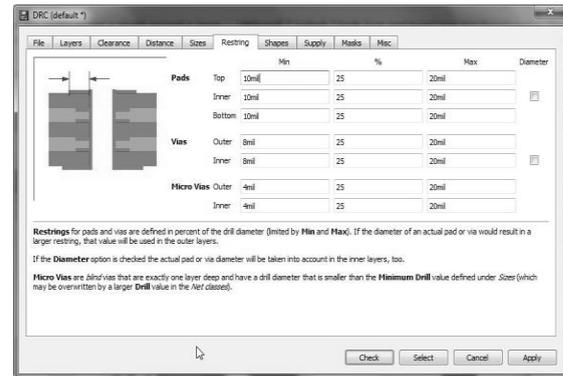
(16.c)



(16.d)



(16.e)



(16.f)

Gambar 16. Tampilan untuk pengecekan aturan rancangan (*drc*)

Berdasarkan Gambar 16 dapat dijelaskan, bahwa terdapat tampilan 10 (sepuluh) bagian, yaitu *file*, *layers*, *clearance*, *distance*, *sizes*, *restring*, *shapes*, *supply*, *masks*, dan *misc*. Dalam hal *Design Rule Check (DRC)* untuk pemeriksaan papan yang dibuat dengan serangkaian aturan untuk penentuan, apakah terdapat kesalahan meski kesalahan tersebut tidak sepenuhnya (smpurna), tetapi akan tertangkap banyak kesalahan secara umum. Kesepuluh bagian tersebut dijelaskan:

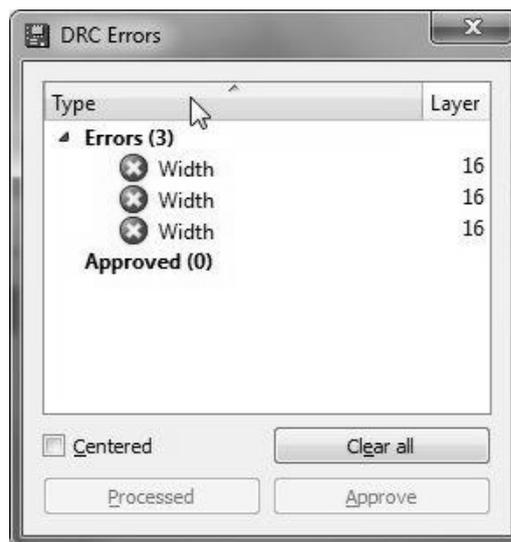
- i) *File* - dimungkinkan pemilihan *file DRC* yang akan digunakan atau sudah dimiliki, dan layanan seperti OSH Park juga dengan file *.drc* yang dapat diunduh dan dimuatkan di sini;
- ii) *Layers* - digunakan versi *freeware* dari EAGLE, jika dimiliki salah satu lisensi berbayar, lapisan dapat ditambahkan dengan perubahan penyiapan menjadi seperti (1*2*15*16);
- iii) *Clearance* – pemberitahuan dari EAGLE, seberapa banyak ruangan diinginkan antara berbagai jenis kontak listrik di papan tulis, untuk kondisi dimana tanda sama dengan 0, *OSH Park* dengan jarak minimum 6 *mils* untuk segala hal yang berada di bawah tanda berbeda atau dapat dengan ukuran 20 *mils*;
- iv) *Distance* – lebar tembaga atau dimensi merupakan jarak dari perutean ke tepi papan, pengeboran atau pelubangan dengan jarak dari *routing* ke lubang di papan tulis;
- v) *Sizes* - ditetapkan ukuran minimum, lebar minimum yang dapat dipilih, *OSH Park* dengan 6 *mils* atau dapat dengan ukuran 24 *mils*. minimum pengeboran untuk *OSH Park* adalah 13*mils*;

- vi) *Restrings* – saat pengontrolan ukuran bantalan lubang, biasanya pada nilai *default* yang sudah ada;
- vii) *Shapes* - dimungkinkan untuk pembuatan bantalan dengan tepi yang membulat, biasanya juga tidak tersentuh untuk yang satu ini;
- viii) *Supply* – pembuangan kalor tersebut sebelumnya, perlu pemberian sedikit kontrol lebih, kotak centang dimungkinkan untuk penyalaan pembuangan;
- ix) *Mask r* – apabila *PCB* dibuat, telah dengan lapisan sebagai penutup dan pelindung untuk semua papan, kecuali untuk area yang ditunjuk oleh *masker stop*, dimana setelah dipabrikasi, atau dilakukan perakitan oleh mesin (atau dengan *oven reflow*), stensil dibuat dan pasta solder diterapkan melalui stensil tersebut ke bantalan yang perlu disolder, pengontrolan ukuran bukaan di *masker stop* dengan nilai *default* yang tersedia; dan
- x) *Misc (miscellaneous)* serba aneka, meliputi:
 - Check Grid* - saat perletakan dan perutean *PCB*, semua yang dilakukan terdapat di kotak dengan ukuran yang ditentukan oleh perintah *Grid*, untuk kondisi dimana *grid* diubah secara parsial melalui tata letak (perutean), dipastikan bahwa semua bagian dipatuhi pengaturan baru;
 - Check Angle* - pengarahannya dengan gaya yang berbeda dari belokan 45 atau 90 derajat, atau jika dilakukan pemindahan bagian setelah perutean, diperiksa kotak ini dibuat EAGLE saat dilakukan hal itu;
 - Check Font*
 - Check Restrict*
 - Max. length difference in differential pairs*, perbedaan panjang maksimum pada bagian-bagian berbeda, merupakan *routing* berbeda di mana dua *trace* yang disalurkan secara berdampingan dan sebagai pembawa sinyal berbeda (dimisalkan: kawat A adalah 1, kawat B diatur ke 0; jika A-B lebih besar dari nol, angka 1 adalah dikirim, jika tidak 0 sedang dikirim). Sejumlah keuntungan untuk tujuan tersebut, salah satunya berupa '*Common Mode Rejection*'. Intisari dari kebanyakan derau listrik sedikit banyak sebagai penambah tegangan pada kawat yang diberikan. Untuk kondisi dimana kedua kabel cukup dekat, tegangan yang sama (sebut saja V), ditambahkan ke keduanya. Penggunaan *routing* berbeda, diinginkan kedua kabel tersebut terpengaruh oleh V yang sama, sehingga V sebagai hasil dari $(A + V) - (B + V)$.

Pemilihan dengan perbedaan panjang menjadi minimal, sehingga kabel sebagai pengambil derau yang sama. Hal ini diungkapkan dipilih perbedaan terhadap apa yang harus dipilih. Faktor celah (*gap*) untuk belokan pada pasangan berbeda pada kasus sinyal dengan kecepatan tinggi di kedua kabel pada pasangan berbeda menjadi panjang yang sama. Bergantung pada perutean, maka hal itu tidak menjadi masalah, sehingga untuk pembuat *trace* yang lebih pendek sesuai dengan yang lama, sebuah 'liku-liku' dimasukkan ke dalam. Faktor *gap* disesuaikan dengan ukuran ini. Setelah perolehan semua pengaturan dengan cara yang benar, dilakukan penge-klik-an *Check*, tahapan pengaturan semua ini diakhiri dengan pengetikan 'drc', sedangkan untuk pegoperasian *DRC* lagi ditekan *enter* dua kali.

Tahapan perolehan hasil pada pengecekan aturan untuk rancangan

Tampilan saat awal perolehan hasil pada pengecekan aturan untuk rancangan (Anonymous, 2012), seperti ditunjukkan pada Gambar 17.



Gambar 17. Tampilan saat awal perolehan hasil pada pengecekan aturan untuk rancangan

Berdasarkan Gambar 17 dapat dijelaskan, bahwa terdapat tiga kesalahan karena keberadaan *trace* yang tidak cukup lebar. Untuk perbaikan lebar *trace* terdapat dua pilihan, yaitu: i) di-klik bagian kanan pada *trace*, klik *Properties*, dan atur lebar atau ii) di-klik *Change ... (the wrench)*, klik *Width*, dan pilih *new width*. Alternatif yang dapat dilakukan, yaitu ketik '*Change width .024*' atau apapun yang diinginkan untuk ubah, lalu klik pada *trace* yang bersinggungan) atau dapat dipelajari cara penggunaan

Change dan berdasarkan pengalaman dalam penggunaan EAGLE akan diperoleh hasil yang jauh lebih baik.

Penjelasan tambahan tentang pengalaman beberapa pelaku, LED di sisi bawah PCB sulit untuk dilihat apakah PCB dipasang *flush* terhadap sesuatu atau bukan, sehingga dapat dihindari terjadi kesalahan dalam hal yang sama. Penggunaan perintah *Ripup*, dilepaskan *trace* yang terarah ke LED, melalui perintah *Mirror*, pindahkan LED ke *layer* atas, kemudian digunakan perintah *Route*, arahkan *trace* di lapisan bawah sekitar setengah jalan ke LED untuk setiap *airwire*. Tombol tengah pada *mouse* di-klik untuk membawa *trace* ke lapisan atas, sehingga dapat diubah lapisan dari 'Bawah' ke 'Atas' dengan kotak *drop-down* di kiri atas. Langkah lanjutan, di-klik pada bantalan LED untuk penyelesaian akhir *trace* dan gambar segera ditampilkan. Setelah selesai dengan proses perutean dan tidak terdapat kesalahan pada *DRC*, maka telah selesai perihal implementasi literasi terhadap aplikasi EAGLE, sehingga dapat dilanjutkan dengan tahapan pabrikasi PCB.

SIMPULAN

Berdasarkan pembahasan, maka dapat ditarik simpulan sesuai tujuan implementasi literasi ini.

- 1) Penjelasan umum terhadap aplikasi EAGLE yang merupakan salah satu dari sekian banyak aplikasi untuk rancangan dalam pembuatan keping papan tercetak (*printed circuit board, PCB*) berbantuan komputer (*computer aided design, CAD*), sehingga dikatakan sebagai *editor* tata letak yang mudah digunakan untuk bantuan rancangan papan rangkaian tercetak (*PCB*). Aplikasi EAGLE terdiri atas tiga modul utama, yaitu *Editor Tata Letak*, *Editor Schematic*, dan *Autorouter* yang disematkan dalam satu antarmuka pengguna. Ketiganya dapat diakses dari EAGLE's User Interface (UI) atau antarmuka pengguna pada EAGLE, sehingga tidak perlu pengubahan *netlists* antara skema dan tata letak. Sejumlah keunggulan aplikasi EAGLE, yaitu (i) lintas program (*cross platform*), dimana aplikasi EAGLE dapat beroperasi pada sistem pengoperasian (*Operating System, OS*), baik Windows, Mac, atau Linux, (ii) kebutuhan untuk penyimpanan yang kecil, terdapat edisi bebas pakai (*free of charge*) atau dapat juga dipilih yang berbiaya murah (*low cost*), dan (iii) dukungan masyarakat (*community support*).

- 2) Catatan penting untuk implementasi literasi terhadap aplikasi *EAGLE*, dikelompokkan pada lima catatan penting, yaitu (a) pandangan (*view*) dasar, (b) berkreasi dengan projek baru dan skematik, (c) pembuatan *board layout*, (d) pembuatan jalur-jalur penghubung antar bagian (*routes of parts*), dan (e) pengecekan aturan untuk rancangan (*design rule check, drc*).

DAFTAR PUSTAKA

- (Afandi, 2017) M.Y. Afandi, A. Goeritno, R. Yatim (2017, Jul. 25). "Minimum System Berbasis Mikrokontroler ATmega32 Berbantuan Sensor Passive Infrared Receiver dan Fingerprint Untuk Sistem Pengamanan Kendaraan Bermotor Roda Empat atau Lebih," di *Prosiding SNATIF ke-4 tahun 2017*, Fakultas Teknik Universitas Muria Kudus. [Online]. 4, hlm. 195-210. Tersedia di: <https://jurnal.umk.ac.id/index.php/SNA/article/view/1264/879>.
- (Anonymous, 2012) Killersquire11. (2012, Sept. 29). *PCB Creation with EAGLE for Beginners*. [Online]. Available: <http://www.instructables.com/id/PCB-Creation-with-Eagle-for-Beginners/>
- (Aono, 2011) K. Aono. 2011. "Application Note: PCB Design with EAGLE," in ECE480 Design Team 5, Department of Electrical & Computer Engineering, Michigan, MI: Michigan State University, pp. 1-33.
- (Bate, 2011) M. Bates. 2011. "Chapter 2. Microcontroller Operation," in *PIC Microcontrollers: An Introduction to Microelectronics*, 3rd Edition. Elsevier Ltd., pp. 17-24.
- (CadSoft, 2010) CadSoft. 2010. *Eagle Easily Applicable Graphical Layout Editor Manual Version 5*. Palm Beach, FL: CadSoft Computer Inc., pp. 37-80.
- (Clarke, 2008) T. Clarke. 2008. "The EAGLE Schematic & PCB Layout Editor - A Guide," in Course Material, Department of Electrical & Electronic Engineering. London: Imperial Collage London, pp. 1-17.
- (Effendi, 2015) R. Effendi, A. Goeritno, R. Yatim. (2015, Nov. 17). "Prototipe Sistem Pendeteksian Awal Pencemaran Air Berbantuan Sensor Konduktivitas dan Suhu Berbasis Mikrokontroler," di *Prosiding Seminar Nasional Sains dan Teknologi (Semnastek) 2015*, Fakultas Teknik, Universitas Muhammadiyah Jakarta. [Online]. 2, hlm. (TE-017) 1-6. Tersedia di: <https://jurnal.umj.ac.id/index.php/semnastek/article/view/430/396>.
- (Ginting, 2016) S.F. Ginting, A. Goeritno, R. Yatim, "Kinerja Sistem Pengontrolan Berbantuan Sensor Voice Recognition dan Mikrokontroler ATmega16 untuk Pengoperasian Aktuator," di *Prosiding SNTI V-2016 Universitas Trisakti*, Jakarta, 18 Mei 2016, hlm. 359-365.
- (Goeritno, 2014) A. Goeritno, D.J. Nugroho, R. Yatim. (2014, Nov. 14). "Implementasi Sensor SHT11 Untuk Pengkondisian Suhu dan Kelembaban Relatif Berbantuan Mikrokontroler," di *Prosiding Seminar Nasional Sains dan Teknologi*

- (*Semnastek*) 2014, Fakultas Teknik, Universitas Muhammadiyah Jakarta. [Online]. 1, hlm. (TE-009) 1-13. Tersedia di: <https://jurnal.umj.ac.id/index.php/semnastek/article/view/233/208>.
- (Goeritno, 2016a) A. Goeritno, S.F. Ginting, "Prototipe Sistem Pengontrolan Berbasis Voice Recognition Sensor dan Mikrokontroler untuk Pengoperasian Aktuator," pada *JUTEKS Jurnal Teknik Elektro dan Sains*, Program Studi Teknik Elektro Fakultas Teknik Universitas ibn Khaldun Bogor, vol. 3, no. 2, Oktober 2016. [Online]. 3(2), hlm. 7-15. Tersedia di: <http://ejournal.uika-bogor.ac.id/index.php/JUTEKS/article/view/331/eng>
- (Goeritno, 2016b) A. Goeritno, Saefurrochman. 2016. "Modul Peranti Elektronika Berbasis Mikrokontroler Untuk Sarana Pembelajaran Sistem Pengontrolan pada Program Studi Teknik Elektronika Sekolah Menengah Kejuruan" di *Prosiding the 4th National Conference on Industrial Electrical and Electronics (NCIEE) 2016*, Jurusan Teknik Elektro, Universitas Sultan Ageng Tirtayasa (Untirta), Cilegon, 12-14 Oktober 2016, hlm. 80-90.
- (Goeritno, 2016c) A. Goeritno, R. Effendi, R. Yatim. (2016, Nov. 8). "Implementasi *Contacting Conductivity Sensor* dan *Thermistor* Berbasis Mikrokontroler ATmega32 Untuk Pendeteksian Awal Kualitas Air," di *Prosiding Seminar Nasional Sains dan Teknologi (Semnastek) 2016*, Fakultas Teknik, Universitas Muhammadiyah Jakarta. [Online]. 3, hlm. (TE-018) 1-12. Tersedia di: <https://jurnal.umj.ac.id/index.php/semnastek/article/view/693/624>.
- (Goeritno, 2016d) A. Goeritno, B.A. Prakoso, B.A. Prakoso (2016, Dec. 7). "Kinerja Sistem Kontrol Berbasis Mikrokontroler Untuk Pemantauan Sejumlah Parameter Fisis pada Analogi Smart Green House," di *Prosiding Simposium Nasional Rekayasa Aplikasi Perancangan dan Industri (RAPI XV) 2016*, Fakultas Teknik, Universitas Muhammadiyah Surakarta. [Online]. 15, hlm. 70-76. Tersedia di: https://publikasiilmiah.ums.ac.id/bitstream/handle/11617/8172/B49_Arief_%20Goeritno.pdf?sequence=1&isAllowed=y.
- (Goeritno, 2017a) A. Goeritno, B.A. Prakoso, B.A. Prakoso. 2017. "Tampilan Sejumlah Parameter Fisis pada Analogi *Smart Greenhouse* Berbantuan Mikrokontroler ATmega32, Bluetooth, dan *Smart Phone* Berbasis Android," di *Prosiding Seminar Nasional Multidisiplin Ilmu*, Universitas Budu Luhur, Jakarta, 22 April 2017, hlm. (ICT) 225-234.
- (Goeritno, 2017b) A. Goeritno, I. Mustofa. (2017, Jun.). "Minimum System Berbasis Mikrokontroler ATmega32 untuk Pemantauan dan Tampilan Kondisi Instalasi Kelistrikan Otobis," di *Jurnal Setrum*, Vol. 6, No.1 Juni 2017. [Online]. 6(1), hlm.55-67. Tersedia di: <https://jurnal.untirta.ac.id/index.php/jis/article/view/1721/2122>
- (Goeritno, 2017c) A. Goeritno, S.F. Ginting, R. Yatim. (2017, Juli, 25). "Pengoperasian Beban Listrik Fase Tunggal Terkendali melalui Minimum System Berbasis Mikrokontroler dan Sensor Voice Recognition (VR)," di *Prosiding SNATIF ke-4*

tahun 2017, Fakultas Teknik Universitas Muria Kudus. [Online]. 3, hlm. 63-78. Tersedia di: <https://jurnal.umk.ac.id/index.php/SNA/article/view/1244/862>.

- (Goeritno, 2017d) A. Goeritno, Sopyandi, R. Yatim. (2017, Juli, 25). "Beban-beban Listrik Terkontrol melalui Minimum System Berbasis Payload Data Handling Berbantuan Mikrokontroler," di *Prosiding SNATIF ke-4 tahun 2017*, Fakultas Teknik Universitas Muria Kudus. [Online]. 4, hlm. 223-238. Tersedia di: <https://jurnal.umk.ac.id/index.php/SNA/article/view/1270/882>.
- (Gong, 2017) L. Gong. (2017, March 13). *Top 9 Best PCB Design Software of 2018*. [Online]. Available: <http://www.seeedstudio.com/blog/2017/03/13/pcb-design-software/>
- (Mathivanan, 2006) N. Mathivanan. 2006. *Microprocessor, PC Hardware, and Interfacing*. New Delhi: Prentice-Hall of India, pp. 212-240.
- (Mustofa, 2016a) I. Mustofa, A. Goeritno, B.A. Prakosa, "Prototipe Sistem Kontrol Berbasis Mikrokontroler untuk Pengaman terhadap Gangguan Hubung Singkat pada Otobis," di *Prosiding SNTI V-2016 Universitas Trisakti*, Jakarta, 18 Mei 2016, hlm. 317-323.
- (Mustofa, 2016b) I. Mustofa, A. Goeritno, B.A. Prakosa. 2016. "Performansi Sistem Kontrol Berbasis Mikrokontroler ATmega32 Untuk Tampilan Kondisi Instalasi Listrik pada Otobis," di *Prosiding the 4th National Conference on Industrial Electrical and Electronics (NCIEE) Jurusan Teknik Elektro Fakultas Teknik Universitas Sultan Ageng Tirtayasa (UNTIRTA)* Cilegon, hal. 100-112.
- (Prakoso, 2016) B.A. Prakoso, A. Goeritno, B.A. Prakosa. 2016. "Prototipe Sistem Pengontrolan Berbasis Mikrokontroler ATmega32 untuk Analogi Smart Green House," di *Prosiding SNTI V-2016 Universitas Trisakti*, Jakarta, 18 Mei 2016, hlm. 338-345.
- (Saefurrochman, 2015) Saefurrochman, A. Goeritno, R. Yatim, D.J. Nugroho. (2015, Jan. 24). "Implementasi Sensor Suhu LM35 Berbantuan Mikrokontroler pada Perancangan Sistem Pengkondisian Suhu Ruangan," di *Prosiding University Research Colloquium 2015*, Universitas Muhammadiyah Surakarta, [Online]. 1 (1), hlm. 147-157. Tersedia di: <https://publikasiilmiah.ums.ac.id/bitstream/handle/11617/5095/17.pdf?sequence=1&isAllowed=y>.
- (Sopyandi, 2016) Sopyandi, A. Goeritno, R. Yatim. 2016. "Prototipe Sistem Pengontrolan Berbasis Payload Data Handling Berbantu Mikrokontroler untuk Instalasi Listrik Rumah Tinggal," di *Prosiding SNTI V-2016 Universitas Trisakti*, Jakarta, 18 Mei 2016, hlm. 331-337.
- (Syahida, 2017) A.I. Syahida, A. Goeritno, Syaiful. 2017. "Penggunaan *System Dynamics Software* untuk Simulasi Model Kebijakan Perburuan Rusa," di *Prosiding Seminar Nasional Multidisiplin Ilmu*, Universitas Budu Luhur, Jakarta, 22 April 2017, hlm. (ICT) 210-217.

(Thalib, 2001) M. Thalib. 2001. *20 Kerangka Pokok Pendidikan Islami*. Bandung, JB: Irsyad Baitus Salam, hlm. 30-34.