PERTEMUAN 1
(RUANG LINGKUP REKAYASA PERANGKAT LUNAK)

A. TUJUAN PEMBELAJARAN

Diharpkan pada tahap ini, mahasiswa dapat memahami beberapa hal berikut.

1. Memahami Ruang Lingkup Rekayasa perangkat lunak.
2. Mengetahui Sasaran dan tujuan Rekayasa Perangkat Lunak.

B. URAIAN MATERI

1. Sejarah RPL (Rekayasa Perangkat lunak)

RPL pertama kali dikembangkan di tahun 1940-an, dimana RPL berfokus pada praktek pembuatan aplikasi, agar para pengembang perangkat lunak mengetahui kualitas suatu aplikasi yang akan digunakan oleh pengguna.

Rekayasa Perangkat Lunak (RPL) adalah salah satu bidang profesi dan juga sebagai mata pelajaran yang mempelajari tentang pengembangan perangkat-perangkat lunak, termasuk dalam pembuatan, pemeliharaan hingga manajemen organisasi dan manajemen kualitasnya suatu perangkat lunak.

IEEE PC Society mencirikan pemrograman komputer sebagai metode untuk menerapkan cara yang dibangun, terkendali dan terukur.

Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan. Adapun ilmu komputer berfokus pada teori dan konsep dasar perangkat komputer. Rekayasa perangkat lunak lebih berfokus pada bagaimana membuat memenuhi keriteria sebagai berikut:

a. Dapat terus di pelihara setelah perankat lunak selesai di buat sering berkembangnya teknologi dan lingkungan *(maintainability).*
b. Dapat di andalkan dengan proses bisnis yang dijalankan dan perubahan yang terjadi (*dependability* dan *robust*).
c. Efisien dari segi sumber daya dan penggunaan.
d. Kemampuan yang di pakai sesuai dengan kebutuhan (*usability)*

Dari kriteria di atas maka perangkat lunak yang baik adalah perangkat lunak yang memenuhi kebutuhan pelanggan atau *user* (pemakai perangkat lunak) atau berorientasi pada pelanggan atau pemakai perangkat lunak, bukan berorientasi pada pembuat atau pengembang perangkat lunak.

a. RPL (Rekaysa perangkat lunak)

RPL secara garis besar di sepakati sebagai *Software Engineering.* Merupakan pembangunan dengan menggunakan nilai prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien dengan menggunakan mesin. Perangkat lunak banyak di buat dan pada akhirnya sering tidak digunakan karena tidak memenuhi kebutuhan pelanggan atau bahkan masalah non-teknis seperti pelanggan pemakai perangkat lunak *(User)* untuk mengubah cara kerja dari manual ke otomatis atau ketidakmampuan *user* menggunakan Komputer. Maka dari itu RPL (Rekayasa Perangkat Lunak) dibutuhkan agar perangkat lunak yang di buat tidak hanya perangkat lunak yang tidak terpakai.

Pengertian RPL tersendiri ialah ilmu yang mendisiplikan untuk membahas semua aspek produksi pada *software*, yang di awali dengan menganalisi kebutuhan penggunanya, menetukan desain yang di gunakan, pengkodean, pengujian hingga perawatan system setelai di pakai.

b. Perspektif pemrograman komputer

Bagian dari komponen manusia dapat ditunjukkan pada tingkat masalah, masalah yang diperiksa adalah disiplin yang diidentifikasi dengan klien yang dapat membantu seseorang untuk memahami premis masalah. Pada tingkat pengaturan, perhatian utama adalah disiplin terkait klien yang memungkinkan ses`eorang ditunjuk dalam pemikiran kritis yang lebih baik. Suatu usaha dapat dianggap secara kolektif tugas dan latihan diselesaikan dalam suatu periode. Sudut pandang mencakup kumpulan aset singkat yang bertemu untuk menangani suatu masalah.

Sarana yang diperlukan untuk suatu tugas diselesaikan dengan menggambarkan peluang, memberikan gambaran dasar yang berlaku untuk prasyarat usaha, dan kemudian menggambarkan tujuan dari usaha tersebut. Setelah mengenali masalah dan aset yang dapat diakses, seseorang harus

membedakan pekerjaan terkait agar tidak mempengaruhi atau dipengaruhi oleh usaha yang layak. Kemudian, pada saat itu mengenali aturan untuk memilih apakah suatu usaha adalah satu yang layak, pemahaman yang mencakup batasproyek, kecurigaan, dan bahaya, serta konsekuensi dari hambatan dan anggapan untuk bahaya proyek. Tugas para eksekutif dapat digambarkan sebagai sekumpulan standar, strategi, instrumen, dan prosedur untuk mengatur, mengatur staf, mengoordinasikan, dan mengendalikan latihan terkait bisnis untuk mencapai target proyek dalam waktu, di bawah biaya, dan keharusan pelaksanaan.

c. Kebutuhan Pemrograman

Sebuah Prasyarat Praktis Apakah bantuan yang harus diberikan oleh kerangka kerja, bagaimana kerangka kerja mendapatkan dan siklus dan bagaimana kerangka kerja menangani keadaan. Demikian juga, itu dengan jelas mencirikan apa yang tidak dilakukan kerangka kerja. Prasyarat utilitarian adalah seluk-beluk seperti sumber informasi, hasil dan kasus khusus terkait.

1) Kebutuhan Non-utilitarian:

Pada umumnya mengandung hambatan pada administrasi atau kapasitas yang diberikan oleh kerangka kerja. Menghitung batas waktu, batasan ukuran perbaikan,norma tertentu. Karena mengidentifikasi dengan persyaratan kerangka secara keseluruhan, ketidak mampuan untuk mengatasi masalah semacam ini menyebabkan kerangka kerja. Untuk kebutuhan semacam ini adalah kecepatan akses, keamanan informasi, ukuran batas kapasitas, perlindungan setiap profil, bahasa pemrograman yang digunakan, kerangka kerja yang digunakan.

Kebutuhan non-praktis dipisahkan menjadi 3 macam, yaitu:

a) Permintaan barang untuk keandalan, kecepatan, kegunaan, batas memori yang diperlukan, dan efektivitas kerang kakerja
b) Permintaan otoritatif untuk prinsip program, dialek pemrograman dan teknik rencana yang digunakan.
c) Permintaan luar untuk masalah pemanfaatan, interoperabilitas dengan kerangka kerja yang berbeda, keabsahan, dan perlindungan.

2) Rencana Pemrograman

Menggabungkan cara untuk menunjukkan rekayasa, segmen, antarmuka, dan kualitas produk dengan Model. Cara untuk mengubah prasyarat kerangka item total. Konfigurasi pemrograman terdiri dari siklus:

a) Peluruhan kerangka di beberapa area
b) Sebutkan kewajiban setiap bidang
c) Menjamin segmen dapat mencapai tujuan.
d) Pemrograman Pembangunan.

Mengelola seluk-beluk kemajuan pemrograman, termasuk perhitungan, pengkodean, penemuan kekurangan dan pengujian.

Model : Peningkatan kerumitan ini dapat dicapai dengan menyiapkan pedoman dan metode yang berbeda untuk mengharapkan perubahan. Tes Pemrograman mengingat pengujian untuk eksekusi pemrograman umum.

Model : Sebuah Model Penentuan Uji/Ukuran Kecukupan Uji (Standar Penghentian).

Model penentuan tes adalah untuk memverifikasi bahwa sekelompok eksperimen memadai untuk alasan yang telah ditentukan sebelumnya. Ukuran kecukupan pengujian bisa dipakai untuk menentukan kapan ujicoba yang memadai akan digunakan, atau sudah selesai.

a) Menguji Viabilitas
   Pengujian viabilitas adalah memeriksa perkembangan eksekusi program. Penentuan tes yang akan dijalankan dapat diarahkan oleh berbagai target, mengingat tujuan yang ingin dicapai.
b) Menguji Deformitas
   Untuk tes yang bermanfaat adalah tes yang membuat kerangka gagal. bervariasi dari pengujian untuk menunjukkan bahwa produk tersebut memenuhi penentuan, dalam hal pengujian tersebut bermanfaat jika tidak ada kekecewaan yang terlihat dalam eksperimen dan kondisi pengujian yang masuk akal.

c) Masalah oracle

Oracle adalah klien atau spesialis yang memutuskan apakah suatu program valid atau palsu dalam tes karena menghasilkan pilihan "lulus" atau "gagal".

d) Batas Pengujian Hipotetis dan Layak

Hipotesis uji memperingatkan agar tidak membawa tingkat kepastian yang tidak beralasan ke dalam serangkaian pengujian.

e) Masalah Cara yang Tidak Layak

Cara yang tidak layak adalah cara aliran kontrol yang tidak dapat dilakukan dengan informasi, terutama dalam pengurangan terprogram dari kontribusi pengujian untuk mengontrol cara kerangka kerja.

f) Tes Pemrograman

Percobaan pengujian akan menunjukkan kekurangan jika produk tidak sesuai.

d. Dukungan Pemrograman

Menggabungkan upaya dukungan setelah produk ditempatkan ke dalam aktivitas.

Model :

1) Dukungan untuk mengatasi kesalahan dalam pemrograman dan mengembangkan variasi pemrograman lebih lanjut ke iklim fungsionalnya

2) Dukungan untuk menambah dan menyesuaikan kapasitas kerangka kerja.

e. Pengaturan Pemrograman

Terkait dengan upaya untuk mengubah desain pemrograman untuk memenuhi kebutuhan eksplisit.

Model :

1) Ubah bukti yang dapat dikenali
2) Ubah kontrol
3) Menjamin bahwa perubahan dilakukan dengan baik
4) Menjawab pertemuan yang berbeda karena kurang beruntung

Pemrograman Eksekutif diidentifikasi dengan administrasi dan estimasi RPL, termasuk penyusunan program proyek Model :

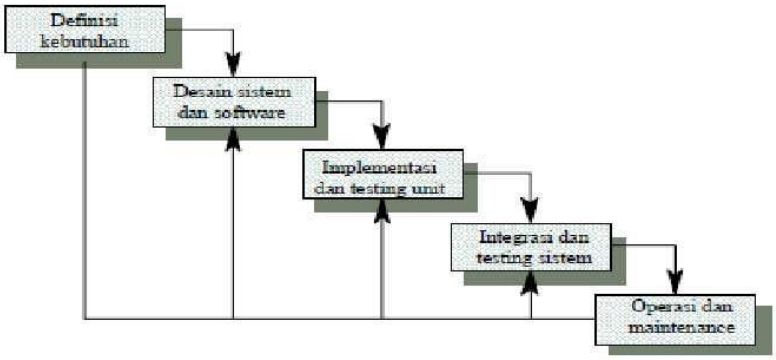
1) Sebuah. Sebagai ukuran asosiasi, pemrograman menjadi lebih baik dan lebih andal diterapkan di seluruh asosiasi
2) Pencapaian interaksi produk adalah sejauh mana siklus itu eksplisit, diawasi, diperkirakan, dikendalikan, dan kuat.

Perangkat dan Teknik Pemrograman Komputer, menggabungkan penyelidikan hipotetis perangkat dan strategi RPL.

Model :

1) Sebuah. (Peralatan)
2) Pemrograman)
3) Brainware (Klien/SDM).

2. Metode air terjun



Gambar 1. 1 Metode air Terjun

Dengan Metode ini, menonjolkan kualitas pemrograman dan siklus hidup.
Model : Kualitas pemrograman dipisahkan menjadi 2, yaitu:

a. Konfirmasi Kualitas Pemrograman (SQA)

Sebuah. Pendekatan Kemajuan Pemrograman

1) Venture Dewan
2) Desain Para eksekuti
3) Peningkatan Kebutuhan/Para eksekutif
4) Penilaian

5) Rencana Pemrograman
6) Pengujian dan seterusnya

b. Kontrol Kualitas Pemrograman (SQC)

1) Tes Satuan
2) Pengujian Penggabungan
3) Uji Kerangka
4) Pengujian Pengakuan

c. Pemrograman Perancangan Siklus

mengelola pelaksanaan, definisi, estimasi, papan, perubahan dan peningkatan langkah-langkah Pemrograman degan Model :

1) Sebuah. Panduan untuk membuat item pemrograman yang hebat.
2) mengatasi masalah pemrogram komputer.
3) Berikan sistem untuk latihan.
4) Berbagai jenis usaha dan memerlukan langkah-langkah pemrograman yang beragam.
5) Item kerja yang dibuat oleh langkah-langkah pemrograman.
6) Sebagai penanda ukuran pemrograman yang tepat.

d. Pemrograman kerangka kerja

Pemrograman yang kenyamanannya lebih terencana untuk tugas-tugas PC. Kerangka kerja Penerjemah bahasa pemrograman (kompiler/mediator)

e. Pemrograman aplikasi

Pemrograman yang nilainya lebih terencana untuk membantu menangani masalah yang dilihat oleh klien.

f. Program bundel instan

Program aplikasi yang dibangun secara asli Sementara itu, menurut aplikasi, produk diisolasi menjadi Pemrograman Kerangka (Framework Programming) Bermacam-macam proyek yang disusun untuk melayani proyek yang berbeda, misalnya editor, driver, dan sebagainya.

g. Pemrograman Konstan

Pemrograman yang digunakan untuk mengkuantifikasi/menguraikan atau mengontrol ukuran jalur informasi dari iklim luar untuk membuat laporan laporan yang ideal

1) Pemrograman Bisnis (Business Programming)

Pemrograman yang memberikan kantor fungsional untuk bisnis atau kantor eksekutif yang dinamis, misalnya kerangka pembukuan, stok, keuangan,dan lainnya

2) Merancang dan Pemrograman Logis

Produk yang digunakan dalam bidang perancangan aplikasi dan pemrograman semacam ini biasanya dikenal dengan penggambaran informasi matematis,perancangan berbantuan komputer (PC Helped Plan), pembuatan kerangka kerja, dan lain-lain.

3) Pemrograman Terpasang

Pemrograman yang digunakan untuk mengontrol item dan kerangka tempat produk disimpan. Umumnya diposisikan di ROM, misalnya Catch in Microwave.

4) Pemrograman PC (Pemrograman PC)

Banyak digunakan dalam aplikasi tunggal, misalnya: penanganan kata, halaman pembukuan, game, DBMS dan lain-lain.

Pemrograman Kesadaran Buatan Manusia (Pemrograman Cerdik Palsu) Dibuat dengan menggunakan prosedur perhitungan non-numerik untuk menangani masalah yang kompleks, digunakan dalam bidang aplikasi penalaran buatan, misalnya: game, kerangka kerja master, organisasi saraf, Super Prolog, dan lain-lain.

5) Model Interaksi Pemrograman Transformatif

Model transformatif adalah model berulang, dijelaskan oleh praktik yang memungkinkan pemrogram untuk mendorong rendisi pemrograman yang lebih lengkap sedikit demi sedikit. Barang dan kebutuhan bisnis sekarang dan lagi berubah dengan kecepatan kemajuan. Dalam situasi ini dan situasi yang berbeda, pemrogram membutuhkan model interaksi yang benar-benar dimaksudkan untuk

mewajibkan perbaikan item setelah beberapa waktu. Model ini tidak mengecualikan pemrograman komputer tradisional.

Model perkembangan meliputi:

a) Model ekspansi

Model bertahap menggabungkan komponen model berurutan langsung (diterapkan berulang-ulang) dengan cara berpikir model iteratif. Model ini menggunakan pengelompokan langsung dalam model yang membingungkan, seiring dengan kemajuan waktu jadwal. Setiap pengelompokan langsung menghasilkan pemrograman yang stabil dan "dapat disampaikan".Misalnya, program penyusunan kata yang dibuat menggunakan pandangan dunia bertahap akan memberikan catatan papan, mengubah, dan melaporkan kapasitas usia dalam peningkatan utama, dll. Pengembangan utama dapat disebut sebagai item utama.

Model ini menyoroti penyampaian item fungsional di setiap augmentasi. Perbaikan awal adalah dalam penampilan hasil akhir yang dipreteli, namun memberikan kemampuan untuk melayani klien dan selanjutnya memberikan tahap penilaian klien.

Pergantian peristiwa secara bertahap, terutama membantu dalam kepegawaian, tidak mungkin memanfaatkan eksekusi penuh dengan waktu cutoff bisnis yang disepakati untuk tugas tersebut. jika item pusat umumnya diterima, staf tambahan mungkin ditambahkan untuk melakukan opsi tambahan.

b) Model berliku

Awalnya diusulkan oleh Boehm (BOE88), itu adalah model ukuran pemrograman perkembangan, mengumpulkan ide berulang model melalui bagian yang terkontrol dan teratur dari model berurutan langsung. Sebuah model yang memiliki potensi untuk kemajuan yang cepat dari bentuk-bentuk pemrograman bertahap. Model puntir dipisahkan menjadi berbagai latihan struktur atau wilayah usaha, termasuk:

Korespondensi klien, tugas yang diperlukan untuk membangun korespondensi yang kuat antara desainer dan klien.

Mengatur, tugas yang diperlukan untuk mengkarakterisasi aset, idealisme, dan proyek data terkait lainnya.

Pemeriksaan bahaya, usaha yang diperlukan untuk melihat peluang, baik administrasi maupun khusus.

Merancang, usaha yang diperlukan untuk membangun setidaknya satu penggambaran aplikasi.

Pengembangan dan pengiriman, usaha yang diperlukan untuk membangun, menguji,(memperkenalkan), dan menawarkan jenis bantuan kepada klien (misalnya mempersiapkan dan mendokumentasikan).

Penilaian klien, upaya untuk mendapatkan kritik klien tergantung pada penilaian penggambaran program, dibuat selama perancangan, dan dilaksanakan selama pembuatan.

Model berliku adalah metodologi yang masuk akal untuk kerangka lingkup besar dan kemajuan pemrograman. Karena pemrograman terus berfungsi saat interaksi bergerak, desainer dan klien memahami, dan merespons dengan lebih baik bahaya dari setiap fase kemajuan. Model belitan menggunakan model sebagai sistem penurun bahaya.

Model berliku membutuhkan kemampuan pemahaman bahaya yang masuk akal, dan sangat bergantung pada keahlian ini untuk membuat kemajuan. Jika bahaya tidak dapat ditemukan dan diawasi, akan ada masalah. Model ini membutuhkan waktu bertahun-tahun hingga kualitasnya yang tak tergoyahkan dapat dipertimbangkan dengan penuh keyakinan.

c) Model pengumpulan segmen

Model ini menggabungkan beberapa atribut dari model memutar. Bersifat transformatif, sehingga membutuhkan cara yang iteratif untuk menangani pembuatan program. Namun, model ini mengumpulkan aplikasi dari segmen pemrograman sebelum dibundel (kadang-kadang disebut kelas).

Pergerakan produk dimulai dengan bukti pembeda dari kelas yang akan datang. Sarat dengan memperhatikan informasi yang akan dikendalikan oleh aplikasi dan perhitungan yang akan diterapkan. Informasi dan perhitungan terkait digabungkan ke dalam kelas. Kelas-kelas ini disimpan di perpustakaan kelas (penimbunan).

Model ini mendorong penggunaan kembali pemrograman, dan penggunaan kembali itu memberikan berbagai manfaat terukur untuk pemrograman komputer.

d) Model formatif yang konsisten

Penggambaran latihan dalam model ini, termasuk latihan ilmiah, bisa dalam keadaan apapun yang direkam secara tiba-tiba. Demikian pula, latihan yang berbeda (rencana atau korespondensi klien) dapat ditangani dengan cara yang tidak berbeda. Semua latihan ada secara bersamaan namun ia hidup dalam keadaan alternatif.

Model ini secara teratur digunakan sebagai pandangan dunia bagi para desainer.

3. Peningkatan pemrograman komputer

Meski baru diluncurkan pada tahun 1968, RPL memiliki sejarah panjang. Dari segi disiplin, RPL masih tergolong muda dan akan terus berkreasi.

Judul peningkatan yang saat ini sedang dibuat meliputi acara di tahun:

a. 1940-an PC pertama yang mengizinkan klien untuk membuat kode program langsung
b. 1950-an Usia awal penerjemah dan dialek skala besar Asli dari penyusun
c. 1960-an Era kedua kompiler PC server terpusat mulai dipopulerkan. Peningkatan pemrograman khusus Ide pemrograman komputer mulai digunakan
d. 1970-an Perangkat peningkatan pemrograman Minikomputer bisnis Perangkat Keras Bisnis (PC) 1980-an Memperluas minat untuk pemrograman

e. 1990-an Pemrograman yang diatur item (OOP) Siklus Cekatan dan Pemrograman Luar Biasa Ekspansi luar biasa dalam batas memori Penggunaan web yang diperluas
f. 2000-an Tahap penerjemah saat ini (Java, .Net, PHP, dan sebagainya)

Komponen pemrograman komputer. Berbagai elemen yang mempengaruhi pengaturan, pelaksana, dan pilihan latihan dan prosedur SQM adalah sebagai berikut:

a. Ruang kerangka kerja yang akan dimiliki produk (dasar keamanan, penting, dan dasar bisnis), Kerangka kerja dan prasyarat pemrograman, Bagian bisnis (luar) atau standar (dalam) untuk digunakan di dalam kerangka.
b. Prinsip-prinsip pemrograman komputer materi eksplisit,Teknik dan instrumen pemrograman yang digunakan untuk kemajuan dan pemeliharaan, dan untuk peningkatan dan penilaian kualitas.
c. Rencana keuangan, staf, asosiasi proyek, rencana, dan pemesanan, semua hal dipertimbangkan,
d. Klien yang diantisipasi dan pemanfaatan kerangka kerja, dan kerangka tingkat kejujuran.

C. SOAL LATIHAN/TUGAS

1. Sebutkan contoh konsep dan tujuan rekayasa perangkat lunak ?
2. Bagaimana cara pengujian viabilitas yang baik dan benar ?
3. Apa yang dimaksud dengan kualitas control pemograman menggunakan metode SQC ?
4. Bagaimana contoh pemograman konstan yang benar menurut anda ?
5. Sebutkan contoh model pengumpulan segmen ?

D. REFERENSI

1. https://www.dosenpendidikan.co.id/rekayasa-perangkat-lunak/
2. https://www.seputarpengetahuan.co.id/2018/04/pengertian-rekayasa-perangkat-lunak-tujuan-kriteria-ruang-lingkup.html

3. https://www.slideshare.net/viiasilviaa/resume-manajemen-resiko-rpl.
4. https://journal.ibs.ac.id/index.php/JEMP/article/download/36/43
5. https://www.slideshare.net/viiasilviaa/resume-manajemen-resiko-rpl.

GLOSARIUM

IEEE PC Society mencirikan pemrograman komputer sebagai metode untuk menerapkan cara yang disengaja, terkendali dan terukur untuk menangani pergantian peristiwa, penggunaan dan pemeliharaan pemrograman.

Oracle adalah klien atau spesialis yang memutuskan apakah suatu program valid atau palsu dalam tes karena menghasilkan pilihan "lulus" atau "gagal".

Model ekspansi Model bertahap menggabungkan komponen model berurutan langsung (diterapkan berulang-ulang) dengan cara berpikir model iteratif. Model ini menggunakan pengelompokan langsung dalam model yang membingungkan, seiring dengan kemajuan waktu jadwal. Setiap pengelompokan langsung menghasilkan pemrograman yang stabil dan "dapat disampaikan".