# **BAB III**

**PEMBAHASAN DAN TEMUAN**

## **Analisis dan Perancangan**

1. Metode Pengembangan Sistem Proses Perangkat Lunak

Menurut riset dari IBM, pengembangan perangkat lunak merupakan istilah dalam ilmu komputer. Maknanya adalah serangkaian aktivitas yang dilakukan untuk merancang, membuat, mengaplikasikan, dan mendukung atau meningkatkan fungsi perangkat lunak. Metode pengembangan perangkat lunak adalah sebuah cara teratur untuk merancang atau meningkatkan fungsi sebuah program agar pengguna lebih mudah dalam mengoperasikan komputer.

Metode Pengembangan Sistem Proses Perangkat Lunak yang digunakan dalam pembuatan Sistem penjadwalan Upacara ini adalah Metode Prototype.

Metode Prototype dalam bahasa Indonesia diartikan dengan istilah purwarupa. Istilah tersebut berarti model awal atau rancangan sementara yang masih membutuhkan berbagai penyesuaian sebelum dinyatakan telah memenuhi hasil yang diinginkan. Terdapat lima tahapan dalam pengembangan perangkat lunak menggunakan metode prototype. Dimulai dengan pengumpulan informasi dari pelanggan ke tim pengembang. Lalu, tim akan merencanakan sistem dan mengerjakan purwarupa perangkat lunak.

Hasilnya kemudian diserahkan kepada penguji untuk dievaluasi. Jika terdapat permasalahan, tim akan merevisi sistem tersebut hingga benar-benar sesuai dengan kebutuhan penguji. Ketika revisi selesai dan telah diterima, berarti perangkat lunak telah siap diterjemahkan ke dalam perangkat keras. Kemudian, dilanjutkan dengan proses uji coba dan berbagai revisi sebelum mulai dapat digunakan.

* Kekurangan dan Kelebihan Metode Prototype

Metode prototype digunakan apabila pemilik proyek tahu benar apa yang diinginkannya, tetapi tidak mengetahui bagaimana cara mengaplikasikannya. Kuncinya terletak pada komunikasi yang baik antara penguji dan pengembang.

Penguji harus menyampaikan kebutuhannya secara jelas. Sementara pengembang juga diharapkan mampu menerjemahkan informasi tersebut agar dapat menghasilkan perangkat lunak yang sesuai. Jika tidak, prototype ini tidak akan menjadi metode yang efektif.

Pengembang memperoleh tantangan besar dari penguji. Sebaliknya, penguji dapat terpuaskan jika pengembang berhasil memenuhi kebutuhannya. Kerja sama kedua pihak akan saling menguntungkan.

* Model Metode Pengembangan Perangkat Lunak

1. Metode Waterfall

Waterfall merupakan metode pengembangan perangkat lunak tradisional yang sistematis. Metode ini memiliki lima tahapan proses, di antaranya Communication, Planning, Modeling, Construction, dan Deployment. Communication merupakan fase di mana pelanggan atau pemilik proyek menyampaikan kebutuhan dan permasalahannya kepada pengembang. Lalu, bersama-sama mengumpulkan data-data yang diperlukan dan merumuskan fitur-fitur perangkat lunak.

Selanjutnya, menginjak pada proses perancangan. Dimulai dengan merumuskan estimasi kerja, kebutuhan sumber daya, serta perencanaan alur kerja. Berlanjut dengan tahap perancangan struktur data, arsitektur, tampilan, dan algoritma perangkat lunak. Rancangan kemudian coba diaplikasikan pada perangkat keras komputer dalam bentuk bahasa pemograman. Construction juga mencakup tahapan uji coba pengoperasian perangkat lunak untuk mengetahui kelemahannya.

Setelah berhasil dibuat, perangkat lunak disebarluaskan untuk diimplementasikan pada perangkat pengguna secara umum. Temuan-temuan dari pengguna, akan menjadi bahan bagi pengembang untuk mengevaluasi dan memperbaiki perangkat lunak lebih jauh lagi.

* Kekurangan dan Kelebihan Metode Waterfall

Metode waterfall melibatkan berbagai proses yang sistematis dan komprehensif. Sumber daya dan tahapan pengerjaannya dikumpulkan secara lengkap sehingga dapat mencapai hasil maksimal. Sayangnya, proses tersebut memakan waktu lebih lama.

Sistem dalam metode waterfall merupakan proses yang baku, sehingga pengembang sulit melakukan improvisasi. Itulah mengapa metode ini dianggap kurang efektif dan seringkali hanya dipakai dalam pengembangan perangkat lunak atau sistem berskala besar.

1. Metode Spiral

Metode spiral menggabungkan dua metode pengembangan yang telah dibahas sebelumnya, yaitu prototype dan waterfall. Pengembang melaksanakan prototyping dengan cara sistematis khas metode waterfall. Umumnya metode spiral diterapkan dalam pengembangan perangkat lunak berskala besar, sekaligus membutuhkan sistem yang kompleks. Setiap prosesnya selalu disertai dengan analisis mendalam mengenai tingkat risiko dan keberhasilan pengembangan.

Pelaksanaan metode spiral dilakukan dalam lima langkah. Pertama adalah komunikasi, yaitu pemilik proyek menyampaikan kebutuhannya kepada pengembang perangkat runak. Dilanjutkan dengan perencanaan mendetail tentang proyek yang digarap.

Langkah perencanaan diikuti dengan analisis untuk mengidentifikasi berbagai kemungkinan yang bisa terjadi selama pengembangan. Kemudian, pengembangan perangkat lunak mulai dijalankan dan setelah jadi akan mendapatkan evaluasi dari pelanggan. Kelima langkah tersebut dilakukan secara berulang-ulang pada setiap tahapan pengembangan. Sejak dari pengembangan konsep, pengembangan prototype, perbaikan, perubahan, hingga pemeliharaan sistem yang telah jadi.

* Kekurangan dan Kelebihan Metode Spiral

Metode spiral menerapkan alur kerja yang kompleks, panjang, dan memakan waktu lama. Metode ini tidak cocok untuk proyek kecil-kecilan, apalagi yang berbujet rendah. Sebaliknya, proyek berskala besar dapat dikerjakan dengan mudah menggunakan metode ini.

Selain itu, metode spiral juga kurang tepat jika dilakukan oleh pengembang perangkat lunak yang belum berpengalaman. Prosesnya bisa menjadi terlalu kompleks bagi pemula yang masih memerlukan banyak latihan.

1. Metode RAD

RAD merupakan singkatan dari Rapid Application Development. Metode ini juga menggunakan pendekatan iteratif dan inkremental, tetapi lebih menekankan pada tenggat waktu dan efisiensi biaya yang sesuai dengan kebutuhan Proses pengembangan dengan Metode RAD dianggap lebih singkat. Pasalnya, semua pihak, baik pelanggan maupun pengembang, terus terlibat secara aktif dalam setiap proses hingga hasil dapat tercapai. Di samping itu, tahapan kerja pada metode ini juga lebih sedikit.

Alur kerja hanya dibagi menjadi tiga tahap yang semuanya padat. Identifikasi tujuan yang langsung diiringi dengan komunikasi dan perancangan, di mana seluruh pihak terlibat aktif dalam setiap perumusannya. Proses ini menjadi tahap awal dari Metode RAD. Tahap kedua masih melibatkan semua pihak, yaitu proses mendesain sistem atau perangkat lunak sesuai kebutuhan. Pelanggan atau pengguna ikut terjun dalam menguji coba perangkat lunak. Perbaikan pun langsung diterapkan jika pengguna menemukan kesalahan.

Ketika pengguna terpuaskan dengan desain perangkat lunak, setelah melalui berbagai perbaikan, barulah proses kerja menginjak pada tahap terakhir, yaitu implementasi. Desain perangkat lunak mulai diterjemahkan dalam bahasa mesin dan bisa digunakan.

* Kekurangan dan Kelebihan Metode RAD

Beberapa kekurangan dari metode RAD, antara lain dilihat dari segi konsistensi dan kemampuan personel. Metode ini membutuhkan pengembang ahli, sekaligus kerjasama yang aktif dan konsisten antara pemilik proyek beserta semua tim.

Tanpa kedua hal tersebut, mustahil menerapkan metode RAD dalam pengembangan perangkat lunak, apalagi yang berskala besar. Namun jika kedua hal itu terakomodasi dengan baik, metode RAD adalah cara paling efektif untuk menghemat waktu dan biaya.

1. Aliran Proses
2. Pengertian Use Case Diagram

Use Case adalah sebuah kegiatan atau juga interaksi yang saling berkesinambungan antara aktor dan juga sistem. Atau dengan kata lain teknik secara umum digunakan, guna mengembangkan software atau sistem informasi, guna memperoleh kebutuhan fungsional dari sistem yang ada.

Komponen tersebut kemudian menjelaskan komunikasi antara actor, dengan sistem yang ada. Dengan demikian, use case dapat dipresentasikan dengan urutan yang sederhana, dan akan mudah dipahami oleh para konsumen Use case ini adalah layanan atau juga fungi yang ada pada sistem untuk para penggunanya. Sedangkan Use case Diagram, adalah gambaran efek fungsionalitas yang diharapkan oleh sistem.

Komponen ini pastinya sangat membantu ketika Anda sedanh menyusun requitment pada sebuah sistem, kemudian mengkomunikasikan rancangan aplikasi tersebut pada konsumen, dan juga merancang test case untuk berbagai fiture yang ada di sistem. Use case ini dapat digunakan atau diaplikasikan ke use case lainnya, sehingga duplikasi fungsionalitas dapat dihindari, caranya adalah dengan menarik keluar fungsional yang common.

1. Fungsi Use Case Diagram
2. Dapat memperlihatkan urutan aktifitas proses yang ada pada sistem.
3. Dapat menggambarkan prose bisnis dan juga urutan aktivitas yang ada dalam sebuah proses.
4. Manfaat Use Case Diagram
5. Untuk memudahkan komunikasi dengan menggunakan domain expert dan juga end user.
6. Adanya Interface yang harus dimiliki oleh sebuah sistem.
7. Memberikan kepastian pemahaman yang pasti, tentang requirement atau juga kebutuhan sebuah sistem.
8. Dapat digunakan untuk mengidentifikasi, siapa yang sedang berinteraksi dengan sistem, dan juga apa yang harus dilakukan untuk sistem tersebut.
9. Biasanya digunakan untuk verifikasi.
10. Komponen Use Case Diagram
11. Sistem

Sebagai batasan sistem yang ada pada relasi dengan actor actor yang biasa menggunakannya dari luar sistem, dan fitur-fitur yang harus disediakan dalam sistem. Perangkat ini akan digambarkan dengan pola segi empat yang akan membatasi semua use case yang ada dalam sistem terhadap pihak yang mana sistem tersebut akan berinteraksi. Sistem tersebut akan dibeli label, sesuai dengan nama label tersebut. Tapi umumnya sistem ini tidak diberi gambar karena tidak terlalu memberikan arti bagi diagram tersebut.

1. Actor

Actor disini sebenarnya bukan bagian dari diagram, akan tetapi actor ini adalah salah satu alat untuk menciptakan use case diagram jadi lebih mudah. Actor ini akan digunakan untuk menjelaskan sesuatu atau juga seseorang yang sedang berinteraksi dengan sistem.

Dimana actor ini akan memberikan informasi pada sistem, hanya akan menerima informasi dari sistem, atau juga keduanya (yaitu memberikan informasi dari sistem, atau menerima informasi dari sistem). Disini Actor juga tidak akan memiliki kontrol terhadap use case itu sendiri, namun hanya diberikan gambaran umum atau juga spesifik, guna memudahkan Anda dalam menggunakan hubungan atau relationship yang ada.

Ada beberapa hal yang memungkinkan actor berhubungan dengan sistem lainnya, antara lain :

* Ketika seseorang ataupun pihak lain akan mengelola sistem tersebut.
* Adanya kepentingan terhadap sistem, dimana arus informasi baik untuk si penerima maupun inputan sistem saling berkepentingan.
* Adanya external resource yang digunakan oleh sistem tersebut.
* Adanya sistem lain yang tengah berinteraksi dengan sistem yang dibuat.

1. Use case

Use Case sendiri adalah gambaran fungsional dari sebuah sistem. Dengan demikian, antara konsumen dan juga pengguna pada sistem tersebut, akan mengerti atau paham mengenai fungsi sistem yang tengah dibangun.

1. Relasi dalam Use Case Diagram
2. Association

Ini adalah teknik mengindentifikasi interaksi yang dilakukan oleh actor tertentu dengan use case tertentu pula. Hal ini digambarkan dengan garis antara actor tehadap use case tersebut. Asosiasi ini biasanya ditandai dengan garis 1 arah (ditandai dengan garis anak panah), jika terjadi komunikasi 2 arah, maka tanda panas tidak akan diperlukan.

1. Generalization

Ini adalah teknik mengindetifikasi relasi antara 2 aktor can juga 2 Use case, dimana salah satunya akan meng- inherit dan juga menambah atau override sifat dari perangkat lainnya. untuk teknik penggambarannya menggunakan garis bermata panah yang kosong. Garis akan diambil dari yang meng-inherit kemudian mengarah ke yang di-inherit.

1. Dependency

Dependency ini terbagi menjadi 2 macam, yaitu include dan juga extend.

1. Include

* Berfungsi untuk mengindentifikasi hubungan antara 2 use case, dimana use case yang satu akan memanggil use case yang lainnya.
* Apabila ada beberapa use case dengan aktifitas yang sama, maka bagian aktifitas tersebut. akan dijadikan aktifitas tersendiri, dengan relasi dependensi use case semula ke use case yang baru.
* Biasanya digambarkan dengan garis putus-putus dengan mata panah notasi include yang pada garis. Kemudian arahkan mata panah sesuai dengan arah yang memanggil.

1. Extend

* Apabila pemanggilan, memerlukan kondisi tertentu maka akan berlaku dependensi.
* Konsep extend ini berbeda dengan konsep extend dalam program Java.
* Dependensi ini akan digambarkan dengan dependensi incude namun dengan arah panah yang berlawanan.

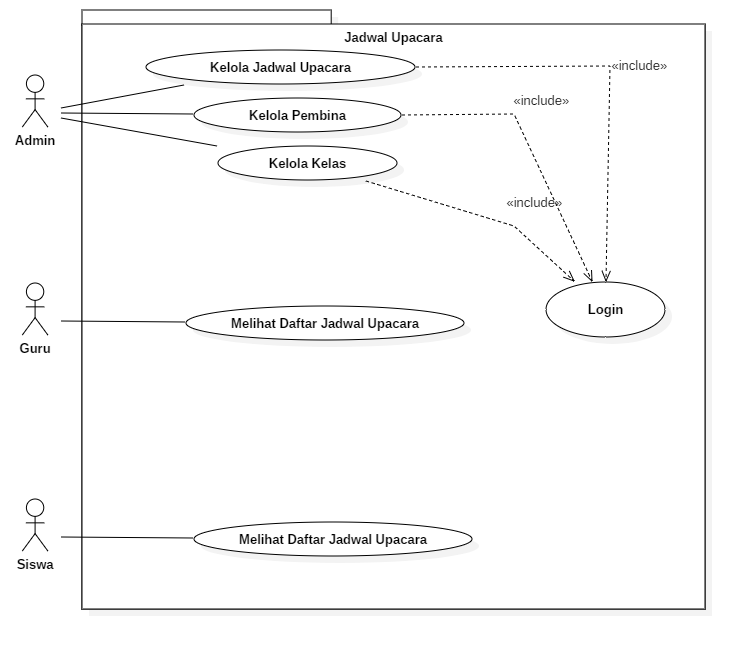
1. Aggregation

Aggregation ini adalah sebuah bentuk association yang maka elemen yang satu akan berisi elemen lainya.

1. Simbol Use Case Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
| Actor | Actor | Merupakan peran orang,  sistem yang lain, atau  alat ketika berhubungan  dengan use case. |
| Use Case | Use Case | Abstraksi dari penghubung  antara aktor dengan  use case. |
| Include | Association | Abstraksi dari penghubung  antara aktor dengan  use case. |
| Generalisasi | Generalisasi | Menunjukkan spesialisasi  aktor untuk dapat  berpartisipasi dengan  use case. |
| Include | Include | Menunjukkan bahwa  suatu use case seluruhnya  merupakan fungsionalitas  dari use case lainnya. |
| Include | Extend | Menunjukkan bahwa  suatu use case  merupakan tambahan  fungsinonal dari use case  lainnya jika suatu kondisi  terpenuhi. |

1. Contoh Use Case Diagram Sistem Penjadwalan Upacara



1. Aliran Kerja
2. Pengertian Activity Diagram

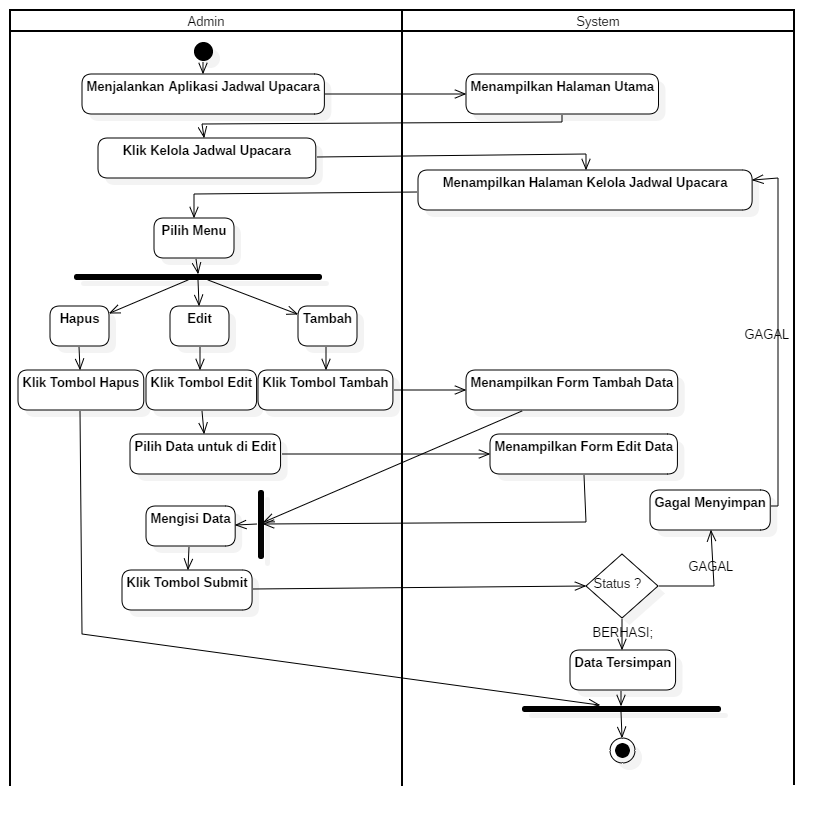
Pengertian activity diagram adalah pemodelan yang dilakukan pada suatu sistem dan menggambarkan aktivitas sistem berjalan. Activity diagram di gunakan sebagai penjelelasan aktivitas program tanpa melihat koding atau tampilan.

Activiry diagram di gambarkan dengan simbol-simbol yang setiap simbolnya memiliki makna dan tujuan. Aktivitas yang perlu diagram adalah sub sistemnya saja tidak perlu detai di dalamnya. Jika semua di buat maka akan sangat Panjang dan banyak.

1. Fungsi Activity Diagram
2. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.
3. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis.
4. Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktu.r
5. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.
6. Activity diagram dibuat berdasarkan sebuah atau beberapa use case pada use case diagram.
7. Elemen-Elemen Activity Diagram
8. Swimlanes, menunjukkan siapa yang bertanggung jawab melakukan aktivitas dalam suatu diagram.
9. Activities State, adalah kegiatan dalam aliran kerja.
10. Action State, adalah langkah-langkah dalam sebuah activity. Action bisa terjadi saat memasuki activity, meningggalkan activity, saat di dalam activity, atau pada event yang spesifik
11. Business object, adalah entitas-entitas yang digunakan dalam aliran kerja.
12. Transition, menunjukkan bagaimana aliran kerja itu berjalan dari satu aktivitas ke aktivitas lainnya.
13. Decision point, menunjukkan dimana sebuah keputusan perlu dibuat dalam aliran kerja.
14. Syncronization, menunjukkan dua atau lebih langkah dalam aliran kerja berjalan secara serentak à Fork dan Join.
15. Start state, menunjukkan dimana aliran kerja itu dimulai.
16. End state, menunjukkan dimana aliran kerja itu berakhir.
17. Simbol Activity Diagram

|  |  |
| --- | --- |
| Simbol Activity Diagram | Kegunaan |
| https://pelajarindo.com/wp-content/uploads/2018/09/Status-Awal.png | Status   awal   aktivitas   sistem,   sebuah diagram aktivitas memiliki sebuah status awal |
| https://pelajarindo.com/wp-content/uploads/2018/09/Aktivitas.png | Aktivitas yangdilakukan sistem,aktivitas  biasanya  diawali  dengan  kata kerja |
| https://pelajarindo.com/wp-content/uploads/2018/09/Percabangan.png | Asosiasi  percabangan  dimana  jika  ada pilihan aktivitas lebih dari satu |
| https://pelajarindo.com/wp-content/uploads/2018/09/Penggabungan.png | Asosiasi   penggabungan   dimana   lebih dari  satu  aktivitas digabungkan  menjadi satu |
| https://pelajarindo.com/wp-content/uploads/2018/09/Status-Akhir.png | Status   akhir   yang   dilakukan   sistem, sebuah    diagram    aktivitas    memiliki sebuah status akhir |
| https://pelajarindo.com/wp-content/uploads/2018/09/Swimlane.png | Swimlane memisahkan  organisasi  bisnis yang     bertanggung     jawab     terhadap aktivitas yang terjadi |

1. Contoh Activity Diagram Kelola Jadwal Upacara



## **Perancangan Antar Muka**

1. Pengertian Mock Up Desain Website

Salah satu kebutuhan yang penting di dalam penyampaian informasi adalah mock up desain website. Mock up merupakan ujung tombak pada presentasi desain. Adanya mockup desain website ini diperlukan guna meyakinkan Anda ketika akan membuat website dengan desain custom. Melalui mock up, Anda dapat melihat visualisasi calon website Anda secara nyata sehingga Anda dapat memberikan masukan kepada desainer jika tampilan mock up dirasa belum sesuai dengan permintaan sebelumnya. Oleh karena itu, tak heran jika mock up merupakan bagian penting dari pembuatan website dengan desain custom.

Mock up selain disebut sebagai visualisasi sebuah konsep desain, bisa juga disebut sebagai gambaran nyata rancangan produk, atau preview sebuah ide yang terlihat seperti wujud aslinya. Anda bisa melihat seperti apa preview dari rancangan halaman web yang akan dibuat. Hal ini akan persis seperti aslinya, hanya saja masih belum memakai aplikasi web dan ini dalam bentuk layout.

Mock up ini sendiri bisa dibuat dengan cara manual dengan mengandalkan kertas, cat warna, dan pena. Akan tetapi, juga bisa dibuat secara lebih modern dengan mengandalkan aplikasi seperti CorelDraw, Adobe Photoshop, Marcomedia Freehand, Adobe Illustrator, dan sebagainya. Bahkan, bisa juga memakai aplikasi khusus untuk mock up desain website.

Web designer yang sudah berpengalaman akan menampilkan gambaran visual seperti konsep asli desain web, sehingga ini akan memberi kemudahan untuk Anda. Anda tak akan kesulitan lagi untuk membayangkan seperti apa konsep halaman web yang Anda pesan dan inginkan.

1. Fungsi Mock Up Desain Web
2. Media presentasi proyek desain web.
3. Menampilkan konsep desain yang akan ditawarkan, seakan-akan nyata.
4. Pedoman teknis untuk merancang halaman web.
5. Kontrol agar display desain yang dibuat tak melenceng dari rencana awal.
6. Memudahkan Anda mendapat gambaran konsep web.