BAB 2
LANDASAN TEORI

2.1 Budidaya Tanaman Padi

2.1.1 Botani Tanaman Padi

Tanaman padi *(Oryza sativa*) adalah tanaman pangan berupa rerumputan yang tersusun dari beberapa ruas. Tanaman padi merupakan jenis tanaman merumpun, yang dalam waktu singkat 1 batang bibit padi dapat membentuk rumpun sejumlah 20 sampai 30 anakan. Tanaman padi pada umumnya memiliki 3 fase pertumbuhan, yaitu fase *vegetatif*, fase *reprodukif*, dan fase pematangan.

Fase *vegetatif* meliputi pertumbuhan tanaman dari mulai berkecambah sampai dengan inisiasi *primordial* malai. Fase *reproduktif* dimulai dari inisiasi *primordia* malai sampai berbunga (*heading*) dan pematangan dimulai dari berbunga sampai masak panen.

Berdasarkan iklim padi tumbuh di daerah tropis/*subtropics* pada 45 derajat LU sampai 45 derajat LS dengan cuaca panas dan kelembaban tinggi dengan musim hujan 4 bulan. Rata-rata curah hujan yang baik adalah 200 mm/bulan atau 1500-2000 mm/tahun. Padi dapat di tanam dimusim kemarau atau hujan, pada musim kemarau produksi padi meningkat asalkan air irigasi selalu tersedia, sedangkan musim hujan walaupun air melimpah produksi dapat menurun karena penyerbukan kurang intensif [9].

2.1.2 Pedoman Budi Daya Padi Sawah

Dalam pembudidayaan padi sawah mencakup proses pembibitan, pemindahan atau penanaman, pemiliharaan (termasuk perlindungan tanaman dan pemupukan), dan panen. Aspek lain yang penting namun bukan termasuk dalam rangkaian bercocok tanam padi sawah adalah proses penyimpanan biji benih.

2.1.2.1 Pembibitan

Dalam melakukan pembibitan terdapat tahap-tahap yang perlu diperhatikan, seperti menentukan benih yang baik sampai dengan melakukan perendaman benih

guna mendapatkan benih yang baik untuk dilakukan pembibitan/persemaian. Berikut adalah tahapan dalam pembibitan padi sawah [10].

1. Persyaratan Benih

Syarat benih yang baik yaitu yang tidak mengandung gabah hampa, potongan jerami, kerikil, tanah dan hama gudang. Warna gabah sesuai dengan aslinya serta daya perkecambahan 80%.

2. Persiapan Benih

Untuk melakukan pembibitan biji benih terlebih dahulu dilakukan perendaman di air mengalir dan dimasukkan kedalam karung goni selama satu malam supaya perkecambahan benih bersamaan.

3. Penyemaian Benih

Untuk satu hektar padi sawah diperlukan 25-40 kg benih, tergantung pada jenis padinya. Lahan persemaian sebaiknya dipersiapkan 50 hari sebelum semai dengan luas kira-kira 1/20 dari area sawah yang akan ditanami.

4. Pemindahan Benih

Bibit yang siap dipindahtanamkan ke sawah berumur sekitar 25-40 hari, berdaun 5-7 helai.

2.1.2.2 Pengolahan Media Tanam

Sebelum melakukan penanaman perlu dilakukan pembajakan sawah dengan membalik tanah dan memasukkan bahan organik yang ada dipermukaan. Pembajakan pertama dilakukan pada awal musim tanam dan dibiarkan 2-3 hari. Setelah itu dilakukan pembajakan kedua yang disusul oleh pembajakan ketiga 3-5 hari menjelang proses penanaman [10].

2.1.2.3 Penanaman

Bibit ditanam dalam larikan dengan jarak tanam 20x20 cm, 25x25 cm, 22x22 cm atau 30x20 cm tergantung pada varitas padi, kesuburan tanah dan musim. Padi dengan jumlah anakan yang banyak memerlukan jarak tanam yang lebih lebar. Pada tanah subur jarak tanam lebih lebar. Jarak tanam di daerah pegunungan lebih rapat karena bibit tumbuh lebih lambat. 2-3 batang bibit ditanam pada kedalaman 3-4 cm [10].

2.1.2.4 Pemeliharaan

1. Pemupukan

Pemupukan padi dilakukan sebanyak tiga kali, pemupukan pertama disebut pupuk dasar. Sewaktu bibit pindah tanam pembibitan perlu dilakukan pada waktu sekitar 8-12 hari setelah tanam. Pemupukan susulan ke satu diberikan sekitar pekan ketiga atau sekitar 21-25 hari setelah tanam, kemudian disusul dengan pemupukan susulan kedua yang diberikan pada umur tanaman mencapai pekan ketiga atau sekitar 30-40 hari setelah tanam [11].

2. Membasmi Hama

Berikut adalah jenis-jenis hama yang sering menyerang padi pada pertaninan di Indonesia [10].

a. Ulat Tentara/Grayak

Ulat grayak menyerang tanaman padi pada semua stadia. Serangan terjadi biasanya pada malam hari sedangkan siang harinya larva ulat grayak bersembunyi pada pangkal tanaman, dalam tanah atau di tempat tempat yang tersembunyi. Seranga ulat ini memakan helai-helai daun dimulai dari ujung daun dan tulang daun utama ditinggalkan sehingga tinggal tanaman padi tanpa helai daun. Pada tanaman yang telah membentuk malai, ulat grayak seringkali memotong tangkai malai, bahkan ulat grayak ini juga menyerang padi yang sudah mulai menguning. Batang padi yang mulai menguning itu membusuk dan mati yang akhirnya menyebabkan kegagalan panen. Serangan saat padi menguning atau keluar malai inilah yang sangat merugikan petani. Gambaran bentuk ulat tentara/grayak dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Ulat Tentara/Grayak

b. Hama Penggerek Batang

Penggerek batang adalah ulat yang hidup dalam batang padi. Hama ini berubah menjadi ngengat berwarna kuning atau coklat. Biasanya 1 larva berada dalam 1 anakan. Ngengat aktif di malam hari. Larva betina menaruh 3 massa telur sepanjang 7-10 hari masa hidupnya sebagai organisme dewasa. Massa telur dari penggerek batang kuning berbentuk cakram dan ditutupi oleh bulu-bulu berwarna coklat terang dari *abdomen* betina. Setiap massa telur mengandung sekitar 100 telur. Penggerek batang padi dapat merusak pertanaman padi pada stadia apa saja, sejak persemaian sampai matang. Bila tanaman masih muda, daun-daun yang di tengah dari anakan yang rusak berwarna coklat dan mati. Kondisi ini disebut sundep. Bila kerusakan timbul setelah terbentuknya malai, maka malai berwarna putih dan disebut beluk. Gambaran bentuk hama penggerek batang yang ada di pertanian dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Hama Penggerek Batang

c. Walang Sangit

Walang sangit merupakan hama yang menghisap cairan bulir pada fase masak susu. Kerusakan yang ditimbulkan walang sangit menyebabkan beras berubah warna, mengapur serta hampa. Hal ini dikarenakan walang sangit menghisap cairan dalam bulir padi. Fase tanaman padi yang rentan terserang hama walang sangit adalah saat tanaman padi mulai keluar malai sampai fase masak susu. Gambaran walang sangit dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Walang Sangit

d. Wereng Coklat

Hama ini dapat menyebabkan tanaman padi mati kering dan tampak seperti terbakar atau puso, serta dapat menularkan beberapa jenis penyakit. Tanaman padi yang rentan terserang wereng coklat adalah tanaman padi yang dipupuk dengan unsur *N* terlalu tinggi dan jarak tanam yang merupakan kondisi yang disenangi wereng coklat.

Hama wereng coklat menyerang tanaman pada mulai dari pembibitan hingga fase masak susu. Gejala serangan adalah terdapatnya *imago* wereng coklat pada tanaman dan menghisap cairan tanaman pada pangkal batang, kemudian tanaman menjadi menguning dan mengering. Gambaran wereng coklat yang ada di pertanian dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Wereng Coklat

e. Tikus

Tikus merusak tanaman pada semua fase pertumbuhan dan dapat menyebabkan kerusakan besar apabila tikus menyerang pada saat primodia. Tikus akan memotong titik tumbuh atau memotong pangkal batang untuk memakan bulir gabah. Tikus menyerang pada malam hari dan pada siang hari tikus bersembunyi di lubang pada tanggul irigasi, pematang sawah, pekarangan, semak atau gulma. Gambaran tikus yang sering menyerang padi dapat dilihat pada Gambar 2.5.



Gambar 2. 5 Tikus Sawah

2.1.2.5 Panen

Padi siap panen yaitu padi dengan 95% bulir yang telah menguning atau sekitar 33-36 hari setelah berbunga, bagian malai masih terdapat gabah hijau dan kadar air gabah 21-26 % serta bulir hijau rendah. Cara memanen padi yaitu dengan menggunakan sabit tajam untuk memotong pangkal batang [10].

2.2 *Game*

Video games merupakan salah satu sarana hiburan yang saat ini mengalami kemajuan yang sangat pesat. Hal ini ditandai dengan banyaknya perangkat dan jenis *game* yang beredar dimasyarakat. Saat ini *video games* tidak lagi dipandang sebagai sarana hiburan bagi kalangan anak-anak saja, tetapi *video games* lebih dikenal sebagai sarana hiburan seluruh kalangan usia.

2.2.1 Pengertian *Game*

Game diambil dari bahasa Inggris yang berarti permainan. Aktivitas yang dilakukan dalam permainan biasanya melibatkan pemain (*player*) didalamnya guna untuk menentukan keputusan dan alur dari permainan tersebut. Pada perkembangan teknologi saat ini kata *game* sering digunakan untuk menyebutkan kata *video game* yang merupakan *game* atau permainan dengan interaksi utamanya melibatkan *video* serta *audio*.

2.2.2 Elemen Dasar *Game*

Dalam suatu *game* terdapat beberapa elemen yang menyusun *game* tersebut, seperti jenis *game*, karakter *game, background*, elemen *sound*/suara, dan gerakan-gerakan dasar.

2.2.3 *Genre Game*

Adapun *genre game* dapat diklasifikasikan menjadi beberapa tipe antara lain adalah [12] :

1. *Action games*

   Game ini dikenali dari aksi didalamnya yang berupa pergerakan, *attack*, reaksi dan pergerakan-pergerakan lainnya. Hal yang lebih ditekankan pada *game* jenis ini adalah aksi didalamnya, bukan dari sudut pandang cerita.

2. *Adventure*

   Dalam *game* tipe ini, *player* akan menyelesaikan misi yang diberikan sepanjang petualangan di dalam *game*. *Player* memulai permainan dengan inventori yang terbatas, senjata dan makanan, kemudian dikirim untuk melakukan suatu *quest*, dan dalam perjalanannya ada tugas yang harus diselesaikan.

3. *Casual Game*

   Yang termasuk dalam *casual game* adalah *board game, card game* dan *game shows*. *Game* ini bertujuan untuk menghibur *player*, dan selain itu dapat juga digunakan untuk melatih, mempelajari dan meningkatkan pengetahuan dan *level skill gamer.*

4. *Educational Game*

   Game edukasi memberikan penekanan pada unsur pendidikan dan pembelajaran. Contoh *game* edukasi yang sederhana dapat berupa pertanyaan pilihan ganda, isian maupun essai.

5. *Role-Playing Game* (RPG)

   Pada *game* bertipe ini memiliki dunia yang luas untuk dieksplorasi oleh *player*, baik untuk mencari harta karun, objek dan status (*leveling*), sambil menghancurkan monster dan objek yang menghalaginya. Dalam RPG ada tujuan yang harus dicapai *player* untuk menyelesaikan permainan.

6. *Strategy*

   Dalam *game* tipe ini dibutuhkan pemikiran dan perencanaan yang benar untuk dapat memenangkan permainan dan biasanya menggunakan sudut pandang *overhead* sehingga *player* dapat melihat seluruh area permainan.

7. *Simulation*

   Simulation game atau *sims* adalah suatu bentuk *game* yang menggambarkan situasi dunia nyata, baik dalam rupa simulasi *driving* maupun dalam hubungannya dengan lingkungan interaksi antara manusia.

8. *Puzzle*

   Puzzle game memiliki tujuan yang sangat sederhana yaitu untuk menyelesaikan *puzzle* yang diberikan.

2.3 Multimedia

Multimedia adalah sebuah fungsi dari komputer yang mempresentasikan dan menggabungkan teks, grafik, suara, *video*, dan animasi sehingga pengguna dapat berinteraksi, menciptakan, berkomunikasi. Definisi tersebut mendefinisikan empat komponen yang harus ada pada multimedia [14].

1. Adanya sebuah komputer untuk mengkoordinasi apa yang bisa dilihat, didengar, dan yang dapat berinteraksi dengan penggunanya.
2. terdapatnya sebuah hubungan ke informasi.
3. terdapat alat navigasi yang memungkinkan pengguna mengakses informasi tersebut.

4. karena multimedia bukan hanya menyaksikan, maka harus ada cara untuk memperoleh, memproses, dan berkomunikasi dengan informasi dan ide.

2.3.1. Elemen Multimedia

Adapun elemen-elemen multimedia adalah sebagai berikut [14].

1. Teks

meskipun dimungkinkan untuk menciptakan sebuah multimedia tanpa teks, tetapi kebanyakan sistem multimedia menggunakan teks karena adalah cara efektif untuk mengkomunikasikan ide-ide dan memberikan petunjuk kepada pengguna. pada multimedia ada empat macam teks yaitu: *printed text*, *scanned text*, *electoric text*, dan *hypertext*.

2. Grafik

Multimedia memungkinkan pengguna untuk melakukan merubah gambar menjadi objek atau *link*. Grafik seringkali muncul sebagai latar dibalik suatu teks. Selain itu gambar juga dapat berfungsi sebagai ikon yang digabungkan dengan teks dan menampilkan pilihan-pilihan, atau dapat ditampilakn secara *fullscreen* untuk menggantikan teks. Grafik terdiri dari beberapa bentuk antara lain: *bitmap*, *vector images*, *clip art*, *digitized pictures*, dan *hyperpictures*.

3. Suara

Suara atau *sound* yang sering digunakan dalam multimedia adalah *waveform audio*, *midi*, *audio CD*, *CD plus*, *Mp3*, dan *hyperaudio*

4. *Video*

Video menyediakan sumber daya yang besar bagi multimedia. terdapat empat jenis *video* yang dapat digunakan pengguna sebagai objek *link* dalam multimedia yaitu: *live video feeds*, *video tape*, *video disc*, dan *digital video*.

5. Animasi

Dalam multimedia, animasi merupakan suatu teknik penggunaan komputer untuk menciptakan pergerakan dilayar. Adapaun animasi yang terdapat dalam multimedia adalah sebagai berikut:

a. *Frame Animation*

   Frame Animation membuat objek bergerak dengan menampilkan serangkaian gambar yang disebut *frame*, dimana objek yang tampilkan bebeda-beda.

b. *Vector Animation*

   Sebuah vektor adalah garis yang memiliki awal, arah, dan panjang. *vector animation* membuat objek bergerak dengan merubah titik awal, arah dan ukuran dari objekdan disuaikan dengan segmen objek.

c. *Computional Animation*

   Computional animation adalah teknik merubah objek untuk dapat bergerak dengan merubah koordinat *x* dan *y* pada layar.

d. *Morphing*

   Morphing berarti transisi suatu bentuk ke bentuk lain dengan menampilkan serangkaian *frame* untuk menciptakan gerakan yang halus.

2.4 AI (*Artificial Intelligence*)

Istilah *intelligence* atau kecerdasan cukup samar-samar. Didalam kamus penjelasan bahwa *intellingence* adalah kapasitas untuk memperoleh dan menerapkan pengetahuan, tetapi definisi tersebut terlalu umum. Karena tidak ada definisi yang lebih baik, *intelligence* dapat diartikan sebagai seperangkat keterampilan yang memungkinkan manusia untuk memecahkan masalah dengan sumber daya yang terbatas. Keterampilan seperti belajar, berpikir abstrak, perencanaan, imajinasi, dan kreativitas mencakup aspek yang paling penting dari kecerdasan manusia.

Pada saat bermain *game* hal yang penting bukan hanya sekedar tentang kekuatan tangan atau reflek yang sempurna, tetapi bagaimana menjadikan *game* sebagai sarana hiburan dan dapat memecahkan masalah pada *game* tersebut. *Game* yang menggunakan konsep *artificial intelligence* merupakan mesin pencari solusi untuk memecahkan masalah dengan cara yang lebih mirip manusia.

Menurut Stuart J. Russell dan Peter Norvig [15], definisi AI dibagi kedalam empat kategori berdasarkan beberapa *textbook* berbeda, yaitu:

1. ***Acting humanly***: *The Turing Test approach*

Pada tahun 1950, Alan Turing mendefinisikan prilaku *intelligence* (kecerdasan) sebagai kemampuan untuk mencapai *level* performa manusia dalam semua tugas yang berkaitan dengan teori, untuk dapat mengelabui pemakainya yang menginterogasinya melalui *teletype* (komunikasi berbasis teks jarak jauh). Jika *intelligence* tidak dapat membedakan yang tidak dapat membedakan yang diinterogasi adalah manusia atau komputer, maka komputer tersebut lolos dari *turing test*. Untuk dapat melakukan pemeriksaan yang benar komputer tersebut harus memiliki kemampuan berikut :

a. *Natural Language Processing*
b. *Knowledge Representation*
c. *Automated Reasoning*
d. *Machine Learning*
e. *Computer Vision*
f. *Robotics*

2. ***Thinking humanly***: *The cognitive modelling approach*

Untuk membuat program yang memilki pola pikir seperti manusia, harus mengetahui dulu bagaimana cara manusia berpikir. Untuk melakukan pendekatan ini dapat dilakukan dengan dua hal yaitu, melalui introspeksi dengan cara mencoba menangkap pemikiran-pemikiran yang biasanya dilakukan manusia, dan dengan cara eksprerimen-eksprerimen psikologi.

3. ***Thinking rationally***: *The laws of thought approach*

Dalam melakukan pendekatan ini terdapat dua masalah utama yang sering dihadapi. Masalah pertama yang timbul adalah untuk membuat pengetahuan informal yang menyatakan pengetahuan tersebut ke dalam *formal term* yang diperlukan oleh notasi logika tidaklah mudah. Kemudian yang kedua terdapat perbedaan besar antara dapat memecahkan masalah kode dan dalam dunia nyata.

4. ***Acting rationally***: *The rational agent approach*

Acting rationally adalah bertindak untuk mencapai suatu tujuan yang dipercayai. Untuk membuat inferensi yang logis merupakan bagian dari suatu *rational agent*, hal ini disebabkan satu-satunya cara untuk melakukan aksi secara

rasional adalah dengan menalar secara logis. Dengan menalar secara logis, maka bisa didapatkan kesimpulan bahwa aksi yang diberikan akan mencapai tujuan atau tidak. Jika mencapai tujuan, maka *agent* dapat melakukan aksi berdasarkan kesimpulan tersebut.

2.5 Algoritma A* (*A Star*)

2.5.1 Deskripsi Algortima A* (*A Star*)

Algortima A* (*A Star*) identik dengan *pathfinding* dalam *game*. Algoritma ini mudah diimplementasikan dan merupakan perbaikan dari metode BFS dengan memodifikasi fungsi *heuristic* nya. Algoritma A* membuang langkah-langkah yang tidak perlu dengan pertimbangan bahwa langkah-langkah yang dibuang sudah pasti merupakan langkah yang tidak akan mencapai solusi yang diinginkan.

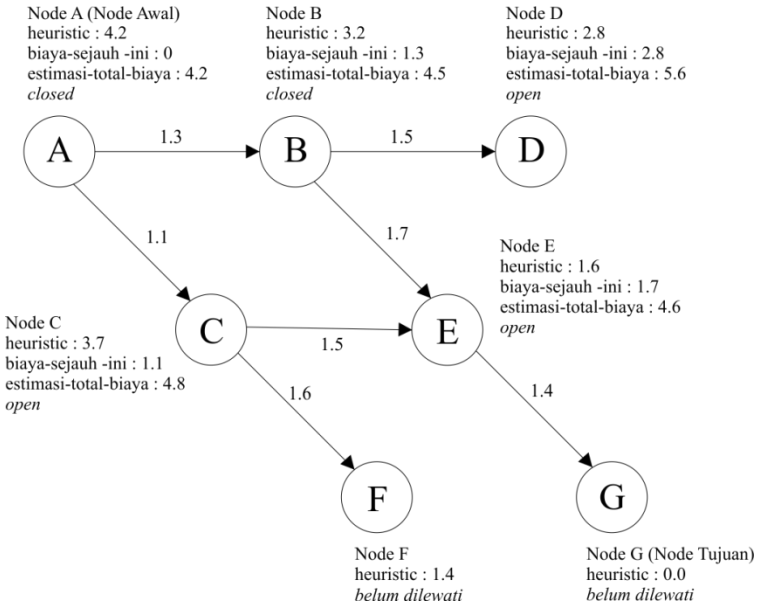
Ada beberapa terminologi dasar yang terdapat pada algoritma ini diantaranya [16]:

1. ***Starting Point*:** *Starting point* adalah sebuah terminologi untuk posisi awal sebuah benda.
2. ***N*:** Adalah posisi *node* yang sedang dijalankan dalam algoritma *pathfinding*.
3. **Simpul (*Nodes*) :** *Node* adalah petak-petak kecil sebagai representasi dari area *pathfinding*. Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.
4. ***Open List*:** *Open List* adalah tempat menyimpan data *node* yang mungkin diakses dari *starting point* maupun *node* yang sedang dijalankan.
5. ***Closed List*:** *Closed List* adalah tempat menyimpan data *node* sebelum *N* yang juga merupakan bagian dari jalur terpendek yang telah berhasil didapatkan.
6. **Harga (*Cost*):** Harga (*cost*) adalah nilai yang diperhitungkan oleh algoritma untuk menemukan jalur terdekat dengan *node* tujuan.
7. **Rintangan (*Unwalkable*):** Rintangan adalah sebuah atribut yang menyatakan bahawa sebuah *node* tidak dapat dilalui oleh *N*

2.5.2 Cara Kerja Algoritma A* (*A Star*)

Prinsip kerja algoritma A* adalah mencari *node open list* yang memiliki nilai *biaya-sejauh-ini* terkecil dari *node* awal. Algoritma akan memilih *node* yang paling memungkinkan untuk sampai ke tujuan dengan jarak terpendek. Untuk menentukan *node* yang paling memungkinkan jadi jalur terpendek diperlukan suatu nilai *heuristic* (nilai *biaya-seberapa-jauh* jarak dari suatu *node* ke *node* tujuan). Keakuratan pencarian pada algoritma tergantung dari keakuratan nilai *heuristic* nya. jika nilai *heuristic* nya akurat maka pencarian akan efesien. Secara rinci algoritma akan bekerja dalam iterasi-iterasi (perulangan). Pada setiap iterasi, algoritma akan melakukan perhitungan dari satu *node* ke *node* berikutnya.

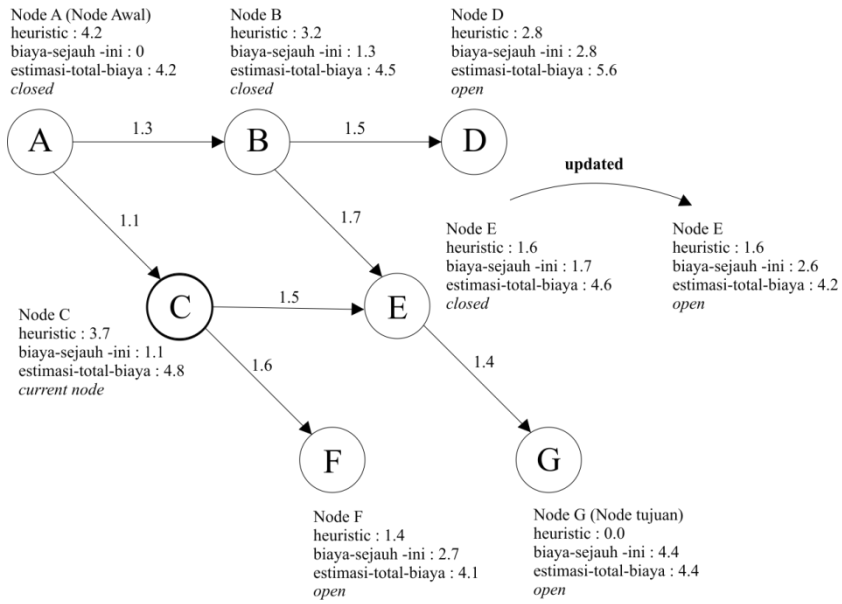
Selama iterasi, algoritma menganggap setiap relasi yang keluar adalah berasal dari *current node*. Setiap relasi yang keluar bertujuan untuk mencari *node* akhir dan menyimpan *total* biaya dari perjalanan tersebut. Selain itu algoritma akan menyimpan biaya lainnya, yaitu *biaya-sejauh-ini* (biaya dari *node* awal ke *current node*) dan *biaya-seberapa-jauh* (biaya dari *current node* ke *node* tujuan) atau lebih dikenal dengan *heuristic*. Kedua nilai biaya tersebut bukanlah hasil perhitungan dari algoritma A*, tetapi ditentukan secara manual dan nilai biaya tersebut tidak boleh bernilai negatif. Untuk lebih jelasnya dapat dilihat pada Gambar 2.6.



Gambar 2. 6 Estimasi Total Biaya Algoritma *A Star* [17]

Algoritma A* akan tetap memberi label *open list* ke *node* yang telah di cek tetapi tidak dijadikan perlintasannya karena *estimasi-total-biaya* nya lebih besar dari yang lainnya dan akan memberi label *closed list* kepada *node* yang menjadi yang memiliki *estimasi-total-biaya* terkecil, hal tersebut memungkinkan algoritma untuk memeriksa *node* yang lebih menjanjikan untuk dijadikan perlintasan selanjutnya. Untuk mendapatkan perkiraan biaya terkecil maka algoritma harus mempunyai *biaya-sejauh-ini* dan nilai *heuristic* terkecil. jika perkiraan biaya tersebut tepat, maka *node* yang lebih dekat dengan *node* tujuan akan diprioritaskan.

Dengan adanya pemberian label *open list* kembali kepada *node* yang tidak jadi perlintasan pertama maka akan sangat menguntungkan, karena algoritma dapat memperbaharui nilai yang telah dicacatnya. Bila terjadi kasus dimana nilai yang baru lebih kecil dari nilai yang sekarang, maka algoritma perlu memperbaharuinya. Algoritma akan melakukan perbandingan pada *biaya-sejauh-ini*, dikarenakan *biaya-sejauh-ini* bukanlah hasil dari perhitungan algoritma, maka nilai dari *biaya-sejauh-ini* dapat dijadikan patokan. Untuk lebih jelasnya dapat dilihat pada Gambar 2.7.

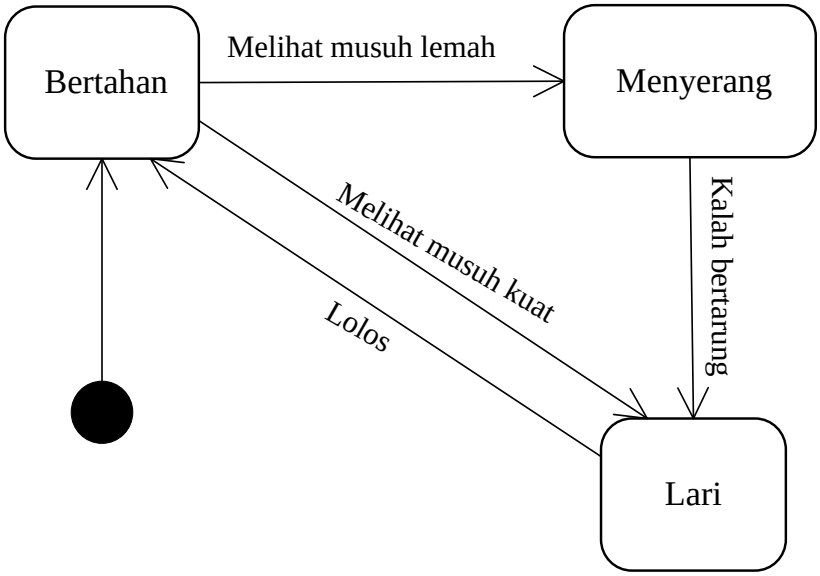


Gambar 2. 7 *Closed Node Update* Algoritma *A Star* [17]

2.6 *Finite State Machines*

Finate State Machines (FSM) adalah sebuah metodologi perancangan sistem kontrol yang menggambarkan tingkah laku atau prinsip kerja sistem dengan menggunakan *state* (Keadaan), *event* (kejadian) dan *action* (aksi) [17].

Dalam *state machine* sistem menempati satu *state* (keadaan). Sistem akan beralih atau bertransisi menuju ke *state* lain jika mendapatkan masukan *event* tertentu. Sistem akan tetap melakukan aksi yang sama pada suatu *state* sampai sistem menerima *event* tertentu baik yang berasal dari perangkat luar atau komponen dari sistem itu sendiri. Setiap *state* terhubung oleh transisi dan setiap transisinya mengarah ke satu *state* lainnya. Transisi keaadan ini umumnya juga disertai oleh aksi yang dilakukan oleh sistem ketika menanggapi masukan yang terjadi. Aksi yang dilakukan tersebut dapat berupa aksi sederhana yang melibatkan rangkaian proses yang relatif rumit. Berikut adalah gambaran *state machines* sederhana.



Gambar 2. 8 *State Machine* Sederhana [17]

Gambar 2.8 menunjukkan *state machine* sederhana yang terdiri dari *state* bertahan, menyerang, dan lari dimana setiap *state* memiliki masing-masing transisi.

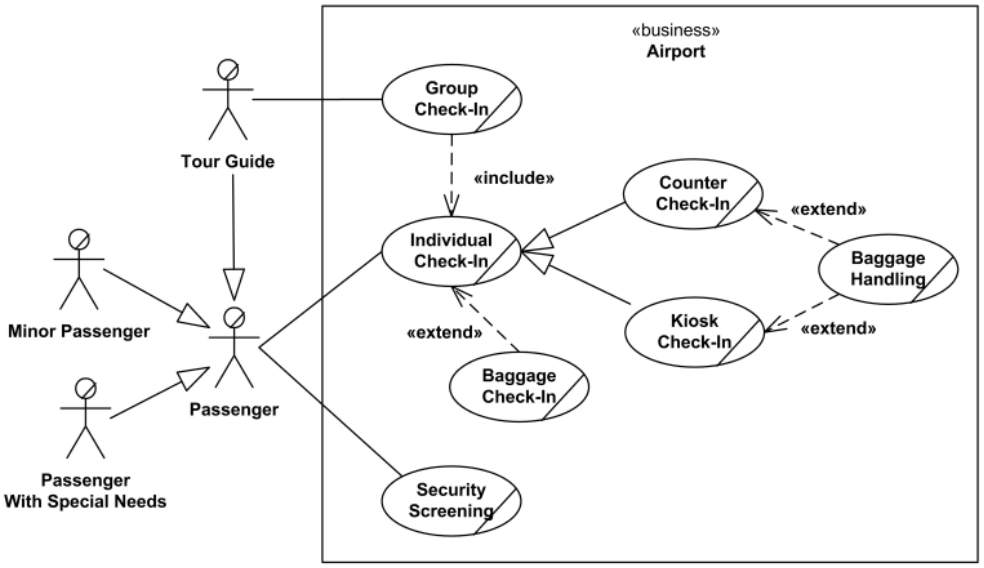
2.7 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami [19]. UML tidak hanya merupakan sebuah bahasa pemograman *visual* saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti *JAVA*, *C++*, *Visual Basic*, atau bahkan dihubungkan secara langsung ke dalam sebuah *object-oriented database*. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, *arsitektur*, *design*, *source code*, *project plan*, *tests*, dan *prototypes*.

Berikut adalah contoh beberapa UML yang digunakan untuk memodelkan aplikasi berorientasi Objek.

1. ***Use Case Diagram***

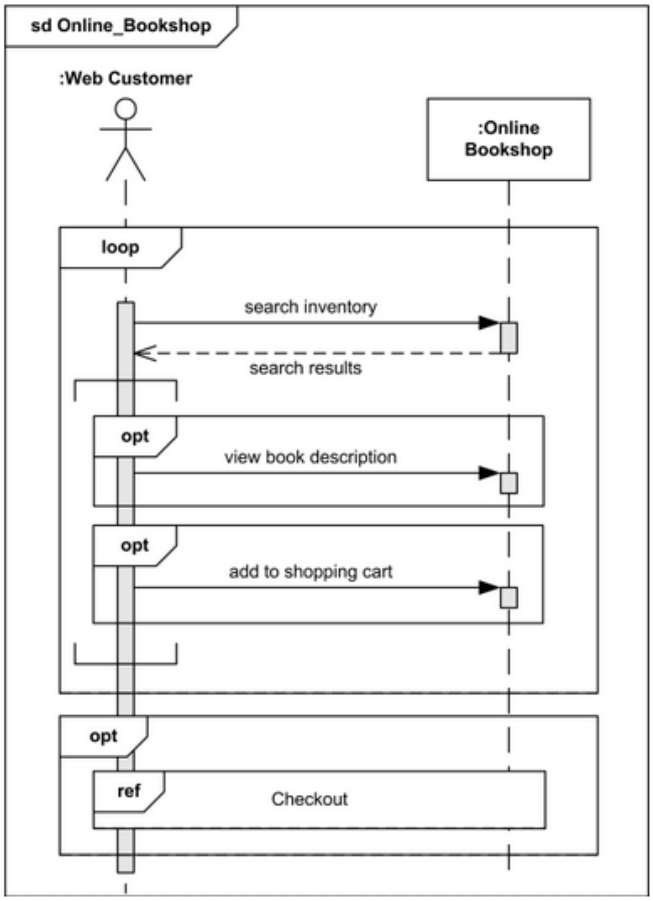
Use case diagram biasanya disebut sebagai *diagram* perilaku yang digunakan untuk menggambarkan serangkaian tindakan (*use case*) terhadap sistem yang harus atau dapat melakukan kolaborasi dengan satu atau lebih pengguna dari luar sistem atau *aktor* [20]. Gambar 2.9 merupakan contoh dari *use case diagram* sistem *check-in* di bandara.



Gambar 2. 9 Contoh *Use Case Diagram Check-in* di Bandara

2. *Sequence Diagram*

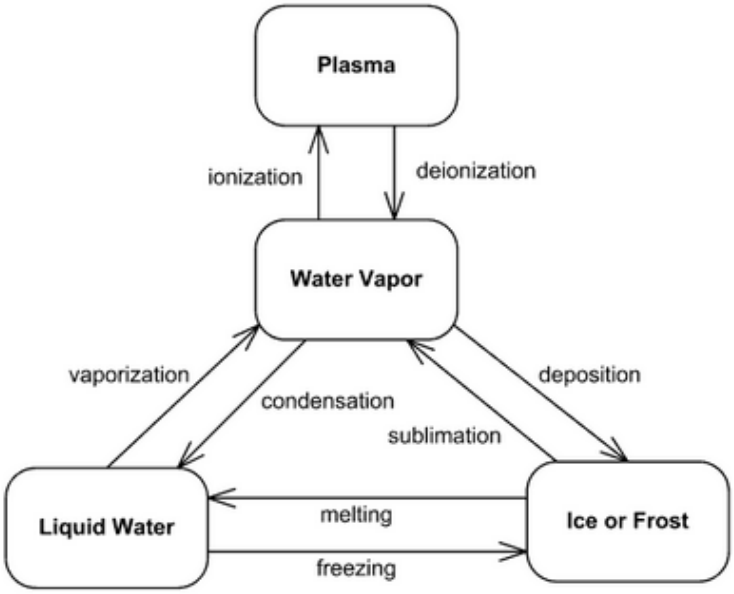
Sequence diagram adalah jenis yang paling umum dari diagram interaksi yang berfokus pada pertukaran pesan antara sejumlah objek [20]. Pada Gambar 2.10 dapat dilihat contoh *sequence diagram online bookshop*.



Gambar 2. 10 Contoh *Sequence Diagram Online Bookshop*

3. ***State Diagram***

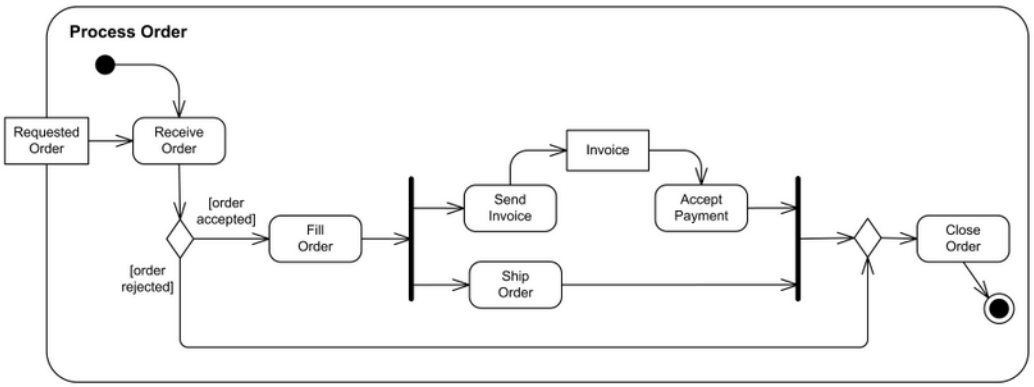
Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa objek lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda. Untuk lebih jelasnya dapat di lihat pada Gambar 2.11.



Gambar 2. 11 Contoh *State Diagram* Fase Air

4. *Activity Diagram*

Activity diagram adalah *diagram* UML yang menunjukkan aliran objek dengan penekanan pada urutan dan kondisi aliran. Tindakan dikoordinasi oleh model kegiatan dapat dimulai karena tindakan lainnya selesai dilaksanakan [20]. Berikut adalah contoh *activity diagram* proses order yang ditunjukkan pada Gambar 2.12.

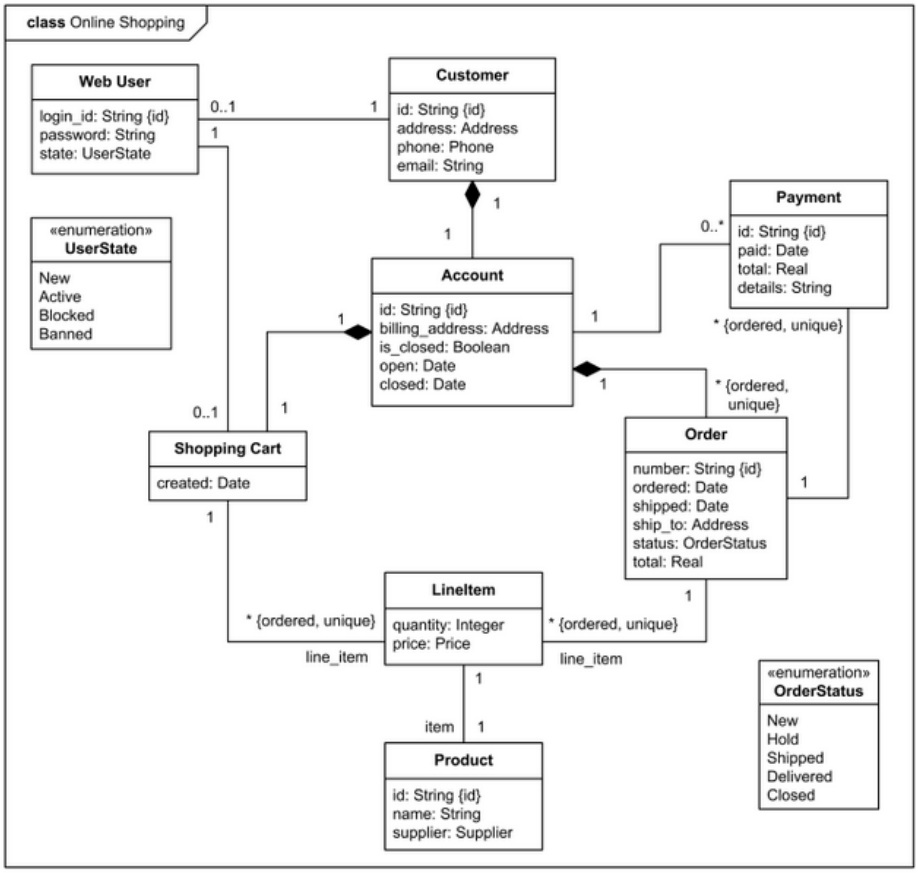


Gambar 2. 12 Contoh *Activity Diagram* Proses Order

5. *Class Diagram*

Class diagram adalah dekripsi kelompok objek dengan properti, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class diagram* dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari

class-class yang ada dan relasinya satu dengan yang lainnya [20]. Sebuah sistem biasanya mempunyai beberapa *class*. Pada Gambar 2.13 dapat dilihat contoh *class diagram online shopping domain*.



Gambar 2. 13 Contoh *Class Diagram Online Shopping Domain*

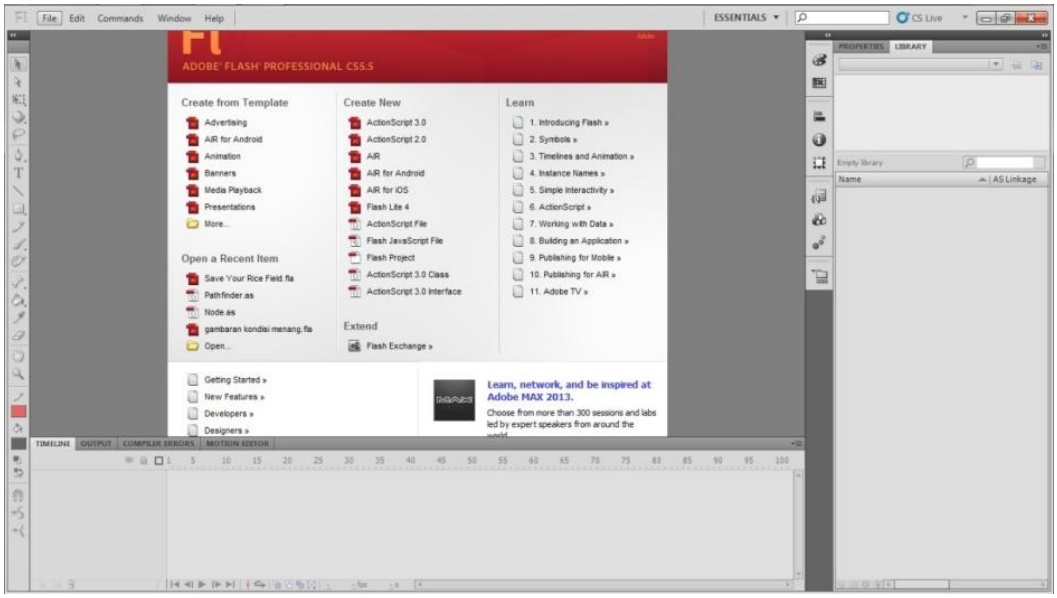
2.8 *Tools* yang digunakan

2.8.1 *Adobe Flash CS5 Professional*

Adobe Flash merupakan salah satu perangkat lunak komputer yang merupakan produk unggulan *Adobe System*. Berkas yang dihasilkan dari perangkat lunak ini mempunya *file extension .swf* dan dapat diputar di *browser web* yang telah di *install Adobe Flash Player*. *Flash* menggunakan bahasa pemrograman yang bernama *ActionScript* yang muncul pertama kalinya pada Flash 5.

Pada saat *Adobe Flash* pertama kali dibuka Secara *default*, *Adobe Flash* menampilkan *Start Page*. Terdapat beberapa pilihan antara lain mengedit *project*,

membuat dokumen *Flash*/*file ActionScript* atau membuat dokumen baru dari *template* yang sudah ada. *Start Page* pada *Adobe Flash CS5 Professional* dapat dilhat pada Gambar 2.14.



Gambar 2. 14 *Start Page Adobe Flash CS5 Professional*

Jendela utama merupakan awal dari pembuatan program, pembuatannya dilakukan dalam kotak *movie* dan *stage* yang didukung oleh *tools* lainnya. Seperti yang pernah dijelaskan dalam sebuah tulisan Jendela kerja *flash* terdiri dari *stage* dan panel-panel. *Stage* merupakan tempat objek diletakkan, tempat menggambar dan menganimasikan objek. Sedangkan panel disediakan untuk membuat gambar, mengedit gambar, menganimasi, dan pengeditan lainnya.

2.8.2 *ActionScript*

ActionScript merupakan bahasa pemrograman berorientasi objek (OOP) yang spesifikasi bahasanya bebasis ECMA *Script* dan ditunjukan untuk berjalan di atas sebuah *runtime environment* (serperti *Flash Player*, *Adobe AIR* atau *Flash Lite*. *Actionscript*) merupakan sebuah program yang mampu berinteraksi dengan bahsa pemrograman lain seperti Javascript, PHP, MySQL, karena pemrograman *actionscript* mempunyai sifat *case sensitive* [18].

Kelebihan terbesar dari *flash* adalah interkativitas terhadap suatu objek. Dalam membangun interaktivitas tersebut digunakan perintah-perintah maupun program yang disebut *actionscript*. Secara umum, *actionscript* pada *flash* hampir sama dengan pemrograman *javascript*. Jadi jika sudah terbiasa menggunakan bahasa pemrograman *javascript*, hal itu akan sangat membantu dalam membuat aplikasi-aplikasi yang ada pada *actionscript*.

Dalam membangun suatu interaktivitas, ada tiga komponen yang penting dalam *actionsctipt* yaitu:

1. *Event* (Kejadian)

   Event merupakan peristiwa atau kejadian untuk mendapatkan aksi sebuah objek. Beberapa *event* pada *actionscript 3.0* adalah sebagai berikut:

   a. ***Mouse Event*** : *event* yang berkaitan dengan penggunaan *mouse*, diantaranya *mouseclick*, *mouseover*, *mousedown*, *mousemove*.
   b. ***Keyboard Event*** : *event* yang berkaitan dengan *keyboard*, seperti *input.*
   c. ***Frame Event*** : *event* yang diletakkan pada *keyframe*.
   d. ***MovieClip Event*** : *event* yang berkaitan dengan *movieclip*.

2. *Action* (Aksi)

   Action merupakan aksi atau kerja yang dikenakan atau diberikan pada suatu objek untuk pembuatan interkasi antar objek. Beberapa *action* dalam *actionscipt 3.0* adalah sebagai berikut:

   a. ***Action Frame*** : *Action frame* adalah *action* yang diberikan pada *keyframe*. *Keyframe* yang telah diberikan *action* biasanya akan ditandai dengan huruf *a*.
   b. ***Action Object*** : *Action* pada objek diberikan bertujuan untuk membuat objek dapat bergerak yang berupa *MovieClip* atau merespon aksi dari *mouse* yang berupa *button*.

3. Target

   Target adalah objek yang dikenai oleh aksi atau perintah. Untuk dapat dikenai aksi atau dalam pemberian aksi, sebuat objek harus diberi label atau memilki nama inisialisasi.