

BAB II

LANDASAN TEORI

2.1 Penjadwalan

Penjadwalan diperlukan ketika beberapa kegiatan harus diproses pada suatu waktu tertentu. Penjadwalan yang baik memaksimalkan efektivitas pemanfaatan setiap kegiatan yang ada, sehingga penjadwalan merupakan bagian yang penting dalam perencanaan dan pengendalian kegiatan.

Bedworth dan Bailey (2007), mengidentifikasi beberapa tujuan dari aktivitas penjadwalan, yaitu yang pertama adalah meningkatkan utilisasi penggunaan sumber daya, atau dengan kata lain mengurangi waktu tunggu dari sumber daya tersebut, sehingga total waktu proses dapat berkurang dapat meningkat. Kedua, mengurangi rata-rata jumlah kegiatan yang menunggu dalam antrian proses ketika sumber daya yang ada masih mengerjakan tugas lain, dan yang ketiga, mengurangi keterlambatan, yaitu menjamin pemenuhan waktu yang tersedia.

2.2 Aritmetika Modulo

Menurut Marsigit (2008), *Modulo* adalah sebuah operasi bilangan yang menghasilkan sisa pembagian dari suatu bilangan terhadap bilangan lainnya. Misalkan dua bilangan a dan b , a modulo b (disingkat $a \bmod b$) adalah bilangan bulat sisa pembagian a oleh b . Misalnya, " $1 \bmod 3$ ", " $4 \bmod 3$ ", dan " $7 \bmod 3$ " memiliki hasil 1, karena ketiga bilangan tersebut memiliki sisa 1 jika dibagi oleh 3, sedangkan " $9 \bmod 3$ " sama dengan 0.

2.3 Kongruensi Modulo

Menurut Rosen (2007), *Kongruensi modulo* dapat digunakan untuk menentukan sisa hasil pembagian bilangan berpangkat. Definisinya : Misalkan n sebuah bilangan bulat positif tertentu. Dua bilangan bulat a dan b dikatakan kongruen modulo n , ditulis : $a \equiv b \pmod{n}$, jika membagi selisih $a - b$, atau jika $a - b = kn$ untuk suatu bilangan bulat k . Dibaca juga, a kongruen dengan b modulo n , atau a kongruen modulo n dengan b . Dari definisi ini, kita bisa menentukan sisa hasil pembagian bilangan berpangkat.

Contoh: Tentukan sisa hasil dari 10^{99} dibagi 7!

Pembahasan :

Kita dapat membentuknya menjadi seperti berikut:

- $10^{99} \equiv x \pmod{7}$.

Mulailah dari bilangan terkecil diatas yang bersisa 1 (artinya $x = 1$) dibagi 7, yaitu 10^6 .

- $10^6 \equiv 1 \pmod{7}$.

Kedua ruas persamaan kongruensi modulo tersebut dipangkatkan 16 diperoleh $(10^6)^{16} \equiv 1^{16} \pmod{7}$.

- $10^{96} \equiv 1 \pmod{7}$.

Kedua ruas dikali 10^3 , diperoleh :

$$10^{96} \times 10^3 \equiv 1 \times 10^3 \pmod{7}$$

$$10^{99} \equiv 10^3 \pmod{7}$$

$$10^{99} \equiv 10 \times 10 \times 10 \pmod{7}$$

$$10^{99} \equiv 3 \times 3 \times 3 \pmod{7}$$

$$10^{99} \equiv 27 \pmod{7},$$

$10^{99} \equiv 6 \pmod{7}$, maka sisa dari 10^{99} dibagi 7 adalah 6.

2.4 Sistem Kompetisi *Round robin*

Menurut Harisman, dkk (2012), turnamen yang setiap tim bertanding dengan tim lainnya hanya sekali disebut Turnamen *Round robin* dan turnamen ini merupakan bentuk khusus dari turnamen reguler. Round-Robin sendiri banyak digunakan dalam suatu kompetisi dengan jumlah peserta yang banyak agar dapat langsung tersisih separuh dari jumlah peserta yang ada. Hal ini dilakukan untuk meminimalkan jumlah pertandingan yang akan dilaksanakan. Sistem kompetisi yang paling umum dipakai adalah sistem kompetisi penuh dan sistem setengah kompetisi. Dalam kompetisi penuh (dalam bahasa Inggris: *Double Round robin*), setiap peserta akan bertemu dengan peserta lainnya dua kali, biasanya satu pertemuan sebagai tuan rumah ("pertandingan kandang") dan satu pertemuan sebagai tamu ("pertandingan tandang"). Dalam sistem setengah kompetisi (*Round robin*), setiap peserta akan bertemu dengan semua peserta lainnya satu kali. Sistem kompetisi penuh dipakai dalam banyak kompetisi liga olahraga penting, seperti sepak bola dan bola basket. Sistem setengah kompetisi biasanya dipakai dalam suatu babak penyisihan suatu turnamen, yang sering kali dilanjutkan dengan sistem gugur.

Round robin menyusun/mengatur semua tim atau para pemain mengambil/memainkan giliran mereka bermain, yang akan menghasilkan pemenang dari turnamen yang telah diselenggarakan. Penerapan *Round robin* dalam penjadwalan pertandingan adalah sebagai berikut.

1. Jika n adalah banyaknya peserta, suatu kompetisi penuh akan memerlukan $n(n - 1)$ pertemuan dan setengah kompetisi memerlukan $\frac{n}{2}(n - 1)$ pertemuan.
2. Jika n genap, maka sebanyak $\frac{n}{2}$ pertemuan dapat diadakan secara paralel, asalkan tempat pertandingan mencukupi.
3. Jika banyaknya peserta ganjil, dapat ditambahkan satu "peserta bohong" (dummy), sehingga dalam setiap putaran ada satu peserta yang tidak bermain.

Diketahui bahwa jumlah *team* yang ikut serta adalah sebanyak N , dimana N adalah genap. Pada *Round* ke- R , team i akan melawan team j , maka berlaku ketentuan sebagai berikut :

$$i + j \equiv R \pmod{N-1} \dots\dots\dots(2)$$

Ketentuan di atas selalu berlaku kecuali jika $i = j$ (karena tidak mungkin ada team yang melawan dirinya sendiri) atau jika ada j yang sudah dipakai di petak dengan i lebih kecil (sebelumnya) (agar tidak terjadi bentrok). Jika ketentuan ini dilanggar, kosongkan dulu nilai j . Jika ternyata terdapat 2 nilai j yang memenuhi, maka ambillah nilai j yang lebih kecil.

Sebagai contoh terdapat 8 pemain/tim yang bermain pada suatu pertandingan. Jika diberlakukan kompetisi penuh, maka dengan menggunakan persamaan $n(n - 1)$ maka jumlah pertandingan (R) = $8 - 1 = 7$ *Round*/ pertandingan. Berikut ini contoh tabel penjadwalan untuk jumlah pemain $N = 8$.

Tabel 2.1. Contoh tabel penjadwalan

| R \ i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---------|---|---|---|---|---|---|---|---|
| Round 1 | 7 | 6 | 5 | A | 3 | 2 | 1 | B |
| Round 2 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Round 3 | 2 | 1 | 7 | 6 | | 4 | 3 | |
| Round 4 | 3 | C | 1 | 7 | 6 | 5 | 4 | D |
| Round 5 | 4 | 3 | 2 | 1 | 7 | | 5 | |
| Round 6 | 5 | 4 | | 2 | 1 | 7 | 6 | |
| Round 7 | 6 | 5 | 4 | 3 | 2 | 1 | 8 | 7 |

Berikut penjelasan sebagian darimana angka-angka itu didapat:

1. Lihat kotak dengan angka 7 di *Round 1*:

$R = 1$, dan $i = 1$, maka untuk menentukan lawan dari i dilakukan perhitungan

dengan persamaan : $(1+j) = 1 \pmod{7}$

Berapa nilai J agar hasilnya $(i+j-1) \pmod{7}$ hasilnya 0 ?

Nilai yang memungkinkan adalah $= 7$.

2. Lihat kotak dengan angka 5 di *Round 1*:

$R = 1$, dan $i = 3$, maka untuk menentukan lawan dari i dilakukan perhitungan

dengan persamaan : $(3+j) = 1 \pmod{7}$

Berapa nilai J agar hasilnya sama $(i+j-1) \pmod{7}$ hasilnya 0?

Jadi nilai j yang memungkinkan adalah $= 5$.

3. Lihat kotak kosong dengan Huruf A:

$R=1$, dan $i=4$, maka untuk menentukan lawan dari i dilakukan perhitungan dengan persamaan : $(4+j) = 1 \pmod{7}$

Berapa nilai J agar hasilnya sama dengan $(i+j-1) \pmod{7}$ hasilnya 0?

Jadi nilai j adalah $= 4$.

Namun, karena ketentuan tidak boleh $i = j$, maka kotak ini harus dikosongkan dahulu.

4. Lihat kotak kosong dengan Huruf B:

$R=1$, $i = 8$, maka untuk menentukan lawan dari i dilakukan perhitungan dengan persamaan : $(8+j) = 1 \pmod{7}$

Berapa nilai J agar hasilnya $(i+j-1) \pmod{7}$ hasilnya 0?

Nilai j yang memenuhi adalah $= 7$.

Namun, karena 7 sudah dipakai di petak sebelumnya ($R=1$, $i=1$), maka 7 tidak dapat digunakan lagi, maka kosongkan dulu petak ini. Untuk petak-petak yang lain, cara perhitungannya sama seperti contoh-contoh di atas. Setelah tabel setengah jadi di atas selesai, kita tinggal mengisi kotak-kotak yang masih kosong dengan nilai yang mungkin. Misalnya, karena di round 1, team 4 dan team 8 keduanya masih kosong, maka 4 akan melawan 8, dan 8 akan melawan 4. Cara yang sama juga kita lakukan untuk kotak kosong yang lain.

Tabel 2.2 Contoh tabel untuk pengisian kotak yang kosong

| R \ i | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|----------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Round 1 | 7 | 6 | 5 | 8 | 3 | 2 | 1 | 4 |
| Round 2 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| Round 3 | 2 | 1 | 7 | 6 | 8 | 4 | 3 | 5 |
| Round 4 | 3 | 8 | 1 | 7 | 6 | 5 | 4 | 2 |
| Round 5 | 4 | 3 | 2 | 1 | 7 | 8 | 5 | 6 |
| Round 6 | 5 | 4 | 8 | 2 | 1 | 7 | 6 | 3 |
| Round 7 | 6 | 5 | 4 | 3 | 2 | 1 | 8 | 7 |

2.5 Sistem Pertandingan

Pengertian sistem pertandingan menurut Kristiandaru (2010), adalah suatu mekanisme yang digunakan untuk menyelesaikan serangkaian kegiatan/proses dalam suatu pertandingan. Dalam kegiatan olahraga dikenal dengan dua sistem pertandingan yaitu sistem gugur dan sistem kompetisi. Untuk menentukan sistem apa yang akan digunakan untuk menyelesaikan serangkaian pertandingan, sangat dipengaruhi oleh beberapa hal diantaranya :

1. Jumlah hari yang tersedia
2. Jumlah peserta pertandingan
3. Fasilitas pertandingan yang tersedia

4. Biaya yang di sediakan
5. Jenis *event* olahraga yang diselenggarakan

2.5.1 Sistem Gugur (*Knock Out*)

Menurut Kristiandaru (2010), sistem gugur dalam suatu pertandingan olahraga merupakan sistem yang mengatur dimana setiap kali pertandingan harus ada pemenang untuk menentukan juara atau menentukan siapa dan posisi dimana dalam babak pertandingan selanjutnya, sistem gugur dikenal ada dua sistem, yaitu sistem gugur tunggal dan sistem gugur ganda.

Pengertian sistem pertandingan gugur tunggal adalah suatu sistem dalam pertandingan dimana peserta yang kalah sudah tidak lagi memiliki kesempatan untuk melanjutkan pertandingan berikutnya. Berbeda dengan sistem pertandingan gugur ganda, pengertian sistem pertandingan gugur ganda yaitu sistem dalam pertandingan dimana jika ada peserta yang satu kali, masih diberi kesempatan melakukan pertandingan selanjutnya dalam kelompok yang berbeda, dan jika mengalami kekalahan yang kedua kalinya, maka baginya secara otomatis akan gugur dan tidak ada kesempatan lagi dalam pertandingan berikutnya.

2.5.2 Sistem Kompetisi

Menurut Kristiandaru (2010), sistem kompetisi merupakan sistem dalam suatu pertandingan yang memberikan kesempatan para peserta untuk melakukan pertandingan lebih dari satu kali, sehingga setiap peserta tidak bertemu satu sama lain. dalam sistem ini juara di tentukan melalui perhitungan dari pengumpulan nilai yang tertinggi. Sistem kompetisi dikenal dua sistem yaitu sistem setengah kompetisi dan kompetisi penuh.

Sistem setengah kompetisi mempunyai pengertian sistem pertandingan yang semua peserta saling bertemu dalam satu kali pertandingan. Adapun sistem pertandingan penuh adalah sistem pertandingan yang semua peserta saling bertemu dalam dua kali pertandingan atau dapat di artikan satu tuan rumah dan sekali kandang, maka dari itu disebut *home and away system*.

2.6 Bola Basket

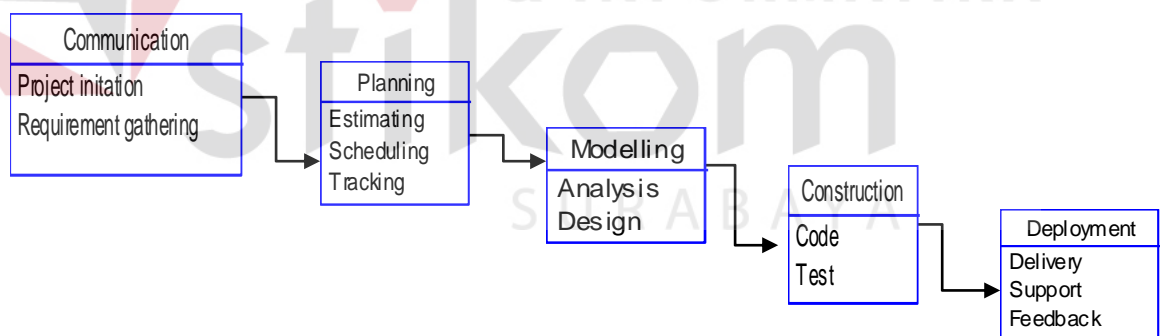
Menurut Rahmani (2014), bola basket adalah olahraga yang menggunakan bola dan dimainkan secara beregu. Bola basket dimainkan oleh 5 orang pemain. Bola basket sangat cocok untuk ditonton karena biasa dimainkan di ruang olahraga tertutup dan hanya memerlukan lapangan yang relatif kecil. Selain itu, permainan bola basket juga lebih kompetitif karena tempo permainan cenderung lebih cepat jika dibandingkan dengan olahraga bola yang lain, seperti voli dan sepak bola. Ada 3 posisi utama dalam bermain basket, yaitu : 1) *Forward*, pemain yang tugas utamanya adalah mencetak poin dengan memasukkan bola ke keranjang lawan, 2) *Defense*, pemain yang tugas utamanya adalah menjaga pemain lawan agar pemain lawan kesulitan memasukkan bola, dan 3) *Playmaker*, pemain yang menjadi tokoh kunci permainan dengan mengatur alur bola dan strategi yang dimainkan oleh rekan-rekan setimnya.

Jumlah pemain dalam permainan bola basket adalah 5 orang dalam satu regu dengan cadangan 5 orang. Sedangkan jumlah wasit dalam permainan bola basket adalah 2 orang. Wasit 1 disebut *Referee* sedangkan wasit 2 disebut *Umpire*. Waktu permainan 4 x 12 menit jika berpedoman dengan aturan *NBA*. Diantara babak 1, 2, 3, dan babak 4 terdapat waktu istirahat selama 10 menit. Bila terjadi skor yang sama pada akhir pertandingan harus diadakan perpanjangan waktu

sampai terjadi selisih skor. Di antara dua babak tambahan terdapat waktu istirahat selama 2 menit. Waktu untuk lemparan ke dalam yaitu 5 detik.

2.7 Sistem Development Life Cycle (SDLC)

Menurut Pressman (2015), *Model system Development Life Cycle (SDLC)* ini biasa disebut juga dengan model *waterfall* atau disebut juga *classic life cycle*. Adapun pengertian dari SDLC ini adalah suatu pendekatan yang sistematis dan berurutan. Tahapan-tahapannya adalah *Communication*, perencanaan (*planning*), pemodelan (*modelling*), konstruksi (*construction*), serta penyerahan *system* perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan. *Model system Development Life Cycle (SDLC)* ditunjukkan pada gambar 1 berikut.



Gambar 2.1. Pengembangan menggunakan Model *Waterfall* (Pressman, 2015)

Penjelasan-penjelasan SDLC Model *Waterfall*, adalah sebagai berikut:

a. Communication

Langkah pertama diawali dengan komunikasi kepada konsumen. Langkah awal ini merupakan langkah penting karena menyangkut penumpulan informasi tentang apa kebutuhan konsumen.

b. Planning

Setelah proses *communication*, kita menetapkan rencana untuk pengerjaan *software* yang meliputi tugas-tugas teknis yang akan dilakukan, resiko yang mungkin terjadi, sumber-sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. Modelling

Pada proses modeling ini menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, *representasi interface*, dan detail (algoritma) *procedural*. Tahapan ini menghasilkan dokumen yang disebut *software requirement*.

d. Construction

Construction merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemah desain dalam bahasa yang bisa dikenali oleh *computer*.

Programmer akan menerjemahkan transaksi yang diminta oleh *user*.

Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software* artinya penggunaan *computer* akan dimaksimalkan dalam tahapan ini.

Setelah pengkodean selesai maka akan dilakukan *testing* terhadap *system* yang

telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap *system* tersebut untuk kemudian bisa diperbaiki.

e. Deployment

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau *system*. Setelah melakukan analisis, desain dan pengkodean maka *system* yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

2.8 Black Box Testing

Menurut Black dan Champion (2009), *Tester* menggunakan *behavioral test* (disebut juga *Black Box Test*), sering digunakan untuk menemukan *bug* dalam *high level operations*, pada tingkatan fitur, profil operasional dan skenario *customer*. *Tester* dapat membuat pengujian fungsional *blackbox* berdasarkan pada apa yang harus sistem lakukan. *Behavioral testing* melibatkan pemahaman rinci mengenai *domain* aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem. *Behavioral test* paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan *bug* umum untuk jenis desain.

Menurut Nidhra dan Dondeti (2012), *black box testing* juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi.