BAB 2
LANDASAN TEORI

1.1 Kota Bandung

Dalam bagian ini akan dibahas tentang sejarah dan *landmark* Kota Bandung.

1.1.1 Sejarah Kota Bandung

Kota Bandung tidak berdiri bersamaan dengan pembentukan Kabupaten Bandung. Kota itu dibangun dengan tenggang waktu sangat jauh setelah Kabupaten Bandung berdiri. Kabupaten Bandung dibentuk pada sekitar pertengahan abad ke-17 Masehi, dengan Bupati pertama tumenggung Wiraangunangun. Beliau memerintah Kabupaten bandung hingga tahun 1681.

Semula Kabupaten Bandung beribukota di Krapyak (sekarang Dayeuhkolot) kira-kira 11 kilometer ke arah Selatan dari pusat kota Bandung sekarang. Ketika kabupaten Bandung dipimpin oleh bupati ke-6, yakni R.A Wiranatakusumah II (1794-1829) yang dijuluki "Dalem Kaum I", kekuasaan di Nusantara beralih dari Kompeni ke Pemerintahan Hindia Belanda, dengan gubernur jenderal pertama Herman Willem Daendels (1808-1811).

Sekitar akhir tahun 1808/awal tahun 1809, bupati beserta sejumlah rakyatnya pindah dari Krapyak mendekali lahan bakal ibukota baru. Mula-mula bupati tinggal di Cikalintu (daerah Cipaganti), kemudian pindah ke Balubur Hilir, selanjutnya pindah lagi ke Kampur Bogor (Kebon Kawung, pada lahan Gedung Pakuan sekarang).

Tidak diketahui secara pasti, berapa lama Kota Bandung dibangun. Akan tetapi, kota itu dibangun bukan atas prakarsa Daendels, melainkan atas prakarsa Bupati Bandung, bahkan pembangunan kota itu langsung dipimpin oleh bupati. Dengan kata lain, Bupati R. A. Wiranatakusumah II adalah pendiri (the founding father) kota

Bandung. Kota Bandung diresmikan sebagai ibukota baru Kabupaten Bandung dengan surat keputusan tanggal 25 September 1810 [**3**].

1.2 *Landmark* Kota Bandung

Setiap kota memiliki identitas kota (*landmark*) sendiri yang berbeda satu dengan lainnya, baik yang berskala regional, nasional, maupun internasional. Bandung memiliki banyak *landmark* [**4**], antara lain :

1. Gedung Sate
2. Monumen Perjuangan
3. Museum Geologi Bandung
4. Jalan layang Pasupati
5. Institut Teknologi Bandung
6. Villa Isola
7. Mesjid Raya Bandung
8. Gedung Merdeka
9. Savoy Homann Bidakara Hotel
10. Monumen Bandung Lautan Api

1.3 Edukasi

Edukasi atau pendidikan ialah suatu usaha yang sadar yang teratur dan sitematis, yang dilakukan oleh orang – orang yang diserahi tanggung jawab untuk mempengaruhi anak agar mempunyai sifat dan tabiat yang sesuai dengan cita – cita pendidikan. Edukasi juga bisa diartikan sebagai bantuan yang diberikan secara sengaja kepada anak dalam pertumbuhan jasmani maupun rohani.

Secara teoritis pengertian mendidik dan mengajar tidaklah sama. Mengajar berarti menyerahkan atau manyampaikan ilmu pengetahuan atau keterampilan dan lain sebagainya kepada orang lain, dengan menggunakan cara–cara tertentu sehingga ilmu–ilmu tersebut bisa menjadi milik orang lain.

Lain halnya mendidik, bahwa mendidik tidak hanya cukup dengan hanya memberikan ilmu pengetahuan ataupun keterampilan, melainkan juga harus

ditanamkan pada anak didik nilai–nilai dan norma–norma susila yang tinggi dan luhur. Dari pengertian di atas diketahui bahwa mendidik lebih luas dari pada mengajar. Mengajar hanyalah alat atau sarana dalam mendidik .dan mendidik harus mempunyai tujuan dan nilai–nilai yang tinggi [**5**].

1.4 Unsur – unsur Edukasi

Unsur – unsur edukasi ada enam [**6**]:

1. Anak didik, yaitu pihak yang menjadi objek utama edukasi.
2. Pendidik, yaitu pihak yang menjadi subjek dari pelaksanaan edukasi.
3. Materi, yaitu bahan atau pengalaman belajar yang disusun menjadi kurikulum.
4. Alat edukasi, yaitu tindakan yang menjadi kelangsungan mendidik.
5. Lingkungan, yaitu keadaan yang berpengaruh terhadap hasil edukasi.
6. Dasar dan landasan edukasi, yaitu landasan yang menjadi fundamental dari segala kegiatan edukasi.

1.4.1 Faktor – faktor yang membatasi kemampuan edukasi

Adapun faktor-faktor yang membatasi kemampuan edukasi diantaranya [**6**]:

1. Faktor anak

   Di dalam anak didik terdapat potensi-potensi yang butuh edukasi dari luar
2. Faktor pendidik

   Guru mempunyai metode penyampaian yang berbeda dan beragam
3. Faktor lingkungan

   Lingkungan sangat berpengaruh baik positif maupun negatif

1.5 Game

Dalam bagian ini akan dibahas pengertian *game,* klasifikasi *game* berdasarkan platform maupun berdasarkan *genre*, serta *game* edukasi.

1.5.1 Pengertian *Game*

Istilah *"video game"* awalnya mengacu pada jenis spesifik perangkat, yaitu sebuah sistem komputer yang menciptakan sinyal tampilan video untuk televisi, tapi kini istilah tersebut telah menjadi istilah yang mencakup segala jenis perangkat [**7**].

Video game dikendalikan oleh komputer melalui interaksi pengguna dan imersi audio-visual, *video game* memungkinkan pemain untuk menanggapi situasi yang menantang dalam dunia fantasi. Sangat penting untuk dicatat bahwa "fantasi" merujuk pada fakta bahwa dunia yang dimaksud adalah maya, dan tidak nyata, misalnya pemain dapat menjelajahi fantasi bermain di liga utama bisbol.

1.5.2 Klasifikasi *Game*

Game bisa diklasifikasikan kedalam beberapa hal, diantaranya klasifikasi berdasarkan *platform*, dan berdasarkan *genre*.

1.5.2.1 Berdasarkan Platform

Tiga kategori utama platform pada *game* adalah *personal computer (PC), console*, dan *mobile* [**7**].

1. *Personal Computer (PC)*

   PC game adalah *game* yang dibuat untuk computer baik berbasis Windows, Mac, ataupun Linux. *PC* menyediakan kekuatan grafis dan pemrosesan yang kuat yang memungkinkan pengembang untuk membuat *game* yang mutakhir. Tetapi kelemahan *game PC* adalah mahal karena pengguna harus menggunakan hardware yang *up-to-date* untuk memainkan *game PC* dengan baik. Selain itu, kelemahan *game PC* bagi pengembang adalah banyaknya varian dari konfigurasi *PC* membuat sulit bagi pengembang untuk memastikan *game* tersebut berjalan dengan benar pada semua *setup PC*.

2. *Console*

   Console adalah *hardware* yang dibuat oleh pihak ketiga seperti Sony, Microsoft, dan Nintendo. *Console* terhubung ke televisi dan tujuan

utamanya adalah untuk bermain *game*. *Game Console* sangat menarik bagi *game* pengembang karena mereka hanya perlu memikirkan satu konfigurasi *hardware* ketika membuat *software* untuk konsol. Sangat kontras dengan *PC* yang memiliki opsi konfigurasi yang tak terbatas.

3. *Mobile*

Mobile platform terdiri atas sesuatu yang *portable* dan bisa digenggam, termasuk ponsel, PDA, iPods, dan *handheld game* seperti Nintendo DSi atau Sony PSP. *Game mobile* memiliki kontrol yang sederhana (terutama jika dibandingkan dengan *PC*).

1.5.2.2 Berdasarkan Genre

Menurut Ernest Adams [**8**], genre *game* bisa dibagi menjadi 9, yaitu:

1. *Action*

Action game adalah *game* dimana kebanyakan dari tantangan yang disajikan merupakan dari tes *physical skill* dan koordinasi pemain. Salah satu sub-genre *action game* adalah *shooters game*, baik yang 2D maupun 3D seperti *First Person Shooters* (FPS).

2. *Strategy*

Strategy game menantang pemain untuk mencapai kemenangan dengan perencanaan, khususnya melalui perencanaan serangkaian tindakan yang dilakukan melawan satu lawan atau lebih. Kemenangan diraih dengan perencanaan matang dan pengambilan keputusan yang optimal.

3. *Role Playing Game (RPG)*

RPG adalah *game* dimana pemain mengontrol satu atau lebih karakter yang biasanya di desain oleh pemain itu sendiri, dan memandu mereka melewati berbagai rintangan yang diatur oleh komputer. Perkembangan karakter dalam hal kekuatan dan kemampuannya adalah kunci dari *game* jenis ini, salah satu contoh *game* jenis ini adalah Dota.

4. *Sports*

Sports game mensimulasikan berbagai aspek dari olahraga atletik nyata maupun imajiner, apakah itu memainkan pertandingan, me-*manage* tim dan karir, atau keduanya. Salah satu contoh *game* jenis ini adalah FIFA 13, dimana pemain bisa memainkan pertandingan, menjadi manajer tim, maupun menjadi pemain dan mengembangkan karirnya sendiri.

5. *Vechicle Simulation*

Vechicle simulation membuat feeling mengendarai kendaraan, baik kendaraan nyata maupun kendaraan imajiner. Performa dan karakteristik kasar mesin harus menyerupai kenyataan, kecuali jika yang didesain adalah kendaraan imajiner. Salah satu contoh *game* jenis ini adalah *Gran Turismo 5*.

6. *Construction and Management Simulation*

CMS game adalah *game* tentang proses. Tujuan pemain bukan untuk mengalahkan musuh, tetapi membangun sesuatu dengan konteks proses yang sedang berjalan. Semakin pemain mengerti dan mengontrol proses, semakin sukses sesuatu yang ia bangun. *Game* seperti ini biasanya menyediakan dua jenis permainan, yaitu mode bebas dimana pemain bebas membangun sesuatu, dan mode misi dimana terdapat skenario hal apa yang harus dibangun oleh pemain.

7. *Adventure*

Adventure game adalah cerita interaktif tentang karakter protagonis yang dimainkan oleh pemain. Penyampaian cerita dan eksplorasi adalah elemen inti dari *game* ini. Penyelesaian teka-teki dan tantangan konseptual adalah bagian besar dari permainan.

8. *Artificial Life and Puzzle Game*

Artificial Life game adalah *game* yang membuat tiruan dari kehidupan sebenarnya. Biasanya ada dua jenis *game* ini, tiruan kehidupan manusia, contohnya The SIMS.

9. *Online Game*

Istilah *online game* disini mengacu kepada *multiplayer game* dimana mesin dari para pemain terhubung dengan jaringan.

1.5.3 *Game* Edukasi

Game edukasi adalah *game* digital yang dirancang untuk pengayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interaktif [**9**].

Kriteria perancangan *game* edukasi yang ideal dibagi menjadi enam *point*, yaitu :

1. Rasa ingin tahu, fantasi dan kontrol
    a. Motivasi intrinsik, menyenangkan untuk digunakan. Pengguna memiliki kontrol terhadap permainan.
    b. Integrasi antara materi edukasi dan aspek fantasi serta permainan
    c. Pengguna memegang kendali permainan dan dapat mengatur kecepatan bermain.
    d. Mendorong keingintahuan, pengguna dapat melakukan eksplorasi bebas. Permainan mengandung rahasia tersembunyi.
    e. Pengguna dapat menyimpan kemajuan permainan.
    f. Menyediakan lebih dari jalur. Ada beberapa cara untuk memenangkan permainan.
    g. Pengguna dapat melakukan *trial-error* untuk menyelesaikan permainan.
    h. Keberhasilan ditentukan oleh pengetahuan, bukan kebetulan.
    i. Task disajikan secara *incremental* dan bertahap.
    j. Simulasi realistik dunia.

k. Materi edukasi disesuaikan dengan materi dunia nyata.

l. Personalisasi, yaitu pengguna dapat mengubah karakter dan objek-objek.

m. Menyediakan banyak pilihan dan tema.

n. Hindari pengulangan, hindari *drill-practice*.

o. Sediakan elemen tidak terduga dan kejutan.

p. Poin tidak hilang saat menjawab pertanyaan.

2. Tantangan

a. Kinerja pemain terukur dan didefinisikan dengan baik.

b. Tantangan diberikan secara terus menerus tapi disesuaikan dengan tingkat pemahaman pemain.

c. Kompleksitas semakin berkembang sejalan dengan kemampuan pengguna. Tersedia banyak level.

d. Memonitor kinerja pengguna, berdasarkan hasil *monitoring* ini kesulitan kemudian disesuaikan.

e. Pengguna dapat melihat *progress*nya setiap saat.

f. Menyediakan *hint* dan instruksi untuk membantu pengguna.

g. Bantuk pengguna menemukan '*flow state'* yaitu keseimbangan antara tantangan dengan kemampuan untuk menghadapi tantangan.

3. Sosialisasi

a. Pemain dapat berkolaborasi bersama.

b. Pemain dapat saling berkompetisi.

c. Pemenang bisa lebih dari satu (*multiple winner*)

4. Pedegogi
    a. Menyebutkan secara eksplisit target umur dan menyesuaikan rancangan permainan sesuai umur tersebut.
    b. Aktivitas pembelajaran dilaksanaan bersamaan dengan permainan.
    c. Objektif pembelajaran jelas.
    d. Memberi kesempatan kepada guru dan orangtua untuk ikut berperan.
    e. Memberikan petunjuk yang jelas sehingga pemain dapat berkonsentrasi kepada isi permainan, bukan bagaimana cara menggunakan permainan.
    f. Menyediakan tutorial sehingga pemain tidak perlu membaca manual.
5. Teknologi
    a. Menggunakan teknologi yang tersedia di sekolah dan masyarakat umum (tidak membutuhkan *hardware* yang terlalu tinggi)
    b. Memberikan lisensi yang terjangkau sekolah.
    c. Menggunakan antarmuka yang intuitif.
6. Pengguna anak-anak dan yang berkebutuhan khusus
    a. Memberikan petunjuk yang dibacakan selain petunjuk tertulis.
    b. Gambar, objek, layar tertata rapi.
    c. Permainannya sendiri merupakan aktivitas yang penting (*play for the sake of play*)

d. Transformasi terlihat. Saat anak melakukan interaksi, ada *instant feedback*.

e. *Input* dan *output* yang mudah terlihat.

f. Tantangan diberikan secara *gradual*.

g. Permainan menyenangkan untuk diulang, beberapa karakter, musik mudah teringat oleh anak.

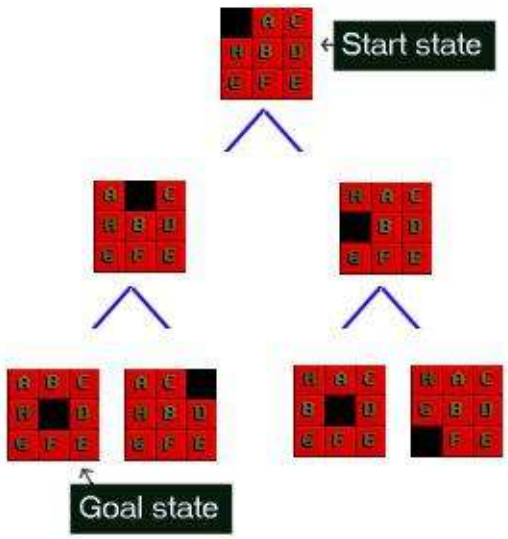
h. Objek dan suara diambil dari kehidupan sehari-hari yang mudah dikenali.

i. Menginspirasi anak, bahkan setelah komputer dimatikan.

1.6 Algoritma A Star

Algoritma A Star (A*) merupakan salah satu jenis algoritma pencarian [**10**].. A Star adalah algoritma pencarian graf yang mencari jalur dari satu titik awal ke sebuah titik akhir yang telah ditentukan. Algoritma A* menggunakan pendekatan heuristik h(x) yang memberikan peringkat ke tiap-tiap titik x dengan cara memperkirakan rute terbaik yang dapat dilalui dari titik tersebut. Setelah itu tiap-tiap titk x tersebut dicek satu-persatu berdasarkan urutan yang dibuat dengan pendekatan heuristik tersebut. Maka dari itulah algoritma A* adalah contoh dari best-first search. Algoritma ini pertama kali ditemukan pada tahun 1968 oleh Peter Hart, Nils Nilsson dan Bertram Raphael. Dalam tulisan mereka, algoritma ini dinamakan algoritma A. Penggunaan algoritma ini dengan fungsi heuristik yang tepat dapat memberikan hasil yang optimal, maka algoritma inipun disebut A*. Beberapa terminologi dasar yang terdapat pada algoritma ini adalah starting point, simpul (*nodes*), A, *open list*, *closed list*, harga (*cost*), halangan (*unwalkable*).

Salah satu penggunaan algoritma A Star dalam menyelesainkan *sliding puzzle* 3*3 dapat dilihat pada gambar 2.1.



Gambar 2.1 Hasil penggunaan A Star [**11**].

Gambar 2.2 menunjukkan algoritma A Star secara garis besar.



```
function A*(start,goal)
    closedset := the empty set    // The set of nodes already evaluated.
    openset := {start}    // The set of tentative nodes to be evaluated, initially containing the start node
    came_from := the empty map    // The map of navigated nodes.

    g_score[start] := 0    // Cost from start along best known path.
    // Estimated total cost from start to goal through y.
    f_score[start] := g_score[start] + heuristic_cost_estimate(start, goal)

    while openset is not empty
        current := the node in openset having the lowest f_score[] value
        if current = goal
            return reconstruct_path(came_from, goal)

        remove current from openset
        add current to closedset
        for each neighbor in neighbor_nodes(current)
            tentative_g_score := g_score[current] + dist_between(current,neighbor)
            if neighbor in closedset
                if tentative_g_score >= g_score[neighbor]
                    continue

            if neighbor not in openset or tentative_g_score < g_score[neighbor]
                came_from[neighbor] := current
                g_score[neighbor] := tentative_g_score
                f_score[neighbor] := g_score[neighbor] + heuristic_cost_estimate(neighbor, goal)
                if neighbor not in openset
                    add neighbor to openset

    return failure

function reconstruct_path(came_from, current_node)
    if current_node in came_from
        p := reconstruct_path(came_from, came_from[current_node])
        return (p + current_node)
    else
        return current_node
```

Gambar 2.2 Algoritma A Star.

1.7 Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei [**12**]. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti:

1. Sangat tidak setuju
2. Tidak setuju
3. Netral
4. Setuju

5. Sangat setuju

Selain pilihan dengan lima skala seperti contoh di atas, kadang digunakan juga skala dengan tujuh atau sembilan tingkat. Suatu studi empiris menemukan bahwa beberapa karakteristik statistik hasil kuesioner dengan berbagai jumlah pilihan tersebut ternyata sangat mirip.

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan "netral" tak tersedia.

1.8 Black Box Testing

Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangk at lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses *database* eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

1.9 Tools

Pada sub bab ini akan dijelaskan *tools* yang digunakan dalam pembangunan *game* edukasi *adventure* teori musik ini.

1.9.1 Object Oriented Paradigm

Konsep dasar dari paradigma berorientasi objek (cara melakukan sesuatu) relatif mudah untuk dipahami dan diterapkan. Alan Kay, penemu Smalltalk, telah mengerjakan *'Personal Computer for Children of all Ages*' sejak 1968, sebagai sasarannya adalah anak-anak. Tidak mengherankan bahwa konsep dasarnya sederhana.

Pendekatan berorientasi objek diciptakan (atau lebih tepatnya, berkembang) karena kesulitan orang – orang yang telah berusaha untuk mendapatkan sistem berkualitas baik, dihasilkan tepat waktu, dan sesuai anggaran, terutama untuk sistem yang besar dengan banyak orang yang terlibat.

Beberapa justifikasi yang biasanya diberikan kepada orientasi objek :

1. Objek lebih mudah dipahami masyarakat
2. Spesialis dapat berkomunikasi lebih baik.
3. Data dan proses tidak dipisahkan secara artifisial.
4. Kode dapat digunakan kembali lebih mudah.
5. Orientasi Obyek matang dan terbukti dengan baik.

1.9.2 Unified Modelling Language (UML)

Pada pertengahan 1990-an, metodologi yang paling terkenal adalah yang ditemukan oleh Ivar Jacobson, James dan Grady Booch. Masing-masing memiliki perusahaan konsultan sendiri, menggunakan metodologi sendiri dan notasi sendiri. Pada tahun 1996, Jacobson dan Rumbaugh bergabung Rasional Corporation (didirikan oleh Booch), dan mereka mengembangkan satu set notasi yang dikenal sebagai *Unified Modeling Language* (UML) [**13**].

UML memiliki 13 jenis diagram. Spesifikasi UML tidak mengatakan di mana diagram harus digunakan dalam metodologi tertentu.

1. *Use Case Diagram* mengkategorikan cara di mana sistem yang digunakan.
2. *Class Diagram* menunjukkan kelas dan bagaimana mereka dapat dipasang bersama-sama (mereka juga dapat menunjukkan *object*).

3. *Object Diagram* memperlihatkan obyek saja dan bagaimana mereka dapat dipasang bersama-sama.
4. *Activity Diagram* menunjukkan aktivitas oleh manusia atau objek dalam cara yang mirip dengan *Flow Chart*.
5. *State Machine Diagram* menunjukkan berbagai state setiap obyek dengan siklus hidup yang menarik atau rumit.
6. *Communication Diagram* menunjukkan pesan yang dikirim antara objek dalam skenario tertentu.
7. *Sequence Diagram* menunjukkan informasi yang sama dengan *communication diagram*, tetapi menekankan urutan daripada koneksi.
8. *Package Diagram* menunjukkan bagaimana kelas terkait dikelompokkan bersama-sama, untuk kepentingan pengembang.
9. *Deployment Diagram* menunjukkan mesin, proses dan artefak dikerahkan untuk jadi seorang sistem.
10. *Component Diagram* menunjukkan komponen dapat digunakan kembali (benda atau subsistem) dan *interface* mereka.
11. *Interaction Overview Diagram* menunjukkan langkah-langkah individu dari suatu kegiatan menggunakan *sequence diagram*.
12. *Timing Diagram* menunjukkan kendala waktu yang tepat untuk pesan dan menyatakan objek.
13. *Composite Structure Diagram* menunjukkan bagaimana benda cocok bersama dalam suatu agregasi atau komposisi, menampilkan antarmuka dan berkolaborasi benda.

1.9.3 ActionScript 3.0

ActionScript 3.0 adalah adalah bahasa berorientasi objek untuk membuat aplikasi dan konten multimedia ber-*script* untuk pemutaran di *runtime* Flash *client* (seperti Flash Player dan Adobe AIR). Dengan sintaks yang mirip dengan Java dan C#, bahasa ActionScript seharusnya akrab bagi programmer berpengalaman [**14**].

Beberapa fitur kunci dari bahasa ActionScript 3.0 adalah :

1. *First-class support* untuk *constructs* berorientasi objek umum, seperti kelas, objek, dan interface
2. *Single-threaded execution model*
3. *Runtime type-checking*
4. *Compile-time type-checking* opsional
5. Fitur dinamis seperti *runtime* pembuatan fungsi konstruktor baru dan variable
6. *Runtime exceptions*
7. *Direct support* untuk CML sebagai tipe data *built-in*
8. *Package* untuk pengorganisasian *code libraries*
9. *Namespaces* untuk kualifikasi *identifier*
10. *Regular expressions*