

MODUL

Konsep Pemodelan Sistem Perangkat Lunak

A. Tujuan Pembelajaran

1. Menjelaskan konsep rekayasa perangkat lunak
2. Menjelaskan komponen dan karakteristik perangkat lunak
3. Menjelaskan prinsip analisis dan desain
4. Menjelaskan ragam pemodelan perangkat lunak

B. Materi Pembelajaran

1. Definisi Perangkat Lunak

Perangkat lunak (*software*) adalah kumpulan instruksi atau program komputer (*syntaks/code* yang ditulis dalam bahasa pemrograman) ketika dijalankan menyediakan fungsi dan performan yang diinginkan. Perangkat lunak merupakan struktur data yang memungkinkan program memanipulasi informasi dan dokumen yang menggambarkan cara menggunakan dan mengoperasikan program tersebut (*Pressman*). Perangkat lunak merupakan program komputer yang berfungsi menghubungkan antara pengguna dan komputer yang digunakan. Beberapa teori menyatakan bahwa perangkat lunak merupakan sebagai media penerjemah perintah yang diberikan oleh pengguna kepada komputer untuk selanjutnya diproses melalui perangkat keras komputer tersebut.

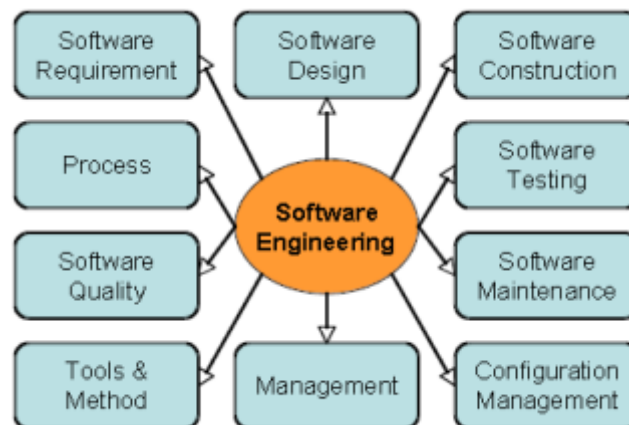
Menurut IEEE & ISO 1997, perangkat lunak adalah kumpulan program komputer (*syntaks/code*), prosedur, fungsi, modul (kumpulan program yang memiliki tujuan dan fungsi tertentu), dokumentasi dan kumpulan data yang diperlukan untuk mengoperasikan suatu sistem perangkat lunak. Perangkat lunak umumnya digunakan untuk mengontrol perangkat keras yang biasa disebut sebagai *device driver*, melakukan proses perhitungan, berinteraksi dengan perangkat lunak yang lebih mendasar lainnya, seperti sistem operasi dan bahasa pemrograman. Jadi, rekayasa perangkat lunak (*software engineering*) merupakan aplikasi perangkat lunak yang bersifat sistematis, dalam pengembangannya menerapkan prinsip-prinsip rekayasa, metode pengembangan, operasi, dan *maintenance* dari *software* yang menggunakan pendekatan rekayasa.

Rekayasa atau teknik merupakan penerapan ilmu dan teknologi untuk menyelesaikan permasalahan manusia. Hal ini diselesaikan lewat pengetahuan matematika, dan pengalaman praktis yang diterapkan untuk mendesain objek atau proses yang berguna. Para praktisi teknik atau rekayasa profesional disebut perekayasa. Rekayasa perangkat lunak atau *software engineering* dalam bahasa Inggris merupakan bidang ilmu yang mempelajari tentang segala aspek perangkat lunak, seperti cara-cara pengembangan, pemeliharaan, pembuatan, serta manajemen kualitas perangkat lunak. Rekayasa perangkat lunak juga merupakan disiplin rekayasa dengan perangkat lunak yang dikembangkan. Biasanya proses melibatkan penemuan pada keinginan *client* menyusunnya di dalam daftar kebutuhan, merancang arsitektur yang mampu mendukung semua kebutuhan,

perancang, pengodean, pengujian, dan pengintegrasian bagian yang terpisah, menguji keseluruhan, penyebaran, dan pemeliharaan perangkat lunak.

Secara umum ada tiga jenis perangkat lunak yang diketahui hingga saat ini yaitu sistem operasi yang merupakan sebuah penghubung antara pengguna dari komputer dengan perangkat keras komputer. Kedua yaitu perangkat lunak bahasa pemrograman seperti java, C++, pascal, PHP, dll. Dan yang ketiga yaitu perangkat lunak aplikasi yang merupakan perangkat yang digunakan untuk membantu dan memudahkan pekerjaan seseorang misalnya saja Microsoft Excel, Word, dan Power Point.

Ruang lingkup Rekayasa Perangkat Lunak (RPL) dapat digambarkan sebagai berikut:



- *Software Requirement* merupakan kegiatan yang dilakukan untuk mengidentifikasi dan menganalisis kebutuhan perangkat lunak. Hasil akhir tahapan ini adalah spesifikasi dan model perangkat lunak.
- *Software Design* adalah tahapan perancangan arsitektur, komponen, antarmuka, dan karakteristik lain dari perangkat lunak.
- *Software Construction* berhubungan dengan *detail* pengembangan perangkat lunak, termasuk algoritma, pengkodean, pengujian dan pencarian kesalahan.
- *Software Testing* meliputi pengujian pada keseluruhan perilaku perangkat lunak.
- *Software Maintenance* mencakup upaya-upaya perawatan ketika perangkat lunak telah dioperasikan.
- *Software Configuration* management berhubungan dengan usaha perubahan konfigurasi perangkat lunak untuk memenuhi kebutuhan tertentu.
- *Software Engineering Management* berkaitan dengan pengelolaan dan pengukuran RPL, termasuk perencanaan proyek perangkat lunak.
- *Software Engineering Tools and Methods* mencakup kajian teoritis tentang alat bantu dan metode RPL.
- *Software Engineering Process* berhubungan dengan definisi, implementasi pengukuran, pengelolaan, perubahan, dan perbaikan proses RPL.
- *Software Quality* menitik beratkan kualitas dan daur hidup perangkat lunak.

2. Komponen dan Karakteristik Perangkat Lunak

a. Komponen Perangkat Lunak

1) Sistem Operasi

Sistem operasi merupakan komponen utama perangkat lunak sistem. Sistem operasi (disebut juga *platform software*) terdiri dari program utama dan program *low-level* yang mengatur operasi dasar komputer. Memungkinkan perangkat lunak aplikasi untuk berinteraksi dengan komputer. Memungkinkan perangkat lunak aplikasi untuk berinteraksi dengan komputer dan membantu komputer untuk mengelola sumber daya baik itu internal maupun eksternal. Secara khusus, sistem operasi menangani *control* dan penggunaan sumber daya perangkat keras, termasuk ruang *disk*, memori, alokasi CPU *time*, dan perangkat peripheral.

2) *Device Driver*

Membantu komputer mengontrol perangkat peripheral. Driver artinya adalah pemacu yang maksudnya adalah dengan dipasangnya suatu device ke komputer sementara operating sistem kita atau komputer tidak mengenalinya maka driver tadi yang akan memperkenalkan bahwa device yang dipasang itu adalah benar adanya dan bisa digunakan karena Device Driver adalah program komputer yang mengawal jenis-jenis peranti yang dipasangkan (install) pada komputer. Program ini adalah spesifik untuk peranti yang tertentu saja dan tidak boleh digunakan pada peranti yang lain, contoh: mesin pencetak (printer) memerlukan driver untuk berfungsi.

3) Program Utilitas

Adalah sebuah program yang digunakan untuk Meningkatkan kapabilitas program komputer yang telah ada pada computer. Perangkat lunak utilitas merupakan perangkat lunak komputer yang didisain untuk membantu proses analisis, konfigurasi, optimasi, dan membantu pengelolaan sebuah komputer ataupun sistem. Perangkat lunak utilitas harus dibedakan dengan perangkat lunak aplikasi yang memungkinkan pengguna melakukan berbagai hal dengan komputer seperti mengetik, melakukan permainan, merancang gambar, dan lain-lain. Perangkat lunak utilitas lebih memfokuskan penggunaannya pada pengoptimasian fungsi dari infrastruktur yang terdapat dalam sebuah komputer. Karena fungsinya, perangkat lunak utilitas umumnya tidak ditujukan untuk pengguna secara umum, melainkan ditujukan untuk pengguna yang memiliki pemahaman atas cara kerja sistem komputer yang cukup baik.

Kebanyakan perangkat keras utilitas ini dibuat secara khusus untuk melakukan fungsi tertentu pada suatu area komputasi secara spesifik, seperti memformat harddisk, atau melakukan pengecekan konektifitas jaringan. Namun dalam perkembangannya sejumlah perangkat lunak utilitas terkadang pula dipaketkan dalam satu paket utilitas yang ditujukan untuk beragam kebutuhan.

b. Karakteristik Perangkat Lunak

Karakteristik Proses Perangkat Lunak terdiri dari:

- 1) *Understandability*, membuat proses secara eksplisit didefinisikan dan bagaimana sehingga mudah untuk mengerti definisi proses

- 2) *Visibility*, Aktifitas proses menghasilkan hasil yang jelas sehingga tahapan proses yang dilakukan terlihat
- 3) *Supportability*, Aktifitas Proses dapat didukung atas CASE tools
- 4) *Acceptability*, Penerimaan atas proses yang terdefinisi dan yang digunakan oleh Engineer selama pembangunan Produk Perangkat Lunak.
- 5) *Reliability*, Proses didesain dalam suatu metode untuk dihindarkan dari kesalahan
- 6) *Robustness*, Proses dapat meneruskan dalam masalah yang tidak diharapkan terjadi
- 7) *Maintainability*, Proses yang merefleksikan atas perubahan terhadap permintaan atau perbaikan proses yang diidentifikasi
- 8) *Rapidity*, bagaimana cepat dapat berjalan atas proses pengiriman atau implementasi sebuah sistem dari Spesifikasi yang ada sampai selesai

Dalam Buku Software Engineering Ian Sommerville, Perangkat Lunak mempunyai Karakteristik sebagai berikut:

- 1) *Maintainability* (Dapat Dirawat), Perangkat Lunak harus dapat memenuhi perubahan kebutuhan
- 2) *Dependability*, Perangkat Lunak harus dapat dipercaya
- 3) *Efisiensi*, Perangkat Lunak harus efisien dalam penggunaan resource
- 4) *Usability*, Perangkat Lunak harus dapat digunakan sesuai dengan yang direncanakan

3. Prinsip Analisis dan Desain

Masing-masing metode analisis memiliki titik pandang yang unik. Tetapi semua metode analisis dihubungkan oleh serangkaian prinsip operasional:

- a) Domain informasi dari suatu masalah harus direpresentasikan dan dipahami.
- b) Fungsi-fungsi yang akan dilakukan oleh perangkat lunak harus didefinisikan.
- c) Tingkah laku perangkat lunak (sebagai suatu urutan kejadian eksternal) harus diwakilkan.
- d) Model-model yang menggambarkan informasi, fungsi, dan tingkah laku harus dipecah-pecah dalam suatu cara yang membongkar suatu detail dalam bentuk lapisan.
- e) Proses analisis harus bergerak dari informasi dasar ke detail implementasi.

Dengan mengaplikasikan prinsip-prinsip tersebut, analisis mendekati suatu masalah secara sistematis. Domain informasi diuji sehingga fungsi itu dapat dipahami secara lebih lengkap. Model-model digunakan sehingga karakteristik fungsi dan tingkah laku dapat dikomunikasikan dengan cara yang rapi. Pembagian diterapkan untuk mengurangi keruwetan. Pandangan esensial dan implementasi dari perangkat lunak diperlukan untuk mengakomodasi batasan logis yang dibebankan oleh persyaratan pemrosesan dan batasan fisik yang dibebankan oleh elemen sistem yang lain. Perekayasa perangkat lunak yang mempercayai

prinsip tersebut akan dapat lebih mengembangkan spesifikasi perangkat lunak yang kemudian akan menjadi dasar yang kuat bagi desain.

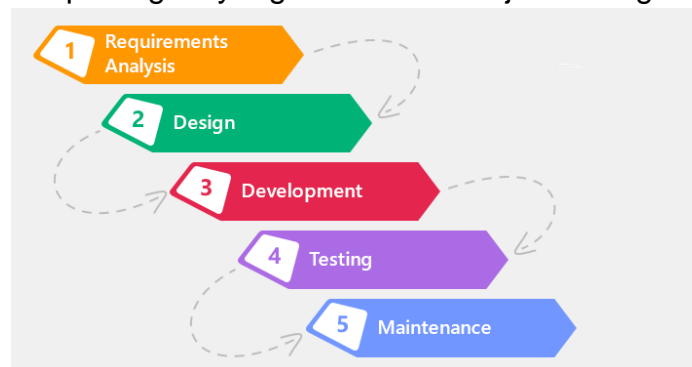
4. Ragam Pemodelan Perangkat Lunak

Model proses perangkat lunak merupakan cara untuk memproses sebuah perangkat lunak dari nol menjadi sebuah perangkat lunak yang siap untuk digunakan. Berikut merupakan beberapa contoh model proses perangkat lunak yang biasa digunakan:

a) Model *Waterfall*

Model air terjun ini merupakan model klasik yang bersifat sistematis dalam membuat suatu perangkat lunak dan juga paling sering digunakan.

Pada fase analisis fungsi, kemampuan operasi perangkat lunak yang akan dibuat harus diketahui terlebih dahulu. Kemudian apabila analisis telah selesai dilakukan maka didesainlah perangkat lunak yang akan dibuat. Setelah desain selesai lalu desain tersebut diterjemahkan kedalam kode-kode dengan bahasa pemrograman yang diinginkan, misalnya saja C++. Setelah kode selesai dibuat diadakanlah proses pengetesan terhadap perangkat lunak yang baru dibuat agar diketahui apakah perangkat yang dibuat bisa berjalan dengan benar atau tidak.



Gambar 1. Model *Waterfall*

Tahapan-tahapan pada model air terjun, yaitu:

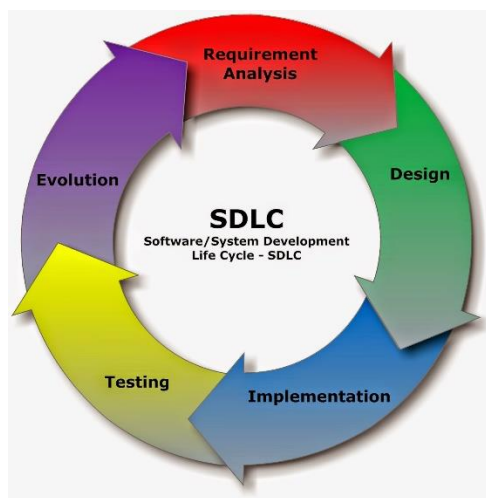
- 1) Rekayasa sistem perangkat lunak (*System Engineering*) merupakan bagian dari sebuah sistem untuk mendapatkan gambaran yang meluas pada arah sistem.
- 2) Analisis (*Analysis*) merupakan tahapan sistem perangkat lunak yang fokusnya lebih terarah ke perangkat lunak dan berusaha mengetahui aspek 'what' dengan melibatkan pemakaian dan pengembangan.
- 3) Perancangan (*Design*) merupakan tahapan sistem perangkat lunak yang membentuk representasi dan dapat dievaluasi kualitasnya sebelum tahap *coding* dilakukan.
- 4) Pengembangan program (*Coding*) merupakan rancangan yang mengarah ke dalam bentuk yang dapat dimengerti komputer.
- 5) Pengujian (*Testing*) merupakan tahapan sistem perangkat lunak yang berfokus pada rincian logikal dari perangkat lunak, bertujuan mengungkapkan

dan menghilangkan kesalahan-kesalahan yang ada sehingga perangkat lunak bekerja sesuai dengan yang diharapkan.

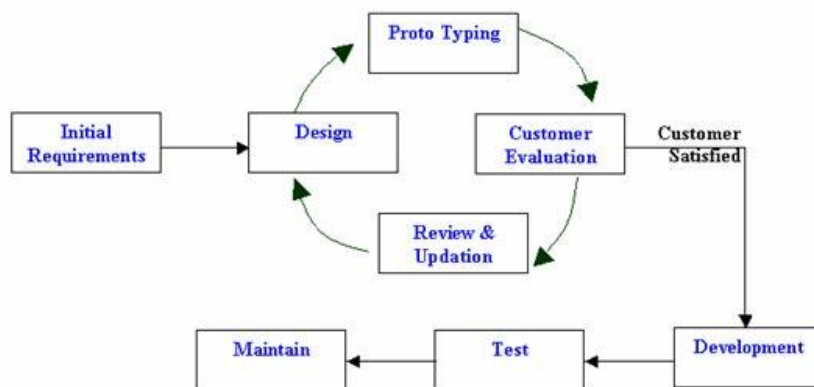
- 6) Pemeliharaan (*Maintenance*) merupakan kegiatan-kegiatan koreksi kesalahan dan penyesuaian perangkat lunak terhadap perubahan lingkungannya.

b) Model *Prototype*

Metode *prototype* merupakan suatu paradigma baru dalam metode pengembangan perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC (*Software Development Life Cycle*) atau waterfall development model.



Gambar 2. Model SDLC



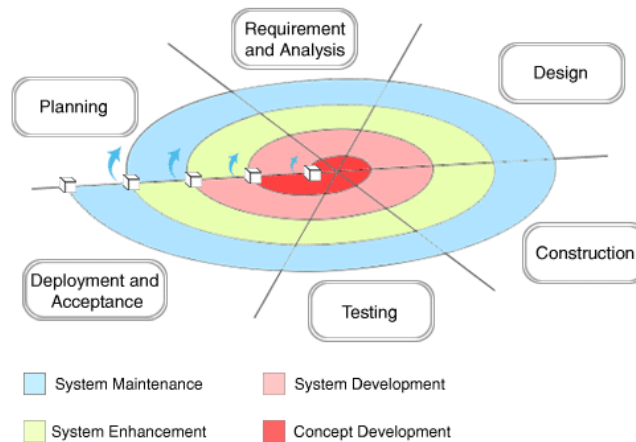
Proto Type Model

Gambar 3. Model *Prototype*

c) Model Spiral

Model ini mengadaptasi dua model perangkat lunak yang ada yaitu model prototyping dengan pengulangannya dan model waterfall dengan pengendalian

dan sistematikanya. Model ini dikenal dengan sebutan Spiral Boehm. Pengembang dalam model ini memadupadankan beberapa model umum tersebut untuk menghasilkan produk khusus atau untuk menjawab persoalan-persoalan tertentu selama proses pengerjaan proyek.



Gambar 4. Model *Spiral*

d) Model RAD

Rapid Application Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

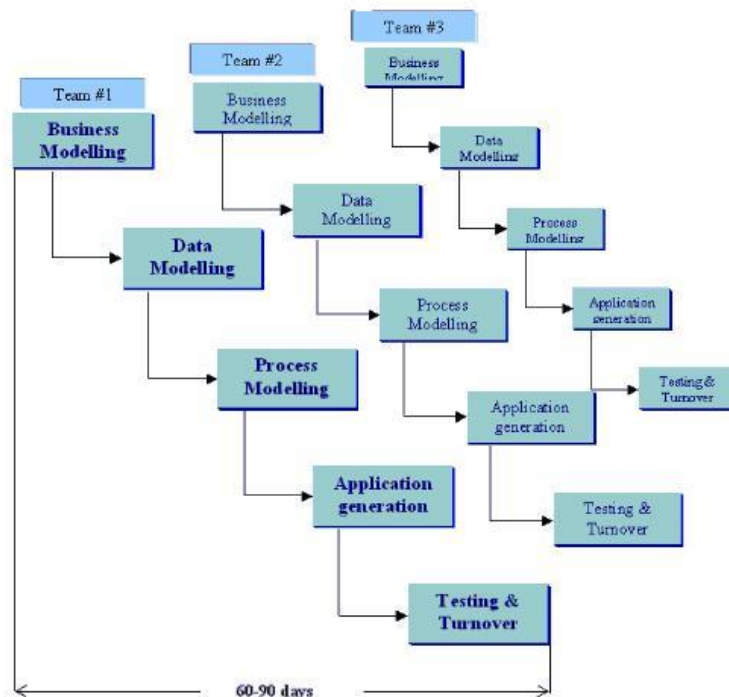
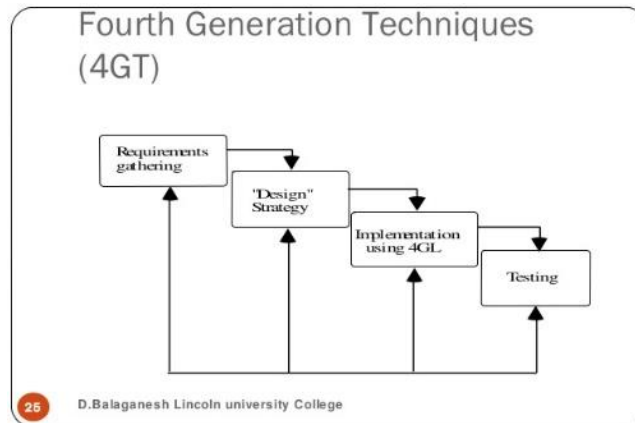


Figure 1.5 – RAD Model

Gambar 5. Model RAD

e) Model 4GT

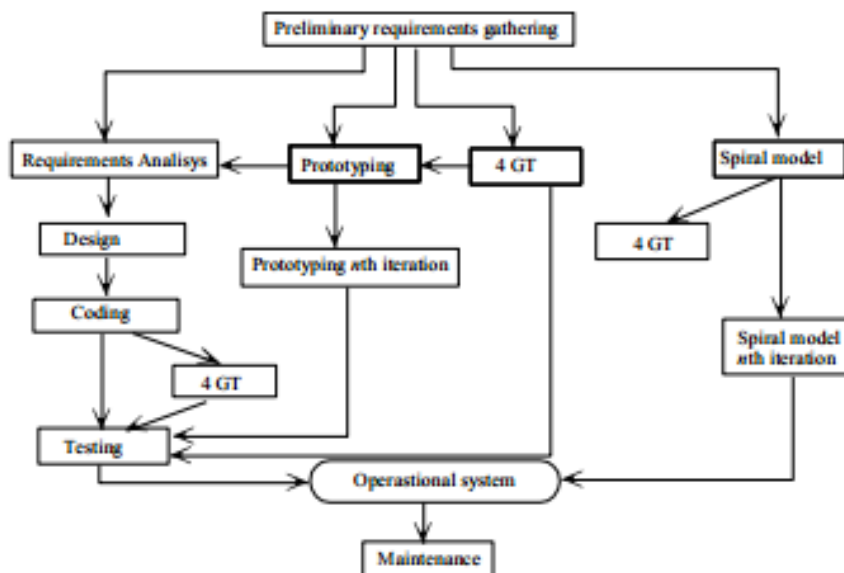
Istilah Fourth Generation Techniques (4GT) mencakup seperangkat peralatan perangkat lunak yang berfungsi sebagai perangkat bantu yang memudahkan seorang pengembang software mengaplikasikan beberapa karakteristik software pada tingkat yang tinggi, yang akan menghasilkan source code dan object code secara otomatis sesuai dengan spesifikasi (persyaratan khusus) yang dibuat oleh sang pengembang perangkat lunak.



Gambar 6. Model 4GT

f) Gabungan Beberapa Model Pengembangan

Gabungan beberapa model pengembangan ini memiliki tujuan yaitu untuk mengambil kelebihan masing-masing paradigma. Berikut ini merupakan gambaran dari gabungan beberapa model pengembangan:



Gambar 7. Gabungan dari beberapa model pengembangan

C. Soal Tugas

- 1) Sebutkan pengertian dari perangkat lunak dan rekayasa perangkat lunak !
- 2) Jelaskan yang dimaksud dengan program komputer !
- 3) Jelaskan perbedaan dari model *waterfall* dan *prototype* !
- 4) Apa kegunaan dari model pengembangan perangkat lunak ?
- 5) Menurut kalian, model pengembangan perangkat lunak mana yang mudah dilakukan ketika kalian ingin mengembangkan sebuah program komputer ? Berilah alasan kalian !

D. Referensi

- 1) Mulyanto, Aunur Rofiq. (2008). *Rekayasa Perangkat Lunak untuk Sekolah Menengah Kejuruan (Jilid 1)*. Jakarta: Direktur Pembinaan Sekolah Menengah Kejuruan.
- 2) Munif, Abdul. (2019). *Pemodelan Perangkat Lunak (C3) Kelas XI*. Malang: PT. Kuantum Buku Sejahtera.
- 3) Wardani, Ratna. (2012). *Rekayasa Perangkat Lunak*. Universitas Negeri Yogyakarta: Jurusan Pendidikan Teknik Elektronika.