

## MPI Component Change log

//0.34 Feb 2023

: fungsi dasar MPI component, dokumentasi dalam format PDF, sampel penggunaan kode dalam FLA

Kode MPI Component	Fungsi	Contoh Penggunaan
<b>NAVIGASI DASAR</b>		
click(ins, opt1, opt2=null)	Mengatur tombol ketika di klik <b>ins</b> : instance name tombol <b>opt1</b> : frame tujuan / nama fungsi / nama label <b>opt2</b> : parameter untuk fungsi opt1	//pindah menuju frame tertentu mpi.click("tombolHome", 10); //pindah menuju frame dengan label mpi.click("tombolHome", "beranda"); //menjalankan fungsi tanyaKeluar mpi.click("tombolKeluar", tanyaKeluar); //pindah ke frame tertentu dan memainkan timeline mpi.click("tombolMulai", 10, "play"); //mengirim parameter ke fungsi mpi.click("tombolSubmit", submitScore, 80);
setTransisi(tp)	Menambahkan efek transisi halaman  <b>tp</b> : tipe transisi	//mematikan transisi mpi.setTransisi(""); //transisi efek fade mpi.setTransisi("fade"); //transisi efek wipe mpi.setTransisi("wipe"); //transisi efek dengan movieclip untuk transisi //movieclip memiliki linkage dan transisi tepat ditengah durasi movieclip mpi.setTransisi("mc", "linkageMC"); //movieclip memiliki linkage dan transisi terjadi pada frame tertentu, misal frame 20 mpi.setTransisi("mc", "linkageMC", 20);
geser(mcN, num, efek)	Menggeser konten movieclip sebesar num. Dapat digunakan untuk sistem galeri/photoslide, materi yang terdiri dari beberapa halaman dsb <b>mcN</b> : instance name movieclip <b>num</b> : jumlah pergeseran <b>efek</b> : efek pergeseran	//menggeser movieclip 1 langkah mpi.geser("kontenMC"); //menggeser movieclip 2 langkah mpi.geser("kontenMC", 2); //menggeser movieclip 1 langkah dengan efek fade mpi.geser("kontenMC", 1, "fade"); //menggeser movieclip 1 langkah dengan efek slide mpi.geser("kontenMC", 1, "slide");  //contoh dengan 2 tombol untuk menggeser galeri mpi.click("nextG", geserGaleri, 1); mpi.click("prevG", geserGaleri, -1); galeri.stop();  function geserGaleri(num):void{ mpi.geser("galeri", num, "slide"); }

<b>AUDIO dan VIDEO</b>		
suara(snd)	Menambahkan suara <b>snd</b> : linkage suara	//memainkan suara dengan linkage "bgm" 1x mpi.suara("bgm"); //memainkan suara dengan linkage "bgm" loop mpi.suara("bgm", "loop"); //menghentikan suara "bgm" mpi.suara("bgm", "stop"); //memainkan suara dengan volume tertentu mpi.suara("bgm", "play", 75);
suaraTombol(ins, snd)	Mengatur suara yang dikeluarkan oleh tombol ketika diklik <b>ins</b> : instance name tombol <b>snd</b> : linkage suara	//untuk suara per tombol mpi.suaraTombol("tombolHome", "suara1"); //untuk semua tombol memiliki suara yang sama mpi.suaraTombol("semua", "suaraKlik");
aturSuara(mcN)	Mengatur on off suara, membutuhkan movieclip dengan 2 frame. Frame 1 : on dan frame 2: off <b>mcN</b> : instance name movieclip	//onOff suara mpi.atuSuara("onOffMC");
video(mcN)	Mengatur video yang telah diimport ke stage. Suara aktif akan mati sesaat saat video dimainkan dan akan kembali setelah pindah dari frame video. <b>mcN</b> : instance name video	mpi.video("videoKu");
<b>FUNGSI GLOBAL</b>		
acak(num):int	Menghasilkan bilangan acak antara 0 - num	//acak dadu var acakDadu = 1+ mpi.acak(6);
acakArray(arr):Array	Mengacak isi array	//mengacak soal var soalTemp = acakArray(soal);
arah(kec, sudut)	Mendapatkan kecepatan pergeseran berdasarkan sudut tertentu.  Menghasilkan objek x dan y	var arahBola = mpi.arah(10, 45); mpi.gerak("bolaMC", gerakBola);  function gerakBola(){ bolaMC.x+=arahBola.x; bolaMC.y+=arahBola.y; }
bukaSWF(swflink, nfunc=null)	Membuka file SWF  <b>swflink</b> : nama file SWF yang akan dibuka <b>nfunc</b> : fungsi yang dipanggil setelah file terbuka dan selesai dijalankan	mpi.bukaSWF("intro.swf", selesaiIntro);  function selesaiIntro(){ gotoAndStop(2); }

bukaWeb(weblink)	Membuka link website	<pre> mpi.click("tombolInstagram", buka); function buka() {     mpi.bukaWeb("www.instagram.com/wandah_w"); } </pre>
clone(src:MovieClip)	Menduplikasi movieclip yang berada di stage dan sudah memiliki linkage  <b>src</b> : linkage movieclip	<pre> //menduplikasi MC gelembung yang sudah ada di stage  var gel = mpi.clone("gelembungMC"); gel.x = 200; gel.y = 300; </pre>
dataWeb(weblink, nfunc=null)	Membuka data dari link online. Misalnya digunakan untuk berkomunikasi dengan PHP untuk mendapatkan data tertentu  <b>Weblink</b> : link website yang dipanggil <b>Nfunc</b> : fungsi setelah mendapat respon dari website	<pre> mpi.dataWeb("www.wandah.org/submitScore.php", submitBerhasil);  function submitBerhasil(){     mpi.popUp("popupMC", "Submit Score", "nilai telah tersimpan"); } </pre>
drag(mcN, nfunc=null, limit="")	Drag movieclip  <b>mcN</b> : instance name movieclip <b>nfunc</b> : fungsi setelah mouse dilepas <b>limit</b> : batasan area drag	<pre> mpi.drag("mcA", pasang);  function pasang() {     if (mpi.jarak("mcA", "mcB")&lt;20) {         mcA.x = mcB.x;         mcA.y = mcB.y;     }else{         mpi.resetMC("mcA");     } }  //membatasi area drag hanya bisa horizontal mpi.drag("mcA", pasang, "x"); //membatasi area drag hanya bisa vertikal mpi.drag("mcA", pasang, "y"); //membatasi area drag berdasarkan kordinat //struktur batasan kordinat "x1,y1,x2,y2" mpi.drag("mcA", pasang, "100,20,150,200"); </pre>
dist(x1, y1, x2, y2)	Menghitung jarak antara dua kordinat x1,y1,x2 dan y2	<pre> var jarakTitik = mpi.dist(0,100, 200, 300); </pre>
gerak(mcN, gfunc, delay = 1)	Fungsi yang akan dijalankan secara terus menerus selama movieclip ada di stage  <b>mcN</b> : instance name movieclip <b>nfunc</b> : fungsi yang dijalankan <b>delay</b> : waktu eksekusi, semakin tinggi semakin lambat.	<pre> mpi.gerak("bolaMC", gerakBola);  function gerakBola(){     bolaMC.x+=10; } </pre>

	Secara default fungsi akan dijalankan 30 kali per detik	
getColor(x:Number, y:Number)	Mengambil warna layar yang ada pada kordinat x dan y	mpi.log(mpi.getColor(100,200));
hits(ob1, ob2)	Mendeteksi tabrakan antara movieclip ob1 dan ob2. Sama seperti kode hitTestObject	if (mpi.hits("mc1", "mc2")) mpi.log("tabrakan");
hitxy(ob1, px, py)	Mendeteksi tabrakan antara movieclip ob1 dan kordinat px, py. Sama seperti kode hitTestPoint	if (mpi.hits("mc1", 200, 200)) mpi.log("tabrakan");
jarak(ob1, ob2)	Menghitung jarak antara 2 objek ob1 dan ob2	if (mpi.jarak("mcA", "mcB")<20){ mpi.log("menempel"); }
jarakSudut(s1, s2)	Menghitung jarak sudut tersempit antara 2 sudut s1 dan s2	mpi.jarakSudut(125, 75);
keluarLayar(mcN)	Kondisi apakah movieclip keluar dari layar  <b>mcN</b> : instance name movieclip	var arahBola = mpi.arah(10, 45); mpi.gerak("bolaMC", gerakBola);  function gerakBola(){ bolaMC.x+=arahBola.x; bolaMC.y+=arahBola.y; if (mpi.keluarLayar("bolaMC")) mpi.resetMC("bolaMC"); }
kuis(mcN, soal, smax, nfunc)	Menampilkan kuis pada sebuah movieclip.  <b>mcN</b> : instance name movieclip <b>soal</b> : variabel array soal <b>smax</b> : jumlah soal maksimal <b>nfunc</b> : fungsi setelah kuis  Gunakan movieclip dari MPI Component agar lebih mudah diimplementasikan	var soal:Array = [[berisi array soal]];  mpi.kuis("kuisMC", soal, 5, selesaiKuis);  function selesaiKuis(){ //untuk menampilkan popup, harus memiliki MC popup yang sudah memiliki linkage mpi.popup("popupMC", "Kuis Selesai", "Nilai anda "+kuisMC.score, kembali); }  //menambahkan opsi jawaban kuisMC.opsi = "A"; //"a", "1"  //mengatur jumlah jawaban (default = 4) kuisMC.jumlahJawaban = 3;  //soal gambar, berupa moviclip "gambarKuis" kuisMC.setGambar("gambarKuis");

		<pre>//menjawab dengan waktu maksimal 5 detik mpi.timer("timerKuis", 5); //menambahkan suara kuis pada kuis dengan suara yang memiliki linkage kuisMC.suaraSoal = ""; kuisMC.suaraJawab = ""; kuisMC.suaraBenar = "sound1,sound2"; kuisMC.suaraSalah = "insound2,insound3 "; kuisMC.suaraWaktuHabis = ""; kuisMC.suaraSoalHabis = "";</pre>
soalXML(str)	<p>Membuka soal untuk kuis dari file bertipe XML.</p> <p><b>Struktur xml</b></p> <pre>&lt;dataSoal&gt;   &lt;soal&gt;     &lt;tanya&gt;Siapakah penemu mesin uap ?&lt;/tanya&gt;     &lt;jawab1&gt;James Watt&lt;/jawab1&gt;      &lt;jawab2&gt;Davinci&lt;/jawab2&gt;     &lt;jawab3&gt;Issac Newton&lt;/jawab3&gt;     &lt;jawab4&gt;James bond&lt;/jawab4&gt;     &lt;gambar&gt;0&lt;/gambar&gt;   &lt;/soal&gt; &lt;/dataSoal&gt;</pre>	<pre>mpi.kuis("kuisMC", mpi.soalXML("dataSoal.xml"), 5, selesaiKuis);</pre>
kursor(mcN, hideK=true)	<p>Menjadikan movieclip mcN menjadi kursor</p> <p><b>mcN</b> : instance name movieclip yang akan dijadikan kursor  <b>hideK</b> : menyembunyikan kursor asli</p>	<pre>mpi.kursor("kursorMC");  //untuk mengembalikan kursor asli mpi.kursor("");</pre>
log(str)	<p>Sama dengan trace, menampilkan str pada output panel saat dijalankan. Digunakan untuk proses debug</p>	<pre>mpi.debug = true; //default untuk mengaktifkan log mpi.log("hallo Dunia");</pre>
login(mcN)	<p>Untuk menambahkan fitur pemain dan akan direkam ke database lokal</p> <p><b>mcN</b> : instance name movieclip</p>	<pre>mpi.login("formMC");  //setelah login, nama pemain dapat diakses Mpi.log(mpi.nama);</pre>

	Gunakan movieclip dari MPI Component agar lebih mudah diimplementasikan	
popup(mcN)	Menambahkan popup ke layar <b>mcN</b> : linkage movieclip	//menambahkan popup sederhana mpi.popup("popupMC"); //menambahkan popup dengan text judul dan isi, menggunakan template yang ada di library mpi.popup("popupMC", "Judul Popup", "Ini isi popup"); //menambahkan popup dengan fungsi tertentu ketika popup ditutup mpi.popup("popupMC", "Submit score", "Apakah akan mensubmit nilai?", submitScore); //menambahkan popup pada kordinat tertentu mpi.popup("popupMC", "", "", 200, 300); //menambahkan popup yang akan hilang secara otomatis setelah menunggu sesaat var mulaiMC = mpi.popup("popupMC"); mulaiMC.waktuTunggu = 2; //menambahkan popup yang akan hilang secara otomatis setelah menunggu sesaat lalu menjalankan fungsi tertentu var mulaiMC = mpi.popup("popupMC", "", "", mulaiPermainan); mulaiMC.waktuTunggu = 2;
hapusPopup()	Menghapus popup secara manual. Popup secara otomatis bisa terhapus dari layar, namun jika diperlukan dapat dihapus menggunakan kode berikut	mpi.hapusPopup();
puzzle(mcN, gambar, nfunc)	Membuat puzzle secara otomatis  <b>mcN</b> : instance name movieclip container untuk puzzle, di dalamnya sudah ada tombol acak  <b>gambar</b> : linkage gambar yang akan dijadikan puzzle <b>nfunc</b> : fungsi selesai puzzle  Puzzle membutuhkan movieclip kepingMC	mpi.puzzle("puzzleGame", "gambarPuzzle", selesaiPuzzle); mpi.timer("timerPuzzle", 60); //non aktifkan timer sebelum puzzle mulai diacak timerPuzzle.aktif = false;  function selesaiPuzzle(){ mpi.popup("popupMC", "Puzzle Selesai", "Anda berhasil menyelesaikan puzzle"); puzzleGame.acakBtn.visible = true; }
resetMC(mcN)	Merestet posisi sebuah movieclip. Movieclip akan kembali ke posisi awal yang direkam saat aplikasi dipublish.	mpi.resetMC("mcA");

	<b>mcN</b> : instance name movieclip	
rotasi(x1:int, y1:int, x2:int, y2:int)	Menghitung sudut tersempit di antara dua titik kordinat	var sudut = mpi.rotasi(powerMC.x, powerMC.y, sensorMC.x, sensorMC.y);
scroll(mcN, val = 100, cfunc = null)	Fitur scroll yang ditampilkan pada sebuah movieclip.  <b>mcN</b> : instance name movieclip <b>val</b> : nilai awal dari scroller <b>cfunc</b> : fungsi yang dijalankan saat update posisi scroll	//mengatur suara dengan movieclip scrollMC mpi.suara("bgm", "loop"); mpi.scroll("scrollMC", mpi.volumeSuara, aturVolume);  function aturVolume(){ mpi.suara("bgm", "loop", scrollMC.val); }
scrollMC(mcN, dir, val)	Untuk menggeser movieclip / mengaktifkan fungsi scroll pada movieclip. Movieclip di dalamnya harus memiliki instance name "kontenMC".  <b>mcN</b> : instance name movieclip <b>dir</b> : arah pergeseran "x" atau "y" <b>val</b> : nilai pergeseran	//fitur scroller movieclip "materiMC" secara vertikal mpi.scroll("scrollMC", 0, slideV);  function slideV(){ mpi.scrollMC("materiMC", "y", scrollMC.val); }
tambahMC(mcN, px = 0, py = 0)	Menambahkan movieclip dari library ke stage  <b>mcN</b> : linkage name movieclip <b>px</b> : kordinat x <b>py</b> : kordinat y  Beda dengan clone adalah, clone digunakan ketika movieclip sudah berada di stage.	var bola = mpi.tambahMC("bolaMC", 100, 200);
hapusMC(mcN)	Menghapus movieclip yang ada di stage	mpi.hapusMC("bola");
tampilkanTanggal(ins)	Menampilkan tanggal aktif pada sebuah dynamic text <b>ins</b> : instance name dynamic text	mpi.tampilkanTanggal("tanggalTxt"); //menampilkan tanggal secara terpisah tanggalTxt.text = mpi.hari+", "+mpi.tanggal+" "+mpi.bulan+" "+mpi.tahun;
tampilkanWaktu(ins)	Menampilkan waktu aktif pada sebuah dynamic text <b>ins</b> : instance name dynamic text	mpi.tampilkanWaktu("waktuTxt");
teks(txtN, txt, nfunc = null)	Menampilkan teks pada sebuah dynamic text dengan efek typewriter	mpi.teks("ketikTxt", "Untuk menampilkan teks dengan efek mesin ketik, buatlah sebuah dynamic text.

	<p><b>txtN</b> : instance name dynamic text</p> <p><b>txt</b> : teks yang akan ditampilkan pada dynamic text</p> <p><b>nfunc</b> : fungsi yang akan dijalankan setelah teks ditampilkan seutuhnya.</p>	<p>Tambahkah instance name, dan pastikan menekan opsi EMBED");</p> <p>//jika ingin menampilkan fungsi setelah efek teks selesai</p> <pre>nextBtn.visible = false; mpi.click("nextBtn", 1); mpi.teks("ketikTxt", "Silahkan tekan tombol next untuk kembali ke beranda", balikHome);</pre> <pre>function balikHome(){     nextBtn.visible = true; }</pre>
teksHTML(txtN, txt)	<p>Menampilkan teks pada sebuah dynamic text dengan format HTML</p> <p><b>txtN</b> : instance name dynamic text</p> <p><b>txt</b> : teks yang akan ditampilkan pada dynamic text</p> <p>Kode HTML yang dapat ditampilkan adalah : &lt;b&gt;, &lt;i&gt;, &lt;u&gt;, &lt;font&gt;, &lt;br&gt;, /n</p>	<pre>mpi.teksHTML("rumusTxt", "H&lt;sub&gt;2&lt;/sub&gt;O itu air &lt;sup&gt;tm&lt;/sup&gt;"); mpi.teksHTML("rumusTxt2", "H~2~SO~4~ memiliki ion^+1^");</pre>
timer(mcN, waktu=5, cfunc=null)	<p>Mengatur waktu mundur. Perlu ditampilkan pada movieclip yang didalamnya terdapat hitungan mundur secara frame by frame.</p> <p><b>mcN</b> : instance name movieclip</p> <p><b>waktu</b> : waktu dalam detik</p> <p><b>cfunc</b>: fungsi setelah waktu habis</p>	<pre>mpi.timer("timer1MC", 10, fungsiSelesaiTimer);</pre> <pre>function fungsiSelesaiTimer(){     mpi.popup("popupMC", "Peringatan", "Waktu telah habis"); }</pre>
<b>VIRTUAL LAB</b>		
baterai (mcN)	<p>Mengatur baterai</p> <p><b>mcN</b> : instance name movieclip</p> <p>Gunakan movieclip <b>vlab-bateraiMC</b> agar bekerja dengan baik</p>	<pre>mpi.baterai("bateraiMC"); bateraiMC.voltase = 1.5;</pre>
burner (mcN)	<p>Mengatur burner</p> <p><b>mcN</b> : instance name movieclip</p>	<pre>mpi.burner("burnerMC"); burnerMC.dragY = false; burnerMC.menyala = true;</pre>



	Gunakan movieclip <b>vlab-burnerMC</b> agar bekerja dengan baik	//mengatur api burnerMC.waktu_api = 3; burnerMC.radius = 100; burnerMC.kalor = 10; burnerMC.idApi = 1;
caliper(mcN, cfunc = null)	Mengatur burner  <b>mcN</b> : instance name movieclip  Gunakan movieclip <b>vlab-caliperMC</b> agar bekerja dengan baik	mpi.caliper("caliper");
diagram(mcN, tipe, dataSet)	Membuat diagram pada kontainer  <b>mcN</b> : instance name movieclip <b>tipe</b> : jenis diagram "batang" "histogram" "garis" "lingkaran" "radar"  <b>dataSet</b> : array data	var dataSet = []; dataSet[0] = ["Jan", 34]; dataSet[1] = ["Feb", 16]; dataSet[2] = ["Mar", 44]; dataSet[3] = ["Apr", 22]; dataSet[4] = ["Mei", 32]; dataSet[5] = ["Jun", 46]; dataSet[6] = ["Jul", 23];  mpi.diagram("diagramMC1", "batang", dataSet);  //mpi.skalaV = 5; //jumlah garis skala pada diagram //mpi.warnaDiagram="mono"; //untuk diagram batang agar hanya 1 warna //mpi.warna = [0xff00ff, kode warna, kode warna, dst] jika ingin mengubah warna diagram //mpi.outlineDiagram = 0; //untuk menentukan tebal outline; 0 berarti tanpa outline
dummy(mcN, dfunc = null)	Membuat objek yang belum didefinisikan kodenya  <b>mcN</b> : instance name movieclip <b>dfunc</b> : fungsi yang dijalankan dummy setiap saat	mpi.dummy("contohMC"); //secara otomatis akan menjadikan contohMC menjadi objek virtual lab.
garis(x1, y1, x2, y2, tbl = 1, clr = 0x000000)	Membuat garis  <b>x1, y1, x2, y2</b> : kordinat garis <b>tbl</b> : tebal garis <b>clr</b> : warna garis (hex)  jika ada kordinat kartesian di layar garis akan dihitung berdasarkan kordinat kartesian.	//tanpa kordinat kartesian mpi.garis(100,20, 300, 400, 2, 0xffff00);  //dengan kordinat mpi.koordinat("kordMC"); kordMC.gridW = 25; kordMC.huruf = 10; kordMC.skalaX = 5; kordMC.skalaY = 5; mpi.updateMC("kordMC");

	Jika tidak ada, perhitungan menggunakan kordinat layar.	<pre>//menggambar gelombang sinus for (var i=0; i&lt;380;i++){     mpi.garis(i/5, 40 * Math.sin(i/40), (i+1)/5, 40 * Math.sin((i+1)/40), 2, 0xffc000); }</pre>
gear(mcN, notch, rot=0, radO=10, radI=10)	<p>Membuat gear</p> <p><b>mcN</b> : instance name movieclip  <b>notch</b> : jumlah gigi  <b>rot</b> : rotasi gear  <b>radO</b> : radius luar  <b>radI</b> : radius dalam</p>	<pre>mpi.gear("gear1", 20); mpi.gear("gear2", 10,15);  mpi.gerak("gear1", putarGear);  function putarGear(){     gear1.rotation+=1;     gear2.rotation-=2; }</pre>
glass(mcN)	<p>Mengatur glass</p> <p><b>mcN</b> : instance name movieclip</p> <p>Gunakan movieclip <b>vlab-beakerMC</b> agar bekerja dengan baik</p>	<pre>mpi.glass("glass1"); //pengaturan default glass, dapat diubah sesuai kebutuhan glass1.volume = 0; glass1.volumeMaks = 100; glass1.suhu = 20; glass1.gravitasi = true; glass1.warna = "biru"; glass1.titikUap = 100; glass1.titikDidih = 100; glass1.kalorJenis = 1; glass1.drag = true; glass1.auto = true; glass1.tuang = false;</pre>
gelembung(mc, px=0, py=0)	<p>Menambahkan gelembung ke dalam gelas</p> <p><b>mcN</b> : instance name movieclip  <b>px</b> : kordinat x  <b>py</b> : kordinat y</p> <p>membutuhkan linkage <b>bubbleMC</b></p>	<pre>mpi.gelembung(glass, glass.elektroda1.x, glass.elektroda1.y); mpi.gelembung(glass, glass.elektroda2.x, glass.elektroda2.y);</pre>
hapusGambar()	Menghapus gambar yang ditambahkan ke layar melalui perintah garis, titik, panah dan sebagainya	mpi.hapusGambar();
kabel(mcN)	<p>Mengatur kabel</p> <p><b>mcN</b> : instance name movieclip</p>	<pre>mpi.kabel("kabelMC"); kabelMC.warna = "biru";</pre>

	<p>Gunakan movieclip <b>vlab-kabelMC</b> agar bekerja dengan baik</p> <p>Mebutuhkan simbol <b>vlab-tipeKabel</b> untuk opsi pengaturan kabel</p>	
lampu(mcN)	<p>Mengatur lampu</p> <p><b>mcN</b> : instance name movieclip</p> <p>Gunakan movieclip <b>vlab-lampuMC</b> agar bekerja dengan baik</p>	<pre>mpi.lampu("lampuMC"); lampuMC.voltaseAktif = 0; //mati lampuMC.voltaseAktif = 1; //menyala lampuMC.voltaseAktif = 2; //menyala terang</pre>
koordinat(mcN)	<p>Membuat kordinat kartesian pada movieclip</p> <p><b>mcN</b> : instance name movieclip</p> <p>di dalam movieclip harus terdapat instance name</p> <p><b>x1</b>: untuk kordinat x+  <b>x0</b>: untuk kordinat x-  <b>y1</b>: untuk kordinat y+  <b>y0</b>: untuk kordinat y-</p>	<pre>mpi.koordinat("kordMC"); kordMC.gridW = 30; kordMC.huruf = 10; kordMC.skalaX = 5; kordMC.skalaY = 5;  //kordMC.gridX = false; //kordMC.gridY = false; mpi.updateMC("kordMC");  //menambahkan garis pada kordinat mpi.garis(2,3, -3, -2, 2, 0xff0000);</pre>
panah(x1, y1, x2, y2, tbl = 1, clr = 0x000000)	<p>Membuat panah</p> <p><b>x1, y1, x2, y2</b>: kordinat panah  <b>tbl</b>: tebal panah  <b>clr</b>: warna panah (hex)</p> <p>jika ada kordinat kartesian di layar panah akan dihitung berdasarkan kordinat kartesian. Jika tidak ada, perhitungan menggunakan kordinat layar.</p>	<pre>mpi.koordinat("kordMC"); mpi.panah(1,1, -4,-5);</pre>
pegas(mcN, putaran=10, warna="hijau")	<p>Mengatur pegas, mengubah movieclip menjadi pegas.</p> <p><b>mcN</b> : instance name movieclip</p>	<pre>mpi.pegas("pegasMC"); mpi.gerak("pegasMC", gerakPegas); pegasMC.drag = false; //gerakkan pegas var speedY = 3; var posY = 0;  function gerakPegas(){     posY+=speedY;     if (posY&gt;50    posY&lt;-50) speedY*=-1;     pegasMC.scaleY = 1+posY/100; }</pre>

<pre>pipet(mcN, pfunc = null)</pre>	<p>Mengatur pipet</p> <p><b>mcN</b> : instance name movieclip  <b>pfunc</b>: fungsi saat pipet di drop</p> <p>Gunakan movieclip <b>vlab-pipetMC</b> agar bekerja dengan baik</p>	<pre>mpi.glass("glass1"); glass1.volume = 80; glass1.drag = false; glass1.jenisIsi = "NaOH";  mpi.glass("glass2"); glass2.volume = 20; glass2.volumeMaks = 30; glass2.warna = "merah"; glass2.drag = false; glass2.jenisIsi = "H2SO4"; mpi.updateMC("glass2");  mpi.pipet("pipet", cekPipet);  mpi.vlabLantai = 600;  function cekPipet(){     if (pipet.jenisIsi == "H2SO4" &amp;&amp; pipet.drop == glass1){         glass1.warna = "ungu";         mpi.updateMC("glass1");     } }</pre>
<pre>putar(mcN, val=0, nfunc=null, minL=0, maxL=360)</pre>	<p>Memutar movieclip digunakan untuk mengatur sudut, mengatur volume dsb</p> <p><b>mcN</b> : instance name movieclip  <b>val</b> : nilai awal sudut putaran  <b>nfunc</b> : fungsi yang dijalankan  <b>minL</b> : sudut minimal  <b>maxL</b> : sudut maksimal</p>	<pre>mpi.putar("knobMC", 0, updateNilai);  function updateNilai(){     nilai_txt.text = String(Math.round(knobMC.val/3)/10); }</pre>
<pre>rollMenu(mcN, nfunc=null, ndata=0)</pre>	<p>Membuat sistem roll menu</p> <p><b>mcN</b> : instance name movieclip  <b>nfunc</b> : fungsi yang dijalankan  <b>ndata</b> : nomor data yang dimunculkan pertama kali</p> <p>Gunakan movieclip <b>vlab-rollMenu</b></p>	<pre>rollMC.data1 = "Air (H~2~O)"; //karakter ~ untuk subscript rollMC.data2 = "Minyak"; rollMC.data3 = "Air Garam"; rollMC.data4 = "NaCL^+1^"; //karakter ^ untuk superscript mpi.rollMenu("rollMC", ubahIsi);  mpi.glass("glass1"); glass1.volume = 80; glass1.drag = false;  function ubahIsi(){     if (rollMC.pilihan == 1) glass1.warna = "biru";</pre>

		<pre>         if (rollMC.pilihan == 2) glass1.warna = "kuning";         if (rollMC.pilihan &gt;= 3) glass1.warna = "putih";         mpi.updateMC("glass1");     } </pre>
sempoa(mcN, sfunc = null)	<p>Membuat simulasi sempoa</p> <p><b>mcN</b> : instance name movieclip  <b>sfunc</b>: fungsi saat sempoa dioperasikan</p> <p>Gunakan movieclip <b>vlab-sempoaMC</b> agar bekerja dengan baik</p>	<pre> mpi.sempoa("sempoaMC", traceNilai);  function traceNilai(){     trace(sempoaMC.nilai); } </pre>
simulasi(nfunc = null) stopSimulasi() resetSimulasi()	<p>Perintah loop untuk menjalankan simulasi</p> <p><b>nfunc</b> : fungsi yang dijalankan</p>	<pre> mpi.click("btn_mulaiSimulasi", "start-vlab", rumusSuhu); mpi.click("btn_resetSimulasi", "reset-vlab");  function rumusSuhu(){     burnerMC.menyala = true;     //T = Q / (m.c)     var kenaikanSuhu = burnerMC.kalori / (beakerMC.volume * beakerMC.kalorJenis);     //jika api menyentuh gelas     if (beakerMC.sumberPanas == burnerMC){         //suhu berubah         beakerMC.suhu += kenaikanSuhu;     } } </pre>
slider(mcN, nfunc = null, val=0, vmin=0, vmax =100)	<p>Membuat slider untuk pengaturan objek lainnya</p> <p><b>mcN</b> : instance name movieclip  <b>nfunc</b>: fungsi saat slider digeser  <b>val</b> : nilai awal  <b>vmin</b> : nilai minimum  <b>vmax</b> : nilai maksimum</p> <p>Gunakan movieclip <b>vlab-slider</b> agar bekerja dengan baik</p>	<pre> mpi.slider("slider", aturSuhu); slider.nilai = 20; slider.kelipatan = 1; slider.desimal = 2; slider.nilaiMin = 0; slider.nilaiMax = 100;  mpi.termometer("termo1"); termo1.skala = "C"; termo1.auto = false;  function aturSuhu(){     termo1.suhu = slider.nilai; } </pre>
termometer(mcN)	<p>Mengatur termometer</p> <p><b>mcN</b> : instance name movieclip</p>	<pre> mpi.termometer("termo"); termo.skala = "F"; termo.auto = false; termo.skalaBawah = "0"; termo.skalaAtas = "300"; </pre>

	Gunakan movieclip <b>vlab-termometerMC</b> agar bekerja dengan baik	
titik(x1, y1, r = 5, tbl = 1, sclr = 0x000000, fclr = 0xff0000)	Membuat titik  <b>x1, y1</b> : kordinat titik <b>r</b> : radius titik <b>tbl</b> : tebal garis <b>sclr</b> : warna garis tepi(hex) <b>fclr</b> : warna isi(hex)  jika ada kordinat kartesian di layar garis akan dihitung berdasarkan kordinat kartesian. Jika tidak ada, perhitungan menggunakan kordinat layar.	mpi.koordinat("kordMC"); mpi.titik(1,1);
tripod(mcN)	Mengatur movieclip sebagai tripod	
txt(str, px=0, py=0, tw=100, th=100, tt="Arial", ta="center", te="normal")	Membuat teks  <b>str</b> : teks yang dimunculkan <b>px</b> : kordinat x <b>py</b> : kordinat y <b>tw</b> : lebar teks <b>th</b> : tinggi teks <b>tt</b> : jenis huruf <b>ta</b> : alignment <b>te</b> : efek huruf "bold", "italic"	var kata = mpi.txt("halo", 100, 100); addChild(kata);
uap(mc)	Menambahkan uap pada movieclip  <b>mcN</b> : instance name movieclip	mpi.uap("glass");
zoom(mcN, zval = 2)	Menampilkan movieclip zoom agar dapat melihat detail dengan pembesaran default 2 kali  <b>mcN</b> : instance name movieclip <b>zval</b> : nilai pembesaran  movieclip <b>vlab-zoomMC</b> harus dicopy ke project	mpi.zoom("zoom");  //atau melalui tombol mpi.click("zoomBtn", tampilkanZoom); function tampilkanZoom(){ mpi.tambahZoom("zoomMC"); }

MPI Component dapat digunakan secara umum (Public Domain) dengan mencantumkan credit wandah.org atau mensitasi buku Wandah Wibawanto.

Request penambahan fitur dapat dilakukan melalui email [wandah@wandah.com](mailto:wandah@wandah.com)