

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai pembuatan *remastering* sistem operasi (OS) linux sudah banyak dilakukan. Terdapat beberapa judul penelitian yang berhubungan dengan *remastering* sistem operasi (*Operating System*) linux. Penelitian ini juga dapat digunakan sebagai bahan perbandingan dan referensi dalam menulis.

Patra Nizar Salsabilla (2019) membuat distro “*Remastering Sistem Operasi Linux Ubuntu 17.10 Guna Menunjang Kegiatan Belajar Mengajar Matematika Dikelas 2 SD Muhammadiyah Sidoarum*” pada pembuatan dan perancangan sebuah sistem operasi Linux dengan tema yaitu belajar matematika, yang nantinya akan dipergunakan oleh pengajar atau guru matematika agar dapat memanfaatkan ilmu teknologi komputer .

Muhammad Yandi (2010) membuat distro “*Remastering Linux Ubuntu 9.10 Untuk Kegiatan Mata Kuliah Multimedia*”. Pada perancangan sebuah OS yang memiliki basis berupa open source akan memaketkan semua *software* yang dibutuhkan kedalam satu wadah dvd dimana isinya adalah *software* multimedia yang digunakan pada matakuliah praktikum multimedia di laboratorium komputer.

Muhammad Asef Yusriyadi (2010) melakukan penelitian mengenai “*Mandriva 2010 Remaster Power Packflash For Distro Linux Desktop*”. Pada penelitian ini dimana didalam distro ini terdapat berbagai *software* standar yang dapat dipergunakan sesuai kegunaannya. Sistem operasi ini menggunakan *remastersys*.

Bhamana Apta Reyhendendra (2020) merancang distro “*Remastering Sistem Operasi Linux Ubuntu 18.04 Guna Menunjang Kegiatan Belajar Mengajar Al – Qur’an Dikelas 4 SD Muhammadiyah Sapen Yogyakarta*” pada pembuatan dan perancangan sebuah sistem operasi Linux dengan tema yaitu belajar Al – Qur’an, yang nantinya akan dipergunakan oleh pengajar atau guru Agama agar dapat memanfaatkan ilmu teknologi komputer sebagai sarana dalam kegiatan belajar mengajar Agama khususnya dikelas 4 sekolah dasar.

2.2 Dasar Teori

2.2.1 Sistem Operasi

Sistem Operasi atau bisa disebut dengan OS (*Operating System*) merupakan sebuah perangkat lunak (*Software*) yang mempunyai fungsi sebagai sistem yang dapat mengatur serta manajemen suatu perangkat keras (*Hardware*) dan operasi dasar sistem yang ada didalamnya. Selain itu juga dapat mengoperasikan *software* yang mana sudah diinstall sebelumnya, seperti program pengolah data atau kata dan web browser.

Secara garis besar, sebuah *hardware* yaitu berupa komputer dan disaat itu dinyalakan serta dioperasikan maka keberadaan suatu sistem operasi yang ada didalamnya tersebut berada pada tingkat atau lapisan paling utama dan ditempatkan pada memori komputer nya. Dan *software* lainnya yang telah diinstall akan dapat dioperasikan setelah sistem operasi sudah berjalan terlebih dahulu.

Jika suatu sistem komputer terdiri dari berbagai lapisan dan tingkatan, maka sistem operasi memiliki tugas utama yaitu berupa menghubungkan sebuah perangkat keras dengan perangkat lunak nya. Selain memiliki tugas utama nya sistem operasi ini juga mempunyai tugas-tugas lain nya yaitu menjamin akan kelancaran beberapa *software-software* berbeda yang berjalan dan beroperasi secara bersamaan dengan baik dan lancar. Disaat *software-software* berbeda tersebut sedang beroperasi secara bersamaan maka secara otomatis sistem operasi melakukan tugas lain nya berupa mengatur dan manajemen waktu penggunaan memory, sehingga semua kegiatan dan proses nantinya tidak akan saling terganggu satu sama lain nya. Hal ini dilakukan oleh sebuah prosesor (CPU) yang telah dipasang.

Berikut ini adalah bagian-bagian dari sistem operasi yaitu :

1. Mekanisme *Booting* adalah memasukkan kernel ke memory
2. Kernel adalah sebuah program inti yang dimiliki oleh sistem operasi
3. *Command Interpreter* adalah program yang memiliki tugas untuk membaca perintah berupa input dari sang pengguna

4. Pustaka adalah program yang berisi fungsi-fungsi dasar dan standar
5. *Driver* adalah sebuah komponen perangkat lunak yang memiliki tugas sebagai penghubung antara sistem operasi dengan *hardware*

Sistem operasi juga memiliki beberapa jenis atau tipe yaitu jenis yang pertama bernama *desktop OS* merupakan sistem operasi yang memiliki ukuran kapasitas besar serta kompleks sedangkan sistem operasi yang memiliki ukuran kapasitas kecil disebut dengan *real-time OS*. Contoh sistem operasinya adalah seperti windows, symbian, linux, solaris dan lain-lainnya.

2.2.2 Open Source Software

Eric Raymond merupakan orang yang pertama kali memperkenalkan dan mempopulerkan sistem *Open Source Software*. Arti dari kata *Open Source* didalam sistem ini adalah kode programnya dapat diakses oleh setiap orang, jadi setiap orang dapat melihat kelebihan dan kekurangan dari sistem tersebut serta jikalau mampu, akan dapat dikembangkan dan diperbaiki tanpa adanya pembayaran.

Linux merupakan salah satu perangkat lunak yang dikembangkan melalui sebuah sistem yang berbasis *open source*. *Software* ini juga bisa disebut dengan *freeSoftware*. Artinya setiap orang dapat mengakses, memiliki, mengoperasikannya dan dapat dikembangkan tanpa adanya pembayaran apapun. Asalkan tidak menghilangkan atau masih tetap menyertakan kode asal program *software* tersebut.

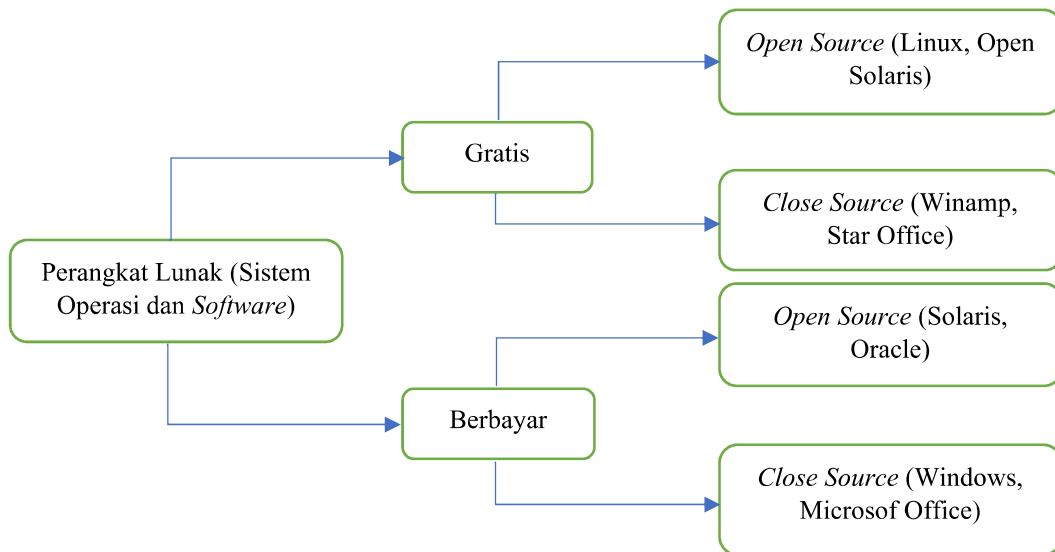
Free software adalah salah satu implementasi dari konsep *open source*. Banyak orang yang salah mengartikan bahwa *free software* itu adalah *software* yang gratisan. Tetapi menurut *Free Software Foundation*, arti dari *free* disini memiliki 4 kebebasan, yaitu :

1. Kebebasan dalam mengoperasikan *software* guna keperluan apapun.
2. Kebebasan dalam menganalisis program akan berjalan, dan mengadaptasinya untuk keperluan sendiri.
3. Kebebasan untuk mendistribusikan kembali kepada orang lain.
4. Kebebasan untuk memperbaiki program, dan menerbitkan perbaikan tersebut secara luas.

Jadi, arti konsep *free* disini bukanlah berarti gratis melainkan bebas. Ada banyak vendor yang mengkomersilkan *free software* mereka, tetapi banyak juga yang benar-benar menggratiskannya.

2.2.3 Free Open Source Software

Menurut David Wheeler (2000), secara umum program yang dinamakan *Free Open Source Software* (FOSS) merupakan suatu program sistem yang memiliki lisensi dan lisensinya itu memberikan sang pengguna atau *user* dapat bebas dalam mengoperasikan, menganalisis, dan dapat mengembangkannya lebih baik lagi tanpa adanya pembayaran *royalty*. Serta dapat bertolak belakang dengan *Close Source* yang menutup kode sumber untuk diketahui masyarakat dan harus membayar *royalty* kepada penggunanya. Berikut gambar perbandingan sistem operasi dan *software Open Source* dan *Close Source* :



Gambar 2.1 Perbandingan *Open Source* dan *Close Source*

Pengertian dari *open source software* itu sendiri harus mempunyai standard antara lain :

1. Pembagian dari sebuah program sistem perangkat lunak yang secara bebas, artinya didapatkan secara gratis atau tidak bayar.
2. Kode sumber atau *source code* dari sebuah program *software* harus dapat dicantumkan ditempat yang nantinya dapat digunakan oleh setiap

orang, misalnya dalam menggunakan jaringan internet semua orang dapat *download software* gratis.

3. Dalam memelihara sebuah karakter dari suatu *source code* yang dimiliki oleh si pembuat *software* maka *licence* pada *software* dapat melarang pendistribusian *source code* yang telah di rubah, kecuali *licence* itu memberi izin pendistribusian *patch file* (sebagaimana file *software*) yang bertujuan merubah *software* tersebut dengan disertakan *source code* dari *software* asal.
4. Sebuah *licence* yang berbasis *open source* dilarang untuk membuat suatu diskriminasi baik itu kelompok maupun individu.
5. Dilarang menghalangi pengguna untuk memanfaatkan fasilitas *open source* dalam berbagai bidang tertentu. Sebagai contoh, tidak ada pembatasan program tersebut terhadap penggunaan dalam bidang bisnis, atau terhadap pemanfaatan dalam bidang riset genetic.
6. Sebuah lisensi dilarang memproduksi suatu program yang mana program *software* tersebut hanya memiliki kepentingan dan keuntungan untuk sendiri. Hak-hak yang tercantum pada suatu program tidak boleh tergantung pada apakah program tersebut merupakan bagian dari satu distribusi perangkat lunak tertentu atau tidak. Sekalipun program diambil dari distribusi tersebut dan digunakan atau didistribusikan selaras dengan lisensi program itu.

Dengan demikian, ada satu hal yang perlu kita garis bawahi yaitu mengenai pengertian dari *free* ini, dimana arti *free* dapat dideskripsikan melalui lima kegiatan antara lain :

1. Kebebasan dalam mengoperasikan sebuah program.
2. Kebebasan dalam mengetahui suatu *source code* program *software*.
3. Kebebasan dalam mendistribusikan program *software*.
4. Kebebasan dalam mengembangkan program *software*.
5. Kebebasan dalam menjual program *software*.

2.2.4 Sejarah Linux

Profesor Andrew Tanenbaum adalah orang yang pertama kali membuat sebuah sistem operasi dengan nama Minix. Bermula dari sistem operasi Minix inilah sistem operasi Linux terbentuk yaitu gabungan nama antara nama si pencipta dengan nama sistem operasinya UNIX.

Sedangkan Linus B. Torvald merupakan orang yang pertama kali mengoperasikan sistem operasi Minix, ia seorang mahasiswa Universitas Helsinki. Selama dia mengoperasikan sistem operasi Minix ini lalu ia mendapatkan beberapa pandangan mengenai kelebihan dan kekurangan sistem operasi Minix tersebut. Dan ditahun 1991 Linus B. Torvald merancang dan membuat sebuah sistem operasi yang baru dengan nama Linux yang mana turunan dari UNIX. Pada saat itu sistem nya belum stabil, lalu pada Oktober 1991 Linus B. Torvald meminta bantuan kepada programmer lain untuk dapat mengembangkan sistem operasi Linux ini menjadi lebih stabil dan baik lagi. Dan akhirnya ditahun 1994 Linus B. Torvald meluncurkan sistem operasi Linux versi 1.0, dimana di versi 1.0 ini sistem operasinya sudah dilengkapi dengan *feature symmetric multiprocessing* yaitu kemampuan dalam mengatur dan membagi tugas dari banyaknya CPU serta *feature preemptive multiprocessing* yaitu kemampuan dalam mengatur dan membagi sumber daya CPU saat banyaknya penggunaan *software*.

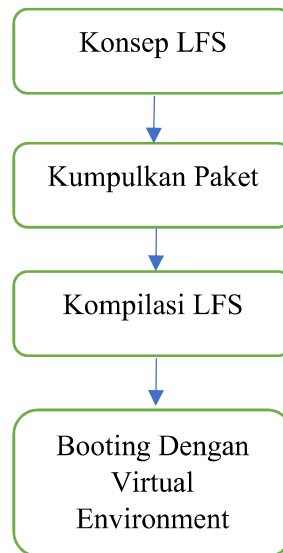
2.2.5 Konsep Pembuatan Distribusi Linux

Dalam pembuatan sebuah distro perangkat lunak harus memiliki tujuan berupa memberikan manfaat yang banyak serta distro tersebut dapat dikembangkan lebih baik lagi. Adapun metode – metode pembuatan distro yaitu :

2.2.5.1 LFS (Linux From Scratch)

Linux From Scratch merupakan salah satu metode pembuatan sebuah distro perangkat lunak secara individu atau mandiri. Arti dari mandiri disini adalah *Software* yang telah di instal *source code* nya didapat melalui *pristine code* atau kode sumber yang murni. Adapaun kekurangan dari metode ini adalah distro yang sudah dibuat sifatnya tidak bisa dikembangkan lagi biasanya digunakan untuk diri

sendiri atau pribadi, distro yang nantinya akan diproduksi tidak dapat dikerjakan dalam satu ISO yang *installable* dan sangat sulit untuk dapat didistribusikan lagi.



Gambar 2.2 Bagan Sederhana Pembuatan Distro Dengan Konsep LFS

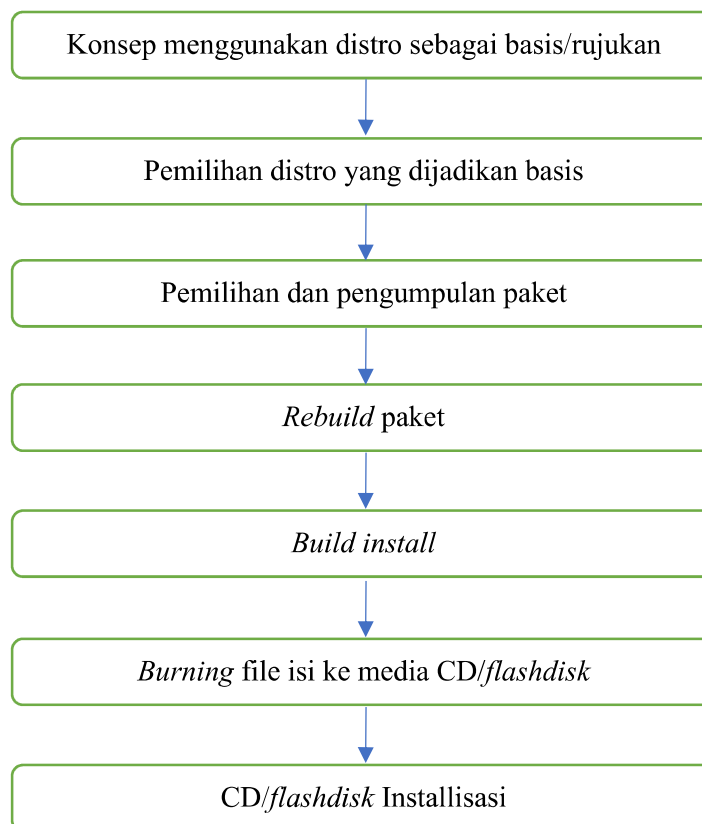
Berikut ini adalah proses - proses pembuatan distro LFS secara singkat :

1. Rancanglah semua paket-paket yang perlukan.
2. Rancanglah direktori secara statik dengan cara mengkompilasi paket-paket sistem LFS yang telah dirancang sebelumnya sebagai tempat pembuatan file LFS.
3. Selanjutnya adalah merancang sistem-sistem LFS yang telah dikompilasi dan terinstall agar dapat melakukan sebuah *booting*, dengan cara menggunakan sebuah sistem yang bernama virtual environment, maka nanti LFS akan membangun sebuah sistem yang nantinya dapat membantu proses kustomisasi dan kecepatan loading distro tersebut karena telah disesuaikan dengan platform nya dimana distro akan ditempatkan. Maka proses ini akan berjalan dengan memindahkan semua *library* yang terdapat dalam satu file yang diposisikan pada setiap file dengan fungsinya masing-masing. Sehingga nanti setiap paket akan dapat berdiri sendiri tanpa tergantung dengan *library* yang ada, akan tetapi apabila dilakukan suatu proses update maka file yang melakukan perubahan adalah keseluruhan sistem nya. Karena proses kompilasi ini sehingga paket yang ada dapat berdiri sendiri

adalah salah satu bagian dari proses pembangunan sistem LFS. Perubahan tersebut nantinya akan menambah waktu pembuatan yang pada dasarnya dapat dimanfaatkan untuk mengoptimalkan paket *software* yang ada.

2.2.5.2 Remaster

Metode yang kedua ini merupakan metode pembuatan distro yang banyak orang atau pengembang gunakan, sekitar 80% distro linux yang sudah beredar dan diproduksi yang dibuat menggunakan konsep metode *remaster* ini. Dikarenakan kepopuleran metode ini banyak pengembang sudah menjadikan metode ini sebagai rujukan atau basis dalam membuat sebuah distro perangkat lunak. Kelebihan dari metode ini salah satunya adalah distronya dapat dikembangkan lagi oleh pengembang lain nya untuk jadi lebih baik lagi.



Gambar 2.3 Bagan Sederhana Pembuatan Distro Dengan Konsep *Remaster*

2.2.6 Bagian Sistem Linux

Sistem linux terdiri dari tiga bagian kode penting yaitu:

1. Kernel

Bertanggung jawab untuk mengatur dan menjaga seluruh kerangka utama sebuah inti sari dari suatu sistem operasi, seperti memori virtual dan proses – proses keseluruhan sistem.

2. System Libraries

Memilih beberapa berkas fungsi standar agar *software-software* nya dapat terhubung dengan kernel dan mengimplementasikan nya agar semua fungsi sistem operasi dapat beroperasi dengan baik dan maksimal. *System Libraries* juga mengoleksi serta menjaga suatu fungsi *system call* jikalau nanti dibutuhkan guna mengatur fungsi-fungsi tersebut.

3. System Utilities

Merupakan suatu rancangan sistem yang melakukan pekerjaan manajemen secara individu dan khusus.

2.2.7 Linux Ubuntu 18.04

Ubuntu merupakan sebuah distribusi atau turunan dari Linux yang memiliki basis terhadap debian dan mempunyai interface desktop. Nama ubuntu berasal dari sebuah konsep ideologi benua Afrika lebih tepatnya di Afrika Selatan, jadi kata ubuntu diambil dari sebuah bahasa kuno Afrika, yang memiliki arti rasa berperikemanusiaan kepada manusia lain nya.

Linux ubuntu telah meluncurkan beberapa versi nya salah satu yang terbarunya adalah linux ubuntu versi 18.04 yang mana pada versi ini menggunakan sistem Gnome, dengan tampilan animasi windowsnya lebih halus dan lebih bagus serta yang lebih pentingnya adalah versi ini sistem nya lebih stabil dibandingkan dengan versi terdahulunya. Adapun kelebihan lain dari versi 18.04 salah satunya adalah dapat secara langsung menginstall program atau *software* lain nya secara mudah dikarenakan di versi ini sudah disediakan Apps Store yang bernama Ubuntu *Software* jadi pengguna hanya mencari *Software* yang diinginkan lalu klik install kemudian tunggu dan selesai.

2.2.8 Zekr

Zekr adalah sebuah peranti lunak al-Quran open source yang dibangun menggunakan bahasa pemrograman Java. Zekr dapat berfungsi sebagai alat bantu untuk menelusuri dan melakukan pencarian terhadap teks al Quran. Zekr dibuat dalam bentuk aplikasi yang bersifat "universal", "open source", dan dapat dijalankan pada berbagai platform.

Zekr memiliki kemampuan untuk mengakomodasi beberapa jenis fasilitas tambahan (add-on) antara lain: terjemahan dalam berbagai bahasa, themes, serta bacaan Quran dari berbagai qari.

Sejak versi 0.7.0, Zekr memungkinkan pencarian ke dalam setiap terjemahan baik dengan metode indeks (berbasis Lucene) maupun dengan metode biasa.

Zekr tersedia dalam bentuk installer NSIS untuk platform Microsoft Windows dan Application Bundle untuk Mac OS. Untuk Linux, disediakan arsip dalam format tar.gz, dan khusus untuk debian telah tersedia installer dalam bentuk .deb.

2.2.9 Elforkane

Elforkane adalah aplikasi Al Qur'an. Semacam Zekr dan Ayat yang sudah tersedia sebelumnya. Elforkane dibangun diatas Qt (Bahasa Pemrograman) dan bisa berjalan native diatas Linux dengan baik. Kelebihan Elforkane dibanding kedua aplikasi sebelumnya adalah dalam hal tampilan UI yang lebih baik.

2.2.10 QiOO (*Extension Libre Office Writer*)

QiOO adalah sebuah ekstensi untuk OpenOffice yang berfungsi untuk menampilkan ayat-ayat Al-Qur'an beserta terjemahannya dalam bentuk teks di dokumen OpenOffice atau LibreOffice.

Quran in Open Office (QiOO) yang merupakan ekstensi dari Open Office atau dapat juga dipakai di LibreOffice lingkungan Linux maupun Windows atau Mac OS.

2.2.11 Komposisi Warna

Dalam merancang sebuah *software* sistem operasi, salah satu aspek yang perlu diperhatikan adalah desain antarmuka *software* tersebut. Untuk dapat membuat sebuah desain yang nyaman dilihat oleh *user*, maka hal yang perlu dilakukan adalah memilih warna yang terbaik. Warna memberikan peran penting untuk kenyamanan interaksi *user* dengan antar muka *software*. Berikut ini merupakan tabel komposisi kombinasi warna pada desain antarmuka, yang terdiri dari komposisi warna terbaik hingga komposisi warna terburuk dalam hubungannya dengan aspek psikologis *user*.

Tabel 2.1 Komposisi Kombinasi Warna Terbaik

Background	Garis dan Teks (Normal)	Garis dan Teks (Tebal)
Putih	Biru (94%), Hitam (63%), Merah (25%)	Hitam (69%), Biru (63%), Merah (31%)
Hitam	Putih (75%), Kuning (63%)	Kuning (69%), Putih (59%), Hijau (25%)
Merah	Kuning (75%), Putih (56%), Hitam (44%)	Hitam (50%), Kuning (44%), Putih (44%), Cyan (31%)
Hijau	Hitam (100%), Biru (56%), Merah (25%)	Hitam (69%), Merah (63%), Biru (31%)
Biru	Putih (81%), Kuning (50%), Cyan (25%)	Kuning (38%), Magenta (31%), Hitam (31%), Cyan (31%), Putih (25%)
Cyan	Biru (69%), Hitam (56%), Merah (37%)	Merah (56%), Biru (50%), Hitam (44%), Magenta (25%)
Magenta	Hitam (63%), Putih (56%), Biru (44%)	Biru (50%), Hitam (44%), Kuning (25%)
Kuning	Merah (63%), Biru (63%), Hitam (56%)	Merah (75%), Biru (63%), Hitam (50%)

(Sumber : <https://www.beritanet.com/2008/07/25/komposisi-warna-terbaik-terburuk/>)

Tabel 2.2 Komposisi Kombinasi Warna Terburuk

Background	Garis dan Teks (Normal)	Garis dan Teks (Tebal)
Putih	Kuning (100%), Cyan (94%)	Kuning (94%), Cyan (75%)
Hitam	Biru (87%), Merah (44%), Magenta (25%)	Biru (81%), Magenta (31%)
Merah	Magenta (81%), Biru (44%), Hijau & Cyan (25%)	Magenta (69%), Biru (50%), Hijau (37%), Cyan (25%)
Hijau	Cyan (81%), Magenta (50%), Kuning (37%)	Cyan (81%), Magenta & Kuning (44%)
Biru	Hijau (62%), Merah & Hitam (37%)	Hijau (44%), Merah & Hitam (31%)
Cyan	Hitam (81%), Kuning (75%), Putih (31%)	Kuning (69%), Hijau (62%), Putih (56%)
Magenta	Hijau (75%), Merah (56%), Cyan (44%)	Cyan (81%), Hijau (69%), Merah (44%)
Kuning	Putih & Cyan (81%)	Putih (81%), Cyan (56%), Hijau (25%)

(Sumber:<https://www.beritanet.com/2008/07/25/komposisi-warna-terbaik-terburuk/>)

Pada tabel dihalaman sebelumnya dapat dilihat bahwa hal yang harus diperhatikan dalam melakukan pemilihan warna saat mendesain perangkat lunak antara lain sebagai berikut :

- a. Menghindari perpaduan warna yang membuat mata menjadi lelah, seperti warna cyan, magenta, kuning secara bersama.
- b. Pengguna dengan usia tua, sebaiknya menggunakan warna yang tajam.
- c. Dalam hal ini kombinasi yang terbaik adalah warna biru dengan kuning.
- d. Perpaduan warna hijau dan biru secara psikologis menimbulkan kesan yang tidak baik. Warna berlawanan yang dapat dipadukan adalah warna biru–kuning dan hijau-merah.

2.2.12 Teori Kognitif

Teori kognitif merupakan teori bahwa belajar adalah suatu proses perubahan pemikiran dan pemahaman yang dapat diukur dan dilihat. Teori ini lebih berorientasi pada kasus bagaimana siswa belajar berpikir. Fokus studinya adalah pertanyaan perkembangan kognitif. Pada guru yang terpenting adalah bagaimana dapat berpengaruh dalam perkembangan berpikir dan bagaimana guru dapat mengkondisikan pengajaran pada *level* perkembangan kognitif para siswa.

Jean Piaget adalah salah satu pakar yang pintar telah *publish* tentang studi perkembangan kognitif. Pemikiran Piaget ini lebih berkonsentrasi pada pendidikan anak tetapi program pendidikan orang dewasa pun banyak yang memakai konsep Piaget ini. Piaget mengakui bahwa pada dasarnya setiap manusia mengalami perkembangan pada tingkat berpikirnya melalui tahapan yang rumit. Setiap tahapan ditunjukkan dengan pemilihan konsep sebagai skema. Skema itu merupakan suatu program atau strategi yang di gunakan oleh manusia pada saat berinteraksi dalam lingkungannya.

Hasil interaksi tersebut adalah pengalaman. Pengalaman itu disatukan ke dalam pola perilaku. Apabila pengalamannya tidak memadai untuk menjelaskan hal-hal baru, nantinya manusia akan mengembangkan skema baru dan menyelesaikan dengan informasi baru. Tanpa strategi yang terkait dengan lingkungan, maka aspek tertentu seolah-olah tidak ada untuk kita. Bagi seorang anak, untuk mampu menyusun benda secara berurutan di perlukan waktu yang cukup lama. Cakupan kognitif anak tumbuh melalui perkembangan skema yang lebih kompleks untuk di satukan dengan lingkungannya. Sumber penyebab ini semua adalah penyesuaian.

Sebagai ahli psikologi perkembangan kognitif, Piaget telah mengetahui bahwa perkembangan skema terjadi dengan urutan yang sama dengan kecepatan yang di ukur oleh kematangan fisiologis kita. Atas dasar itu, maka Piaget membenarkan perkembangan kognitif sebagai berikut.

1. Tahapan sensorimotor (0-2 tahun). Dimana anak berpikir dengan merujuk pada perilaku yang bersifat praverbal. Pada tahapan ini anak peduli pada benda sebagai benda.

2. Tahapan praoperasional (2-7 tahun). Tahapan ini di bagi dua bagian yaitu:

- Tahapan prakonseptual (2-4 tahun). Artinya awal dari intelegensi konseptual dimana anak sudah dapat mengembangkan fungsi simbol.
- Tahapan intuitif (4-7). Artinya anak belum dapat berpikir secara logis.

3. Tahapan operasional (7-16 tahun). Tahapan ini dibagi menjadi dua bagian yaitu:

- Tahapan konkrit operasional (7-11 tahun). Artinya pada tahap ini anak sudah mulai dapat berpikir secara rasional yang diawali dengan mempersepsi benda-benda konkrit
- Tahapan formal operasional (11-16 tahun). Artinya anak usia sekolah lanjutan ini mulai memunculkan kemampuan untuk memecahkan masalah secara logis. Tahap berpikir ini di anggap reflektif karena anak bernalar dengan dasar asumsi formal.